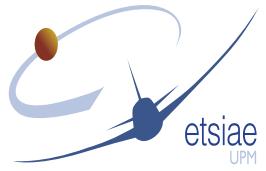




POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
AERONÁUTICA Y DEL ESPACIO

“ESTUDIO DE LA LOCOMOCIÓN DE UN
ROBOT CON PATAS COMO SISTEMA
DINÁMICO NO LINEAL”

TUTOR: Dr. EZEQUIEL DEL RÍO FERNÁNDEZ
ÓSCAR RODRÍGUEZ FERNÁNDEZ
CIENCIAS Y TECNOLOGÍAS AEROESPACIALES
CURSO 2013/2014

Prólogo

Este documento pretende recoger todo lo relacionado con el proceso de elaboración del Trabajo de Fin de Grado sobre la locomoción de un robot con patas como sistema dinámico no lineal.

Querría expresar mi agradecimiento a mi familia y amigos, por su apoyo durante estos años de carrera, así como a mi tutor, Ezequiel del Río Fernández, por su ayuda durante la realización de este proyecto.

- Madrid, 19 de julio de 2014

Óscar Rodríguez Fernández

Índice general

Índice de Figuras	v
Índice de Tablas	ix
Índice de Códigos	xI
1. Introducción	1
1.1. Motivación	1
1.2. Modos de Movimiento en Insectos de 6 patas	2
1.3. Descripción de las señales de las patas	3
2. Utilización de los Anillos de Toda-Rayleigh como CPG	7
2.1. Mecánica del Anillo de Toda-Rayleigh	7
2.2. El circuito equivalente	8
2.3. El nuevo sistema de Toda-Rayleigh	11
2.4. Conclusiones y Objetivos del Proyecto	13
3. Definición geométrica del problema	15
3.1. Ligaduras geométricas	16
3.2. Cálculo de los momentos de inercia	16
3.2.1. Momento de inercia de las patas sin apoyar	16
3.2.2. Momento de inercia de las patas apoyadas	16
4. Dinámica del movimiento del robot	19
4.1. Introducción	19
4.2. Planteamiento de las ecuaciones	19
4.2.1. Ecuaciones para las patas apoyadas	20
4.2.2. Resolución del sistema de ecuaciones de las patas apoyadas	21
4.2.3. Ecuaciones para la(s) pata(s) en fase vuelo	23
4.2.4. Modelo de ecuaciones dinámicas a resolver	23
4.2.5. Transición entre la fase de vuelo y la fase de apoyo	23
4.3. Simulación numérica de la dinámica del robot	24
4.3.1. Funcionamiento del programa	24
4.3.2. Resultados de la simulación	27
4.3.3. Discusión de los resultados de la simulación	33
5. Dinámica del movimiento del robot utilizando el sistema de Toda-Rayleigh como Central Pattern Generator	35
5.1. Introducción	35

5.2. La función de Toda-Rayleigh y su sistema de control automático	37
5.3. Modificación de las ecuaciones	38
5.3.1. Transición entre la fase de vuelo y la fase de apoyo	42
5.4. Simulación numérica de la dinámica del robot	42
5.4.1. Funcionamiento del programa	42
5.4.2. Resultados de la simulación	46
5.5. Discusión de los resultados obtenidos en la simulación	60
6. Conclusiones y trabajos futuros	71
Apéndices	73
A. Código momentos mecánicos constantes	75
B. Código momentos proporcionados por Toda-Rayleigh	91
Bibliografía	115

Índice de Figuras

1.1.	Esquema de los tres modos de movimiento hexápodos. A) Nomenclatura utilizada. Las letras L y R representan los lados derecho (R) e izquierdo (L), mientras que los subíndices indican el número de pata. B) Modo <i>Metachronal</i> (baja velocidad). Los segmentos gruesos representan la fase de vuelo o de avance de la pata levantada, mientras que la línea discontinua se corresponde con la fase de apoyo. Tan sólo se levanta una de las patas a la vez. C) Modo <i>Caterpillar</i> (velocidad media). D) Modo <i>Tripod</i> (alta velocidad)	2
1.2.	Esquema simplificado de las patas del robot	3
1.3.	Ángulo de referencia para la generación de las señales del CPG	4
1.4.	Evolución temporal de los ángulos de las de los diferentes modos de movimiento hexápodos: A) <i>Metachronal</i> B) <i>Caterpillar</i> C) <i>Tripod</i>	5
2.1.	Anillo de Toda. A) 6 masas están acopladas formando un anillo unidas por "muelles exponenciales". B) Fuerza exponencial actuando entre cada par de masas acopladas.	7
2.2.	Esquema de bloques del anillo de Toda-Rayleigh. Cada nodo del anillo (limitado por la línea de puntos) se compone de dos bloques principales: un doble condensador (DC) y una resistencia no lineal (NR)	8
2.3.	Esquema de la implementación electrónica el anillo de Toda-Rayleigh	9
2.4.	Modos oscilatorios generados por el circuito de Toda-Rayleigh de 6 unidades y su relación con los modos hexápodos. La parte superior muestra el voltaje medido de cada una de las seis unidades mientras que la parte inferior muestra las correspondientes relaciones de fase. A) Modo $m = 1$, correspondiente al modo <i>Metachronal</i> . B) $m = 2$, correspondiente al modo <i>Caterpillar</i> . C) $m = 3$ modela el modo <i>Tripod</i>	10
2.5.	Esquema que muestra la sustitución del doble condensador (DC) por el circuito electromecánico que incluye la dinámica de la extremidad. Ambos bloques siguen ecuaciones equivalentes.	12
3.1.	Esquema simplificado de la geometría del robot	15
3.2.	Esquema del funcionamiento de la guía	17
4.1.	Momentos y fuerzas que actúan sobre el robot en un instante genérico . . .	19
4.2.	Condición de ligadura	20
4.3.	Diagrama de flujo del script principal del código de integración de momentos mecánicos constantes	25
4.4.	Diagrama de flujo del script dinamica_completa.m	26
4.5.	Ángulos de las extremidades en función del tiempo para el modo <i>Metachronal</i> .	28
4.6.	Posición del Centro de Masas en función del tiempo para el modo <i>Metachronal</i>	28

4.7.	Velocidad del Centro de Masas en función del tiempo para el modo <i>Metachronal</i>	29
4.8.	Ángulos de las extremidades en función del tiempo para el modo <i>Caterpillar</i>	30
4.9.	Posición del Centro de Masas en función del tiempo para el modo <i>Caterpillar</i>	30
4.10.	Velocidad del Centro de Masas en función del tiempo para el modo <i>Caterpillar</i>	30
4.11.	Ángulos de las extremidades en función del tiempo para el modo <i>Tripod</i>	31
4.12.	Posición del Centro de Masas en función del tiempo para el modo <i>Tripod</i>	32
4.13.	Velocidad del Centro de Masas en función del tiempo para el modo <i>Tripod</i>	32
5.1.	Funciones $\theta(t)$ resultantes de la primera aproximación al sistema de Toda-Rayleigh	36
5.2.	Función de Toda-Rayleigh $\frac{\Phi I_{dc}}{J}$	36
5.3.	Esquema de la implementación del control automático en el sistema electro-mecánico	37
5.4.	Momentos y fuerzas que actúan sobre el robot en un instante genérico	39
5.5.	Diagrama de flujo del script main.m	43
5.6.	Diagrama de flujo del script dinamica_robot.m	45
5.7.	Funciones $\theta(t)$ resultantes de la simulación del modo <i>Metachronal</i>	47
5.8.	Funciones $\dot{\theta}(t)$ resultantes de la simulación del modo <i>Metachronal</i>	48
5.9.	Funciones $\ddot{\theta}(t)$ resultantes de la simulación del modo <i>Metachronal</i>	48
5.10.	Estado de las patas en función del tiempo: 0 para la fase de apoyo o i para la fase de vuelo de la pata i para el modo <i>Metachronal</i>	49
5.11.	Momentos introducidos por el sistema de Toda-Rayleigh para el modo <i>Metachronal</i>	49
5.12.	Posición del centro de masas del robot para el modo de avance <i>Metachronal</i>	50
5.13.	Velocidad del centro de masas del robot para el modo de avance <i>Metachronal</i>	50
5.14.	Valor de V_{sh} para el modo <i>Metachronal</i>	51
5.15.	Funciones $\theta(t)$ resultantes de la simulación del modo <i>Caterpillar</i>	52
5.16.	Funciones $\dot{\theta}(t)$ resultantes de la simulación del modo <i>Caterpillar</i>	52
5.17.	Funciones $\ddot{\theta}(t)$ resultantes de la simulación del modo <i>Caterpillar</i>	53
5.18.	Estado de las patas en función del tiempo: 0 para la fase de apoyo o i para la fase de vuelo de la pata i para el modo <i>Caterpillar</i>	53
5.19.	Momentos introducidos por el sistema de Toda-Rayleigh para el modo <i>Caterpillar</i>	54
5.20.	Posición del centro de masas del robot para el modo de avance <i>Caterpillar</i>	54
5.21.	Velocidad del centro de masas del robot para el modo de avance <i>Caterpillar</i>	55
5.22.	Valor de V_{sh} para el modo <i>Caterpillar</i>	55
5.23.	Funciones $\theta(t)$ resultantes de la simulación del modo <i>Tripod</i>	56
5.24.	Funciones $\dot{\theta}(t)$ resultantes de la simulación del modo <i>Tripod</i>	57
5.25.	Funciones $\ddot{\theta}(t)$ resultantes de la simulación del modo <i>Tripod</i>	57
5.26.	Estado de las patas en función del tiempo: 0 para la fase de apoyo o i para la fase de vuelo de la pata i para el modo <i>Tripod</i>	58
5.27.	Momentos introducidos por el sistema de Toda-Rayleigh para el modo <i>Tripod</i>	58
5.28.	Posición del centro de masas del robot para el modo de avance <i>Tripod</i>	59
5.29.	Velocidad del centro de masas del robot para el modo de avance <i>Tripod</i>	59
5.30.	Valor de V_{sh} para el modo <i>Tripod</i>	60
5.31.	Fuerzas de rozamiento para el modo <i>Metachronal</i>	61
5.32.	Fuerzas de rozamiento para el modo <i>Caterpillar</i>	61
5.33.	Fuerzas de rozamiento para el modo <i>Tripod</i>	62

5.34. Comparación de las simulaciones para el modo <i>Metachronal</i>	63
5.35. Funciones $\theta(t)$ para el modo <i>Metachronal</i>	64
5.36. Funciones $\dot{\theta}(t)$ para el modo <i>Metachronal</i>	64
5.37. Posición del centro de masas para el modo <i>Metachronal</i>	64
5.38. Velocidad del centro de masas para el modo <i>Metachronal</i>	65
5.39. Comparación de las simulaciones para el modo <i>Caterpillar</i>	65
5.40. Funciones $\theta(t)$ para el modo <i>Caterpillar</i>	66
5.41. Funciones $\dot{\theta}(t)$ para el modo <i>Caterpillar</i>	66
5.42. Posición del centro de masas para el modo <i>Caterpillar</i>	66
5.43. Velocidad del centro de masas para el modo <i>Caterpillar</i>	67
5.45. Funciones $\theta(t)$ para el modo <i>Tripod</i>	67
5.44. Comparación de las simulaciones para el modo <i>Tripod</i>	68
5.46. Funciones $\dot{\theta}(t)$ para el modo <i>Tripod</i>	68
5.47. Posición del centro de masas para el modo <i>Tripod</i>	69
5.48. Velocidad del centro de masas para el modo <i>Tripod</i>	69

Índice de Tablas

4.1.	Parámetro físicos del problema	27
4.2.	Condiciones iniciales del modo <i>Metachronal</i>	27
4.3.	Secuencia de levantamiento del modo <i>Metachronal</i>	28
4.4.	Condiciones iniciales del modo <i>Caterpillar</i>	29
4.5.	Secuencia de levantamiento del modo <i>Caterpillar</i>	29
4.6.	Condiciones iniciales del modo <i>Tripod</i>	31
4.7.	Secuencia de levantamiento del modo <i>Tripod</i>	31
5.1.	Parámetros geométricos y másicos del robot	46
5.2.	Parámetros propios de la función de Toda-Rayleigh	46
5.3.	Condiciones iniciales en las posiciones de las extremidades para el modo <i>Metachronal</i>	47
5.4.	Condiciones iniciales en las velocidades angulares de las extremidades para el modo <i>Metachronal</i>	47
5.5.	Voltaje externo para estabilizar el modo <i>Metachronal</i>	47
5.6.	Condiciones iniciales en las posiciones de las extremidades para el modo <i>Caterpillar</i>	51
5.7.	Condiciones iniciales en las velocidades angulares de las extremidades para el modo <i>Caterpillar</i>	51
5.8.	Voltaje externo para estabilizar el modo <i>Caterpillar</i>	51
5.9.	Condiciones iniciales en las posiciones de las extremidades para el modo <i>Tripod</i>	56
5.10.	Condiciones iniciales en las velocidades angulares de las extremidades para el modo <i>Tripod</i>	56
5.11.	Voltaje externo para estabilizar el modo <i>Tripod</i>	56

Índice de Códigos

A.1. principal.m	75
A.2. parametros.m	81
A.3. dinamica_completa.m	82
A.4. RK4_din.m	85
A.5. dinamica_apoyadas.m	86
A.6. dinamica_levantadas.m	88
A.7. centrodemasas.m	89
B.1. main.m	91
B.2. setup.m	96
B.3. RK4.m	98
B.4. dinamica_robot.m	101
B.5. TODAg.m	106
B.6. derTODAg.m	108
B.7. RKcontrol.m	109
B.8. CONTROL.m	110
B.9. cinematica_CM.m	110

Capítulo 1

Introducción

1.1. Motivación

Recientemente, ha aparecido un creciente interés por entender la dinámica neuronal, los sistemas de procesamiento de la información y los sistemas de control que utilizan los seres vivos. Una diferencia importante con respecto a robots basados en control mediante el procesamiento digital de señales es que, en el caso digital, el sistema realiza millones de operaciones por segundo, mientras que las neuronas tan sólo realizan algunas decenas. Esto, unido al tiempo necesario para la comunicación entre neuronas alejadas entre sí, ha motivado la búsqueda de mecanismos de control en insectos basados en sistemas dinámicos.

La locomoción de los animales parece estar determinada por un Generador Central de Patrones, CPG por sus siglas en inglés, que es una red intra-espinal de neuronas capaces de generar la salida oscilatoria necesaria para el control de las extremidades. El estudio de este generador central de patrones es crucial para entender tanto el comportamiento animal como las funciones de estas redes neuronales. Además, también es importante para el diseño de robots inspirados en estos generadores de patrones que sean capaces de moverse eficientemente en casi todos los ambientes y terrenos.

El movimiento de las patas parece ser una parte intrínseca del generador de patrones más que simplemente un reflejo de su actividad [1]. La arquitectura de este CPG apenas es observable en vivo. Sin embargo, se pueden inferir aspectos importantes de su estructura a partir de la observación de ciertos aspectos de los modos de movimiento tales como la fase dentro del ciclo de movimiento en la cual cada extremidad se posa en el suelo. Hecho esto, se pueden reproducir modelos fenomenológicos utilizables en el diseño de robots.

De hecho, la mayoría de los modos de movimiento de los animales poseen un grado de simetría y unas características universales que no están alejadas del comportamiento de anillos de osciladores acoplados. Por ejemplo, se puede considerar un anillo de osciladores no lineales para reproducir el Generador Central de Patrones de Locomoción de insectos y otros animales [2–6]. Las transiciones entre los diferentes modos de caminar se podrían modelar cambiando entre los diferentes patrones de oscilación del anillo.

1.2. Modos de Movimiento en Insectos de 6 patas

A continuación se realiza una breve descripción de las simetrías observables en los modos de movimiento de un insecto de 6 patas (se pueden ver más detalles en [3]). Se utiliza la siguiente nomenclatura: las extremidades de izquierda y derecha están numeradas comenzando por la frontal y señalizadas con la letra L y R respectivamente (figura 1.1).

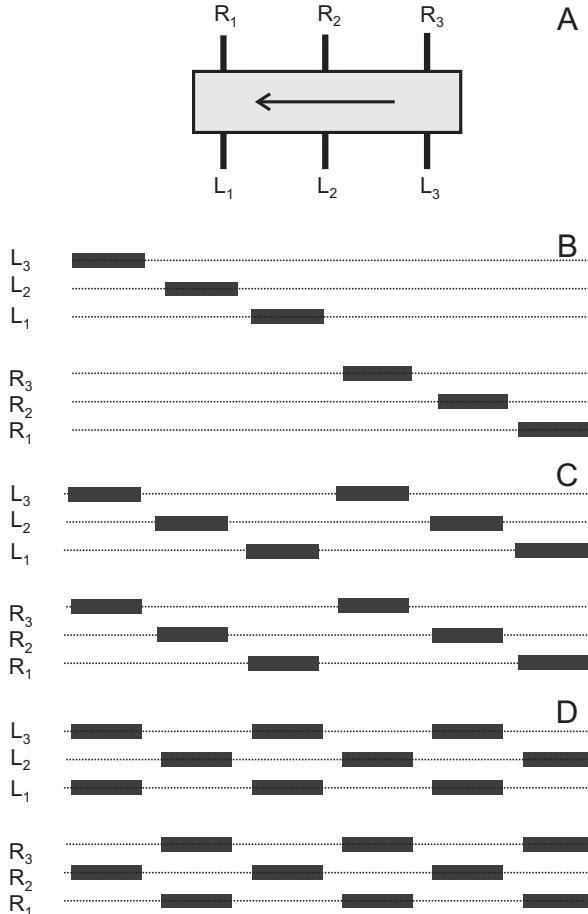


Figura 1.1: Esquema de los tres modos de movimiento hexápodos. A) Nomenclatura utilizada. Las letras L y R representan los lados derecho (R) e izquierdo (L), mientras que los subíndices indican el número de pata. B) Modo *Metachronal* (baja velocidad). Los segmentos gruesos representan la fase de vuelo o de avance de la pata levantada, mientras que la línea discontinua se corresponde con la fase de apoyo. Tan sólo se levanta una de las patas a la vez. C) Modo *Caterpillar* (velocidad media). D) Modo *Tripod* (alta velocidad).

Cuando un insecto se mueve lentamente, suele recurrir al modo llamado *Metachronal* (figura 1.1.B). Este modo puede describirse como una onda propagándose desde atrás hacia adelante (primero en el lado izquierdo y luego en el derecho) de acuerdo con el siguiente esquema:

$$L_3, L_2, L_1, R_3, R_2, R_1$$

En este modo, las extremidades adyacentes de cada segmento (R_1 y R_2 , R_2 y R_3) tienen un desfase de 60° . Las extremidades de cada segmento (L_1 y R_1) tienen entre sí un desfase de medio período o 180° .

El modo *Caterpillar* es un modo de velocidad media el que las extremidades de ambas mitades se mueven de forma síncrona (figura 1.1.C) de acuerdo con el esquema:

$$(L_3R_3), (L_2R_2), (L_1R_1)$$

Este modo es más frecuente en insectos del tipo ciempiés que en otro tipo de insectos.

Para desplazamientos relativamente más rápidos, los insectos adoptan el modo *Tri-pod* (figura 1.1.D). En este modo, la pata anterior y posterior de uno de los lados y la pata central del otro se mueven en fase conjuntamente. Las extremidades de ambas mitades tienen un desfase de 180° y las patas adyacentes de cada uno de los lados también tienen un desfase de 180° , moviéndose con el siguiente patrón:

$$(L_3L_1R_2), (L_2R_3R_1)$$

En el caso de insectos, y otros animales como crustáceos, los diferentes modos de caminar se van ajustando dinámicamente para adaptarse al ambiente. Por ejemplo, cuando un pequeño obstáculo se encuentra en el camino, la fase de vuelo puede cambiarse o hacerse más corta para asegurar la suavidad del movimiento del animal. Esta característica es uno de los mayores desafíos en el diseño de robots, ya que es necesario un ajuste en tiempo real de los ciclos del CPG, realimentándolo en tiempo real con información del ambiente.

1.3. Descripción de las señales de las patas

En la figura 1.2 se ha modelizado, de forma sencilla, las patas de un insecto palo con sus componentes principales: coxa (o muslo), femur, tibia y tarsus. Como demostraron Cruse y sus colaboradores [1, 13], a efectos de locomoción tan sólo son relevantes los tres grados de libertad observados en la figura 1.2: θ - extensión-retracción del muslo, α - elevación-bajada del femur, y β - flexión-extensión de la tibia.

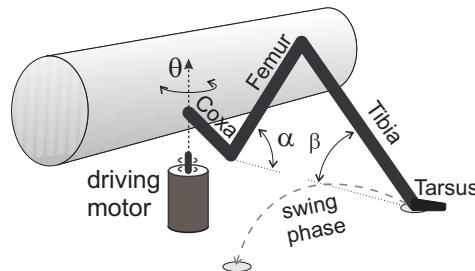


Figura 1.2: Esquema simplificado de las patas del robot

Cada una de las patas pueden encontrarse únicamente en una fase: fase de apoyo o fase de vuelo. Cuando las patas se encuentran en fase de apoyo, se puede conseguir el movimiento del robot mediante la rotación del ángulo θ con ayuda de un servomotor. Para mantener la pata fijada al suelo sin deslizar, el movimiento del muslo (ángulo θ) se acompaña con un ajuste continuo de los ángulos α y β . Cuando la extremidad se encuentra en la fase de vuelo ($\frac{d\theta}{dt} > 0$ en el caso de avance), el ángulo α se incrementa levantando la pata del suelo para volver a disminuir cuando se apoye la pata.

El ángulo θ determina el comportamiento de los ángulos α y β durante todas las fases de movimiento. Esto sugiere que sólo es necesario un CPG que controle la dinámica de los diferentes θ_n , generando los diferentes modos oscilatorios o modos de movimiento, para conseguir la locomoción del robot, puesto que el comportamiento de los demás ángulos queda determinado.

Utilizando como variable de control el ángulo θ que forman cada una de las patas con una referencia perpendicular al cuerpo del insecto (figura 1.3), se han realizado simulaciones de las diferentes señales que tendría que generar el CPG para reproducir los modos de caminar típicos de los insectos.

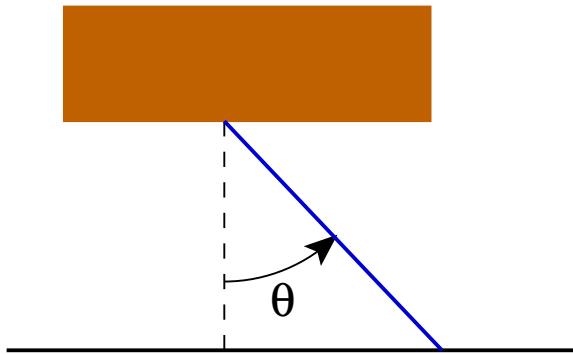
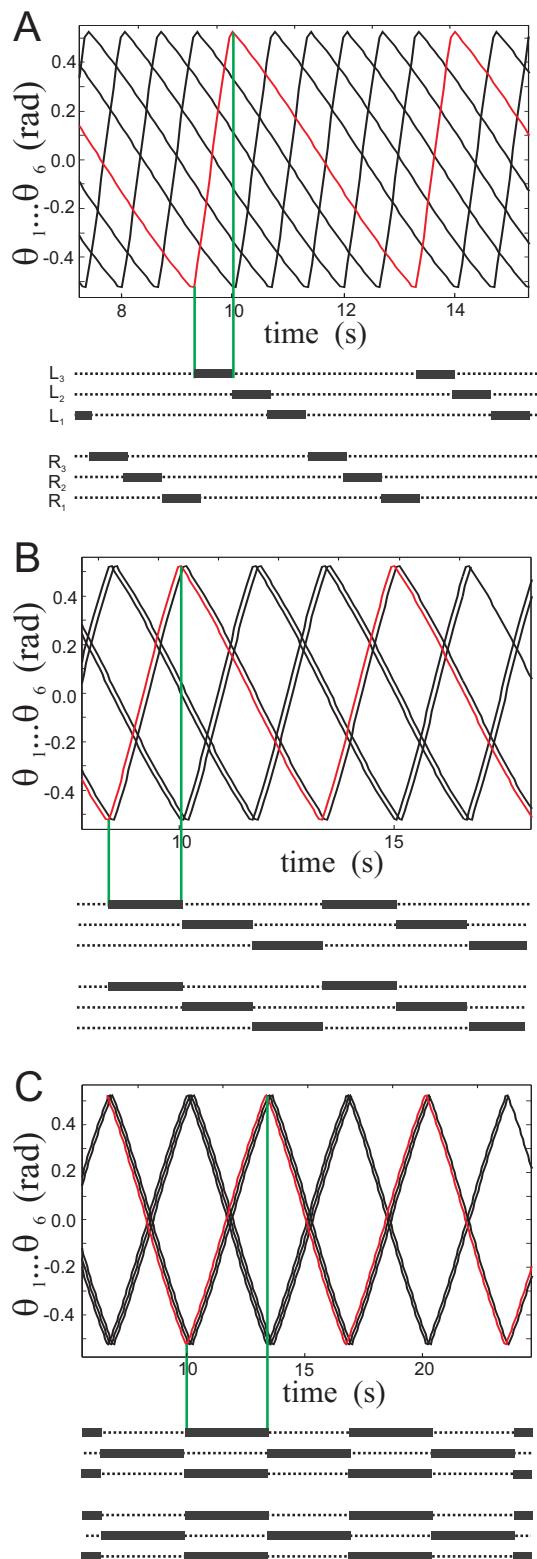


Figura 1.3: Ángulo de referencia para la generación de las señales del CPG

Las señales resultantes de estas simulaciones se pueden observar en la figura 1.4. Se puede observar como, durante las diferentes fases de vuelo, la pendiente de la señal correspondiente es positiva mientras que en las fases de apoyo esta es negativa. Esto es debido a que en la simulación se ha considerado el modo de avance del insecto.

Figura 1.4: Evolución temporal de los ángulos de las de los diferentes modos de movimiento hexápodos: A) *Metachronal* B) *Caterpillar* C) *Tripod*



Capítulo 2

Utilización de los Anillos de Toda-Rayleigh como CPG

Para modelar el comportamiento del CPG, se debe encontrar un sistema dinámico capaz de generar las señales observadas en la figura 1.4, correspondientes a los diferentes modos de movimiento. El sistema propuesto es un anillo de osciladores acoplados denominado anillo de Toda-Rayleigh.

2.1. Mecánica del Anillo de Toda-Rayleigh

Toda [10,11] proporciona las soluciones exactas para un sistema mecánico formando un anillo (y su circuito equivalente) (figura 2.1A). Seis masas se unen mediante muelles especiales cuya fuerza se incrementa exponencialmente con el decrecimiento de distancia entre masas unidas (figura 2.1.B). Bajo los límites adecuados (pequeña amplitud de las oscilaciones), la interacción exponencial de Toda se reduce al oscilador armónico.

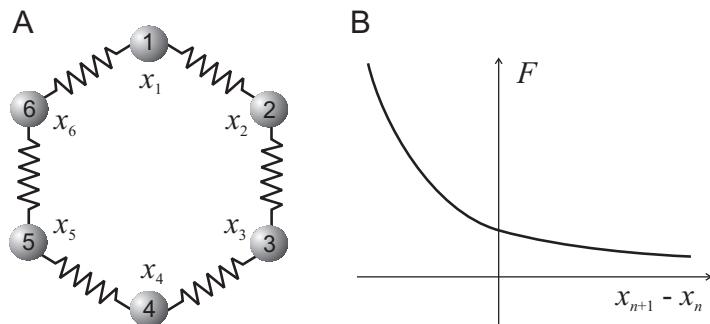


Figura 2.1: Anillo de Toda. A) 6 masas están acopladas formando un anillo unidas por "muelles exponenciales". B) Fuerza exponencial actuando entre cada par de masas acopladas.

Este sistema de Toda es conservativo. Sin embargo, su implementación como circuito electrónico o sistema mecánico conlleva inevitablemente unas pérdidas de energía [14]. Para poder mantener las oscilaciones, es necesario proveer energía al sistema. Un balance de energía-disipación fue propuesto por Lord Rayleigh [12]. Este incluye una no-linealidad cúbica que regula el balance de disipación-aporte [7,9].

En vista de esto, se obtiene un sistema compuesto conservativo-disipativo. La forma canónica del modelo de Toda-Rayleigh [9] está dada por:

$$\ddot{x}_n + \omega_0^2(e^{x_n-x_{n+1}} - e^{x_{n-1}-x_n}) - (\mu - \dot{x}_n^2)\dot{x}_n = 0 \quad (2.1)$$

donde ω_0 es la frecuencia de las oscilaciones lineales y μ es el parámetro de Rayleigh.

En el caso realmente amortiguado ($\mu < 0$), el sistema sólo tiene una solución globalmente estable ($x_{n+1} - x_n = 1$). Para $\mu = 0$, el sistema presenta una bifurcación simétrica de Hopf [9, 15] y para $\mu > 0$, el balance de energía sólo admite una serie de soluciones discretas. Aparecen en el sistema $N-1$ modos diferentes de oscilación (cinco, en el caso de un anillo de 6 masas). Estos modos se corresponden con los ciclos límite estables en el espacio de fases de $2N$ dimensiones del sistema. Estos modos representan ondas no lineales, similares a aquellas observadas en la figura 1.4, que se propagan por el anillo y que se pueden identificar en función del número de picos de la onda m . Para un anillo de 6 masas (figura 2.1) el modo $m = 1$ se corresponde con una onda de un sólo pico, es decir, su longitud es igual a la del anillo, $m = 2$ posee dos picos o longitud igual a la mitad de la del anillo mientras que $m = 3$ se corresponde con un modo tal que las masas vecinas se mueven en antifase. El signo del número de modo se corresponde con la dirección de propagación de la onda (horaria o antihoraria). Cabe destacar que, dado que cada modo se corresponde con un ciclo límite estable, sólo puede aparecer un único modo en el anillo al mismo tiempo sin permitir la superposición.

2.2. El circuito equivalente

El anillo de Toda-Rayleigh puede reproducirse mediante un circuito electrónico analógico. Una descripción detallada del mismo y de todos sus componentes puede encontrarse en [9]. Este circuito ha sido diseñado y construido en el laboratorio de Sistemas No Lineales del Departamento de Física Aplicada a la Ingeniería Aeronáutica de la Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio. En la figura 2.2 se muestra un esquema del mencionado circuito.

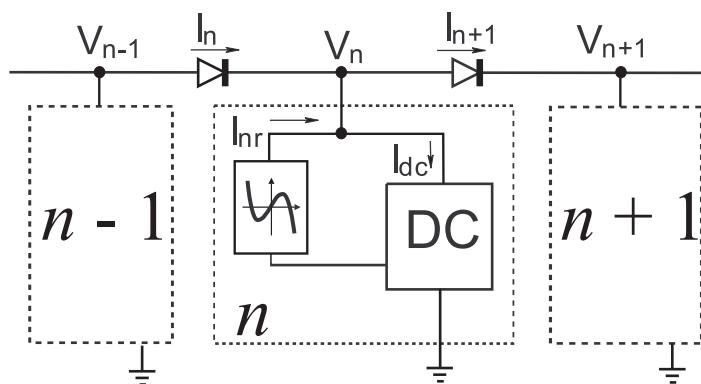


Figura 2.2: Esquema de bloques del anillo de Toda-Rayleigh. Cada nodo del anillo (limitado por la línea de puntos) se compone de dos bloques principales: un doble condensador (DC) y una resistencia no lineal (NR)

En [9] se puede ver una descripción detallada de los diferentes componentes del circuito y su funcionamiento, así como de la ecuación diferencial que permite obtener

el voltaje V_n en cada una de las unidades que componen el anillo.

Cada una de las unidades del anillo está compuesta por dos bloques principales: un doble condensador (DC) y una resistencia no lineal (NR). El nexo entre las diferentes unidades se realiza mediante diodos.

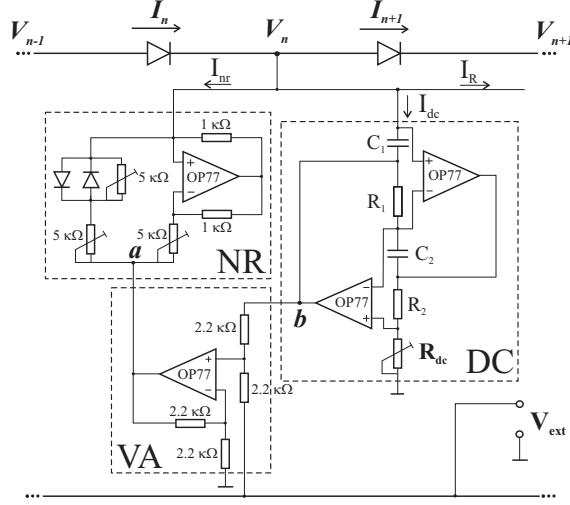


Figura 2.3: Esquema de la implementación electrónica el anillo de Toda-Rayleigh

De acuerdo con las características de un doble condensador se obtiene:

$$\frac{d^2V_n}{dt^2} = aI_{dc} \quad (2.2)$$

donde a es un parámetro que depende de los componentes internos del doble condensador. Aplicando las leyes de Kirchhoff para hallar el valor de I_{dc}

$$\frac{d^2V_n}{dt^2} = a(I_n - I_{n+1} + I_{nr}) \quad (2.3)$$

y teniendo en cuenta que se puede modelizar el comportamiento de los diodos según:

$$I_n = I_s e^{\frac{V_{n-1}-V_n}{V_t}} \quad (2.4)$$

y que la corriente a través de la resistencia no lineal es una función cúbica del voltaje aplicado a sus terminales ΔV de la forma

$$I_{nr}(\Delta V) = b(\tilde{V}^2 - \Delta V^2)\Delta V \quad (2.5)$$

$$\Delta V = \frac{1}{\omega_v^2 R_{dc} C_1} \frac{dV_n}{dt} - V_{ext} \quad (2.6)$$

donde b y \tilde{V} son constantes y V_{ext} es un voltaje de control externo que permite, mediante su ajuste, estabilizar los diferentes modos de oscilación, se llega a la ecuación diferencial 2.7 que determina el voltaje de la unidad n :

$$\frac{d^2V_n}{dt^2} = a\{I_s \left[e^{\frac{V_{n-1}-V_n}{V_t}} - e^{\frac{V_n-V_{n+1}}{V_t}} \right] + I_{nr}\} \quad (2.7)$$

Comparando 2.1 con 2.5, 2.6 y 2.7 se puede observar que, dinámicamente, el circuito descrito por los voltajes V_n es equivalente al sistema mecánico del anillo (V_n es equivalente a x_n).

Experimentalmente, se han medido las señales, en términos de voltaje, que produce este circuito equivalente para los diferentes modos de oscilación. Durante la realización de los experimentos, se ha comprobado cómo el sistema es muy robusto frente a perturbaciones, proporcionando siempre señales similares. Dichas señales se presentan a continuación:

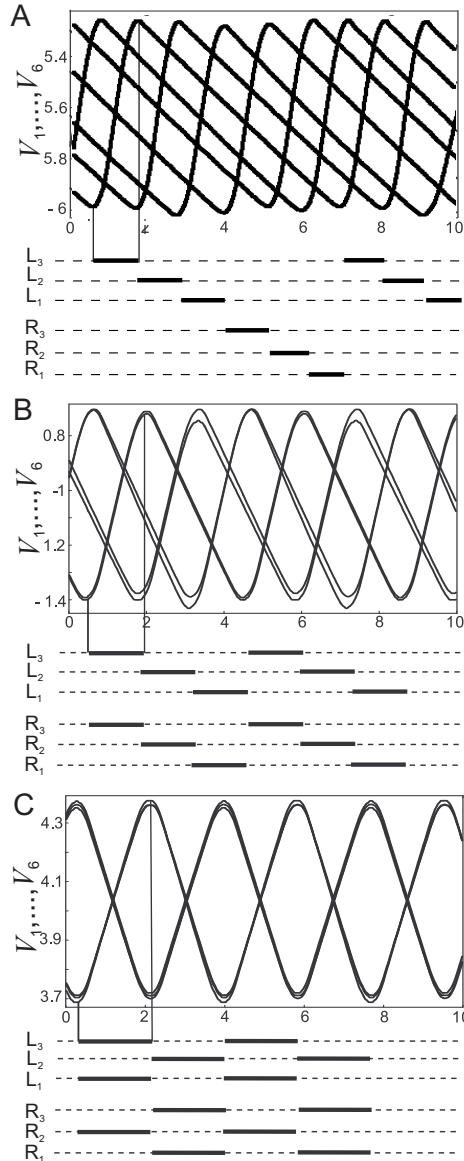


Figura 2.4: Modos oscilatorios generados por el circuito de Toda-Rayleigh de 6 unidades y su relación con los modos hexápodos. La parte superior muestra el voltaje medido de cada una de las seis unidades mientras que la parte inferior muestra las correspondientes relaciones de fase. A) Modo $m = 1$, correspondiente al modo *Metachronal*. B) $m = 2$, correspondiente al modo *Caterpillar*. C) $m = 3$ modela el modo *Tripod*.

Como se puede observar en la figura 2.4, las señales generadas por el sistema de Toda-Rayleigh se corresponden aproximadamente con las necesarias para generar los diferentes modos de caminar de los insectos (figura 1.4). Podemos asociar cada una de las extremidades del robot con una unidad del anillo de Toda-Rayleigh a través de un motor. Cada motor gira de acuerdo con la señal de voltaje de la correspondiente unidad del sistema, transformando el voltaje en un ángulo.

$$V_n \rightarrow \theta_n \quad (2.8)$$

En el modo de avance considerado, cuando la derivada del voltaje sea positiva, la extremidad se encontrará en la fase de vuelo, estando en la fase de apoyo cuando la derivada sea negativa. La coordinación entre cada una de las patas se conseguirá debido al acoplamiento entre cada una de las unidades del anillo. Esto conduce a la conclusión de que se puede utilizar el sistema de Toda-Rayleigh para asumir las funciones del CPG en la locomoción del robot. Sin embargo, el sistema planteado produciría siempre las mismas señales de movimiento, sin tener en cuenta las fuerzas externas que puedan influir sobre las patas o el cuerpo del robot. En la siguiente sección se propone una implementación electromecánica del sistema de Toda-Rayleigh que incluye las extremidades del robot.

2.3. El nuevo sistema de Toda-Rayleigh

Se propone ahora una nueva implementación del anillo de Toda-Rayleigh donde se incluye el estado del robot como una variable. De esta forma se consigue un CPG cuya dinámica dependa del entorno, pudiendo variar las características del movimiento para adaptarse al mismo. Esto se consigue sustituyendo el doble condensador del circuito por los motores que controlan el movimiento de las extremidades, tal y como se explica a continuación.

En el caso más general, la dinámica de un motor sigue la siguiente ecuación:

$$J \frac{d^2\theta}{dt^2} + f \frac{d\theta}{dt} = I\Phi - T \quad (2.9)$$

donde θ es el ángulo de rotación del motor y, por lo tanto, se corresponde con el ángulo de la extremidad correspondiente θ_n , $I\Phi$ es el momento electromagnético, T es un momento constante que incluye la fricción de Coulomb interna, el parámetro f es debido a la fricción viscosa interna, que puede tomarse como una constante y J es el momento de inercia, que dependerá de la distribución geométrica de masas y suponemos constante. La corriente que circula por el devanado del motor I está dada por:

$$\ell \frac{dI}{dt} + rI + \Phi \frac{d\theta}{dt} = V_{in}(t) \quad (2.10)$$

donde V_{in} es el voltaje de entrada, ϕ el flujo magnético a través del aire, ℓ y r la auto-inductancia y resistencia del motor.

Normalmente, cada motor de las extremidades está operado por el voltaje generado por el CPG, mientras que la corriente que circula por él está determinada por 2.10.

En este caso, los motores no trabajan mediante un voltaje de entrada sino mediante la corriente I que aparece en 2.9. Además, en el caso de los motores pequeños que se utilizarían, el flujo magnético está generado por un imán permanente, por lo que se puede aproximar $\Phi = \Phi_0$ como una constante.

Para poder implementar la ecuación 2.9 en el circuito del sistema de Toda-Rayleigh, necesitamos utilizar un convertidor ángulo-voltaje acoplado al rotor del motor. Normalmente estos convertidores son dispositivos ópticos que no afectan al comportamiento del motor.

$$V_n(t) = V_0\theta_n(t) \quad (2.11)$$

V_0 es la constante de conversión. De esta forma, la nueva variable $V_n(t)$ evoluciona según:

$$\frac{J}{V_0} \frac{d^2V}{dt^2} + \frac{f}{V_0} \frac{dV}{dt} + T = \Phi_0 I \quad (2.12)$$

Si separamos la corriente I en los valores:

$$\Phi I = \Phi(I_f + I_T + I_{dc}) \quad (2.13)$$

y ajustamos el valor de I_f y I_T de forma que:

$$I_f = \frac{f}{\Phi_0 V_0} \frac{dV}{dt}, \quad I_T = \frac{T}{\Phi_0} \quad (2.14)$$

Entonces, de (2.12) se pasa a:

$$\frac{d^2V}{dt^2} = \frac{V_0}{J} \Phi_0 I_{dc} \quad (2.15)$$

Esta ecuación es matemáticamente equivalente a la que describe el comportamiento de un doble condensador, pudiendo reemplazarlo del circuito por su equivalente electromecánico. En la ecuación 2.7, la constante a será reemplazada por el factor $\frac{V_0}{J\Phi_0}$, que describe las propiedades mecánicas del conjunto motor-extremidad y la parte derecha de la ecuación será I_{dc} .

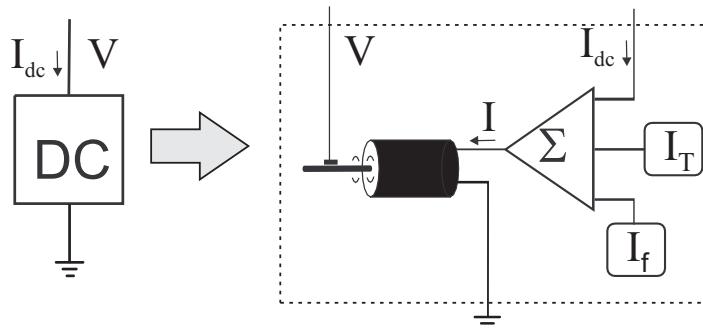


Figura 2.5: Esquema que muestra la sustitución del doble condensador (DC) por el circuito electromecánico que incluye la dinámica de la extremidad. Ambos bloques siguen ecuaciones equivalentes.

De esta forma se obtiene un sistema donde, en lugar de utilizar un doble condensador, se utiliza la dinámica de los motores (extremidades) dada por 2.15. De esta manera, el nuevo CPG incorpora un lazo de realimentación con información del conjunto cuerpo-extremidad-motor. Este CPG debe oscilar autónomamente proporcionando

funciones $\theta(t)$ para cada pata similares a las descritas anteriormente (Figuras 1.4 y 2.4), proporcionando la locomoción autónoma del robot.

Como ejemplo de cómo este nuevo CPG tiene en cuenta el ambiente, se propone el caso en el que el robot avanza en un terreno inclinado. Debido a su propio peso, aparecerá una fuerza tangencial sobre las patas, lo que produce un momento adicional sobre los motores. Este momento modifica la ecuación 2.7 introduciendo un término adicional $-\alpha g$, donde g es la aceleración de la gravedad y α la constante adecuada.

$$\frac{d^2W_n}{dt^2} = C \left\{ I_s \left[e^{\frac{W_{n-1}-W_n}{V_t}} - e^{\frac{W_n-W_{n+1}}{V_t}} \right] + I_{nr} \left(\frac{dW_n}{dt}, V_{ext} \right) - \frac{1}{R} W_n - \alpha g \right\} \quad (2.16)$$

Esta solución está relacionada con la solución de 2.7 mediante $W_n = V_n - \alpha g$. De esta forma, el nuevo sistema produce una adaptación al ambiente.

Es importante destacar que la adaptación al entorno es una respuesta del propio sistema locomotor por si mismo, sin ser necesaria ninguna fuerza externa introducida por un sistema de control. Esta respuesta, además, no necesita de ningún sensor, ya que el propio sistema incorpora las fuerzas de las extremidades para generar la señales del CPG.

2.4. Conclusiones y Objetivos del Proyecto

Como se ha visto a lo largo de estos capítulos, es posible utilizar el sistema de Toda-Rayleigh descrito como sistema locomotor de un robot. Sin embargo, es necesario realizar simulaciones numéricas que permitan predecir el comportamiento real de este robot con el fin de seleccionar correctamente las diferentes variables de diseño de cara a su construcción.

El objetivo general del proyecto es estudiar el movimiento del cuerpo del robot, considerado como un sólido rígido dotado de 6 patas e incluir, finalmente, el momento mecánico ΦI_{dc} , dónde I_{dc} proviene de la función de Toda-Rayleigh. Esta función ha sido estudiada en el Proyecto de Fin de Carrera de Omar Gómez Ortega [16]. Si bien las ecuaciones matemáticas son completamente diferentes, la robustez del sistema permite esperar un comportamiento similar en la aplicación a la locomoción de un robot.

Este objetivo se traduce en el siguiente plan de trabajo:

1. Estudio del movimiento del robot aproximado como un sólido rígido con patas sometidas a momentos mecánicos dados.
2. Estudio del movimiento del robot aproximado como un sólido rígido con patas formando parte del anillo de Toda-Rayleigh, donde los momentos mecánicos son generados por la propia dinámica del sistema. Para ello, es necesario simular el nuevo diseño de anillo Toda-Rayleigh completamente. Se parte de la ecuación 2.7 y de los parámetros estudiados en [16]. Una vez simulado el sistema completo, se estudian los modos particulares:

- 2.a Análisis del modo *Metachronal*.

2.b Análisis del modo *Caterpillar*.

2.c Análisis del modo *Tripod*.

Este estudio proporcionará valiosa información sobre la construcción final de un robot con patas basado en la metodología descrita.

Capítulo 3

Definición geométrica del problema

En este capítulo se realizará una descripción de la geometría del robot de 6 patas objeto de estudio.

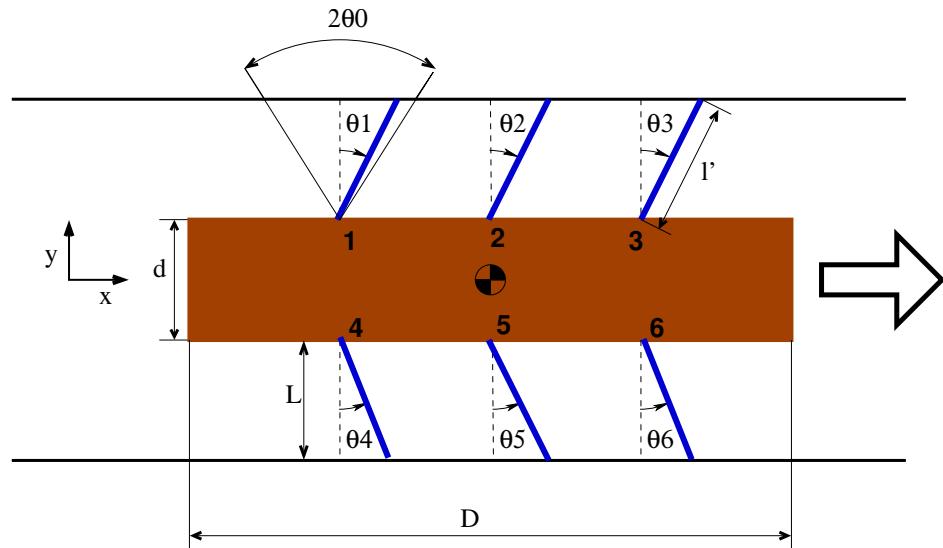


Figura 3.1: Esquema simplificado de la geometría del robot

En la figura 3.1 se puede observar una representación de la geometría del robot. Esquemáticamente, este está compuesto por un cuerpo rectangular de masa M_c uniformemente distribuida y de dimensiones $D \times d$. Las patas se modelizan como varillas rígidas de masa m_p uniformemente distribuida y de longitud total l . Las varillas están articuladas con el cuerpo mediante un sistema de guía que mantiene la distancia entre el cuerpo del robot y los carriles guía de las patas constante y de valor L deslizando las patas por debajo del cuerpo del robot. La longitud l' será la distancia, en cada instante, de la articulación del cuerpo al punto de apoyo de la pata. Se supone que, cuando las patas dejan de estar apoyadas en el suelo, estas sobresalen completamente del cuerpo (la distancia entre la articulación y el extremo de la pata será l). El ángulo θ_i es el que forma cada una de las patas con la perpendicular al cuerpo, y el sentido positivo es el que se define en la figura. La máxima amplitud del movimiento de las patas es igual al ángulo $2 \cdot \theta_0$.

3.1. Ligaduras geométricas

El sistema de guía de las articulaciones del cuerpo permite relacionar las longitudes L , l y l' mediante los ángulos θ_i y θ_0 . Para una pata i cualquiera:

$$L = l \cdot \cos(\theta_0) \quad \Rightarrow \quad l = \frac{L}{\cos(\theta_0)} \quad (3.1)$$

$$L = l'_i \cdot \cos(\theta_i) \quad \Rightarrow \quad l'_i = \frac{L}{\cos(\theta_i)} \quad (3.2)$$

(3.3)

3.2. Cálculo de los momentos de inercia

El cálculo de los momentos de inercia del cuerpo y de la pata que no está apoyada se realiza directamente a partir del esquema propuesto. Sin embargo, el sistema de guía de las patas complica el cálculo del momento de inercia de las patas que están apoyadas, ya que este no es constante, sino que depende del ángulo θ_i .

3.2.1. Momento de inercia de las patas sin apoyar

El momento de inercia de las patas que no están apoyadas en el suelo alrededor de un eje perpendicular a la articulación, de acuerdo con la hipótesis de que sobresale completamente del cuerpo del robot, será igualmente constante y de valor:

$$I = \frac{m_p l^2}{3} = \frac{m_p L^2}{3 \cos^2(\theta_0)} \quad (3.4)$$

3.2.2. Momento de inercia de las patas apoyadas

En la siguiente figura se puede ver, esquemáticamente, el efecto de la guía sobre la posición relativa de la pata con respecto al cuerpo, para una posición genérica del ángulo θ_i .

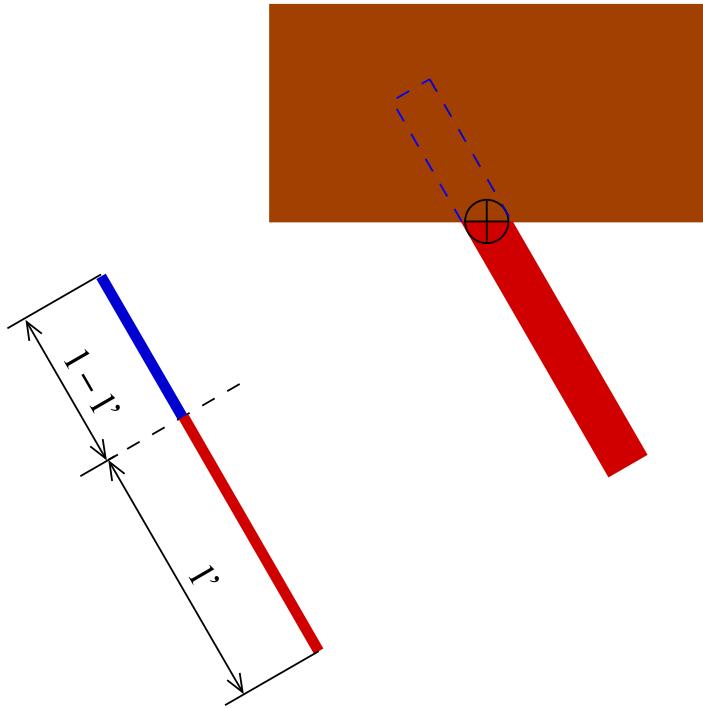


Figura 3.2: Esquema del funcionamiento de la guía

En este caso, el valor del momento de inercia alrededor de un eje perpendicular a la articulación de la pata i será:

$$I_i(\theta_i) = \int r^2 dm = \frac{m_p}{l} \int_{-(l-l')}^{l'} r^2 dr \quad (3.5)$$

$$I_i(\theta_i) = \frac{m_p l^2}{3} \left(1 - 2 \frac{l'}{l} + 2 \left(\frac{l'}{l} \right)^2 \right) \quad (3.6)$$

Introduciendo las ligaduras geométricas anteriores, podemos expresar el momento de inercia en función del ángulo θ_i

$$I_i(\theta_i) = \frac{m_p l^2}{3} \left[1 - 2 \frac{L/l}{\cos(\theta_i)} + 2 \left(\frac{L/l}{\cos(\theta_i)} \right)^2 \right] \quad (3.7)$$

Se encuentra aquí una primera diferencia con respecto a la ecuación 2.9, donde el momento de inercia J era una constante.

Capítulo 4

Dinámica del movimiento del robot

4.1. Introducción

En este capítulo se lleva a cabo una aproximación a la dinámica del movimiento del robot. En esta aproximación, el efecto del sistema de guías de las patas permite limitar el movimiento del robot a la dirección X, es decir, no se desplaza en la dirección Y ni gira alrededor de un eje perpendicular al propio cuerpo. Además, los momentos introducidos por los motores en las patas (m_i) se suponen constantes y conocidos. Este análisis se centra en el período de tiempo en el que una de las patas está sin apoyar en el suelo (pata j), lo que se correspondería con el modo de avance *Metachronal*. La deducción de las ecuaciones para los otros dos modos de movimiento, *Caterpillar* y *Tripod*, se realiza de forma completamente análoga, considerando levantadas 2 ó 3 patas simultáneamente según qué modo se analiza.

4.2. Planteamiento de las ecuaciones

En la siguiente figura se representan las fuerzas y momentos que actúan sobre el robot en un instante genérico, de acuerdo con el criterio de signos establecido. La pata j , aquella no apoyada sobre el suelo, se ha representado en color rojo, mientras que el resto se representan en color azul.

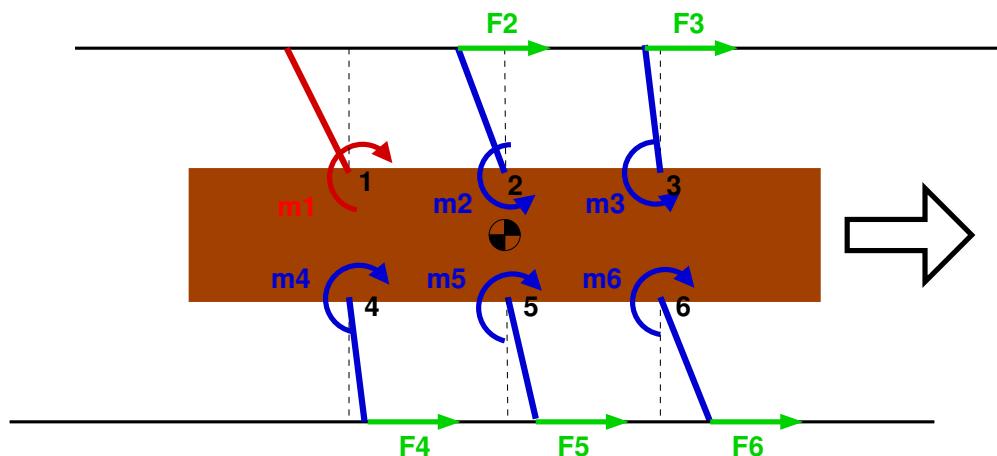


Figura 4.1: Momentos y fuerzas que actúan sobre el robot en un instante genérico

Las fuerzas F_i , representadas en la figura 4.1, son las fuerzas de rozamiento que

aparecen en el apoyo de cada una de las extremidades. Se realiza la hipótesis de que estas fuerzas son lo suficientemente grandes para mantener el apoyo fijo al suelo durante toda la duración de la fase de apoyo.

El movimiento del centro de masas del conjunto queda determinado únicamente por el movimiento de las patas en contacto con el suelo. En este primer modelo, donde los momentos que actúan sobre las patas son externos, constantes y conocidos, esta condición nos permite desacoplar el sistema de ecuaciones de las patas apoyadas y el de la pata (o patas) que se encuentran en la fase de vuelo.

4.2.1. Ecuaciones para las patas apoyadas

Aplicando la ecuación de momentos sobre cada una de las patas apoyadas en el suelo se llega a la siguiente expresión:

$$\frac{d(I_i(\theta_i)\dot{\theta}_i)}{dt} = m_i + LF_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (4.1)$$

$$\dot{I}_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + I_i(\theta_i)\ddot{\theta}_i = m_i + LF_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (4.2)$$

siendo j la pata que se encuentra en la fase de vuelo.

Aplicando ahora la ecuación de cantidad de movimiento para el sistema cuerpo y patas, donde M es la masa total del cuerpo y de las patas:

$$M = M_c + 6m_p \quad (4.3)$$

$$M \ddot{X}_{cm} = \sum_{\substack{i=1 \\ i \neq j}}^6 F_i \quad (4.4)$$

Para acabar de cerrar el sistema de ecuaciones es necesaria una condición de compatibilidad de los desplazamientos. Esta se obtiene relacionando el giro de las patas con el movimiento del cuerpo entero.

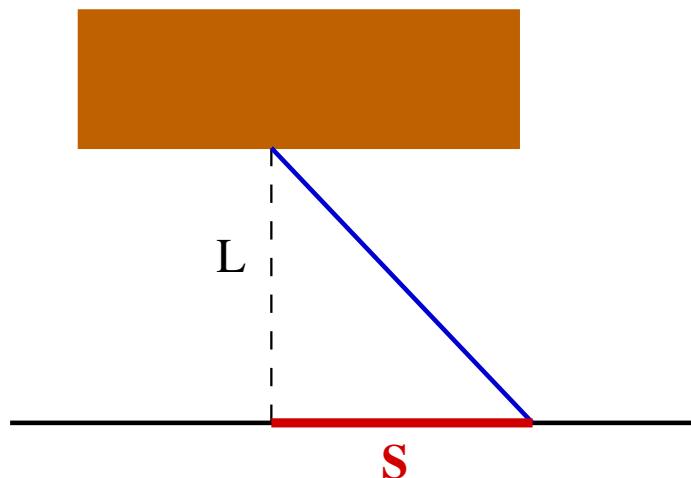


Figura 4.2: Condición de ligadura

$$\frac{S}{L} = \tan(\theta_i) \quad (4.5)$$

Derivando dos veces esta expresión con respecto del tiempo y teniendo en cuenta que la velocidad del centro de masas con respecto al suelo es igual y contraria a la de los extremos de las patas con respecto al cuerpo, se puede obtener la relación de \dot{X}_{CM} y \ddot{X}_{CM} con θ_i , $\dot{\theta}_i$ y $\ddot{\theta}_i$.

$$\dot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\dot{\theta}_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (4.6)$$

$$\ddot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\ddot{\theta}_i - \frac{2L\tan(\theta_i)}{\cos^2(\theta_i)}\dot{\theta}_i^2 \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (4.7)$$

Esta ligadura geométrica es propia del robot y no está contemplada en el anillo original de Toda-Rayleigh. Esta constituiría una segunda diferencia con respecto al sistema de Toda-Rayleigh aislado.

Resumiendo, se obtiene el siguiente sistema de 11 ecuaciones con 11 incógnitas (5 θ_i , 5 F_i y X_{CM}):

$$\dot{I}_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + I_i(\theta_i)\ddot{\theta}_i = m_i + LF_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (4.8)$$

$$M\ddot{X}_{cm} = \sum_{\substack{i=1 \\ i \neq j}}^6 F_i \quad (4.9)$$

$$\ddot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\ddot{\theta}_i - \frac{2L\tan(\theta_i)}{\cos^2(\theta_i)}\dot{\theta}_i^2 \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (4.10)$$

4.2.2. Resolución del sistema de ecuaciones de las patas apoyadas

Para la resolución del sistema de ecuaciones es necesario calcular la derivada del momento de inercia de las patas con respecto del tiempo

$$\dot{I}_i(\theta_i, \dot{\theta}_i) = \frac{m_p l^2}{3} \left[-2 \frac{L/l}{\cos(\theta_i)} + 4 \left(\frac{L/l}{\cos(\theta_i)} \right)^2 \right] \tan(\theta_i) \dot{\theta}_i \quad (4.11)$$

Despejando el valor de la fuerza F_i de la ecuación 4.2 e introduciéndolo en la ecuación 4.4, se llega a la siguiente expresión:

$$F_i = \frac{\dot{I}_i(\theta_i, \dot{\theta}_i)}{L}\dot{\theta}_i + \frac{I_i(\theta_i)}{L}\ddot{\theta}_i - \frac{m_i}{L} \quad (4.12)$$

$$M\ddot{X}_{CM} = \sum_{\substack{i=1 \\ i \neq j}}^6 \left(\frac{\dot{I}_i(\theta_i, \dot{\theta}_i)}{L}\dot{\theta}_i + \frac{I_i(\theta_i)}{L}\ddot{\theta}_i - \frac{m_i}{L} \right) \quad (4.13)$$

Introduciendo ahora la expresión de la aceleración del centro de masas calculada en la ecuación 4.7 se llega a un sistema de 5 ecuaciones diferenciales de segundo orden para

las 5 funciones $\theta_i(t)$, que puede ser integrado numéricamente:

$$-\frac{L}{\cos^2(\theta_k)}\ddot{\theta}_k - \frac{2L\tan(\theta_k)}{\cos^2(\theta_k)}\dot{\theta}_k^2 = \sum_{\substack{i=1 \\ i \neq j}}^6 \left(\frac{\dot{I}_i(\theta_i, \dot{\theta}_i)}{ML}\dot{\theta}_i + \frac{I_i(\theta_i)}{ML}\ddot{\theta}_i - \frac{m_i}{ML} \right) \quad \begin{array}{l} k = 1, 2, \dots, 6 \\ k \neq j \end{array} \quad (4.14)$$

Este sistema se puede expresar matricialmente. En este caso se ha supuesto que la pata j corresponde a la pata 1, aunque el sistema es equivalente sea cual sea la pata j . Utilizando la siguiente nomenclatura, para simplificar la notación:

$$\begin{aligned} A_i &= \frac{L}{\cos^2(\theta_i)} \\ B_i &= \frac{2L\tan(\theta_i)}{\cos^2(\theta_i)} \\ C_i &= \frac{m_p l^2}{3ML} \left[-2\frac{L/l}{\cos(\theta_i)} + 4 \left(\frac{L/l}{\cos(\theta_i)} \right)^2 \right] \tan(\theta_i) \\ D_i &= \frac{I_i(\theta_i)}{ML} \\ E &= \sum_{\substack{i=1 \\ i \neq j}}^6 \frac{m_i}{ML} \end{aligned} \quad (4.15)$$

Podemos escribir el sistema de ecuaciones 4.14 de la forma:

$$\begin{aligned} & - \begin{bmatrix} A_2 & 0 & 0 & 0 & 0 \\ 0 & A_3 & 0 & 0 & 0 \\ 0 & 0 & A_4 & 0 & 0 \\ 0 & 0 & 0 & A_5 & 0 \\ 0 & 0 & 0 & 0 & A_6 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{Bmatrix} - \begin{bmatrix} B_2 & 0 & 0 & 0 & 0 \\ 0 & B_3 & 0 & 0 & 0 \\ 0 & 0 & B_4 & 0 & 0 \\ 0 & 0 & 0 & B_5 & 0 \\ 0 & 0 & 0 & 0 & B_6 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{Bmatrix} = \\ & = \begin{bmatrix} C_2 & C_3 & C_4 & C_5 & C_6 \\ C_2 & C_3 & C_4 & C_5 & C_6 \\ C_2 & C_3 & C_4 & C_5 & C_6 \\ C_2 & C_3 & C_4 & C_5 & C_6 \\ C_2 & C_3 & C_4 & C_5 & C_6 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{Bmatrix} + \begin{bmatrix} D_2 & D_3 & D_4 & D_5 & D_6 \\ D_2 & D_3 & D_4 & D_5 & D_6 \\ D_2 & D_3 & D_4 & D_5 & D_6 \\ D_2 & D_3 & D_4 & D_5 & D_6 \\ D_2 & D_3 & D_4 & D_5 & D_6 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{Bmatrix} - \begin{Bmatrix} E \\ E \\ E \\ E \\ E \end{Bmatrix} \end{aligned} \quad (4.16)$$

Reagrupando obtenemos:

$$\begin{aligned} & \begin{bmatrix} D_2 + A_2 & D_3 & D_4 & D_5 & D_6 \\ D_2 & D_3 + A_3 & D_4 & D_5 & D_6 \\ D_2 & D_3 & D_4 + A_4 & D_5 & D_6 \\ D_2 & D_3 & D_4 & D_5 + A_5 & D_6 \\ D_2 & D_3 & D_4 & D_5 & D_6 + A_6 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{Bmatrix} = \\ & = - \begin{bmatrix} C_2 + B_2 & C_3 & C_4 & C_5 & C_6 \\ C_2 & C_3 + B_3 & C_4 & C_5 & C_6 \\ C_2 & C_3 & C_4 + B_4 & C_5 & C_6 \\ C_2 & C_3 & C_4 & C_5 + B_5 & C_6 \\ C_2 & C_3 & C_4 & C_5 & C_6 + B_6 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{Bmatrix} + \begin{Bmatrix} E \\ E \\ E \\ E \\ E \end{Bmatrix} \end{aligned} \quad (4.17)$$

De este sistema de ecuaciones es posible despejar el valor de $\ddot{\theta}_i = \ddot{\theta}_i(\theta_2, \dots, \dot{\theta}_2, \dots)$ para realizar la integración utilizando un esquema del tipo Runge-Kutta.

$$[D + A] \begin{Bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{Bmatrix} = -[C + B] \begin{Bmatrix} \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{Bmatrix} + \begin{Bmatrix} E \\ E \\ E \\ E \\ E \end{Bmatrix} \quad (4.18)$$

Siendo $\{\ddot{\theta}\}$ el vector cuyas componentes son las distintas $\ddot{\theta}_i$ y $\{\dot{\theta}^2\}$ el vector cuyas componentes son las distintas $\dot{\theta}_i^2$, el sistema queda, finalmente:

$$\{\ddot{\theta}\} = -[D + A]^{-1} [C + B] \{\dot{\theta}^2\} + [D + A]^{-1} \{E\} \quad (4.19)$$

4.2.3. Ecuaciones para la(s) pata(s) en fase vuelo

Como se ha descrito anteriormente, durante la fase de vuelo las patas se encuentran completamente extendidas, por lo que su momento de inercia con respecto a un eje perpendicular a la articulación con el cuerpo es constante. Aplicando la ecuación de momentos sobre la pata apoyada se llega a una única ecuación con una incógnita θ_j fácilmente integrable.

$$\frac{d(I_j \dot{\theta}_j)}{dt} = m_j \quad (4.20)$$

$$I_j \ddot{\theta}_j = m_j \quad (4.21)$$

4.2.4. Modelo de ecuaciones dinámicas a resolver

Siendo m el número de patas levantadas en cada uno de los instantes de tiempo (1,2 ó 3 según el modo de movimiento), podemos modelizar el movimiento del robot, bajo las hipótesis anteriores, mediante un sistema de $6 - m$ ecuaciones diferenciales acopladas y m ecuaciones diferenciales independientes.

$$\{\ddot{\theta}\}_{6-m} = - \left[[D + A]^{-1} [C + B] \right]_{(6-m) \times (6-m)} \{\dot{\theta}^2\}_{6-m} + [D + A]_{(6-m) \times (6-m)}^{-1} \{E\}_{6-m} \quad (4.22)$$

$$I_j \ddot{\theta}_j = m_j \quad (4.23)$$

4.2.5. Transición entre la fase de vuelo y la fase de apoyo

Para poder aplicarse el modelo descrito por la ecuación 4.22 debe satisfacerse la condición de ligadura 4.7 en todo instante de tiempo. Si las condiciones iniciales no son tales que satisfagan dicha ligadura, los resultados obtenidos serán erróneos. En este primer modelo, donde tan sólo se quiere observar la dinámica del robot sometido a unos momentos mecánicos dados y, en general, arbitrarios, no se entra en detalle en la zona de transición entre las fases de vuelo y de apoyo de una pata. Esta transición simplemente se modela como un cambio instantáneo en la velocidad de la pata que pasa a apoyarse de forma que se cumpla la ecuación de compatibilidad 4.6 y, debido a que en el momento de apoyo aparece la fuerza de rozamiento F_i se pasa de aplicar la

ecuación de las patas levantadas 4.23 a incluir la extremidad en el sistema 4.22.

Definimos como un ciclo de integración el tiempo que pasa entre que una de las patas se levanta (en un tiempo $t = t_0$) y alguna de las patas en contacto con el suelo alcanza el valor de amplitud máximo $-\theta_0$ (en un tiempo $t = t_f$). Utilizando el subíndice k para denominar a la pata que pasa de la fase de vuelo a la fase de apoyo en el tiempo t_0 y el subíndice i para una de las patas apoyadas en los instantes $t \leq t_0$, y que por lo tanto ya cumplía la condición de compatibilidad, podemos calcular la velocidad angular de la pata k ($\dot{\theta}_k$) que satisface la ligadura en función de $\dot{\theta}_i$, θ_i y θ_k .

$$\dot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\dot{\theta}_i = -\frac{L}{\cos^2(\theta_k)}\dot{\theta}_k \quad (4.24)$$

$$\dot{\theta}_k = \frac{\cos^2(\theta_k)}{\cos^2(\theta_i)}\dot{\theta}_i \quad (4.25)$$

4.3. Simulación numérica de la dinámica del robot

Para implementar el modelo descrito por las ecuaciones 4.22 y 4.23, se ha optado por desarrollar un programa con una estructura modular utilizando el software comercial MATLAB®. Este programa se apoya en 7 scripts diferentes, un script que contiene el programa principal y seis funciones. El código fuente se encuentra disponible en el apéndice A. A continuación, se realiza una descripción detallada del programa desarrollado.

4.3.1. Funcionamiento del programa

En la figura 4.3 se ha representado el diagrama de flujo del script principal del software. En éste se ha programado una simulación para cada uno de los modos de movimiento analizados: *Metachronal*, *Caterpillar* y *Tripod*. Cada uno de estos modos posee unas condiciones iniciales diferentes en la posición de las patas, de acuerdo con las características de su movimiento, no siendo diferentes las de velocidad, puesto que en los tres modos se parte desde el reposo.

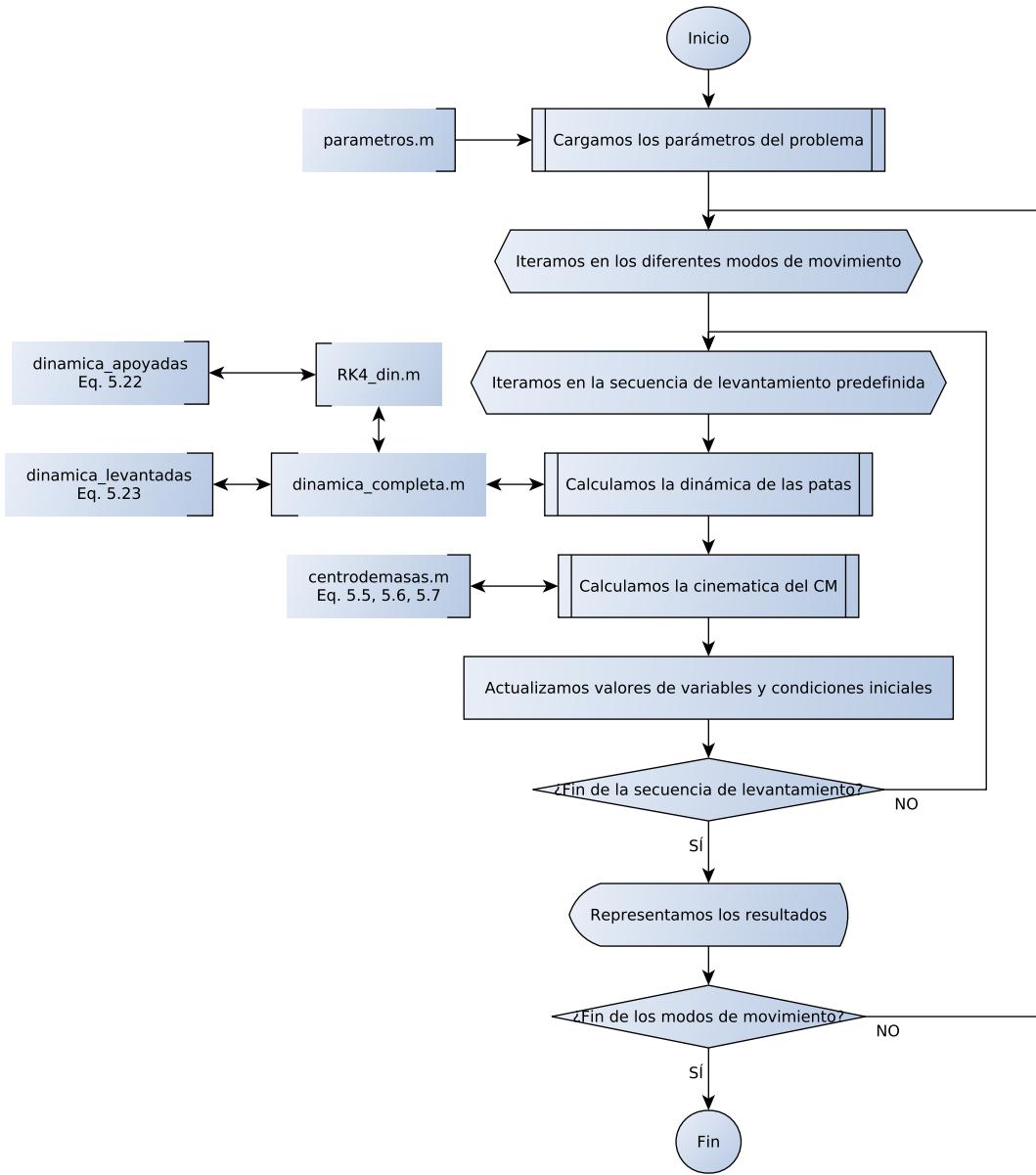


Figura 4.3: Diagrama de flujo del script principal del código de integración de momentos mecánicos constantes

La integración de las ecuaciones se va realizando de manera cíclica. En cada uno de los ciclos, las patas que se encuentran en la fase de vuelo se seleccionan de entre una secuencia preconfigurada de acuerdo a los diferentes modos de movimiento. En la siguiente tabla se muestra, a modo de ejemplo, una posible secuencia de levantamiento para cada uno de los modos. La notación “ $<>$ ” denota la extremidad o extremidades levantadas en cada uno de los ciclos, de acuerdo con la numeración introducida en la figura 4.1.

Modo <i>Metachronal</i>	Modo <i>Caterpillar</i>	Modo <i>Tripod</i>
$<1>, <2>, <3>, <4>, <5>, <6>$	$<1, 4>, <2, 5>, <3, 6>$	$<2, 4, 6>, <1, 3, 5>$

En cada uno de los ciclos se recurre a la función *dinamica_completa.m* para calcular e integrar la dinámica de las extremidades. En la figura 4.4 se observa el diagrama de

flujo que describe el comportamiento de esta función.

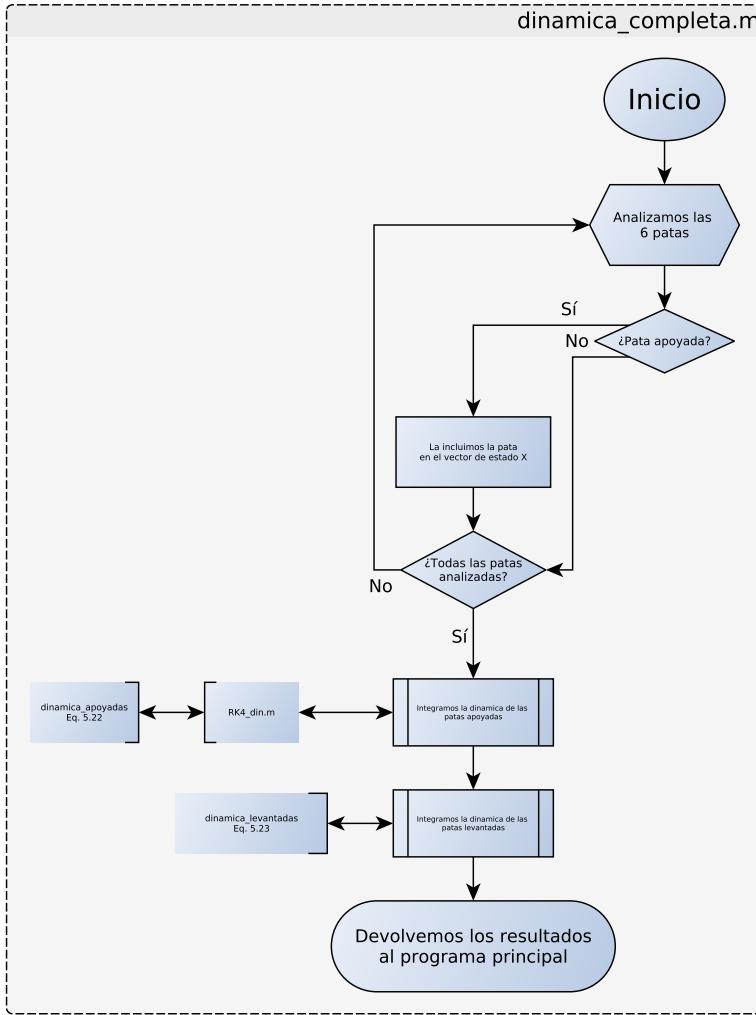


Figura 4.4: Diagrama de flujo del script *dinamica_completa.m*

Primeramente se utiliza la función *RK4_din.m* para, utilizando un esquema Runge-Kutta de orden 4, integrar el movimiento de las patas apoyadas. Para realizar esta integración, se utiliza el script *dinamica_apoyadas.m*, que realiza los cálculos necesarios para obtener la ecuación 4.22. La condición para que finalice la integración de un determinado ciclo es que alguna de las extremidades supere el ángulo de amplitud máximo θ_0 . Una vez realizada la integración del movimiento de las patas apoyadas se pasa a integrar la dinámica de la extremidad o extremidades que se encuentren en fase de vuelo durante el ciclo. Para ello, se utiliza el script contenido en *dinamica_levantadas.m*. En este script, primero se calcula el momento necesario para que la pata pase, en el tiempo que dura ese ciclo, de la posición final del anterior ciclo de integración ($\theta_{f_{n-1}}$) a la posición $+\theta_0$. Una vez realizado este cálculo, se integra la ecuación 4.23. Esta integración se realiza, gracias a que ésta ecuación está desacoplada del resto, de manera analítica. El script "*dinamica_levantadas.m*" es equivalente a integrar la ecuación diferencial 4.23 con las siguientes condiciones, donde m es un momento constante y desconocido a calcular y t_{0n} y t_{fn} los tiempos iniciales y finales del ciclo de integración n. Además se impone que la velocidad angular inicial es nula.

$$I_j \ddot{\theta} = m_j \quad (4.26)$$

$$\theta(t_{0_n}) = \theta_{f_{n-1}} \quad (4.27)$$

$$\theta(t_{fn}) = \theta_0 \quad (4.28)$$

$$\dot{\theta}(t_{0_n}) = 0 \quad (4.29)$$

Una vez realizada la integración de la dinámica de las 6 extremidades del robot, se procede a calcular la cinemática del centro de masas del mismo. Para ello, se utiliza la función *centrodemasas.m*. En ella se calcula el movimiento del centro de masas utilizando las condiciones de ligadura descritas por las ecuaciones 4.5, 4.6 y 4.7 y los valores de $\theta(t)$, $\dot{\theta}(t)$ y $\ddot{\theta}(t)$ de las patas apoyadas previamente calculados.

Una vez terminada la integración de uno de los ciclos, se realiza la transición entre ciclos actualizando las condiciones iniciales de las patas que pasen a apoyarse para que cumplan la ligadura geométrica y el siguiente ciclo comienza cambiando las extremidades que se encuentran en la fase de vuelo, de manera consecutiva hasta que se agota la secuencia predefinida de levantamiento.

Entre cada simulación de los diferentes modos de movimiento se realiza la representación de los resultados en función del tiempo.

4.3.2. Resultados de la simulación

A continuación, se presentan los resultados de las simulaciones numéricas correspondientes a los diferentes modos de movimiento. Los parámetros físicos del problema seleccionados se pueden observar en la tabla 4.1.

L (m)	mp (kg)	θ_0 ($^{\circ}$)	M (kg)	m_i (N· m)
0.3	0.2	35	3	-0.25

Tabla 4.1: Parámetro físicos del problema

Además de estos parámetros, se ha utilizado un paso de integración de 0.005 s y, como se ha comentado anteriormente, se parte desde el reposo.

Modo Metachronal

Para la integración de este modo de movimiento se han utilizado las siguientes condiciones iniciales:

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	$\dot{\theta}_4$	$\dot{\theta}_5$	$\dot{\theta}_6$
$-\theta_0$	$-\frac{2}{3}\theta_0$	$-\frac{\theta_0}{3}$	$\frac{\theta_0}{3}$	$\frac{2}{3}\theta_0$	θ_0	0	0	0	0	0	0

Tabla 4.2: Condiciones iniciales del modo *Metachronal*

La secuencia de levantamiento de extremidades en esta simulación ha sido la siguiente:

Modo Metachronal
$\langle 1 \rangle, \langle 2 \rangle, \langle 3 \rangle, \langle 4 \rangle, \langle 5 \rangle, \langle 6 \rangle, \langle 1 \rangle, \langle 2 \rangle, \langle 3 \rangle, \langle 4 \rangle, \langle 5 \rangle, \langle 6 \rangle$

Tabla 4.3: Secuencia de levantamiento del modo *Metachronal*

A continuación, se presenta la figura 4.5 con los resultados de la integración de la dinámica de este modo de movimiento.

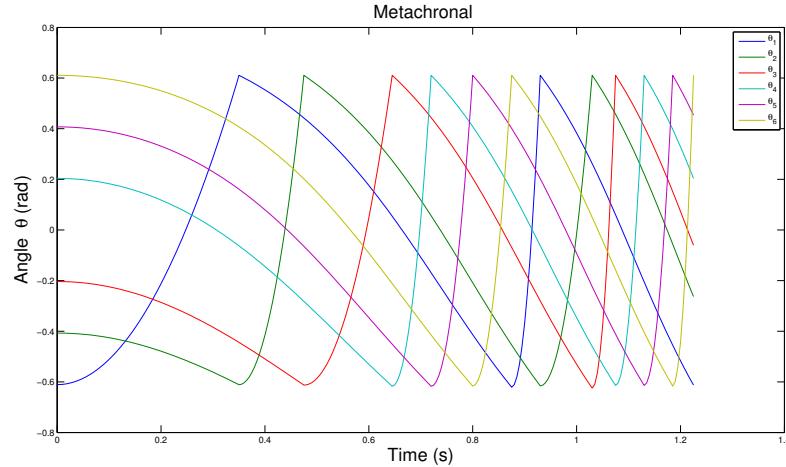


Figura 4.5: Ángulos de las extremidades en función del tiempo para el modo *Metachronal*

A partir de la integración de la dinámica de las extremidades, se calcula la cinemática del centro de masas del robot. Esta cinemática se puede calcular de manera independiente con los valores de θ , $\dot{\theta}$ y $\ddot{\theta}$ de cada una de las patas apoyadas en cada instante del tiempo. Si la integración de la dinámica ha sido correcta, los valores calculados con cada una de las extremidades apoyadas deben de coincidir, como se observa en las figuras siguientes.

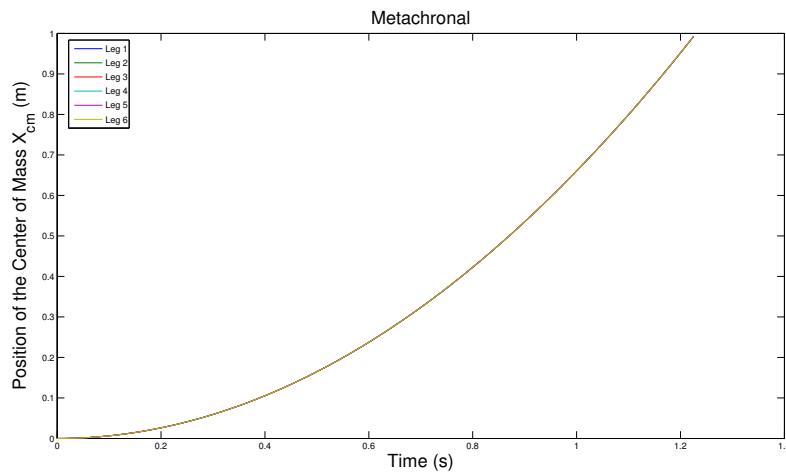


Figura 4.6: Posición del Centro de Masas en función del tiempo para el modo *Metachronal*

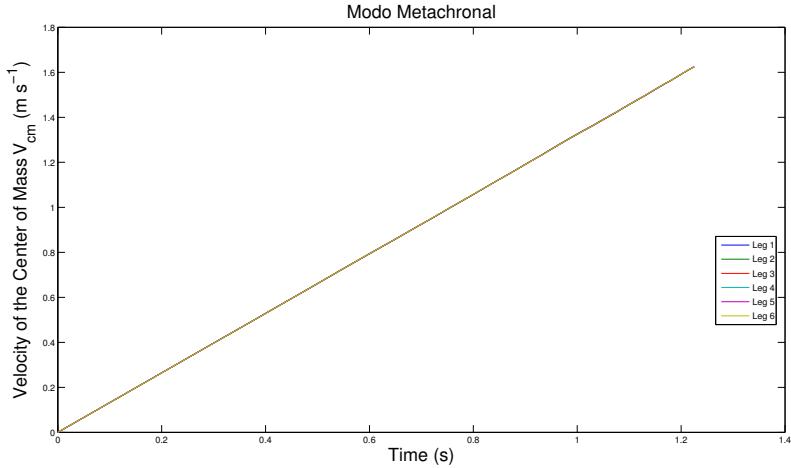


Figura 4.7: Velocidad del Centro de Masas en función del tiempo para el modo *Metachronal*

La coincidencia de la cinemática del centro de masas calculada con los valores de las diferentes extremidades que se observa en las figuras 4.6 y 4.7 permite validar el esquema numérico utilizado en la integración.

Modo Caterpillar

En el caso del modo de avance *Caterpillar* las condiciones iniciales utilizadas son:

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	$\dot{\theta}_4$	$\dot{\theta}_5$	$\dot{\theta}_6$
$-\theta_0$	0	θ_0	$-\theta_0$	0	θ_0	0	0	0	0	0	0

Tabla 4.4: Condiciones iniciales del modo *Caterpillar*

La secuencia de levantamiento de extremidades en esta simulación ha sido la siguiente:

Modo <i>Caterpillar</i>
$<1,4>, <2,5>, <3,6>, <1,4>, <2,5>, <3,6>, <1,4>, <2,5>, <3,6>,$

Tabla 4.5: Secuencia de levantamiento del modo *Caterpillar*

De igual manera que antes, se presentan los resultados de la integración de la dinámica de las extremidades de este modo, seguido de la cinemática del centro de masas.

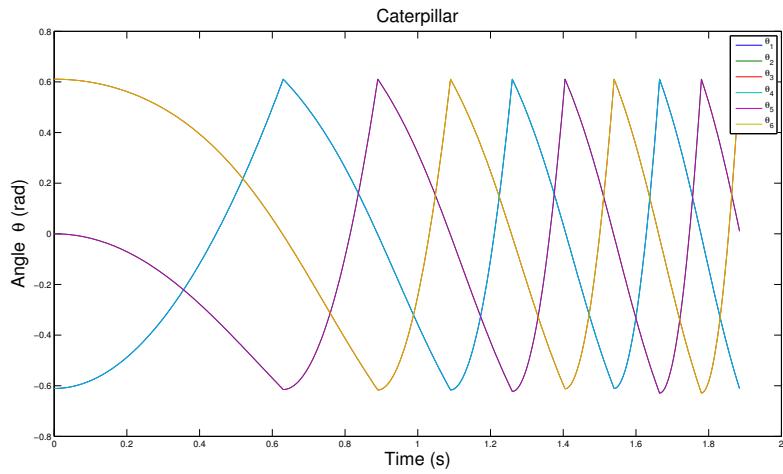


Figura 4.8: Ángulos de las extremidades en función del tiempo para el modo *Caterpillar*

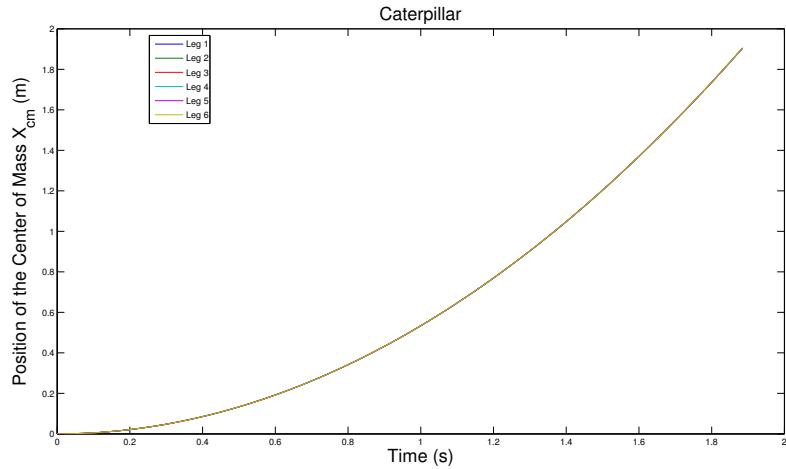


Figura 4.9: Posición del Centro de Masas en función del tiempo para el modo *Caterpillar*

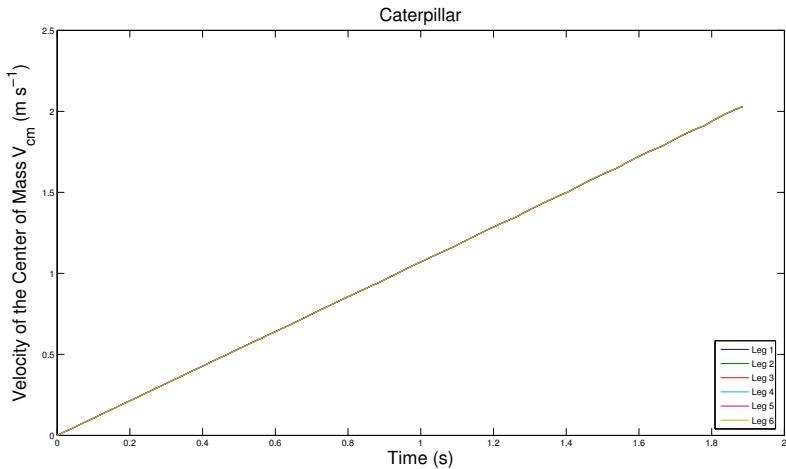


Figura 4.10: Velocidad del Centro de Masas en función del tiempo para el modo *Caterpillar*

Modo Tripod

Se pasa ahora a analizar el modo *Tripod*. Las condiciones iniciales de este modo son:

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_3$	$\dot{\theta}_4$	$\dot{\theta}_5$	$\dot{\theta}_6$
θ_0	$-\theta_0$	θ_0	$-\theta_0$	θ_0	$-\theta_0$	0	0	0	0	0	0

Tabla 4.6: Condiciones iniciales del modo *Tripod*

En este caso, la secuencia de levantamiento de extremidades pasa a ser:

Modo <i>Tripod</i>
$<2,4,6>, <1,3,5>, <2,4,6>, <1,3,5>, <2,4,6>, <1,3,5>, <2,4,6>, <1,3,5>$

Tabla 4.7: Secuencia de levantamiento del modo *Tripod*

Se presentan ahora los resultados de la simulación de este modo de avance.

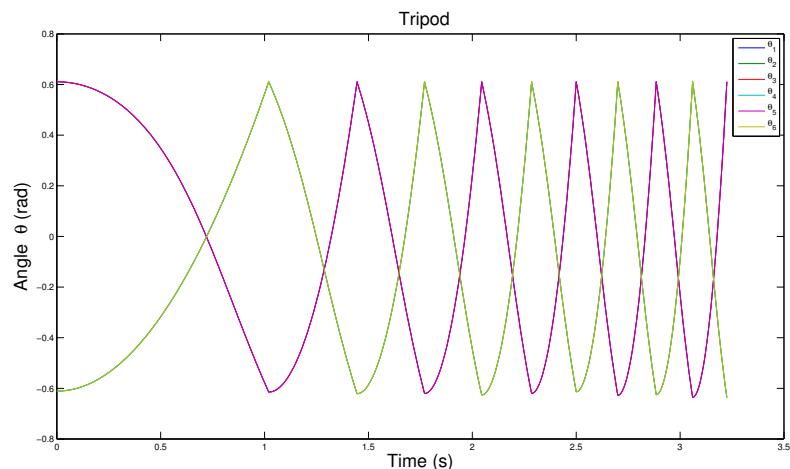


Figura 4.11: Ángulos de las extremidades en función del tiempo para el modo *Tripod*

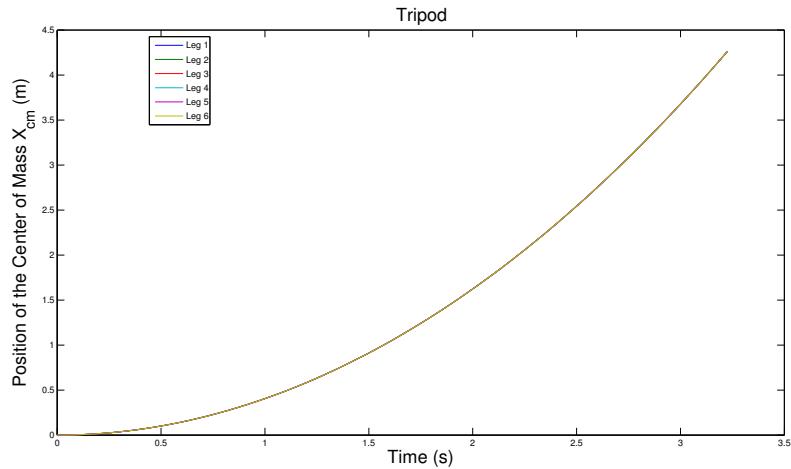


Figura 4.12: Posición del Centro de Masas en función del tiempo para el modo *Tripod*

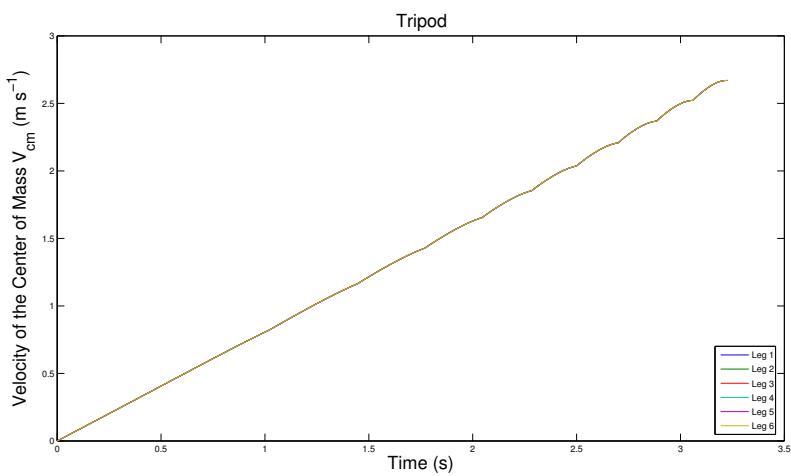


Figura 4.13: Velocidad del Centro de Masas en función del tiempo para el modo *Tripod*

4.3.3. Discusión de los resultados de la simulación

En las figuras 4.5, 4.8 y 4.11 se pueden observar funciones $\theta(t)$ muy similares a las de la figura 1.4, donde se mostraban las señales que provocaban los diferentes modos de movimiento. Sin embargo, la forma de estas señales no es totalmente idéntica. Se puede inferir que el modelo dinámico implementado recoge de forma correcta los diferentes modos de avance de los insectos generados por el CPG. Sin embargo, como era de esperar, los momentos mecánicos de las articulaciones de las patas necesarios para reproducir fielmente el movimiento de éstas no son constantes.

Analizando los resultados de la cinemática del centro de masas, se observa cómo el movimiento generado por momentos mecánicos constantes es un movimiento de aceleración constante. Como se ha comentado anteriormente, este movimiento se puede calcular independientemente con los resultados de cada una de las extremidades apoyadas. En los resultados obtenidos, se ha representado el movimiento calculado a partir de cada una de las diferentes extremidades apoyadas en función del tiempo. Como no podía ser de otra manera, estos resultados coinciden entre sí, asegurando la correcta integración del modelo físico.

Capítulo 5

Dinámica del movimiento del robot utilizando el sistema de Toda-Rayleigh como Central Pattern Generator

5.1. Introducción

En este capítulo, se realiza el estudio de la dinámica del robot incorporando el sistema de Toda-Rayleigh. Este sistema realiza la función del Generador Central de Patrones (CPG), controlando los momentos mecánicos que actúan sobre cada una de las extremidades en función del tiempo.

Tal y como se ha visto en la sección 2.3, los momentos mecánicos introducidos por los motores (ΦI_{dc}), están dados por la expresión 2.15, donde I_{dc} corresponde a la parte derecha de la ecuación 2.7.

En la sección 2.3 también se explica cómo el nuevo sistema de Toda-Rayleigh, que actúa como CPG, debería auto-oscilar, generando los diferentes modos de movimiento. Sin embargo, las ligaduras geométricas introducidas por el robot en las patas apoyadas (ecuación 4.7), impedirían la propagación de la oscilación de una pata a la siguiente, por lo que es fundamental la correcta identificación del instante en que las patas deben levantarse (desapareciendo entonces la ligadura geométrica).

En una primera simulación, se buscó comprobar la robustez de las soluciones del sistema. Como se comentó en apartados anteriores, a pesar de las diferencias con respecto al sistema de Toda-Rayleigh aislado, las funciones $\theta(t)$ resultantes de la simulación deberían de ser similares a las medidas experimentalmente en el circuito equivalente (figura 1.4). En esta modelización, la pata levantada se identificó, simplemente, como aquella en la cual la velocidad angular $\dot{\theta}$ era positiva (estaría avanzando). Este método tan simple de identificar las extremidades en fase de vuelo fallaría en las proximidades de los instantes de apoyo/levantamiento, dónde, al ser $\dot{\theta}$ muy próxima a 0, el método numérico puede introducir saltos en el signo de la misma. Para asegurar la correcta propagación de la oscilación, y que esta no desapareciese debido a la ligadura geométrica, se suponían levantadas no sólo la pata que, efectivamente, se encontraba en fase de vuelo, si no también las patas anteriores y posteriores.

En la figura 5.1 se han representado las funciones $\theta(t)$ resultantes de esta primera aproximación para el modo de movimiento *Metachronal*. Se puede observar como la forma de estas es similar a la de las figuras 1.4 y 2.4, confirmando la robustez de las soluciones del sistema.

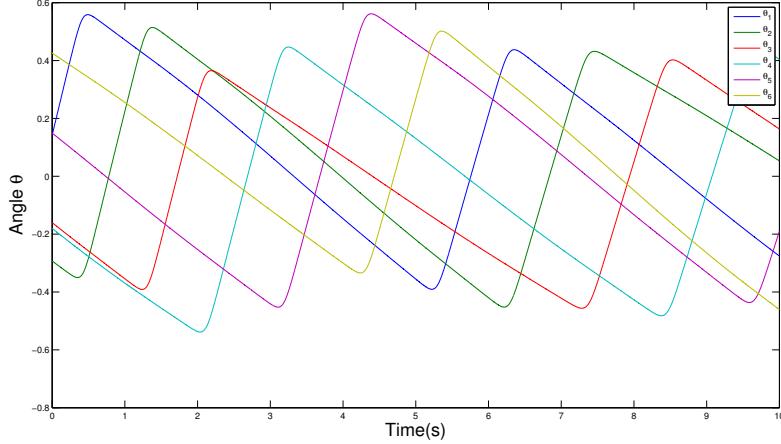


Figura 5.1: Funciones $\theta(t)$ resultantes de la primera aproximación al sistema de Toda-Rayleigh

La figura 5.2 muestra la función de Toda-Rayleigh, que determina el momento mecánico introducido por los motores de las extremidades. Se observa cómo, cuando la pendiente de la función de Toda-Rayleigh es negativa, la pendiente de la función $\theta(t)$ correspondiente es positiva, por lo que esa pata estaría en fase de vuelo o avance.

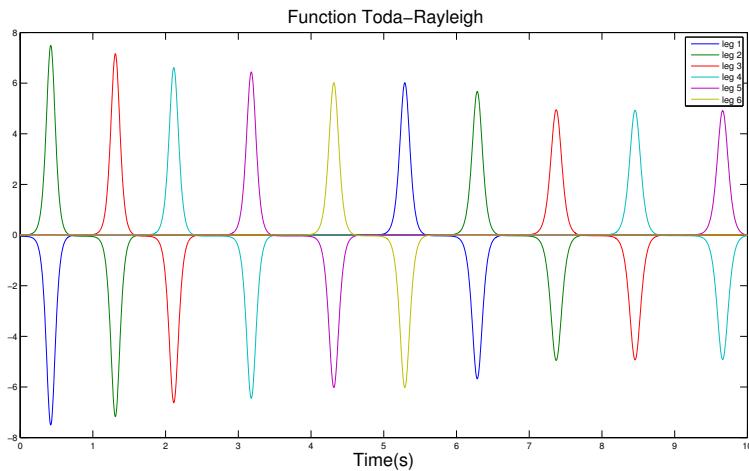


Figura 5.2: Función de Toda-Rayleigh $\frac{\Phi I_{dc}}{J}$

Estos resultados no sólo sirvieron para comprobar que las soluciones obtenidas eran similares a las buscadas, sino que también permitieron desarrollar una nueva metodología de identificación de las extremidades que se encontraban en la fase de vuelo. Este método relaciona el estado de las patas (apoyadas o en vuelo), con la derivada de la función de Toda-Rayleigh. Cuando la derivada sea negativa, las patas se encuentran en fase de vuelo, estando apoyadas en el caso contrario. Esta metodología permite identificar de forma correcta, en todo momento, qué extremidades se encuentran en fase

de apoyo y fase de vuelo, produciéndose la correcta propagación de la oscilación sin necesidad de cancelar las ligaduras en las patas anteriores y posteriores.

5.2. La función de Toda-Rayleigh y su sistema de control automático

Para poder utilizar el sistema de Toda-Rayleigh en la locomoción de un robot se debe asegurar que las oscilaciones de las patas se centren en el cero, o de lo contrario se podría superar la amplitud máxima de oscilación θ_0 , que es un límite mecánico del robot. Para asegurar que las oscilaciones se centren en el 0, en [16] se propone la implementación de un sistema de control automático electrónico compuesto por 6 resistencias R_B conectadas a cada uno de los nodos del circuito, un condensador, un amplificador de constante K y un interruptor para activar ($C_r = 1$) o desactivar el sistema de control ($C_r = 0$). En la figura 5.3 se puede ver un esquema de cómo se implementaría el circuito de control en el sistema electromecánico, donde el bloque DC representa los motores de las extremidades, tal y como se explicó en la sección 2.3.

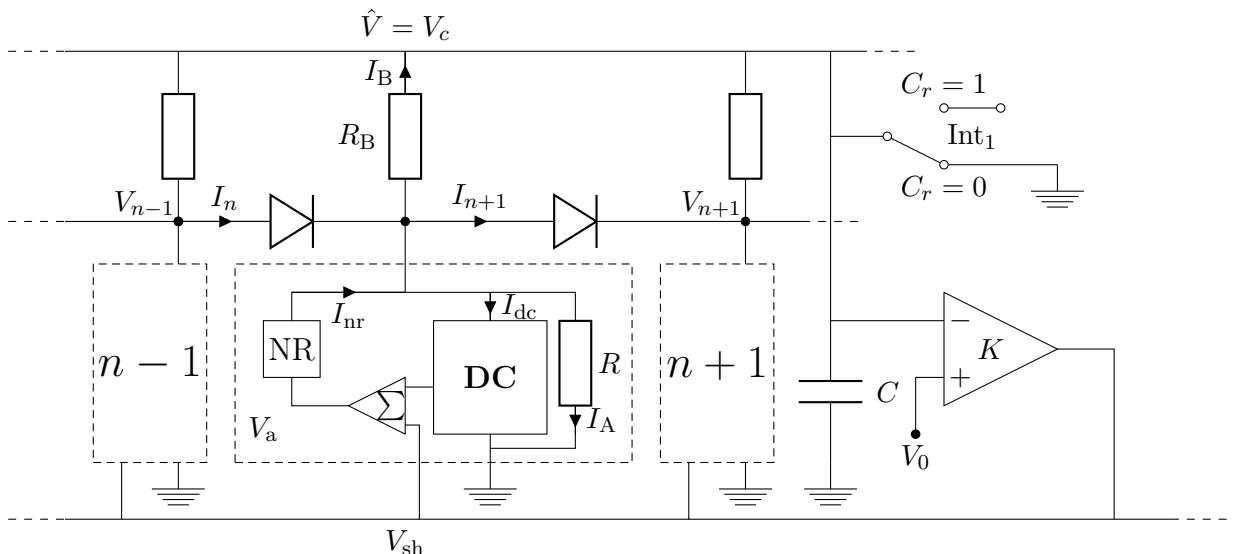


Figura 5.3: Esquema de la implementación del control automático en el sistema electromecánico

Este sistema actúa sobre el voltaje de control, ahora denominado V_{sh} y que es equivalente a V_{ext} de la expresión 2.6, siguiendo la ecuación:

$$V_{sh} = V_0 - KC_r V_C \quad (5.1)$$

donde V_0 es el voltaje externo y constante introducido en el sistema y V_C es el voltaje medio que circula por los nodos, con el que va cargándose el condensador.

Este sistema de control introduce nuevas ecuaciones, fácilmente reducibles a una nueva ecuación diferencial que permite calcular el voltaje medio V_C . El voltaje en cada uno de los nodos n del circuito (o extremidades del robot) será:

$$V_n = V_C + R_B I_B \quad n = 1, 2, \dots, 6 \quad (5.2)$$

Sumando las 6 ecuaciones 5.2 se obtiene:

$$\sum_{i=1}^6 V_i - 6V_c = 6R_B \sum_{i=1}^6 I_i \quad (5.3)$$

Aplicando Kirchhoff para calcular la intensidad I_C que pasa por el condensador:

$$\sum_{i=1}^6 I_i + I_C = 0 \quad (5.4)$$

y teniendo en cuenta la ecuación que modeliza el comportamiento de los condensadores

$$I_C = C\dot{V}_C \quad (5.5)$$

se llega a la ecuación diferencial 5.6 que permite el cálculo del voltaje $V_C(t)$ en función del tiempo.

$$\frac{\sum_{i=1}^6 V_i}{6} - V_C = R_B C \dot{V}_C \quad (5.6)$$

Teniendo en cuenta este sistema de control, la expresión de la función de Toda-Rayleigh que permite calcular los momentos mecánicos introducidos por cada uno de los motores, en su forma final, queda:

$$\frac{V_0}{J} \Phi I_{dc} = \omega^2 \left\{ I_s \left(e^{(\frac{V_{n-1}-V_n}{V_t})} - e^{(\frac{V_n-V_{n+1}}{V_t})} \right) + \gamma \left[\mu_0 - (\dot{V}_n - \beta V_{sh})^2 \right] (\dot{V}_n - \beta V_{sh}) - \frac{V_n}{R} - \frac{V_n - C_r V_c}{R_B} \right\} \quad (5.7)$$

En esta expresión, los únicos parámetros libres son ω , β , R , R_B , C y la constante K del amplificador, puesto que los demás parámetros están relacionados con la implementación electrónica del sistema y son difícilmente modificables.

Mediante la conversión ángulo-voltaje introducida ($V_n \rightarrow \theta_n$), se observa cómo los momentos introducidos por los motores dependen de la dinámica del robot de la forma:

$$T_i = T(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i) \quad (5.8)$$

5.3. Modificación de las ecuaciones

La deducción de las ecuaciones se realiza de manera análoga al caso anterior. Se utiliza el subíndice j para denominar la pata o patas, dependiendo del modo de avance, que se encuentran en la fase de vuelo. Aunque las ecuaciones planteadas se corresponderían con el modo *Metachronal* considerando, nuevamente, la pata 1 como la extremidad en fase de vuelo, las ecuaciones para los demás modos y/o patas en fase de vuelo se deducen de manera similar.

La diferencia fundamental con respecto al modelo dinámico anterior es que, ahora, los momentos no son constantes, sino que estos son función de la propia dinámica del sistema.

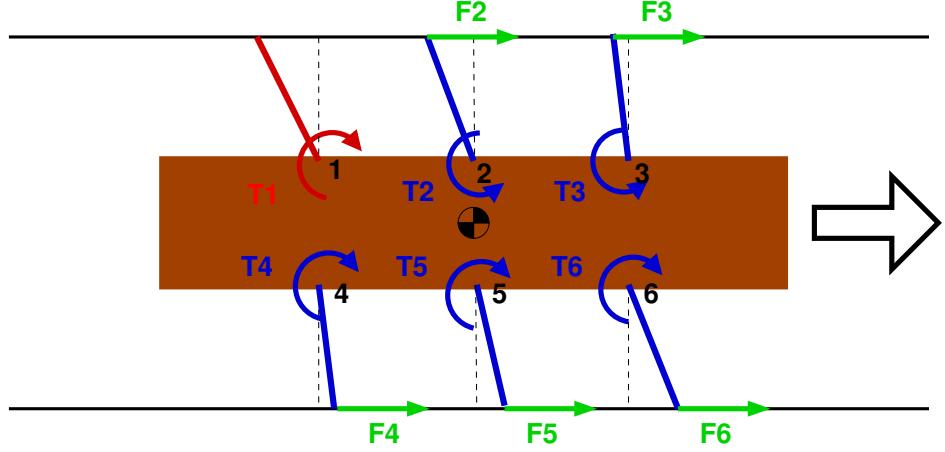


Figura 5.4: Momentos y fuerzas que actúan sobre el robot en un instante genérico

En la figura 5.4 se ha utilizado la nomenclatura T_i para referirse a los momentos proporcionados por el sistema dinámico, para diferenciarlos de los momentos constantes m_i del caso anterior. Como se ha explicado anteriormente, estos momentos dependerán de $\theta_i, \theta_{i-1}, \theta_{i+1}$ y $\dot{\theta}_i$

$$T_i = T(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i) \quad (5.9)$$

Esta dependencia de los momentos con los ángulos de las extremidades anteriores y posteriores impide desacoplar las ecuaciones de las patas en fase de vuelo del resto del sistema, como sí ocurría en el caso anterior.

Las ecuaciones para las extremidades en contacto con el suelo serán:

$$\frac{d(I_i(\theta_i)\dot{\theta}_i)}{dt} = T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i) + LF_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (5.10)$$

$$\dot{I}_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + I_i(\theta_i)\ddot{\theta}_i = T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i) + LF_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (5.11)$$

mientras que las de las patas en fase de vuelo quedarán:

$$I_j\ddot{\theta}_j = T_j(\theta_j, \theta_{j-1}, \theta_{j+1}, \dot{\theta}_j) \quad (5.12)$$

Para completar el sistema de ecuaciones que modela el comportamiento dinámico del robot hay que añadir la ecuación de cantidad de movimiento del centro de masas del robot (5.13) y la condición que modela la ligadura geométrica analizada en el capítulo anterior (5.14).

$$M\ddot{X}_{cm} = \sum_{\substack{i=1 \\ i \neq j}}^6 F_i \quad (5.13)$$

$$\ddot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\ddot{\theta}_i - \frac{2L\tan(\theta_i)}{\cos^2(\theta_i)}\dot{\theta}_i^2 \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (5.14)$$

Agrupando todas las ecuaciones, se obtiene el siguiente sistema que modela la nueva dinámica del robot:

$$\dot{I}_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + I_i(\theta_i)\ddot{\theta}_i = T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i) + LF_i \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (5.15)$$

$$I_j\ddot{\theta}_j = T_j(\theta_j, \theta_{j-1}, \theta_{j+1}, \dot{\theta}_j) \quad (5.16)$$

$$M\ddot{X}_{cm} = \sum_{\substack{i=1 \\ i \neq j}}^6 F_i \quad (5.17)$$

$$\ddot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\ddot{\theta}_i - \frac{2L\tan(\theta_i)}{\cos^2(\theta_i)}\dot{\theta}_i^2 \quad i = 1, 2, \dots, 6 \quad i \neq j \quad (5.18)$$

Siendo m el número de extremidades en fase de vuelo en un instante de tiempo en el caso más general, se obtiene un sistema con $13 - m$ incógnitas (6 funciones $\theta(t)$, $6 - m$ fuerzas de rozamiento $F(t)$ y $X_{CM}(t)$) con sus $13 - m$ ecuaciones ($6 - m$ ecuaciones 5.15 para las patas apoyadas, m ecuaciones 5.16 para las patas en fase de vuelo, 1 ecuación 5.17 para el movimiento del centro de masas y $6 - m$ condiciones de compatibilidad geométrica 5.18), puesto que los diferentes T_i , son funciones conocidas que dependen de θ_i , θ_{i-1} , θ_{i+1} y $\dot{\theta}_i$

Este sistema de ecuaciones se puede reducir a un sistema de 6 ecuaciones diferenciales de segundo orden, donde las únicas incógnitas sean las funciones $\theta_i(t)$. Para ello, se procede, de igual forma que en el capítulo anterior, a despejar la expresión de las fuerzas de rozamiento F_i en función de θ y $\dot{\theta}$ para, a continuación, igualar las ecuaciones 5.17 y 5.18 para obtener

$$F_i = \frac{\dot{I}_i(\theta_i, \dot{\theta}_i)}{L}\dot{\theta}_i + \frac{I_i(\theta_i)}{L}\ddot{\theta}_i - \frac{T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i)}{L} \quad (5.19)$$

$$M\ddot{X}_{CM} = \sum_{\substack{i=1 \\ i \neq j}}^6 \left(\frac{\dot{I}_i(\theta_i, \dot{\theta}_i)}{L}\dot{\theta}_i + \frac{I_i(\theta_i)}{L}\ddot{\theta}_i - \frac{T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i)}{L} \right) \quad (5.20)$$

$$-\frac{L}{\cos^2(\theta_k)}\ddot{\theta}_k - \frac{2L\tan(\theta_k)}{\cos^2(\theta_k)}\dot{\theta}_k^2 = \sum_{\substack{i=1 \\ i \neq j}}^6 \left(\frac{\dot{I}_i(\theta_i, \dot{\theta}_i)}{ML}\dot{\theta}_i + \frac{I_i(\theta_i)}{ML}\ddot{\theta}_i - \frac{T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i)}{ML} \right) \quad k = 1, \dots, 6 \quad (5.21)$$

Pasando ahora a una notación matricial para el caso de $j = 1$ y definiendo las siguientes expresiones para simplificar la notación

$$\begin{aligned} A_i &= \frac{L}{\cos^2(\theta_i)} \\ B_i &= \frac{2L\tan(\theta_i)}{\cos^2(\theta_i)} \\ C_i &= \frac{m_p l^2}{3ML} \left[-2\frac{L/l}{\cos(\theta_i)} + 4 \left(\frac{L/l}{\cos(\theta_i)} \right)^2 \right] \tan(\theta_i) \\ D_i &= \frac{I_i(\theta_i)}{ML} \\ E &= \sum_{\substack{i=1 \\ i \neq j}}^6 \frac{T_i(\theta_i, \theta_{i-1}, \theta_{i+1}, \dot{\theta}_i)}{ML} \end{aligned} \quad (5.22)$$

podemos expresar el sistema de ecuaciones diferenciales de segundo orden completo a resolver:

$$\begin{aligned}
 & - \left[\begin{array}{cccccc} -I_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_6 \end{array} \right] \left\{ \begin{array}{l} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{array} \right\} - \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & B_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & B_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & B_6 \end{array} \right] \left\{ \begin{array}{l} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{array} \right\} = \\
 & = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 0 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 0 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 0 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 0 & C_2 & C_3 & C_4 & C_5 & C_6 \end{array} \right] \left\{ \begin{array}{l} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{array} \right\} + \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D_2 & D_3 & D_4 & D_5 & D_6 \\ 0 & D_2 & D_3 & D_4 & D_5 & D_6 \\ 0 & D_2 & D_3 & D_4 & D_5 & D_6 \\ 0 & D_2 & D_3 & D_4 & D_5 & D_6 \\ 0 & D_2 & D_3 & D_4 & D_5 & D_6 \end{array} \right] \left\{ \begin{array}{l} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{array} \right\} - \left\{ \begin{array}{l} -T_1 \\ E \\ E \\ E \\ E \\ E \end{array} \right\} \\
 \end{aligned} \tag{5.23}$$

que reagrupando términos queda:

$$\begin{aligned}
 & \left[\begin{array}{cccccc} -I_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & D_2 + A_2 & D_3 & D_4 & D_5 & D_6 \\ 0 & D_2 & D_3 + A_3 & D_4 & D_5 & D_6 \\ 0 & D_2 & D_3 & D_4 + A_4 & D_5 & D_6 \\ 0 & D_2 & D_3 & D_4 & D_5 + A_5 & D_6 \\ 0 & D_2 & D_3 & D_4 & D_5 & D_6 + A_6 \end{array} \right] \left\{ \begin{array}{l} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{array} \right\} = \\
 & = - \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 + B_2 & C_3 & C_4 & C_5 & C_6 \\ 0 & C_2 & C_3 + B_3 & C_4 & C_5 & C_6 \\ 0 & C_2 & C_3 & C_4 + B_4 & C_5 & C_6 \\ 0 & C_2 & C_3 & C_4 & C_5 + B_5 & C_6 \\ 0 & C_2 & C_3 & C_4 & C_5 & C_6 + B_6 \end{array} \right] \left\{ \begin{array}{l} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{array} \right\} + \left\{ \begin{array}{l} -T_1 \\ E \\ E \\ E \\ E \\ E \end{array} \right\}
 \end{aligned} \tag{5.24}$$

Con una notación más abreviada, se obtiene:

$$\left[\tilde{D} + \tilde{A} \right] \left\{ \begin{array}{l} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{array} \right\} = - \left[\tilde{C} + \tilde{B} \right] \left\{ \begin{array}{l} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \\ \dot{\theta}_4^2 \\ \dot{\theta}_5^2 \\ \dot{\theta}_6^2 \end{array} \right\} + \left\{ \begin{array}{l} -T_1 \\ E \\ E \\ E \\ E \\ E \end{array} \right\} \tag{5.25}$$

Despejando los valores de $\ddot{\theta}_i$, se obtiene el sistema de ecuaciones diferenciales de segundo orden a integrar mediante un esquema Runge-Kutta para obtener la dinámica de las extremidades del robot, donde la notación \sim se utiliza para diferenciar las matrices de las del modelo de momentos constantes.

$$\{\ddot{\theta}\} = - [\tilde{D} + \tilde{A}]^{-1} [\tilde{C} + \tilde{B}] \{\dot{\theta}^2\} + [\tilde{D} + \tilde{A}]^{-1} \{\tilde{E}\} \tag{5.26}$$

Este sistema de ecuaciones es claramente diferente de las ecuaciones del sistema de Toda-Rayleigh original.

5.3.1. Transición entre la fase de vuelo y la fase de apoyo

Al igual que en el caso anterior, para poder aplicar el modelo dinámico propuesto en 5.26, la ligadura geométrica 4.5 debe satisfacerse en todo instante del tiempo. En particular, las condiciones iniciales de las extremidades en contacto con el suelo deben de cumplir dicha ligadura geométrica. La modelización de la transición entre la fase de vuelo y la fase de apoyo de una extremidad se realiza de igual forma que en el caso anterior, como un cambio de velocidad de la extremidad que pase a apoyarse de tal forma que respete la ligadura. Utilizando nuevamente el subíndice k para la pata que pasa a apoyarse en un instante t_0 y el subíndice i para una pata apoyada en los instantes $t \leq t_0$, y que por lo tanto ya cumplía la condición de compatibilidad, podemos calcular la velocidad angular de la pata k ($\dot{\theta}_k$) que satisface la ligadura geométrica en función de θ_k , θ_i y $\dot{\theta}_i$.

$$\dot{X}_{CM} = -\frac{L}{\cos^2(\theta_i)}\dot{\theta}_i = -\frac{L}{\cos^2(\theta_k)}\dot{\theta}_k \quad (5.27)$$

$$\dot{\theta}_k = \frac{\cos^2(\theta_k)}{\cos^2(\theta_i)}\dot{\theta}_i \quad (5.28)$$

5.4. Simulación numérica de la dinámica del robot

Para la implementación del modelo descrito por la ecuación 5.26 se ha desarrollado un programa basado en 9 scripts de MATLAB®, un script principal y 8 funciones auxiliares. El código de estos scripts se puede ver en el apéndice B. Se incluye ahora una descripción del software desarrollado.

5.4.1. Funcionamiento del programa

En la figura 5.5 se representa el diagrama de flujo del script principal del programa. En él, se puede observar cómo se interrelacionan las diferentes funciones entre sí.

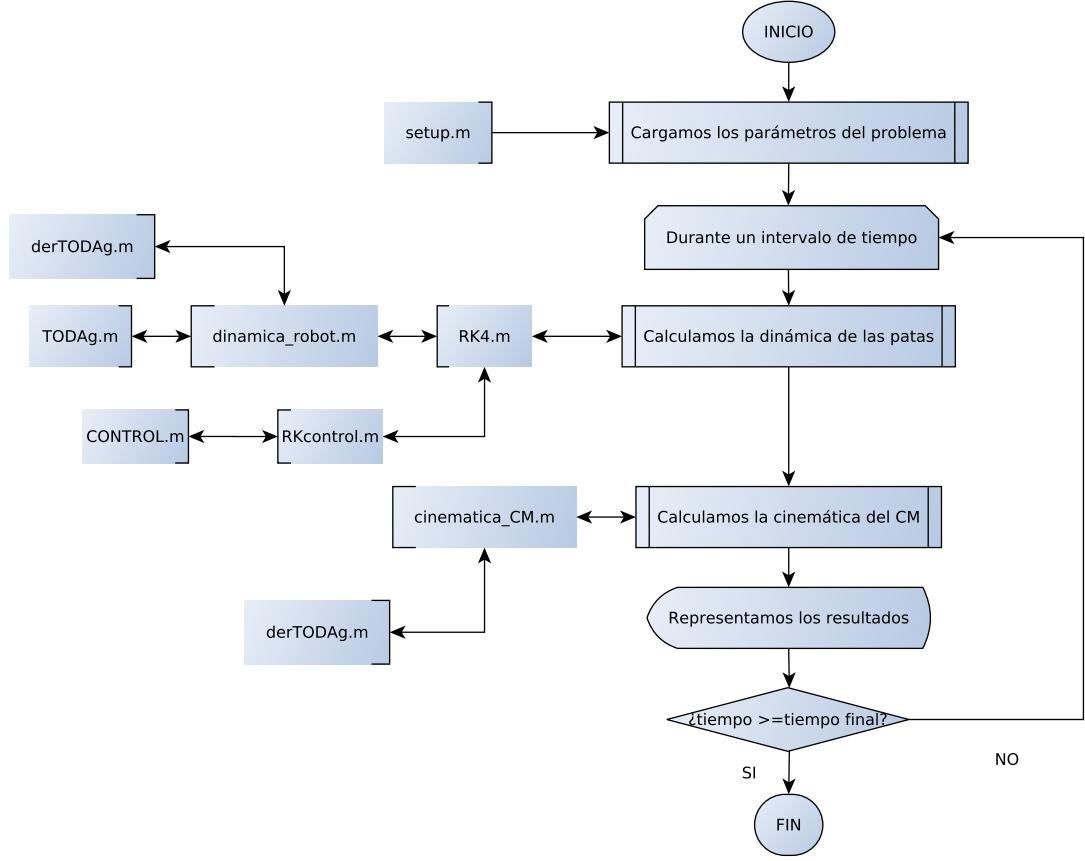


Figura 5.5: Diagrama de flujo del script main.m

La estructura del programa difiere ahora de la del código del modelo con momentos mecánicos constantes. En el anterior programa, la secuencia de levantamiento de las extremidades se imponía de forma externa, según el modo de movimiento analizado, y la integración se iba realizando por ciclos, integrando de forma numérica la dinámica de las patas apoyadas y de forma analítica la dinámica de las patas levantadas, ya que las ecuaciones estaban desacopladas. En este caso, las ecuaciones ya no se encuentran desacopladas y la integración se realiza de manera conjunta (ecuación 5.26) y de forma numérica mediante un esquema Runge-Kutta explícito de 4 etapas. Esta integración ya no se realiza por ciclos, sino que se realiza de forma continua durante un intervalo de tiempo determinado. Los diferentes modos de avance se consiguen en función de las señales generadas por el sistema de Toda-Rayleigh. Los diferentes parámetros de la función *TODAg.m* y las condiciones iniciales del problema definirán qué extremidades se encuentran levantadas en cada uno de los instantes del tiempo.

Analizando de manera más detallada el programa, éste comienza cargando las condiciones iniciales, los parámetros físicos del problema y los parámetros de la función que modela el sistema dinámico. A continuación se realiza la integración mediante un esquema Runge-Kutta de cuarto orden (*RK4.m*). Para realizar esta integración, la función *dinamica_robot.m* es la encargada, apoyándose en el script *TODAg.m* que contiene la función de Toda-Rayleigh, de realizar los cálculos necesarios para obtener el sistema de ecuaciones diferenciales de segundo orden 5.26 en cada uno de los instantes del tiempo. Una vez realizado los cálculos de la dinámica, se utilizan las ecuaciones 4.5,

4.6 y 4.7, contenidas en el script *cinematica_CM.m*, para calcular la cinemática del centro de masas del robot. En este script, se hace uso de la ecuación 5.19 para calcular el valor de las fuerzas de rozamiento que actúan sobre las extremidades. En el caso de que el sistema de control esté activado, previo a la llamada a *dinamica_robot.m* se realiza el cálculo del voltaje en el condensador de control (V_c) mediante otro esquema Runge-Kutta, apoyándose en las funciones *RKcontrol.m* y *CONTROL.m*. Finalizados todos los cálculos, se procede a la representación de los resultados obtenidos.

En este modelo físico, el sistema dinámico es el que controla, en todo momento, la secuencia de levantamiento de las extremidades, y, por lo tanto, los diferentes modos de movimiento, por lo que su correcta caracterización es crítica. Desde el punto de vista de la simulación numérica es fundamental identificar en todo momento qué extremidades se encuentran en la fase de vuelo, para aplicar correctamente las ecuaciones que modelan la dinámica del sistema. Esta identificación se realiza en la función *dinamica_robot.m*. La figura 5.6 muestra el diagrama de flujo de la función *dinamica_robot.m*, que es llamada en cada uno de los instantes del tiempo por la subrutina de integración *RK4.m*.

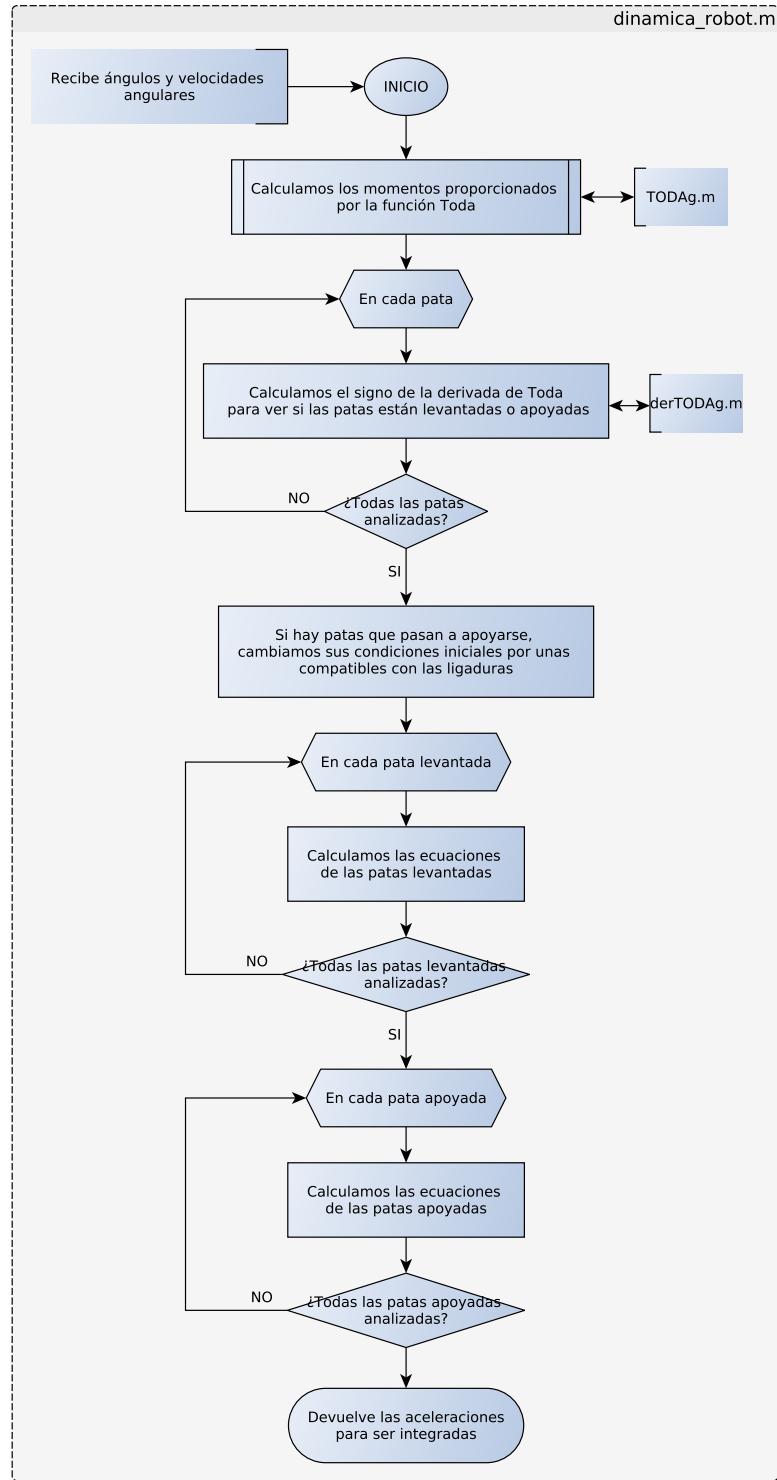


Figura 5.6: Diagrama de flujo del script dinamica_robot.m

En el comienzo de la subrutina, se calcula el valor de la función de Toda-Rayleigh para cada una de las extremidades, equivalente a un momento dividido por un momento de inercia ($\frac{\Phi I_{dc}}{J}$) con los valores de los ángulos y velocidades ángulares en ese instante de tiempo. Una vez realizado el cálculo de esta función, se puede proceder a la identificación de qué patas se encuentran en fase de vuelo y cuáles en fase de apoyo.

Para identificar si una extremidad se encuentra en fase de vuelo o no, se recurre al signo de la derivada de la función de Toda-Rayleigh. Relacionando la forma de esta función con el proceso de propagación de la oscilación se ha observado que, cuando la derivada de esta función es negativa, la extremidad está levantada, estando apoyada en el caso contrario. El signo de esta derivada se calcula utilizando la subrutina *derTODAg.m*, donde se ha implementado la derivada, analítica, de la función de Toda-Rayleigh. Como se ha explicado en el capítulo 2, la función de Toda-Rayleigh está directamente relacionada con la intensidad que circula por los motores de las extremidades. De cara a la implementación mecánica del sistema, se puede comprobar si una de las patas se encuentra en fase de vuelo o en fase de apoyo en función del aumento o la disminución de la intensidad que circula por su motor.

Una vez identificadas qué extremidades se encuentran en fase de vuelo y cuáles en fase de apoyo, se pueden aplicar las ecuaciones adecuadas en cada extremidad para conformar el sistema de ecuaciones 5.26.

5.4.2. Resultados de la simulación

Se presentan ahora los resultados de la simulación de los diferentes modos de movimiento del robot. Los parámetros básicos y geométricos relevantes del robot utilizados en la simulación son:

L	θ_0	m_p	M
0.3 m	35°	0.2 kg	3 kg

Tabla 5.1: Parámetros geométricos y básicos del robot

Con respecto a los parámetros propios de la función de Toda-Rayleigh, los valores utilizados han sido:

Tabla 5.2: Parámetros propios de la función de Toda-Rayleigh

R	ω	β	γ^*	μ^*	R_B	C	K
500 Ω	14.2470	0.7	$1.1250 \cdot 10^{-4}$	51.0260	500 Ω	0.03	15

* El valor de estos parámetros queda determinado por el valor de β y los parámetros propios de los componentes electrónicos

Los valores de estos parámetros se han seleccionado para que, en combinación con unas condiciones iniciales adecuadas y el valor del voltaje externo V_0 que aparece en la ecuación 5.1, se obtengan los diferentes modos de movimiento (*Metachronal*, *Caterpillar* y *Tripod*) de forma estable y con las oscilaciones de las funciones $\theta(t)$ centradas en el origen sin sobrepasar el ángulo de amplitud máxima θ_0 . A continuación se presentan los resultados de la simulación para los diferentes modos de avance o movimiento.

Modo *Metachronal*

Para conseguir la estabilización de este modo se han utilizado las siguientes condiciones iniciales para los ángulos y velocidades angulares:

θ_{01} (rad)	θ_{02} (rad)	θ_{03} (rad)	θ_{04} (rad)	θ_{05} (rad)	θ_{06} (rad)
-0.3954	-0.2313	-0.0639	0.0840	0.2751	0.2297

Tabla 5.3: Condiciones iniciales en las posiciones de las extremidades para el modo *Metachronal*

$\dot{\theta}_{01}$ (rad s ⁻¹)	$\dot{\theta}_{02}$ (rad s ⁻¹)	$\dot{\theta}_{03}$ (rad s ⁻¹)	$\dot{\theta}_{04}$ (rad s ⁻¹)	$\dot{\theta}_{05}$ (rad s ⁻¹)	$\dot{\theta}_{06}$ (rad s ⁻¹)
-0.1141	-0.1269	-0.1334	-0.1330	-0.1241	0.7164

Tabla 5.4: Condiciones iniciales en las velocidades angulares de las extremidades para el modo *Metachronal*

Además de estas condiciones iniciales, se ha activado el sistema de control automático con un valor de voltaje de -1:

C_r	V_0 (V)
1	-1

Tabla 5.5: Voltaje externo para estabilizar el modo *Metachronal*

Se muestran a continuación los diferentes resultados de la simulación.

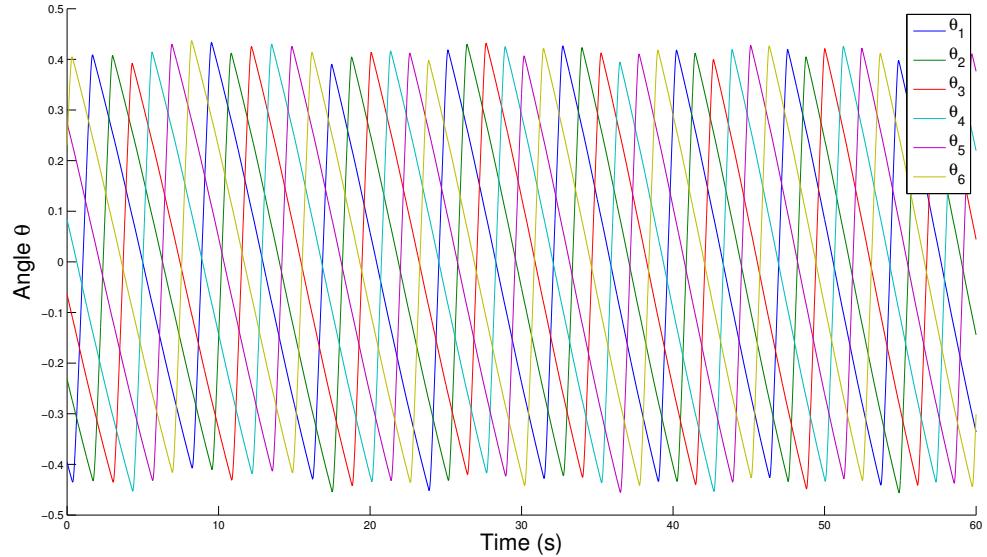


Figura 5.7: Funciones $\theta(t)$ resultantes de la simulación del modo *Metachronal*

En la figura 5.7 se pueden observar las funciones $\theta(t)$ que resultan de la integración del sistema de ecuaciones 5.26 para el modo *Metachronal*. Se observa cómo estas funciones se corresponden con las de la figura 1.4.A.

Las figuras 5.8 y 5.7 muestran, respectivamente, la velocidad y aceleración de las diferentes extremidades el robot.

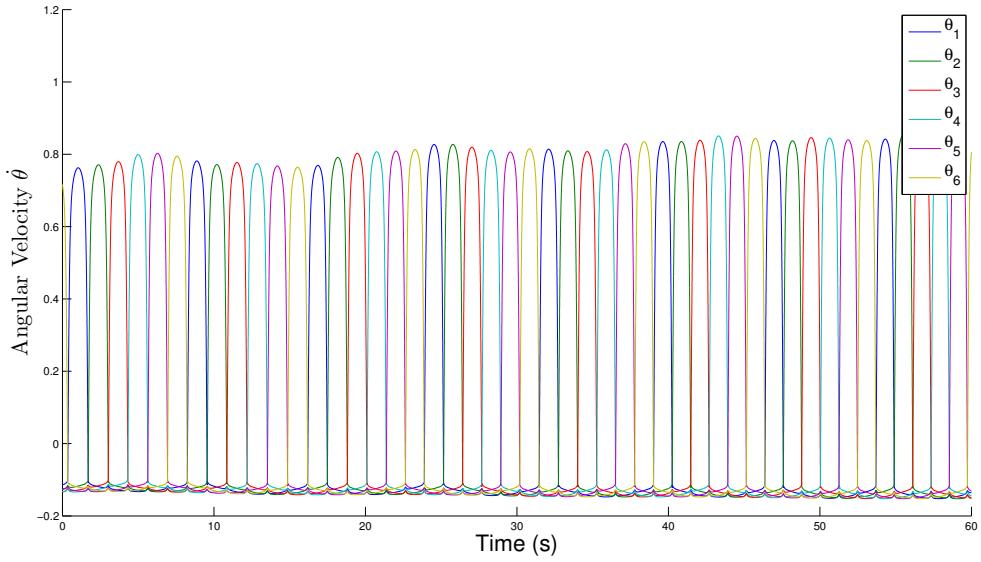


Figura 5.8: Funciones $\dot{\theta}(t)$ resultantes de la simulación del modo *Metachronal*

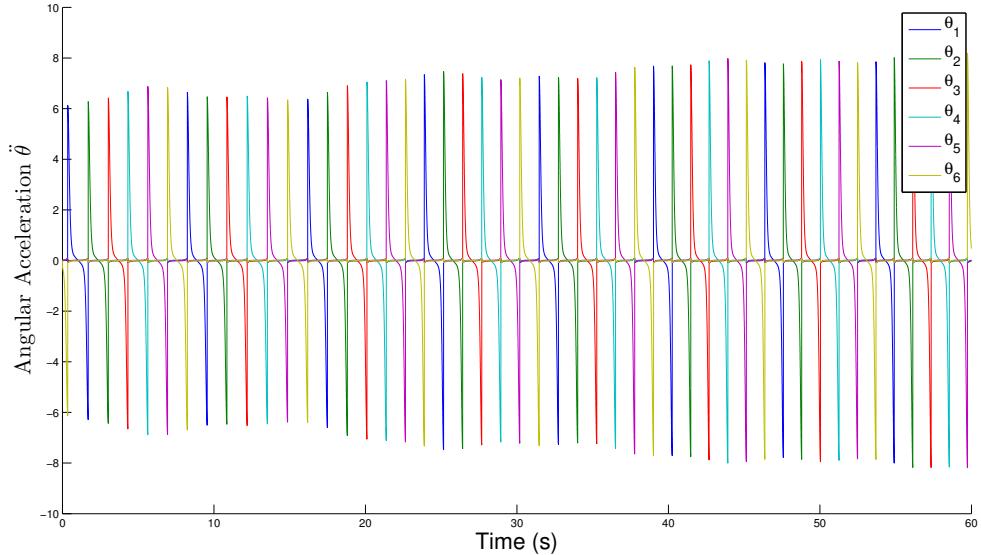


Figura 5.9: Funciones $\ddot{\theta}(t)$ resultantes de la simulación del modo *Metachronal*

Como se ha comentado anteriormente, la identificación del estado de las patas (apoyadas o en fase de vuelo) es crítica a la hora de modelizar el comportamiento del robot, puesto que el sistema de ecuaciones 5.26 varía en función de en qué estado se encuentren las extremidades. Para asegurar que los resultados obtenidos se correspondan, efectivamente, con el modo de avance *Metachronal*, se representa, en función del tiempo, el estado de las extremidades. Este estado se ha representado en la figura 5.10 mediante 0, en el caso de que las patas estén apoyadas, o mediante el número asignado a cada una de las patas en la figura 5.4 si éstas se encuentran en fase de vuelo.

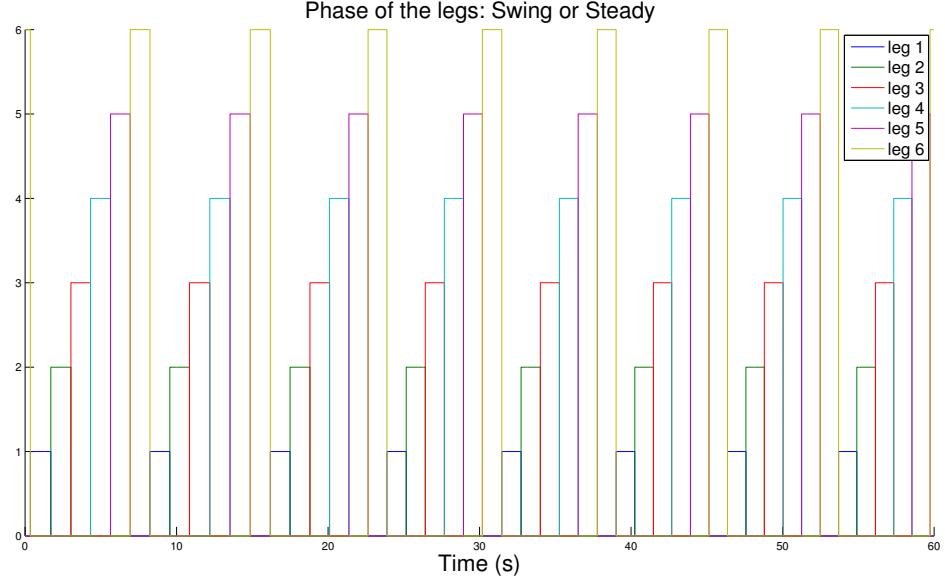


Figura 5.10: Estado de las patas en función del tiempo: 0 para la fase de apoyo o i para la fase de vuelo de la pata i para el modo *Metachronal*

En la figura 5.10 la secuencia de levantamiento-apoyo se corresponde con la del modo *Metachronal*, donde la onda se propaga desde atrás hacia adelante y de izquierda a derecha, siguiendo el patrón: <1>,<2>,<3>,<4>,<5>,<6>.

Se representa en la figura 5.11 los momentos mecánicos ΦI_{dc} introducidos por los motores en las extremidades. Estos momentos son muy próximos a 0 excepto en los alrededores de los instantes de apoyo/levantamiento de las patas, donde se induce un pico de momento positivo para levantar la pata y otro de momento negativo para apoyar la extremidad que se encontraba en la fase de vuelo.

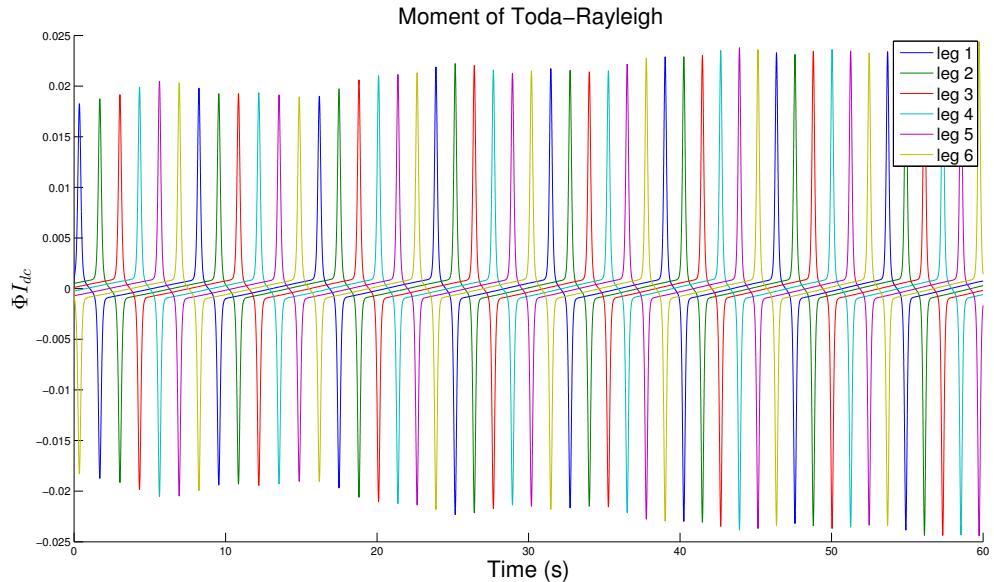


Figura 5.11: Momentos introducidos por el sistema de Toda-Rayleigh para el modo *Metachronal*

Después de representar los resultados individuales para cada una de las extremidades del robot, se representa la cinemática del centro de masas del robot, calculada mediante las ecuaciones 4.5 y 4.6.

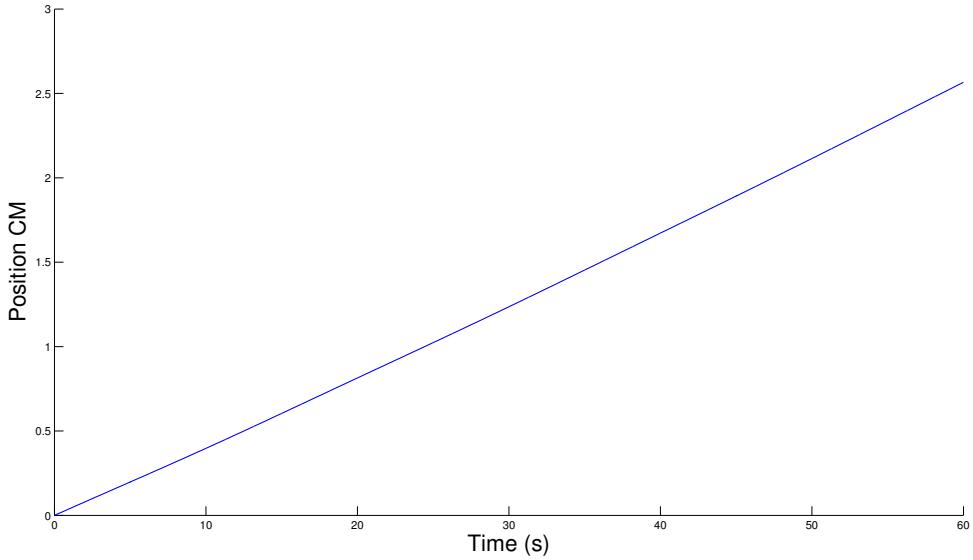


Figura 5.12: Posición del centro de masas del robot para el modo de avance *Metachronal*

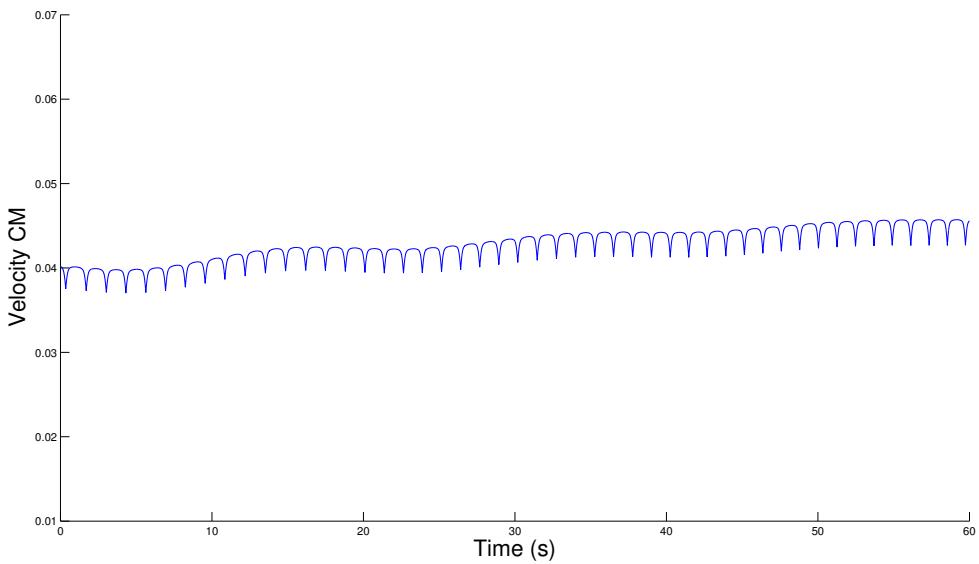


Figura 5.13: Velocidad del centro de masas del robot para el modo de avance *Metachronal*

El movimiento del centro de masas se corresponde, atendiendo a la figura 5.12 con un movimiento de velocidad uniforme. Aunque la figura 5.13, no muestre una velocidad exactamente constante con el tiempo, sino que se observan claramente las zonas de transición entre las fases de apoyo y levantamiento, esta velocidad sí se puede representar por un valor medio aproximadamente constante.

Por último, se representa el valor del voltaje V_{sh} proporcionado por el sistema de control automático. Debido a que se parte de unas condiciones iniciales y un valor de V_0 que estabilizan el modo *Metachronal* y que no existen acciones exteriores que perturben la dinámica del robot, el valor de V_{sh} se mantiene, en todo momento, muy próximo al valor de V_0 introducido, tal y como se puede observar en la figura 5.14.

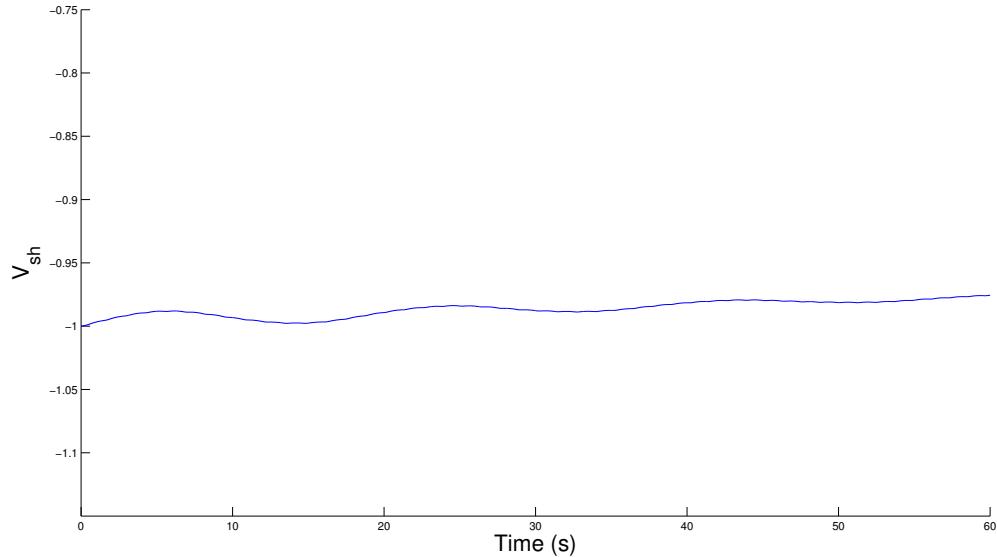


Figura 5.14: Valor de V_{sh} para el modo *Metachronal*

Modo *Caterpillar*

Para obtener la dinámica del modo de avance *Caterpillar* se ha utilizado la siguiente combinación de condiciones iniciales y valor de voltaje externo V_0 , estando nuevamente activado el sistema de control automático.

θ_{01} (rad)	θ_{02} (rad)	θ_{03} (rad)	θ_{04} (rad)	θ_{05} (rad)	θ_{06} (rad)
0.1561	-0.2774	0.1256	0.1561	-0.2774	0.1256

Tabla 5.6: Condiciones iniciales en las posiciones de las extremidades para el modo *Caterpillar*

$\dot{\theta}_{01}$ (rad s $^{-1}$)	$\dot{\theta}_{02}$ (rad s $^{-1}$)	$\dot{\theta}_{03}$ (rad s $^{-1}$)	$\dot{\theta}_{04}$ (rad s $^{-1}$)	$\dot{\theta}_{05}$ (rad s $^{-1}$)	$\dot{\theta}_{06}$ (rad s $^{-1}$)
0.8301	-0.3502	-0.3726	0.8301	-0.3502	-0.3726

Tabla 5.7: Condiciones iniciales en las velocidades angulares de las extremidades para el modo *Caterpillar*

C_r	V_0 (V)
1	-0.5

Tabla 5.8: Voltaje externo para estabilizar el modo *Caterpillar*

En las siguientes figuras se representan los resultados de la simulación de este modo de avance, comenzando nuevamente por las funciones $\theta(t)$. Cabe destacar que las simetrías de este modo producen que se superpongan los resultados de las extremidades que se mueven con la misma fase en las diferentes representaciones.

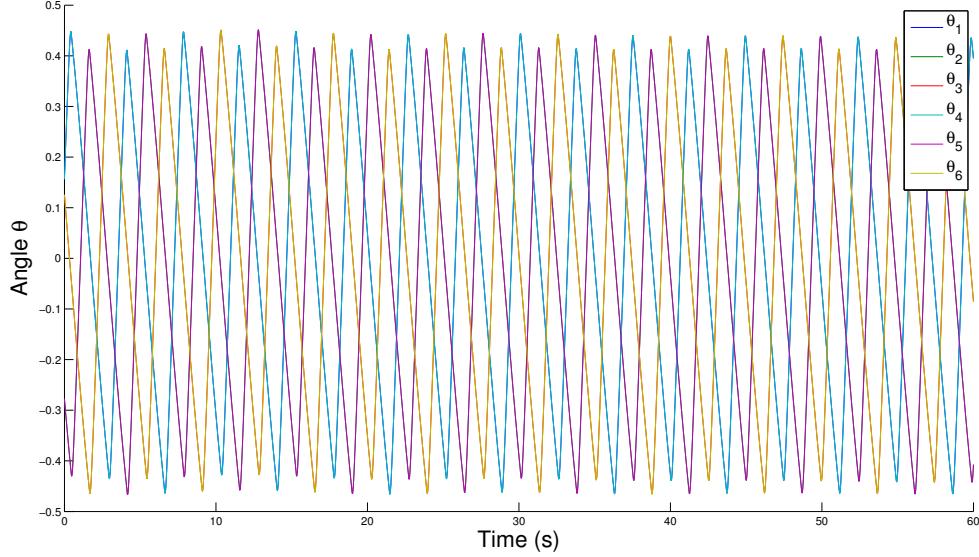


Figura 5.15: Funciones $\theta(t)$ resultantes de la simulación del modo *Caterpillar*

Se observa nuevamente cómo las funciones $\theta(t)$ de la figura 5.15 se corresponden con las de la figura 1.4.B.

A continuación se han representado las velocidades angulares (5.16) y aceleraciones angulares de las extremidades (5.17).

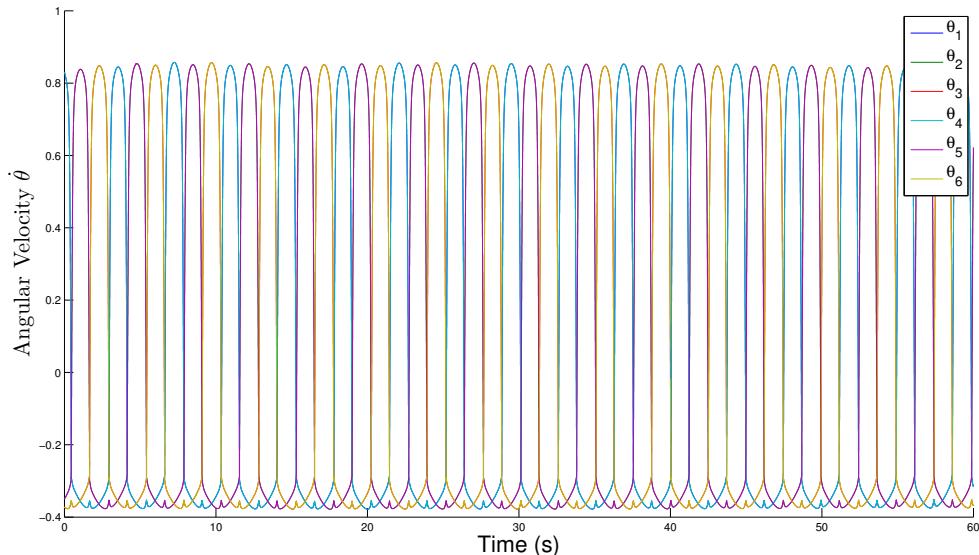


Figura 5.16: Funciones $\dot{\theta}(t)$ resultantes de la simulación del modo *Caterpillar*

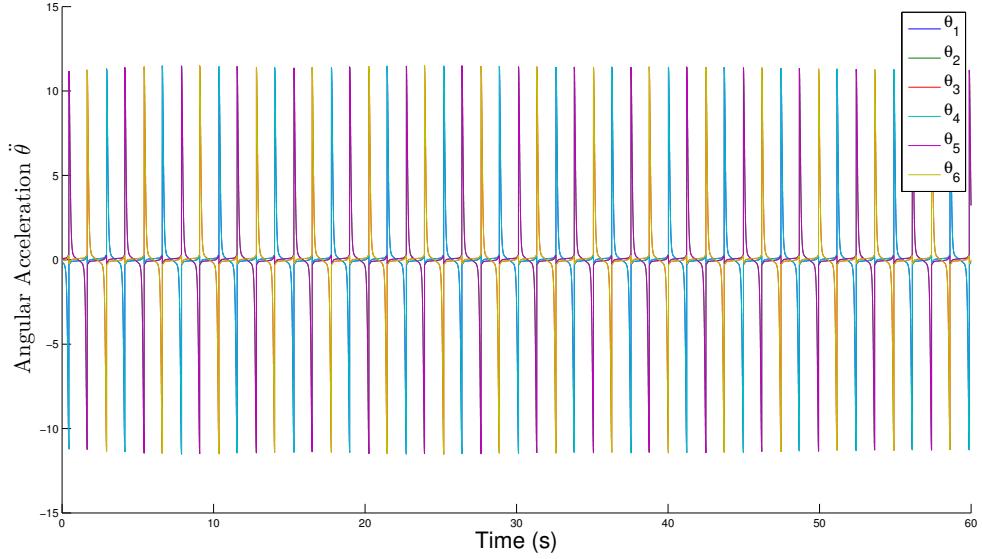


Figura 5.17: Funciones $\ddot{\theta}(t)$ resultantes de la simulación del modo *Caterpillar*

En la figura 5.18 se observa cómo la secuencia de levantamiento de las extremidades se corresponde con la del modo *Caterpillar*, donde en cada instante de tiempo se encuentran dos extremidades levantadas siguiendo la secuencia $<1,4>$, $<2,5>$, $<3,6>$.



Figura 5.18: Estado de las patas en función del tiempo: 0 para la fase de apoyo o i para la fase de vuelo de la pata i para el modo *Caterpillar*

Los momentos introducidos por el sistema de Toda-Rayleigh son similares a los del caso anterior.

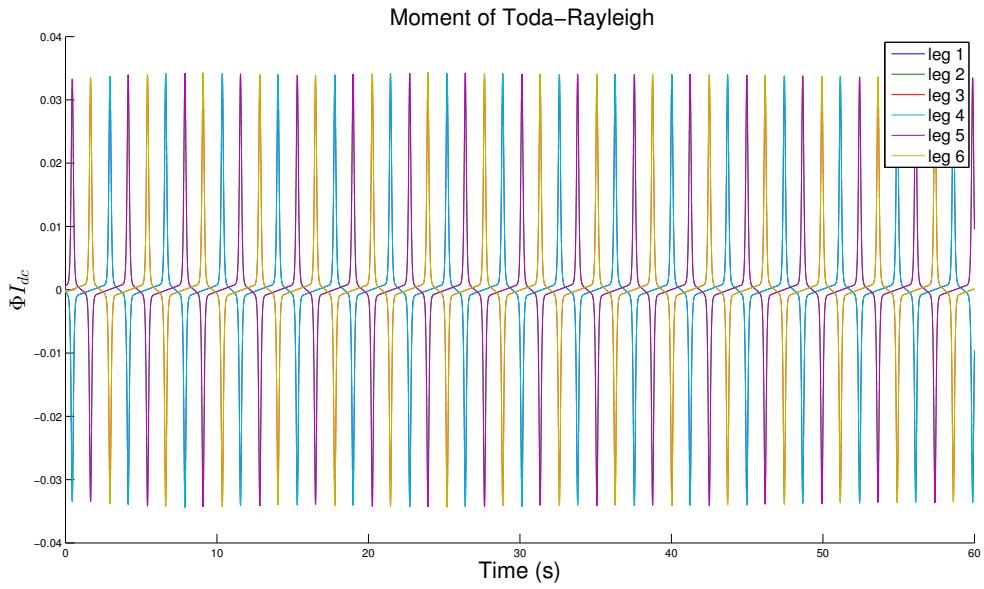


Figura 5.19: Momentos introducidos por el sistema de Toda-Rayleigh para el modo *Caterpillar*

Nuevamente, nos encontramos con un movimiento uniforme en cuanto a la cinemática del centro de masas, observando en la velocidad pequeñas oscilaciones en las fases de transición.

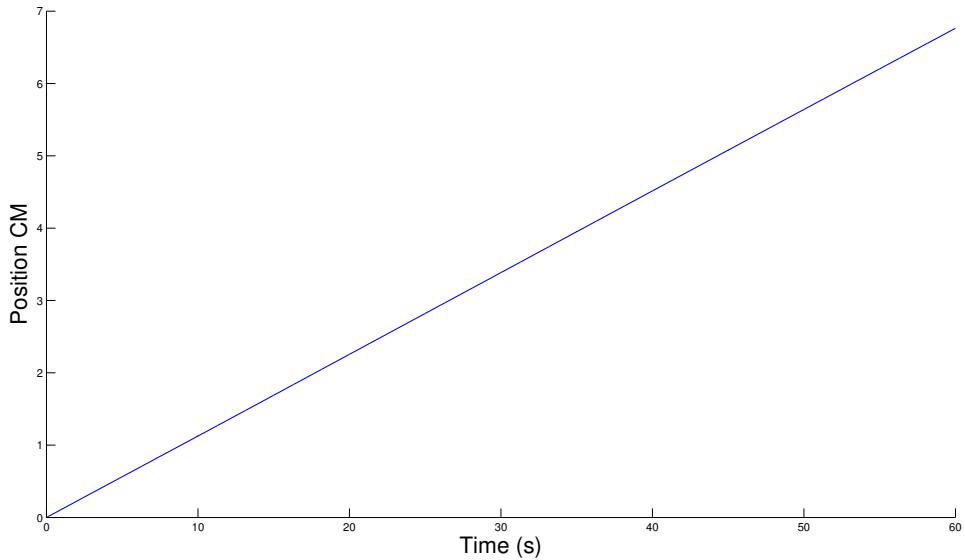


Figura 5.20: Posición del centro de masas del robot para el modo de avance *Caterpillar*

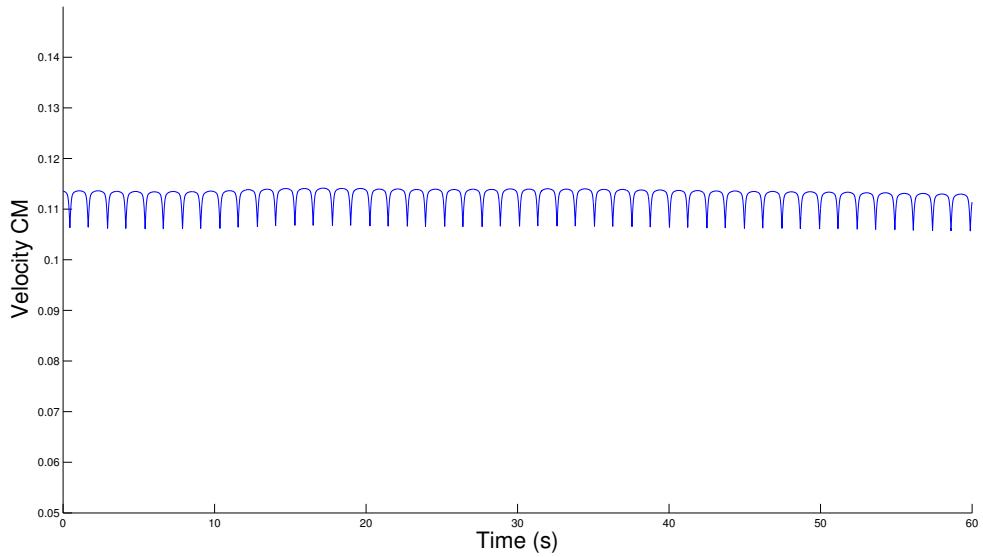


Figura 5.21: Velocidad del centro de masas del robot para el modo de avance *Caterpillar*

Se observa cómo, en este modo, la velocidad es mayor que en el modo de avance *Metachronal* (aproximadamente $0.115 \text{ m}\cdot\text{s}^{-1}$ frente a $0.042 \text{ m}\cdot\text{s}^{-1}$ del modo *Metachronal*), tal y como se explicó en la sección 1.2.

Como era de esperar, debido a la ausencia de perturbaciones exteriores, el valor de V_{sh} introducido por el sistema de control se mantiene muy próximo al valor de V_0 impuesto, ya que este es muy próximo al valor que estabiliza este modo en torno al 0.

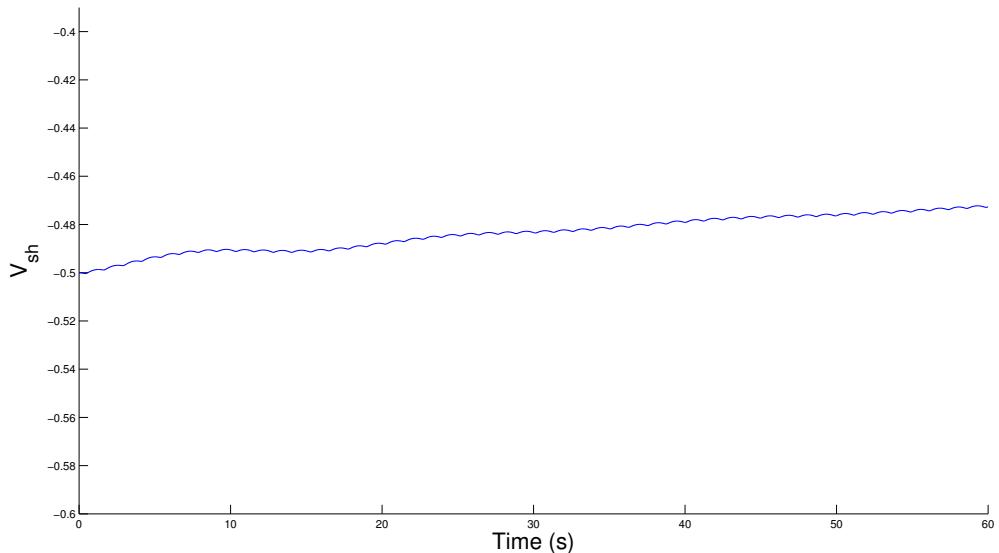


Figura 5.22: Valor de V_{sh} para el modo *Caterpillar*

Modo *Tripod*

El último modo de avance por analizar es el modo *Tripod*. Las condiciones iniciales para generar este modo de forma estable utilizadas y el valor de V_0 han sido:

θ_{01} (rad)	θ_{02} (rad)	θ_{03} (rad)	θ_{04} (rad)	θ_{05} (rad)	θ_{06} (rad)
-0.1657	0.1390	-0.1657	0.1390	-0.1657	0.1390

Tabla 5.9: Condiciones iniciales en las posiciones de las extremidades para el modo *Tripod*

$\dot{\theta}_{01}$ (rad s $^{-1}$)	$\dot{\theta}_{02}$ (rad s $^{-1}$)	$\dot{\theta}_{03}$ (rad s $^{-1}$)	$\dot{\theta}_{04}$ (rad s $^{-1}$)	$\dot{\theta}_{05}$ (rad s $^{-1}$)	$\dot{\theta}_{06}$ (rad s $^{-1}$)
-0.7282	0.8531	-0.7282	0.8531	-0.7282	0.8531

Tabla 5.10: Condiciones iniciales en las velocidades angulares de las extremidades para el modo *Tripod*

C_r	V_0 (V)
1	0.0331

Tabla 5.11: Voltaje externo para estabilizar el modo *Tripod*

Presentamos ahora en la figura 5.23 las funciones $\theta(t)$, que coinciden, una vez más, con las presentadas en la figura 1.4.C.

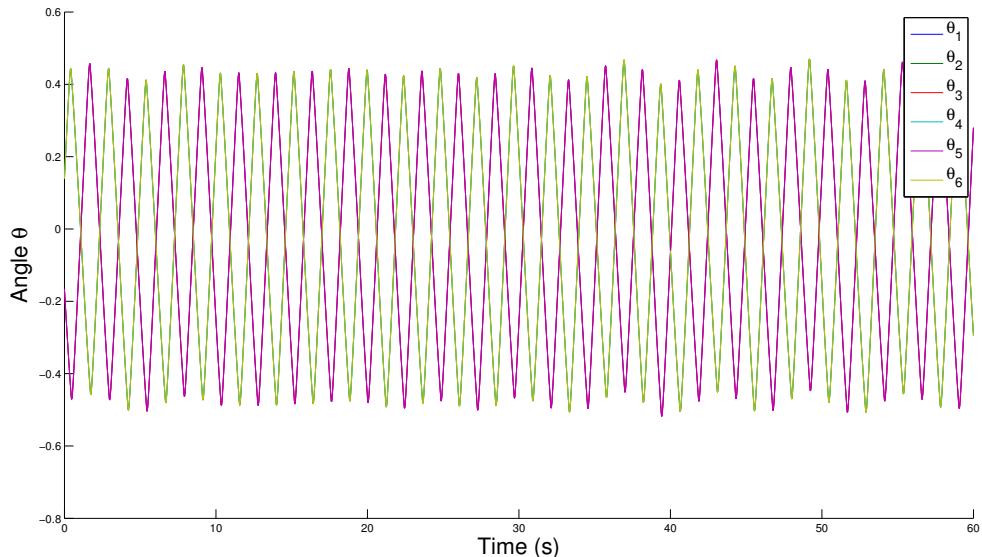


Figura 5.23: Funciones $\theta(t)$ resultantes de la simulación del modo *Tripod*

Se representan ahora las demás variables cinemáticas de las extremidades: las velocidades angulares (5.24) y aceleraciones angulares (5.25).

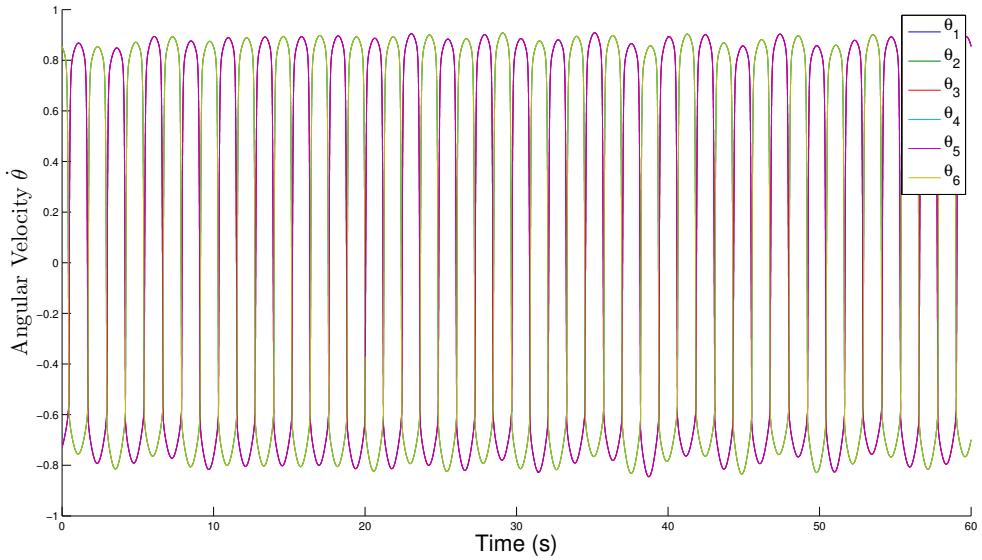


Figura 5.24: Funciones $\dot{\theta}(t)$ resultantes de la simulación del modo *Tripod*

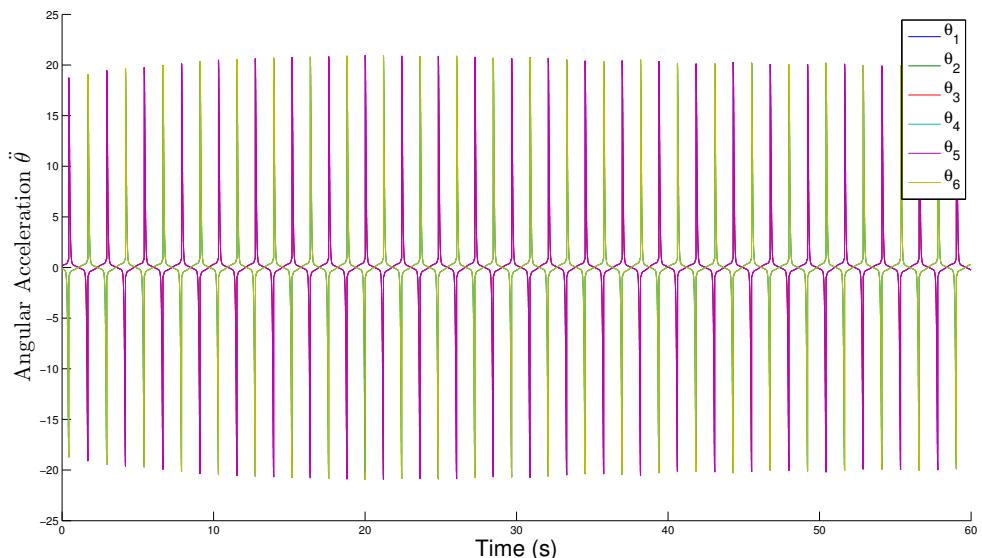


Figura 5.25: Funciones $\ddot{\theta}(t)$ resultantes de la simulación del modo *Tripod*

Cabe destacar, nuevamente, que las simetrías de este modo hacen que se superpongan las soluciones, observándose únicamente en la representación una de las soluciones para las patas apoyadas y otra para las patas en fase de vuelo.

La figura 5.26 muestra cómo las extremidades siguen la secuencia de levantamiento esperada: $\langle 1,3,5 \rangle$, $\langle 2,4,6 \rangle$.

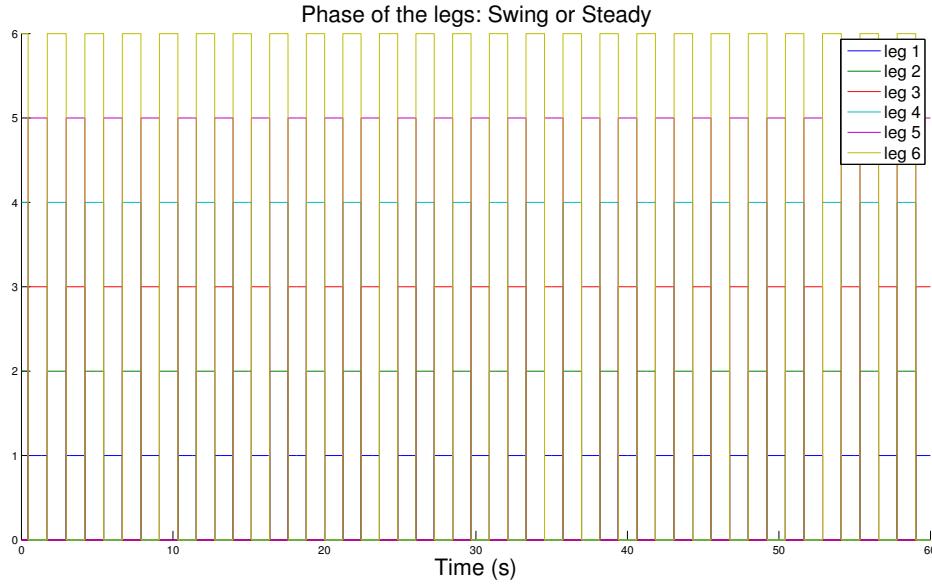


Figura 5.26: Estado de las patas en función del tiempo: 0 para la fase de apoyo o i para la fase de vuelo de la pata i para el modo *Tripod*

Los momentos introducidos por los motores son similares a los anteriores, pero de mayor valor máximo.

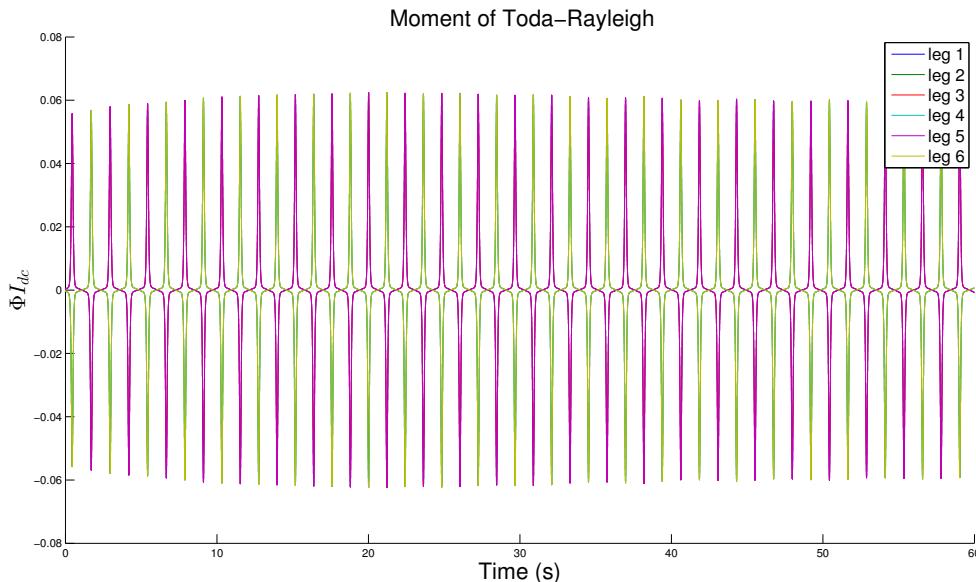


Figura 5.27: Momentos introducidos por el sistema de Toda-Rayleigh para el modo *Tripod*

En cuanto a la cinemática del centro de masas, nos encontramos con un movimiento de velocidad aproximadamente uniforme (al menos se puede aproximar por un valor medio constante), tal y como se ven en las figuras 5.28 y 5.29.

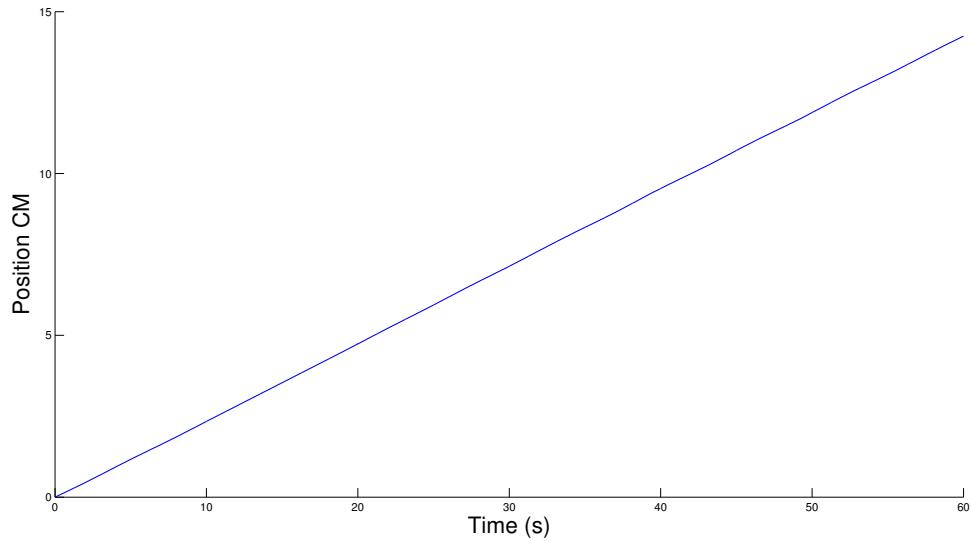


Figura 5.28: Posición del centro de masas del robot para el modo de avance *Tripod*

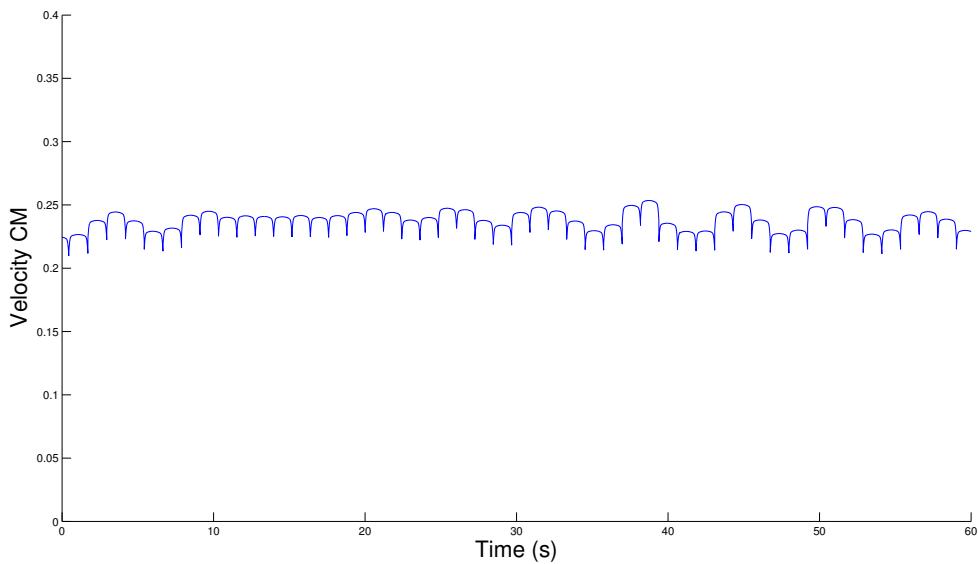


Figura 5.29: Velocidad del centro de masas del robot para el modo de avance *Tripod*

Se puede comprobar cómo, de acuerdo con lo esperado, la velocidad de avance de este modo es la más alta de entre los tres estudiados.

Como último resultado, se representa el valor del voltaje V_{sh} proporcionado por el sistema de control automático, cuyas variaciones con respecto al valor V_0 impuesto no son muy grandes.

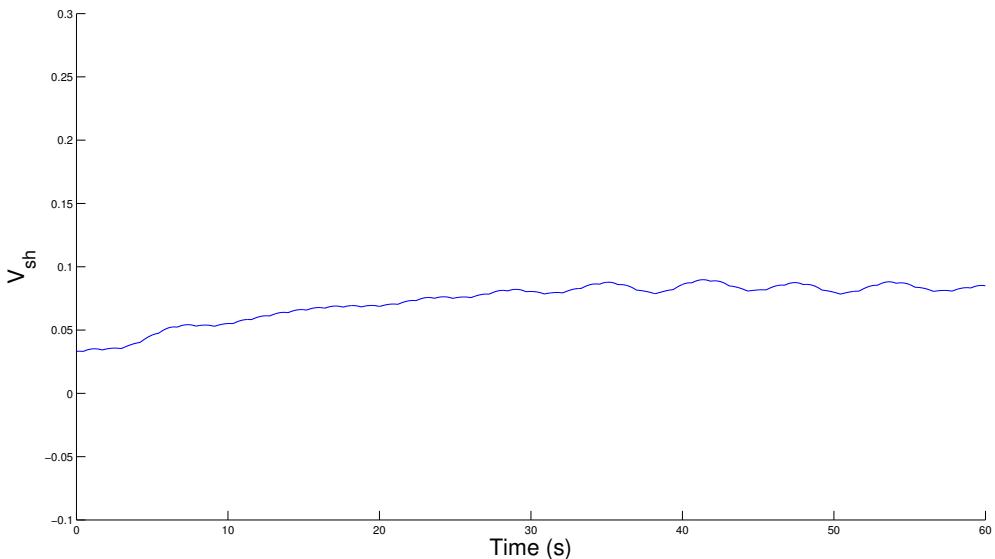


Figura 5.30: Valor de V_{sh} para el modo *Tripod*

5.5. Discusión de los resultados obtenidos en la simulación

Los resultados obtenidos en la sección 5.4.2 prueban que es posible utilizar el sistema de Toda-Rayleigh propuesto como CPG para generar la locomoción de un robot hexápedo.

Una de las hipótesis realizadas para la obtención de estos resultados ha sido que la fuerza de rozamiento que actúa sobre cada una de las extremidades apoyadas es lo suficiente grande como para mantener el apoyo fijo al suelo sin deslizar. En las figuras 5.31, 5.32 y 5.33 se ha representado la evolución de dichas fuerzas con el tiempo para los diferentes modos de movimiento.

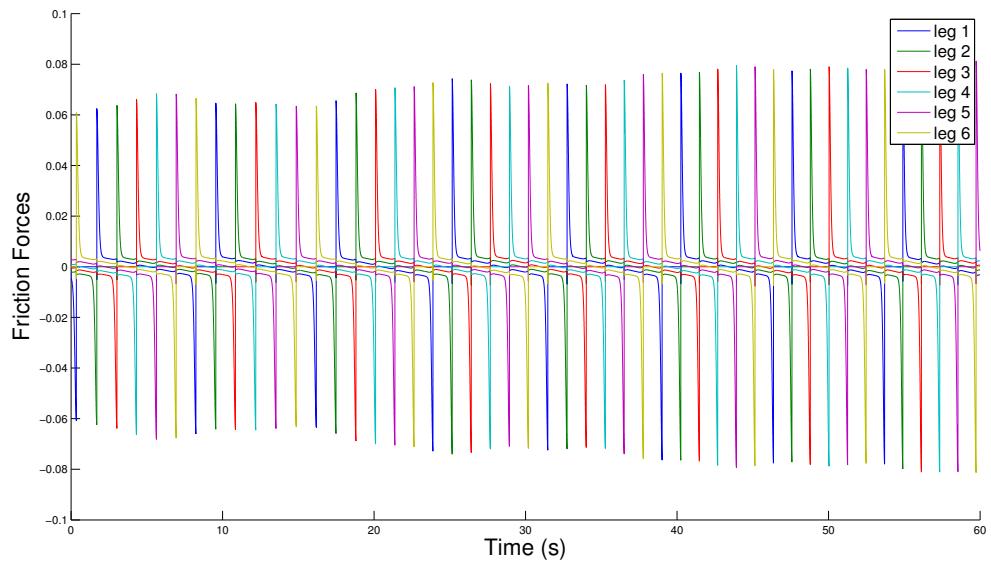


Figura 5.31: Fuerzas de rozamiento para el modo *Metachronal*

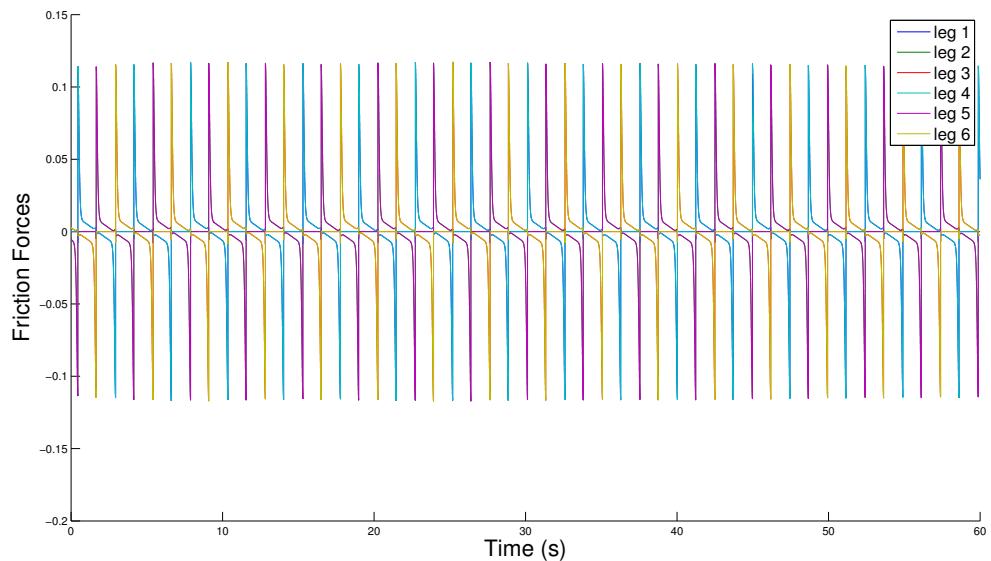


Figura 5.32: Fuerzas de rozamiento para el modo *Caterpillar*

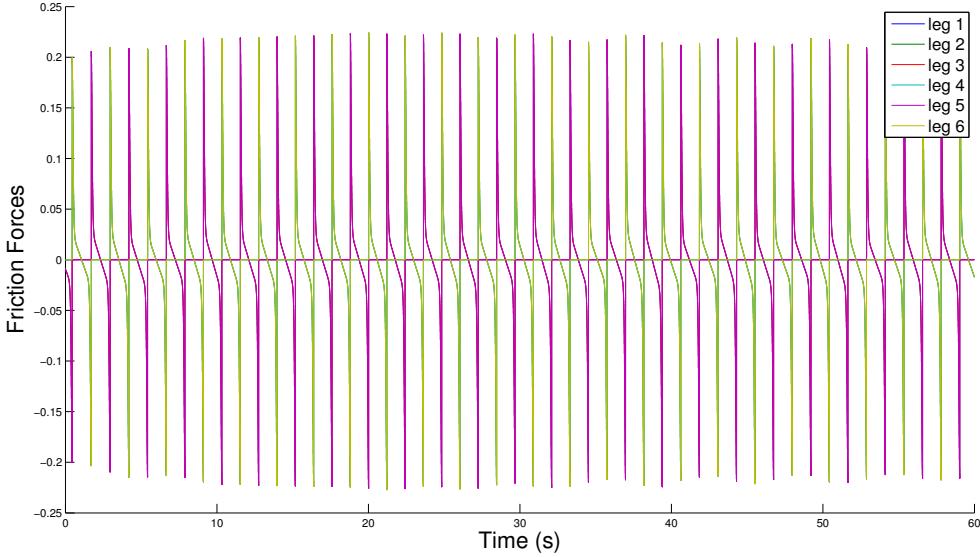


Figura 5.33: Fuerzas de rozamiento para el modo *Tripod*

En las figuras presentadas se observa claramente cómo las patas sufren una percusión en los instantes de apoyo. Esta percusión inicial podría hacer que las patas deslicen en el instante de apoyo, algo que no está recogido en la modelización física.

Otra de las hipótesis realizadas en la modelización del movimiento del robot es que las patas poseen rigidez infinita, algo que no puede conseguirse en la construcción del robot.

A pesar de las hipótesis realizadas en la modelización matemática, debido a la robustez de las soluciones del sistema de Toda-Rayleigh, se espera que los resultados obtenidos en una futura implantación del sistema en un robot sean muy similares a los obtenidos.

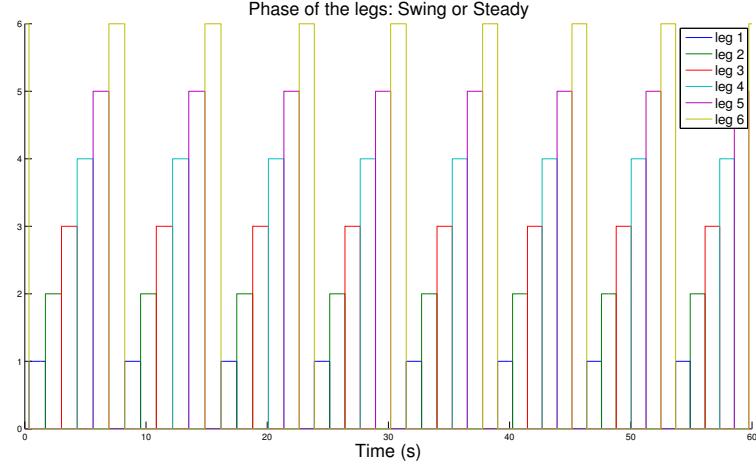
A fin de comprobar esto, se modifica una de las partes más críticas del software: la identificación de la fase de movimiento de las extremidades mediante la derivada de la función de Toda-Rayleigh. Como se ha comentado anteriormente, la identificación del estado de las patas determina el sistema de ecuaciones a resolver para obtener la dinámica del robot. En vez de utilizar la derivada analítica de la función, se aproxima el signo de esta derivada simplemente mediante la resta de los valores de la función en dos instantes del tiempo consecutivos.

$$\operatorname{sgn} \left(\frac{d}{dt} \left(\frac{V_0}{J} \Phi I_{dc} \right) \Big|_{t_j} \right) \approx \operatorname{sgn} \left(\left. \frac{V_0}{J} \Phi I_{dc} \right|_{t_j} - \left. \frac{V_0}{J} \Phi I_{dc} \right|_{t_{j-1}} \right) \quad (5.29)$$

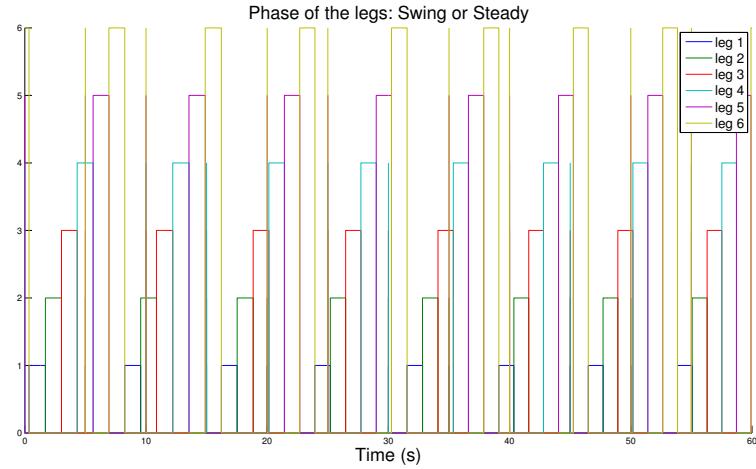
El efecto del error introducido por esta aproximación a la hora de identificar el estado de las extremidades puede utilizarse para predecir el comportamiento del sistema real una vez construido.

A continuación, se muestran los resultados de esta nueva simulación para los diferentes modos de movimiento.

En la figura 5.34 se observa cómo la aproximación de la derivada introduce errores en la identificación del estado de las patas en comparación con la derivada analítica.



(a) Estado de las patas utilizando la derivada analítica



(b) Estado de las patas approximando la derivada

Figura 5.34: Comparación de las simulaciones para el modo *Metachronal*

A pesar de estos errores, los resultados obtenidos de la integración de la dinámica del robot siguen siendo similares a los deseados.

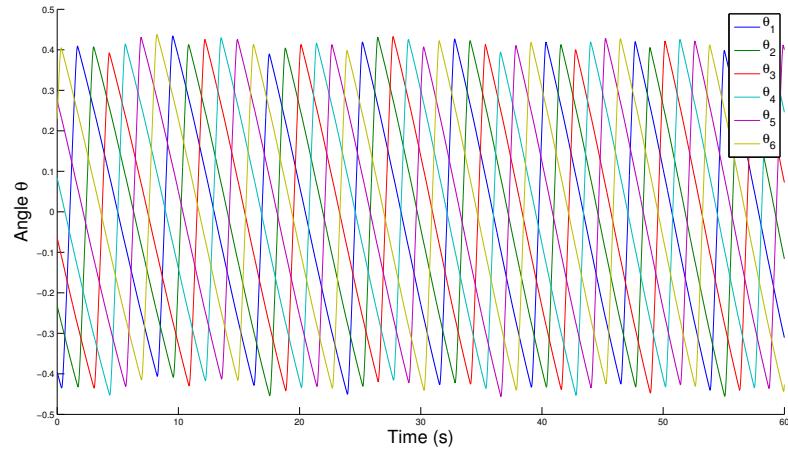


Figura 5.35: Funciones $\theta(t)$ para el modo *Metachronal*

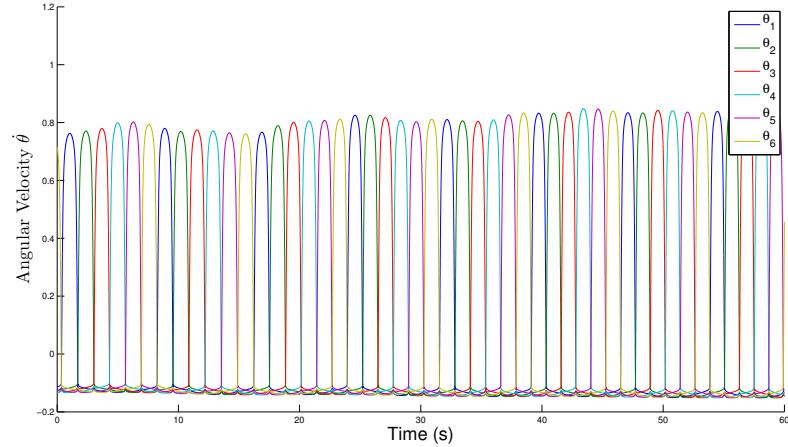


Figura 5.36: Funciones $\dot{\theta}(t)$ para el modo *Metachronal*

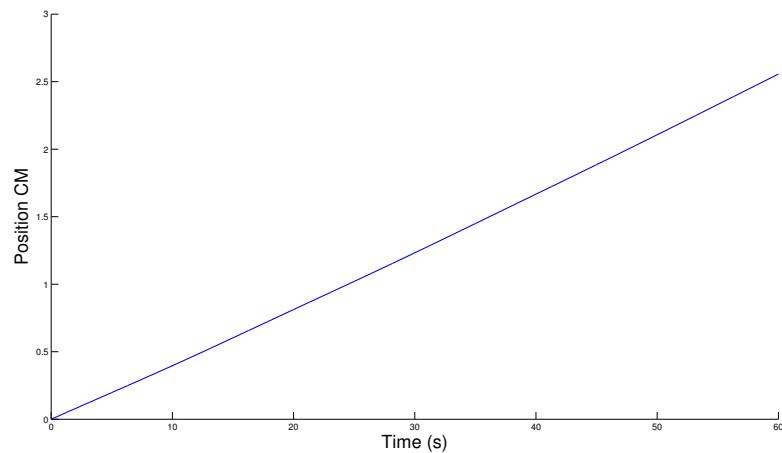


Figura 5.37: Posición del centro de masas para el modo *Metachronal*

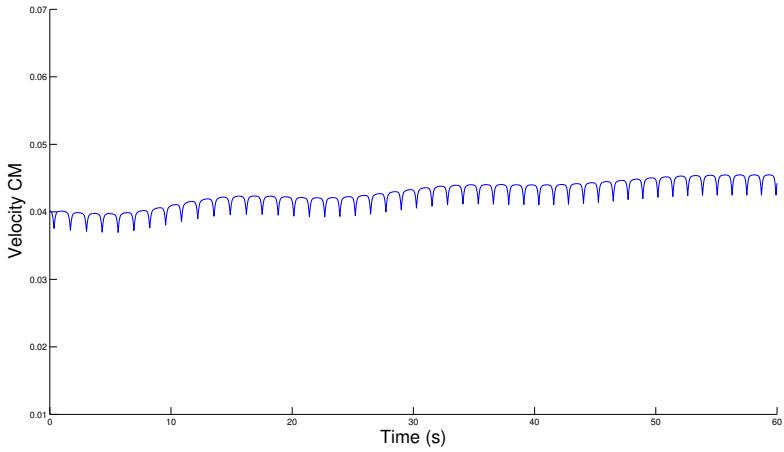
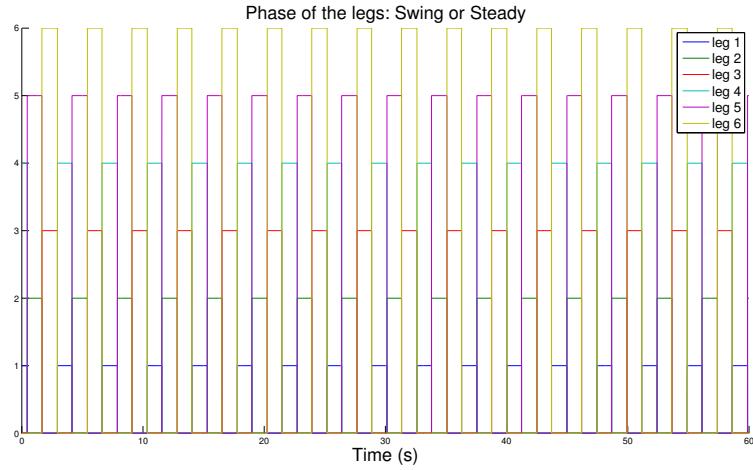
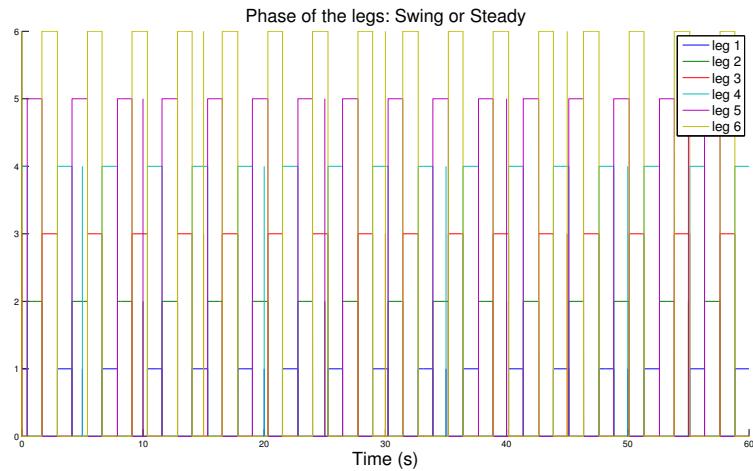


Figura 5.38: Velocidad del centro de masas para el modo *Metachronal*

Los resultados para los otros modos de movimiento son similares.



(a) Estado de las patas utilizando la derivada analítica



(b) Estado de las patas aproximando la derivada

Figura 5.39: Comparación de las simulaciones para el modo *Caterpillar*

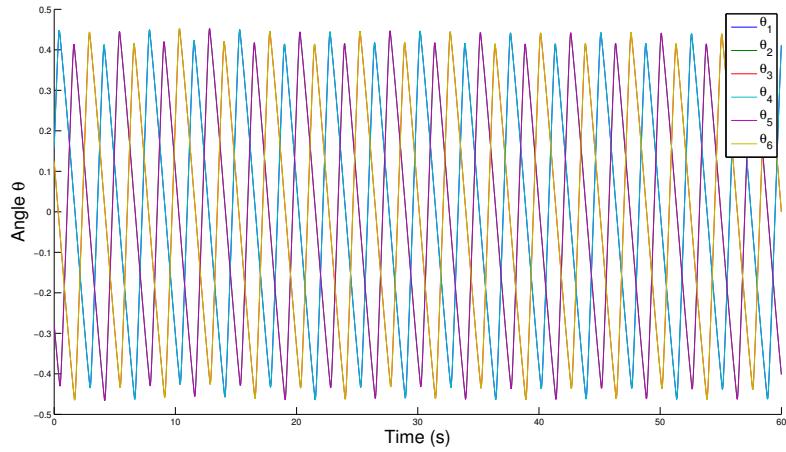


Figura 5.40: Funciones $\theta(t)$ para el modo *Caterpillar*

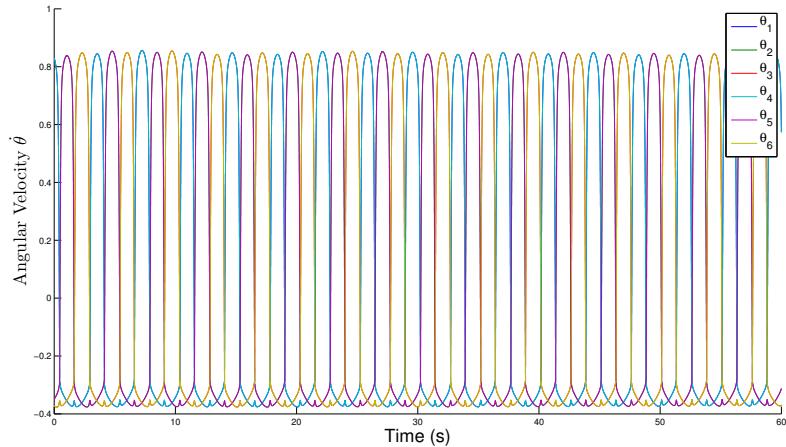


Figura 5.41: Funciones $\dot{\theta}(t)$ para el modo *Caterpillar*

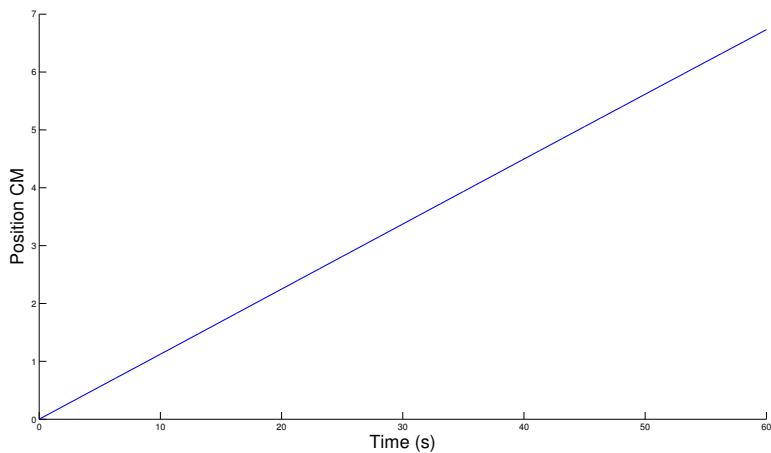


Figura 5.42: Posición del centro de masas para el modo *Caterpillar*

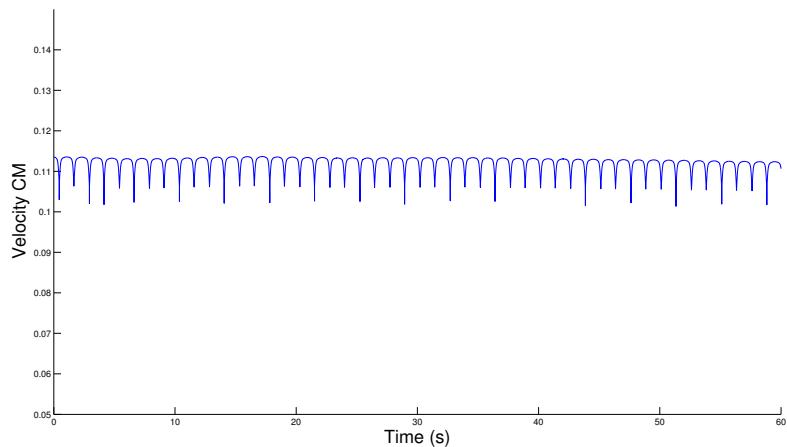


Figura 5.43: Velocidad del centro de masas para el modo *Caterpillar*

En el caso del modo *Tripod*, no se aprecian grandes diferencias con esta aproximación:

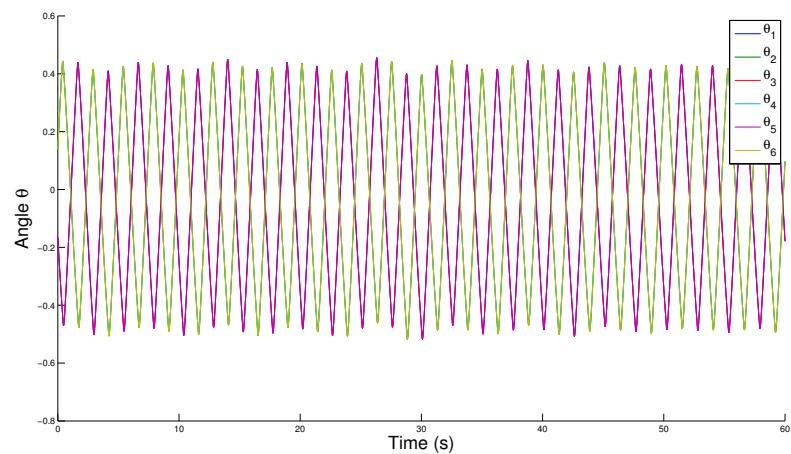


Figura 5.45: Funciones $\theta(t)$ para el modo *Tripod*



(a) Estado de las patas utilizando la derivada analítica



(b) Estado de las patas aproximando la derivada

Figura 5.44: Comparación de las simulaciones para el modo *Tripod*

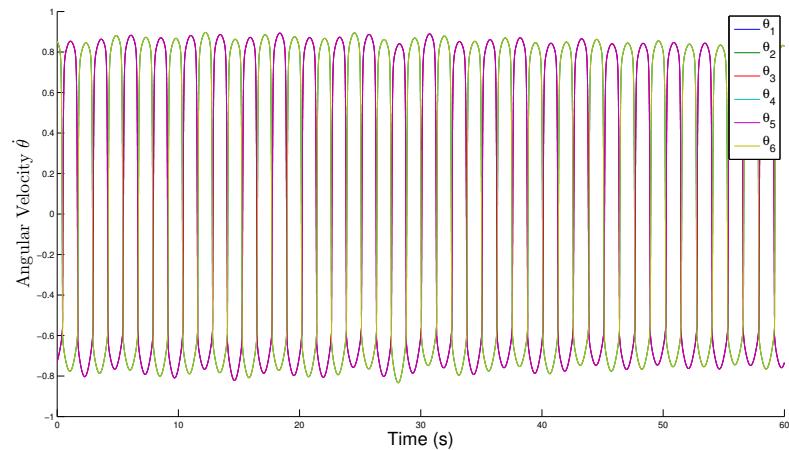


Figura 5.46: Funciones $\dot{\theta}(t)$ para el modo *Tripod*

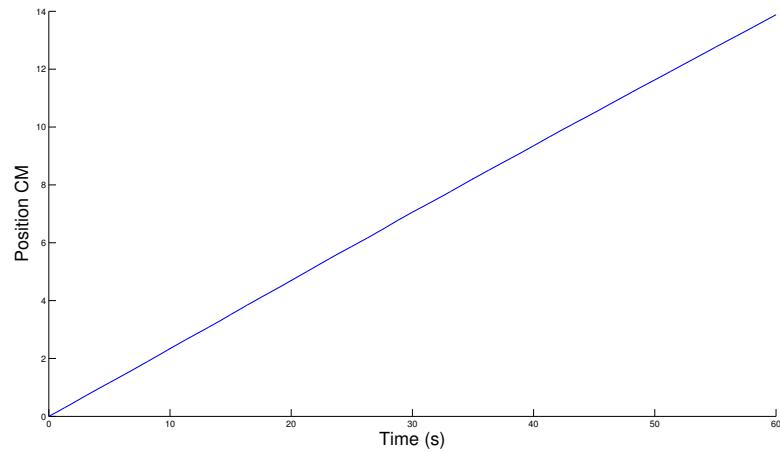


Figura 5.47: Posición del centro de masas para el modo *Tripod*

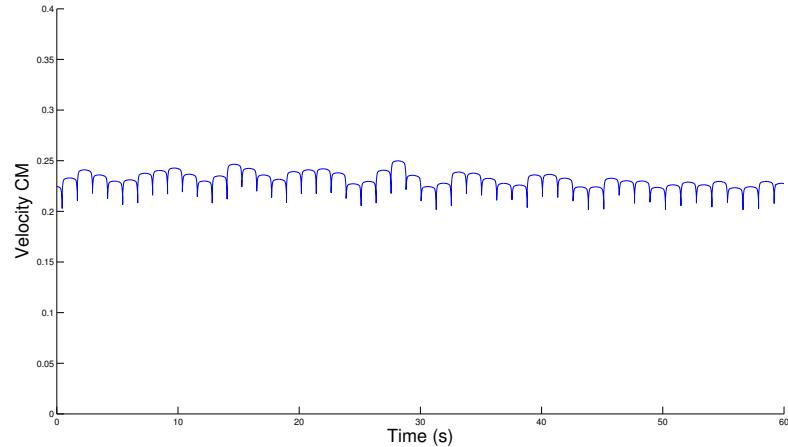


Figura 5.48: Velocidad del centro de masas para el modo *Tripod*

A pesar de los errores introducidos en el modelo por esta aproximación, se consiguen generar las funciones $\theta(t)$ necesarias para la locomoción del robot con los modos de movimiento analizados. Los errores introducidos se acumulan, sobre todo en los instantes de transición entre las fases de vuelo y apoyo, puesto que no siempre se siguen exactamente las secuencias características de levantamiento. Siendo estos instantes los más críticos de cara al movimiento del robot (donde se generan las percusiones debido a las fuerzas de rozamiento), el hecho de que se sigan generando las funciones $\theta(t)$ buscadas demuestra la robustez del sistema de cara a su implementación real.

Capítulo 6

Conclusiones y trabajos futuros

A lo largo de este documento, se ha estudiado la aplicación de un sistema electromecánico que asuma las funciones del Central Pattern Generator de los insectos para conseguir la locomoción de un robot hexápedo. La metodología propuesta sustituye el esquema clásico sensor-controlador-actuador por un sistema dinámico no lineal que genera, de forma autónoma, las oscilaciones necesarias para generar el movimiento del robot, imitando los diferentes modos típicos de desplazamiento de los insectos en los que se inspira este Central Pattern Generator.

Partiendo de un circuito electrónico que generaba señales $\theta(t)$, en términos de voltaje, similares a las necesarias para generar los diferentes modos de avance de los insectos, se estudió la conversión de este sistema a un sistema electromecánico capaz de controlar, de forma efectiva, el movimiento real de un robot hexápedo. A pesar de las fuertes ligaduras geométricas impuestas por la implementación de este sistema para la locomoción autónoma de un robot, queda probado que se siguen generando las mismas funciones $\theta(t)$ que originan los modos de movimiento que inspiraron el desarrollo de esta metodología.

El objetivo de este trabajo era modelizar y validar el sistema propuesto y no realizar un diseño de la implementación real final. Por ello, no deben de tomarse los diferentes parámetros utilizados como los únicos válidos para el correcto funcionamiento del sistema. Los diferentes valores de los parámetros característicos y de diseño han sido estimados tratando de evitar incompatibilidades que impidan la correcta implementación real del sistema.

Habiéndose cumplido los objetivos propuestos en el desarrollo de este proyecto, los siguientes pasos deberían de estar enfocados a la implementación real del sistema. En [17], se propone un diseño mecánico para construir las extremidades del robot, controladas por el Central Pattern Generator propuesto mediante los ángulos θ_i . La información contenida en este Trabajo de Fin de Grado, junto con la del Proyecto de Fin de Carrera de [16], realizados en colaboración, permite seleccionar los parámetros finales de diseño de acuerdo con las características constructivas y las especificaciones del robot final.

Apéndices

Apéndice A

Código momentos mecánicos constantes

Código A.1: principal.m

```
1 % Programa principal
2 clc
3 clear all
4 close all
5 % Valores iniciales
6 [theta00, thetap0, Xcm0, Xcm_p0, L, mp, theta_0, M, Ip, mi, dt] =
    parametros(); % Parametros iniciales
7
8
9 %% MODO DE MOVIMIENTO METACHRONAL
10 theta0 = theta00(1,:);
11
12 t0 = 0;
13
14 % Inciamos las variables de representación
15 tplot = [];
16 Xcm_plot_1 = [];% Por cinemática
17 Xcm_p_plot_1=[];% Por cinemática
18 Xcm_pp_plot_1=[];% Por cinemática
19 thetапlot = [];
20 thetapp_plot = [];
21 thetap_plot = [];
22
23 Xcm_01 = Xcm0;
24 Xcm_p_01 = Xcm_p0;
25
26
27 for pata_j =[1 2, 3, 4, 5, 6, 1 2, 3, 4, 5, 6];
28     pnl = 6 - size(pata_j,2);
29     [theta, thetap, thetapp, t] = dinamica_completa(theta0,
        thetap0,L, mp, theta_0, M, Ip, mi, pata_j, t0, dt, pnl);
30
31 % Calculamos la cinemática del centro de masas a partir de
    la de las patas
```

```

32 [Xcm_1,Xcm_p_1, Xcm_pp_1] = centrodemasas(theta, thetap,
33   thetapp, L,t,Xcm_01,Xcm_p_01, pata_j);
34
35 % Vamos juntando los valores de los ángulos en variables
36 % independientes
37 % para la representación
38 thetplot = vertcat(thetplot, theta);
39 thetap_plot = vertcat(thetap_plot, thetap);
40 thetapp_plot = vertcat(thetapp_plot, thetapp);
41
42 Xcm_plot_1 = vertcat(Xcm_plot_1, Xcm_1);
43 Xcm_p_plot_1 = vertcat(Xcm_p_plot_1, Xcm_p_1);
44 Xcm_pp_plot_1 = vertcat(Xcm_pp_plot_1, Xcm_pp_1);
45
46 tplot = [tplot,t];
47
48 % Reiniciamos los valores inciales con los finales de la
49 % anterior
50 % integración y cambiamos la condición inicial de las patas
51 % que se
52 % apoyan.
53 theta0 = theta(end,:);
54 theta0(pata_j) = theta_0;
55 thetap0 = thetap(end,:);
56 Xcm0 = Xcm_1(end);
57 Xcm_p0 = Xcm_p_1(end);
58 if pata_j ==6
59   Xcm_01 = Xcm_1(end,4);
60   Xcm_p_01 = Xcm_p_1(end,4);
61   thetap0(pata_j)=thetap(end,4)*cos(theta0(pata_j))^2/cos(
62     theta(end,4))^2;
63
64 else
65   Xcm_01 = Xcm_1(end,6);
66   Xcm_p_01 = Xcm_p_1(end,6);
67   thetap0(pata_j)=thetap(end,6)*cos(theta0(pata_j))^2/cos(
68     theta(end,6))^2;
69 end
70
71 t0 = t(end);
72 end
73
74 % Representamos
75 m=1;
76 figure (m)
77 m=m+1;
78 h = plot (tplot, thetplot);

```

```

77 title (' Metachronal ','FontSize', 20);
78 xlabel ('Time (s)', 'FontSize', 20);
79 ylabel ('Angle \theta (rad)', 'FontSize', 20)
80 legend (h, '\theta_1', '\theta_2', '\theta_3', '\theta_4', '\
theta_5', '\theta_6')
81
82
83 figure(m)
84 m=m+1;
85 plot (tplot, Xcm_plot_1)
86 title ('Metachronal', 'FontSize', 20);
87 xlabel ('Time (s)', 'FontSize', 20);
88 ylabel ('Position of the Center of Mass X_{cm} (m)', 'FontSize',
20);
89 legend ('Leg 1', 'Leg 2', 'Leg 3', 'Leg 4', 'Leg 5', 'Leg 6', 'Location',
'Best');
90
91 figure(m)
92 m=m+1;
93 plot (tplot, Xcm_p_plot_1)
94 title ('Modo Metachronal', 'FontSize', 20);
95 xlabel ('Time (s)', 'FontSize', 20);
96 ylabel ('Velocity of the Center of Mass {V}_{cm} (m s^{-1})', 'FontSize',
20);
97 legend ('Leg 1', 'Leg 2', 'Leg 3', 'Leg 4', 'Leg 5', 'Leg 6', 'Location',
'Best');
98
99
100
101
102 %% MODO DE MOVIMIENTO CATERPILLAR
103
104 %
105 [theta00, thetap0, Xcm0, Xcm_p0,~,~,~,~,~,~,~] = parametros()
106 % Parametros iniciales
107 t0 = 0;
108 theta0 = theta00(2,:);
109 % Inciamos las variables de representación
110 tplot = [];
111 Xcm_plot_1 = [];% Por cinemática
112 Xcm_p_plot_1=[];% Por cinemática
113 Xcm_pp_plot_1=[];% Por cinemática
114 thetапlot = [];
115 thetapp_plot = [];
116 thetap_plot = [];
117
118 Xcm_01 = Xcm0;
119 Xcm_p_01 = Xcm_p0;
120
121 secuencia = {[1 4], [2 5],[3 6],[1 4], [2 5],[3 6],[1 4], [2

```

```

    5] ,[3 6}};

122 for i = 1:size(secuencia,2)
123     pata_j =cell2mat(secuencia(i));
124     pnl = 6 - size(pata_j,2);
125     [theta, thetap,thetapp, t] = dinamica_completa(theta0,
126         thetap0,L, mp, theta_0, M, Ip, mi, pata_j, t0, dt, pnl);

127 % Calculamos la cinemática del centro de masas a partir de
128 % la de las patas
129 [Xcm_1,Xcm_p_1, Xcm_pp_1] = centrodemasa(theta, thetap,
130     thetapp, L,t,Xcm_01,Xcm_p_01, pata_j);

131 % Vamos juntando los valores de los ángulos en variables
132 % independientes
133 % para la representación
134 thetplot = vertcat(thetplot, theta);
135 thetap_plot = vertcat(thetap_plot, thetap);
136 thetapp_plot = vertcat(thetapp_plot, thetapp);

137 Xcm_plot_1 = vertcat(Xcm_plot_1, Xcm_1);
138 Xcm_p_plot_1 = vertcat(Xcm_p_plot_1, Xcm_p_1);
139 Xcm_pp_plot_1 = vertcat(Xcm_pp_plot_1, Xcm_pp_1);

140 tplot = [tplot,t];

141

142

143 % Reiniciamos los valores inciales con los finales de la
144 % anterior
145 % integración y cambiamos la condición inicial de las patas
146 % que se
147 % apoyan.
148 theta0 = theta(end,:);
149 theta0(pata_j) = theta_0;
150 thetap0 = thetap(end,:);
151 Xcm0 = Xcm_1(end);
152 Xcm_p0 = Xcm_p_1(end);
153 for j=1:6
154     if j ~= pata_j
155         ref = j;
156         break
157     end
158 end

159 for j = 1:size(pata_j,2)
160     Xcm_01 = Xcm_1(end,ref);
161     Xcm_p_01 = Xcm_p_1(end,ref);
162     thetap0(pata_j(j))=thetap(end,ref)*cos(theta0(pata_j(j)))
163         )^2/cos(theta(end,ref))^2;
164 end

t0 = t(end);

```

```

165 end
166
167
168
169 % Representamos
170 m=4;
171 figure (m)
172 m=m+1;
173 h = plot (tplot, thetaplot);
174 title ('Caterpillar','FontSize', 20);
175 xlabel ('Time (s)', 'FontSize', 20);
176 ylabel ('Angle \theta (rad)', 'FontSize', 20)
177 legend (h, '\theta_1', '\theta_2', '\theta_3', '\theta_4', '\theta_5', '\theta_6')
178
179
180 figure(m)
181 m=m+1;
182 plot (tplot, Xcm_plot_1)
183 title ('Caterpillar', 'FontSize', 20);
184 xlabel ('Time (s)', 'FontSize', 20);
185 ylabel ('Position of the Center of Mass X_{cm} (m)', 'FontSize',
20);
186 legend ('Leg 1', 'Leg 2', 'Leg 3', 'Leg 4', 'Leg 5', 'Leg 6', 'Location',
'Best');
187
188 figure(m)
189 m=m+1;
190 plot (tplot, Xcm_p_plot_1)
191 title ('Caterpillar', 'FontSize', 20);
192 xlabel ('Time (s)', 'FontSize', 20);
193 ylabel ('Velocity of the Center of Mass {V}_{cm} (m s^{-1})', 'FontSize',
20);
194 legend ('Leg 1', 'Leg 2', 'Leg 3', 'Leg 4', 'Leg 5', 'Leg 6', 'Location',
'Best');
195
196
197
198 %% MODO DE MOVIMIENTO TRIPOD
199
200 %
201 [theta00, thetap0, Xcm0, Xcm_p0,~,~,~,~,~,~,~] = parametros\(\)
; % Parametros iniciales
202 t0 = 0;
203 theta0 = theta00(3,:);

204 %
205 % Inciamos las variables de representación
206 tplot = [];
207 Xcm_plot_1 = [];% Por cinemática
208 Xcm_p_plot_1=[];% Por cinemática
209 Xcm_pp_plot_1=[];% Por cinemática

```

```

210 thetaplot = [];
211 thetapp_plot = [];
212 thetap_plot = [];
213
214 Xcm_01 = Xcm0;
215 Xcm_p_01 = Xcm_p0;
216
217 secuencia = {[ 2 4 6], [ 1 3 5], [ 2 4 6], [ 1 3 5],[ 2 4 6], [
218   1 3 5],[ 2 4 6], [ 1 3 5],[ 2 4 6],[ 1 3 5]};
219 for i = 1:size(secuencia,2)
220     pata_j =cell2mat(secuencia(i));
221     pnl = 6 - size(pata_j ,2);
222     [theta, thetap,thetapp, t] = dinamica_completa(theta0,
223         thetap0,L, mp, theta_0, M, Ip, mi, pata_j, t0, dt, pnl);
224
225 % Calculamos la cinemática del centro de masas a partir de
226 % la de las patas
227 [Xcm_1,Xcm_p_1, Xcm_pp_1] = centrodematas(theta, thetap,
228         thetapp, L,t,Xcm_01,Xcm_p_01, pata_j);
229
230 % Vamos juntando los valores de los ángulos en variables
231 % independientes
232 % para la representación
233 thetaphot = vertcat(thetaplot, theta);
234 thetaphot = vertcat(thetaphot, thetap);
235 thetapp_plot = vertcat(thetapp_plot, thetapp);
236
237 Xcm_plot_1 = vertcat(Xcm_plot_1, Xcm_1);
238 Xcm_p_plot_1 = vertcat(Xcm_p_plot_1, Xcm_p_1);
239 Xcm_pp_plot_1 = vertcat(Xcm_pp_plot_1, Xcm_pp_1);
240
241 tplot = [tplot,t];
242
243
244 % Reiniciamos los valores inciales con los finales de la
245 % anterior
246 % integración y cambiamos la condición inicial de las patas
247 % que se
248 % apoyan.
249 theta0 = theta(end,:);
250 theta0(pata_j) = theta_0;
251 thetap0 = thetap(end,:);
252 Xcm0 = Xcm_1(end);
253 Xcm_p0 = Xcm_p_1(end);
254 for j=1:6
255     if j ~= pata_j
256         ref = j;
257         break
258     end
259 end

```

```

254     for j = 1:size(pata_j,2)
255         Xcm_01 = Xcm_1(end,ref);
256         Xcm_p_01 = Xcm_p_1(end,ref);
257         thetap0(pata_j(j))=thetap(end,ref)*cos(theta0(pata_j(j)))
258             )^2/cos(theta(end,ref))^2;
259     end
260
261     t0 = t(end);
262
263
264
265 % Representamos
266 m=7;
267 figure(m)
268 m=m+1;
269 h = plot (tplot, thetап);
270 title (' Tripod','FontSize', 20);
271 xlabel ('Time (s)', 'FontSize', 20);
272 ylabel ('Angle \theta (rad)', 'FontSize', 20)
273 legend (h, '\theta_1', '\theta_2', '\theta_3', '\theta_4', '\theta_5', '\theta_6')
274
275
276 figure(m)
277 m=m+1;
278 plot (tplot, Xcm_plot_1)
279 title ('Tripod', 'FontSize', 20);
280 xlabel ('Time (s)', 'FontSize', 20);
281 ylabel ('Position of the Center of Mass X_{cm} (m)', 'FontSize',
282 20);
282 legend ('Leg 1', 'Leg 2', 'Leg 3', 'Leg 4', 'Leg 5', 'Leg 6', 'Location',
283 'Best');
284
284 figure(m)
285 m=m+1;
286 plot (tplot, Xcm_p_plot_1)
287 title ('Tripod', 'FontSize', 20);
288 xlabel ('Time (s)', 'FontSize', 20);
289 ylabel ('Velocity of the Center of Mass {V}_{cm} (m s^{-1})', 'FontSize',
290 20);
290 legend ('Leg 1', 'Leg 2', 'Leg 3', 'Leg 4', 'Leg 5', 'Leg 6', 'Location',
291 'Best');

```

Código A.2: parametros.m

```

1 % Parámetros y valores iniciales del problema
2 function [theta0, thetap0, Xcm0, Xcm_p0, L, mp, theta_0, M, Ip,
3 mi,dt] = parametros()
4 %

```

```

4  % - Variables de salida
5  %     theta0, thetap0 :: Valores iniciales de ángulos y
6  %                         velocidades angulares
7  %     Xcm0, Xcm_p0      :: Valores iniciales de posicion y
8  %                         velocidad del CM
9  %     L                  :: Distancia entre el cuerpo y el punto
10 %        de apoyo de las patas
11 %     mp                 :: Masa de cada una de las patas
12 %     theta_0             :: Semiángulo de amplitud maxima
13 %     M                  :: Masa del conjunto cuerpo-patas
14 %     Ip                 :: Momento de inercia de las patas
15 %        levantadas
16 %     mi                 :: Momentos mecanicos sobre las patas
17 %        apoyadas
18 %     dt                 :: Paso de integracion temporal
19 %

```

```

20
21
22 Xcm0 = 0;
23 Xcm_p0 = 0;
24 L = 0.3;
25 mp = 0.20;
26 theta_0 = 70*pi/(2*180);
27 % Condiciones Metachronal
28 theta0(1,:) = [-theta_0, -theta_0*2/3, -theta_0/3, theta_0/3,
    theta_0*2/3, theta_0];
29 % Condiciones Caterpillar
30 theta0(2,:) = [-theta_0, 0, +theta_0, -theta_0, 0, +theta_0];
31 % Condiciones Tripod
32 theta0(3,:) = [theta_0, -theta_0, theta_0, -theta_0, +theta_0, -
    theta_0];
33 thetap0 = [ 0, 0, 0, 0, 0, 0];
34 M = 3;
35 mi = 0.25*[-1, -1, -1, -1, -1, -1];
36 dt = 0.005;
37 Ip = mp*L ^2/(3*cos(theta_0))^2;
38
39 return

```

Código A.3: dinamica_completa.m

```

1  % Esta función recibe los valores de los 6 ángulos, las 6
2  % velocidades ángulares y que pata está en movimiento a tiempo
3  % t0 y realiza
4  % la integracion hasta un tiempo tf, donde una pata alcanza el
5  % angulo de
6  % amplitud maxima
7
8  function [theta, thetap, thetapp, t] = dinamica_completa(theta0,
    thetap0, L, mp, theta_0, M, Ip, mit, pata_j, t0, dt, pn1);

```

```

7
8 %
=====

9 % - Variables de entrada
10 %      theta0, thetaph0, Xcm0, Xcm_p0 :: ángulos, velocidades
11 %                                         angulares, posición y
12 %                                         velocidad iniciales.
13 %      L, mp, theta_0, M, Ip           :: parámetros
14 %                                         característicos
15 %      mit                          :: momentos mecánicos
16 %      pata_j                      :: patas que se encuentran
17 %                                         levantadas en el instante
18 %                                         de tiempo concreto
19 %      t0, dt                      :: tiempo inicial, paso
20 %                                         temporal
21 %      pnl                         :: numero de patas no
22 %                                         levantadas
23 %

24 % - Variables de salida
25 %      theta                        :: matriz cuyas filas
26 %                                         almacenan los 6 ángulos
27 %                                         de las patas para cierto
28 %                                         tiempo
29 %      thetaph                     :: matriz cuyas filas
30 %                                         almacenan la velocidad
31 %                                         angular de las 6 patas
32 %                                         para cierto tiempo
33 %      thetapp                     :: matriz cuyas filas
34 %                                         almacenan la aceleración
35 %                                         angular de las 6 patas
36 %                                         para cierto tiempo
37 %      t                           :: tiempo
38 %

39 % - Utiliza las funciones
40 %      RK4_din.m
41 %      Dinamica_levantadas
42 %

=====

43
44 % Construimos el vector del estado inicial para la integración,
45 % eliminando
46 % la coordenada de la pata levantada (esta se integrará mediante
47 % otra
48 % subrutina)
49
50 k= 1;
for i=1:6
    % Solo rellenamos el vector X0 y aumentamos el valor de k,
    % si la pata

```

```

51 % no coincide con la que está levantada
52 if i ~= pata_j;
53     X0(1,k) = theta0(i);
54     X0(1,k+pn1) = thetap0(i);
55     mi(k) = mit(i);
56     k= k+1;
57 end
58 end
59 % Realizamos la integración con la subrutina preparada para tal
60 % fin
61 [x,t]=RK4_din(X0,@dinamica_apoyadas, L, mp, theta_0, M, mi, dt,
62 pn1);
63 % Sumamos al tiempo el tiempo inicial de la integración
64 t = t + t0;
65
66
67
68
69 % Reintroducimos los valores de los ángulos con la numeración
70 % original del
71 % 1 al 6
72 k = 1;
73 theta = zeros(size(t,2),6);
74 thetap = zeros(size(t,2),6);
75 thetapp = zeros(size(t,2),6);
76
77 for i=1:6
78     if i ~= pata_j
79         theta(:,i) = x(:,k);
80         thetap(:,i)=x(:,k + pn1);
81         for l = 1:size(t,2)
82             [U_p] = dinamica_apoyadas(x(l,:), L, mp, theta_0, M,
83                                         mi, pn1);
84             thetapp(l,i) = U_p(k+pn1);
85         end
86         k= k +1;
87     else
88         % Aquí entrarian las subrutinas para calcular la pata
89         % levantada, se
90         % dejan sus valores en 0 por ahora
91         [theta(:,i),thetap(:,i), thetapp(:,i)]=
92             dinamica_levantadas(t-t0, Ip,theta_0, theta0(1,i));
93     end
94 end
95 return

```

Código A.4: RK4_din.m

```

1 % Función Runge-Kutta de 4 pasos, particularizada para el
2 % movimiento del robot
3 function [x,t] = RK4_din(x0,funcion, L, mp, theta_0, M, mi, dt,
4 % 
5 % - Variables de entrada
6 %     x0          :: Vector fila con la condición inicial
7 %     funcion      :: funcion que almacena la ecuacion
8 %                   a resolver
9 %     L, mp, theta_0, M :: parametros caracteristicos
10 %    mi           :: Vector que almacena los momentos
11 %                   mecanicos
12 %    dt           :: paso de integracion
13 %    pnl          :: numero de patas no levantadas
14 %
15 % - Variables de salida
16 %     x           :: Matriz que almacena, por filas,
17 %                   los angulos y velocidades angulares
18 %                   de las patas apoyadas
19 %     t           :: Vector que almacena los tiempos
20 %                   de integracion
21 %
22 %

23
24 control = true;
25 x(1,:) = x0;
26 t(1) = 0;
27 i = 0;
28 while control
29     i = i+1;
30     k1 = funcion(x(i,:), L, mp, theta_0, M, mi,pnl);
31     k2 = funcion(x(i,:)+ dt*k1/2, L, mp, theta_0, M, mi,pnl);
32     k3 = funcion(x(i,:)+ dt*k2/2, L, mp, theta_0, M, mi,pnl);
33     k4 = funcion(x(i,:)+ dt*k3/2, L, mp, theta_0, M, mi,pnl);
34
35     x(i+1,:) = x(i,:)+(dt/6)*(k1 +2*k2 + 2*k3 + k4);
36     t(i+1) = t(i) + dt;
37     % comprobamos si alguno de los ángulo se pasa de theta0, en
38     % cuyo caso
39     % paramos la integracion
40     for j=1:pnl
41         if abs(x(i+1,j)) >= (theta_0)
42             control = false;
43         end
44     end

```

```

44
45    % Introducimos este criterio de parada de la integración
        % para que, si
46    % las patas no se mueve, no entre en un bucle infinito al no
        % conseguir
47    % alcanzar el valor theta0
48    if i > 10000
49        control = false;
50    end
51 end
52 return

```

Código A.5: dinamica_apoyadas.m

```

1  % Función que proporciona la ecuación que modela la dinámica de
    % las patas apoyadas en la forma
2  % {U'} = [A(U)]{U} + {F} para la integración mediante Runge-
    % Kutta.
3
4 function [U_p] = dinamica_apoyadas(U, L, mp, theta_0, M, mi, pnl
    )
5 %



---


6 % - Variables de entrada
7 %     U                      :: Vector que almacena, por este orden,
8 %                               los ángulos y, a continuación, las
9 %                               derivadas de las patas apoyadas
10 %     L, mp, theta_0, M :: Parametros caracteristicos del
    % problema
11 %
12 %     mi                    :: Vector que almacena los 5 momentos
    % de las patas
13 %
14 %     pnl                  :: numero de patas no levantadas
15 % - Variables de salida
16 %     U                      :: Vector que almacena, por este orden,
    % velocidad angular y la aceleracion
    % angular de las patas apoyadas
17 %
18 %
19 %
20 %



---


21
22 % Parametros internos de la funcion
23 E = 0;
24 tol = 1e-5; % tolerancia para que la matriz sea o no singular
25
26 % por comodidad de las expresiones, extraemos los ángulos del
    % vector U
27 theta(:,1) = U(1:pnl);
28 control = true;
29 while control
30 %Construimos los coeficientes de las matrices

```

```

31     for i= 1:pnl
32         A(i) = L/(cos(theta(i)))^2;
33         B(i) = 2*L*tan(theta(i))/(cos(theta(i))^2);
34         C(i) = (mp*L/(3*M*(cos(theta_0)^2)))*(-2*cos(theta_0)/
35             cos(theta(i)) + 4 *(cos(theta_0)/cos(theta(i)))^2)*tan(
36             theta(i));
37         I = (mp*L^2/3*(cos(theta_0)^2))*(1-2*cos(theta_0)/cos(
38             theta(i)) + 2*(cos(theta_0)/cos(theta(i)))^2);
39         D(i) = I/(M*L);
40         E = E + mi(i)/(M*L);
41     end
42     %Construimos las matrices
43     for i=1:pnl
44         for j = 1:pnl
45             if j == i
46                 AD (i,j) = D(i) + A(i);
47                 CB (i,j) = C(i) + B(i);
48             else
49                 AD(i,j) = D(j);
50                 CB(i,j) = C(j);
51             end
52         end
53     end
54
55     % Comprobamos si la matriz AD se puede invertir
56     if abs(det(AD)) <= tol
57         % si la matriz es singular, le sumamos una cantidad
58         % pequeña a los
59         % ángulos, de forma que deje de ser singular, pero
60         % apenas afecte a
61         % la resolución del sistema
62         theta = theta + theta/100000;
63     else
64         control = false;
65     end
66 end
67 % rellenamos un vector con el valor de las velocidades angulares
68 % al cuadrado
69 for i=1:pnl
70     theta_p2(i,1) = U(pnl+i)^2;
71 end
72 % Calculamos el valor de las aceleraciones angulares
73 theta_pp = -inv(AD)*CB*theta_p2 + inv(AD)*E*ones(pnl ,1);
74
75 % Rellenamos el valor del vector U'
76 U_p = vertcat(U(pnl+1:2*pnl)', theta_pp);
77 % Devolvemos el valor de U_p como vector fila, para usarlo en el
78 % Runge-Kutta
79 U_p = U_p';
80 return

```

Código A.6: dinamica_levantadas.m

```

1 % Subrutina que calcula la dinámica de las patas en fase de
2 % vuelo,
3 %
4 function [theta,thetap, thetapp] = dinamica_levantadas(t, Ip,
5 % theta_0, theta1)
6 %
7 %
8 %
9 %
10 % - Variables de entrada
11 % t :: Vector que almacena el intervalo
12 % de integración (tiempo comienza
13 % en 0)
14 % Ip :: Momento de inercia de las patas
15 % levantadas
16 % theta_0 :: Semiamplitud maxima de las patas
17 % thetap0 :: Velocidad angular inicial
18 %
19 %

20 %
21 % Allocatamos memoria
22 thetapp= zeros (size(t,2),1);
23 thetap= zeros (size(t,2),1);
24 theta= zeros (size(t,2),1);
25 %
26 % Calculamos el momento (constante) que tiene que aplicarse para
27 % que la
28 % pata pase de -theta_0 a theta_0 en el intervalo de tiempo
29 % considerado
30 %
31 m = (2*Ip/(t(end)^2))*(theta_0 - theta1);
32 %
33 for i = 1:size(t,2)
34 % Integraremos el movimiento de la pata de manera analítica
35 thetapp(i,1) = m/Ip;
36 thetap(i,1) = (m/Ip)*t(i);
37 theta(i,1) = (m/(2*Ip))*t(i)^2 + theta1;
38 end
39 return

```

Código A.7: centrodemapas.m

```

1 % Calculo de la cinematica del centro de masas a partir de la
2 % cinemática de
3 % cada una de las patas apoyadas
4
5 function [Xcm,Xcm_p, Xcm_pp] = centrodemapas(theta, thetap,
6 %
7 % - Variables de entrada
8 % theta :: Matriz que almacena, por filas, los
9 % valores de los angulos de todas las
10 % patas para cierto instante
11 % thetap :: Matriz que almacena, por filas, los
12 % valores de velocidad angular de todas
13 % las patas para cierto instante
14 % thetapp :: Matriz que almacena, por filas, los
15 % valores de aceleracion de todas las
16 % patas para cierto instante
17 % L :: parametro caracteristico
18 % X0, Xp0 :: Posicion y velocidad inicial del CM
19 %
20 % - Variables de salida
21 % Xcm, Xcm_p, Xcm_pp :: Matrices que almacenan, por filas,
22 % los valores de posicion, velocidad
23 % y aceleracion del CM para cierto
24 % instante
25
26 Xcm =zeros(size(t),6);
27 Xcm_p =zeros(size(t),6);
28 Xcm_pp =zeros(size(t),6);
29 Xcm(1,:) = X0;
30 Xcm_p(1,:) = Xp0;
31
32 for i =1:size(t,2)
33     for j = 1:6
34         if i ~= size(t,2)
35             Xcm(i+1,j)= -L*(tan(theta(i+1,j))-tan(theta(1,j)))+
36             X0; % Posición CM
37             end
38             Xcm_p(i,j) = -(L*thetap(i,j)/(cos(theta(i,j))^2)); %+
39             vel_pata(j); % Velocidad CM
40             Xcm_pp(i,j) =-( L*thetapp(i,j)/(cos(theta(i,j))^2) + (2*
41             L*tan(theta(i,j))*(thetap(i,j)^2))/(cos(theta(i,j))^2)
42             ); % Aceleración CM
43         end
44     end

```

```
41
42 for i=1:6
43     if i ~= pata_j
44         ref = i;
45         break
46     end
47 end
48
49 for j = 1:size(pata_j,2)
50     Xcm(:,pata_j(j)) = Xcm(:,ref);
51     Xcm_p(:,pata_j(j)) = Xcm_p(:,ref);
52     Xcm_pp(:,pata_j(j)) = Xcm_pp(:,ref);
53 end
```

Apéndice B

Código momentos proporcionados por Toda-Rayleigh

Código B.1: main.m

```
1 % Programa principal
2 %=====
3 % — Utiliza las funciones:
4 %     setup.m
5 %     RK4.m
6 %     cinematica_CM
7 %=====
8
9
10 clc
11 clear all
12 close all
13
14 tic
15 % Incorporamos los parametros inciales del programa
16 [n,omega, gamma, mu, R, theta0, thetap0,theta_pp0, Xcm0, Xcm_p0
17 , L, mp, theta_0, M,tin,dt,tf, beta,RB,V0,K,C,Cr,ac0]=
18     setup();
19
20 % Iniciamos variables
21 global estado
22 m = 1;
23 t0 = 0;
24 t = t0;
25 iter = floor(tin/dt);
26 V_sh0 = V0;
27
28 % Iniciamos las gráficas
29 % CINEMATICA DE LAS PATAS
30 fig_angulos = m;
31 m = m+1;
32 figure (fig_angulos);
33 clf(fig_angulos);
```

```

33 xlabel ('Time (s)', 'FontSize', 20);
34 ylabel ('Angle \theta', 'FontSize', 20);
35 hold on;
36
37 fig_vel = m;
38 m = m+1;
39 figure (fig_vel)
40 set(fig_vel, 'Visible', 'Off')
41 clf(fig_vel);
42 xlabel ('Time (s)', 'FontSize', 20);
43 ylabel ('Angular Velocity $\dot{\theta}$', 'Interpreter', 'Latex
    ', 'FontSize', 20)
44 hold on;
45
46 fig_acel = m;
47 m = m+1;
48 figure(fig_acel)
49 set(fig_acel, 'Visible', 'Off')
50 clf(fig_acel);
51 xlabel ('Time (s)', 'FontSize', 20);
52 ylabel ('Angular Acceleration $\ddot{\theta}$', 'Interpreter', 'Latex
    ', 'FontSize', 20)
53 hold on;
54
55
56 % FUNCTION TODA
57 fig_toda = m;
58 m = m+1;
59 figure(fig_toda)
60 set(fig_toda, 'Visible', 'Off')
61 clf(fig_toda);
62 xlabel ('Time (s)', 'FontSize', 20);
63 ylabel ('$\Phi_{dc}$', 'Interpreter', 'Latex', 'FontSize', 20)
64 title ('Moment of Toda-Rayleigh', 'FontSize', 20)
65 hold on;
66
67 % FUERZAS DE ROZAMIENTO
68 fig_fuer = m;
69 m=m+1;
70 figure(fig_fuer)
71 set(fig_fuer, 'Visible', 'Off')
72 clf(fig_fuer);
73 xlabel ('Time (s)', 'FontSize', 20);
74 ylabel ('Friction Forces', 'FontSize', 20)
75 hold on;
76
77
78 % ESTADO DE LAS PATAS: 0 = APOYADA
79 fig_estado = m;
80 m = m+1;
81 figure(fig_estado)

```

```

82 set(fig_estado,'Visible','Off')
83 clf(fig_estado);
84 xlabel ('Time (s)', 'FontSize',20);
85 title ('Phase of the legs: Swing or Steady', 'FontSize',20);
86 hold on;
87
88 %
89 % % CINEMATICA CM
90 fig_pos_cm = m;
91 m=m+1;
92 figure(fig_pos_cm)
93 set(fig_pos_cm,'Visible','Off')
94 clf(fig_pos_cm);
95 xlabel ('Time (s)', 'FontSize',20);
96 ylabel ('Position CM', 'FontSize',20)
97 hold on;
98
99 fig_vel_cm = m;
100 m=m+1;
101 figure(fig_vel_cm)
102 set(fig_vel_cm,'Visible','Off')
103 clf(fig_vel_cm);
104 xlabel ('Time (s)', 'FontSize',20);
105 ylabel ('Velocity CM', 'FontSize',20)
106 hold on;
107
108
109
110 if Cr ==1
111     fig_V_sh = m;
112     m = m+1;
113     figure(fig_V_sh )
114     set(fig_V_sh , 'Visible','Off')
115     xlabel ('Time (s)', 'FontSize',20);
116     ylabel ('V_{sh}', 'FontSize',20)
117     hold on;
118 end
119
120
121
122 while t(end) < tf-dt
123 %% ===== CALCULOS =====
124 if (t0 + iter*dt) > tf +dt
125     iter = mod(tf,tin)/dt;
126 end
127 % Integracion del movimiento de las patas mediante la
128 % subrutina RK4.m
129 [x,t,theta_pp,V_sh, ac,salida] = RK4([theta0, thetap0], L,
130 mp, theta_0, M, dt, omega, gamma, mu, R,n, t0, beta,RB,V0,
131 K,C,Cr,theta_pp0,iter,ac0,V_sh0);
132 % Calculamos la cinematica del CM

```

```

130 [Xcm,Xcm_p, Xcm_pp, XCM, XCM_p,F] = cinematica_CM(x(:,1:n),x
131   (:,n+1:2*n), theta_pp, L,t,n, Xcm0, Xcm_p0,omega, gamma,mu
132   ,R,V_sh,beta,RB,theta_0,ac,mp);
133
134 % Separamos en angulos y velocidades angulares
135 theta = x(:,1:n);
136 theta_p = x(:,n+1:2*n);
137
138 %% %% ===== REPRESENTACION DE RESULTADOS =====
139
140 % CINEMATICA DE LAS PATAS
141 figure (fig_angulos);
142 plot (t, theta);
143
144 figure (fig_vel)
145 set(fig_vel,'Visible','Off')
146 plot (t, theta_p);
147
148 figure(fig_acel)
149 set(fig_acel,'Visible','Off')
150 plot (t, theta_pp);
151
152
153 % FUNCIÓN TODA
154 figure(fig_toda)
155 set(fig_toda,'Visible','Off')
156 plot (t, ac)
157
158 % Fuerzas Rozamiento
159 figure(fig_fuer)
160 set(fig_fuer,'Visible','On')
161 plot (t, F)
162
163 % ESTADO DE LAS PATAS: 0 = APOYADA
164 figure(fig_estado)
165 set(fig_estado,'Visible','Off')
166 paso = (t(end) - t0)/(size(estado,1) -1);
167 plot ([t0:paso:t(end)], [1*estado(:,1),2*estado(:,2),3*estado
168   (:,3),4*estado(:,4),5*estado(:,5),6*estado(:,6)])
169
170 % % CINEMATICA CM
171 figure(fig_pos_cm)
172 set(fig_pos_cm,'Visible','Off')
173 plot (t, XCM)
174
175 figure(fig_vel_cm)
176 set(fig_vel_cm,'Visible','Off')
177 if t0 == 0; % No ploteamos los primeros valores de velocidad

```

```

    , que están, en general, mal calculados
178     plot (t(3:end), XCM_p(3:end))
179 else
180     plot (t, XCM_p)
181 end

182
183 % VOLTAJE DE CONTROL
184 if Cr ==1
185     figure(fig_V_sh )
186     set(fig_V_sh , 'Visible','Off')
187     plot (t, V_sh)
188     V_sh0 = V_sh(end);
189 end
190 % Actualizamos condiciones iniciales
191 theta0 = theta(end,:);
192 thetap0 = theta_p(end,:);
193 theta_pp0 = theta_pp(end,:);
194 t0 = t(end);
195 Xcm0 = XCM(end);
196 Xcm_p0 = XCM_p(end);
197 ac0 = ac(end,:);
198 if salida
199     break
200 end
201 clear -global estado;
202 global estado;
203 end

204
205 % ESTADO DE LAS PATAS: 0 = APOYADA
206 figure(fig_estado)
207 h = legend ('leg 1', 'leg 2', 'leg 3', 'leg 4', 'leg 5', 'leg 6')
208 ;
209 set(h, 'FontSize',15)

210 % CINEMATICA DE LAS PATAS
211 figure (fig_angulos);
212 h = legend ( '\theta_1', '\theta_2', '\theta_3', '\theta_4', '\theta_5', '\theta_6');
213 set(h, 'FontSize',15)

214
215 figure (fig_vel)
216 h = legend ( '\theta_1', '\theta_2', '\theta_3', '\theta_4', '\theta_5', '\theta_6');
217 set(h, 'FontSize',15)

218
219 figure(fig_acel)
220 h = legend ( '\theta_1', '\theta_2', '\theta_3', '\theta_4', '\theta_5', '\theta_6');
221 set(h, 'FontSize',15)

222
223 % FUNCIÓN TODA

```

```

224 figure(fig_toda)
225 h =legend ('leg 1', 'leg 2', 'leg 3', 'leg 4', 'leg 5', 'leg 6')
226 ;
227 set(h, 'FontSize',15)
228
229 % FUERZAS DE ROZAMIENTO
230 figure(fig_fuer)
231 h =legend ('leg 1', 'leg 2', 'leg 3', 'leg 4', 'leg 5', 'leg 6')
232 ;
233 set(h, 'FontSize',15)
234
235 if Cr ==1
236     figure(fig_V_sh )
237 end
238
239
240
241 t(end)

```

Código B.2: setup.m

```

1 % Incorpora al programa los parametros y valores iniciales del
   problema
2 % considerado
3 function [n,omega, gamma, mu, R, theta0, thetap0,theta_pp0 ,
   Xcm0, Xcm_p0, L, mp, theta_0, M,tin,dt,tf, beta, RB,V0,K,C,Cr ,
   ac0]= setup()
4 %=====
5 % - Variables de salida
6 %      n       :: Numero de patas.
7 %      omega   :: Parametro omega de la funcion de Toda.
8 %      gamma   :: Parametro gamma de la funcion de Toda.
9 %      mu      :: Parametro mu de la funcion de Toda.
10 %      R       :: Parametro R de la funcion de Toda.
11 %      theta_0 :: Condicion inicial para angulos,
   dimension (1,n).
12 %      thetap0 :: Condicion inicial para Velocidades angulares,
   dimension (1,n).
13 %      Xcm0    :: Posición inicial del Centro de Masas
14 %      Xcm_p0  :: Velocidad inicial del Centro de Masas.
15 %      L       :: distancia desde el cuerpo al punto de apoyo
   de las patas
16 %      mp      :: masa de las patas
17 %      theta_0 :: maxima amplitud de las patas
18 %      M       :: masa del conjunto cuerpo + patas
19 %      tin     :: Duración del intervalo de integración
20 %      dt      :: Paso de tiempo por iteracion.
21 %      tf      :: Tiempo final de integracion
22 %      beta    :: Parametro beta de la funcion Toda
23 %      RB      :: Resistencia del control

```

```

27 %      V0      :: Voltaje de control
28 %      K       :: Constante de amplificación del controlador
29 %      C       :: Capacidad del condensador de control
30 %      Cr      :: Variable de activación del controlador
31 %      ac0     :: Valor inicial de los momentos
32 %                      (se inicia a ceros)
33 %=====
34 global I1
35
36 %Parametros fijos del problema
37 n = 6;
38 Imax = 45e-6;    %[A]
39 Vtilde = 1.447;  %[V]
40 Rdc = 681;       %[Ohm]
41
42
43 % Parametros masicos y geometricos
44 L = 0.3;
45 mp = 0.20;
46 theta_0 = 70*pi/(2*180);
47 Mcuerpo = 1.8;
48 M = Mcuerpo +n*mp;
49 I1= mp*L^2 /(3*cos(theta_0))^2; % momento de inercia de la pata
estirada
50
51 global estado;
52 ac0 = zeros(1,6);
53 estado = [];
54
55 Xcm0 = 0;
56 Xcm_p0 = 0;
57 theta_pp0 = [0 0 0 0 0 0];
58
59
60 % Parametros integracion
61 tin = 5; % Intervalo de integración
62 dt = 0.001; % Paso de integración
63 tf = 60; % Tiempo final de integración;
64
65
66 % Parametros de Toda-Rayleigh
67 R = 500;           %[Ohm]
68 omegav = sqrt(I1)*10;      %[1/t]
69 beta = 0.7;        %[1/t]
70
71
72 %Parametros de Control
73 RB = R;           %[Ohm]
74 K = 2;
75 C = 15/R;
76 Cr = 1;

```

```

77
78
79 %Parametros de la funcion Toda
80 omega = sqrt(omegav*omegav*Rdc); % Ya incluye la IL
81 gamma = (3*sqrt(3)*Imax)/(2*beta*beta*beta*Vtilde*Vtilde*Vtilde)
82 ;
83 mu = Vtilde*Vtilde*beta*beta;
84
85 %% Modo Metachronal
86 % theta0 = [-0.3954 -0.2313 -0.0639 0.0840 0.2751
87 % 0.2297];
88 % thetap0 = [ -0.1141 -0.1269 -0.1334 -0.1330 -0.1241
89 % 0.7164];
90 %
91 % ref = 4;
92 % for i = 1:6
93 % if i ~= 6
94 % thetap0 (1,i) = cos(theta0(i))^2*thetap0(ref)/cos(theta0(
95 % ref))^2;
96 % end
97 % end
98 % V0 =-1;
99
100 %% Modo Caterpillar
101 theta0 = [ 0.1561 -0.2774 0.1256 0.1561 -0.2774
102 0.1256];
103 thetap0 = [ 0.8301 -0.3502 -0.3726 0.8301 -0.3502
104 -0.3726];
105 ref = 3;
106 for i = 2:n
107 if i ~= 4 && i ~= 1
108 thetap0 (1,i) = cos(theta0(i))^2*thetap0(ref)/cos(theta0(
109 ref))^2;
110 end
111 end
112 V0 =-0.5;
113
114 %% Modo Tripod
115 % theta0 = [-0.1657 0.1390 -0.1657 0.1390 -0.1657
116 % 0.1390];
117 % thetap0 = [ -0.7282 0.8531 -0.7282 0.8531 -0.7282
118 % 0.8531];
119 %
120 % V0 = 0.0331;

```

Código B.3: RK4.m

```

1 % Función Runge-Kutta de 4 pasos, particularizada para el
2 % movimiento del robot
3 function [x,t, theta_pp, V_sh, ac, salida] = RK4(x0, L, mp,

```

```

    theta_0, M, dt, omega, gamma, mu, R,n, t0, beta, RB,V0,K,C,Cr,
    theta_pp0,m,ac0,V_sh0)
4 %=====
5 % - Variables de entrada:
6 %      x0                               :: vector filas con
7 %                                         el estado (
8 %
9 %      angulos,                           velocidades
10 %                                         angulares)
11 %
12 %
13 %      inicial                          :: parametros
14 %                                         fisicos del
15 %                                         problema
16 %      L, mp, theta_0,M                :: paso de
17 %                                         integracion
18 %      dt                             :: parametros
19 %                                         de la funcion
20 %                                         Toda-Rayleigh
21 %      omega, gamma, mu, R, V_sh, beta, V0
22 %                                         :: numero de patas
23 %                                         :: tiempo inicial de
24 %                                         integracion
25 %      RB, k C, Cr                   :: parametro del
26 %                                         control de la
27 %                                         funcion de
28 %                                         Toda-Rayleigh
29 %      theta_pp0                     :: valor inicial de
30 %                                         la aceleracion
31 %      m                            :: numero de pasos
32 %                                         de integracion
33 %      ac0                          :: valor inicial
34 %                                         del momento
35 %                                         introducido por
36 %                                         los motores
37 %      V_sh0                        :: Valor inicial del
38 %                                         voltaje de
39 %                                         control V_sh
40 %
41 %
42 %
43 %      x                            :: matriz cuyas filas
44 %                                         son los angulos y
45 %                                         velocidades angulares
46 %                                         a tiempo t
47 %
48 %
49 %      t                            :: vector columna que
50 %                                         almacena el tiempo
51 %
52 %      theta_pp                      :: matriz cuyas filas
53 %                                         son la aceleracion
54 %                                         angular de las patas
55 %                                         a tiempo t
56 %
57 %      V_sh                         :: vector que contiene
58 %                                         el voltaje externo de

```

```

51 %
52 %
53 %      ac
54 %
55 %
56 %
57 %      salida
58 %
59 %
60 %
61 %
62 %
63 % - Utiliza las funciones:
64 %      TODAg.m
65 %      dinamica_robot.m
66 %      RKcontrol.m
67 %
68 %=====
69
70 clear -global x;
71 global x; % definimos x como global para poder cambiar la
             velocidad de las patas cuando se apoyan
72 global Il;
73
74 % Creamos las variables de salida
75 Vc = zeros(m+1,1);
76 x = zeros(m+1,12);
77 theta_pp = zeros(m+1,6);
78 t = zeros(m+1,1);
79 theta_pp(1,:) = theta_pp0;
80 t(1,1) = t0;
81 x(1,:)= x0;
82 V_sh(1) = V_sh0;
83 Vc(1) = (V0 - V_sh(1))/K;
84 salida = false;
85 ac = ac0;
86 for i = 1:1:m
87     if i ~= 1
88         ant = i-1;
89     else
90         ant = 1;
91     end
92
93     if Cr == 1
94         V_sh(i+1) = V0 - Cr*K*Vc(i);
95         Vc(i+1) = RKcontrol(x(i,1:n),Vc(i),RB,C,dt);
96     else
97         V_sh(i+1) = V0;
98     end
99
100 T = TODAg (x(i,1:n), x(i,n+1:2*n), omega, gamma,mu,R,n,V_sh(

```

la función de Toda-Rayleigh
:: matriz cuyas filas son el momento introducido por los motores
:: variable lógica que indica si se ha superado el ángulo máximo y se debe parar la integración

```

    i+1),beta,RB,Vc(i),Cr);
101 ac = [ac;Il*T];
102 k1 = dinamica_robot(x(i,:), L, mp, theta_0, M, omega, gamma,
103   mu, R,n, V_sh(i+1), beta,RB,Vc(i),Cr,i,theta_pp(i,:),x(
104     ant,:),theta_pp(ant,:),V_sh(i));
105 k2 = dinamica_robot(x(i,:)+dt*k1/2, L, mp, theta_0, M,
106   omega, gamma, mu, R,n, V_sh(i+1), beta,RB,Vc(i),Cr,i,
107   theta_pp(i,:),x(ant,:),theta_pp(ant,:),V_sh(i));
108 k3 = dinamica_robot(x(i,:)+dt*k2/2, L, mp, theta_0, M,
109   omega, gamma, mu, R,n, V_sh(i+1), beta,RB,Vc(i),Cr,i,
110   theta_pp(i,:),x(ant,:),theta_pp(ant,:),V_sh(i));
111 k4 = dinamica_robot(x(i,:)+dt*k3/2, L, mp, theta_0, M,
112   omega, gamma, mu, R,n, V_sh(i+1), beta,RB,Vc(i),Cr,i,
113   theta_pp(i,:),x(ant,:),theta_pp(ant,:),V_sh(i));
114
115 x(i+1,:) = x(i,:)+(dt/6)*(k1 +2*k2 + 2*k3 + k4);
116 t(i+1,1) = t(i,1) + dt;
117 theta_pp(i+1,:)= k1(n+1:end); % Aceleraciones de las patas
118
119 % comprobamos si alguno de los ángulo se pasa de theta0, en
120 % cuyo caso
121 % paramos la integracion
122 for j=1:n
123   if abs(x(i+1,j)) >= (theta_0);
124     salida = true;
125     fprintf('Se para la integracion porque la pata %d ha
126       alcanzado la amplitud maxima: %g \n', j, rad2deg
127       (x(i+1,j)))
128   end
129 end
130 if salida
131   x = x(1:i+1,:);
132   t = t(1:i+1,:);
133   theta_pp = theta_pp(1:i+1,:);
134   V_sh=V_sh(1:i+1);
135   break
136 end
137 end
138 return

```

Código B.4: dinamica_robot.m

```

1 % Función que proporciona la dinámica del robot en la forma
2 % {U'} = [A(U)]{U} + {F} para la integración mediante Runge-
3 % Kutta.
4
5 function [U_p] = dinamica_robot(U,L,mp, theta_0, M, omega, gamma
6   , mu, R,n, V_sh, beta,RB,Vc,Cr, paso,thetapp,U_ant,thetapp_ant
7   ,V_sh_ant)
8
9 =====

```

```

6  % - Variables de entrada:
7  %      U                               :: vector fila que
8  %      almacena, por este orden, los ángulos y, a continuación, las
9  %      derivadas de las patas.
10 %      L, mp, theta_0,M                :: parametros fí
11 %      sicos del problema
12 %      omega, gamma, mu, r, V_sh, beta, RB :: parametros de la
13 %      funcion Toda-Rayleigh
14 %      Vc                                :: Voltaje en el
15 %      condensador de control
16 %      Cr                                :: Variable para
17 %      activar (1) o desactivar del control (0)
18 %      n                                 :: numero de patas
19 %      paso                             :: indice del paso
20 %      temporal actual
21 %      thetapp                          :: aceleraciones de
22 %      las patas
23 %      U_ant, thetapp_ant             :: valores de á
24 %      ngulos,velocidades y aceleraciones en el instante anterior
25 %      V_sh_ant                         :: valor del voltaje
26 %      externo en el instante anterior
27
28 % - Variables de salida:
29 %      U_p                            :: derivada de U.
30 %      Vector fila que almacena, por este orden, velocidades
31 %      angulares y aceleraciones de las patas
32
33 % - Utiliza las funciones
34 %      TODAg.m
35 %      derTODAg.m
36
37 %
38 =====
39
40 % Parametros internos
41 E = 0;
42 tol = 1e-5; % tolerancia para que la matriz sea o no singular
43 control = true;
44 theta_pp = zeros(n,1);
45 estadop = ones(1,6);
46 Apoyadas = 0;
47 Levantadas = 0;
48 cambio = 0;
49 global Il
50 global estado
51 global x
52
53 % por comodidad de las expresiones, extraemos los ángulos del
54 % vector U
55 theta(:,1) = U(1:n);

```

```

42 thetap(:,1) = U(n+1:2*n);
43 theta_ant(:,1) = U_ant(1:n);
44 thetap_ant(:,1) = U_ant(n+1:2*n);
45
46 % Primero calculamos el valor de la función Toda en cada una de
   las patas
47 T = TODAg(theta, thetap, omega, gamma, mu, R, n, V_sh, beta, RB, Vc, Cr)
     ;
48
49 % Calculamos el valor de la derivada de la función Toda
50 dT = derTODAg(theta, thetap, thetapp, omega, gamma, mu, R, n, V_sh,
   beta, RB); % En el instante de integración
51 if paso ~= 1
52     dT_ant = derTODAg(theta_ant, thetap_ant, thetapp_ant, omega,
   gamma, mu, R, n, V_sh_ant, beta, RB); % En el instante anterior
53 end
54
55
56 %% CONDICIÓN DE PATAS LEVANTADAS/APOYADAS
57
58 k = 0;
59 l= 0;
60 for i = 1:1:n
61     condicion = false;
62
63 if paso ~= 1
64     if sign(dT(i))~= 1
65         condicion = true;
66         % Recogemos el caso en el que pasa a apoyarse
67         %(en la anterior iteración estaba apoyada) como si todav
           ía
68         %no se hubiese apoyado para asegurar que se transmita la
           oscilación correctamente
69     elseif sign(dT_ant(i))~= 1
70         condicion = true;
71     end
72 else
73     if sign(dT(i))~= 1
74         condicion = true;
75     end
76 end
77
78 if condicion
79     % Guardamos en un vector la numeración real de las patas
       apoyadas,
80     % si no están ya incluida
81     if i ~= Levantadas
82         l = l+1;
83         Levantadas(l) = i;
84     end
85 end

```

```

86     end
87 for i=1:1:n
88     if i ~= Levantadas % pata apoyadas
89         k = k+1;
90         Apoyadas(k) = i;
91         estadop(i) = 0;
92     end
93 end
94
95 npa = size(Apoyadas,2); % numero de patas apoyadas
96 npl = size (Levantadas,2); % numero de patas levantadas
97
98 % ----- CAMBIO DE VELOCIDAD DE LAS PATAS QUE PASAN A APOYARSE
99
100 % Descartamos el caso de la primera iteración
101 if size(estado,1) > 0
102     % Comprobamos si alguna de las patas pasa de levantada a
103     % apoyada y
104     % cambiamos su velocidad angular
105     k = 0;
106     for i=1:6
107         if estado(end,i) - estadop(1,i) == 1 % Se apoya cuando
108             pasa de 1 a 0
109             k = k+1;
110             cambio(k) = i; % almacenamos en un vector de
111             % longitud variables las patas que pasan a apoyarse
112             % en este instante
113         end
114     end
115     if cambio ~= 0
116         % buscamos una pata que este apoyada y que no haya
117         % cambiado de estado para
118         % utilizarla de referencia
119         if Apoyadas ~= 0
120             for i=1:npa
121                 if i ~= cambio % si la pata esta apoyada y antes
122                     no estaba levantada, la tomamos de referencia
123                     ref = Apoyadas(i);
124                     break
125                 else
126                     ref= i; % caso de que todas las patas
127                     cambien de estado, utilizamos una
128                     cualquiera de referencia
129                 end
130             end
131         else
132             ref = cambio(1); % caso limite: todas han cambiado
133             de estado al apoyarse, se utiliza una cualquiera
134             de referencia
135         end
136     end
137 
```

```

126     for i=1:size(cambio,2)
127         thetap(cambio(i),1) = thetap(ref)*cos(theta(cambio(i)))
128             )))^2/cos(theta(ref))^2;
129         x(paso, n + cambio(i)) = thetap(cambio(i),1); %
130             cambiamos la velocidad en la matriz global
131     end
132 end
133 % Actualizamos la matriz de estado
134 estado = [estado; estadop];
135 %% CALCULO DE LAS ECUACIONES DEL PROBLEMA
136
137 % Patas levantadas
138 if Levantadas ~= 0
139     for i=1:npl
140         % Ecuacion de la pata levantada
141         theta_pp(Levantadas(i),1) = T(Levantadas(i));
142     end
143 end
144
145 % Patas apoyadas
146 if Apoyadas ~= 0
147     % Creamos el sistema de ecuaciones de las patas apoyadas
148     A = zeros (npa,1);
149     B = zeros (npa,1);
150     C = zeros (npa,1);
151     D= zeros (npa,1);
152     AD = zeros (npa,npa);
153     CB = zeros (npa,npa);
154     while control
155         for k =1:npa
156             A(k) = L/(cos(theta(Apoyadas(k))))^2;
157             B(k) = 2*L*tan(theta(Apoyadas(k)))/(cos(theta(
158                 Apoyadas(k)))^2);
159             C(k) = (mp*L/(3*M*(cos(theta_0)^2)))*(-2*cos(theta_0)/
160                 cos(theta(Apoyadas(k))) + 4 *(cos(theta_0)/cos(
161                     theta(Apoyadas(k))))^2)*tan(theta(Apoyadas(k)));
162             I = (mp*L^2/3*(cos(theta_0)^2))*(1-2*cos(theta_0)/
163                 cos(theta(Apoyadas(k))) + 2*(cos(theta_0)/cos(
164                     theta(Apoyadas(k))))^2);
165             D(k) = I/(M*L);
166             E = E + I*T(Apoyadas(k))/(M*L); % Se multiplica por
167                 Il porque su valor ya esta incluido en el momento
168         end
169
170         for i=1:npa
171             for j = 1:npa
172                 if j == i
173                     AD (i,j) = D(i) + A(i);
174                     CB (i,j) = C(i) + B(i);

```

```

169         else
170             AD(i,j) = D(j);
171             CB(i,j) = C(j);
172         end
173     end
174 end
175 % Comprobamos si la matriz AD se puede invertir
176 if abs(det(AD)) <= tol
177     % si la matriz es singular, le sumamos una cantidad
178     % pequeña a los
179     % ángulos, de forma que deje de ser singular, pero
180     % apenas afecte a
181     % la resolución del sistema
182     theta = theta + theta/100000;
183     disp ('Problema, posible bucle infinito')
184 else
185     control = false;
186 end
187
188 % rellenamos un vector con el valor de las velocidades
189 % angulares al
190 % cuadrado
191 theta_p2 = ones(npa,1);
192 for k=1:npa
193     theta_p2(k,1) = thetap(Apoyadas(k))^2;
194 end
195
196
197 % Introducimos los valores calculados en la numeracion
198 % orginal
199 for k =1:npa
200     theta_pp(Apoyadas(k),1)= theta_pp_apoyadas(k);
201 end
202
203 %
204 % Rellenamos el valor del vector U'
205 U_p = vertcat(thetap, theta_pp);
206 %U_p = vertcat(U(n+1:2*n)', theta_pp)
207 % Devolvemos el valor de U_p como vector fila, para usarlo en el
208 % % Runge—Kutta
209 %U_p = theta_pp;
210 U_p = U_p';
211 return

```

Código B.5: TODAg.m

```

1 % Calculo del valor de la función de Toda-Rayleigh
2
3 function f = TODAg(Xn,XnPrim,omega,gamma,mu,R,n,V_sh,beta,RB,Vc,
4 % Cr)
5 % =====
6 % - Variables de entrada:
7 % Xn :: Valor de theta en un paso de integracion [V]
8 % 1xn.
9 % XnPrim :: Valor de la primera derivada de theta en un
10 % paso de integracion [V] 1xn.
11 % omega :: Parametro del problema
12 % gamma :: Parametro del problema
13 % mu :: Parametro del problema [V^2/s^2].
14 % n :: Numero de nodos del problema.
15 % R :: Resistencia conectada a tierra [Ohm].
16 % V_sh :: Voltaje de control [V].
17 % beta :: parametro del problema [1/s].
18 % RB :: Resistencia del sistema de control [V].
19 % Vc :: Valor de Vc para un paso de integracion [V].
20 % Cr :: Variable de activacion y desactivacion del
21 % control (0 desactivado, 1 activado).
22 %
23 % =====
24
25
26 %Parámetros internos
27 Is = 1.5e-9;
28 Vt = 0.05;
29
30 %Creamos la variable de salida.
31 f = zeros(1,n);
32 %Calculamos f para las patas de la 1 a la n.
33 for i = 1:1:n
34
35 %Guardamos la pata anterior a la que estamos iterando, si es
36 %la primera
37 %la anterior sera la ultima.
38 if i == 1
39     mmenos = n;
40 else
41     mmenos = i-1;
42 end

```

```

43 %Guardamos la pata siguiente a la que estamos iterando, si
44 %es la
45 %ultima la siguiente sera la primera.
46 if i == n
47     mmas = 1;
48 else
49     mmas = i+1;
50 end
51
52 %Calculamos f.
53 f(i) = omega*omega*...
54     (Is*(exp((Xn(mmenos)-Xn(i))/Vt)-exp((Xn(i)-Xn(mmas))/Vt)
55     )...
56 +gamma*(mu-(XnPrim(i)-beta*V_sh)*(XnPrim(i)-beta*V_sh))
57     *(XnPrim(i)+beta*V_sh)...
58 -Xn(i)/R - (Xn(i)-Cr*Vc)/RB);
59 end
60
61 end

```

Código B.6: derTODAg.m

```

1 % Función que calcula la derivada analítica de la función de
2 % Toda-Rayleigh
3
4 function f = derTODAg(Xn ,XnPrim ,XnPrim2 ,omega ,gamma ,mu ,R ,n ,V_sh ,
5 % beta ,RB)
6 %
7 % - Variables de entrada:
8 %      Xn       :: Valor de theta en un paso de integracion [V]
9 %      1xn.
10 %      XnPrim   :: Valor de la primera derivada de theta en un
11 %      paso de integracion [V] 1xn.
12 %      omega     :: Parametro del problema
13 %      gamma    :: Parametro del problema
14 %      mu        :: Parametro del problema [V^2/s^2].
15 %      n         :: Numero de nodos del problema.
16 %      R         :: Resistencia conectada a tierra [Ohm].
17 %      V_sh     :: Voltaje de control [V].
18 %      beta      :: parametro del problema [1/s].
19 %      RB        :: Resistencia del sistema de control [V].
20 %
21 % - Variables de salida
22 %      f         :: Valor de la derivada de la función de Toda-
23 %      Rayleigh 1xn
24 %
25 %

```

```

22
23 %Parámetros internos
24 Is = 1.5e-9;
25 Vt = 0.05;
26
27 %Creamos la variable de salida.
28 f = zeros(1,n);
29 %Calculamos f para las patas de la 1 a la n.
30 for i = 1:1:n
31
32     %Guardamos la pata anterior a la que estamos iterando, si es
            la primera
33     %la anterior sera la ultima.
34     if i == 1
35         mmenos = n;
36     else
37         mmenos = i-1;
38     end
39
40     %Guardamos la pata siguiente a la que estamos iterando, si
            es la
41     %ultima la siguiente sera la primera.
42     if i == n
43         mmas = 1;
44     else
45         mmas = i+1;
46     end
47
48     %Calculamos f.
49     f(i) = omega*omega*...
50         (Is*((XnPrim(mmenos)-XnPrim(i))/Vt)*exp((Xn(mmenos)-Xn(
51             i))/Vt) - ((XnPrim(i)-XnPrim(mmas))/Vt)*exp((Xn(i)-Xn(
52                 mmas))/Vt))...
53         +gamma*(2*XnPrim2(i)*(XnPrim(i)-beta*V_sh)*(XnPrim(i)-
54             beta*V_sh)+(mu-(XnPrim(i)-beta*V_sh)*(XnPrim(i)-beta*
55                 V_sh))*XnPrim2(i))...
56         -XnPrim(i)/R - XnPrim(i)/RB);
57 end
58 end

```

Código B.7: RKcontrol.m

```

1 % Runge-Kutta que calcula el voltaje medio en el condensador de
      control
2 function [VcEx] = RKcontrol(Xn,Vc,RB,C,dt)
3 %
4 =====
5 % - Variables de entrada:
6 %       Xn          :: Valor de theta en un paso de integracion [V]
7 %       1xn.

```

```

6   %      Vc      :: Valor de Vc para un paso de integracion [V].
7   %      RB      :: Resistencia del sistema de control [V].
8   %      C       :: Capacitancia del condensador del control [F].
9   %      dt      :: Paso de integración [s].
10 %
11 % - Variables de salida:
12 %      f       :: Valor de Vc en el siguiente instante de
13 %                  tiempo [V/s].
14 %
15 % - Funciones utilizadas:
16 %      CONTROL.m
17 %
18 =====
19
20
21
22
23
24

```

Código B.8: CONTROL.m

```

1   % La funcion CONTROL(Xn,Vc,RB,C) calcula la solucion de la
2   % ecuacion de control.
3   function f = CONTROL(Xn,Vc,RB,C)
4   %
5   =====
6
7
8
9
10 %
11 % - Variables de entrada:
12 %      Xn      :: Valor de theta en un paso de integracion [V]
13 %      1xn.
14 %
15 %      Vc      :: Valor de Vc para un paso de integracion [V].
16 %      RB      :: Resistencia del sistema de control [V].
17 %      C       :: Capacitancia del condensador del control [F].
18 %
19 % - Variables de salida:
20 %      f       :: Solucion de la ecuacion de control [V/s].
21 %
22
23
24

```

Código B.9: cinematica_CM.m

```

1   % Calculo de la cinematica del centro de masas a partir de la
2   % cinemática de

```

```

2 % cada una de las patas (ecuaciones de ligadura 2.5 y 2.6)
3 function [Xcm,Xcm_p, Xcm_pp, XCM, XCM_p,F] = cinematica_CM(theta
    , thetap, thetapp, L,t,n, X0, Xp0,omega, gamma,mu,R,V_sh,beta,
    RB,theta_0,ac,mp)
4 %
=====

5 % - Variables de entrada:
6 %     theta, thetap, thetapp          :: matrices que
    almacenan, por filas, posicion, velocidades y aceleraciones
    angulares en funcion del tiempo.
7 %     L                               :: parametro geométrico
    del problema
8 %     t                               :: vector columna que
    almacena el tiempo
9 %     n                               :: numero de patas
10 %    X0, Xp0                         :: posicion y velocidad
    incial del CM
11 %    omega, gamma,mu,R,V_sh,beta,RB :: parámetros de la
    función de Toda-Rayleigh

12
13 % - Variables de salida:
14 %     U_p                           :: derivada de U.
    Vector fila que almacena, por este orden, velocidades
    angulares y aceleraciones de las patas
15 %     Xcm,Xcm_p, Xcm_pp, XCM, XCM_p   :: matrices que
    almacenan posicion, velocidad y aceleracion del CM calculadas
    con cada una de las patas en funcion de t
16 %     XCM, XCM_p                     :: vector con la
    posicion del CM, utilizando solo una de las patas apoyadas en
    cada instante (para salida de resultados

17
18 % - Utiliza las funciones:
19 %     derTODAg.m
20 %

=====

21
22
23
24
25 % Iniciamos las variables
26 Xcm =X0*ones(size(t,1),6);
27 Xcm_p =Xp0*ones(size(t,1),6);
28 Xcm_pp =zeros(size(t,1),6);
29 XCM = X0*ones(size(t,1),1);
30 XCM_p= Xp0*ones(size(t,1),1);
31 Xcm(1,:) = X0;
32 Xcm_p(1,:) = Xp0;
33 XCM(1,:) = X0;
34 XCM_p(1,:) = Xp0;

```

```

35 X_0 = X0*ones(1,6);
36 init = ones(1,6);
37 Levantadas_anterior = zeros(1,6);
38 F = zeros(size(t,1),6);
39
40 for j =1:size(t,1)
    %% CRITERIO DE PATAS LEVANTADAS
    % Analizamos que pata o patas estan levantadas a tiempo j
    clear Apoyadas
    clear Levantadas
    clear error
    Apoyada = false;
    Levantadas = 0;
    k= 0;
    l = 0;
    % Calculamos el valor de la derivada de la función Toda
    dT = derTODAg(theta(j,:), thetap(j,:),thetapp(j,:), omega,
                    gamma,mu,R,n,V_sh(j),beta,RB);
    if j ~= 1
        dT_ant = derTODAg(theta(j-1,:), thetap(j-1,:),thetapp(j
                    -1,:), omega, gamma,mu,R,n,V_sh(j-1),beta,RB);
    end
    55
    for i=1:1:n
        condicion = false;
        if j ~= 1
            if sign(dT(i))~= 1
                condicion = true;
                % Recogemos el caso en el que pasa a apoyarse
                % como si todavía no se hubiese apoyado
            elseif sign(dT_ant(i))~= 1
                condicion = true;
            end
        else
            if sign(dT(i))~= 1
                condicion = true;
            end
        end
        66
        if condicion % condicion de pata apoyada
            l = l+1;
            Levantadas(l) = i;
        else
            k = k+1;
            Apoyadas(k) = i;
            Apoyada = true;
        end
        79
    end
    81 if Apoyada
        for i = 1:size(Apoyadas,2)
            if any(Apoyadas(i) == Levantadas_anterior)

```

```

84           X_0(:) = XCM(j);
85           init(:) = j;
86       end
87   end
88   for i = 1:size(Apoyadas,2)
89       if j ~= size(t,1)
90           Xcm(j+1,Apoyadas(i))= -L*(tan(theta(j+1,Apoyadas
91               (i))) -tan(theta(init(Apoyadas(i)),Apoyadas(i))
92               )) +X_0(Apoyadas(i)); % Posición CM
93       end
94       Xcm_p(j,Apoyadas(i)) = -(L*thetap(j,Apoyadas(i))/(
95           cos(theta(j,Apoyadas(i)))^2)); %+ vel_pata(Apoyadas
96               (i)); % Velocidad CM
97       Xcm_pp(j,Apoyadas(i)) =-( L*thetapp(j,Apoyadas(i))/(
98           cos(theta(j,Apoyadas(i)))^2) + (2*L*tan(theta(j,
99               Apoyadas(i)))*(thetap(j,Apoyadas(i))^2))/(cos
100                  theta(j,Apoyadas(i)))^2)); % Aceleración CM
101   end
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121

```

```

122      Ip = (mp*L/(3*(cos(theta_0)^2)))*(-2*cos(theta_0)/
123          cos(theta(j,Apoyadas(k))) + 4 *(cos(theta_0)/cos
124          (theta(j,Apoyadas(k))))^2)*tan(theta(j,Apoyadas(
125          k)));
126      F(j,Apoyadas(k)) = Ip*thetap(j,Apoyadas(k))^2/L +
127          I*thetapp(j,Apoyadas(k))/L - ac(j,Apoyadas(k))/L
128          ;
129      end
130  else
131      if j ~= size(t,1)
132          XCM(j+1) = XCM(j);
133      end
134      if j ~= 1
135          XCM_p(j)= XCM_p(j-1);
136      end
137
138  end
Levantadas_anterior = Levantadas;

```

Bibliografía

- [1] Dean J, Kindermann T, Schmitz J, Schumm M and Cruse H, 1999. “Control of walking in the stick insect: from behavior and physiology to modeling,” *Autonomous Robots*, vol. 7, pp. 271-288.
- [2] Collins J J and Stewart I, 1993a. “Coupled nonlinear oscillators and the symmetries of animal gaits,” *Nonlinear Sci.*, **3**, 349–392 (1993)
- [3] Collins J J and Stewart I, 1993b “Hexapodal gaits and coupled nonlinear oscillator models,” *Biol. Cyb.*, **68**, 287–298 (1993)
- [4] Collins J J and Stewart I, 1994. “A group-theoretic approach to rings of coupled biological oscillators,” *Biol. Cyb.*, **71**, 95–103 (1994)
- [5] Golubitsky M, Stewart I, Buono P L and Collins J J, 1998. “A modular network for legged locomotion,” *Physica D*, **115**, 56–72, (1998).
- [6] Golubitsky M, Stewart I, Buono P L and Collins J J, 1999. “The role of symmetry in animal locomotion,” *Nature*, **401**, 693–695 (1999)
- [7] Del Rio E, Makarov V A, Velarde M G and Ebeling W, 2003. “Mode transitions and wave propagation in a driven-dissipative Toda-Rayleigh ring,” *Phys. Rev. E*, **67**, pp. 056208–056217 (2003)
- [8] Makarov V A, Ebeling W and Velarde M G, “Soliton-like waves on dissipative Toda lattices,” *Int. J. Bifurcation Chaos*, **10**, pp. 1075-1089 (2000)
- [9] Makarov V A, del Rio E, Ebeling W and Velarde M G, 2001. “Dissipative Toda-Rayleigh lattice and its oscillatory modes,” *Phys. Rev. E*, **64**, 036601–36615 (2001)
- [10] Toda M, *Theory of Nonlinear Lattices*. Berlin: Springer. (1981)
- [11] Toda M, *Nonlinear Waves and Solitons*. Dordrecht: Kluwer. (1983)
- [12] Rayleigh J. W, *The Theory of Sound*. 2nd ed. New York: Dover reprint, sec. 68a. (1945)
- [13] Schmitz J, Dean J, Kindermann T, Schumm M and Cruse H, 2001, “A biologically inspired controller for hexapod walking: simple solutions by exploiting physical properties,” *Biol. Bull.*, **200**, 195–200 (2001)
- [14] Singer A C and Oppenheim A V, 1999, “Circuit implementations of soliton systems,” *Int. J. Bifurcation Chaos*, **9**, 571–590 (1999)

- [15] Yu. A. Kuznetsov, *Elements of Applied Bifurcation Theory*. Berlin: Springer-Verlag. (1995)
- [16] Gómez Ortega, Omar, *Estudio de la dinámica de un anillo de Toda para su aplicación a la locomoción de un robot con patas*. Madrid, 2014
- [17] del Río E. and Velarde M.G. “A prototype 2N-legged (insect-like) robot. A non-linear dynamical system approach”