

# Modelado Dinámico e Identificación de un Robot Hexápodo

Mazzanti, Matías - Reto Reynal, Juan

Laboratorio 7

DEPTO. DE FÍSICA – FCEYN - UBA

DICIEMBRE 2018

ALUMNOS: Matías Mazzanti - 110/13 - mazzantimatiass@gmail.com  
Juan Reto Reynal - 777/12 - juanreto@gmail.com

LUGAR DE TRABAJO: Laboratorio de Robótica y Sistemas Embebidos,  
DC – FCEyN - UBA  
<https://robotica.dc.uba.ar/?lang=es>

DIRECTOR DEL TRABAJO: De Cristóforis, Pablo - pdecris@gmail.com

CODIRECTOR: Pessacg, Facundo - facu.pessacg@gmail.com

# Índice

<b>1. Motivaciones</b>	<b>1</b>
<b>2. Introducción Teórica</b>	<b>1</b>
2.1. Cinemática directa e inversa . . . . .	2
2.2. Modelo electromecánico de un servomotor . . . . .	4
2.3. Modelo dinámico de una pata . . . . .	5
2.4. Modelo dinámico completo de una pata . . . . .	7
2.5. Patrones de Movimiento ( <i>Gaits</i> ) y formulación del problema . . . . .	7
<b>3. Desarrollo experimental</b>	<b>9</b>
3.1. Características de los servomotores Dynamixel AX-12 . . . . .	9
3.2. Cinemática . . . . .	12
3.2.1. Cinemática Directa . . . . .	12
3.2.2. Cinemática Inversa . . . . .	12
3.3. Caracterización y simulación de un servomotor . . . . .	12
3.4. Dinámica de una pata completa . . . . .	13
<b>4. Resultados y discusiones</b>	<b>14</b>
4.1. Cinemática . . . . .	14
4.1.1. Cinemática Directa . . . . .	15
4.1.2. Sensibilidad del modelo con los parámetros de D-H. . . . .	16
4.1.3. Cinemática Inversa . . . . .	17
4.2. Espacio de Trabajo - <i>Workspace</i> . . . . .	17
4.3. Resultados dinámica completa de una pata . . . . .	18
4.3.1. Caracterización y simulación de un servomotor. . . . .	19
4.3.2. Simulación completa de una pata . . . . .	21
4.3.3. Análisis de los errores . . . . .	22
4.3.4. Análisis de la velocidad en los umbrales . . . . .	23
4.3.5. Análisis de los errores en la coxa . . . . .	25
<b>5. Conclusiones</b>	<b>25</b>
5.1. Conclusiones Laboratorio 6 . . . . .	25
5.2. Conclusiones Laboratorio 7 . . . . .	26
<b>6. Referencias</b>	<b>27</b>

# Índice de figuras

1.	Robot PhantomX AX Metal Hexapod Mark III . . . . .	1
2.	Ángulos de las articulaciones. . . . .	2
3.	Parámetros del modelo de Denavit-Hartenberg. . . . .	2
4.	Transformación entre el cuerpo y la coxa. . . . .	3
5.	Sistemas de coordenadas de las articulaciones. . . . .	3
6.	Esquema electromecánico de un motor de corriente continua. . . . .	5
7.	Configuración de una pata. . . . .	6
8.	Combinación de los modelos de una cadena de cuerpos rígidos y electromecánico. . . . .	7
9.	Fases de Gait estándar. . . . .	8
10.	Gait estándar (comercial). . . . .	8
11.	Planteo del problema: detectar colisiones. . . . .	8
12.	Gait adaptativo. . . . .	9
13.	Potenciómetro (sensor de posición) señalado con una flecha. . . . .	10
14.	Potenciómetro (sensor de posición) de color azul. . . . .	10
15.	Configuraciones angulares posibles del servomotor Dynamixel AX-12 . . . . .	10
16.	Simulación de una interpolación lineal entre una posición inicial y final. . . . .	11
17.	Odroid XU4 montada en el hexápodo. 8.5cm x 5.5 cm. . . . .	14
18.	Simulación de 1 pata. . . . .	15
19.	Simulación del Robot Completo. . . . .	15
20.	Error en la coordenada z para los parámetros $\theta_F^{Off}$ y $\theta_T^{Off}$ . . . . .	16
21.	Error en la norma para los parámetros $\theta_F^{Off}$ y $\theta_T^{Off}$ . . . . .	16
22.	Error en la coordenada z para los parámetros $a_C, a_F$ y $a_T$ . . . . .	17
23.	Error en la norma para los parámetros $a_C, a_F$ y $a_T$ . . . . .	17
24.	Espacio de trabajo: Plano XY. . . . .	18
25.	Espacio de trabajo: Plano YZ. . . . .	18
26.	Relación lineal entre la tensión aplicada al motor y la velocidad del mismo. . . . .	19
27.	Simulación de la velocidad del motor para el caso estacionario sin torque de gravedad. . . . .	19
28.	Relación lineal entre la tensión aplicada al motor y la velocidad del mismo para diferentes casos. . . . .	20
29.	Simulación de la velocidad del motor para el caso con carga y torque de gravedad. . . . .	21
30.	Posición medida en ticks en función del tiempo junto a la simulación de la pose real y de la pose objetivo. $\Delta t = 1s$ . . . . .	21
31.	Posición medida en ticks en función del tiempo junto a la simulación de la pose real y de la pose objetivo. $\Delta t = 0.5s$ . . . . .	21

32.	Error de la posición real y objetivo en función del tiempo. . . . .	22
33.	Error de la posición real y objetivo en función del tiempo, con torque de gravedad a favor. . . . .	23
34.	Error de la posición real y objetivo en función del tiempo, con torque de gravedad en contra. . . . .	23
35.	Errores en la coxa - Pata estirada. . . . .	25
36.	Errores en la coxa - Pata contraída. . . . .	25

# 1. Motivaciones



Figura 1: Robot PhantomX AX Metal Hexapod Mark III

La principal ventaja de los robots hexápodos es su capacidad para desplazarse en diversos tipos de terrenos [6] [7], que resultan imposibles de atravesar con sistemas de locomoción tradicionales basados en ruedas u orugas. Una aplicación importante de estos robots es en misiones de búsqueda y rescate urbano (USAR, *Urban, Search and Rescue*) en edificios derrumbados o áreas afectadas por desastres.

Sin embargo, la gran cantidad de grados de libertad (*degrees of freedom (DoF)*) que poseen (en general entre 18 y 24) dificulta el diseño de algoritmos de control del movimiento del robot. Es por ello que es necesario usar el modelo cinemático del robot en la planificación de los movimientos de los motores. Además, el modelo dinámico del robot puede ser utilizado para caminar sobre terrenos irregulares [8], detectar colisiones o incluso estimar la pose del robot. Debido a la complejidad de estos sistemas, producto de la cantidad de grados de libertad involucrados, se suelen utilizar simplificaciones de los modelos cuya calidad impacta directamente en la performance del robot.

Este trabajo propone evaluar la calidad del modelo cinemático de un robot hexápodo. Además, generalizar el modelo dinámico teniendo en cuenta la dinámica de una pata completa. De esta forma se espera mejorar la traversabilidad y control de los robots tipo hexápodos a través de la detección de colisiones. Dichas detecciones no son consideradas por los modelos comerciales como el utilizado en este trabajo.

# 2. Introducción Teórica

A continuación se explican las bases teóricas del trabajo que permiten describir matemáticamente y físicamente un robot hexápodo: el modelo de la cinemática directa e inversa, el modelo dinámico de una pata completa contemplando las ecuaciones electromecánicas de un servomotor y las ecuaciones

de Euler-Lagrange para modelar una cadena de cuerpos rígidos. También se explican los diferentes patrones de movimiento (*gaits*) a utilizar junto al planteo del problema de cómo realizar la detección de colisiones.

## 2.1. Cinemática directa e inversa

Se puede pensar el problema de dos maneras distintas: el modelo cinemático directo (MCD), que permite determinar la posición y orientación final de cada extremo de la pata a partir de asignar ángulos a cada articulación que se muestran en la Fig. 2; y el modelo cinemático inverso (MCI), que permite calcular los posibles ángulos de cada motor para determinar una posición final de cada pata deseada.

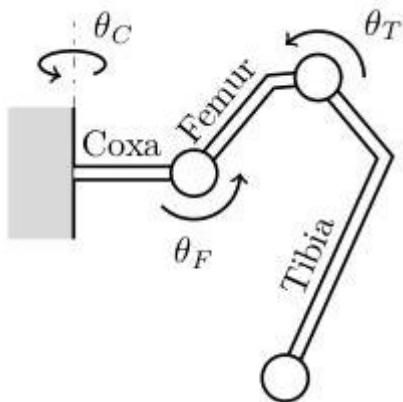


Figura 2: Ángulos de las articulaciones.

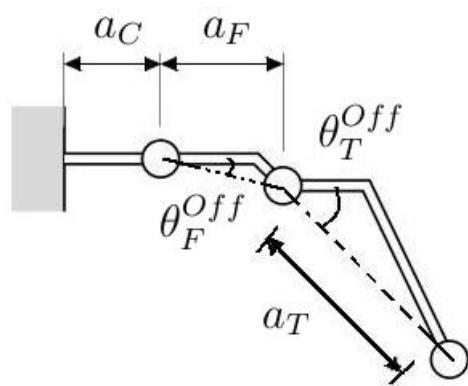


Figura 3: Parámetros del modelo de Denavit-Hartenberg.

Un robot hexápodo está compuesto por cuerpos rígidos (eslabones) unidos en serie por medio de articulaciones. Para describir la relación entre las articulaciones de la plataforma se utiliza un sistema de coordenadas homogéneas. La matriz de transformación homogénea  $T$  que relaciona dos sistemas de referencia contiene una matriz de rotación  $R$  y una traslación dada por un vector  $\mathbf{r}$ , que tiene la siguiente expresión:

$$T = \begin{bmatrix} R & \mathbf{r} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Entonces una transformación homogénea es una matriz que relaciona dos sistemas de referencia que pueden estar rotados y/o trasladados uno respecto del otro. Una propiedad muy importante y útil de estas transformaciones es que se pueden componer y de esta forma se puede hallar la transformación que vincula la posición final de cada pata con el centro del robot como se lo explica a continuación.

Para poder describir la cinemática del robot se tienen que definir de forma apropiada los sistemas de coordenadas de cada articulación. Se utilizó la convención estándar de Denavit-Hartenberg (D-H) [5], la cual define una transformación homogénea entre dos o más sistemas de coordenadas

homogéneas, dados ciertos parámetros característicos como la longitud de las articulaciones y los ángulos subyacentes entre las articulaciones ( $\theta_{Off}$ , de *offset*) como se observa en la Fig. 3.

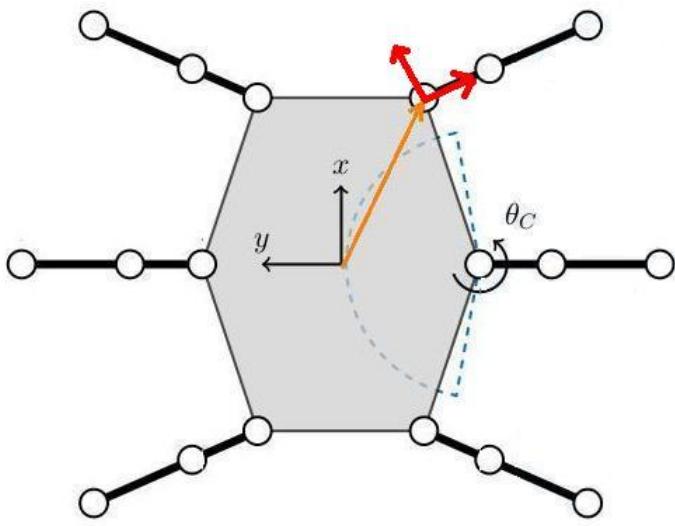


Figura 4: Transformación entre el cuerpo y la coxa.

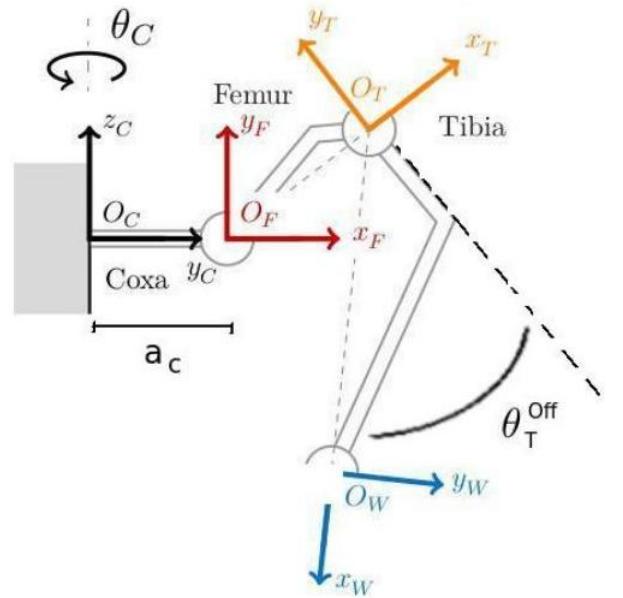


Figura 5: Sistemas de coordenadas de las articulaciones.

Para describir por medio de la cinemática directa la posición de la punta de la pata (*foottip*) respecto del centro del robot hace falta en primer lugar una transformación,  $P_l$ , entre el sistema de coordenadas del cuerpo y la coxa como se muestra en la Fig. 4, donde el vector naranja representa la traslación entre ambos sistemas de coordenadas y donde la rotación para alinear la orientación debe ser llevada a cabo entre los ejes negros  $x, y$  y los marcados en rojo de la coxa. En segundo lugar, una vez llegado a la coxa de una de las patas se debe componer las matrices de transformación entre cada articulación como se muestra en la Fig. 5 para poder finalmente determinar la posición de la *foottip*. Se hace notar que el procedimiento para cada una de las patas es análogo por la simetría del robot. Matemáticamente, la posición de la *foottip* resulta:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = P_l A_F^C(\theta_C) A_T^F(\theta_F) A_W^T(\theta_T) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Las matrices  $A_i^{i-1}$  relacionan dos articulaciones consecutivas, es decir, la matriz  $A_F^C(\theta_C)$  es la matriz de transformación entre la coxa y el femur. Se hace notar que se agrega un sistema de referencia adicional en la punta de la pata para poder computar la *foottip* donde el vector  $(0, 0, 0, 1)$  está compuesto en sus primeras tres componentes por el vector nulo que expresa la posición de la *foottip* visto desde su propio sistema de referencia y el 1 de la cuarta componente está puesto para conservar las dimensiones y simetría de las matrices de transformación.

Por medio de la cinemática inversa se pueden determinar los ángulos de las articulaciones conocida la posición final de la pata. Se utilizó el modelo propuesto por [2] que consiste en obtener los ángulos de los motores por medio de argumentos puramente geométricos como es estándar en la literatura. Se hace notar que se obtuvieron diferencias con los resultados obtenidos por la referencia citada arriba, pudiéndose optimizar la cinemática inversa implementada en el robot real. Para futuras referencias se muestran los resultados obtenidos de los ángulos de las articulaciones por medio de la cinemática inversa implementados en el robot real:

$$\theta_{Coxa} = \text{atan2}(y_{aux}, x_{aux}) - \beta_l$$

$$\theta_{Femur} = \frac{\arccos(a_F^2 - a_T^2 + p^2)}{2a_F a_T} - \theta_F^{Off}$$

$$\theta_{Tibia} = -\pi + \arccos\left(\frac{a_T^2 + a_F^2}{2a_F a_T}\right) - \theta_T^{Off}$$

## 2.2. Modelo electromecánico de un servomotor

El primer paso para poder desarrollar un modelo dinámico del cuerpo completo consiste en tener en cuenta el modelo electromecánico de los motores utilizados en este tipo de robots.

En general los robots hexápodos utilizan servomotores, que suelen estar compuestos por un motor de corriente continua, un *encoder* (dispositivo que proporciona la posición angular del motor), un driver que maneja la potencia que el motor necesita y un microcontrolador, que se encarga del control y la comunicación con el motor. Se utilizó el modelo estándar [1]. Dicho modelo establece una relación entre los fenómenos mecánicos y los fenómenos eléctricos del motor, que viene dada por la Ec. 1:

$$J_{eff}\ddot{\theta}(t) = \frac{r k_m}{R_a} u(t) - B_{eff}\dot{\theta}(t) - r\tau_{fs} - r\tau_{fc} - r^2\tau_g \quad (1)$$

donde  $J_{eff}$  es el momento de inercia efectivo del motor,  $r$  es la reducción del motor,  $k_m$  es la constante del torque,  $R_a$  es la resistencia del motor,  $B_{eff}$  es la constante efectiva de fricción viscosa,  $\tau_{fs}$  y  $\tau_{fc}$  son los coeficientes de fricción estática y de Coulomb respectivamente y  $\tau_g$  es el torque de gravedad.

Las variables que dependen del tiempo  $u(t)$ ,  $\dot{\theta}(t)$  y  $\ddot{\theta}(t)$  son la diferencia de potencial aplicada al motor, la velocidad y la aceleración del motor respectivamente. En la Fig. 6 se observa un esquema de un motor de corriente continua, donde por conveniencia las constantes ya definidas son combinaciones de las mostradas.

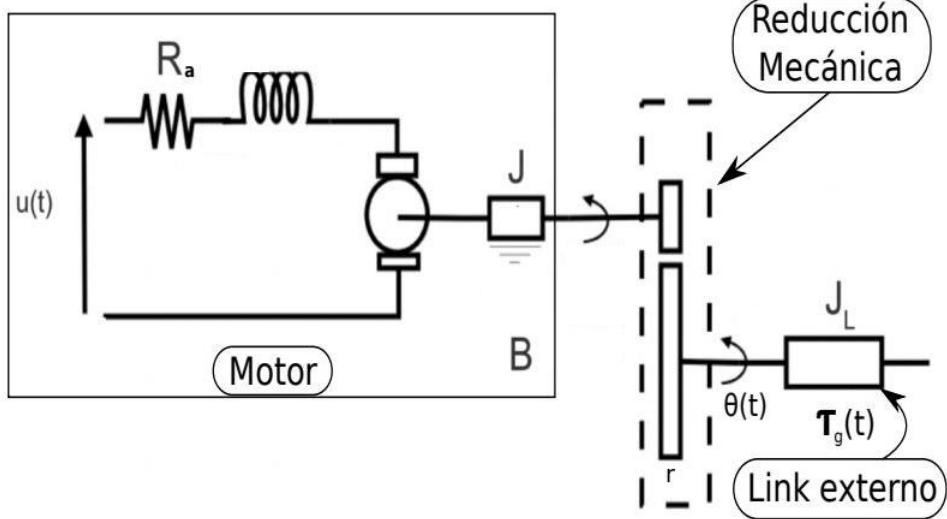


Figura 6: Esquema electromecánico de un motor de corriente continua.

Se observa que la Ec. 1 para el caso estacionario se simplifica pues se tiene que la aceleración  $\ddot{\theta}$  y el coeficiente de fricción estática  $\tau_{fs}$ , son nulos. Si además no se considera el torque de gravedad y redefiniendo algunas constantes se obtiene:

$$\dot{\theta}_{est} = \alpha_1 u + \alpha_2 sgn(\dot{\theta}_{est}) \quad (2)$$

donde  $\alpha_1 = \frac{rk_m}{R_a B_{eff}}$ ,  $\alpha_2 = \frac{r\tau_c}{B_{eff}}$  y  $sgn(\dot{\theta}_{est})$  es una función partida que vale 1, 0 ó -1 si  $\dot{\theta}_{est}$  es positiva, nula ó negativa respectivamente.

### 2.3. Modelo dinámico de una pata

Una pata completa compuesta de las 3 articulaciones coxa, femur y tibia puede ser modelada como una cadena de cuerpos rígidos, la cual puede ser descripta utilizando el formalismo de Euler-Lagrange donde el torque de cada articulación es:

$$\tau = \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta}$$

donde  $\mathcal{L}$  es el lagrangiano del sistema que está formado por la diferencia entre la energía cinética  $K$  y la energía potencial  $P$ , que tienen las siguientes expresiones:

$$K(\theta) = \frac{1}{2} \dot{\theta}^T D(\theta) \dot{\theta}$$

$$P(\theta) = \sum_{i=1}^n g^T a_{c_i}(\theta) m_i$$

donde  $g^T$  es el vector de gravedad traspuesto,  $a_{c_i}$  indica la posición del centro de masa de cada articulación, cuyo módulo se muestra en la Fig. 7.

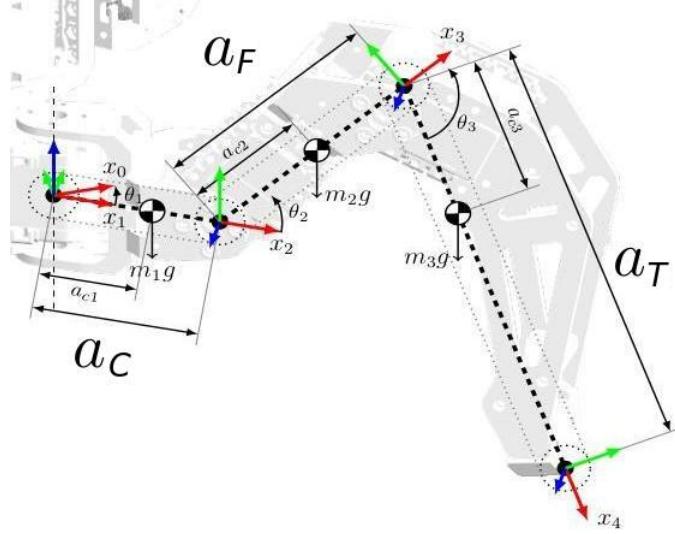


Figura 7: Configuración de una pata.

Resolviendo las ecuaciones se puede escribir de forma matricial el torque de cada articulación:

$$\boxed{\mathbf{D}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau}} \quad (3)$$

donde  $\mathbf{C}$  es el vector centrífugo y de Coriolis,  $\mathbf{G}$  es el vector de gravedad y  $\mathbf{D}$  es la matriz de inercia que tiene la siguiente expresión:

$$\mathbf{D}(\boldsymbol{\theta}) = \sum_{i=1}^n [m_i J_{v_i}^T J_{v_i} + J_{w_i}^{T0} R_i I_i^0 R_i^T J_{w_i}]$$

con  $m_i$  la masa de cada articulación,  $J_{v_i}$  y  $J_{w_i}$  los jacobianos asociados a la velocidad de traslación y de rotación respectivamente,  $R_i$  la matriz de rotación que relaciona la orientación del sistema base fijo a la coxa y cada articulación  $i$ -ésima respectivamente y finalmente  $I_i$  es el tensor de inercia de cada articulación.

Teniendo en cuenta la pata completa, se obtiene la siguiente matriz  $\mathbf{D}_w$  que coincide con las ecuaciones de [9] (ver pág. 173) por un camino alternativo:

$$\mathbf{D}_w = \begin{bmatrix} I_{133} + I_{233}\cos^2(\theta_2) + I_{333}\cos^2(\theta_2 + \theta_3) & 0 & 0 \\ 0 & I_{233} + I_{333} & I_{333} \\ 0 & I_{333} & I_{333} \end{bmatrix}$$

Para verificar la validez del resultado, se consideró el caso que no tiene en cuenta la coxa que es una configuración idéntica a la de un brazo robótico de dos grados de libertad y que coincide con la Ec.

6.79 de [5] donde la matriz  $\mathbf{D}_w$  toma la siguiente forma:

$$\mathbf{D}_w = \begin{bmatrix} I_{233} + I_{333} & I_{333} \\ I_{333} & I_{333} \end{bmatrix}$$

## 2.4. Modelo dinámico completo de una pata

Si se tienen en cuenta el modelo electromecánico dado por la Ec. 1 y el modelo dinámico de la cadena de cuerpos rígidos de la Ec. 3, se pueden combinar ambas ecuaciones para obtener el modelo dinámico completo de una pata.

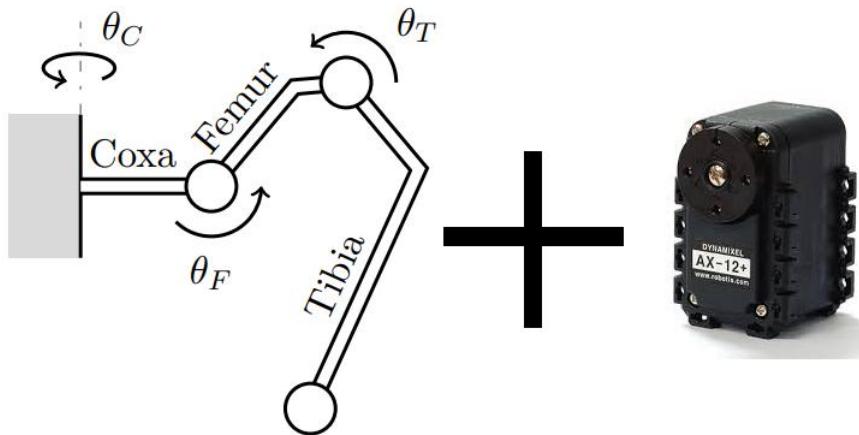


Figura 8: Combinación de los modelos de una cadena de cuerpos rígidos y electromecánico.

De esta forma se puede obtener una expresión final para la tensión  $\mathbf{V}$  aplicada a cada servomotor que tiene la misma forma funcional que la Ec. 3:

$$\mathbf{D}'(\theta)\ddot{\theta} + \mathbf{C}'(\theta)\dot{\theta} + \mathbf{F}_{fric}(\dot{\theta}) + \mathbf{G}'(\theta) = \mathbf{K}'\mathbf{V} \quad (4)$$

## 2.5. Patrones de Movimiento (*Gaits*) y formulación del problema

Los robots hexápodos pueden caminar de diversas maneras y de forma repetitiva, pudiéndose distinguir dos fases. La fase estable en la cual todas las patas están apoyadas sobre el terreno (*support phase*). La fase de movimiento (*swing phase*) en la cual una cantidad a determinar de patas están en movimiento sin estar tocando el terreno. Este proceso puede ser esquematizado en la Fig.9.

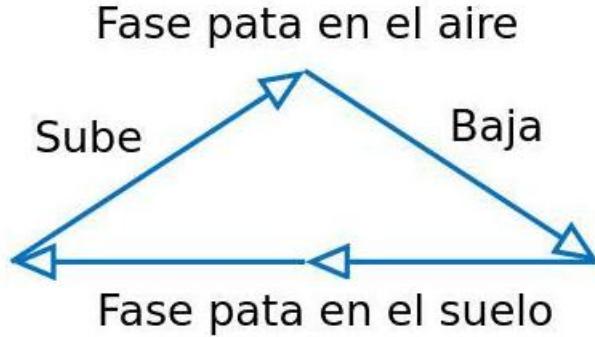


Figura 9: Fases de Gait est\'andar.

Algunos ejemplos de los diferentes *gaits* son el *pentapod gait* donde se mueve una pata a la vez ó el *tripod gait* donde se mueven tres patas a la vez. En cualquier caso el movimiento general de una pata cuando la plataforma se est\'a moviendo para adelante es el que se muestra en la Fig. 10, donde la pata en primer lugar es levantada del piso, luego avanza y vuelve a bajar. Este proceso en el caso del *tripod gait* se realiza moviendo 3 patas en simult\'aneo, 2 de un lado del robot y otra del lado contrario, mientras que mantiene 3 patas apoyadas sobre la superficie (*support phase*), alternando dicho proceso.

El problema para este tipo de *gaits* que vienen de fabrica en los robots, es que no se pueden adaptar a las irregularidades de los terrenos a atravesar, pues no prev\'en los posibles obst\'aculos ó colisiones de una pata como se muestra en la Fig. 11. Esto se debe a que dichos *gaits* comerciales est\'an pensados para terrenos planos sin ning\'un obst\'aculo y donde generalmente el usuario manipula al hex\'apodo en tiempo real.

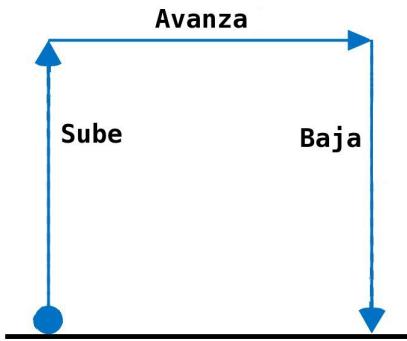


Figura 10: Gait est\'andar (comercial).

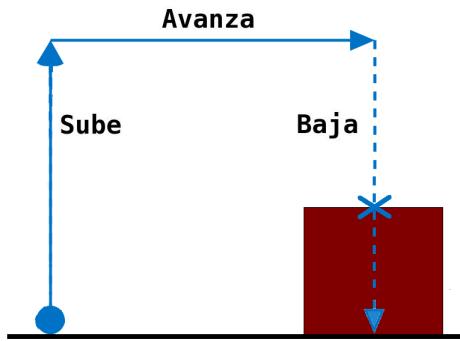


Figura 11: Planteo del problema: detectar colisiones.

Este trabajo propone un enfoque minimalista sin utilizaci\'on de sensores adicionales para detectar posibles colisiones de cualquiera de las patas del robot que le permitan adaptar su movimiento a distintos terrenos en forma aut\'onoma que en principio pueden ser desconocidos para el robot.

Por \'ultimo, en la Fig. 12 se muestra como una pata, ante la detecci\'on de una posible colisi\'on en bajada, frena su movimiento y contin\'ua con el siguiente ciclo. Al mismo tiempo muestra diferentes

posibilidades de movimiento cuando el terreno es irregular. En este caso la altura de la pata respecto del cuerpo no es fija y por lo tanto el *gait* tiene que ser capaz de adaptarse a dicha condición.

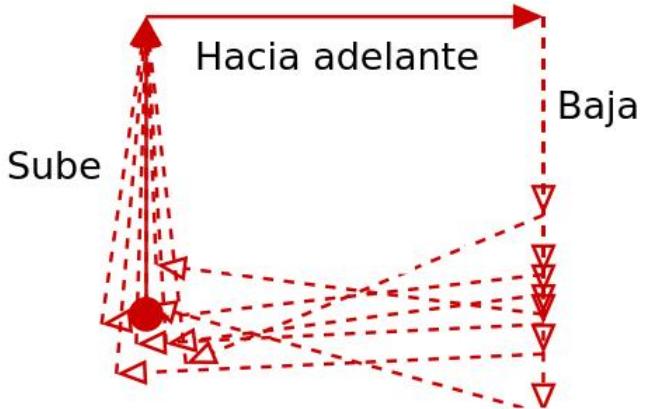


Figura 12: Gait adaptativo.

### 3. Desarrollo experimental

#### 3.1. Características de los servomotores Dynamixel AX-12

La plataforma usada es un PhantomX AX Metal Hexapod Mark III de la compañía Trossen Robotics. Este posee 6 patas con 3 servomotores en cada una de ellas teniendo un total de 18 servomotores de la marca Dynamixel, modelo AX-12. Éstos son alimentados mediante una fuente de 12V. A lo largo de los diferentes experimentos se utilizaron éstos en sus dos modos de operación disponibles.

El primero de ellos es el modo rueda (*Wheel mode*), teniendo un rango de movimiento sin restricciones como un motor de corriente continua, en el cual se le asigna a los motores comandos de velocidad, es decir se asigna una velocidad angular. El control interno que tiene el servomotor para verificar tiempo a tiempo la velocidad asignada que se encuentra en ese instante, no es provista por el fabricante en el *datasheet* [4]. Mediante un estudio se pudo demostrar que posee algún tipo de control interno en la velocidad.

El segundo modo utilizado es el modo articular (*Joint mode*). Dicho modo tiene una limitación de movimiento, pudiendo hacer sólo  $300^\circ$ . El servomotor posee un sensor de posición, que es un potenciómetro lineal de  $10\text{ k}\Omega$  que se muestra en las Figs. 13 y 14, que da lecturas de ángulos en forma absoluta, medidas en *ticks*. Se hace notar que dicho sensor de posición puede ser utilizado en ambos modos de operación. Éste tiene una resolución de 1024 posiciones pues la salida del microcontrolador interno Atmega8 que posee el servomotor tiene una señal de salida de 10 bits ( $2^{10} = 1024$ ), es decir 1024 *ticks*, pudiendo asignar cualquiera de estas sin importar la posición inicial. Es decir, 1 tick equivale a  $0.29^\circ$  ó  $5,1.10^{-3}\text{rad}$ . Se hace notar que en el *datasheet* figura una resolución errónea de  $0.35^\circ$  que sale de considerar que el servomotor puede girar en un rango de  $360^\circ$ , que es imposible

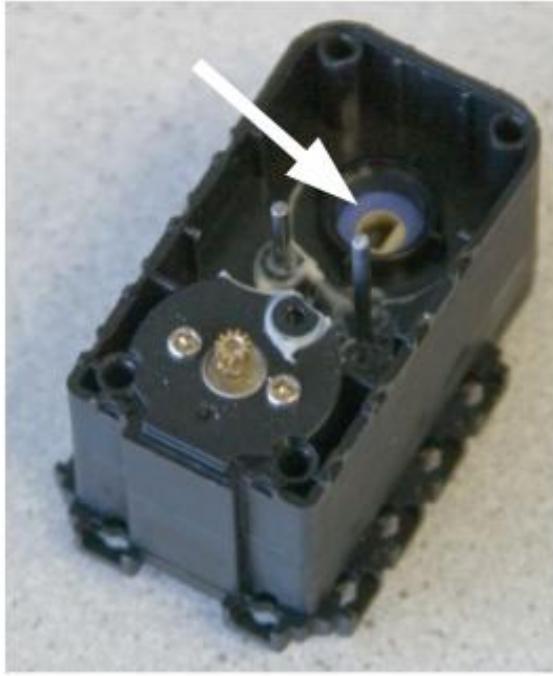


Figura 13: Potenciómetro (sensor de posición) señalado con una flecha.



Figura 14: Potenciómetro (sensor de posición) de color azul.

pues la posición 1024 que es la última está asociada al ángulo de  $300^\circ$  de giro, como se muestra en la Fig.15. Lo anterior fue verificado en la literatura y coincide con [10].

En general, el modo *Joint* se implementa interpolando la posición inicial con la posición final deseada dando el tiempo en el cual uno quiere que tarde ese movimiento. Esto se realiza de esta manera, dando un tiempo de interpolación, ya que si sólo se le asigna una posición final, el servomotor se posicionara en dicho ángulo, dándole tensión continua al servomotor, es decir velocidad máxima, hasta que determine que se encuentra en la posición deseada. Los problemas que conlleva esto, además de ser brusco, es que puede dañar al servomotor por el momento de inercia generado y el frenado abrupto.

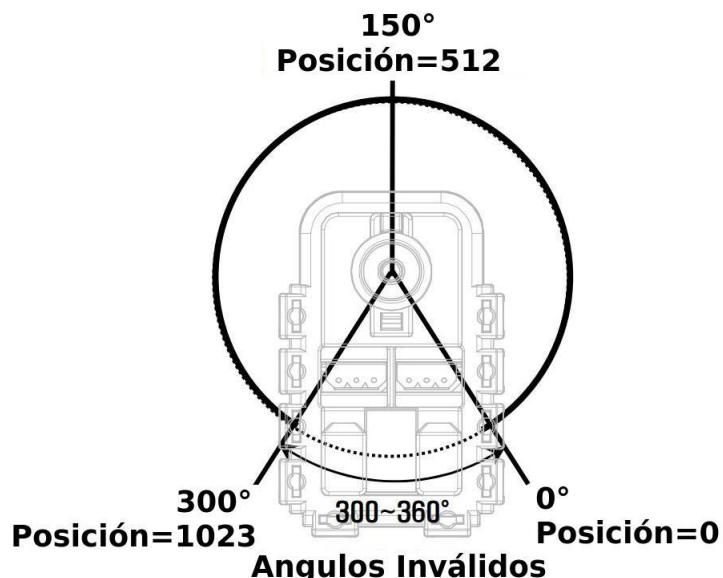


Figura 15: Configuraciones angulares posibles del servomotor Dynamixel AX-12

El servomotor cuenta en el modo *joint* con un control proporcional  $P$  en la posición, como muestra la Ec. 5.

$$Vel = K(X_{goal} - X_{real}) \quad (5)$$

La interpolación lineal genera una lista de posiciones objetivo intermedias entre una posición inicial y una final, con un paso temporal fijo de 16ms por servomotor, que es el tiempo que tarda el microcontrolador interno del servomotor en mandar y recibir la respuesta de un comando.

El controlador  $P$ , tiempo a tiempo toma las mediciones actuales y las compara con las posiciones objetivos de la interpolación, corrigiendo la velocidad. Iterando esto, se logra llegar a la posición final en el tiempo dado. En la Fig. 16 se puede ver una interpolación lineal simulada, en la cual se muestra la posición inicial y final asignadas en función del tiempo demandado para realizar dicho movimiento. En esta imagen se observa que el ancho temporal de cada paso es de 16ms, que se debe a la demora propia que tiene el microcontrolador del servomotor, entre que manda un comando de lectura y obtiene respuesta, como ya se mencionó. El paso espacial de la interpolación, medido en ticks, depende del tiempo de interpolación deseado y la distancia relativa entre las posiciones inicial y final.

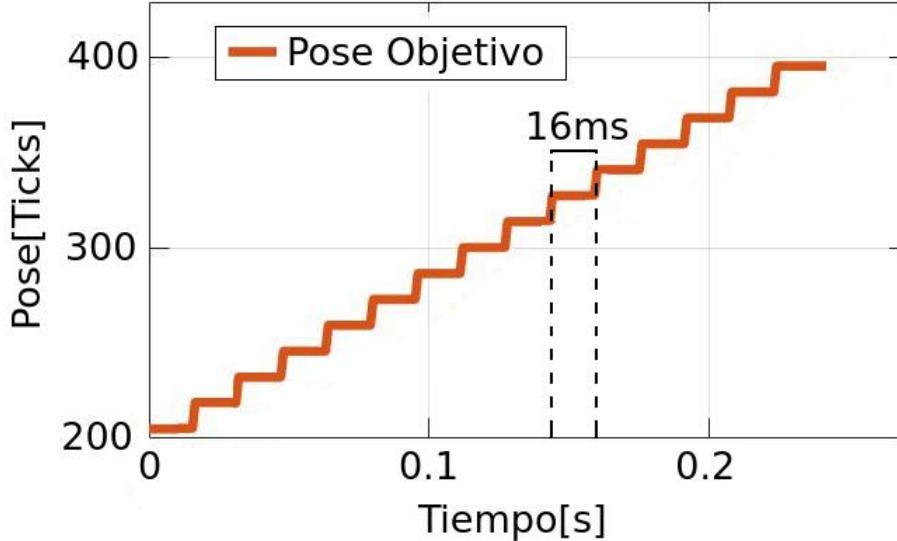


Figura 16: Simulación de una interpolación lineal entre una posición inicial y final.

Los comandos de lectura y de escritura del servomotor, son enviados en lenguaje C++ mediante computadora de escritorio a través del puerto serial hacia la placa ArbotiX, la cual se encuentra en el cuerpo del Hexapodo. A través de dicho puerto se asignaron los ángulos de cada articulación para una pata:  $\theta_{Coxa}$ ,  $\theta_{Femur}$  y  $\theta_{Tibia}$ , para el caso de la cinematica directa o posiciones finales para el caso de la cinematica inversa.

## 3.2. Cinemática

Como se explicó en la introducción, la mayor parte de los experimentos realizados, fueron sobre una de las patas del robot y no sobre el cuerpo entero. Esto se debe a la simetría del problema. Para los casos de la utilización del modo articular, lo que se hizo fue asignarles a cada servomotor de una misma pata, diferentes posiciones angulares para producir una posición final total.

### 3.2.1. Cinemática Directa

En el caso de la cinemática directa, se fueron asignado diferentes posiciones angulares, midiendo luego la posición final del efecto, es decir de la *foottip*. Esto se hizo para el fin de determinar la validez del MCD, poder obtener confiabilidad en nuestra simulación y optimizar los parámetros del mismo. Estas mediciones se hicieron de forma manual, con la utilización de un plano cuadriculado donde se apoyó la plataforma y con una regla nivel. Se asignó un error de medio centímetro, por la forma de medir y por la propia precisión del instrumento.

### 3.2.2. Cinemática Inversa

Para verificar el MCI, se utilizó la cinemática directa. Dadas una posición y orientación deseadas de la punta final de una pata, se utilizó el modelo de la cinemática inversa para obtener las coordenadas angulares de cada motor. A continuación, se utilizó la simulación de la cinemática directa optimizada, que se explica en la sección de resultados, para calcular las coordenadas finales de dicha pata. De esta manera si las coordenadas deseadas coinciden con las coordenadas finales, el modelo cinemático inverso es consistente con el modelo cinemático directo.

## 3.3. Caracterización y simulación de un servomotor

Para la caracterización del propio servomotor se utilizó el modo rueda, en el cual el servomotor se comporta como un motor de corriente continua convencional. Se generó un programa de C++ para generar una posición inicial y final, a partir de comandos de velocidad. El movimiento se basa en ir de una posición a la otra a velocidad constante aumentando ésta en cada nuevo recorrido. En este caso, se tomaron las lecturas de velocidad en todo momento.

El modelo electromecánico fue simulado en MATLAB a partir del modelo de la Ec. 2 y se corroboró dicho modelo con las mediciones adquiridas en el servomotor del robot real. La velocidad del servomotor es generada y controlada por un microcontrolador incorporado como se mencionó antes, a través de la modulación por ancho de pulsos (PWM: *Pulse-width modulation*). Se asumió que el PWM es directamente proporcional a la tensión aplicada al servomotor. En adelante, la velocidad expresada en términos de PWM representará la tensión aplicada al motor directamente.

Para estimar los parámetros del modelo electromecánico se realizó un análisis de la relación entre la velocidad estacionaria medida del motor y el PWM asignado. Mediante un programa escrito en C++, utilizando la librería de los servomotores, se asignaron una posición inicial y final, las cuales se recorrían a velocidades constantes diferentes, para reproducir las condiciones del caso estacionario, adquiriendo las mediciones de velocidad y tensión en cada recorrido. Además se tomaron velocidades promedios para cada tensión dada. Por último, se simuló la velocidad del servomotor mediante el método de Euler utilizando el modelo electromecánico, para comparar las mediciones reales sobre el servomotor.

### 3.4. Dinámica de una pata completa

Finalmente, se utilizó a la plataforma para que detecte colisiones. En este caso se montó de forma tal que quedaran las patas en el aire, con el cuerpo elevado de la mesa donde se encontraba, de forma estática. Luego se generó una trayectoria, desde una posición inicial hasta una final en un determinado tiempo, en la cual se colocó algún objeto/obstrucción en el medio de la trayectoria. En toda esta sección de experimentos se utilizó el modo *joint*. A su vez, no sólo se tomaron posiciones (posición angular) tiempo a tiempo, si no que se pedían las posiciones objetivo, provistas por la interpolación. Con la diferencia entre estos dos valores, tanto los del robot real como los simulados, se pudieron obtener los errores en los cuales uno puede asegurar que no hay colisiones. Cuando estos errores, los cuales la plataforma calcula tiempo a tiempo, superan cierto umbral asignado, se toma el criterio que es debido a una colisión, pudiendo así mediante el código en C++ frenar el movimiento y continuar con el siguiente ciclo del *gait*.

Por último, se probó el modelo dinámico completo para la plataforma en su totalidad. Para ello se utilizó una Odroid XU4 (computadora embebida), que se muestra en la Fig. 17 la cual tiene cargado los patrones de movimientos, desarrollados por [11]. Dicho ordenador permitió un mayor proceso de cómputo y la posibilidad de utilizar el sistema operativo ROS (*Robot Operating System*), que es un framework de programación de robots. Los conocimientos sobre el uso de este sistema operativo, fueron adquiridos en la materia optativa del DC, Introducción a la Robótica Móvil, cursada por ambos autores de este trabajo. ROS cuenta con bibliotecas (funcionalidades de código) con los patrones de movimientos, donde se le agregaron los umbrales recién explicados.

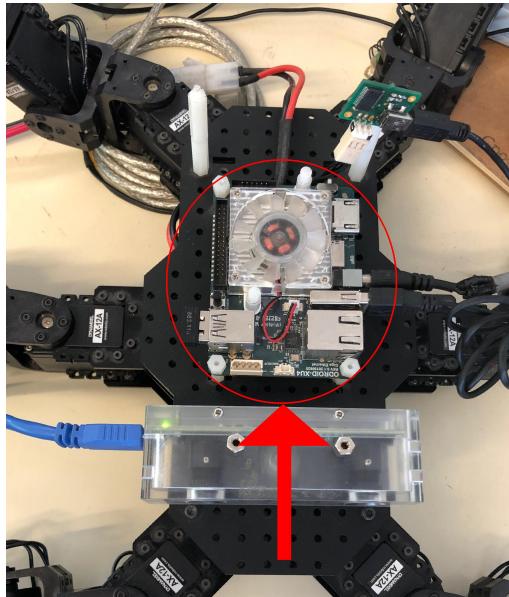


Figura 17: Odroid XU4 montada en el hexápedo. 8.5cm x 5.5 cm.

Para la verificación completa del sistema, se utilizaron diferentes terrenos en el cual, luego de haber implementado los umbrales, la plataforma atravezó satisfactoriamente. Se acompañan [videos](#) que muestran el funcionamiento de la plataforma con diferentes umbrales, tanto subestimados, sobreestimados y óptimo.

## 4. Resultados y discusiones

En esta sección se muestran los resultados obtenidos. En primer lugar, la validación del MCD y MCI, el estudio de la sensibilidad del modelo ante cambios en los parámetros de D-H. Se caracteriza el espacio de trabajo de una pata individual, pudiéndose extender luego al cuerpo entero que permite la planificación de trayectorias seguras (libre de colisiones). Se realizó la caracterización y simulación de un servomotor, verificando el modelo electromecánico elegido. Por último, se muestran los resultados de la simulación completa de una pata y se lo compara con las mediciones reales. Se hace un estudio en los errores de las posiciones esperadas y medidas, pudiendo con ello extraer la información necesaria para establecer umbrales de colisión.

### 4.1. Cinemática

El primer paso en el estudio de la cinemática del hexápedo es el estudio sobre una pata individual vista desde el centro de masa del robot. Esto se debe a que por tener 6 patas idénticas, se facilita luego la descripción global. La cinemática de una pata es la descripción del movimiento de las articulaciones del mismo, sin tener en cuenta las fuerzas o momentos que intervienen. Los métodos cinemáticos se utilizan para poder describir cualquier trayectoria deseada.

#### 4.1.1. Cinemática Directa

Para verificar la validez del MCD del robot se compararon múltiples posiciones finales del extremo de una pata del robot real con una simulación implementada en MATLAB.

Para realizar dicha simulación, se midieron los parámetros del modelo, que se muestran en la Fig. 3 de la introducción teórica. Se realizaron experimentos de simples movimientos de una pata donde se midieron las posiciones finales que fueron comparadas con las simulaciones con las mismas condiciones que el experimento real. La simulación fue realizada en primer lugar para 1 sola pata como se muestra en la Fig. 18 y luego se la extendió para el robot completo como se ve en la Fig. 19, para verificar la simetría del modelo y la validez de los sistemas de referencia elegidos de acuerdo a la convención de D-H.

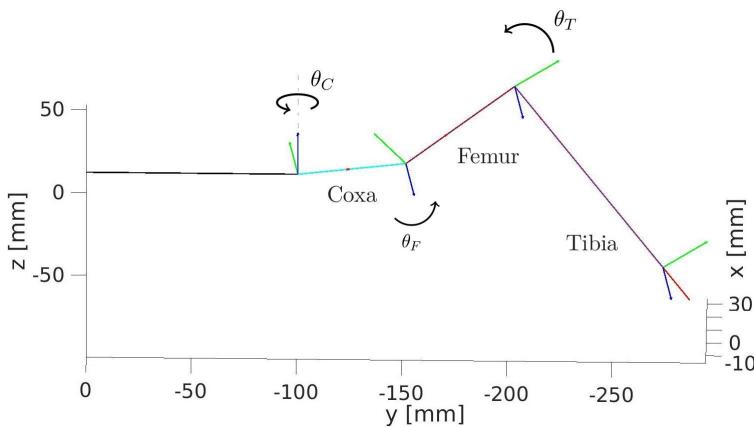


Figura 18: Simulación de 1 pata.

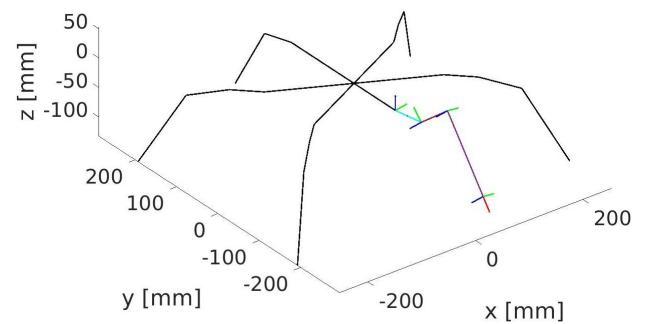


Figura 19: Simulación del Robot Completo.

Una vez obtenidos los primeros resultados, se observó que la diferencia en las mediciones, con respecto a las simulaciones, eran mayores a la incertezza considerada. Es por ello que se optimizaron los parámetros de D-H, para reducir el error entre la simulación y las mediciones con el robot real.

En la siguiente tabla se pueden observar los resultados de las mediciones y las simulaciones antes y después de haber optimizado los parámetros. Se muestran 5 puntos aleatorios de los 54 medidos.

$\theta_c$	$\theta_F$	$\theta_T$	$\ X_s, Y_s\ $ [mm]	$\ X_{Op}, Y_{Op}\ $ [mm]	$\ Z_s\ $ [mm]	$\ Z_{Op}\ $ [mm]
$25^\circ$	$0^\circ$	$75^\circ$	10	0	10	4
$0^\circ$	$0^\circ$	$75^\circ$	11	1	10	4
$-40^\circ$	$10^\circ$	$40^\circ$	3	2	18	1
$-20^\circ$	$90^\circ$	$-60^\circ$	4	8	1	5
$20^\circ$	$60^\circ$	$70^\circ$	20	1	4	1

Donde  $\|X_s, Y_s\|$  es el error de la norma en el plano  $XY$  entre la medición y la simulación. El plano  $XY$  es el que se muestra en la Fig. 4 en la sección de introducción teórica, es decir el plano paralelo a

la plataforma. El error  $\|X_{Op}, Y_{Op}\|$  es el mismo error con los parámetros optimizados. Análogamente se muestran los errores en  $Z$ .

Como se puede observar en la tabla, el error se redujo para la mayoría de las mediciones. Se sumaron los errores de las 54 mediciones con la simulación optimizada para las 3 coordenadas, dando un error total de 679mm representando una tercera parte del error total no optimizado. Los parámetros hallados a partir de la optimización son:  $\theta_F^{off} = -0.19^\circ$ ,  $\theta_T^{off} = -0.73^\circ$ ,  $a_C = 50.53\text{mm}$ ,  $a_F = 68.28\text{mm}$ ,  $a_T = 129.98\text{mm}$ . Como se mencionó anteriormente, al tener todas las patas iguales, estos parámetros pueden ser utilizados para cada una de ellas. Se hace notar que estos parámetros optimizados no poseen incertezas pues no son mediciones reales y son los valores que se utilizarán de forma exacta en las futuras simulaciones.

#### 4.1.2. Sensibilidad del modelo con los parámetros de D-H.

Utilizando los parámetros optimizados, se realizó un análisis de la sensibilidad de los mismos para evaluar su influencia en la validez del modelo. Para simular una medición imprecisa de alguno de los parámetros, se agregaron distintos niveles de error y se analizó la relación entre el error en la posición medida y el porcentaje de error agregado. Se analizaron los errores en la coordenada  $Z$  y en la norma, es decir en el plano  $XY$ , por separado.

En la Fig. 20 se observa un gráfico del tipo boxplot de los errores en la coordenada  $Z$  en función del porcentaje de error agregado a la medición de los parámetros  $\theta_F^{off}$  y  $\theta_T^{off}$ . Los errores son mayores para mediciones imprecisas de  $\theta_T^{off}$  que para  $\theta_F^{off}$ , obteniéndose errores en  $Z$  de hasta 20mm para un 21% de error agregado. Para los mismos parámetros se consideró el error en el plano que se muestra en la Fig. 21 observándose que la dispersión de los errores es mayor para  $\theta_T^{off}$  que para  $\theta_F^{off}$ . De ambos gráficos se desprende que resulta de mayor importancia medir de forma precisa el parámetro  $\theta_T^{off}$  y que por lo tanto la implementación del modelo cinemático resulta más sensible a dicho parámetro.

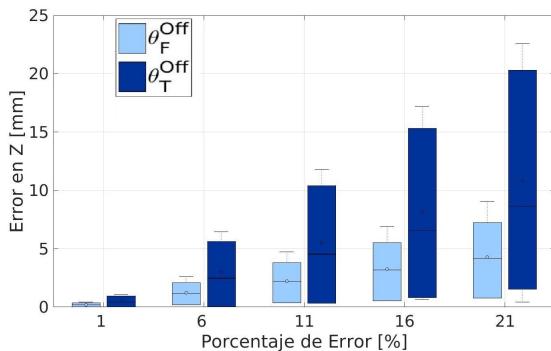


Figura 20: Error en la coordenada  $Z$  para los parámetros  $\theta_F^{off}$  y  $\theta_T^{off}$ .

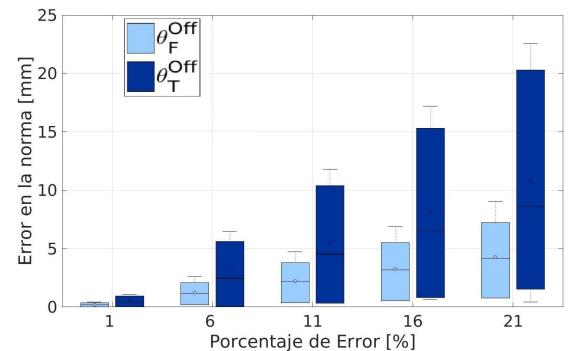


Figura 21: Error en la norma para los parámetros  $\theta_F^{off}$  y  $\theta_T^{off}$ .

En la Fig. 22 se muestra el boxplot de los errores en la coordenada  $Z$  en función del porcentaje

de error agregado a las mediciones de los parámetros  $a_C$ ,  $a_F$  y  $a_T$ . Se observa que el parámetro  $a_C$  no presenta errores a pesar de que el error agregado aumenta debido a la simetría de translación en  $Z$ , dado que  $a_C$  está en un plano definido que no varía. El parámetro  $a_T$  presenta mayores errores que el parámetros  $a_F$  por lo que el modelo resulta más sensible a dicho parámetro.

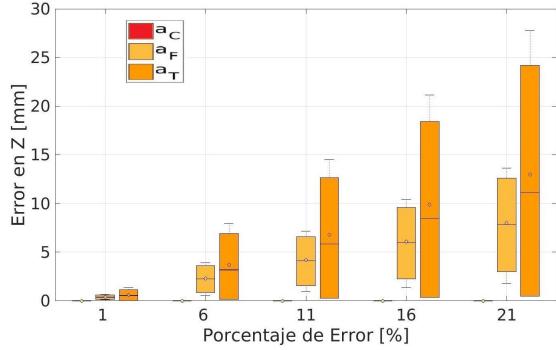


Figura 22: Error en la coordenada  $z$  para los parámetros  $a_C, a_F$  y  $a_T$ .

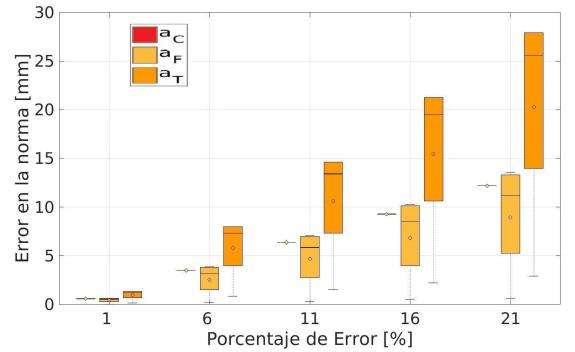


Figura 23: Error en la norma para los parámetros  $a_C, a_F$  y  $a_T$ .

Del boxplot de la Figura 23 se desprende que el parámetro  $a_T$  es el que mayor error en la norma induce debido a ser el parámetro de mayor longitud. Además se observa que el parámetro  $a_C$  no presenta dispersión para cada porcentaje de error agregado. Esto se debe a la simetría en el plano.

#### 4.1.3. Cinemática Inversa

Para verificar el MCI, se utilizó la cinemática directa. Dadas una posición y orientación deseadas de la punta final de una pata, se utilizó el modelo de la cinemática inversa para obtener las coordenadas angulares de cada motor. A continuación, se utilizó la simulación de la cinemática directa optimizada en la sección anterior para calcular las coordenadas finales de dicha pata. De esta manera si las coordenadas deseadas coinciden con las coordenadas finales, el modelo cinemático inverso es consistente con el modelo cinemático directo. Se verificó que se llegaba a la posición deseada con una incertidumbre del orden del error experimental considerado de medio centímetro. Se hace notar que la validez de este modelo es de carácter implementativo, funcional.

Uno de los inconvenientes que tiene este tipo de robot fue la no unicidad de la solución. Estudiando esto se vio que había dos casos particulares, uno de los cuales no era de nuestro interés, ya que arrojaba soluciones fuera del espacio de trabajo. Se solucionó agregando dichos casos al modelo como casos particulares.

## 4.2. Espacio de Trabajo - *Workspace*

El espacio de trabajo de un robot está definido como el conjunto de posiciones que pueden ser alcanzadas por la punta de la pata. La determinación del espacio de trabajo es fundamental para realizar la planificación de trayectorias del robot y evitar posibles obstáculos/collisiones.

Se realizó un barrido de los posibles ángulos que puede tomar la coxa para una sola pata, asignándoselos a los motores del robot real, sin considerar posibles colisiones obteniéndose  $\theta_{Coxa} \in [-90^\circ, 90^\circ]$ . Lo mismo se realizó para el femur y para la tibia, resultando  $\theta_{Femur} \in [-105^\circ, 105^\circ]$  y  $\theta_{Tibia} \in [-120^\circ, 75^\circ]$ . Se asociaron incertezas de la mínima precisión del sensor de posición que es de  $0.29^\circ$ . Los valores mostrados en grados fueron convertidos a partir de valores en *ticks* que son los implementados en la plataforma real. Es por ello que se muestran valores redondeados.

Utilizando la simulación de la cinemática directa se variaron todos los ángulos posibles de cada articulación guardando la posición final alcanzada. En la Fig. 24 se muestra el espacio de trabajo de una pata del robot en el plano XY.

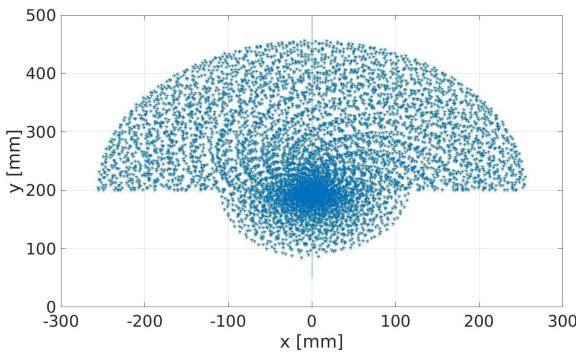


Figura 24: Espacio de trabajo: Plano XY.

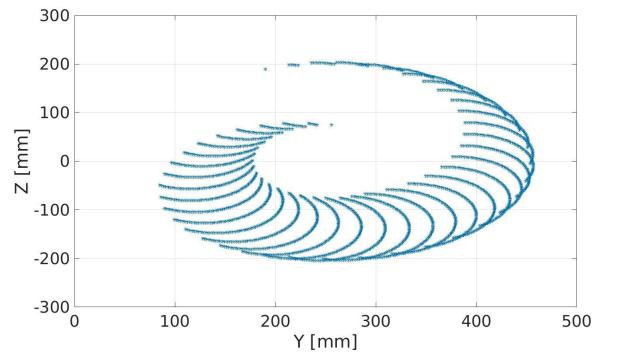


Figura 25: Espacio de trabajo: Plano YZ.

En la Fig. 25 se muestra el espacio de trabajo en el plano YZ. El origen de coordenadas de ambas figuras está asociado al origen del sistema de referencia del robot situado en el centro de su cuerpo.

### 4.3. Resultados dinámica completa de una pata

El primer paso para el estudio de la detección de las colisiones, es conocer la dinámica del hexá-podo. A partir de ésta se puede simular el movimiento de una pata y determinar la posición en cada punto de la trayectoria que realiza cada pata en un *gait*. Nuevamente por la simetría del problema, se estudia una pata individual vista desde el centro de masa del robot, pudiendo luego realizar la descripción completa.

Las simulaciones son comparadas con las mediciones sobre la plataforma real, pudiéndose determinar el error cometido por la simulación. De esta manera se pueden asignar umbrales, en los cuales, si el error es mayor a estos, el patrón de movimiento de dicha pata acaba, para continuar con el próximo, como se explicó en la introducción teórica, en la subsección 2.5 de *gaits*.

Los resultados se ordenan de la siguiente manera. En primer lugar se caracteriza los servomotores utilizados, mostrando la validez del modelo electromecánico y se implementa luego dicha información, para la descripción de los torques que experimentan cada articulación de la pata. Luego se considera a una pata como una cadena de cuerpos rígidos y se describe con el formalismo de Lagrange como se

explicó en la introducción teórica. Con la combinación de estos dos modelos se obtiene la dinámica completa para una pata.

#### 4.3.1. Caracterización y simulación de un servomotor.

Se realizó la identificación de un servomotor del hexápodo para estudiar la validez del modelo electromecánico. Se hace notar que los presentes resultados sólo tienen en cuenta el modelo electromecánico y no se considera todavía el formalismo de Lagrange en cuanto cadena de cuerpos rígidos. Para el caso estacionario se tenía que la velocidad venía dada por la siguiente expresión lineal:

$$\dot{\theta}_{est} = \alpha_1 u + \alpha_2 sgn(\dot{\theta}) \quad (6)$$

En la Fig. 26 se observan los resultados experimentales obtenidos y el ajuste realizado con la Ec. 6. Se observó que la relación lineal se cumple en todo el rango de tensiones aplicadas al motor. De esta forma se estimaron dos constantes para velocidades positivas y dos constantes para velocidades negativas. Los valores promedios obtenidos son:  $\alpha_1^{prom} = 0.53V^{-1}s^{-1}$ ,  $\alpha_2^{prom} = 0.02s^{-1}$ . En el gráfico de la Fig. 26 se observa que existe una rango de voltajes para los cuales la velocidad es nula. Esta región se llama zona muerta y representa los valores de tensión en la cual el servomotor no logra romper la fricción estática para comenzar el movimiento. El ajuste del gráfico de la Fig. 27 fue obtenido integrando la Ec. 1 mediante el método de Euler y se observa que la simulación se ajusta a los datos medidos. De esta forma se pudo confirmar la validez del modelo electromecánico elegido.

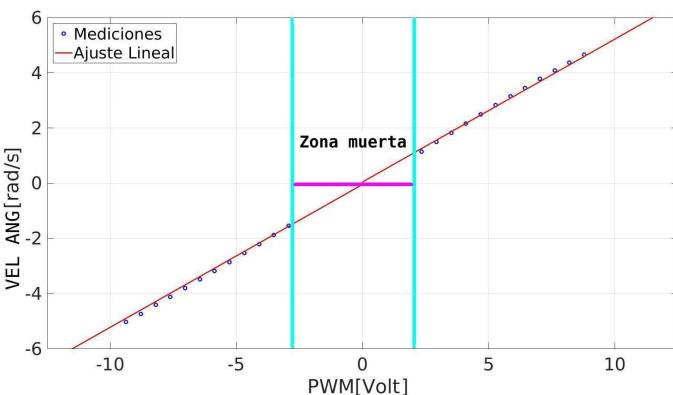


Figura 26: Relación lineal entre la tensión aplicada al motor y la velocidad del mismo.

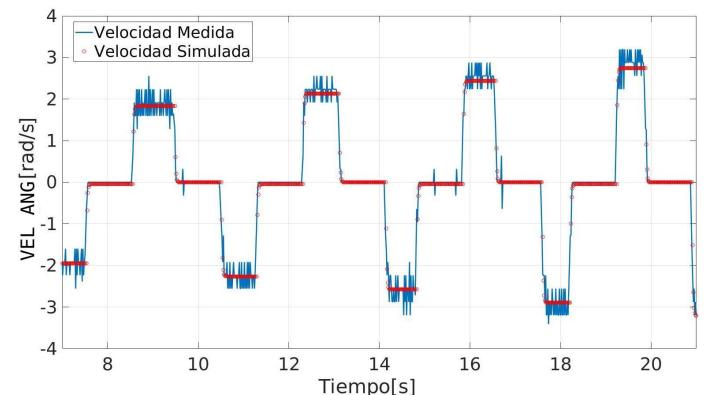


Figura 27: Simulación de la velocidad del motor para el caso estacionario sin torque de gravedad.

Una vez caracterizados los servomotores sin carga adicional, es decir el caso del servomotor aislado sin estar conectado al robot, se tomaron mediciones del servomotor asociado a la coxa de una pata para caracterizarlos con carga. La carga sería en este caso las dos articulaciones conectadas a la coxa que son el femur y la tibia respectivamente. El movimiento que es medido es el de rotación de la coxa con el robot estático.

Como se muestra en el gráfico de la Fig. 28 el servomotor no es simétrico, es decir cuando gira en sentido antihorario (velocidades angulares positivas) tiene una cierta relación con la tensión aplicada que es distinta para el de la rotación horaria para cada caso en particular (con y sin carga). Además se observa que la pendiente en el giro horario para el caso sin carga es menor que los casos con carga mientras que para el giro horario pasa lo contrario. La importancia de esta experiencia fue el corroborar que el comportamiento lineal se mantiene en las diferentes configuraciones de una pata. Más adelante se realiza un análisis de comparación entre el caso de la pata contraída y estirada que muestra este caso en mayor profundidad considerando el modelo dinámico completo.

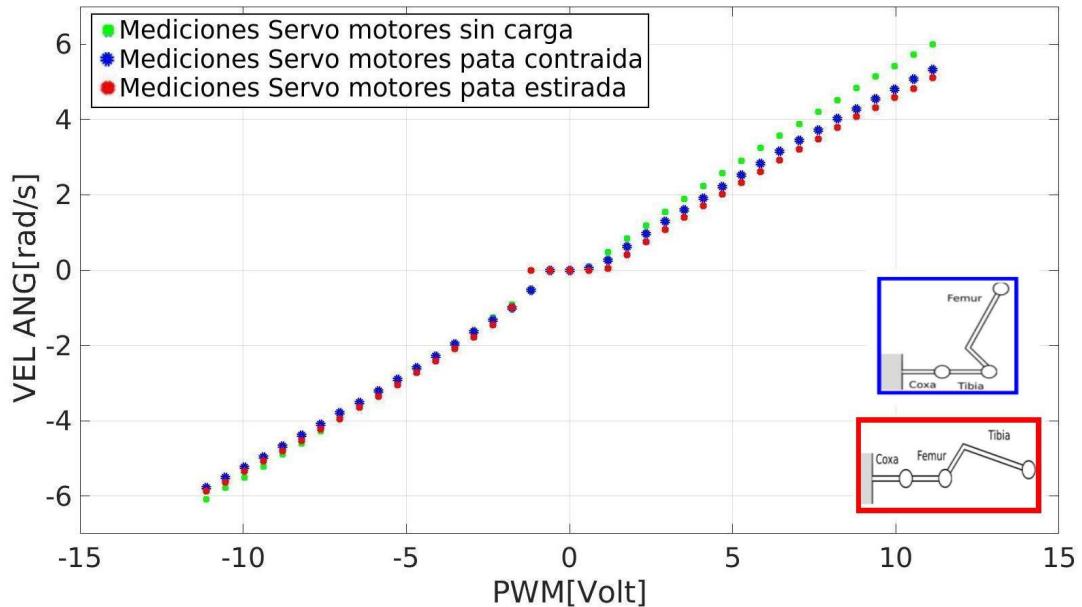


Figura 28: Relación lineal entre la tensión aplicada al motor y la velocidad del mismo para diferentes casos.

En la Fig. 29 se muestra la velocidad en función del tiempo para el caso de una pierna completa. Se puede observar que la forma funcional de la simulación no concuerda con las mediciones reales donde las velocidades constantes son máximas y mínimas. Luego de repetir para diferentes casos esta experiencia se concluyó que el modo *Wheel mode* posee internamente un control interno, no mencionado en el *datasheet*, el cual permite al servomotor establecer las velocidades consignas de manera casi inmediata. Dicho control en velocidades no estaba considerado en la simulación, produciendo así la diferencia ya dicha.

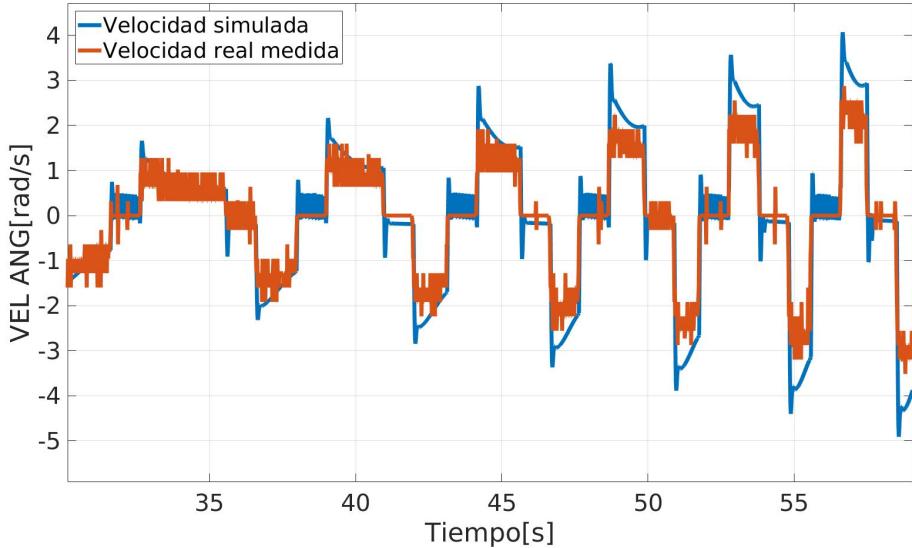


Figura 29: Simulación de la velocidad del motor para el caso con carga y torque de gravedad.

#### 4.3.2. Simulación completa de una pata

En el gráfico de las Figs. 30 y 31 se muestran las mediciones de la posición (puntos azules) de una articulación en función del tiempo para un simple movimiento de 1 y 0.5 segundos de duración respectivamente entre una posición inicial y una posición final elegidas en la implementación en C++. En naranja se muestra la pose objetivo simulada que se desea obtener en cada instante de la interpolación. En la curva azul continua se muestra la simulación.

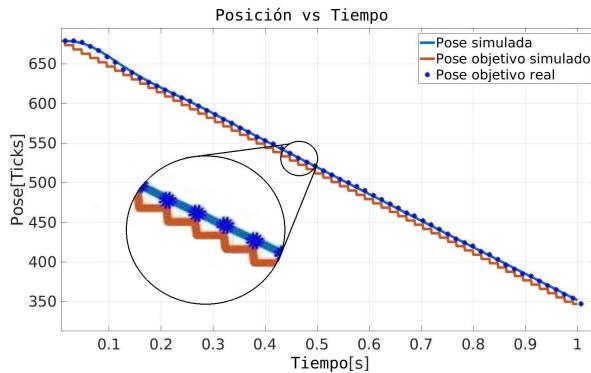


Figura 30: Posición medida en ticks en función del tiempo junto a la simulación de la pose real y de la pose objetivo.  $\Delta t = 1s$ .

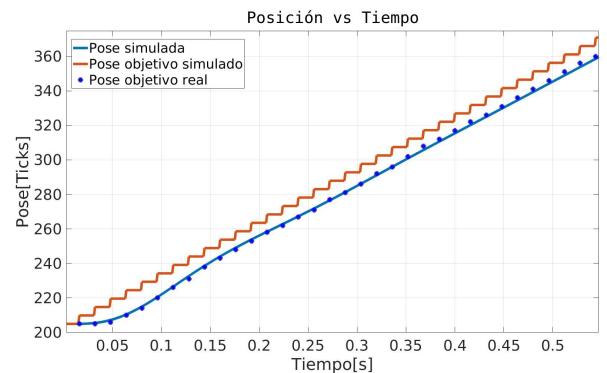


Figura 31: Posición medida en ticks en función del tiempo junto a la simulación de la pose real y de la pose objetivo.  $\Delta t = 0.5s$ .

Se puede observar como en ambos casos la simulación concuerda perfectamente con las mediciones. No se cuantificó la precisión de la simulación ya que lo que es de nuestro interés, es verificar que el modelo dinámico completo reproduce el comportamiento real de una pata. Los datos tienen una diferencia con la interpolación, debido a que el servo no llega a la posición deseada antes de que cambie el objetivo. Este efecto se agrava a medida que la exigencia temporal es menor. Se puede observar como este comportamiento también es reproducido por la simulación, ya que contempla

tanto la interpolación lineal como el controlador proporcional.

Se hace notar que las posiciones son posiciones angulares mostradas en *ticks* como es estándar en la literatura, donde 1 *tick* equivale a  $0.29^\circ$  como se explicó en la descripción experimental.

#### 4.3.3. Análisis de los errores

Una vez probado que la simulación recrea la realidad, se comenzó con el estudio de los errores entre la posición real medida y la posición instantánea deseada por la interpolación. Esto se hizo tanto para los datos reales como para los simulados. En la Fig. 32 se observa el error entre la posición real y la posición objetivo para el caso de una exigencia temporal de 1 segundo.

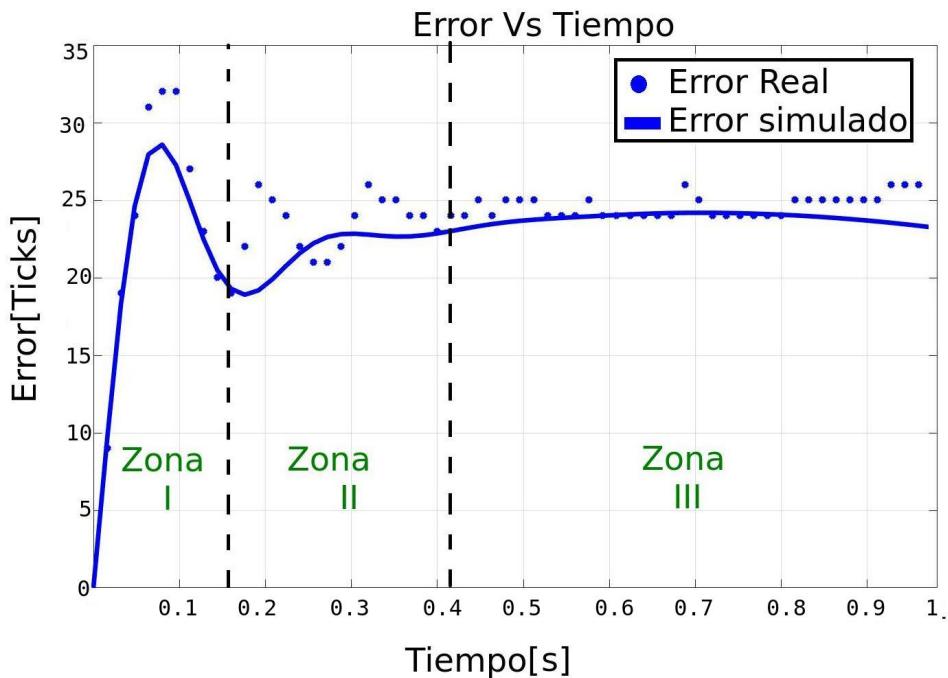


Figura 32: Error de la posición real y objetivo en función del tiempo.

En dicha figura se pueden observar 3 zonas. La **Zona I**, donde se encuentra su pico máximo, describe el inicio del movimiento, en donde el motor rompe la fricción estática y comienza a acelerar. Dicho proceso es algo brusco por el perfil de aceleración que tiene el servomotor. Al suceder esto se genera una gran inercia en la cual la posición real sobrepasa a la posición objetivo y así aumenta su error. Luego de ésta empieza a corregir la posición con el controlador proporcional ya descripto, iniciando la **Zona II**. En esta nueva zona se observa una leve diferencia en la forma funcional entre la simulación y los datos reales. En la simulación se puede ver una oscilación amortiguada crítica, ya que el modelo dinámico utilizado solo considera la inercia de un cuerpo rígido. En cambio los datos reales muestran una oscilación subamortiguada. Esto es debido a oscilaciones mecánicas no consideradas en el modelo, provenientes de los materiales de la propia pata y de la rigidez de los plásticos de las articulaciones. Dicho efecto depende del caso y esto será estudiado más adelante. La **Zona III** consiste en la estabilización del sistema, en el cual llega a una velocidad constante sin vibraciones no deseadas.

Es en esta zona en la cual se toma más adelante el valor promedio y se considera como *offset* del umbral. Por último, se resalta que los errores de la **Zona I** no son considerados en la detección de colisiones. Por este motivo, se asigna un tiempo fijo en el cual el robot empieza a comparar el error con el umbral.

#### 4.3.4. Análisis de la velocidad en los umbrales

Resulta importante determinar de que parámetros depende el *offset* del umbral. Para ello se vio como dependía la velocidad de la propia pata en dichos errores. Se realizaron diferentes mediciones entre una misma posición inicial y final pero a diferente exigencia temporal de llegada, es decir para diferentes velocidades. En las Figs. 33 y 34 se puede observar los errores entre la posición real y objetivo en función del tiempo para diferentes velocidades. En el primer caso, el movimiento era descendente, es decir la posición inicial se encontraba a mayor altura que la posición final, teniendo el torque de gravedad positivo, es decir a favor del movimiento. En cambio en el segundo caso, el movimiento es el inverso, yendo de esta última posición final hacia la posición inicial, es decir teniendo el torque de gravedad en contra del movimiento. Se puede observar como en ambos casos, el error aumenta a medida que aumenta la velocidad. Este efecto se debe a que a mayor velocidad el control pierde gradualmente precisión, no logrando ajustar en el debido tiempo.

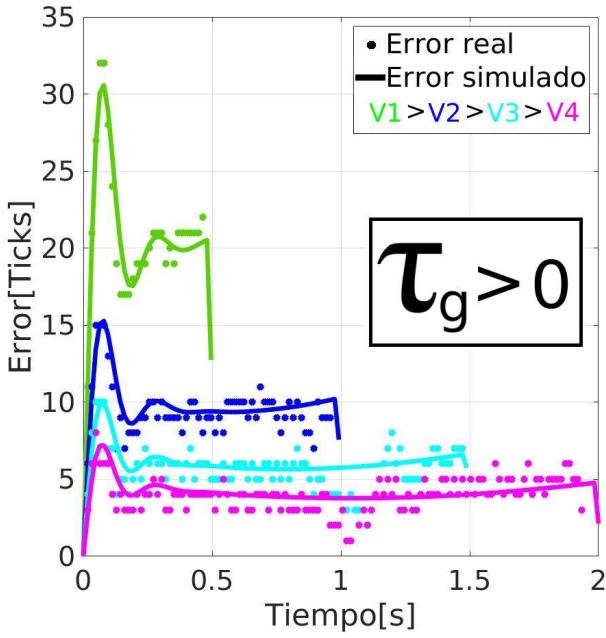


Figura 33: Error de la posición real y objetivo en función del tiempo, con torque de gravedad a favor.

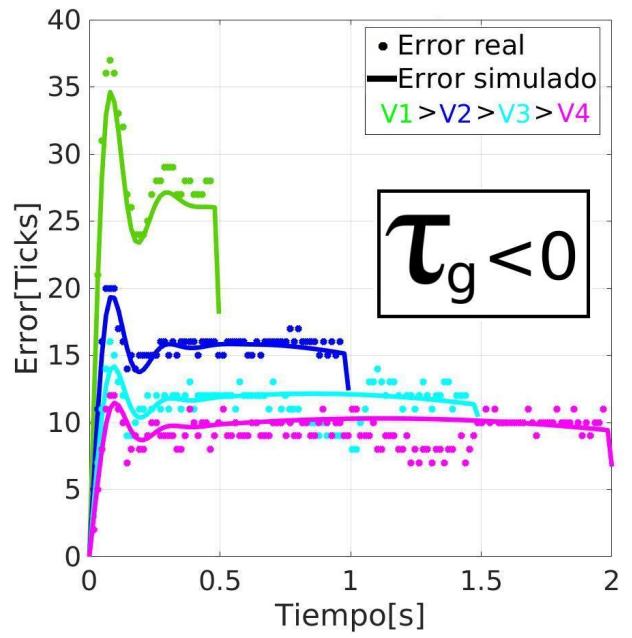


Figura 34: Error de la posición real y objetivo en función del tiempo, con torque de gravedad en contra.

Otra punto importante a observar, es la diferencia en la curvatura en la **Zona III** en ambos gráficos. Dicha diferencia se debe al torque de gravedad, mostrando que la propia simulación contempla dichos casos. Esta torque resulta a favor o en contra dependiendo el sentido de movimiento, modificando

el error. Si no hubiese contribución de la gravedad en el movimiento, no habría curvatura alguna. Dicho comportamiento se puede ver en la sección posterior, donde los errores de coxa no presentan dicha curvatura, por tener un movimiento en el plano perpendicular a la gravedad. Para finalizar se observa que otro efecto de la gravedad es que cuando el movimiento es en contra de esta, los errores son mayores para cada velocidad, en relación a los movimientos a favor de la gravedad.

El análisis en velocidad de los umbrales, tiene como fin implementativo que el propio robot al elegir un patrón de movimiento con una velocidad particular en cada tramo, pueda simular internamente a partir del modelo dinámico completo propuesto y encontrar el *offset* del umbral correcto. A dicho *offset* del umbral se agrega un valor fijo de 5 *ticks* para toda velocidad. Este valor se eligió luego de analizar múltiples casos.

Si el umbral elegido es bajo y alrededor del mismo valor del error que el robot comete normalmente por simple ruido y no por obstáculos, ninguna pata termina de completar si quiera un ciclo. Esto ocurre ya que el error entre cualquier posición intermedia real y la objetivo de la interpolación, es mayor que el umbral elegido. En el video del siguiente [link](#) se muestra este caso donde el robot producto de este umbral bajo pierde estabilidad y se cae parcialmente en la dirección de la pata que encuentra en la *swing phase*.

Si el umbral elegido es alto, el error cometido llega a superar el umbral luego de un cierto tiempo en el que en consecuencia el cuerpo del robot se levanta y aumentando la exigencia de los servomotores, pudiendo en algún caso sobrecargarlos. De esta manera llega a tocar el piso pero no detecta el suelo continuando con el patrón de movimiento, como se muestra en el siguiente video cuyo [link](#) se acompaña debajo. Con este umbral si el robot tuviera colisiones no podría detectarlas correctamente respondiendo de forma inmediata, teniendo que esperar a que el error sea mayor al umbral. Esto conlleva a sobrecargar los servomotores.

Ahora bien, si el umbral elegido es el óptimo, el robot detecta el piso correctamente y ni bien toca la superficie frena su movimiento y continúa con el patrón correctamente como se muestra en el siguiente [link](#). Finalmente se muestra un [video](#) en el que el robot debe atravesar un terreno irregular con el umbral óptimo mostrando que el robot logra adaptar su movimiento, capacidad que no estaba provista por el fabricante.

Links:

- Umbral bajo, del orden del ruido promedio → [www.youtube.com/watch?v=byRyuZdhoEA](http://www.youtube.com/watch?v=byRyuZdhoEA)
- Umbral alto, del orden de errores de colisiones → [www.youtube.com/watch?v=2wELCMmjwNA](http://www.youtube.com/watch?v=2wELCMmjwNA)
- Umbral óptimo → [www.youtube.com/watch?v=yFYMdWken8](http://www.youtube.com/watch?v=yFYMdWken8)
- *Gait* adaptativo → [www.youtube.com/watch?v=HJKMdAtVQf0](http://www.youtube.com/watch?v=HJKMdAtVQf0)

#### 4.3.5. Análisis de los errores en la coxa

Finalmente se analizó la dependencia del momento de inercia en los errores. Para ello se realizaron mediciones de la posición del servomotor de la articulación asociada a la coxa de una pata en dos casos. Una en la cual la pata se encontraba totalmente estirada, la cual poseía el mayor momento de inercia posible y otra en la cual se encontraba contraída intentando tener el menor momento de inercia posible. El caso de estudio en coxa es de interés para analizar las posibles colisiones que pudiera tener en el movimiento de avance del robot durante un *gait*. En las Figs. 35 y 36 se pueden observar los errores entre la posición real y objetivo en función del tiempo, tanto para la simulación y los datos reales, para una misma velocidad para el caso de la pata estirada y contraída respectivamente.

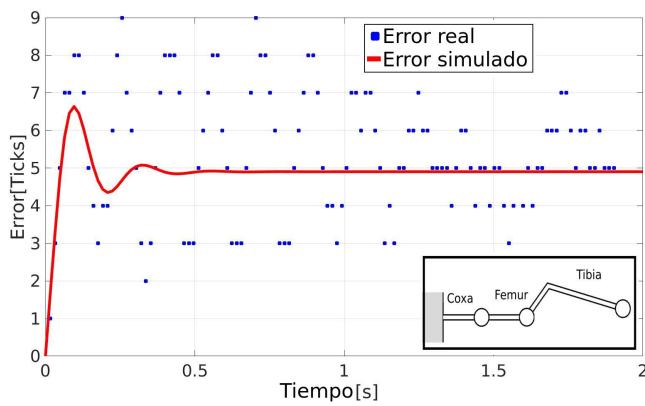


Figura 35: Errores en la coxa - Pata estirada.

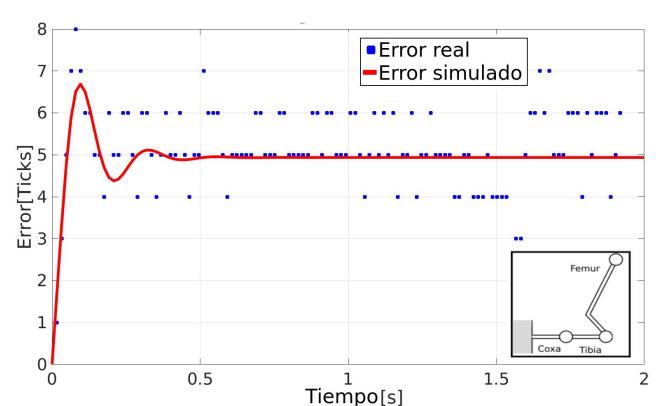


Figura 36: Errores en la coxa - Pata contraída.

Se puede ver que en el caso con el mayor momento de inercia los datos reales presentan en la **Zona II** oscilaciones subamortiguadas, generando mucho ruido. Esto se debe como se explicó antes a que al aumentar el momento de inercia, se generan oscilaciones mecánicas las cuales no son consideradas en nuestro modelo dinámico. A su vez se puede ver como en el caso de pata contraída estas oscilaciones decrecen en amplitud. En este caso en particular, la diferencia del error entre uno y otro no supera los 4 ticks, es decir  $1.16^\circ$ . Cabe resaltar que se eligió una velocidad lo suficientemente baja para mostrar el efecto de las vibraciones de manera visible en los resultados. A velocidades más altas la diferencia en los errores entre los diferentes momentos de inercia aumenta considerablemente. Este efecto hace que nuestra precisión en la detección de colisiones en el plano XY con respecto a la plataforma, sea menos precisa con una dependencia más fuerte al patrón de movimiento y velocidad.

## 5. Conclusiones

### 5.1. Conclusiones Laboratorio 6

Se implementó el modelo de la cinemática directa en el robot real pudiéndose verificar el modelo tanto para una pata como para el robot entero. Se optimizaron los parámetros de D-H, mejorando la

confianza del mismo. Se estudió la influencia de los parámetros en el modelo, observándose que  $\theta_T^{off}$  y  $a_T$  son los que más error inducen por lo que el modelo resulta más sensible a la medición precisa de dichos parámetros.

Por otro lado se implementó y validó el modelo de la cinemática inversa en el robot real. Se determinó el espacio de trabajo, fundamental para realizar la planificación de trayectorias del robot y evitar posibles obstáculos/collisiones.

Por último, se caracterizó y simuló el servomotor utilizado. Se determinó la relación entre la velocidad y la tensión de alimentación medida en PWM. Se verificó el modelo propuesto de velocidad realizando un ajuste por el método de integración de Euler de las velocidades medidas en el motor.

## 5.2. Conclusiones Laboratorio 7

Se amplió el estudio de la caracterización de los servomotores, analizando su comportamiento con carga, siendo ésta la propia pata. Se observó la linealidad en el comportamiento y se pudo validar el modelo electromecánico para cualquier caso.

Se implementó y simuló la dinámica completa de una pata pudiéndose verificar el modelo propuesto. De esta forma se reprodujo el comportamiento real de una pata. Se estudiaron los errores entre la posición objetivo y la medida/simulada dependiendo el caso. Con esto se determinaron los umbrales a partir del modelo dinámico completo. Se analizó la dependencia de la velocidad en el cálculo de los umbrales, permitiendo la implementación en el robot real. Al elegir un patrón de movimiento con una velocidad particular en cada tramo, el robot puede de esta forma simular internamente a partir del modelo dinámico completo propuesto y encontrar el *offset* del umbral correcto. Se estudiaron los errores en diferentes planos de movimiento, considerando el espacio de trabajo, viendo que por la configuración del sistema, estos son mayores en el plano XY, considerando el movimiento de la coxa.

Se implementaron los umbrales en los *gaits* reales como se mostró en los [videos](#), dándole así la capacidad de atravesar diferentes terrenos, mejorando las prestaciones de los *gaits* comerciales. Se probaron los *gaits* adaptativos en diferentes terrenos difíciles de atravesar, obteniéndose resultados satisfactorios.

## Referencias

- [1] KRISHNAN R. (2001) *Electric Motor Drives: Modeling, Analysis, and Control*. Virginia, Estados Unidos. Prentice Hall.
- [2] MRVA J. (2014) *Design of Motion Primitives for a Hexapod Walking Robot Operating in a Rough Environment*. Czech Technical University. Praga, República Checa.
- [3] PESSACG F. (2015) *Fisica Aplicada a Manipuladores Robóticos*. Buenos Aires, Argentina. UBA, Facultad de Ciencias Exactas y Naturales, Departamento de Física.
- [4] Dynamixel AX-12 Manual. [http://www.trossenrobotics.com/images/productdownloads/AX-12\(English\).pdf](http://www.trossenrobotics.com/images/productdownloads/AX-12(English).pdf)
- [5] M. W. SPONG, S. HUTCHINSON, AND M. VIDYASAGAR (2006) *Robot modeling and control, vol. 3*. Wiley New York.
- [6] F. TEDESCHI AND G. CARBONE (2014) *Design issues for hexapod walking robots, Robotics, vol. 3, no. 2, pp. 181–206*.
- [7] D. BELTER, P. LABECKI, AND P. SKRZYP CZYNSKI (2014) *An exploration-based approach to terrain traversability assessment for a walking robot, in Design issues for hexapod walking robots, Robotics, vol. 3, no. 2, pp. 181–206*.
- [8] M. F. SILVA, J. T. MACHADO, AND A. LOPES (2004) *Fractional order control of a hexapod robot, Nonlinear Dynamics, vol. 38, no. 1-4, pp. 417–433*.
- [9] RICHARD M. MURRAY, ZEXIANG LI AND S. SHANKAR SAstry (1994) *A Mathematical Introduction to Robotic Manipulation*, CRC Press.
- [10] ARNO MENSINK (2008) *Characterization and modeling of a Dynamixel servo*, University of Twente, The Netherlands.
- [11] T. FISCHER (2018) *Adaptive motion gait for a hexapod platform. Developed by with the ATC group of the Czetch Technical University*.