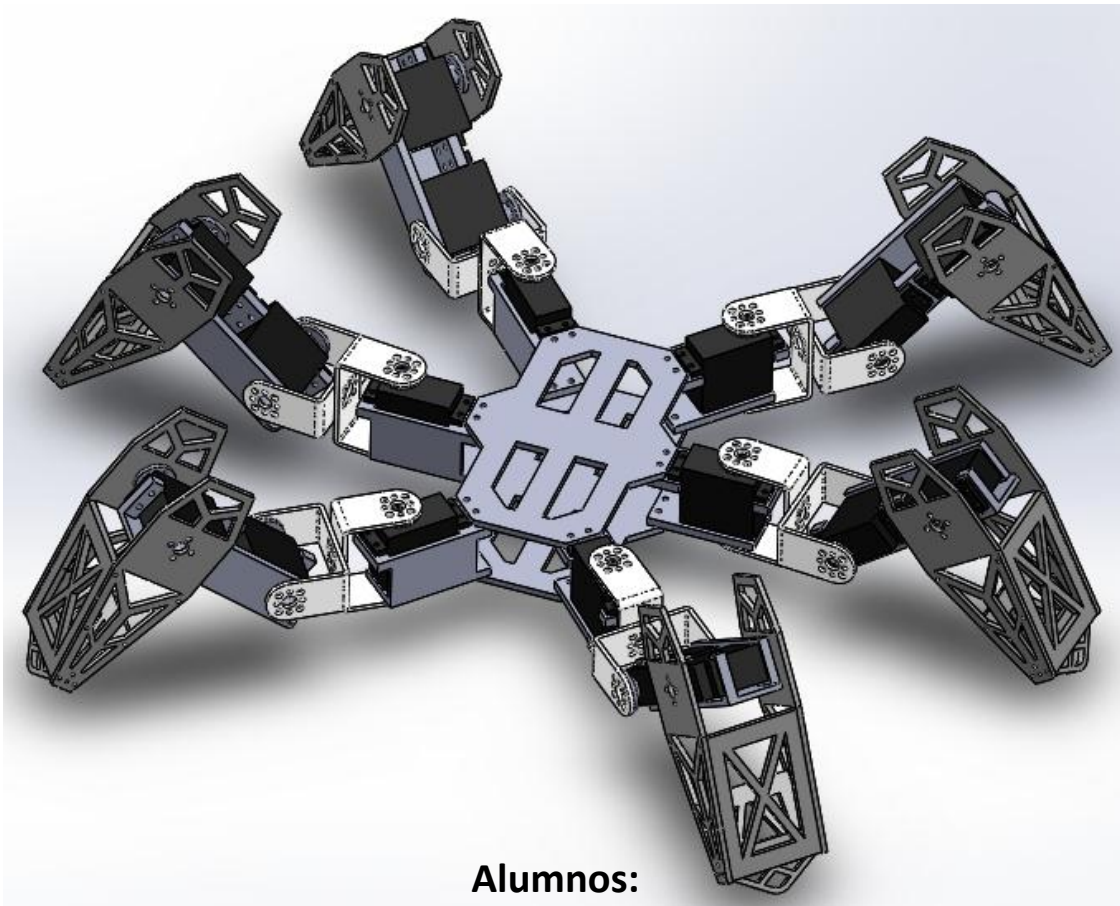


Universidad Nacional de Cuyo

Facultad de Ingeniería

# ROBÓTICA II

Proyecto final 2016: Robot Hexápodo



**Alumnos:**

Sesto, Federico Sebastián      09990

Ullmann, Gustavo      10227

Mendoza, diciembre de 2016

### **Objetivos**

El objetivo de este proyecto es el de experimentar y realizar etapas esenciales en la creación de un robot, como por ejemplo diseñar una estructura lo suficientemente rígida y de peso reducido, desarrollar y simular la dinámica y proponer la realización del montaje de un prototipo. Para la ejecución de estas actividades será necesario aplicar los contenidos teóricos y prácticos estudiados durante el cursado de Robótica 2. Para ello, se desarrollará un robot hexápodo de 18 grados de libertad (3 grados de libertad por cada extremidad). El propósito de este robot móvil es el de exploración de zonas de terreno accidentado o de difícil acceso.

## Resumen

En este trabajo se realiza las etapas de obtención de modelo dinámico de un robot hexápodo de 18 grados de libertad mediante la implementación de los algoritmos computacionales de Newton-Euler y de Euler-Lagrange en MATLAB. Posteriormente se realiza la simulación de estas ecuaciones en el software Simulink, donde se observa el comportamiento a lazo abierto del sistema, a lazo cerrado después de agregarle un controlador PID a cada articulación y los esfuerzos de consigna máximos para determinar el tamaño de actuadores requeridos. Finalmente se proponen los sensores y actuadores necesarios para el funcionamiento del robot.

## **Introducción**

En el siguiente trabajo se buscará determinar la dinámica directa e inversa de un robot móvil terrestre de 6 patas, más precisamente un hexápodo. El mismo cuenta con tres grados de libertad por extremidad, lo que le brinda dieciocho grados de libertad en total al robot. El fin de este robot es el de recorrer y explorar terrenos irregulares.

## Modelo cinemático

El modelo cinemático correspondiente a este robot se encuentra detallado en el proyecto final correspondiente a la cátedra Robótica 1.

## Modelo Dinámico

Para la obtención del modelo dinámico directo se recurre a la formulación de Newton-Euler y de Euler-Lagrange, considerando la pata del hexápodo como un robot serie de tres grados de libertad. Se considera que los eslabones poseen propiedades de masa y longitud distintas entre sí, pero se asume que la masa total de cada eslabón se encuentra situada en el extremo de las mismas. Esta consideración se debe a que es el escenario más desfavorable para los actuadores del robot, además de que simplifica la implementación de los algoritmos sucesivos.

## Método de Lagrange

Este método se basa en las consideraciones energéticas del mecanismo a evaluar, esto es, obtener las ecuaciones dinámicas del mismo partiendo de elaborar las ecuaciones correspondientes a la energía cinética y potencial de cada eslabón.

$$\sum_i^n \frac{d}{dt} \left( \frac{\delta L}{\delta \dot{\theta}_i} \right) - \frac{\delta L}{\delta \theta_i} = \tau_i$$

Mediante la aplicación de un algoritmo computacional es posible obtener el modelo dinámico del mecanismo planteado.

Se implementó el algoritmo computacional de Lagrange Euler en el software MATLAB, donde fue necesario brindar como datos, la matriz de pseudoinercias, los vectores de distancia del centro de masa respecto al sistema de coordenadas de cada eslabón, las matrices de transformación de cada eslabón con sus parámetros Denavit-Hartenberg, entre otros.

El resultado obtenido, dado que el software realizó todas las operaciones de forma simbólica es una larga sumatoria de términos simbólicos, que no se encuentra en forma reducida, para lo cual es necesario aplicar diversas factorizaciones y simplificaciones hasta llegar a la siguiente expresión: (página siguiente).

$$T = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = D \cdot \ddot{q} + H + C$$

$$D = \begin{bmatrix} m_3(l_1 + l_2 c_2 + l_3 c_{23})^2 + m_2(l_1 + l_2 c_2)^2 + l_1^2 m_1 & 0 & 0 \\ 0 & l_2^2(4m_2 + m_3) + 4l_3^2 m_3 + 4l_2 l_3 m_3 c_3 & m_3 l_3(l_2(c_{23} c_2 + s_2 s_{23}) + l_3) \\ 0 & m_3 l_3(l_2(c_{23} c_2 + s_2 s_{23}) + l_3) & l_3^2 m_3 \end{bmatrix}$$

$$H = \begin{bmatrix} -2\dot{\theta}_1 \dot{\theta}_3 m_3 l_3 s_{23}(c_1(l_1 + l_2 c_2 + l_3 c_{23})) - 2\dot{\theta}_2 \dot{\theta}_1(m_3(l_2 s_2 + l_3 s_{23})(l_1 + l_2 c_2 + l_3 c_{23}) + l_2 m_2 s_2(l_1 + l_2 c_2)) \\ (\dot{\theta}_3^2 + 2\dot{\theta}_2 \dot{\theta}_3) m_3 l_3 l_2(c_{23} s_2 - s_{23} c_2) + \dot{\theta}_1^2(m_3(l_2 s_2 + l_3 s_{23})(l_1 + l_2 c_2 + l_3 c_{23}) + l_2 m_2 s_2(l_1 + l_2 c_2)) \\ \dot{\theta}_1^2 \cdot (m_3 l_3 s_{23}(l_1 + l_2 c_2 + l_3 c_{23})) + \dot{\theta}_2^2 \cdot (m_3 l_3 l_2(s_{23} c_2 - c_{23} s_2)) \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \\ g \cdot m_3 \cdot (l_2 \cdot c_2 + l_3 \cdot c_{23}) + g \cdot l_2 \cdot m_2 \cdot c_2 \\ g \cdot m_3 \cdot l_3 \cdot c_{23} \end{bmatrix}$$

$$\ddot{q} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}$$

Donde  $s_2 = \sin(\theta_2)$ ;  $c_2 = \cos(\theta_2)$ ;  $s_3 = \sin(\theta_3)$ ;  $c_3 = \cos(\theta_3)$

Se anexa al final del informe el código de MATLAB utilizado.

### **Método de Newton-Euler**

Este método se basa en las consideraciones vectoriales del mecanismo a evaluar, esto es, obtener las ecuaciones dinámicas del mismo partiendo de elaborar las ecuaciones correspondientes a las fuerzas y torques a las que se comete cada eslabón del mecanismo.

$$\sum F = m\dot{v} \wedge \sum T = I\omega + \omega \times (I\omega)$$

De forma similar al caso anterior se implementó el algoritmo computacional en MATLAB, ingresando esta vez con las matrices de rotación asociadas a los eslabones, las condiciones iniciales de velocidad y aceleración de la base, realizando iteraciones hacia delante y hacia tras finalmente se obtuvo una expresión simbólica que resulta necesario factorizar y simplificar, de la cual se obtiene la misma expresión obtenida con el método de Euler Lagrange.

Se anexa al final del informe el código de MATLAB utilizado.

### Simulación del modelo dinámico

Para poder estimar el comportamiento de la extremidad del robot móvil, se resuelve implementar en Simulink el modelo dinámico del sistema. El mismo consiste en una función donde se cargan los valores de las matrices  $D$ ,  $C$  y  $H$ , cumpliendo con la ecuación de estados:  $\tau = D \cdot \ddot{q} + H + C$ . Además se le adiciona a esta ecuación el torque producido por fricción viscosa que sigue una ecuación lineal de la siguiente forma:  $T_{fi} = cf \cdot \dot{\theta}_i$  y un posible torque de perturbación producido por fuerzas externas.

Cargando los diversos parámetros del sistema, el código implementado en la simulación toma la siguiente forma:

```
function [qdd1,qdd2,qdd3] =  
fcn(q1,q2,q3,w1,w2,w3,tau1,tau2,tau3,f1,f2,f3)  
%#codegen  
m1=0.15; % 150 gramos  
m2=0.15; % 150 gramos  
m3=0.05; %50 gramos  
g=9.8;  
l1=0.1;%10 cm  
l2=0.1;  
l3=0.07;  
  
%Matriz de perturbaciones  
ta=[tau1;tau2;tau3];  
dist=[f1;f2;f3]; %Matriz influencia de la fricción  
  
%Variables articulares  
theta1=q1;  
theta2=q2;  
theta3=q3;  
  
s23=sin(theta2 +theta3);  
c23=cos(theta2 +theta3);  
c1=cos(theta1);  
s1=sin(theta1);  
c2=cos(theta2);  
s2=sin(theta2);  
c3=cos(theta3);  
s3=sin(theta3);  
  
D11=m3*(l1 + l2*c2 + l3*c23)^2 + m2*(l1 + l2*c2)^2 + l1^2*m1;  
D12=0;  
D13=0;  
D21=0;  
D22=m3*((l2)^2 + 2*l2*l3*(s2*s23 +c2*c23) + (l3)^2 )+ l2^2*m2;  
D23=m3*l3*(l2*(c23*c2 +s2*s23) + l3);  
D31=0;  
D32=m3*l3*(l2*(c23*c2 +s2*s23) + l3);  
D33=l3^2*m3;  
  
H1 = - 2*w1*w3*m3*l3*s23*(c1*(l1 + l2*c2 + l3*c23)) -  
2*w1*w2*(m3*(l2*s2+l3*s23)*(l1+l2*c2+l3*c23) + l2*m2*s2*(l1 + l2*c2));
```



```

H2 = w3^2*m3*l3*l2*(c23*s2 - s23*c2) +
w1^2*(m3*(l2*s2+l3*s23)*(l1+l2*c2+l3*c23) + l2*m2*s2*(l1 + l2*c2)) +
2*w2*w3*m3*l3*l2*(c23*s2 - s23*c2);

H3 = (w1^2*(m3*l3*s23*(l1 + l2*c2 + l3*c23)) + w2^2*(m3*l3*l2*(s23*c2 -
c23*s2)));

C1=0;
C2=g*(m3*(l2*c2+l3*c23) +l2*m2*c2);
C3=g*m3*l3*c23;

M=[D11,D12,D13;
    D21,D22,D23;
    D31,D32,D33]; %Matriz de inercias

H=[H1;H2;H3]; %Matriz de coriolis

C=[C1;C2;C3]; %Matriz de gravedad

qdd = inv(M)*(- C - H + ta - dist); %Calculo de aceleraciones

qdd1 = qdd(1);
qdd2 = qdd(2);
qdd3 = qdd(3);

```

Como primer paso se analiza el comportamiento de una sola extremidad del robot a lazo abierto, para la segunda articulación de la misma, con las articulaciones 1 y 3 estáticas. Se impone la restricción de anular los efectos de la gravedad para observar concretamente los efectos de la fricción.

Consigna:

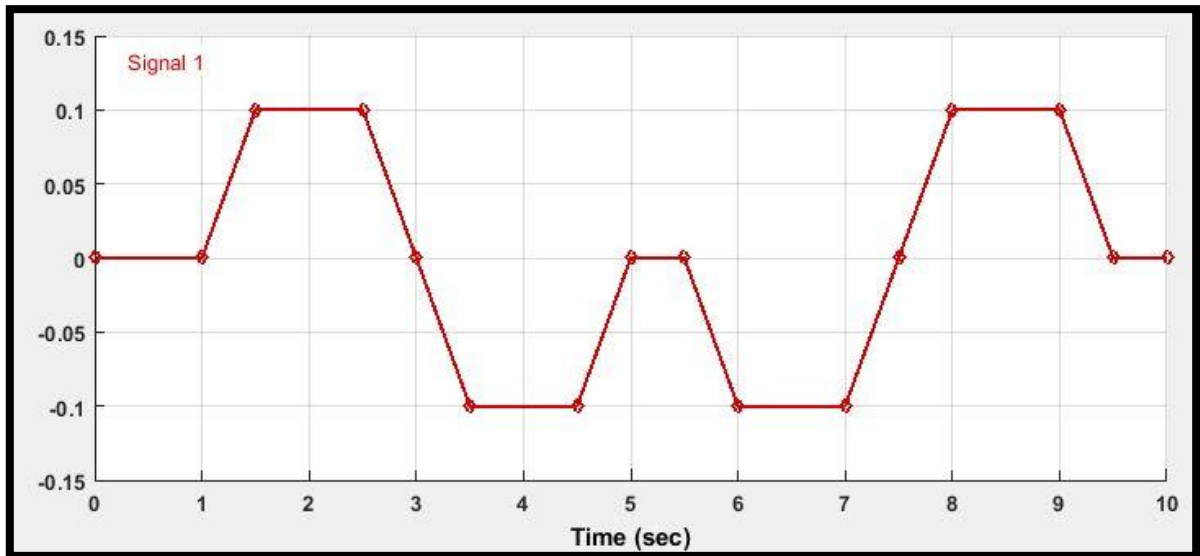


Imagen 1 Consigna de torque

Sin considerar las fuerzas de fricción ni de gravedad ( $g=0$ ), el resultado es el siguiente:

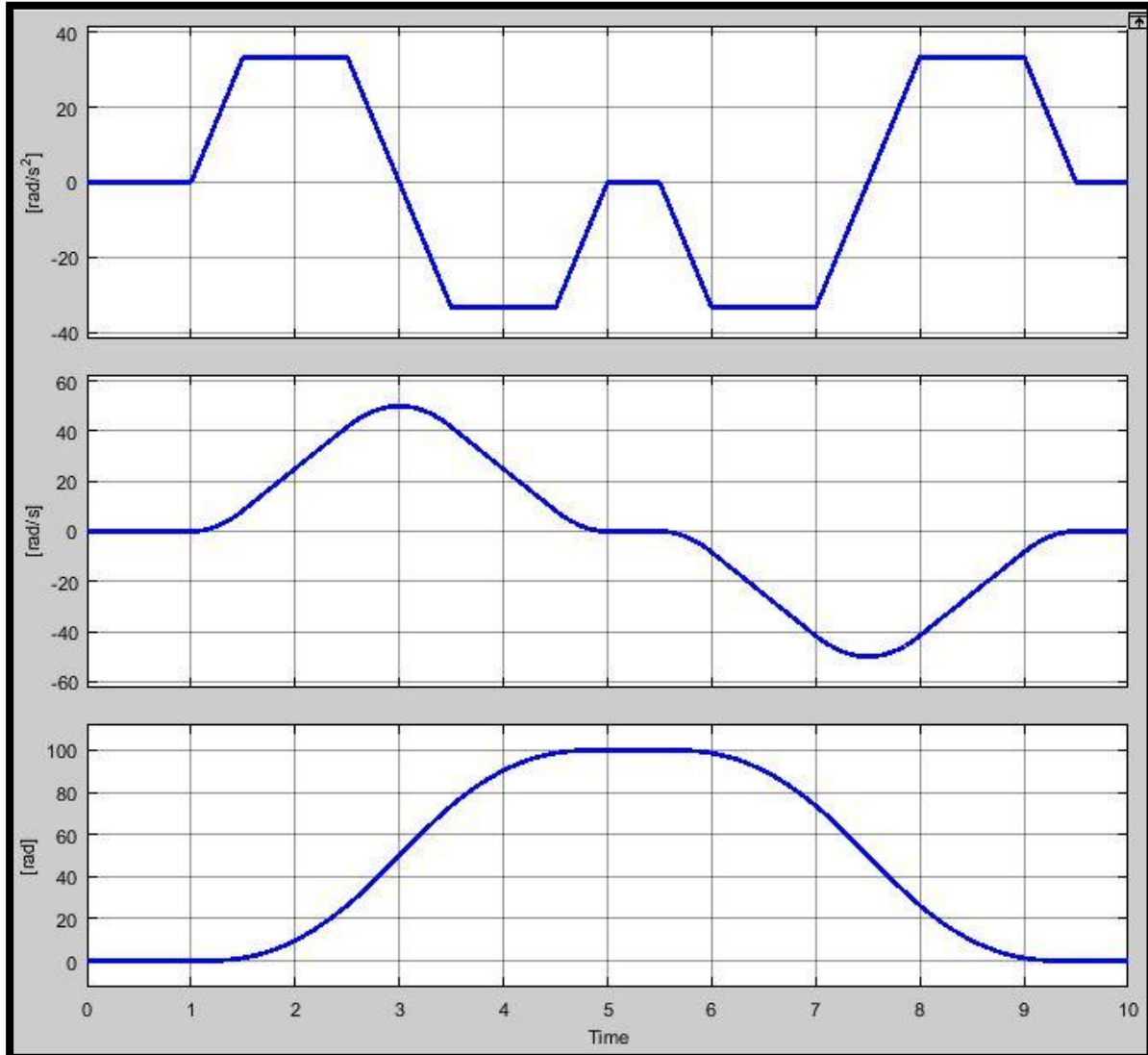


Imagen 2 Respuesta a lazo abierto - Gravedad y fricción nulas – Primer articulación

Se observa que el sistema sigue la consigna establecida sin errores.

Se utiliza el modelo de Stribeck para simular los torques de fricción en las articulaciones. Este modelo consiste en una fuerza de arranque para velocidad angular nula, fricción de coulomb y fricción viscosa (proporcional a la velocidad de rotación de la articulación). El modelo resulta

$$F = F_C + (F_{Str})^{-C_{str} \cdot \omega} + F_v$$

$$F_C = \text{signo}(\omega) \cdot \mu \cdot N \rightarrow \text{Fricción de Coulomb}$$

$\mu$ : Factor de fricción estático del material (Para plastico = 0.05)

$N$ : Componente normal de peso (Valor maximo 30N)

$F_{Str}$ : Fricción de arranque de Stribeck

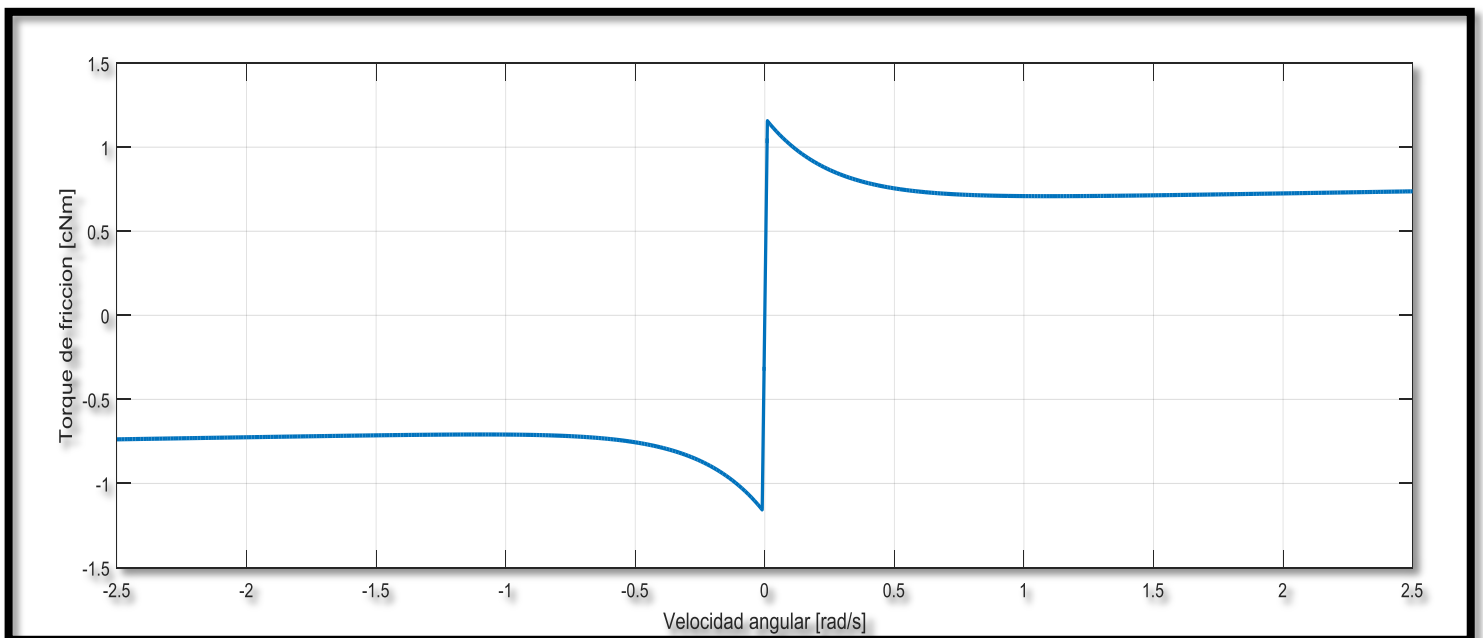
$C_{str}$ : Coeficiente de decrecimiento de fricción dinámica ( $C_{str} = 4 \left[ \frac{s}{rad} \right]$ )

$\omega$ : Velocidad angular  $\left[ \frac{rad}{s} \right]$

$$F_v = C_{fv} \cdot \omega \rightarrow \text{Fricción viscosa}$$

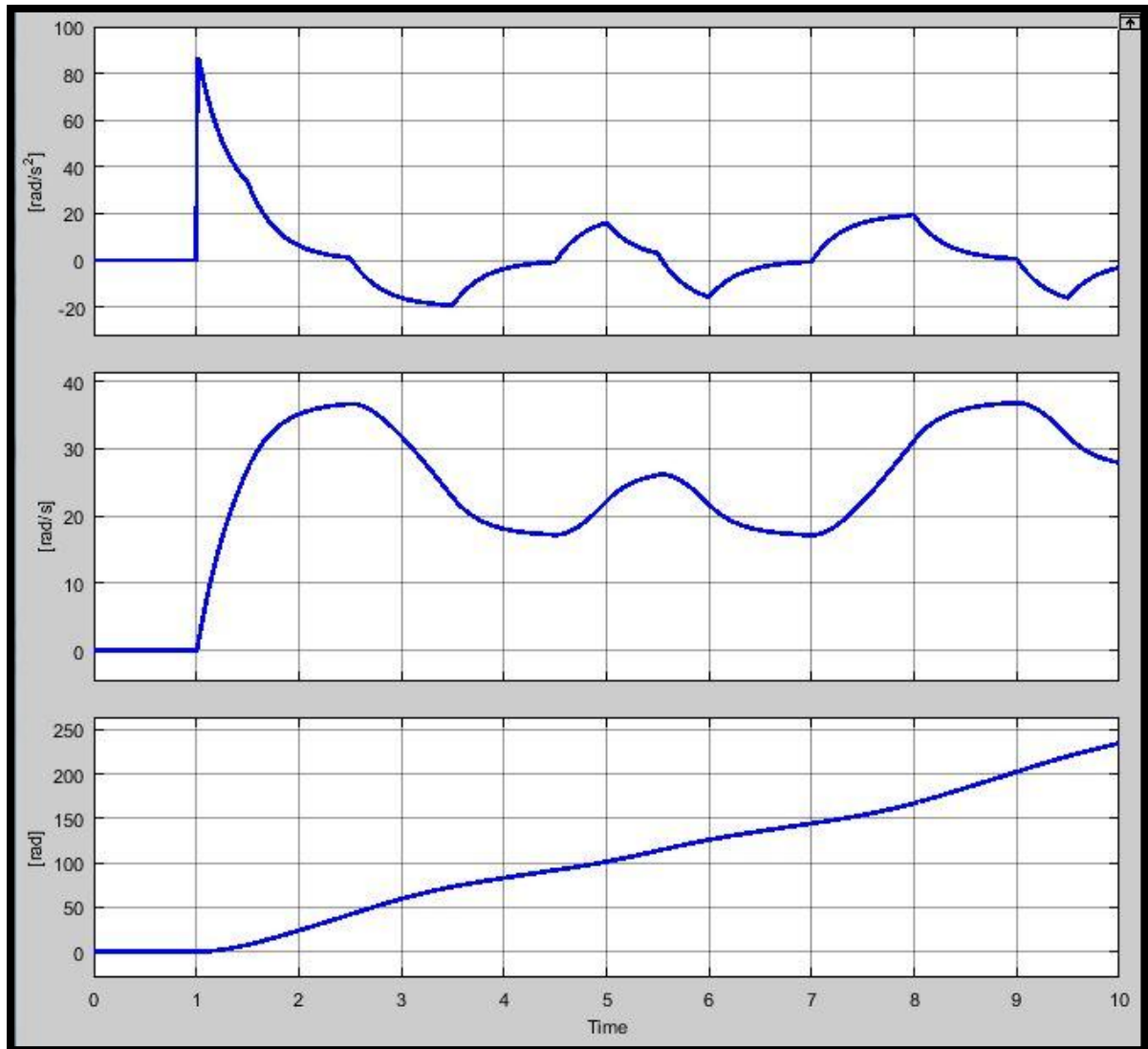
$C_{fv}$ : coeficiente de fricción viscosa ;  $C_{fv} = 0.025 \left[ \frac{Nms}{rad} \right]$

Graficando los torques de fricción obtenidos en función de la velocidad angular se obtiene:



Se anexa el código utilizado al final del informe.

Resultando el comportamiento a lazo abierto de la segunda articulación de la extremidad, dejando las otras dos articulaciones estáticas:



**Imagen 3 Respuesta a lazo abierto con fricción - Gravedad nula - Primer articulación**

Ante la misma consigna de torque es evidente como el sistema es incapaz de seguir la trayectoria establecida, resultando en un movimiento totalmente diferente al deseado.

## Influencia del movimiento de las articulaciones restantes

En las ecuaciones dinámicas se puede observar que la segunda articulación posee términos debidos a la posición relativa de la tercera articulación respecto a la segunda así como de la velocidad angular de la primera articulación, por lo que se deduce que el comportamiento de la segunda articulación se ve afectado por el movimiento de las demás articulaciones.

Nuevamente despreciando los efectos de rozamientos y de gravedad en la articulación para observar la influencia de las demás articulaciones sobre el movimiento de la articulación analizada se establece la misma consigna de posición a las últimas dos articulaciones, anulando los efectos de la fricción y se obtiene el siguiente resultado para la segunda articulación:

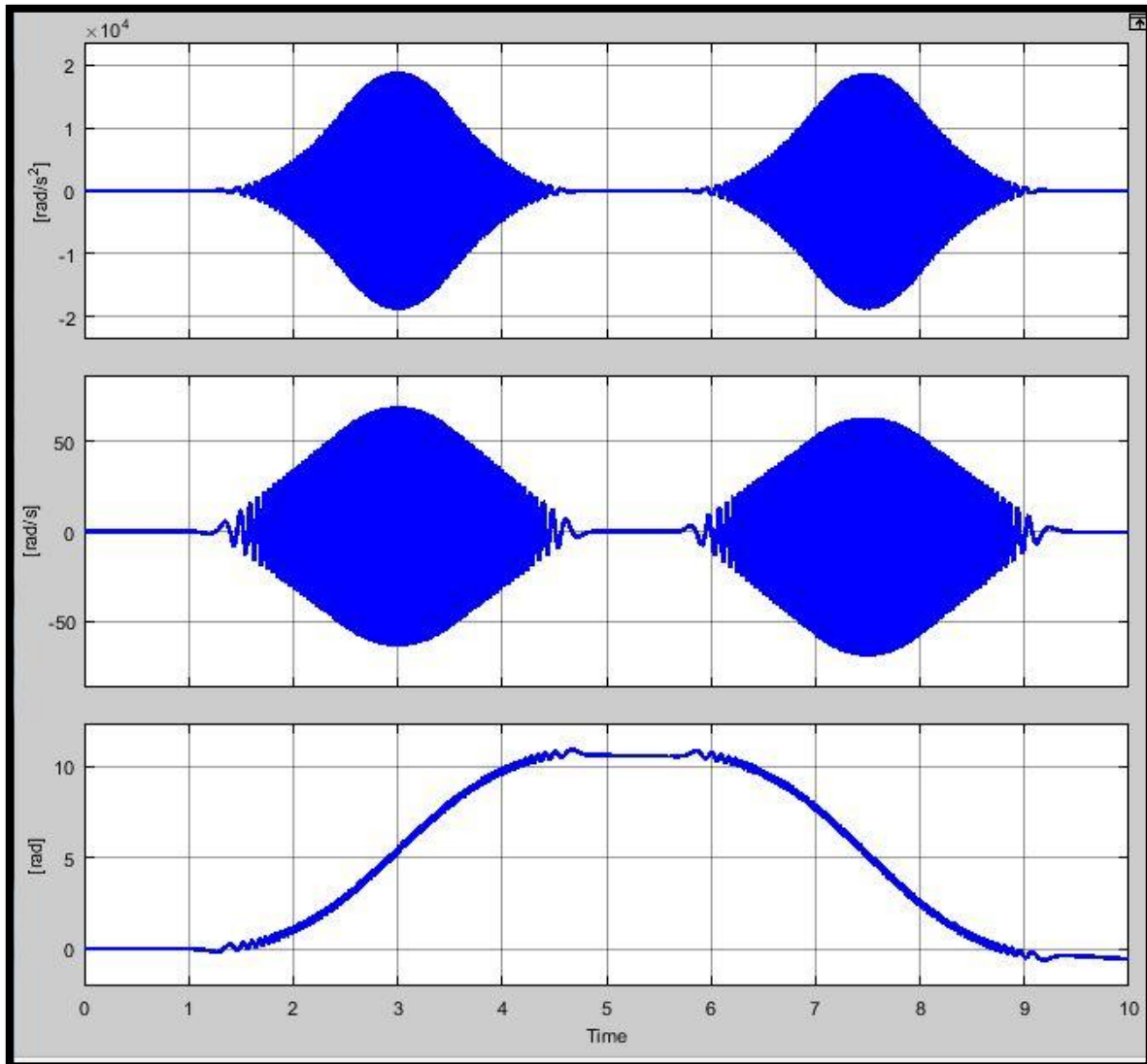


Imagen 4 Respuesta segunda articulación ante consigna de posición sobre la segunda y tercera articulación

La respuesta en conjunto de las dos articulaciones se observa en la imagen 5:

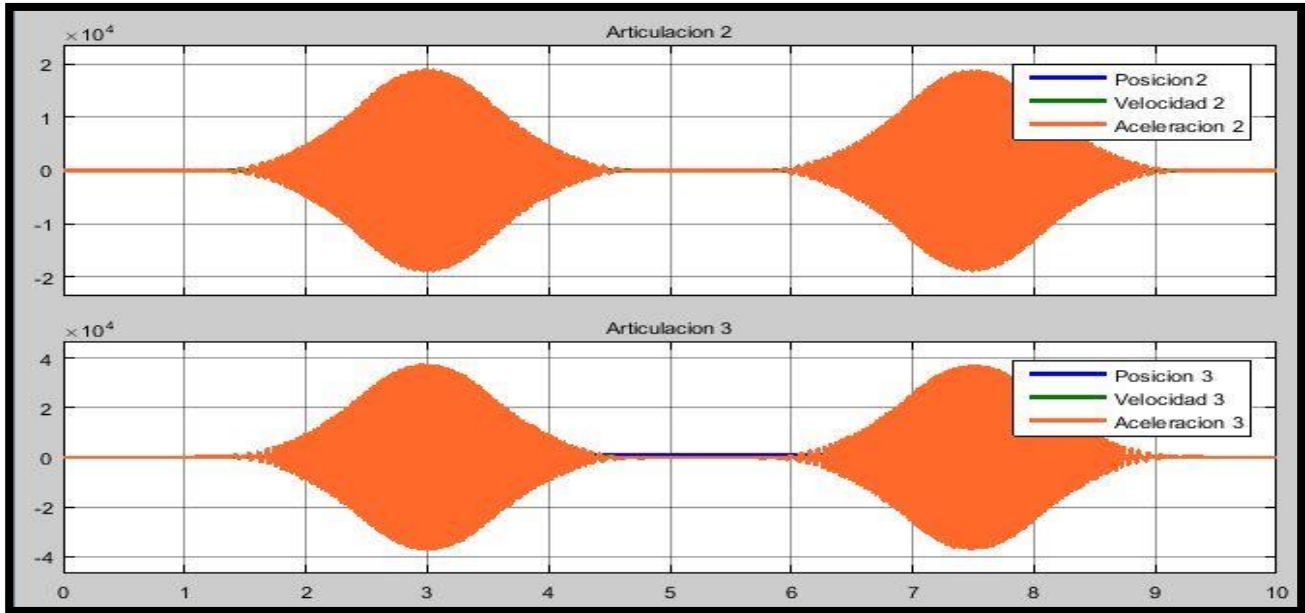


Imagen 5 Respuesta en conjunto de articulaciones 2 y 3

Se puede observar como la tercer articulación al poseer menos masa que la segunda, la magnitud de su aceleración angular es mayor que la correspondiente a la segunda, esto se traduce en que el movimiento de la tercera articulación transmitirá un torque hacia la segunda articulación de mayor magnitud y sentido opuesto al torque de consigna, acelerando en sentido opuesto al caso analizado previamente. Se procede a analizar el caso para el cual se consideran las fuerzas de fricción. El resultado se observa en la imagen 6.

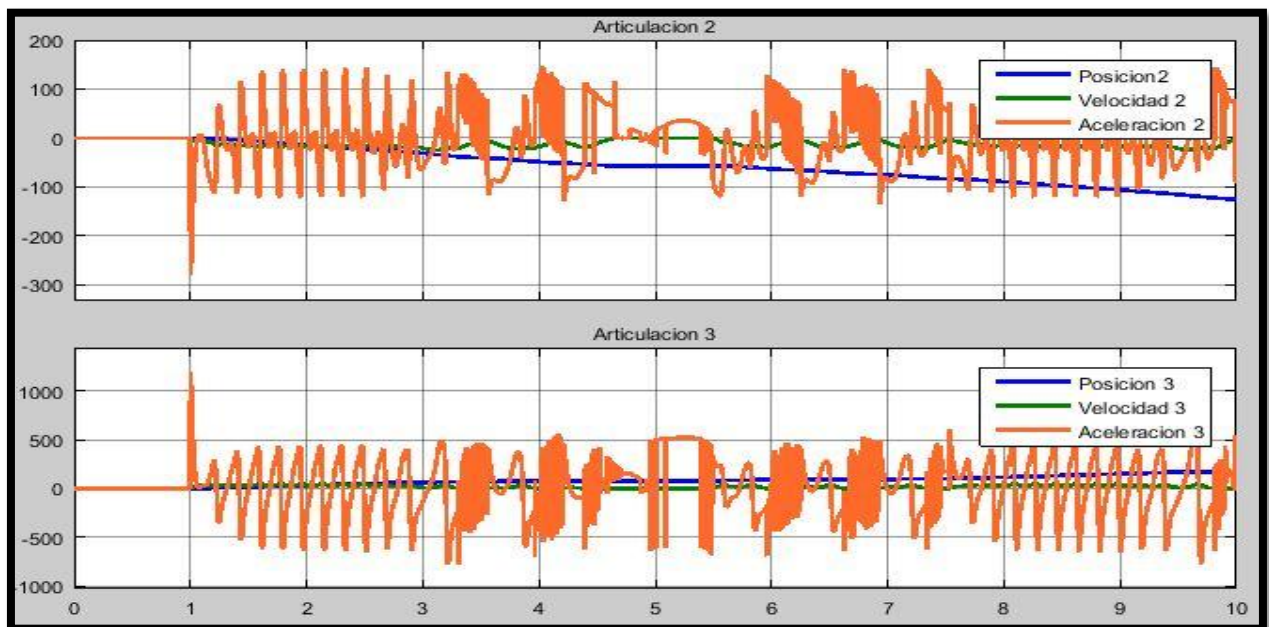


Imagen 6 Respuesta en conjunto de articulaciones 2 y 3 considerando fuerzas de fricción

Se puede observar nuevamente la influencia de la tercera articulación sobre la segunda en comparación al caso donde se restringía el movimiento de la tercera articulación.

**Agregando ahora los efectos de la gravedad:**

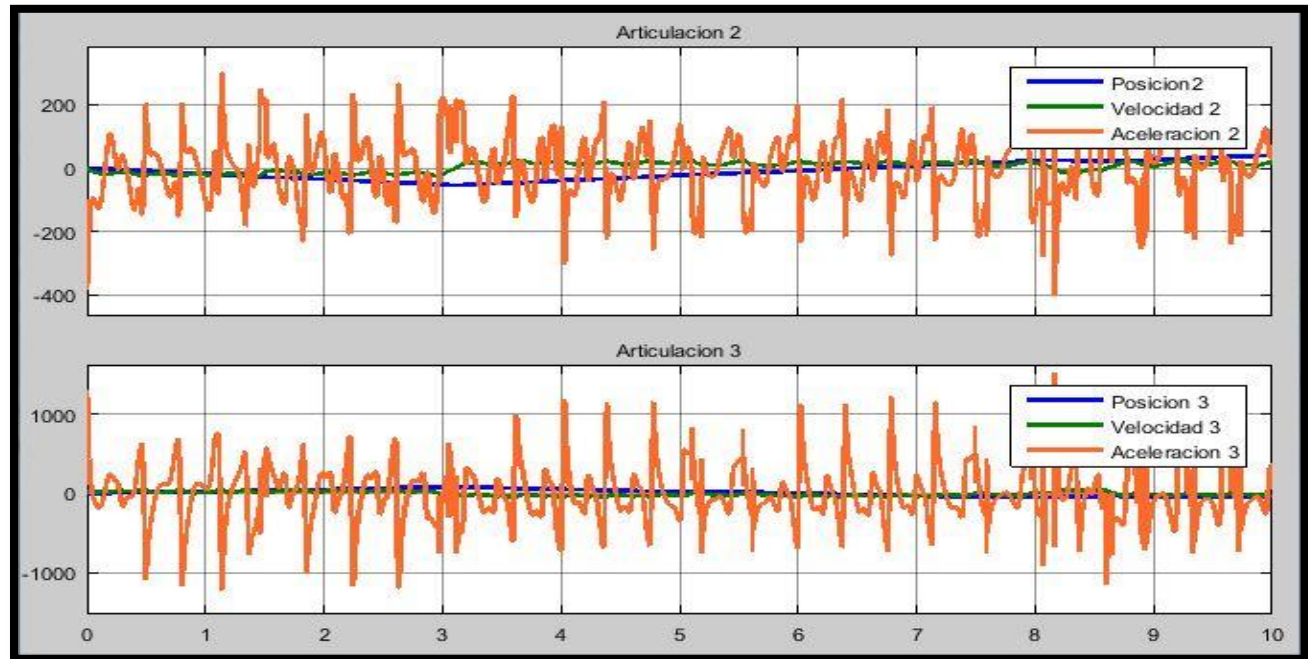


Imagen 7 Respuesta en conjunto de articulaciones 2 y 3 considerando fuerzas de fricción y de gravedad

Lo observado en esta simulaciones y considerando que el robot posee dieciocho articulaciones en total, se puede concluir que no se puede considerar cada eslabón de las extremidades del robot por separado, es decir ignorar el efecto de los demás grados de libertad sobre el movimiento de una articulación en particular.



## Control del robot

De lo observado en el análisis de influencia entre articulaciones, se desprende el hecho de que el control de la extremidad del robot no puede realizarse de forma tal que se considere cada eslabón de la extremidad por separado, es decir, ignorando la influencia del movimiento de los demás eslabones de la cadena cinemática, por lo cual el método de control a utilizar deberá ser de tipo multiarticular.

Analizando el diseño constructivo del robot, se propone de antemano utilizar como actuadores de cada articulación, servomotores analógicos, con vista a utilizar servomotores digitales. Estos actuadores poseen un elevado factor de reducción  $K$ , lo cual permite dentro de lo razonable, desacoplar el movimiento de los eslabones entre sí o mejor dicho, considerar que la influencia de cada articulación no afecta el movimiento de las demás.

Se opta por utilizar un controlador PID para cada articulación de las extremidades.

### Controlador PID

Dado el gran requerimiento computacional que requiere el sistema completo (control de 18 articulaciones), será necesario utilizar un controlador de bajo coste computacional, que sea capaz de cumplir con el seguimiento de las consignas establecidas a diferencia de la sobrecarga que produciría por ejemplo un sistema con métodos de parámetros evolutivos. Además dada las diversas aplicaciones a las que se someterá el robot, resulta tedioso el análisis de todos los escenarios en caso de querer utilizar un sistema de ganancias variables.

Entonces para compensar los efectos de las perturbaciones, se le adiciona al sistema un controlador PID, brindado por Simulink, a cada articulación. Como se observa en la siguiente figura:

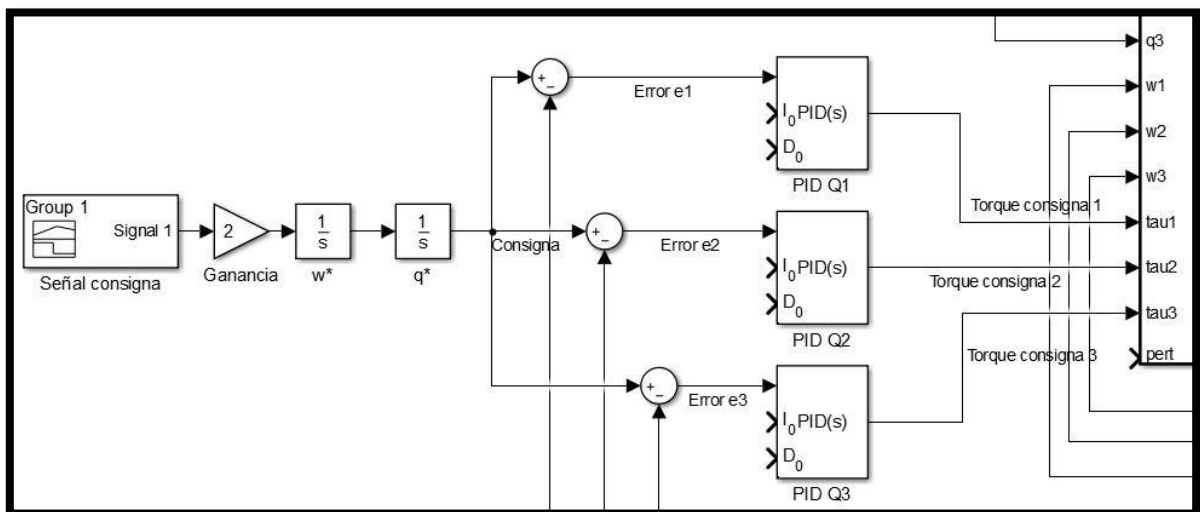


Imagen 8 Controlador PID de las tres articulaciones de una pata

El modelo en bloques en Simulink entonces resulta:

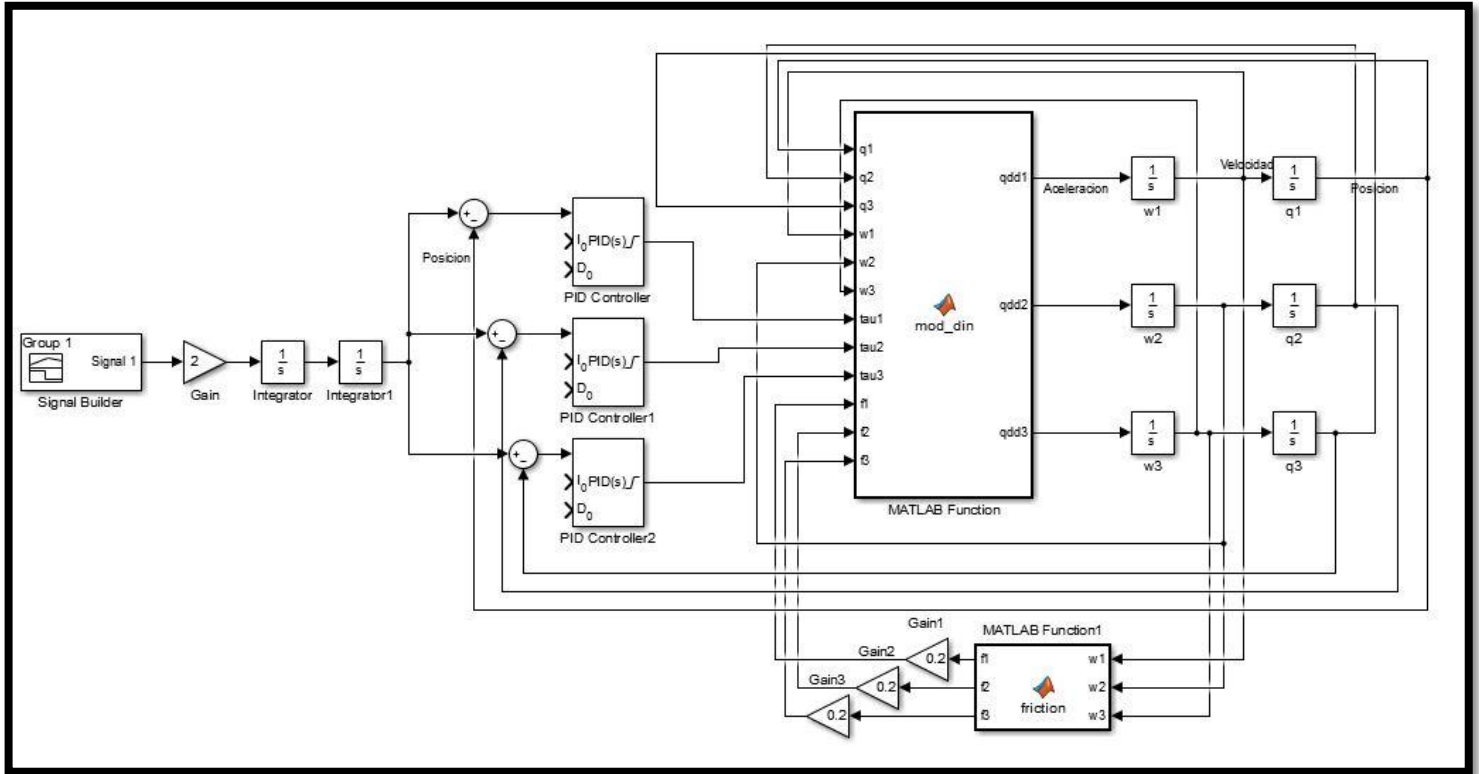


Imagen 9 Sistemas mecánico y de control - Modelo completo

Se introduce posteriormente la siguiente consigna de torque, que integrada dos veces, da una consigna de posición.

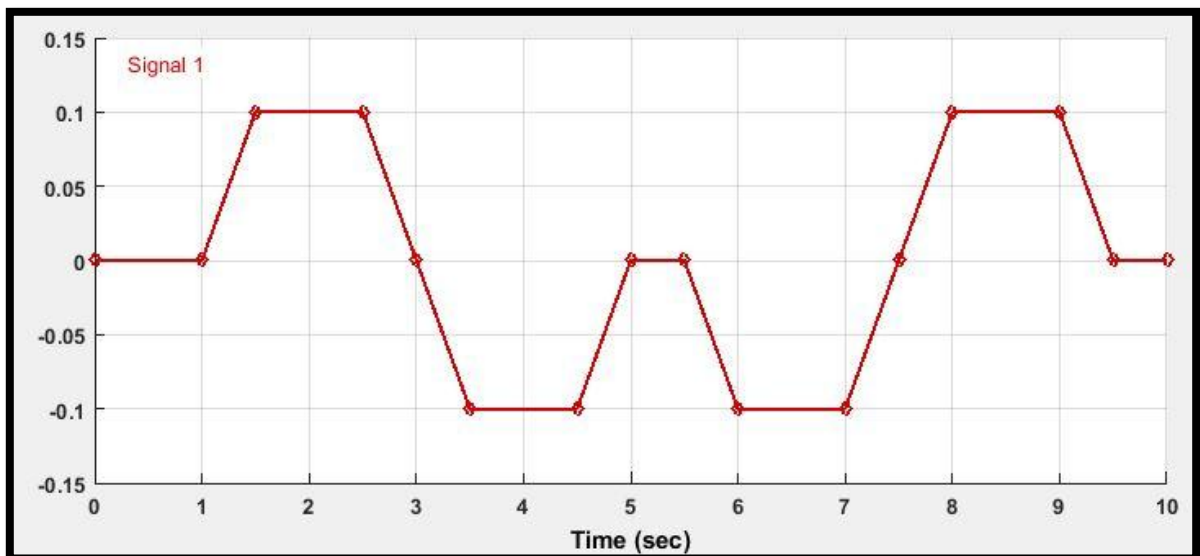


Imagen 10 Señal de torque consigna

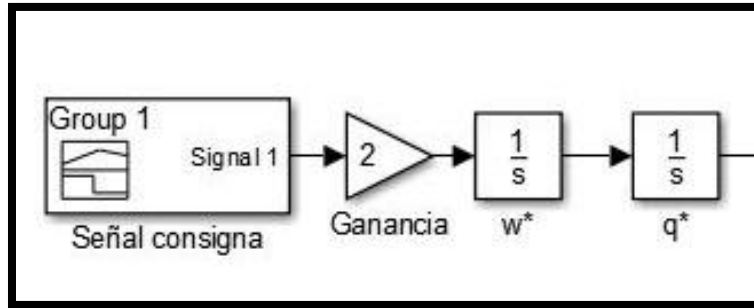


Imagen 11 Torque de consigna - Ganancia e integradores que permiten obtener consigna de posición

La señal de consigna resulta por lo tanto:

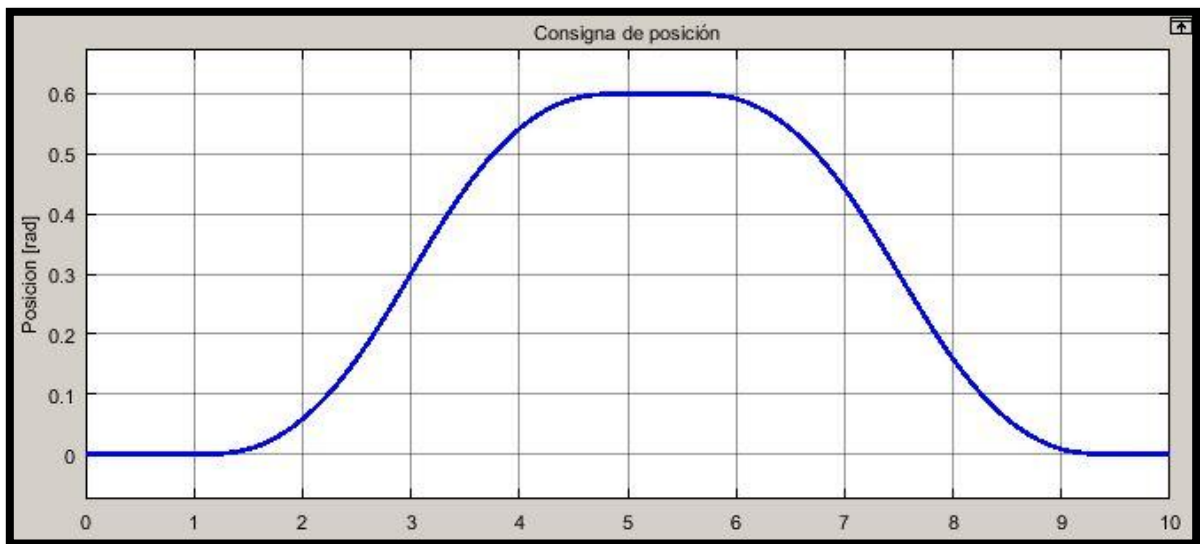


Imagen 12 Consigna de posición resultante en radianes

Se sintonizan los lazos de control con la herramienta de ajustes de parámetros para el controlador PID que brinda Simulink:

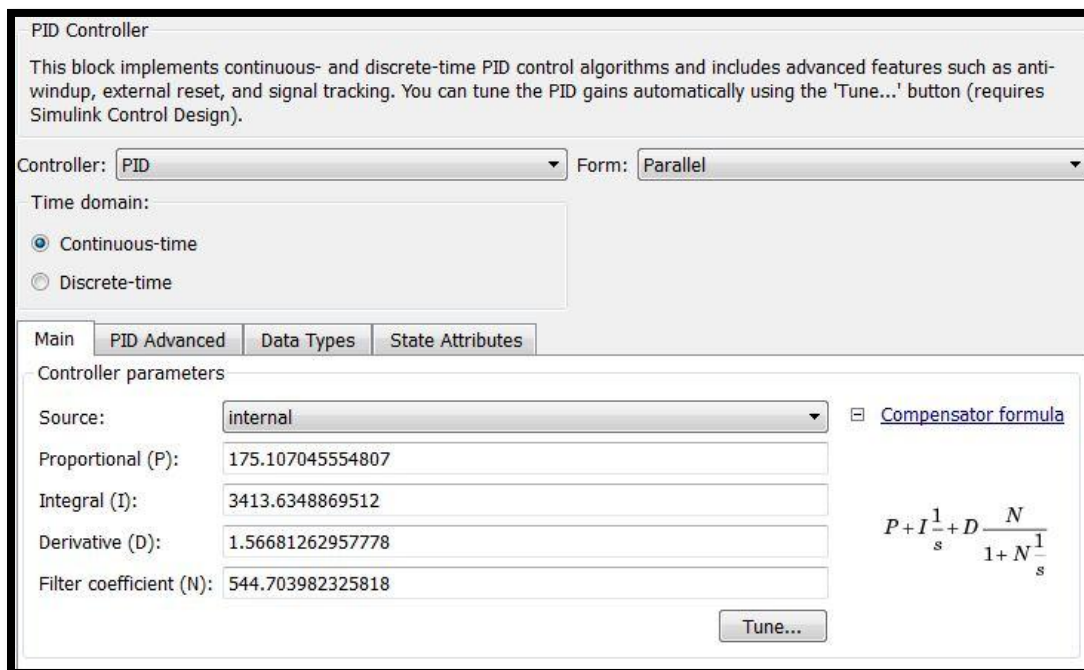


Imagen 13 Herramienta de configuración de controlador PID

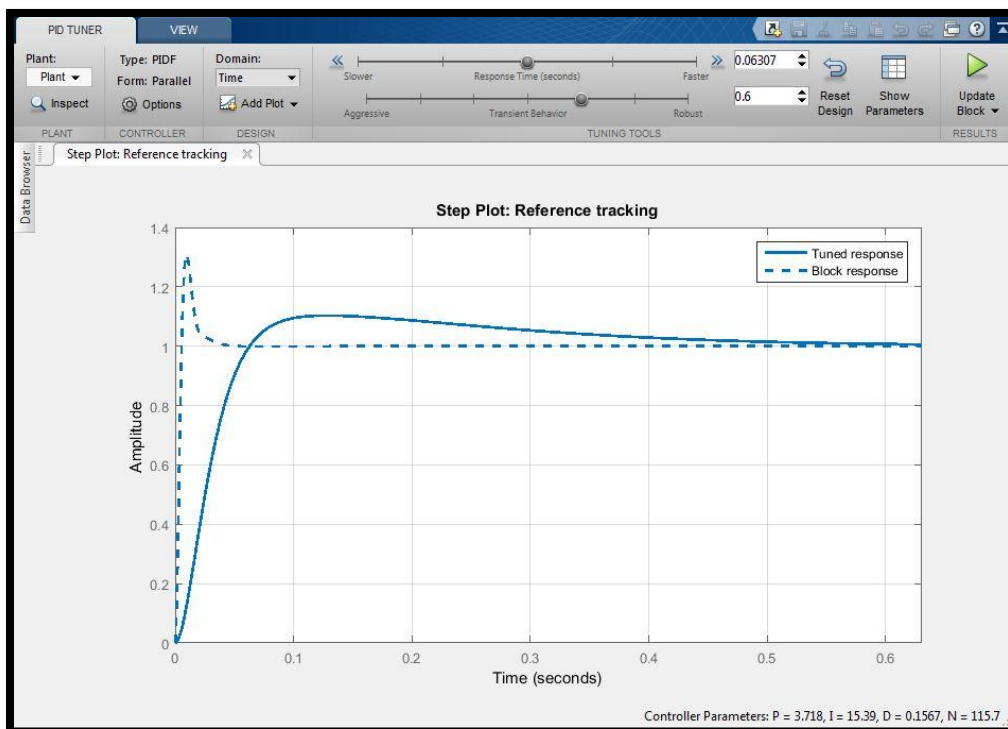


Imagen 14 Sintonizador de ganancias de PID

Luego de la sintonización, la respuesta de cada articulación de una sola extremidad del robot ante la misma consigna es la siguiente:

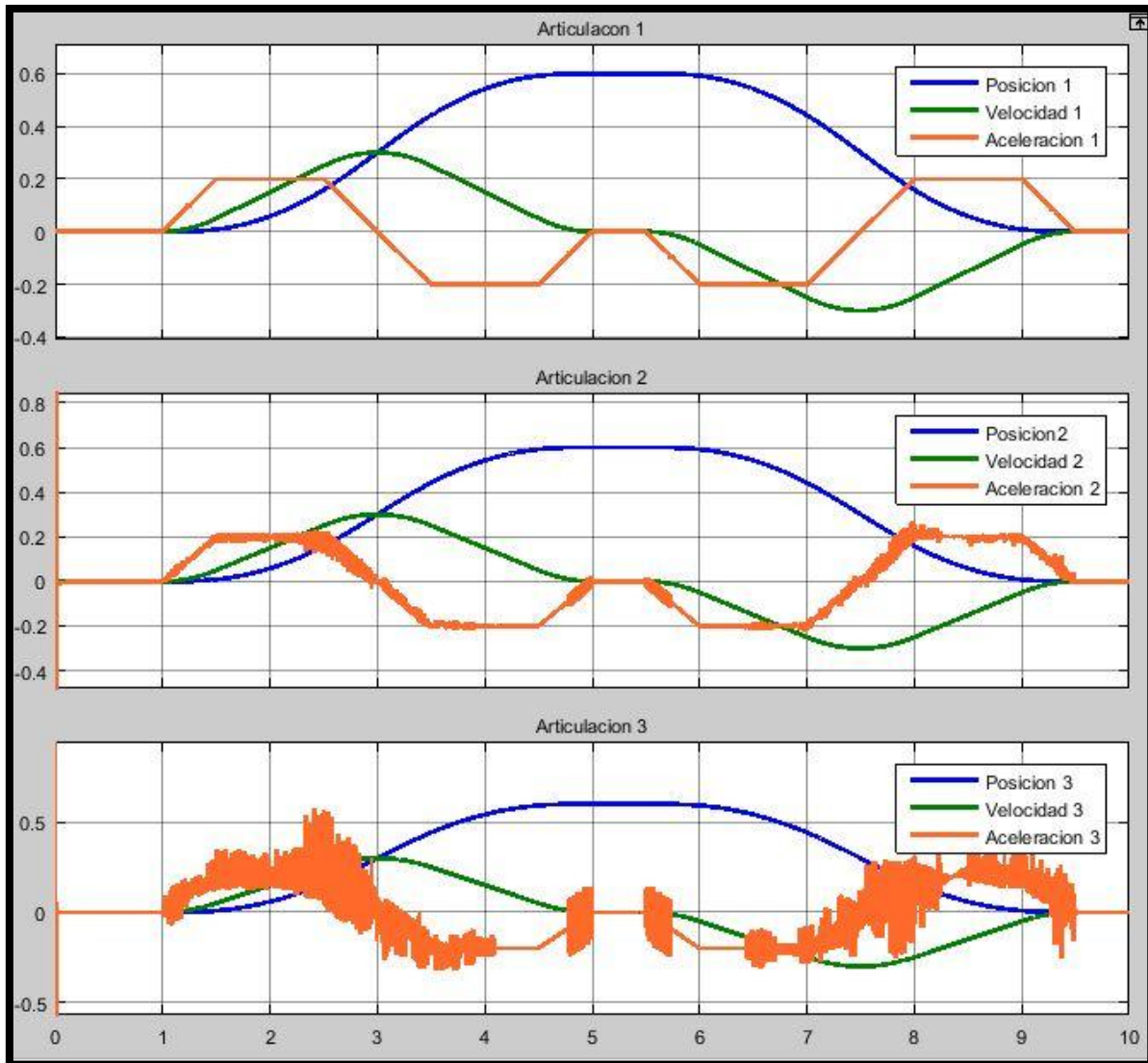


Imagen 15 Respuesta a lazo cerrado de las tres articulaciones

Se obtiene un seguimiento aceptable sin errores de estado estacionario. El transitorio observado al inicio de la simulación se debe al 'golpe' que genera en el tiempo  $t=0$  el adición instantáneo de la influencia de la gravedad. El ruido observado en la aceleración (naranja) se debe a los cambios en el torque de fricción.

En color azul se observa la posición en radianes ( $rad$ ), en verde la velocidad angular en radianes por segundo ( $\frac{rad}{s}$ ) y en naranja la aceleración en radianes por segundo al cuadrado ( $\frac{rad}{s^2}$ ) de la articulación.

El esfuerzo de accionamiento, es decir el torque aplicado por el actuador en cada articulación es el siguiente:

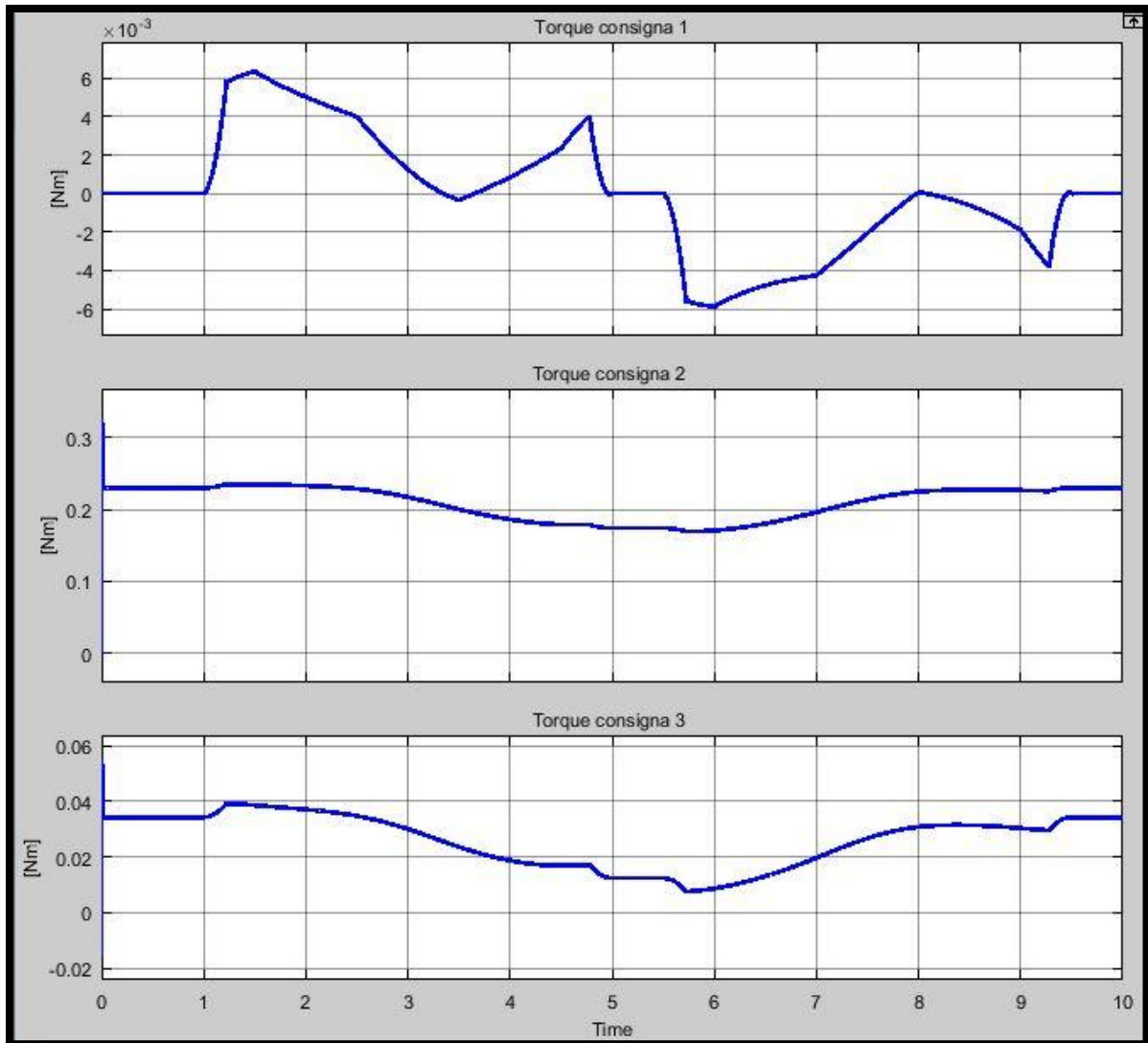


Imagen 16 Torques consigna en cada articulación

Este torque es el medido a la salida del controlador PID. El valor máximo (pico) de torque para un cambio de 0,6 radianes ( $\approx 35^\circ$ ) en un tiempo de 4 segundos es de  $0,4 \text{ Nm} = 4,08 \text{ kgf.cm}$ .

Los torques de fricción correspondientes a cada articulación resultan:

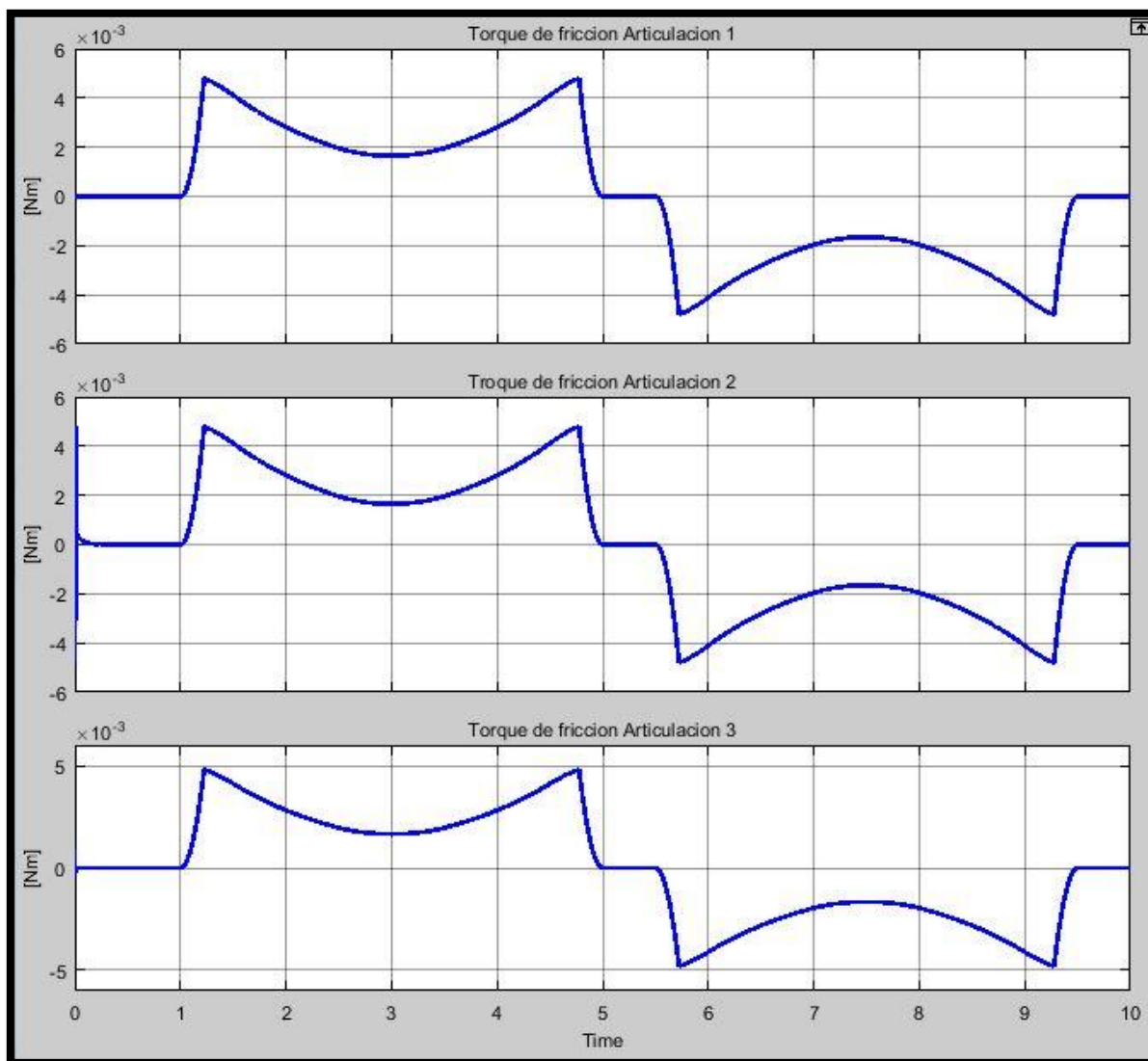


Imagen 17 Torques de fricción de cada articulación



## Navegación

El robot poseerá dos métodos de navegación, autónomo y guiado a través de control remoto. En el primer caso se selecciona el punto inicial y final del recorrido y el robot deberá alcanzar el destino por sus propios medios. Para esto utilizará todos los sensores descritos en el apartado anterior, en especial el GPS que brindará su posición absoluta en la superficie terrestre y una cámara montada a bordo, que mediante un algoritmo de IA reconocerá y evitará obstáculos.

Se propone utilizar un control proporcional integral (PI) para evaluar la diferencia de posición del robot con la trayectoria establecida, considerando al centro de masa del robot completo como punto a posicionar. Esta diferencia de posición u orientación resultará en una consigna de movimiento general del tipo 'Avanzar', 'Retroceder', 'Girar', etc., a una velocidad proporcional al error que posea la posición del hexápodo. Este controlador se desarrollara en detalle como trabajo final de otras cátedras correspondientes a Ingeniería Mecatrónica.

Para navegación por control remoto, el operario del robot establecerá comunicación por radio con el hexápodo recibiendo en su terminal los estados de posición, orientación, inclinación del terreno, imágenes obtenidas de la cámara, distancia a obstáculos, estado de la batería entre otros. En base al criterio del operario, este enviara comandos de movimiento al robot del mismo tipo que los generados por el controlador proporcional integral propuesto, donde la placa controladora de servos y el bloque de generación de trayectoria se encargaran del movimiento final del robot.

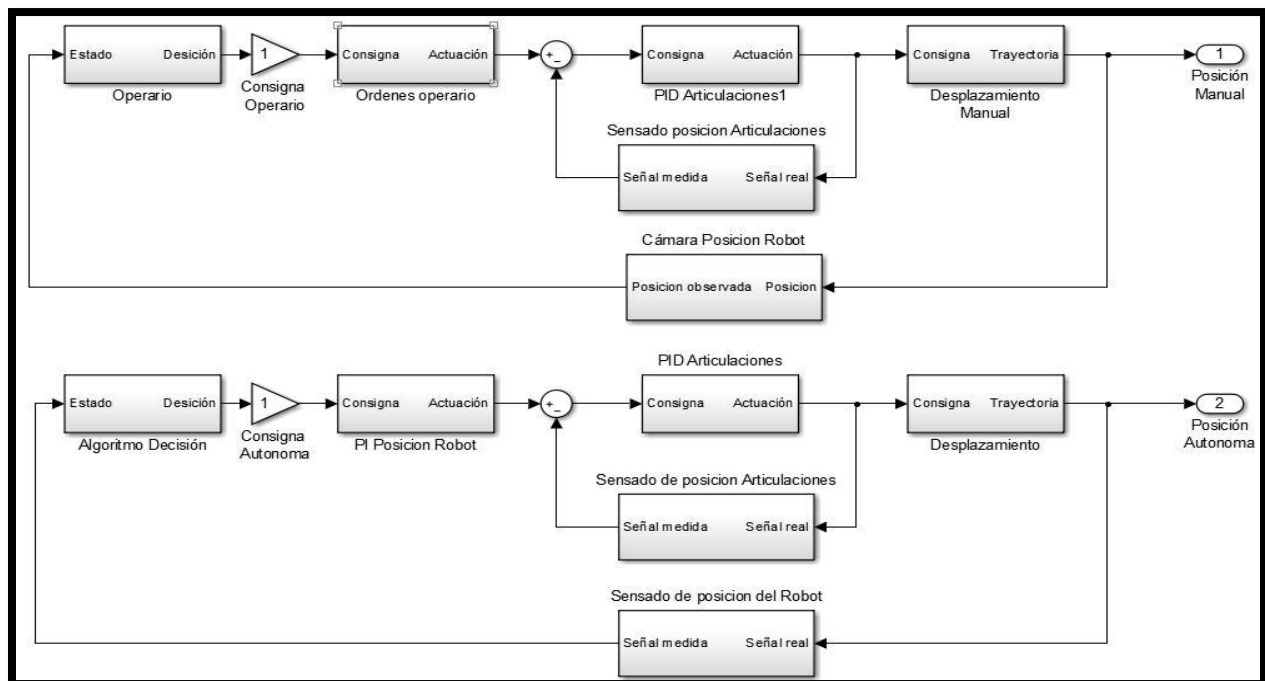


Imagen 18 Diagrama de bloques de método de control del robot



## Estabilización

Un requerimiento fundamental para cumplir correctamente con los objetivos de desplazamiento del robot, evasión de obstáculos y para el eventual caso de transporte de objetos es la correcta inclinación del cuerpo del hexápodo respecto al suelo, es decir que el cuerpo se mantenga siempre horizontal sin importar la pendiente del terreno, considerando que los límites físicos del robot permiten una estabilización de hasta 30°.

El desarrollo de este controlador se llevó a cabo como proyecto final de la cátedra Control y Sistemas de la Facultad de Ingeniería de la Universidad Nacional de Cuyo y excede los objetivos de este informe, por lo tanto se describirá de forma breve su funcionamiento.

A partir de la información obtenida del acelerómetro y giróscopo integrado ubicado en el cuerpo del robot se puede conocer los ángulos de inclinación (pitch y roll) relativos a un sistema de ejes perpendiculares entre si y normales a la dirección de la fuerza de gravedad. En base a esta información y conociendo la geometría ya establecida del robot, se calcula la altura que deberán de ascender o descender las extremidades del robot para cumplir con la consigna establecida de mantener permanentemente el cuerpo de forma horizontal.

## Coordinación de movimientos

Dada la complejidad que implica la coordinación entre las extremidades del robot a fin de evitar colisiones entre ellas y al mismo tiempo navegar por el terreno será necesario como mínimo contar con un algoritmo encargado de enviar las señales correspondientes de consigna al controlador PID de cada extremidad en concordancia con las consignas procedentes del sistema de navegación y del controlador de estabilidad.

Además deberá cumplir con los requisitos de un relativo bajo costo computacional junto con una capacidad para realizar todo tipo de maniobras, en el caso de robot hexápodo las mismas deberán ser: Paso simple, Paso doble y Paso Trípole. La secuencia de pasos se puede observar mejor en la imagen 22.

Paso simple: Se realiza el desplazamiento de a una extremidad por vez, manteniendo estáticas las cuatro extremidades restantes..

Paso doble: Se realiza el desplazamiento de a dos extremidades por vez, mientras se mantienen estáticas las cuatro extremidades restantes.

Paso trípole: se desplazan tres extremidades en simultáneo. Las mismas formaran un triángulo para mantener la estabilidad estática del hexápodo.

Estas secuencias de pasos se han obtenido de forma experimental por otros autores mediante observación de diversos animales en la naturaleza. Estas secuencias de pasos no solo permiten tener distintas velocidades de desplazamiento sino también una mejor maniobrabilidad en terrenos accidentados. A su vez la utilización simultanea de menos de tres patas permite

transportar cargas mayores que en el caso de paso trípode, pero reduciendo de esta manera la velocidad de desplazamiento.

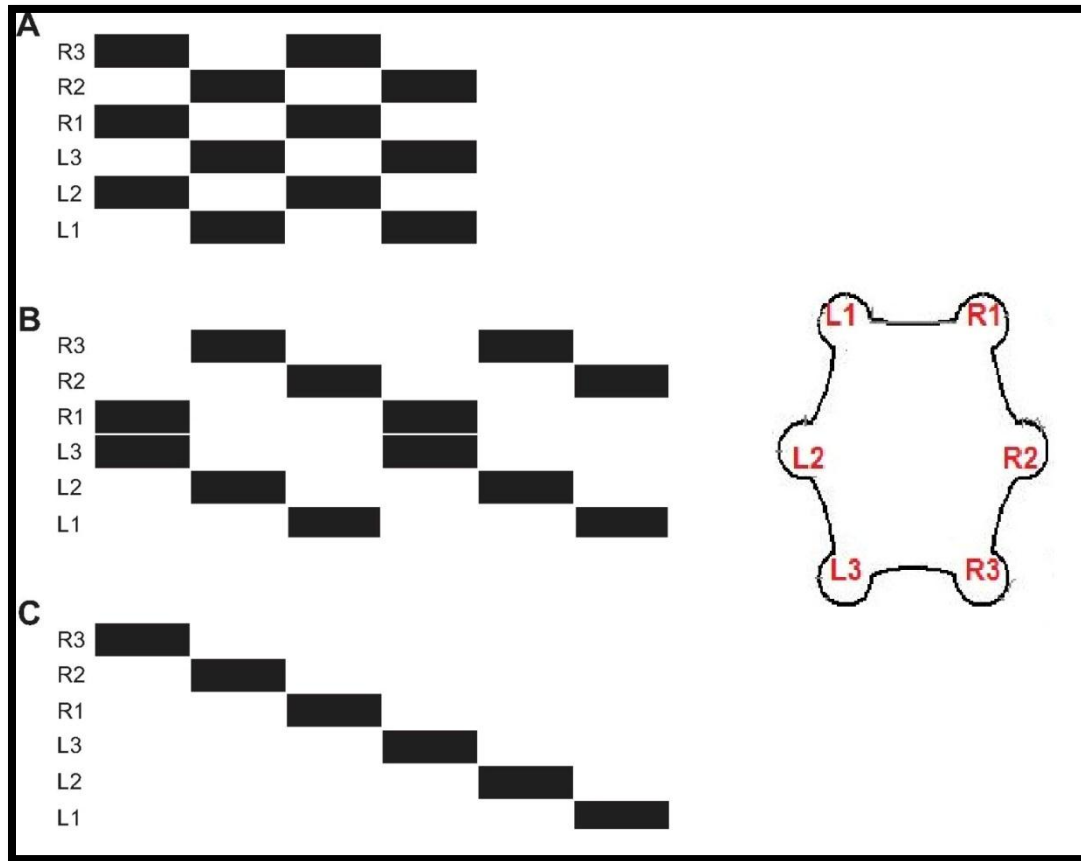


Imagen 19 Secuencia de pasos del hexápodo: A- Paso trípode; B- paso Doble; C- Paso simple

Se propone realizar la implementación del algoritmo como una función o proceso en el microcontrolador a bordo del robot, la cual no se incluye en el informe actual.

### Verificación de esfuerzos estáticos

Para determinar los esfuerzos totales resultantes en cada articulación es necesario determinar además, el torque máximo para cada articulación en el momento en el cual el robot realiza una maniobra de desplazamiento. Dado que la forma más sencilla de este robot de desplazarse es dejar tres patas estáticas mientras las otras tres se elevan y avanzan hasta apoyarse en el suelo, las tres patas quietas deberán soportar el peso total del robot de forma estática, lo acarrea la necesidad de calcular el torque que deberá aplicarse en cada articulación para evitar que el robot se desplome.

Se considera el peso total del robot en 3 kg (29.4 N), cada pata deberá soportar 9.8 N. Esta fuerza se distribuye entre las dos articulaciones que se ven afectadas por la gravedad. Para la segunda articulación el esfuerzo estático máximo será:

$$\tau_{e2} = P_m \cdot d_2 = 9.8 \text{ N} \cdot 0.065 \text{ m} = 0.637 \text{ Nm}$$

Este esfuerzo aparecerá en la configuración en la cual el segundo eslabón se encuentre completamente horizontal. Para la tercera articulación el esfuerzo estático máximo será:

$$\tau_{e3} = P_m \cdot d_3 + P_2 \cdot d_{23} = 9.8 \text{ N} \cdot (0.065 + 0.077) \text{ m} + 1 \text{ N} \cdot 0.077 \text{ m} = 1.46 \text{ Nm}$$

Dado que este valor es mayor que el torque suministrado por el actuador, deberá evitarse la configuración en la cual la tercera articulación se encuentre a completamente horizontal estando el eslabón 2 también horizontal. Esto se debe a que el brazo de palanca es máximo y el ángulo que forma con la componente del peso es 90°. Se deberá verificar que la configuración de posición de la pata sea tal que minimice el torque aplicado en la tercera articulación.

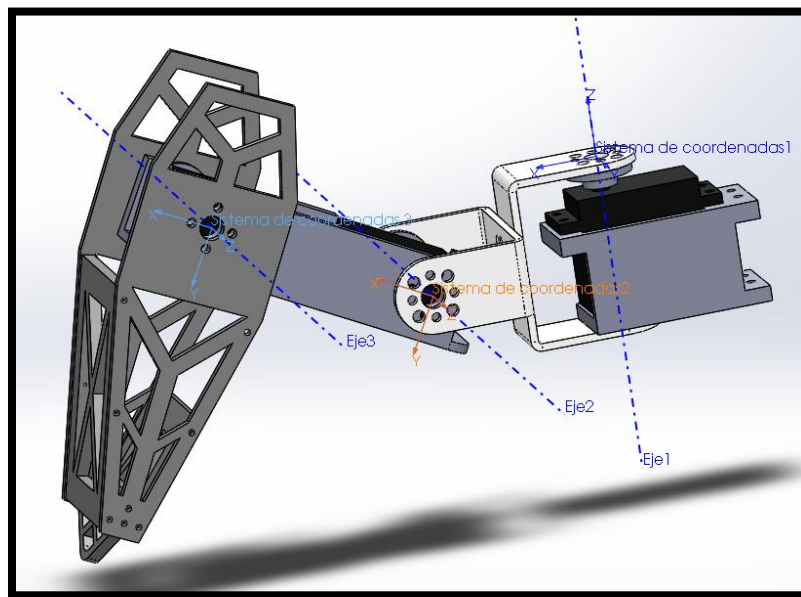


Imagen 20 Posición de apoyo de una pata

## Selección de sensores y actuadores

### Sensores del robot:

Dado el objetivo del robot de recorrer diversas zonas surge la necesidad de orientar y posicionar correctamente el robot en el espacio. Para esto es necesario que el mismo pueda detectar obstáculos cercanos, conocer su ubicación en el espacio y su orientación respecto a una referencia. Esto lleva a determinar los siguientes sensores:

**Ultrasonido:** Se propone utilizar el sensor ultrasónico de distancia HC-04. La implementación de este sensor permite detectar y medir la distancia de objetos que obstaculicen la pasada del robot. En base a estas mediciones el robot decidirá qué acción tomar frente a los mismos.

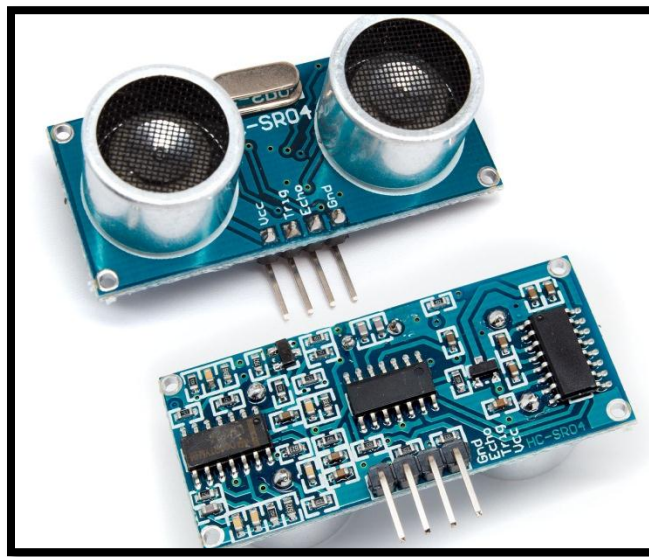


Imagen 21 Sensor ultrasónico HC SR04

**Cámara:** A través de un algoritmo basado en reconocimiento de objetos, se podrá reconocer obstáculos a lo largo del recorrido del robot, además de facilitar el modo de navegación a través de control remoto. Se propone utilizar el siguiente módulo de cámara para Raspberry por su fácil implementación y disponibilidad en el mercado.

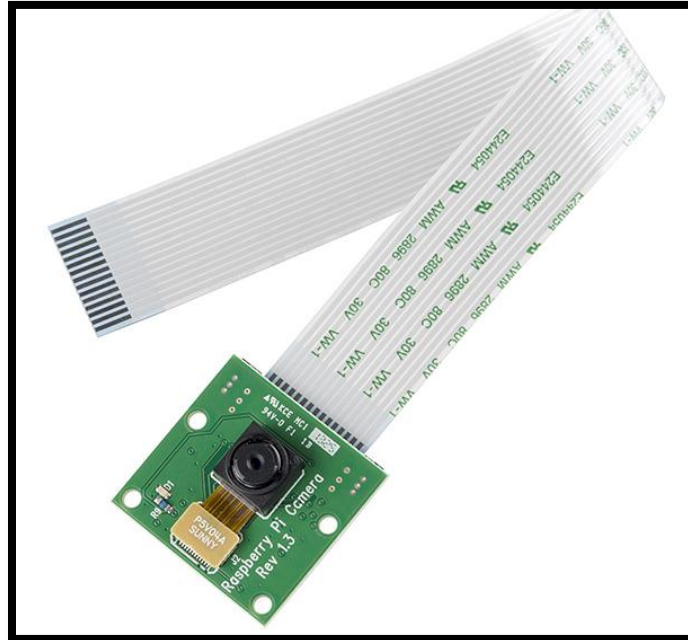


Imagen 22 Modulo de cámara 5 Mpx

**Comunicación:** Modulo Xbee. Utilizado para establecer una comunicación por radiofrecuencia punto a punto de forma inalámbrica con un alcance máximo de 300 m en ambientes abiertos. Permite el envío de consignas de navegación y de relevamiento del estado propio del robot. Cabe aclarar que para poder efectuar la comunicación se requiere de dos módulos, ambos funcionando como transmisor y receptor siendo uno móvil y el otro en la estación remota.

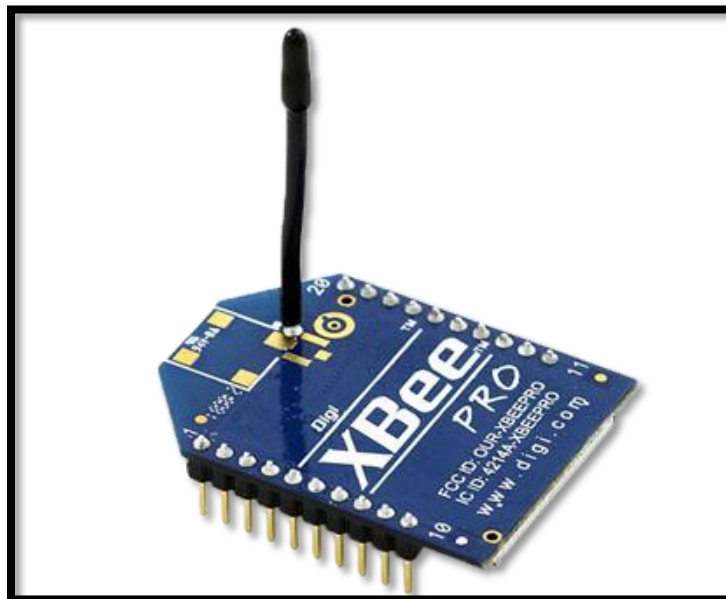


Imagen 23 Modulo RF Xbee

**Posicionamiento:** Modulo GPS. El mismo permite obtener las coordenadas de latitud y longitud del robot en el globo terráqueo. Una correcta implementación de este sensor con un software que integre la aplicación de Google Earth permitirá no solo conocer su ubicación, sino además la posibilidad de planificar trayectorias de recorrido espacial. El modelo de GPS a utilizar resulta ser Ublox Neo 6M.

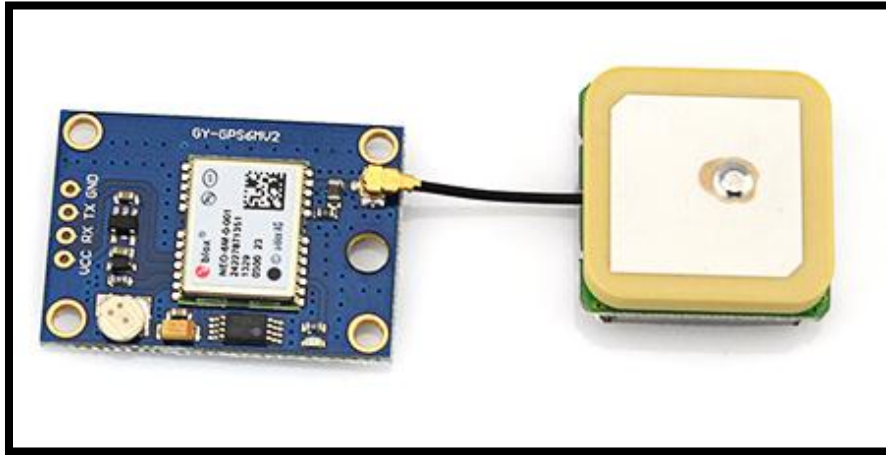


Imagen 24 GPS Neo Ublox 6m

**Orientación:** Para cumplir con el requerimiento de conocer la orientación del robot en el espacio, se propone utilizar el modulo integrado de acelerómetro y giróscopo MPU-6050. Este circuito integrado, posteriormente al tratamiento de las señales adquiridas brinda la orientación angular del mismo respecto a un plano de referencia.

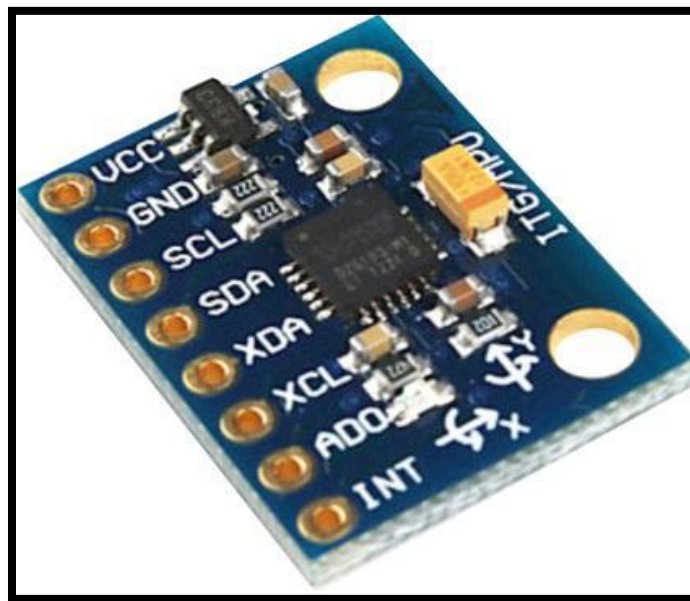
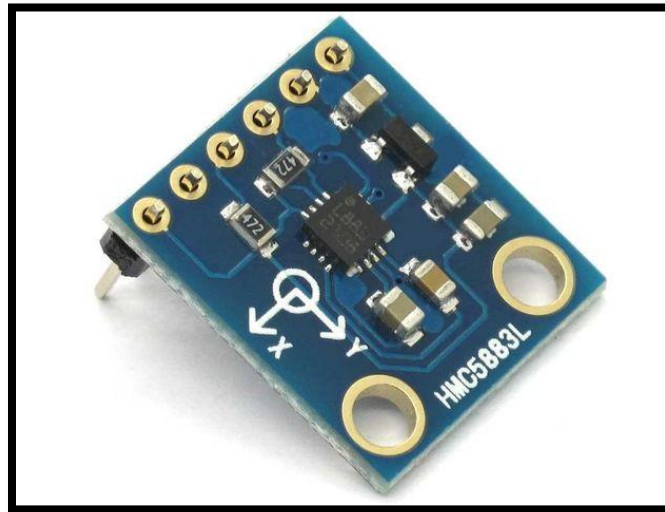


Imagen 25 Acelerómetro MPU6050

Además se propone utilizar un módulo magnetómetro HMC5883L para poder determinar la orientación del hexápodo respecto a los polos magnéticos del globo terrestre. Esta información obtenida en conjunto con la información brindada por el modulo GPS brindan la información necesaria para determinar la ubicación y orientación del robot durante la ejecución de una trayectoria.



**Imagen 26 Modulo magnetómetro HMC5883L**



### Actuadores del robot

Para la construcción del prototipo del hexápodo y para su utilización en ensayos de comportamiento y de adición de otros sistemas, se propone utilizar como actuador de cada articulación de las extremidades, el siguiente servomotor:

Tower Pro Mg996r:



Imagen 27 Servo Digital TowerPro 996r

Especificaciones:

- Peso: 55 g
- Dimensión: 40.7 x 19.7 x 42.9 mm aprox.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)  
Este torque es equivalente a  $11 \text{ kgf} \cdot \text{cm} = 1,07 \text{ Nm}$
- Velocidad de operación: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V) (Sin carga externa)
- Voltaje de operación: 4.8 V a 7.2 V
- Corriente de funcionamiento: 500 mA – 900 mA (6V)
- Rango de temperatura: 0 °C - 55 °C

Dado que en la etapa de simulación dinámica no se superaron valores de  $5 \text{ kgf} \cdot \text{cm}$ , aun incluyendo valores sobredimensionados de peso y torques de fricción. Este actuador posee un margen adecuado en el caso de que se requiera una exigencia dinámica mayor, pero en base a lo establecido en la verificación de esfuerzos estáticos, se deberá asegurar que la configuración de posición de la pata sea tal que minimice el torque aplicado en la tercera articulación, para evitar el desplome del robot.



### Costos de producción

Ítem	Cantidad	Costo unitario	Costo total
Servo digital 996r	18	\$250	\$4500
Placa controladora de servos	1	\$400	\$800
Estructura del robot (Impresión 3D)	1	\$2000	\$2000
Placa central de control	1	\$1000	\$1000
Batería de gel	1	\$500	\$500
Módulo Xbee	2	\$600	\$1200
Xbee Xplorer	1	\$200	\$200
Modulo GPS	1	\$500	\$500
Módulo Acelerómetro	1	\$200	\$200
Módulo Ultrasónico	1	\$80	\$80
Bulonería	1	\$500	\$500
Total			\$11980

## **Conclusiones**

En este proyecto se analiza la dinámica de un robot móvil de seis patas (Hexápodo), mediante los métodos de Newton–Euler y de Euler–Lagrange implementados computacionalmente en el software MATLAB. Los resultados de estos métodos brindan ecuaciones que se simularon en la herramienta Simulink del mismo software.

Los resultados de estas simulaciones permitieron observar los efectos introducidos por el término de fricción viscosa correspondiente a las articulaciones. Posteriormente se adiciona un controlador PID por articulación que cierra el lazo de control compensando estos efectos de perturbación. Finalmente se analizaron los torques requeridos en las articulaciones para el correcto dimensionamiento de los actuadores.

Se destaca el crecimiento y la experiencia en el ámbito académico además de afrontar un proyecto por gusto propio sumado a la investigación sobre el estado actual de los temas teóricos. Para todo esto fue necesario aplicar los conceptos adquiridos durante el transcurso de la cátedra Robótica II.

### **Bibliografía**

- Apuntes de cátedra Robótica II 2016. Facultad de Ingeniería. Universidad Nacional de Cuyo.
- Barrientos, Antonio. Fundamentos de Robótica. 2da edición ed. McGraw Hill 1997.

## ANEXOS

### Implementación computacional de algoritmo Lagrange Euler

```
%Modelo dinámico mediante implementación de algoritmo Lagrange Euler
%Definición de variables simbólicas
syms t g;
syms theta1 theta2 theta3;
syms tita1(t) tita2(t) tita3(t);
syms d_w1 d_w2 d_w3;
syms c1 s1;
syms c2 s2;
syms c3 s3;
syms w1 w2 w3;
syms m1 m2 m3;
syms l1 l2 l3;

c1=cos(theta1);
c2=cos(theta2);
c3=cos(theta3);
g_g=[0 0 -g 0];
s1=sin(theta1);
s2=sin(theta2);
s3=sin(theta3);

w=sym([w1;w2;w3]);
m_m=sym([m1;m2;m3]);
J_j=sym(zeros(4,4,3,1));
thet=sym([theta1;theta2; theta3]);
U=sym(zeros(4,4,3,3));

%Matriz D
D_e=sym(zeros(3));

%Matriz c
c=sym([0;0;0]);

%Distancia centro de masa de cada eslabón
r=sym(zeros(4,1,3));
r1=[l1;0;0;1];
r2=[l2;0;0;1];
r3=[l3;0;0;1];
r(:,1,1)=r1;
r(:,1,2)=r2;
r(:,1,3)=r3;

%Vector Aceleración angular
q_dot_dot=[d_w1;d_w2;d_w3];

%Matrices de transformación D-H
A_01=[c1 -s1 0 0; s1 c1 0 0; 0 0 1 0; 0 0 0 1];
A_12=[c2 0 s2 l1*c2; s2 0 -c2 l1*s2; 0 1 0 0; 0 0 0 1];
A_23=[c3 -s3 0 l2*c3; s3 c3 0 l2*s3; 0 0 1 0; 0 0 0 1];
```

```
A_02=A_01*A_12;
A_03=A_02*A_23;
A=sym(zeros(4,4,3,1));

A(:, :, 1, 1)=A_01;
A(:, :, 1, 2)=A_02;
A(:, :, 1, 3)=A_03;

%%Matrices de Pseudoinercias
J1=[m1*l1^2 0 0 m1*l1; 0 0 0 0;0 0 0 0;m1*l1 0 0 m1];
J2=[m2*l2^2 0 0 m2*l2; 0 0 0 0;0 0 0 0;m2*l2 0 0 m2];
J3=[m3*l3^2 0 0 m3*l3; 0 0 0 0;0 0 0 0;m3*l3 0 0 m3];
%%

J_j(:, :, 1, 1)=J1;
J_j(:, :, 1, 2)=J2;
J_j(:, :, 1, 3)=J3;

for i=1:3
    for j=1:3
        U(:, :, i, j)=diff(A(:, :, 1, i), thet(j));
    end
end

U_u=sym(zeros(4,4,3,3,3));

for i=1:3
    for j=1:3
        for k=1:3
            U_u(:, :, i, j, k)=diff(U(:, :, i, j), thet(k));
        end
    end
end

for i=1:3
    for j=1:3
        for k=max(i, j):3
            D_e(i, j)= D_e(i, j) + trace( U(:, :, k, j)*J_j(:, :, 1, k)*(U(:, :, k, i).') );
        end
    end
end

%%Matrices auxiliares
h=sym(zeros(3,3,3));

for i=1:3
    for k=1:3
        for m=1:3
            for j=max(i, max(k, m)):3
                h(i, k, m) = h(i, k, m) + trace(U_u(:, :, j, k, m)*J_j(:, :, 1, j)*(U(:, :, j, i).') );
            end
        end
    end
end
%Matriz H
```

```
H=sym([0;0;0]);

for i=1:3
    for k=1:3
        for m=1:3
            H(i) = H(i) + h(i,k,m)*w(k)*w(m);
        end
    end
end

for k=1:3
    for j=1:3
        c(k) = c(k) - (m_m(j)* g_g * ( U(:, :, j, k)*r(:, 1, j)));
    end
end

%Modelo Dinámico final
T_LE=sym(D_e*q_dot_dot + H + c)
```

## Implementación computacional de algoritmo Newton Euler

```
%Newton Euler 3dof
%Variables simbólicas
syms g;
syms theta1 theta2 theta3;
syms w1 w2 w3;
syms d_w1 d_w2 d_w3;
syms m1 m2 m3;
syms l1 l2 l3;
syms c1 s1;
syms c2 s2;
syms c3 s3;

%Matrices rotación
%Directas
R_01 = [c1,-s1,0; s1,c1,0; 0,0,1];
R_12 = [c2,0,s2; s2,0,-c2; 0,1,0];
R_23 = [c3,-s3,0; s3,c3,0; 0,0,1];
R_34 = [1,0,0; 0,1,0; 0,0,1];

%Inversas
R_10 = R_01.';%[c1,s1,0; -s1,c1,0; 0,0,1];
R_21 = R_12.';%[c2,s2,0; 0,0,-1; -s2,c2,0];
R_32 = R_23.';%[c3,s3,0; -s3,c3,0; 0,0,1];
R_43 = [1,0,0; 0,1,0; 0,0,1];

%Vectores
%Distancia del centro de masa al sistema de referencia
s_1=[l1 0 0].';
s_2=[l2 0 0].';
s_3=[l3 0 0].';
%Distancia entre sistemas de referencia
p_1=[0;0;0]; %Se considera que S1 y S0 están en la misma posición
p_2=[l1;0;0];
p_3=[l2;0;0];
%Vector de eje Z
z_0=[0;0;1];

%fuerzas externas
f_4=[0;0;0];
n_4=[0;0;0];

%Condiciones iniciales de la base
w_0=[0;0;0];
dw_0=[0;0;0];
v_0=[0;0;g];
a_0=[0;0;0];

%Iteraciones Hacia adelante
%Velocidades angulares
w_1 = sym(R_10*(w_0+z_0*w1));
w_2 = sym(R_21*(w_1+z_0*w2));
```

```
w_3 = sym(R_32*(w_2+z_0*w3));

%Aceleraciones angulares
dw_1 = sym(R_10*(dw_0+z_0*d_w1) + cross(w_0,z_0*w1));
dw_2 = sym(R_21*(dw_1+z_0*d_w2) + cross(w_1,z_0*w2));
dw_3 = sym(R_32*(dw_2+z_0*d_w3) + cross(w_2,z_0*w3));

%Aceleraciones lineales
v_1 = cross(dw_1,p_1) + cross(w_1,cross(w_1,p_1)) + R_10*v_0;
v_2 = cross(dw_2,p_2) + cross(w_2,cross(w_2,p_2)) + R_21*v_1;
v_3 = cross(dw_3,p_3) + cross(w_3,cross(w_3,p_3)) + R_32*v_2;

%Aceleraciones lineales del centro de masa
a_1 = cross(dw_1,s_1) + cross(w_1,cross(w_1,s_1)) + v_1;
a_2 = cross(dw_2,s_2) + cross(w_2,cross(w_2,s_2)) + v_2;
a_3 = cross(dw_3,s_3) + cross(w_3,cross(w_3,s_3)) + v_3;

%Iteraciones hacia dentro
%Fuerzas aplicadas
f_3 = R_34*f_4 + m3*a_3;
f_2 = R_23*f_3 + m2*a_2;
f_1 = R_12*f_2 + m1*a_1;

%Pares aplicados sobre eslabon
n_3 = R_34*(n_4 + cross(R_43*p_3,f_4)) + cross(p_3+s_3,m3*a_3);
n_2 = R_23*[n_3 + cross(R_32*p_2,f_3)] + cross(p_2+s_2,m2*a_2);
n_1 = R_12*[n_2 + cross(R_21*p_1,f_2)] + cross(p_1+s_1,m1*a_1);

%Torques sobre articulación
t_3 = n_3.'*R_32*z_0;
t_2 = n_2.'*R_21*z_0;
t_1 = n_1.'*R_10*z_0;

T_NE=sym([t_1;t_2;t_3])
```



## Código modelo de fricción

```
function [f1,f2,f3] = friction(w1,w2,w3)

w_th=0.01;
F_coulomb=zeros(3);
F_visc=zeros(3);
F_strib=zeros(3);

%Coulomb
offset=0;
coef_coulomb=0.025*27;
F_coulomb(1)=sign(w1)*(coef_coulomb + offset);
F_coulomb(3)=sign(w2)*(coef_coulomb + offset);
F_coulomb(2)=sign(w3)*(coef_coulomb + offset);

%Viscoso
coef_visc=0.025;
F_visc(1)=coef_visc*w1;
F_visc(2)=coef_visc*w2;
F_visc(3)=coef_visc*w3;

%Stribeck
breakaway=0.5;
f_strib=4;
F_strib(1)=(breakaway)*exp(-f_strib*abs(w1))*sign(w1);
F_strib(2)=(breakaway)*exp(-f_strib*abs(w2))*sign(w2);
F_strib(3)=(breakaway)*exp(-f_strib*abs(w3))*sign(w3);

if abs(w1)<=w_th
    f1=(w1/w_th)*(coef_visc*w_th + sign(w1)*F_coulomb(1)+
(breakaway)*exp(-f_strib*w_th));
else
    f1=F_strib(1)+F_coulomb(1)+F_visc(1);
end
if abs(w2)<=w_th
    f2=(w2/w_th)*(coef_visc*w_th + sign(w2)*F_coulomb(2)+
(breakaway)*exp(-f_strib*w_th));
else
    f2=F_strib(2)+F_coulomb(2)+F_visc(2);
end
if abs(w3)<=w_th
    f3=(w3/w_th)*(coef_visc*w_th + sign(w3)*F_coulomb(3)+
(breakaway)*exp(-f_strib*w_th));
else
    f3=F_strib(3)+F_coulomb(3)+F_visc(3);
end
```

### Planilla de división de tareas

Titulo	Descripción	Responsable
Modelo Dinámico: Método de Lagrange	Obtención del modelo dinámico del hexápodo mediante la implementación computacional del algoritmo de Lagrange Euler.	Ullmann
Modelo Dinámico: Método de Euler	Obtención del modelo dinámico del hexápodo mediante la implementación computacional del algoritmo de Newton Euler.	Sesto
Simulación Modelo dinámico	Análisis del comportamiento de las ecuaciones dinámicas obtenidas previamente.	Ullmann
Controlador PID	Adicionado de un controlador PID implementado en Simulink para el cumplimiento de consignas de posición de una extremidad del robot.	Sesto
Esfuerzos estáticos	Análisis de los momentos estáticos producidos en las articulaciones de una extremidad del robot para verificación de los esfuerzos requeridos en los actuadores.	Ullmann
Sensores y actuadores	Descripción de elementos a utilizar para la construcción del robot, junto con sus características más representativas.	Sesto (60%) Ullmann (40%)
Método de control del robot	Descripción del tipo de controlador que gobierna la coordinación de movimiento de extremidades cuando se le introducen consignas de desplazamiento al robot.	Ullmann
Costos de producción	Descripción de costos monetarios de los diversos componentes para la construcción del robot. Los mismos fueron obtenidos de fuentes online y de negocios locales.	Sesto