

**PASSIVE DYNAMIC SWING LOCOMOTION FOR A 6DOF  
COPLANAR ARTICULATED CHAIN**

**JUAN SEBASTIÁN IGUA ROBLES**

**ANDRÉS FELIPE TRUJILLO BUITRAGO**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE ELECTRÓNICA  
BOGOTÁ D.C.  
27 DE MAYO DE 2011**

**PASSIVE DYNAMIC SWING LOCOMOTION FOR A 6DOF  
COPLANAR ARTICULATED CHAIN**

**Trabajo de grado presentado para obtener el título de Ingeniero Electrónico**

**Trabajo de Investigación  
Carrera de Ingeniería Electrónica  
Director  
Ing. KAMILO ANDRÉS MELO B, M.Sc.  
Ing. CARLOS ALBERTO PARRA Ph.D.**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE ELECTRÓNICA  
BOGOTÁ D.C.  
27 DE MAYO DE 2011**

**PONTIFICIA UNIVERSIDAD JAVERIANA**  
**FACULTAD DE INGENIERÍA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA**



**RECTOR MAGNÍFICO**

**R.P. JOAQUÍN SÁNCHEZ S.J.**

**DECANO ACADÉMICO**

**ING. FRANCISCO J. REBOLLEDO**

**DECANO DEL MEDIO UNIVERSITARIO**

**R.P. SERGIO BERNAL S.J.**

**DIRECTOR DE CARRERA**

**ING. JUAN MANUEL CRUZ**

**DIRECTOR DEL PROYECTO**

**ING. KAMILO MELO**

## NOTA DE ADVERTENCIA

"La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia".

Artículo 23 de la Resolución No. 13, del 6 de julio de 1946, por la cual se reglamenta lo concerniente a Tesis y Exámenes de Grado en la Pontificia Universidad Javeriana.

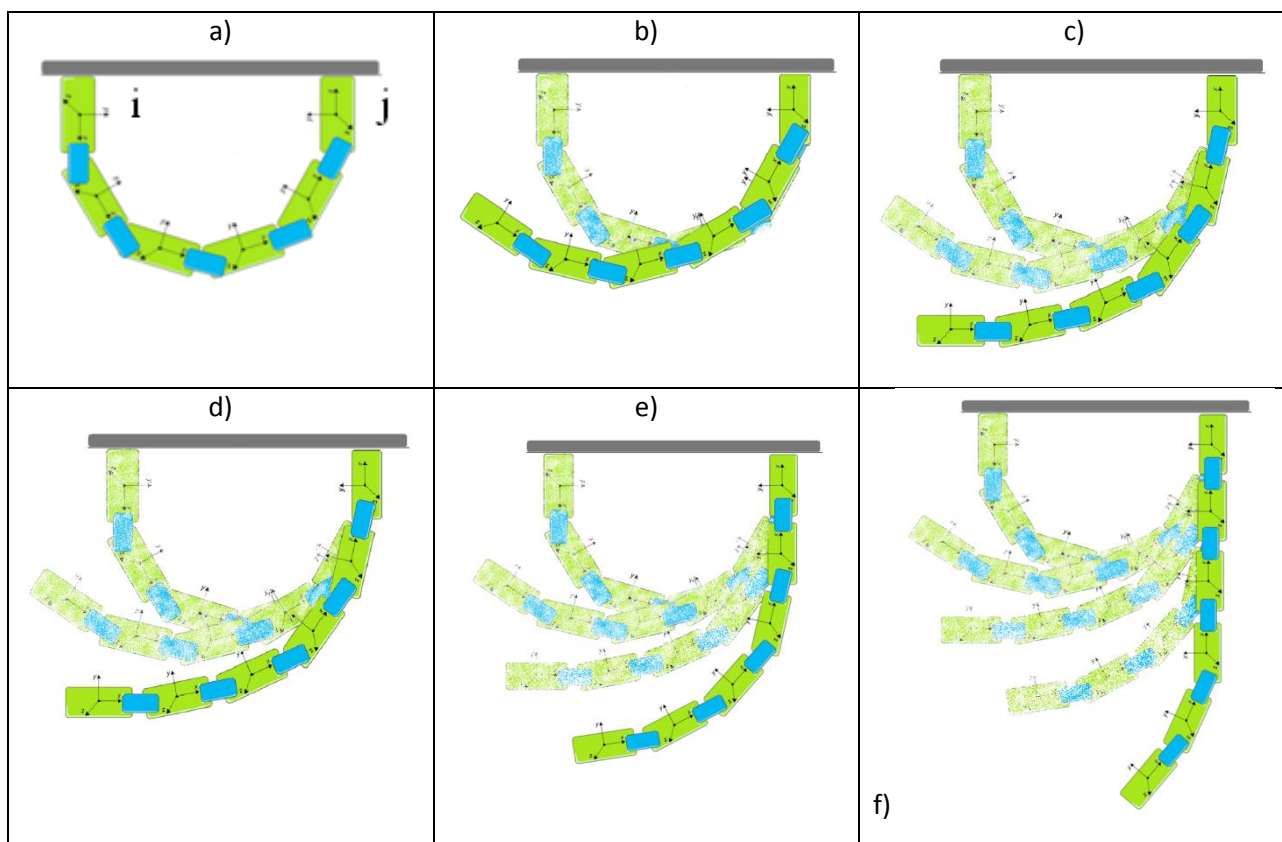
## Contenido

1. INTRODUCCIÓN .....	- 6 -
1.2 Objetivos .....	- 8 -
1.2.1 Objetivo General .....	- 8 -
1.2.2 Objetivos Específicos .....	- 8 -
2. MARCO TEÓRICO .....	- 9 -
2.1 Cinemática.....	- 9 -
2.1.1 Transformaciones homogéneas .....	- 9 -
2.1.2 Centro de gravedad.....	- 11 -
2.2 Dinámica .....	- 11 -
2.2.1 Formulación Euler-Lagrange .....	- 11 -
2.2.2 Tensor de inercia .....	- 12 -
2.3 Ecuación catenaria.....	- 13 -
3. DESARROLLO .....	- 16 -
3.1 Especificaciones mecánicas de la cadena articulada.....	- 16 -
3.2 Representación en variables de estado .....	- 17 -
3.3 Modelo en SimMechanics™ .....	- 25 -
3.3.1 Validación del modelo .....	- 30 -
3.4 Posición inicial de mínima energía de la cadena articulada.....	- 33 -
3.5 Variables de junta y torques de entrada .....	- 40 -
4. CONCLUSIONES .....	- 43 -
5. BIBLIOGRAFÍA.....	- 45 -
6. ANEXOS 1: ALGORITMOS DE MATLAB .....	- 47 -
7. ANEXOS 2: GRÁFICAS Y TABLAS .....	- 59 -

# 1. INTRODUCCIÓN

Las transiciones entre terrenos regulares a irregulares, presentan ciertos inconvenientes para robots móviles, ya que para cada uno su forma de locomoción varía respecto al ambiente en que se mueve. Estas formas o estrategias de locomoción no solo dependen del medio en donde se desplazan sino que también de la forma en que el robot está constituido mecánicamente. En el caso de un robot constituido como una cadena modular o también llamado cadena serial articulada como el que pretende simular este proyecto, se puede pensar en un ambiente de locomoción que implique que la cadena robótica ejecute un movimiento de “swing”.

Swing es el movimiento por el cual la cadena articulada que está sujeta de los extremos *i* y *j* **Figura 1a**, se desprende del extremo *i* y se balancea o columpia (**Figura 1b-g**) para llegar al punto de agarre *k*, ver **Figura 1h**.



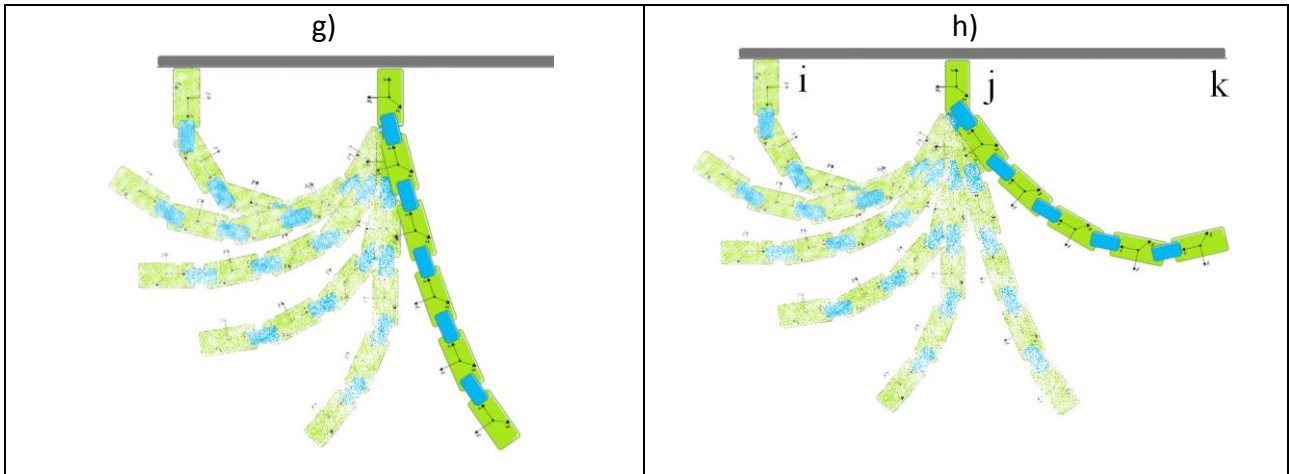


Figura 1. Secuencia del movimiento “swing” de la cadena articulada

En el pasado se han trabajado temáticas relacionadas, donde se resalta la investigación que realizaron las Universidades de Nagoya y Tsukuba en Japón [1][2][3][4][5] donde diseñaron e implementaron el control de velocidad angular del movimiento de swing (término que Toshio Fukuda acuñó para representar el movimiento de este robot). El objetivo de este control era lograr la continuidad del desplazamiento del mismo imitando el movimiento biológico de los simios. Este movimiento se llama *brachiation* [4][5].

Lo eficiente de esta locomoción se puede evidenciar en el gasto de energía; cuando la cadena suelta un punto de agarre la energía potencial es aprovechada para generar un cambio en la energía cinética, disminuyendo el trabajo que tienen que hacer los actuadores para alcanzar el siguiente punto de agarre.

Este proyecto consiste en implementar un modelo práctico para la representación de una cadena articulada, con él se podrá analizar el movimiento de la cadena mediante simulaciones que representen la cinemática y dinámica del sistema, logrando establecer un entorno para generar trayectorias que un robot real podría ejecutar.

En el proyecto primero se determinará el modelo cinemático de la cadena articulada y se conocerá la posición de cada módulo y del sistema en general referidos a un marco de referencia inercial. Luego se determinará el modelo dinámico analítico escalable para después encontrar las matrices del modelo en espacio de estados que corresponde al mecanismo. Los modelos anteriormente nombrados serán implementados y simulados en MATLAB®. Posteriormente, se desarrollará en el “toolbox” SimMechanics™ el modelo mecánico de la cadena articulada, para comparar y validar el modelo analítico de la dinámica del robot antes descrito. En seguida se determinarán las posiciones iniciales de la cadena articulada suspendida, para la realización del movimiento “swing” con el mínimo uso de energía. Por medio de la formulación matemática llamada Catenaria se encontrarán varias posiciones iniciales para el cual el sistema se encuentra en estabilidad estacionaria. A partir de cada una de estas posiciones iniciales se libera la cadena de un extremo y se determina la posición máxima de su último módulo y del centro de gravedad de la cadena articulada. La posición inicial que genere el movimiento para el cual la posición final en y del último módulo (y la posición del centro de gravedad) sea la más alta, es la configuración que requiere menor trabajo mecánico para realizar el movimiento, sin contar las pérdidas de energía, basándose en un modelo ideal. Una vez encontrada la posición inicial de la cadena articulada, usando SimMechanics™, se determinan los torques necesarios para completar todo el movimiento de “swing”,

donde los ángulos iniciales para este desarrollo son los ángulos finales obtenidos de la configuración catenaria que fue escogida previamente.

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

Determinar el modelo dinámico de una cadena articulada coplanar de 6DOF que represente un movimiento “swing”

### **1.2.2 Objetivos Específicos**

1. Determinar un modelo en espacio de estados de una cadena articulada de seis módulos que conforman 6DOF.
2. Validar este modelo, al compararlo con un modelo de parámetros físicos implementado en la herramienta SimMechanics™ de MATLAB®.
3. Especificar los puntos de agarre y la trayectoria de los eslabones, de la cadena articulada en simulación, en donde al ejecutar el movimiento de “swing” se logre alcanzar el punto más alto posible, con el extremo liberado del robot.
4. Determinar usando dinámica inversa, las variables de posición y torque de cada una de las juntas entre los eslabones de la cadena articulada, basándose en el modelo dinámico al ejecutar el movimiento de “swing” especificado anteriormente.



## 2. MARCO TEÓRICO

### 2.1 Cinemática

Para analizar cualquier movimiento que realiza una cadena articulada de  $n$  módulos se plantea un modelo cinemático, donde se obtienen ecuaciones que relacionan la posición de cada módulo  $i$ , siendo  $i$  el módulo específico del que se está hablando, con respecto a un eje coordenado que se sujeta a la base del robot. A este eje coordenado se le conoce como marco de referencia inercial.

#### 2.1.1 Transformaciones homogéneas

Para representar la posición de un marco de referencia  $o_i x_i y_i z_i$  con respecto a otro marco de referencia  $o_{i+1} x_{i+1} y_{i+1} z_{i+1}$  y siendo congruente con la convención Denavit-Hartenberg [6] se recurre a la matriz de transformación  $H$ , la cual se define de la siguiente manera:

$$H_{i+1}^i = A_i A_{i+1} = \begin{bmatrix} R_{i+1}^i & o_{i+1}^i \\ 0 & 1 \end{bmatrix} \quad (1)$$

Donde  $A_i$  es una matriz homogénea de transformación que representa la posición y orientación de  $o_i x_i y_i z_i$  con respecto de  $o_{i+1} x_{i+1} y_{i+1} z_{i+1}$ .  $R_{i+1}^i$  es la matriz de rotación que define la orientación de  $o_{i+1} x_{i+1} y_{i+1} z_{i+1}$  con respecto a  $o_i x_i y_i z_i$ , y  $o_{i+1}^i$  es un vector que define la posición del origen del marco  $i + 1$  con respecto al marco  $i$ .

Para simplificar el análisis cinemático directo de la cadena articulada se utiliza la convención de Denavit-Hartenberg. La convención de DH consiste en plantear la ubicación de los marcos de referencia de tal forma que las matrices de transformación solo dependan de  $\theta_i$ . Dado que para este caso solo se usan juntas rotacionales.

En esta convención cada matriz  $A_i$  es representada por el producto de cuatro transformaciones básicas: Una rotación en el eje  $z$  con un ángulo de  $\theta_i$  seguida por una traslación en el eje  $z$  con una distancia de  $d_i$  seguida por otra traslación en el eje  $x$  con una distancia de  $a_i$  y finalmente seguida por una rotación en el eje  $x$  con un ángulo de  $\alpha_i$ .

$$A_i = \text{Rot}_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} \text{Rot}_{x,\alpha_i}$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Donde los cuatro parámetros asociados a la junta  $i$  son  $\theta_i, a_i, d_i, \alpha_i$ . Cada uno representa respectivamente el ángulo de la junta, el largo del módulo, el "offset" del módulo (definido como el desplazamiento a lo largo del eje z) y el "twist" del módulo (definido como la orientación con respecto al eje x). La matriz  $A_i$  depende solo de  $\theta_i$ , siendo los demás parámetros constantes.

Así como la posición de un módulo  $i$  con respecto al marco de referencia inercial se define a través de una matriz de transformación. La velocidad angular y lineal se definen a través del jacobiano.

Entonces la velocidad de un módulo  $i$  con respecto al marco de referencia inercial está definida por la matriz

$$\xi_i = J_i \dot{q} \quad (3)$$

Donde  $\dot{q}$  representa las velocidades angulares de las juntas. Además se puede expresar de la siguiente manera

$$\xi_i = \begin{bmatrix} v_i^0 \\ \omega_i^0 \end{bmatrix} \quad (4)$$

Para el vector de las velocidades lineales  $v_i^0$  y angulares  $\omega_i^0$  vistas desde el marco de referencia inercial en este caso el 0. Por lo tanto el jacobiano para un módulo  $i$  sería

$$J_i = \begin{bmatrix} J_{v_i} \\ J_{\omega_i} \end{bmatrix} \quad (5)$$

En la (5)  $J_{v_i}$  se refiere al jacobiano de velocidad lineal y  $J_{\omega_i}$  al jacobiano de velocidad angular.

El jacobiano para juntas rotacionales, de acuerdo con [6] y siendo  $n$  el número total de módulos de la cadena articulada, se define como

$$J_i(q) = \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix}_{3 \times n} \quad (6)$$

Donde

$z_{i-1}$  es la componente rotacional de la matriz de transformación del marco  $i - 1$  al marco inercial.

$(o_n - o_{i-1})$  Representa la diferencia entre la posición del módulo  $n$  y la posición del módulo  $i - 1$ .

La (5) se reescribe como

$$v_i = J_{v_i}(q) \dot{q} \quad \omega_i = J_{\omega_i}(q) \dot{q} \quad (7)$$

Siendo  $v_i$  la velocidad lineal de un módulo  $i$  con respecto al marco de referencia inercial y  $\omega_i$  la velocidad angular de un módulo  $i$  con respecto al marco de referencia inercial.

## 2.1.2 Centro de gravedad

El centro de gravedad es el punto donde se encuentra la media ponderada de las masas de todas las partículas de un cuerpo [7]. El vector de posición del centro de gravedad con respecto al marco inercial se define como:

$$\vec{r}_{cm} = \frac{\sum_i m_i \vec{r}_i}{\sum_i m_i} \quad (8)$$

Donde

$m_i$  Es la masa del módulo  $i$

$\vec{r}_i$  Es el vector de posición del centro de gravedad del módulo  $i$

## 2.2 Dinámica

En esta sección se tienen en cuenta las fuerzas y los torques que influyen sobre la cadena articulada para plantear las ecuaciones de movimiento. Las ecuaciones de movimiento o ecuaciones dinámicas describen la relación que existe entre las fuerzas, que actúan sobre el sistema, y el movimiento del robot [6].

### 2.2.1 Formulación Euler-Lagrange

La forma matricial de las ecuaciones de Euler-Lagrange es

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (9)$$

$D(q)$  es la matriz de inercia para un numero  $n$  de módulos y se define como

$$D(q) = \sum_{i=1}^n [m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)] \quad (10)$$

$C(q, \dot{q})$  es una matriz conformada por  $q, \dot{q}$  y los símbolos de Christoffel. Estos se definen

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \quad (11)$$

Donde  $d_{kj}$ ,  $d_{ki}$  y  $d_{ij}$  son elementos de la matriz  $D(q)$ .

$g(q)$  es la matriz de las derivadas parciales de las energías potenciales de cada módulo con respecto a las variables de junta

$$\phi_k = \frac{\partial P}{\partial q_k} \quad (12)$$

Donde  $P$  es la energía potencial total ( $P_1 + P_2 \dots + P_n$ ). Para determinar la energía potencial de un módulo  $i$  se emplea la siguiente ecuación

$$P_i = g^T r_{ci} m_i \quad (13)$$

Donde

$r_{ci}$  es el vector de posición del centro de gravedad del módulo  $i$ .

$m_i$  es la masa del módulo  $i$ .

Y  $\tau$  es la matriz que contiene los torques que influyen sobre cada módulo por las fuerzas que actúan sobre ellos. A partir de la forma general (9) se reescriben las ecuaciones de Euler-Lagrange para llegar a

$$\sum_i d_{kj}(q) \ddot{q}_j + \sum_{i,j} c_{ijk}(q) \dot{q}_i \dot{q}_j + \phi_k(q) = \tau_k \quad k = 1, \dots, n \quad (14)$$

A partir de (14) se obtiene un sistema de  $n$  ecuaciones y  $n$  incógnitas, siendo las segundas derivadas de las variables de junta  $\ddot{q}$  las incógnitas del sistema. Al solucionar este sistema de ecuaciones se obtiene la ecuación de estado de la cadena articulada.

$$\dot{x} = Ax + B\tau \quad (15)$$

Donde  $x$  es la matriz de las variables de estado en función a las variables de las juntas y sus derivadas.

### 2.2.2 Tensor de inercia

El tensor de inercia describe la distribución de masa que existe en un cuerpo siendo una generalización de todos los momentos de inercia del cuerpo con respecto a los ejes coordenados. Éste se define como

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (16)$$

Donde

$$\begin{aligned} I_{xx} &= \iiint (y^2 + z^2) \rho(x, y, z) dx dy dz \\ I_{yy} &= \iiint (x^2 + z^2) \rho(x, y, z) dx dy dz \\ I_{zz} &= \iiint (x^2 + y^2) \rho(x, y, z) dx dy dz \\ I_{xy} &= I_{yx} = - \iiint x y \rho(x, y, z) dx dy dz \\ I_{xz} &= I_{zx} = - \iiint x z \rho(x, y, z) dx dy dz \\ I_{yz} &= I_{zy} = - \iiint y z \rho(x, y, z) dx dy dz \end{aligned}$$

Siendo  $\rho(x, y, z)$  la función que representa la densidad del cuerpo en el espacio. Los términos que se encuentran en la diagonal del tensor de inercia se llaman momentos de inercia principales. Los demás

términos se llaman productos cruz de inercia. Si la distribución de masa del cuerpo es simétrica con respecto al marco de referencia del mismo (en otras palabras  $\rho(x,y,z)$  es uniforme), los términos productos cruz de inercia son cero [6].

### 2.3 Ecuación catenaria

Se considera un cable suspendido en el aire con sus extremos estacionarios ubicados en la misma línea horizontal. Donde el cable posee una longitud  $l$  y una densidad lineal  $\mu$  (Masa por unidad de longitud).

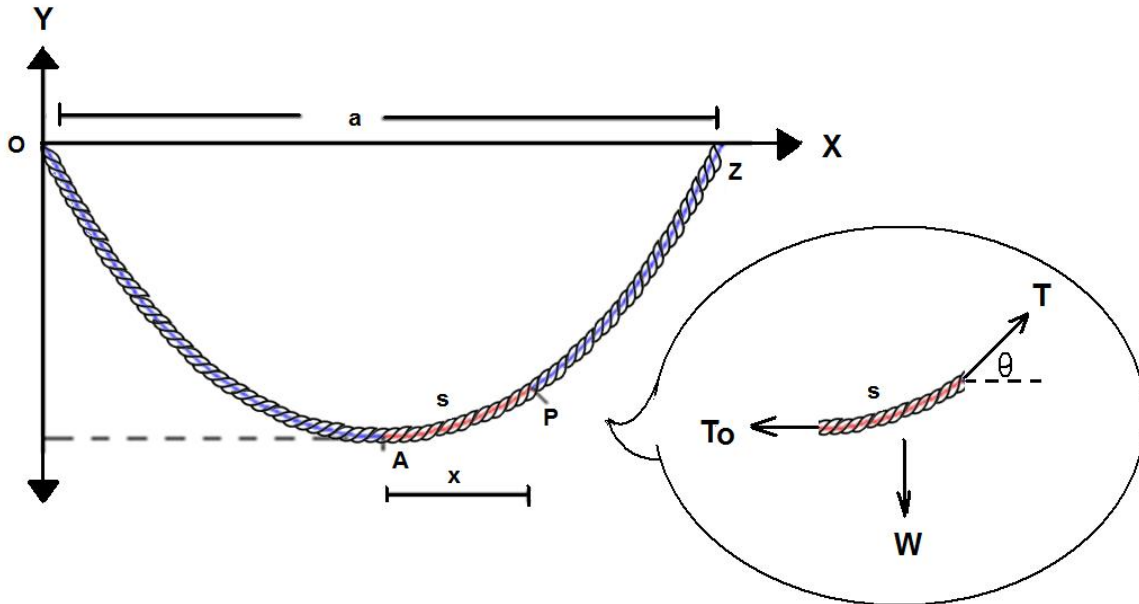


Figura 2. Cable suspendido. [8]

De acuerdo con el diagrama de cuerpo libre del cable, habría tres fuerzas que intervienen en el intervalo AP. El peso  $W$  que iría en dirección  $-y$ , la tensión  $T_0$  que ejerce el intervalo OA y la tensión  $T$  que ejerce el intervalo ZP.

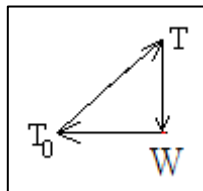


Figura 3. Sumatoria de fuerzas.

Por la figura 25 se entiende que la sumatoria de fuerzas da cero y que la pendiente en el punto P es igual a  $W/T_0$ . Entonces se tiene la siguiente relación:

$$\text{Pendiente en P} = \frac{W}{T_0} \tag{17}$$

El peso es proporcional al largo del cable en la sección AP. Ahora se define la altura del cable como la variable  $y$  y la densidad se define  $\mu$  (Densidad de masa por longitud). Donde  $s$  es la longitud de la cuerda en la sección AP. Entonces el peso en esa sección sería igual a  $W = s \cdot \mu$ . Por otra parte se conoce que la formula de la longitud de una curva en un determinado intervalo  $(a,b)$  se expresa de la siguiente forma [9]:

$$\text{Longitud de la curva en } (a,b) = \int_a^b \sqrt{\left(\frac{dy}{dx}\right)^2 + 1} dx \quad (18)$$

Se aplica la ecuación anterior al modelo de la cuerda y como resultado queda la siguiente ecuación:

$$s = \int_0^x \sqrt{\left(\frac{dy}{dx}\right)^2 + 1} dx \quad (19)$$

Adicionalmente se conoce que la derivada de  $y$  es la pendiente en el punto P.

$$\frac{dy}{dx} = \frac{W}{T_0} = \frac{s \cdot \mu}{T_0}$$

$$s = \frac{T_0}{\mu} \frac{dy}{dx} \quad (20)$$

Se reemplaza la ecuación 2 en la ecuación 1 y se obtiene lo siguiente:

$$\frac{T_0}{\mu} \frac{dy}{dx} = \int_0^x \sqrt{\left(\frac{dy}{dt}\right)^2 + 1} dt$$

$$\frac{dy}{dx} = \frac{\mu}{T_0} \int_0^x \sqrt{\left(\frac{dy}{dt}\right)^2 + 1} dt$$

Se aplica la derivada con respecto a  $x$  en ambos lados:

$$\frac{d^2y}{dx^2} = \frac{\mu}{T_0} \sqrt{\left(\frac{dy}{dt}\right)^2 + 1} \Big|_x$$

$$\frac{d^2y}{dx^2} = \frac{\mu}{T_0} \sqrt{\left(\frac{dy}{dx}\right)^2 + 1}$$

$$y'' = \frac{\mu}{T_0} \sqrt{(y')^2 + 1} \quad (21)$$

Donde la condición inicial para satisfacer la ecuación es  $y(0)=0$ . Por la tanto la ecuación  $y(x)$  se expresa de la siguiente forma:

$$y = \frac{T_0}{\mu} \cosh\left(\frac{\mu}{T_0} x\right) - \frac{T_0}{\mu}$$

$$y = c \cdot \cosh\left(\frac{x}{c}\right) - c \quad (22)$$

### 3. DESARROLLO

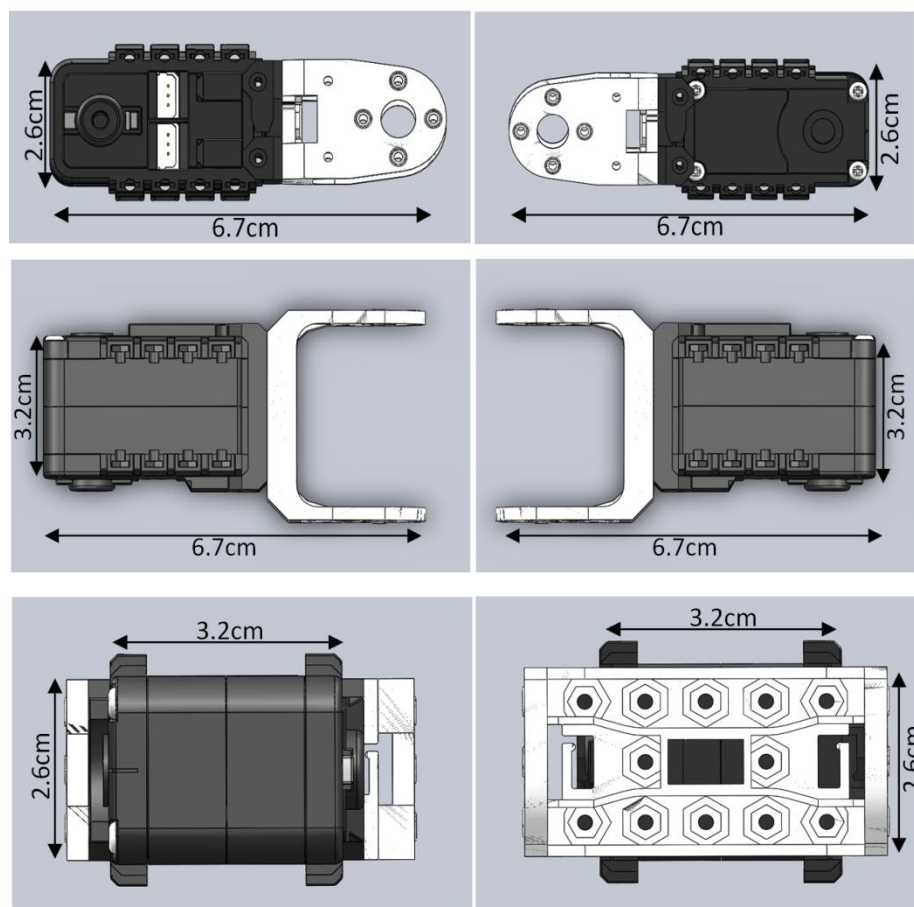
#### 3.1 Especificaciones mecánicas de la cadena articulada

En el desarrollo del trabajo de grado presente, las especificaciones físicas para la cadena articulada fueron obtenidas de la plataforma robótica Bioloid de ROBOTIS. Donde los servos a utilizar son DYNAMIXEL AX-12 [10].

Donde las especificaciones son las siguientes:

Módulo (H x W x D) = 2.6cm x 6.7cm x 3.2cm

Peso = 55g



**Figura 4.** CAD de los módulos de la cadena articulada y sus respectivas dimensiones (Vista frontal, superior y lateral) [10];**Error! No se encuentra el origen de la referencia.**



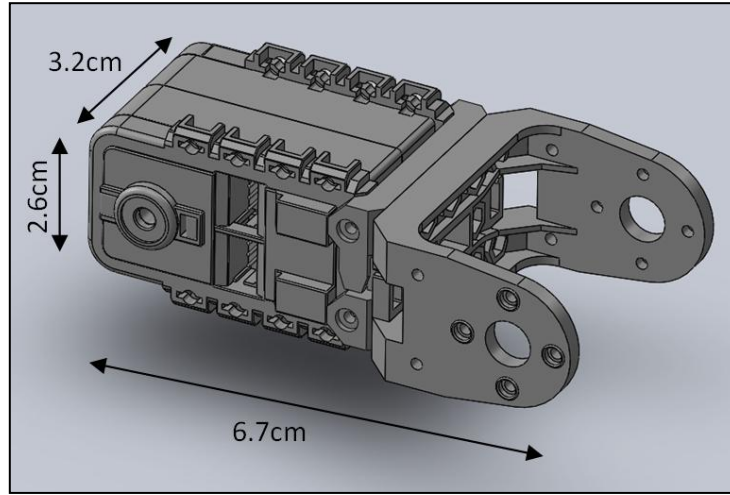


Figura 5. CAD del un módulo y sus respectivas dimensiones

### 3.2 Representación en variables de estado

Inicialmente se desarrollará el modelo dinámico para dos módulos. Con este se busca analizar y entender el procedimiento de la obtención del mismo. Para luego extrapolarlo para seis módulos encontrando un algoritmo o patrón de la extrapolación e implementarla en un código de programación.

Por medio de la (14) se encuentran las ecuaciones de Euler-Lagrange para dos módulos, el desarrollo se encuentra en [6].

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{121}\dot{q}_1\dot{q}_2 + c_{211}\dot{q}_1\dot{q}_2 + c_{221}\dot{q}_2^2 + \phi_1 = \tau_1 \quad (23)$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{112}\dot{q}_1^2 + \phi_2 = \tau_2 \quad (24)$$

Donde  $q_1$  y  $q_2$  son las coordenadas generalizadas del sistema.

Usando MATLAB® se desarrolla un código (ANEXOS 1 Ecu. A1-1) que genera las ecuaciones de Euler-Lagrange para la cadena articulada de 2 módulos y además las gráficas de las variables de junta [11]. A partir de estas ecuaciones se plantea la matriz de espacio de estados para esta configuración.

Las variables de estado son

$$x_1 = q_1 \quad x_2 = \dot{q}_1 \quad x_3 = q_2 \quad x_4 = \dot{q}_2 \quad (25)$$

Luego la matriz de estado es de la forma

$$\dot{x} = Ax + B\tau \quad (26)$$

Donde  $\tau$  son los torques de entrada del sistema.

Mediante el código que se encuentra en ANEXOS 1. Ecu. A1-1 se pueden obtener las funciones  $\dot{x}_2$  y  $\dot{x}_4$ . Estas funciones no se ilustran a continuación, ya que el resultado es un poco extenso y engorroso, por

ser dependiente de variables simbólicas. Por esta razón se muestra el código para que el lector pueda corroborar lo anterior.

Con el análisis de 2 módulos se pueden deducir las expresiones para una cadena articulada de n módulos a partir de la forma generalizada de las ecuaciones de Euler-Lagrange (14). De esta ecuación solo hace falta deducir una expresión que simplifique la obtención de  $D(q)$ . Donde se tiene presente que para la energía cinética de la cadena articulada el Jacobiano se emplea en el marco de referencia del centro de masa de cada módulo. A partir de este punto, se realizará el análisis del modelo dinámico para n módulos para una cadena articulada co-planar.

Para comenzar, se calcula el tensor de inercia para los módulos, teniendo como referencia los ejes en SimMechanics™, con la (16). Por lo tanto para el primer módulo se tiene, como esta fijo al mundo se trata diferente a los demás [12]:

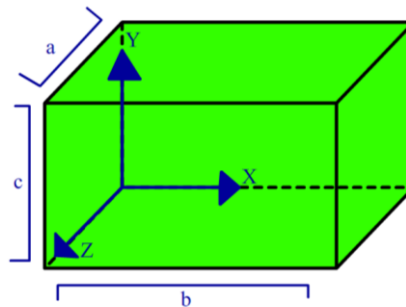


Figura 6. Modelo para el cálculo del tensor de inercia del primer módulo.

Los términos en la diagonal del tensor de inercia son

$$I_{xx} = \frac{m(a^2 + c^2)}{3}$$

$$I_{yy} = \frac{m(a^2 + b^2)}{3}$$

$$I_{zz} = \frac{m(b^2 + c^2)}{3}$$

Luego el tensor de inercia para el primer módulo es

$$I_1 = \begin{bmatrix} \frac{m(a^2 + c^2)}{3} & \frac{-mbc}{4} & \frac{-mab}{4} \\ \frac{-mbc}{4} & \frac{m(a^2 + b^2)}{3} & \frac{-mac}{4} \\ \frac{-mab}{4} & \frac{-mac}{4} & \frac{m(b^2 + c^2)}{3} \end{bmatrix}$$

Donde  $a=3.2\text{cm}$ ,  $b=6.7\text{cm}$  y  $c=2.6\text{cm}$ .

Para los demás módulos se tiene

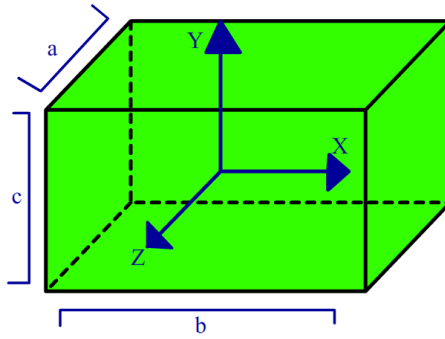


Figura 7. Modelo para el cálculo del tensor de inercia de los otros módulos.

$$I_{xx} = \frac{m(a^2 + c^2)}{12}$$

$$I_{yy} = \frac{m(a^2 + b^2)}{12}$$

$$I_{zz} = \frac{m(b^2 + c^2)}{12}$$

Luego el tensor de inercia es

$$I_{2,3,4,5,6} = \begin{bmatrix} \frac{m(a^2 + c^2)}{12} & 0 & 0 \\ 0 & \frac{m(a^2 + b^2)}{12} & 0 \\ 0 & 0 & \frac{m(b^2 + c^2)}{12} \end{bmatrix}$$

Para aplicarlo al código de la ecuación de estado el único cambio que se realiza es  $I_{yy} = I_{xx}$ , debido a que el eje de SimMechanics™ es el eje x en el código.

Después de encontrar los tensores de inercia cabe comentar que por simplicidad de este manipulador implementado (cadena articulada coplanaria) se reduce la complejidad de la obtención de la matriz  $D(q)$ . Ya que las velocidades angulares son

$$w_1 = \dot{q}_1 k, w_2 = (\dot{q}_1 + \dot{q}_2)k, \dots, w_n = (\dot{q}_1 + \dot{q}_2 + \dots + \dot{q}_n)k$$

Por lo que  $w_i$  está alineado al eje k, se tiene que el producto de  $w_i^T I_i w_i$  se reduce al  $I_{zz}$  para cada módulo  $i$ . Entonces de ahora en adelante  $I_{zz}$  tomará la anotación  $I_i$

Entonces la matriz de inercia generalizada es

$$D(q)_n = \sum_{i=1}^n m_i J_{vci}^T J_{vci} + \begin{bmatrix} I_1 + I_2 + \dots + I_n & I_2 + \dots + I_n & \dots & I_n \\ I_2 + \dots + I_n & I_2 + \dots + I_n & \dots & I_n \\ \vdots & \vdots & \ddots & \vdots \\ I_n & I_n & I_n & I_n \end{bmatrix}_{n \times n} \quad (27)$$

Donde  $J_{vci}$  es el jacobiano de velocidad lineal del centro de masa que se calcula usando (7)

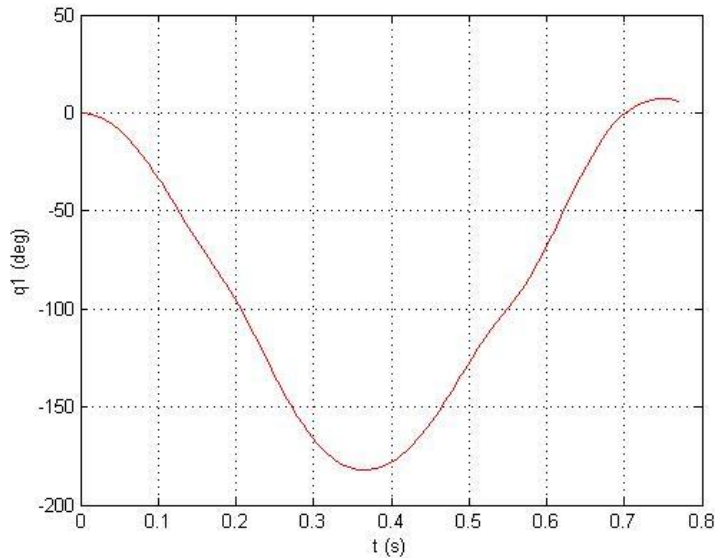
Para el sistema de 6 módulos las variables de estado son

$$\begin{aligned} x_1 &= q_1 & x_2 &= \dot{q}_1 & x_3 &= q_2 & x_4 &= \dot{q}_2 & x_5 &= q_3 & x_6 &= \dot{q}_3 \\ x_7 &= q_4 & x_8 &= \dot{q}_4 & x_9 &= q_5 & x_{10} &= \dot{q}_5 & x_{11} &= q_6 & x_{12} &= \dot{q}_6 \end{aligned} \quad (28)$$

Y la matriz de estados de la configuración de 6 módulos es de la misma forma que en (26).

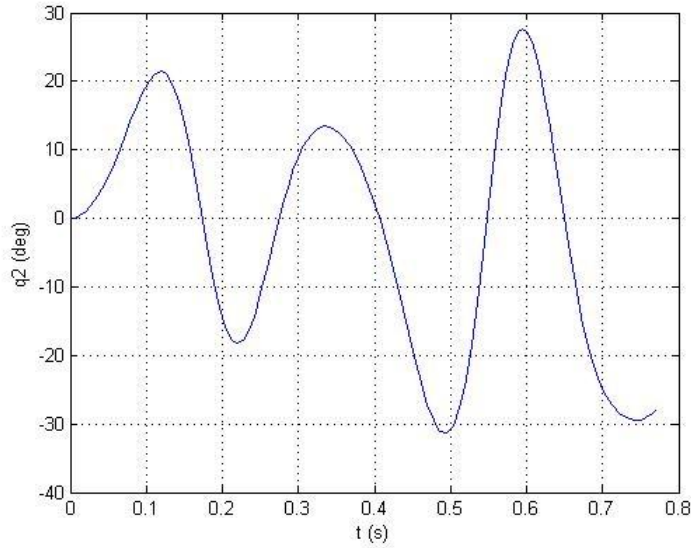
Mediante el código que se encuentra en ANEXOS 1 **Ecu. A1-5** se pueden obtener las funciones  $\dot{x}_2, \dot{x}_4, \dot{x}_6, \dot{x}_8, \dot{x}_{10}$  y  $\dot{x}_{12}$ . Que representan las ecuaciones de  $\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, \ddot{q}_4, \ddot{q}_5$  y  $\ddot{q}_6$  respectivamente.

Para comenzar se especifican las condiciones iniciales de las variables de junta en  $q_{10} = 0^\circ$  y  $q_{20} = 0^\circ$  para el código de la cadena articulada de 2 módulos. El tiempo de simulación es de 0.77 segundos por que en esta ventana de tiempo se puede observar el movimiento de swing completo, y también porque después de este tiempo se obtienen movimientos complejos y muy diferentes al movimiento que se está analizando en este trabajo debido a que en la simulación no está restringido el movimiento de la junta a  $\mp 90^\circ$  o a  $\mp 150^\circ$  como en la practica la mecánica de la estructura de los módulos o servomotores si lo está.



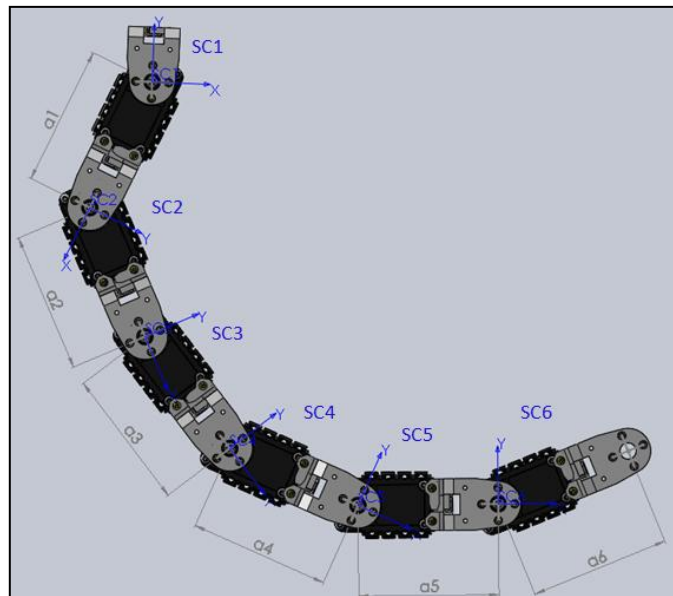
**Figura 8.** Variable de junta  $q_1$  de la cadena de 2 módulos

En la **Figura 8** se observa el movimiento sinusoidal del primer módulo, empieza en  $0^\circ$  y luego desciende y asciende hasta  $-180^\circ$  para luego volver a la posición inicial de  $0^\circ$  en  $t = 0.7$  segundos aproximadamente.



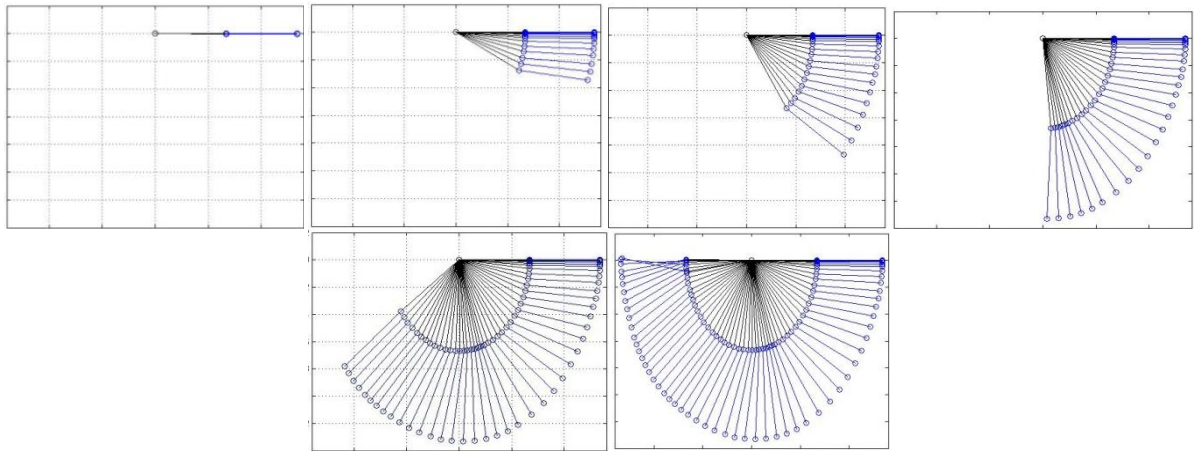
**Figura 9.** Variable de junta  $q_2$  de la cadena de 2 módulos (masa=55g y largo=6,7cm)

El marco de referencia desde el cual se mide  $q_2$  es el del módulo 1 (**SC1, Figura 10**). En la **Figura 9** se observa con la diferencia de amplitudes de  $q_2$ , como el módulo 2 tiene dependencia del módulo 1. Cuando el módulo 1 desciende el módulo 2 es afectado, ya que el extremo está sujeto al módulo 1 y este genera una fuerza de reacción de dirección (-y). Gracias a esto el módulo 2 empieza a rotar y  $q_2$  aumenta. Una vez el módulo 1 llega a  $100^\circ$  comienza a detenerse y el módulo 2 sigue su trayectoria lo cual hace que  $q_2$  comience a disminuir en  $t=0,2$  segundos. Luego el proceso se repite.



**Figura 10.** Cadena articulada de 6 módulos

Lo anteriormente nombrado se puede observar en la **Figura 11**, donde está ilustrado el movimiento de swing de la cadena de dos módulos.



**Figura 11.** Movimiento de Swing de la cadena de 2 módulos (masa=55g y largo=6,7cm)

Para continuar el código se extrapola a seis módulos, esto se puede evidenciar en los ANEXOS 1. Donde se encuentra el código para una cadena de cinco y seis módulos (Es redundante poner los demás algoritmos debido a que a partir del algoritmo de 5 y 6 módulos se pueden deducir los demás). Donde las gráficas de los códigos para la cadena de dos, tres, cuatro y cinco módulos se expondrán posteriormente. En el siguiente capítulo que hace referencia a la validación del modelo teórico. La razón del porque no se expone las gráficas del código para una cadena de seis módulos se explicará a continuación.

En el transcurso de la implementación de cada código, se pudo evidenciar el cambio de tiempos para la obtención de las variables de estado. Estos tiempos se veían comprometidos con el aumento de los módulos en la cadena articulada. En otras palabras, los resultados del código iban aumentando exponencialmente. Esto se debe a que las ecuaciones de cada módulo aumentaban por la interacción entre estos. Si el numero de módulos aumenta, también aumenta las fuerzas de interacción que estos ejercen los unos con los otros. Esto fue un detalle que aunque estaba previsto, no se pensó que iba a afectar tan drásticamente los tiempos de ejecución de cada código, para dos, tres, cuatro y cinco módulos.

Lo que dificulto la obtención de los resultados del código de la cadena articulada para seis módulos. La cual nunca se pudo simular a su totalidad y eventualmente se concluyó que no era práctico ni viable la utilización de esta para alcanzar los siguientes objetivos planteados en el presente trabajo de grado.

A continuación se mostrarán los tiempos que se obtuvieron por cada algoritmo únicamente en la instrucción que hace referencia al solucionador de ecuaciones (solve) simulados finalmente en la herramienta llamada MAPLE 14. Se realizó el cambio de software de MATLAB® a Maple, ya que en Matlab los tiempos aumentaban hasta 10 veces más que los tiempos de Maple. Cabe aclarar que estos tiempos varían o dependen del computador utilizado para implementar o ejecutar los códigos. Para los resultados que se muestran a continuación se empleó un computador con las siguientes especificaciones. Fabricante: DELL, Modelo: Studio 1558, Procesador: Intel(r) Core(TM) i7 CPU Q720 @1,60GHz 1.60 GHz, Memoria instalada (RAM) 8,00GB, tipo de sistema: 64-bit SO, ATI Radeon HD 5470 1GB. Donde la columna tiempo hace referencia al tiempo de cálculo de las variables. Y la columna del tiempo total hace referencia al tiempo en que tardaba MAPLE en arrojar el resultado en la pantalla.

Memoria						
	1ra prueba	2da prueba	3ra prueba	4ta prueba	5ta prueba	Promedio
Dos módulos	0.68MB	0.68MB	0.68MB	0.68MB	0.68MB	0.68MB
Tres módulos	0.68MB	0.68MB	0.68MB	0.68MB	0.68MB	0.68MB
Cuatro módulos	35.68MB	35.68MB	33.99MB	35.99MB	33.99MB	35.07MB
Cinco módulos	253.70MB	222.64MB	247.99MB	270.79MB	235.23MB	246.07MB
Tiempo						
	1ra prueba	2da prueba	3ra prueba	4ta prueba	5ta prueba	Promedio
Dos módulos	0.03s	-	-	0.01s	0.01s	0.01s
Tres módulos	0.03s	0.01s	0.03s	0.01s	0.03s	0.022s
Cuatro módulos	1.51s	2.99s	1.54s	1.59s	1.59s	1.59s
Cinco módulos	75.09s	78.55s	79s	71.34s	70.83s	74.962s
Tiempo total						
	1ra prueba	2da prueba	3ra prueba	4ta prueba	5ta prueba	Promedio
Dos módulos	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
Tres módulos	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
Cuatro módulos	<3 s	<2 s	<2 s	<2 s	<2 s	<3 s
Cinco módulos	<5 horas 17 min	< 5 horas 24 min	< 5 horas 34 min	< 4 horas 59 min	< 4 horas 41 min	<5 horas 15 min

**Tabla 1.** Resultados en Maple de la función solve, para los códigos 2,3,4 y 5

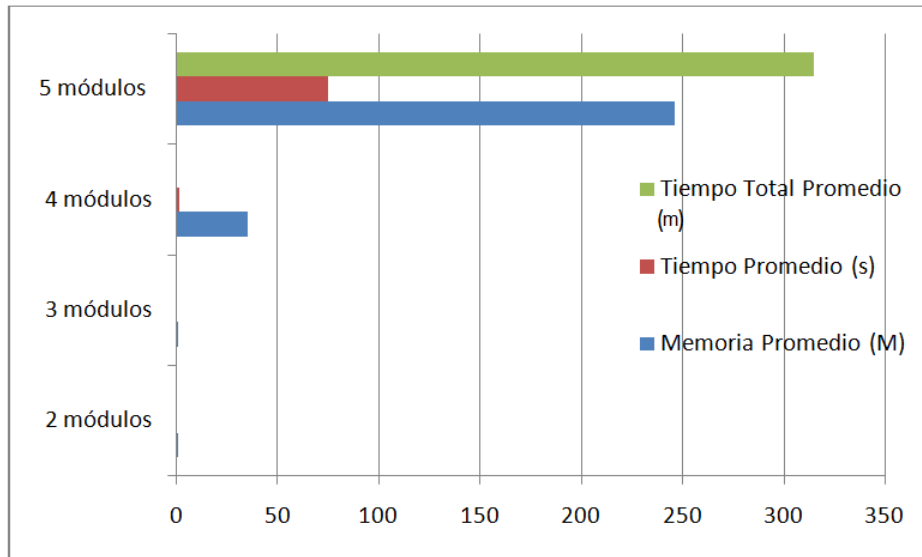
A pesar de los extensos y forzosos intentos, el código para seis módulos se simulo con las siguientes características.

#### Para seis módulos

Memoria	Tiempo:	Tiempo total
2750.70M	892.67s	> 5 días
2623.64M	842.15s	> 5 días
2907.99M	915.3s	> 5 días

**Tabla 2.** Tiempo en que era suspendida la simulación del código de 6

Para ver más claro el aumento considerable del tiempo se puede observar la siguiente gráfica.

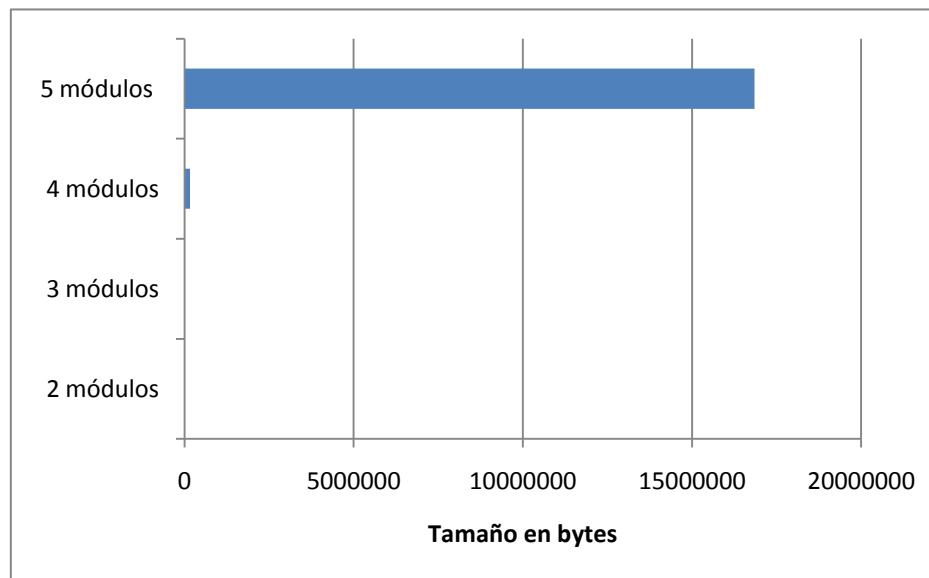


**Figura 12.** Tiempo (en segundos y minutos) y memoria (Megabytes) de obtención de datos por cada código

Otro indicador fueron los resultados de la función solve, en otras palabras las variables de estado del modelo para cada módulo. Ya que para los anteriores resultados de los códigos de dos, tres, cuatro y cinco módulos. Las variables de estado iban aumentando significativamente, dificultado su uso y su visualización en caracteres presentados en la pantalla. Esto se puede evidenciar con la siguiente tabla.

	Tamaño de variables de estado	Tamaño aproximado
2 módulos	619 bytes	619 bytes
3 módulos	10.590 bytes	10.3 KB
4 módulos	167.386 bytes	163 KB
5 módulos	16.847.348 bytes	16MB

**Tabla 3.** Aumento del tamaño de las variables de estado del código de 2, 3, 4 y 5



**Figura 13.** Tamaño (en bytes) de las variables de estado para los códigos de 2, 3, 4 y 5



Por los resultados obtenidos anteriormente, se puede decir que la implementación del modelo teórico en código para una cadena articulada de seis o más módulos es impráctico, no es viable y demorado. Por esta razón se construye el modelo de la cadena articulada en SimMechanics, siendo está una herramienta mas rápida y eficiente que el código. En la siguiente sección se validará el modelo teórico desarrollado en código con un modelo desarrollado en SimMechanics™. Por medio de esa validación y el problema de tiempos para el algoritmo de seis módulos descrito previamente. En el presente trabajo de grado para una cadena de seis articulaciones se simulará en la herramienta de SimMechanics™. Pero este tema será abordado con más claridad en la siguiente sección. Aunque el modelo en espacio de estados para la configuración de 6 módulos no pudo ser determinado, las ecuaciones de Euler-Lagrange van a servir para un análisis posterior.

### 3.3 Modelo en SimMechanics™

Para comprobar los modelos en espacio de estados para 2, 3, 4 y 5 módulos que se obtuvieron mediante las ecuaciones de Euler-Lagrange se usa una extensión del toolbox Simscape™ de MATLAB® llamada SimMechanics™.

Antes de explicar el desarrollo del modelo en simulink cabe hacer la aclaración de que la escogencia de las posiciones iniciales de los módulos es arbitraria en el momento de crear el modelo mecánico, lo importante es que sea consistente la geometría del sistema y la relación de posiciones de un módulo a otro. Por otra parte, se tiene presente que en momento de simular el mecanismo, las posiciones iniciales con las que va comenzar el sistema tienen que ser iguales que las del código. En otras palabras, se tiene presente que hay dos condiciones iniciales, la primera hace referencia a la geometría del sistema y la segunda a la posición inicial del sistema en que va ser simulado (Ver Bloque IC ANEXOS).

Para simular los dos módulos en SimMechanics™ primero se tiene que tener una referencia global. En SimMechanics™ esta referencia se llama WORLD.

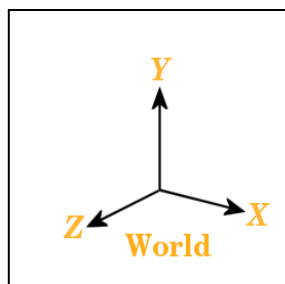


Figura 14. Los ejes de coordenadas por defecto en SimMechanics™ [19]

- Por defecto de la herramienta y está apuntando hacia arriba (Ver **Figura 14**)
- Por defecto de la herramienta x está apuntando hacia la derecha (Ver **Figura 14**)
- Por defecto de la herramienta z está apuntando hacia fuera de la hoja (por la regla de la mano derecha, Ver **Figura 14**)

Donde la gravedad por defecto va en la dirección -y (puede ser modificada). Por motivos prácticos se emplea el sistema de coordenadas por defecto. Luego se define una tierra o “ground” que es el origen de nuestro modelo mecánico que va a estar fijo o va ser un punto estacionario en el sistema. La tierra es un requisito para cualquier sistema, pero siempre debe haber al menos una tierra o un error será presentado en

la simulación. La razón es porque el sistema necesita un soporte estacionario para simular todo el mecanismo.

En este caso se escoge la posición  $[0,0,0]$  para “ground”, por lo tanto su sistema de coordenadas (CS) también estará ubicado en ese punto. (Figura 15)

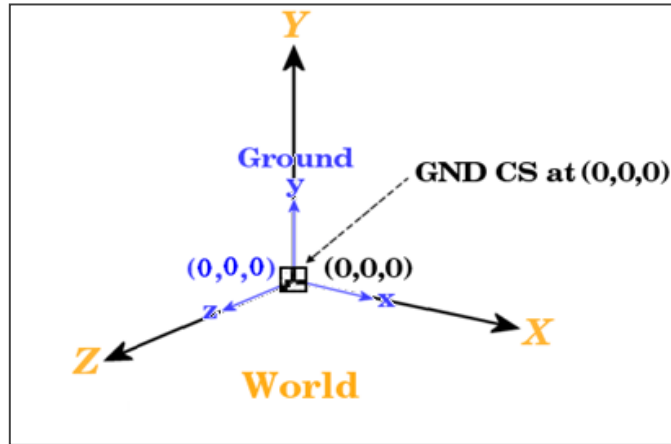


Figura 15. Ejes de coordenada de “world” y “ground” [19]

En el transcurso de la explicación del modelo mecánico se mostrará su semejante en el diagrama de bloques en Simulink, entonces el diagrama de bloques iría quedando de la siguiente forma (Ver Figura 16). El bloque de “Machine Environment” tiene que ir antes del ground para simular la gravedad ( $g = 9.81$ ) en el sistema mecánico y para escoger el modelo dinámico que se va emplear, en este caso el “Forward Dynamics”, para más detalles de este bloque ver anexos 3.

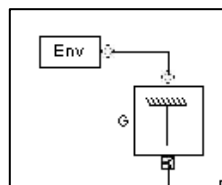


Figura 16. Primeros bloques de la cadena articulada de dos módulos

Luego se crea el bloque que va unir el “ground” con el cuerpo o en este caso con el primer módulo, donde la relación de posición se hace por medio de las sistemas de coordenadas de CS (GND) a CS1 que es la posición del origen del módulo. El bloque de unión que concuerda a nuestro mecanismo es el bloque de “revolute”, el cual es un pivote o una unión de rotación. Se asigna como parámetro el eje de rotación en  $z$   $[0,0,1]$  (Ver Figura 17).

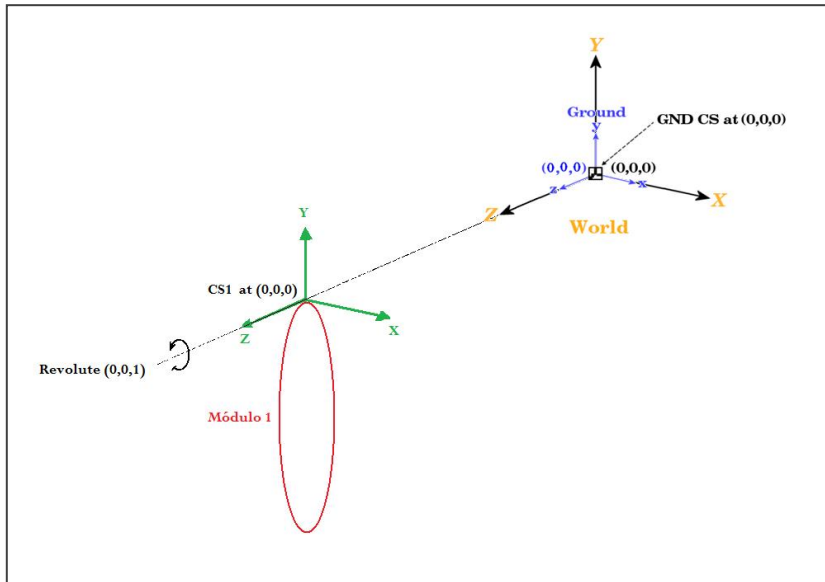


Figura 17. Relación de sistemas de coordenadas con respecto al bloque de rotación o “revolute” [19]

Después se especifica la posición del cuerpo o del primer módulo en el sistema. Donde el origen del cuerpo debe coincidir con la posición del ground para que sea consistente el sistema. Si esto no se cumple la herramienta genera un error. El bloque a utilizar es el “Body”. El cuál es la representación de un cuerpo en el sistema que posee masa, inercia y centro de masa.

Para el módulo 1 se tiene que la longitud es de 6.7 cm y que la masa es de 55 g. Los marcos de referencia cambian con respecto a los de la ecuación de estado. El eje x en la ecuación de estado es el eje y en SimMechanics™ (Figura 6). La matriz de inercia es igual a  $\begin{bmatrix} \left(\frac{m_1 \cdot (c^2 + a^2)}{3}\right), & \left(-\frac{m_1 \cdot (c+b)}{4}\right), & \left(-\frac{m_1 \cdot (a+b)}{4}\right) \\ \left(-\frac{m_1 \cdot (c+b)}{4}\right), & \left(\frac{m_1 \cdot (b^2 + a^2)}{3}\right), & \left(-\frac{m_1 \cdot (a+c)}{4}\right) \\ \left(-\frac{m_1 \cdot (a+b)}{4}\right), & \left(-\frac{m_1 \cdot (a+c)}{4}\right), & \left(\frac{m_1 \cdot (c^2 + b^2)}{3}\right) \end{bmatrix}$  por el corrimiento en el punto de rotación en CS1. El centro de gravedad está en el centro del módulo (Ver Figura 18).

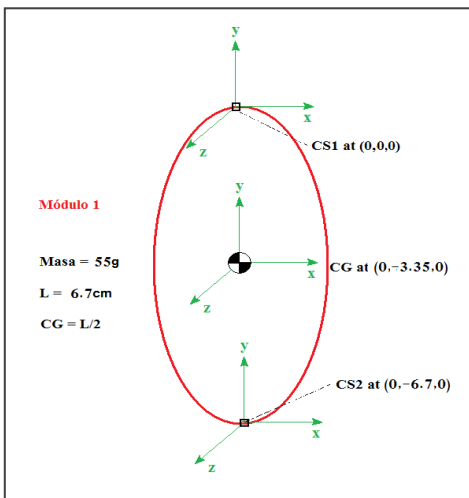


Figura 18. Primer módulo con sus características. CS1 sistema de coordenadas del origen del módulo. CG sistema de coordenadas del centro de gravedad. CS2 sistema de coordenadas del final del módulo.

Por lo tanto el esquemático del primer módulo sería de la siguiente forma.

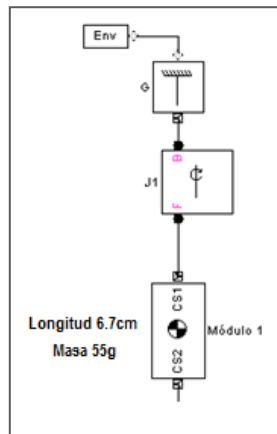


Figura 19. Visualización del primer módulo en simulink

Luego se repite el proceso para el segundo módulo teniendo en cuenta la posición geométrica de éste con respecto al eje de coordenadas global (Ver **Figura 20**). Y siguiendo la metodología de anotación se tiene que para el origen del módulo 2 se expresa CS1 (Coordinate System 1), el final CS2 (Coordinate System 2) y la matriz de inercia es  $\left[ \left( \frac{m_1 \cdot (c^2 + a^2)}{12} \right), 0, 0; 0, \left( \frac{m_1 \cdot (b^2 + a^2)}{12} \right), 0; 0, 0, 0, \left( \frac{m_1 \cdot (c^2 + b^2)}{12} \right) \right]$ .

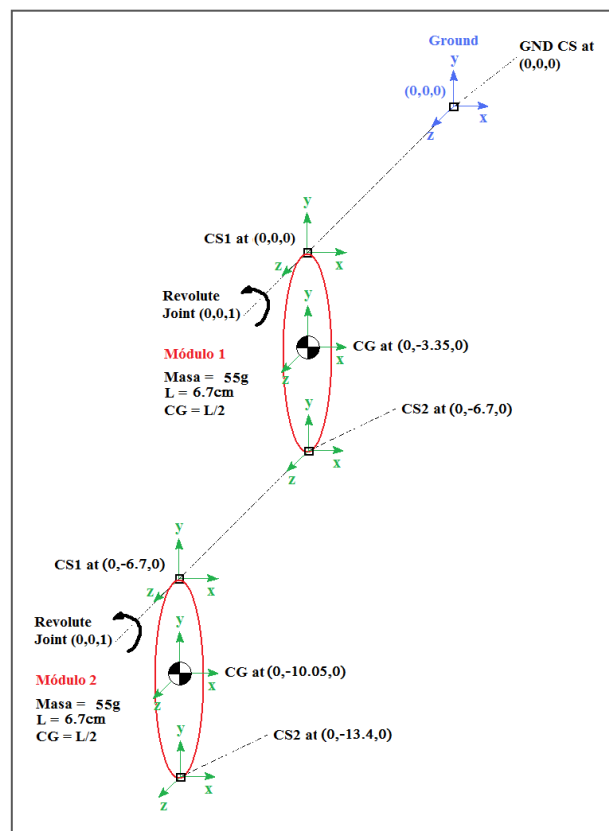


Figura 20. Relación de los sistemas de coordenadas de la cadena articulada de dos módulos

Por último se agregan dos bloques a cada módulo los cuales son para establecer la posición inicial de los módulos y dos sensores para medir la variación del ángulo, velocidad angular y aceleración angular en el tiempo.

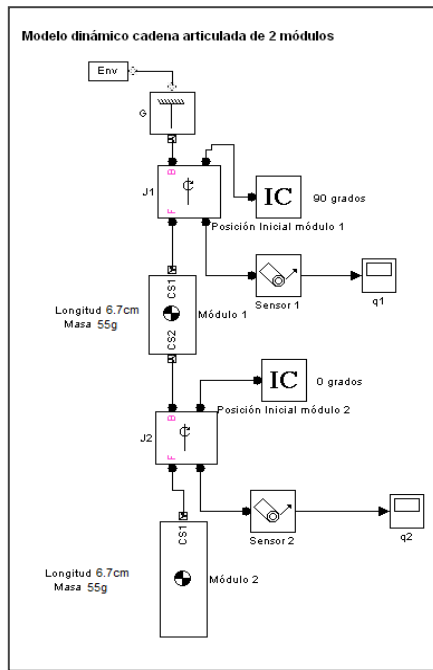


Figura 21. Modelo dinámico de la cadena articulada de 2 módulos en SimMechanics™ [9]

Teniendo en cuenta las siguientes convenciones:

MS – Medio movimiento de swing.

SC – Swing completo.

Las gráficas de las variables de junta en SimMechanics™ para 2 módulos son

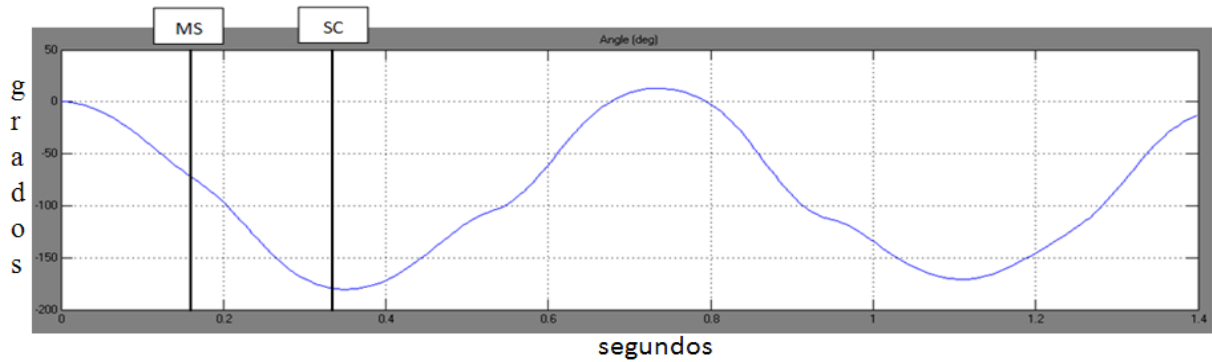


Figura 22. Variable de junta  $q_1$  de la cadena de 2 módulos (masa=55g y largo=6,7cm)

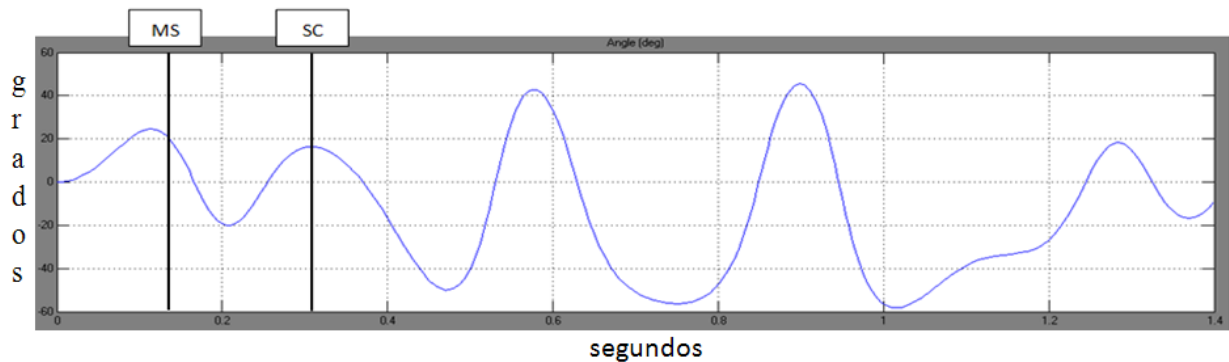


Figura 23. Variable de junta  $q_2$  de la cadena de 2 módulos (masa=55g y largo=6,7cm)

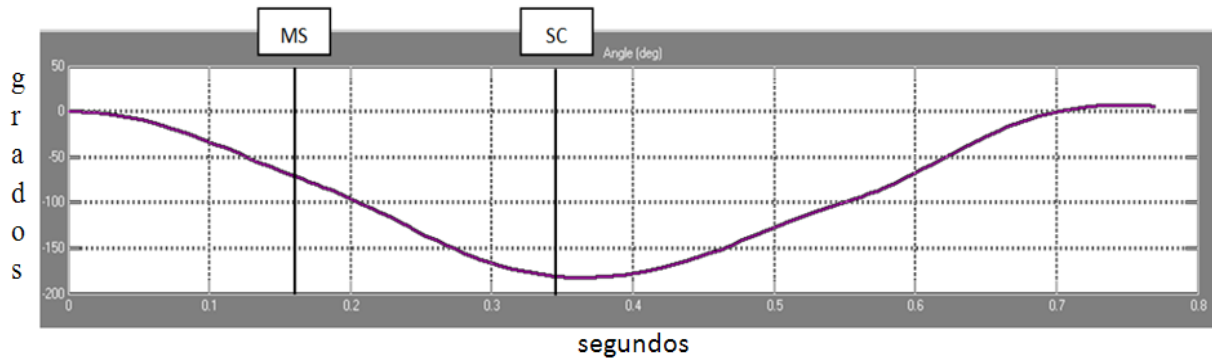
Las gráficas para las demás configuraciones de SimMechanics™ se encuentran en ANEXOS 4.

### 3.3.1 Validación del modelo

Una vez obtenidas todas las variables de junta, por medio de SimMechanics™, se usan para validar el modelo de las ecuaciones de estado. Para esto se comparan las gráficas obtenidas en SimMechanics™ con las gráficas obtenidas por medio de las ecuaciones de estado mediante el punto máximo del error absoluto (valor absoluto de la resta de las gráficas) y la correlación de las gráficas.

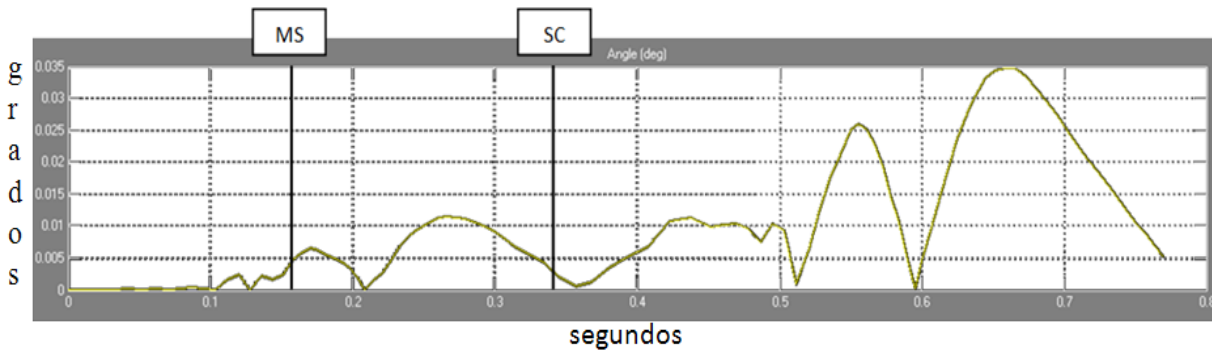
#### Cadena articulada de 2 módulos

##### Variable de junta $q_1$



**Figura 24.** Gráfica de SimMechanics™ y de la ecuación de estado para  $q_1$  (cadena articulada de 2 módulos).

En la **Figura 24** se muestra como la gráfica de SimMechanics™ sigue a la gráfica de la ecuación de estado para la variable de junta  $q_1$ . El error está representado por la siguiente gráfica



**Figura 25.** Error de  $q_1$  entre SimMechanics™ y las ecuaciones de estado (cadena articulada de 2 módulos).

Cabe aclarar que la correlación se obtiene de una función interna de MATLAB® llamada corr.

Correlación entre gráficas= 0.999999982692717

Max. error absoluto = 0.0347°

### Variable de junta $q_2$

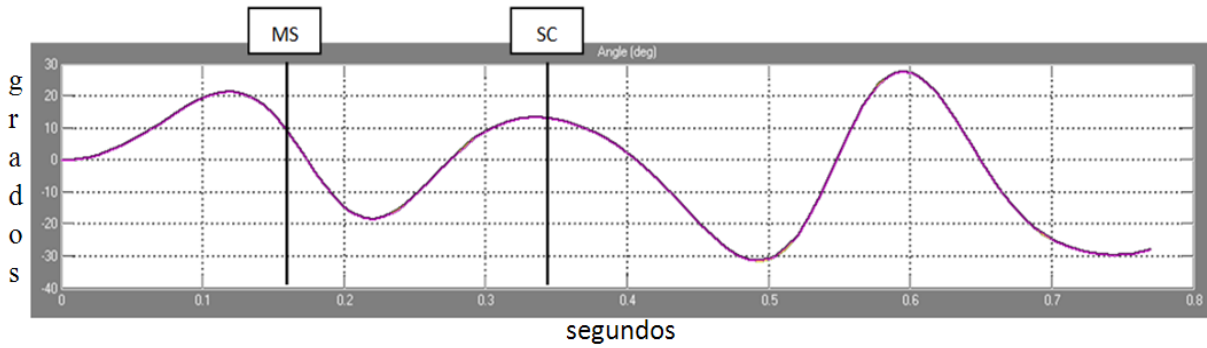


Figura 26. Gráfica de SimMechanics™ y de la ecuación de estado para  $q_2$  (cadena articulada de 2 módulos).

En la **Figura 26** se muestra como la gráfica de SimMechanics™ sigue a la gráfica de la ecuación de estado para la variable de junta  $q_2$ . El error está representado por la siguiente gráfica

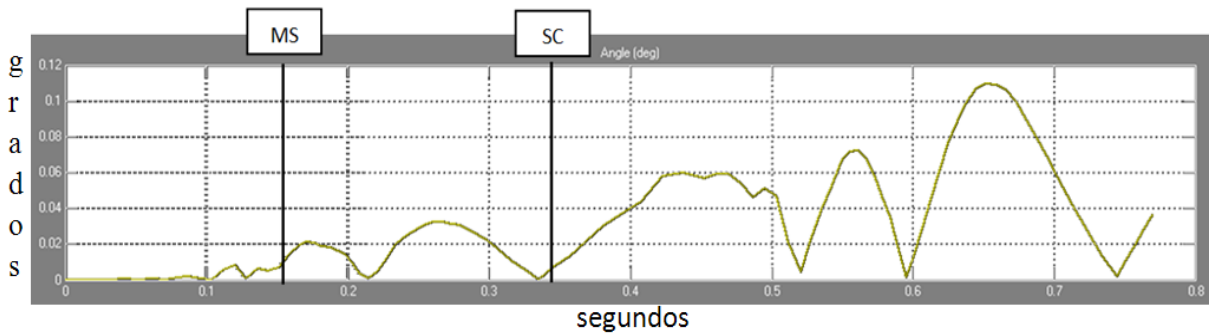


Figura 27. Error de  $q_2$  entre SimMechanics™ y las variables de estado (cadena de 2 módulos).

Correlación entre gráficas = 0.999997181120045

Max. error absoluto= 0.1100°

En este documento solo se muestran las gráficas de error para las demás configuraciones. Las gráficas de las variables de junta superpuestas se encuentran en **ANEXOS 3**. Los valores máximos de los errores absolutos junto con la correlación de las gráficas se muestran en una tabla al final de la sección.

### Cadena articulada de 3 módulos

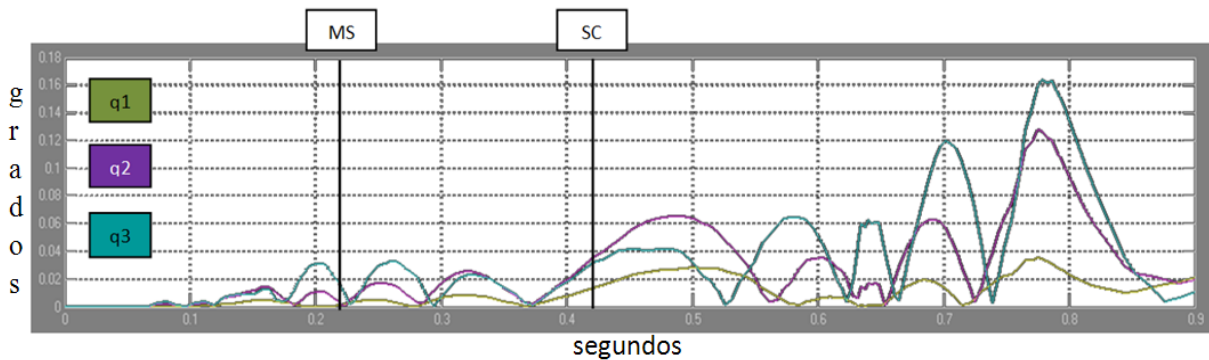


Figura 28. Error entre SimMechanics™ y las variables de estado para la cadena articulada de 3 módulos.

La gráfica de error en la variable de junta  $q_1$  es menor que en  $q_2$ . Por otro lado la gráfica de error en la variable de junta  $q_2$  es menor que en  $q_3$ .

Cadena articulada de 4 módulos

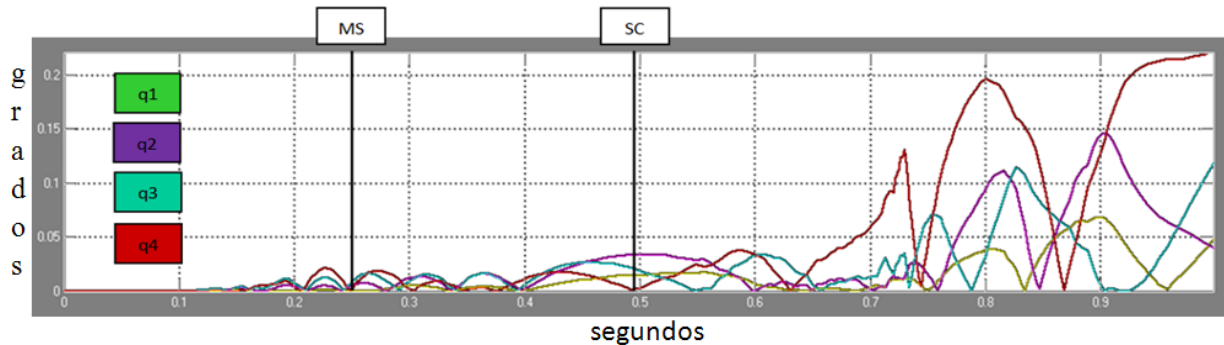


Figura 29. Error entre SimMechanics™ y las variables de estado para la cadena articulada de 4 módulos.

En la **Figura 29** se observa como las gráficas de error presentan la tendencia  $q_1 < q_2 < q_3 < q_4$ .

Cadena articulada de 5 módulos

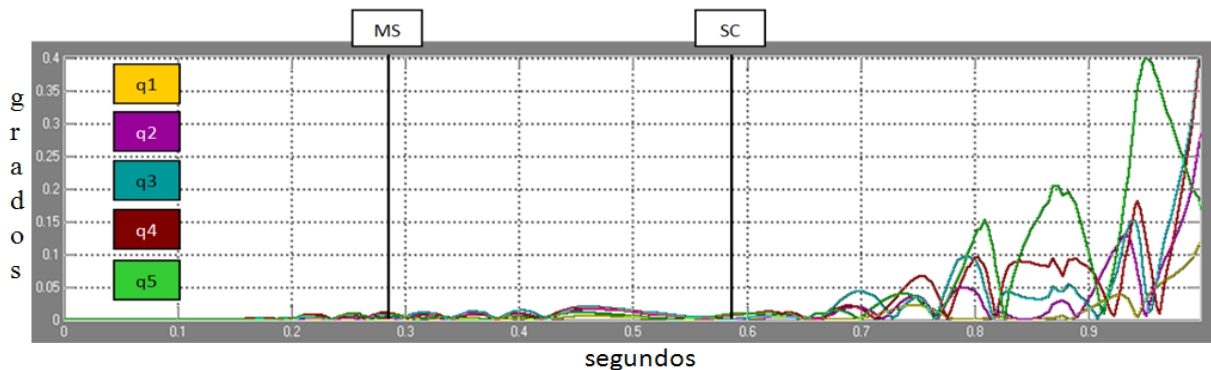


Figura 30. Error entre SimMechanics™ y las variables de estado para la cadena articulada de 5 módulos.

En la **Figura 30** se observa como las gráficas de error presentan la tendencia  $q_1 < q_2 < q_3 < q_4 < q_5$ .

En la **Tabla 4** se muestran los errores para todas las configuraciones. Se usan 8 cifras significativas para comparar las correlaciones teniendo en cuenta que las simulaciones se hicieron bajo las mismas condiciones y con el mismo software.



Cadena articulada de 2 módulos									
q1					q2				
CORR		MAX			CORR		MAX		
0.99999998		0.03471907°			0.99999718		0.11000924°		
Cadena articulada de 3 módulos									
q1			q2			q3			
CORR		MAX	CORR		MAX	CORR		MAX	
0.99999998		0.03557992°	0.99999844		0.12766780°	0.99999786		0.16351922°	
Cadena articulada de 4 módulos									
q1		q2			q3		q4		
CORR	MAX	CORR	MAX		CORR	MAX	CORR	MAX	
0.99999612	0.06839049°	0.99993734	0.14674045°		0.99984866	0.11908252°	0.77571825	0.22211339°	
Cadena articulada de 5 módulos									
q1		q2		q3		q4		q5	
CORR	MAX	CORR	MAX	CORR	MAX	CORR	MAX	CORR	MAX
0.99999564	0.09432780°	0.99992445	0.20317064°	0.999980374	0.31111847°	0.99985813	0.30105905°	0.99996263	0.40059182°

Tabla 4. Punto máximo del error absoluto y correlación para distintas configuraciones

Con los resultados que se encuentran en la **Tabla 4** se validan los modelos planteados con la ecuación de estado para cada configuración. El punto máximo del error absoluto es despreciable en comparación con las variaciones de las variables de junta para todas las configuraciones. Así mismo las correlaciones entre las gráficas de SimMechanics™ y las gráficas de la ecuación de estado se aproximan, en la mayoría de los casos, a 1. La correlación es una medida que muestra la similitud entre dos funciones: Si el resultado es 1 las 2 funciones son iguales. Si el resultado es -1 las 2 gráficas son inversas.

### 3.4 Posición inicial de mínima energía de la cadena articulada.

Para garantizar la posición inicial de mínima energía se usa la ecuación catenaria (que genera curvas como las de la **Figura 31**) que asegura el equilibrio estático del modelo. Esto significa que los módulos de la cadena en su posición de inicio de movimiento no van a realizar torques.

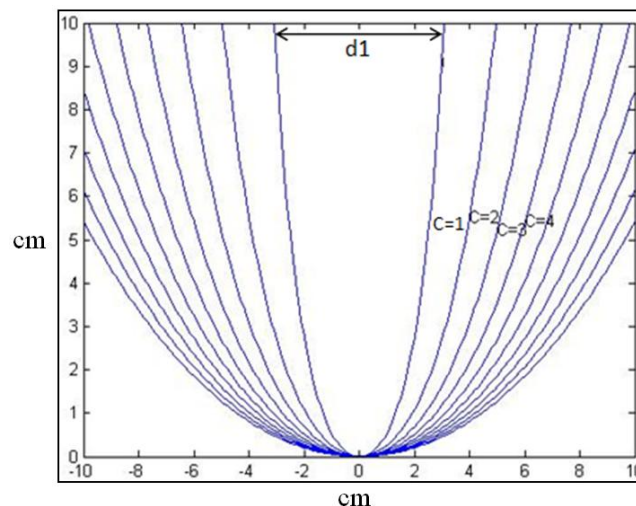
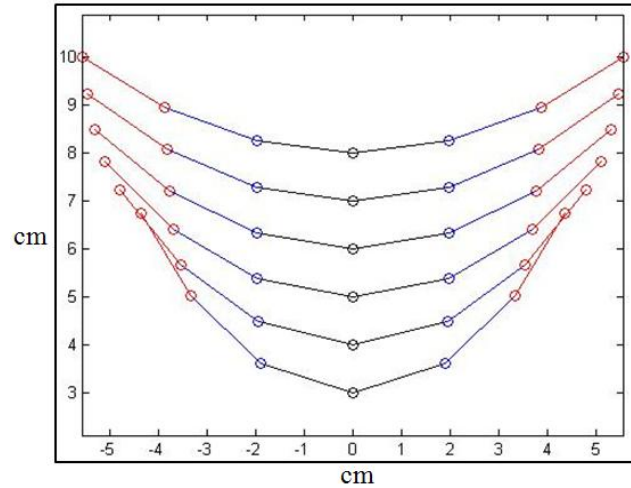


Figura 31. Gráfica de la ecuación de catenaria para c=1,2,3 ...10

El paso a seguir es discretizar la ecuación catenaria para (n) módulos. Esto se desarrolla con un algoritmo en MATLAB®. El cual consiste en encontrar dos puntos en la curva catenaria que tengan una distancia d. Para esto se ubica el primer punto en  $x_0 = 0$  y  $y_0 = 0$ , luego se plantea la distancia euclidiana  $d^2 = \Delta X^2 + \Delta Y^2$ , para  $y = c \cdot \cosh\left(\frac{x}{c}\right)$ . Por lo tanto se obtiene una ecuación con una incógnita y dos posibles soluciones. Este paso se repite (n) veces y luego se determina el ángulo de cada módulo con la fórmula  $\theta_i = \text{atan}\left(\frac{y_i}{x_i}\right)$ . Esto determina todos los ángulos iniciales para varios puntos de agarre de la cadena articulada (ver **Figura 32**).



**Figura 32.** Posiciones iniciales de la cadena articulada de 6 módulos

Para encontrar la posición inicial de mínima energía hay que determinar por medio de cinemática directa cual es la posición del último módulo y del centro de gravedad para cada configuración. Para esto inicialmente a cada módulo se le asigna un marco de referencia  $o_i x_i y_i z_i$  y una variable de junta  $q_i$ , que en este caso representa el ángulo de rotación  $\theta_i$ . El marco de referencia inercial  $o_0 x_0 y_0 z_0$  está fijado al módulo 1 (En la gráfica se utiliza la terminología **SC1** para ser consecuentes con el modelo hecho en SimMechanics™), ver **Figura 33**.

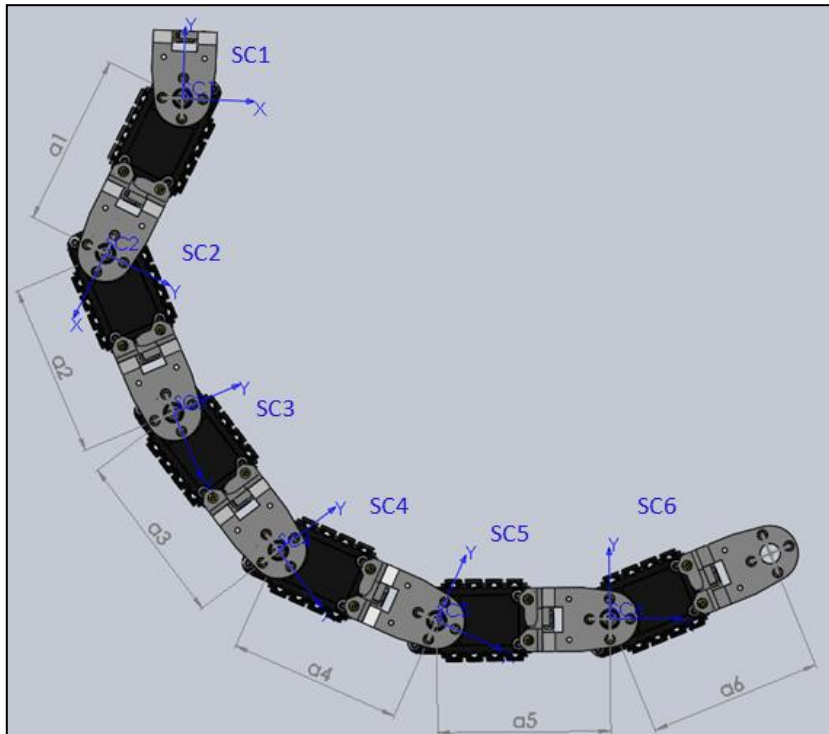


Figura 33. Marcos de referencia.

Cuando se han escogido todos los marcos de referencia se determinan los parámetros DH, como lo muestra la **Tabla 2**:

Módulo	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$a_1$	0	0	$\theta_1^*$
2	$a_2$	0	0	$\theta_2^*$
3	$a_3$	0	0	$\theta_3^*$
4	$a_4$	0	0	$\theta_4^*$
5	$a_5$	0	0	$\theta_5^*$
6	$a_6$	0	0	$\theta_6^*$

Tabla 2. Parámetros DH. \*variable.

Los términos  $\alpha_i$  (el "twist" del módulo) y  $d_i$  (el "offset" del módulo) son cero debido a que la cadena articulada es planar.

Se aplican las siguientes convenciones para simplificar la notación de las ecuaciones:

- $c_n = \cos\theta_n$
- $s_n = \text{sen}\theta_n$
- $c_{123\dots n} = \cos(\theta_1 + \theta_2 + \theta_3 + \dots + \theta_n)$
- $s_{123\dots n} = \text{sen}(\theta_1 + \theta_2 + \theta_3 + \dots + \theta_n)$

Aplicando la (2) se obtienen todas las matrices de transformación

$$\begin{aligned}
A_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_2 &= \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
A_3 &= \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_4 &= \begin{bmatrix} c_4 & -s_4 & 0 & a_4 c_4 \\ s_4 & c_4 & 0 & a_4 s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
A_5 &= \begin{bmatrix} c_5 & -s_5 & 0 & a_5 c_5 \\ s_5 & c_5 & 0 & a_5 s_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_6 &= \begin{bmatrix} c_6 & -s_6 & 0 & a_6 c_6 \\ s_6 & c_6 & 0 & a_6 s_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{29}$$

Utilizando (1)

$$H_{i+1}^i = A_i A_{i+1} = \begin{bmatrix} R_{i+1}^i & o_{i+1}^i \\ 0 & 1 \end{bmatrix}$$

Se obtienen las matrices de transformación  $H$

$$\begin{aligned}
H_1^0 &= A_1 & H_2^0 &= A_1 A_2 & H_3^0 &= A_1 A_2 A_3 \\
H_4^0 &= A_1 A_2 A_3 A_4 & H_5^0 &= A_1 A_2 A_3 A_4 A_5 & H_6^0 &= A_1 A_2 A_3 A_4 A_5 A_6
\end{aligned} \tag{30}$$

$$H_6^0 = \begin{bmatrix} c_{123456} & -s_{123456} & 0 & a_6 c_{123456} + a_5 c_{12345} + a_4 c_{1234} + a_3 c_{123} + a_2 c_{12} + a_1 c_1 \\ s_{123456} & c_{123456} & 0 & a_6 s_{123456} + a_5 s_{12345} + a_4 s_{1234} + a_3 s_{123} + a_2 s_{12} + a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solo se muestra  $H_6^0$  ya que a partir de ella se pueden deducir las demás matrices de transformación.

Mediante la (30) se deduce la expresión general para determinar la matriz de transformación entre módulos.

$$H_{i+1}^i = \begin{bmatrix} \cos(\theta_i + \theta_{i+1}) & -\text{sen}(\theta_i + \theta_{i+1}) & 0 & a_{i+1} \cos(\theta_i + \theta_{i+1}) + a_i \cos(\theta_i) \\ \text{sen}(\theta_i + \theta_{i+1}) & \cos(\theta_i + \theta_{i+1}) & 0 & a_{i+1} \text{sen}(\theta_i + \theta_{i+1}) + a_i \text{sen}(\theta_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

Con la matriz  $H_{i+1}^i$  se determina la posición de un módulo  $i$  con respecto al marco de referencia inercial.

$$x_i^0 = a_n c_{123\dots i} + \dots + a_2 c_{12} + a_1 c_1 \tag{32}$$

$$y_i^0 = a_i s_{123\dots i} + \dots + a_2 s_{12} + a_1 s_1 \tag{33}$$

$$z_i^0 = 0 \tag{34}$$

Se determina el vector de posición del centro de gravedad de cada módulo y por medio de la ecuación (8) que se muestra a continuación nuevamente

$$\vec{r}_{cm} = \frac{\sum_i m_i \vec{r}_i}{\sum_i m_i}$$

Se calcula el centro de gravedad de toda la cadena articulada.

Finalmente para cada posición inicial se obtienen las gráficas de las variables de junta en el tiempo. Con (33) se calcula y se gráfica la posición en y del último módulo. Se itera este proceso para las distintas configuraciones y se comparan todas las gráficas de posición del último módulo buscando el punto máximo en y. Este proceso se repite empleando la posición del centro de gravedad (8). Se comparan ambos resultados en el mismo tiempo para tener un punto de comparación estable y por último se escoge la posición inicial de mínima energía. Este proceso de discretización solo se realizó para la cadena articulada de 6 módulos (para cumplir el objetivo del trabajo, pero es posible discretizar las cadenas de 2, 3, 4 y 5 módulos) y se implementó solamente en SimMechanics™ (debido a que los tiempos de simulación en SimMechanics™ son menores a los del código y también a la simplicidad que presenta SimMechanics™ a la hora de modificar los parámetros físicos).

En la siguiente tabla se encuentran algunas de las posiciones iniciales probadas en la cadena articulada para un tiempo de 0.4200 s, se escogió este tiempo ya que en algunas configuraciones después de 0.4200 s la cadena articulada sufría deformaciones que generaban un movimiento de swing no uniforme. La tabla con todas las C que fueron probadas se encuentra en ANEXOS 2.

C	Tiempo (s)	Último módulo	Centro de gravedad	Distancia d (m)
		Posición en y (m)	Posición en y (m)	
5	0.4200	-0.2091	-0.1483	0.2113
6	0.4200	-0.2362	-0.1532	0.2322
13	0.4200	-0.2909	-0.1472	0.3183
15	0.4200	-0.2897	-0.1447	0.3318
25	0.4200	-0.2731	-0.1375	0.3685
28	0.4200	-0.2685	-0.1362	0.3742
35	0.4200	-0.2595	-0.1338	0.3830
37	0.4200	-0.2574	-0.1333	0.3848
44	0.4200	-0.2512	-0.1318	0.3895
45	0.4200	-0.2505	-0.1316	0.3900
50	0.4200	-0.2471	-0.1308	0.3921
51	0.4200	-0.2465	-0.1307	0.3925
60	0.4200	-0.2420	-0.1296	0.3950
70	0.4200	-0.2384	-0.1287	0.3968

Tabla 3. Altura máxima alcanzada con diferentes posiciones iniciales.

Los valores más pequeños para C (4,5,6,7,8...10) presentan una configuración inicial que genera un movimiento violento indeseado el cual produce un gasto energético innecesario. Para los valores más grandes de C (50, 51, 60... 70) se puede apreciar que la altura que alcanza el último módulo y el centro de gravedad es mayor al igual que la distancia d. Al ser la distancia d mayor el problema surge al momento de implementar la cadena articulada, ya que la posición de agarre de la cadena no es implementable debido a limitaciones físicas. Por estas razones se escogió un valor intermedio de C que permita a la cadena agarrarse de la horizontal y al mismo tiempo que no genere ningún movimiento violento al

liberarse. Luego el valor escogido para C es 28. La gráfica de esta posición inicial se muestra a continuación

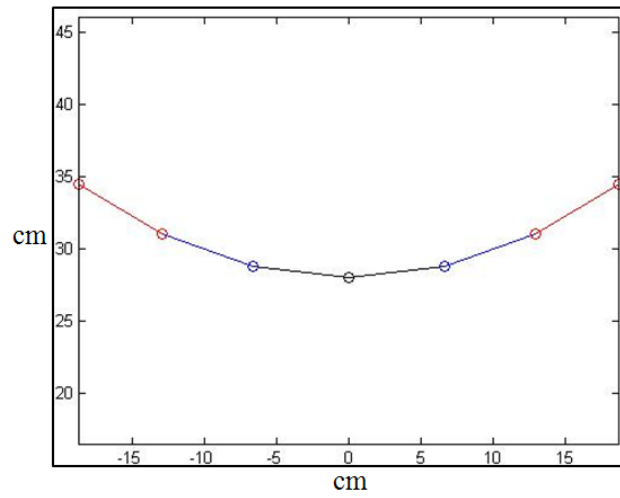


Figura 34. Posición inicial C=28.

Las gráficas para C=28 de posiciones en y se muestran a continuación

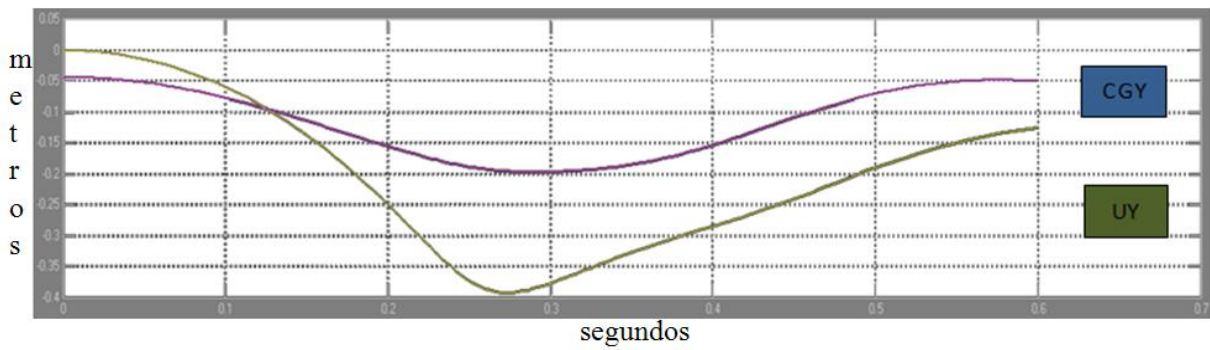
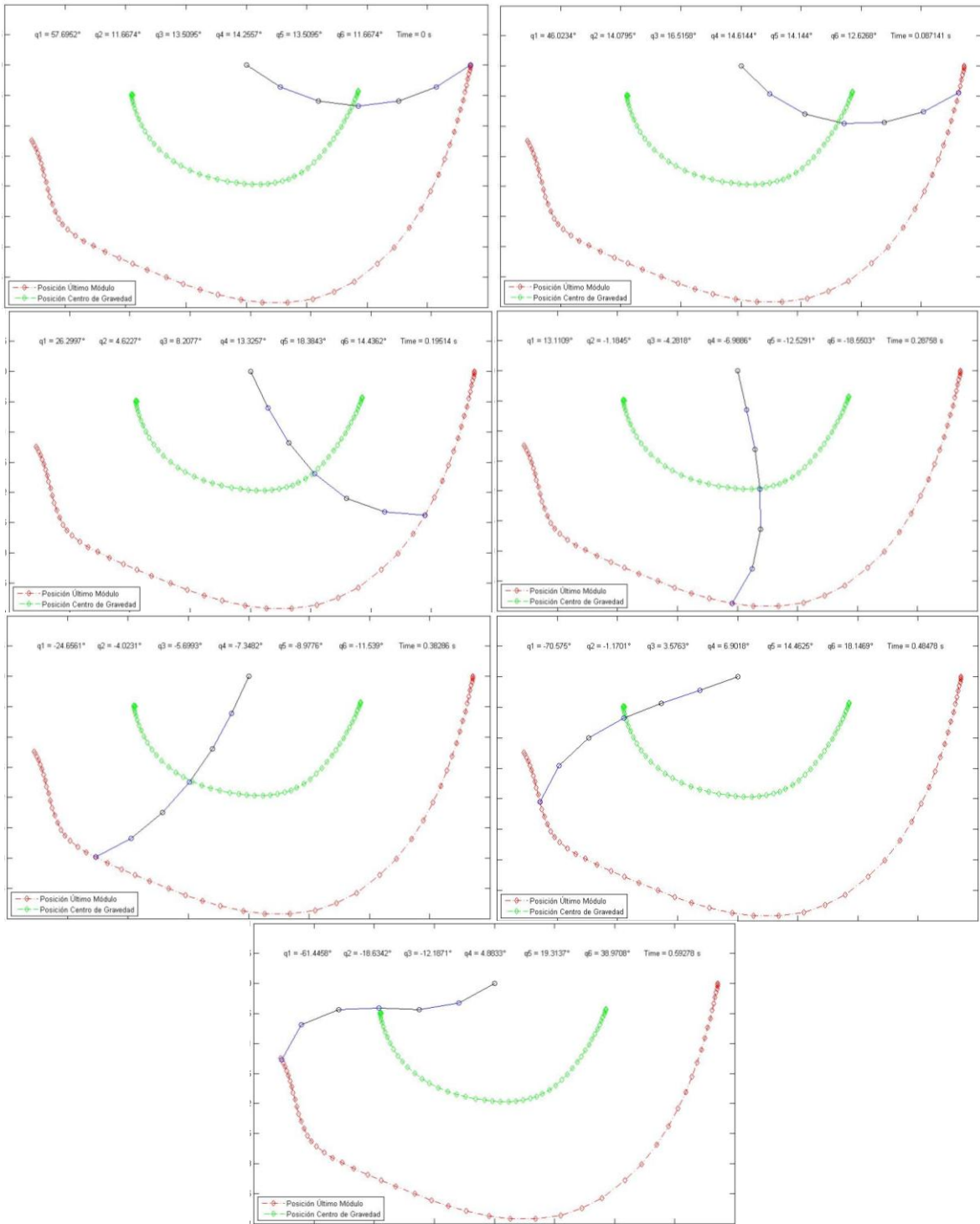


Figura 35. CGY es la posición en y del centro de gravedad. UY es la posición en y del último módulo. C=28



**Figura 36.** Movimiento swing para  $C=28$ , donde se muestra la trayectoria de posición del centro de gravedad del sistema y la posición del último módulo para  $t = 0s, 0.087s, 0.195s, 0.287s, 0.382s, 0.484s, 0.592s$ . (Unidades en metros)

### 3.5 Variables de junta y torques de entrada

A partir de la posición inicial que genera el movimiento de swing con el mínimo consumo de energía hay que encontrar cuales deben ser los torques de entrada para que la cadena articulada termine el movimiento de swing. El método a implementar consiste en dos partes, la primera es usar los torques que genera la gravedad sobre la cadena en medio movimiento de swing. Para luego utilizar los mismos para terminar la otra mitad del movimiento.

Para esto, se calculan los torques debidos a la gravedad usando la ecuación de Euler-Lagrange para seis módulos que se encuentra en el algoritmo para medio movimiento de swing. Pese a que no se pudo obtener las variables de estado mediante el algoritmo del modelo teórico. Se puede usar la ecuación Euler-Lagrange la cual si se pudo obtener. Donde se implementa la (14).

$$\sum_i d_{kj}(q)\ddot{q}_j + \sum_{i,j} c_{ijk}(q)\dot{q}_i\dot{q}_j + \phi_k(q) = \tau_k \quad k = 1, \dots, n$$

La ecuación anterior representa los torques de cada módulo dependiendo de los ángulos, las velocidades angulares y las aceleraciones angulares de estos. Estos valores se obtienen a partir de la simulación de 6 módulos en SimMechanics™. Las gráficas se presentan a continuación:

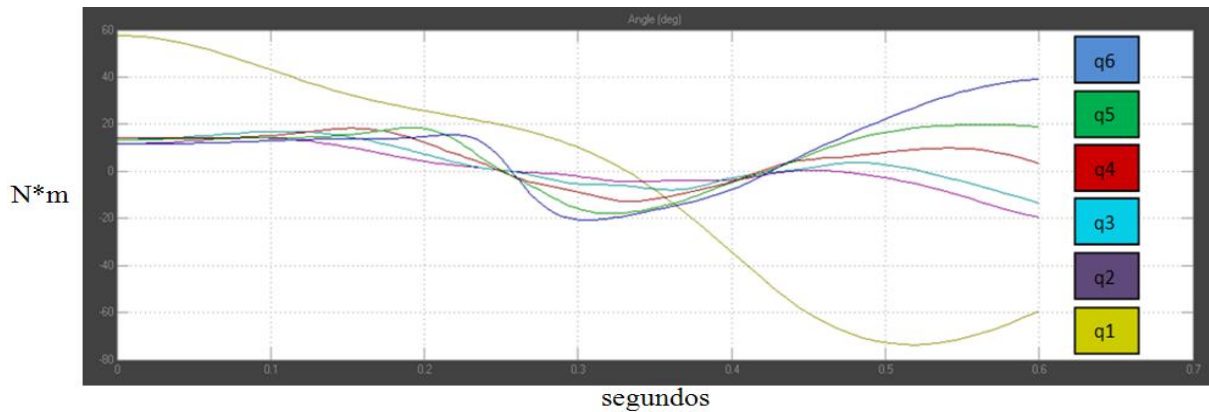


Figura 37. Ángulos de cada módulo en el tiempo, para C=28

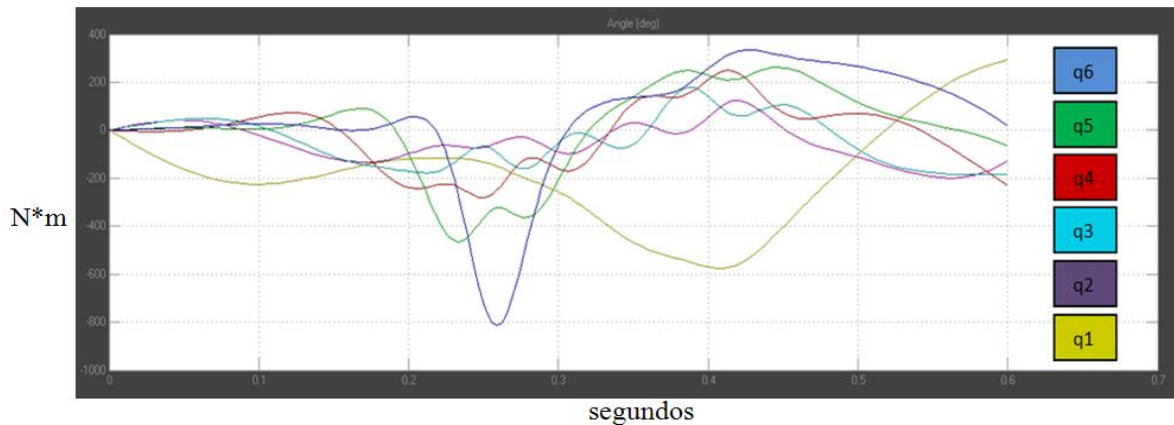


Figura 38. Velocidades angulares de cada módulo en el tiempo, para C=28



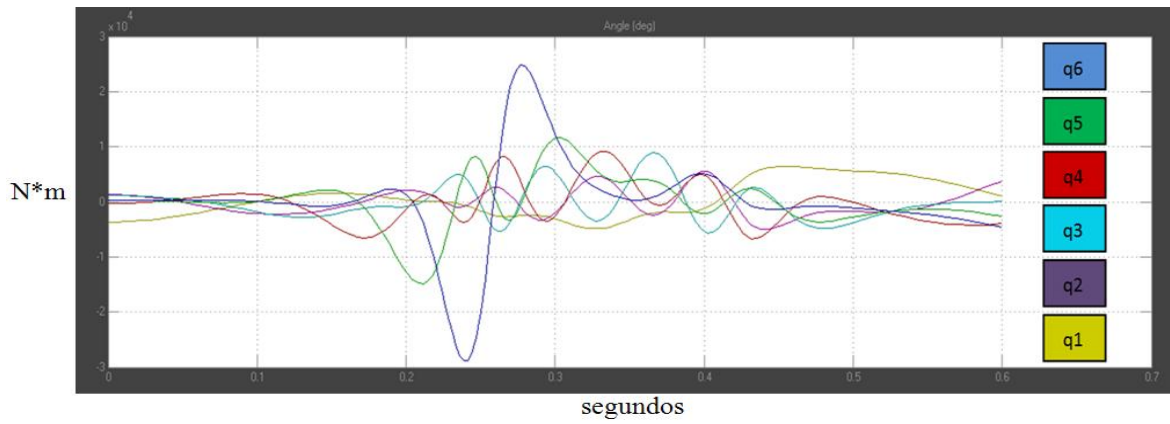
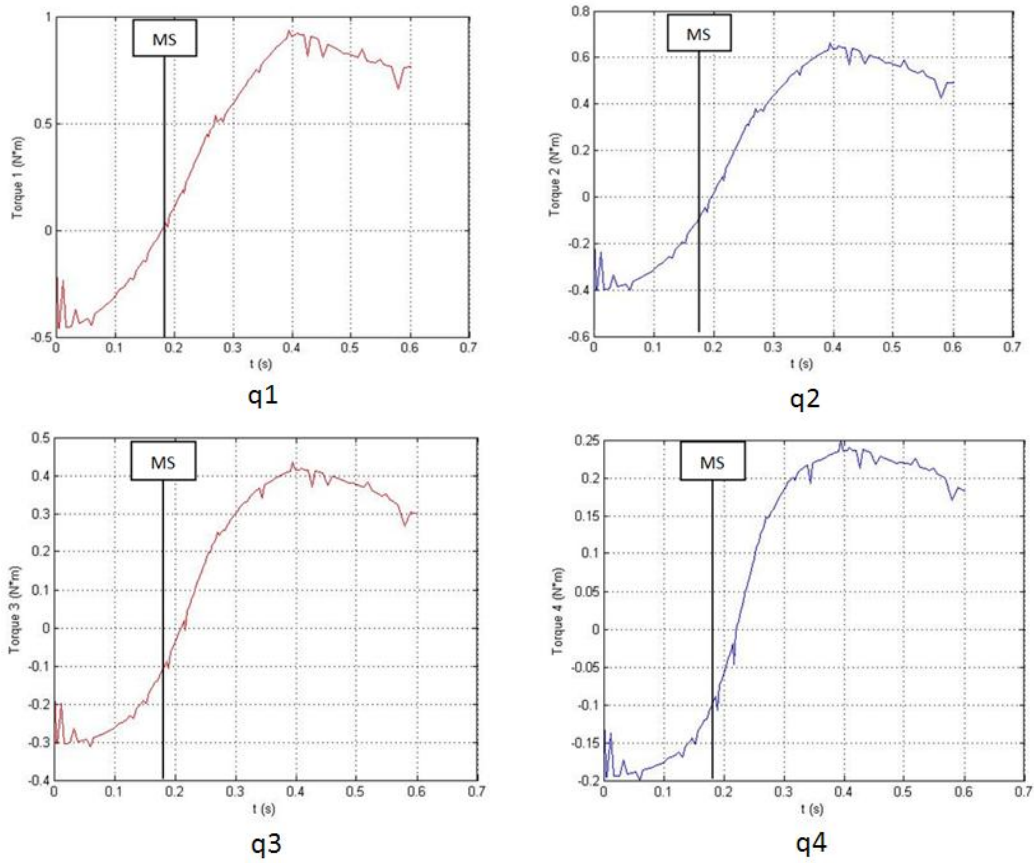
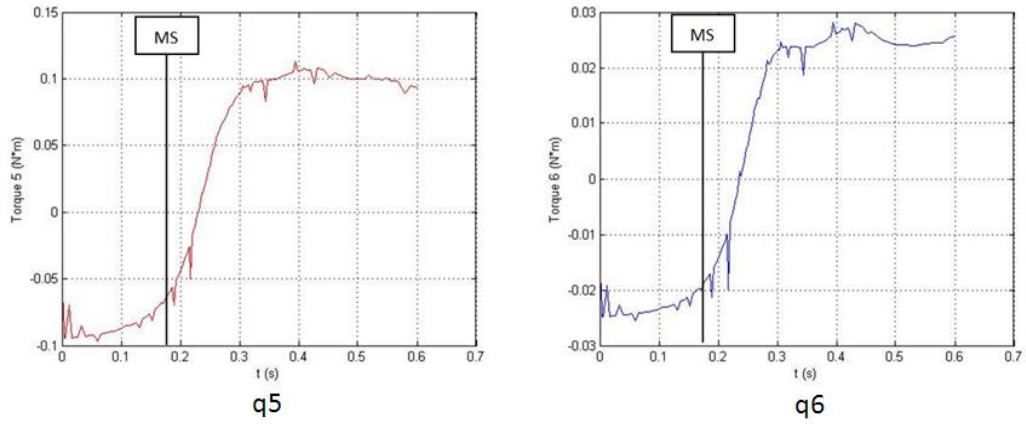


Figura 39. Aceleraciones angulares de cada módulo en el tiempo, para C=28

Estas variables se evalúan en la (14) y los torques resultantes para cada módulo son los mostrados en la Figura 40





**Figura 40.** Torques evaluados con los ángulos, velocidades y aceleraciones obtenidas por SimMechanics™, para C=28

En la **Figura 40** se observa como los torques inicialmente son negativos debido a que la cadena articulada se mueve a favor de la gravedad y se vuelven positivos en el punto donde la cadena va en contra de la gravedad. Con estos torques se determinan cuales van a ser los torques de la segunda mitad del movimiento de swing. Esto se hace tomando las gráficas de torques hasta MS y reflejarlas en  $y$  y luego en  $x$  para encontrar la terminación de las gráficas de torque para todo el movimiento de swing.

## 4. CONCLUSIONES

En este trabajo de grado de carácter investigativo se analizó el modelo dinámico pasivo de una cadena articulada de seis módulos. Donde se empleo una ecuación matemática (catenaria) para determinar la posición inicial de la cadena suspendida que empleará la mínima energía en el movimiento de swing. Luego se determinaron los torques por medio de las variables de posición, velocidad y aceleración angular con base al modelo dinámico del movimiento swing. En el transcurso del trabajo de grado presente se pudo analizar y concluir lo siguiente:

Se determinó un modelo en espacio de estados para la cadena articulada de 6DOF a partir de las ecuaciones de Euler-Lagrange ampliando el algoritmo que se implemento para la cadena articulada de dos módulos. Por otro lado, esta representación en variables de estado no es viable debido a que los tiempos de simulación crecen de forma exponencial con el número de módulos de la cadena articulada **Figura 11**. Esta es una de las razones por las cuales se implementó el modelo en SimMechanics, el cual permite reducir los tiempos de simulación.

Se validó el modelo en espacio de estados comparándolo con el modelo de parámetros físicos implementado en SimMechanics para las configuraciones de 2, 3, 4, y 5 módulos. Esta comparación se hizo a través del error absoluto y de la correlación entre gráficas.

Para valores pequeños de C las posiciones iniciales de la cadena articulada generan movimientos fuera del rango  $\pm 90^\circ$  los cuales no se pueden implementar en la estructura real. Igualmente estos movimientos a manera de “látigo” producen un consumo de energía innecesario. Para los valores más grandes de C la distancia entre los puntos de agarre (d) es mayor (ver Tabla 5), lo que genera un problema al implementar la estructura real debido a que la mecánica de la cadena no permite un agarre adecuado de la misma. Por estas razones se escogió un valor intermedio de  $C=28$ . En este documento se muestran criterios para escoger la posición inicial de la cadena articulada (a partir de la C de la ecuación catenaria). Es posible usar otros criterios para determinar una posición inicial diferente a la escogida y ejecutar el mismo análisis.

Se especificaron los puntos de agarre de la posición inicial de mínima energía para la cadena articulada. Los cuales, en una futura implementación de un robot real, reducen los torques que habría que exigirle a la cadena para que ejecute el movimiento de swing completo. Y el simple hecho de minimizar la energía utilizada en el movimiento de swing, es una ventaja para cualquier sistema que tiene como fuente baterías portables.

La gráfica (Ver **Figura 34**) de posición en y del centro de gravedad muestra como la energía potencial inicial de la cadena articulada se transforma en energía cinética para luego convertirse de nuevo en energía potencial. Esto demuestra el principio de conservación de la energía. Por otro lado la gráfica de posición en y del último módulo muestra como la energía potencial inicial se transforma en energía cinética, pero luego una parte de la energía cinética se vuelve a transformar en energía potencial mientras la otra parte continua siendo cinética. Es por esta razón que el último módulo no retorna a su altura inicial.

El uso de la herramienta de Simulink puede ser muy útil para el desarrollo e implementación de un robot. Ya que el modelo dinámico que este simula es igual de valido al teórico, como se pudo validar en los

resultado de la **Tabla 4**. A su vez porque los tiempos de esta herramienta son considerablemente menores a los empleados por el algoritmo teórico. Otra ventaja que SimMechanics posee, es la facilidad de cambiar los parámetros físicos de los modelos. Esto resulta ser muy conveniente y práctico en el momento de realizar varias pruebas para un mejor entendimiento y análisis de los sistemas mecánicos.

El desarrollo de este trabajo abre las puertas para futuros proyectos en donde se pueden implementar las variables de junta  $q_i$ , las velocidades angulares  $\dot{q}_i$  y las aceleraciones angulares  $\ddot{q}_i$  en un robot real para verificar experimentalmente los resultados. Así mismo proponer una estrategia de planeamiento de movimiento, para completar el movimiento de swing de la cadena articulada y por último establecer métricas de eficiencia energética y desempeño del movimiento.

## 5. BIBLIOGRAFÍA

- [1] Toshio Fukuda, Shigetaka Kojima, Kousuke Sekiyama y Yasuhisa Hasegawa, Noviembre 2006 , “Energy Efficient Swing-Back Control for Brachiation Robot”, Nagoya University, Nagoya: Japón.
- [2] Jun Nakanishi y Toshio Fukuda, Abril 2000, “A Leaping Maneuver for a Brachiating Robot”, Nagoya University, Nagoya: Japón.
- [3] Yasuhisa Hasegawa, Yoshikuni Ito y Toshio Fukuda, Abril 2000, “Behavior Coordination and its Modification on Brachiation-type Mobile Robot”.
- [4] Jun Nakanishi, Toshio Fukuda y Daniel E. Koditschek, Mayo 1999, “Brachiation on a Ladder with Irregular Intervals” , Nagoya University, Nagoya: Japón.
- [5] Jun Nakanishi, Toshio Fukuda y Daniel E. Koditschek, Mayo 1998 ,“Experimental Implementation of a “Target Dynamics” Controller on a Two-link Brachiating Robot” ”, Nagoya University, Nagoya: Japón.
- [6] Mark W. Spong, Seth Hutchinson, y M. Vidyasagar, 1989 “Robot Modeling and Control”, Primera edición.
- [7] Tipler Mosca, Paul A. Gene, “Física para la ciencia y la tecnología - volumen 1”, Quinta Edición, Editorial Reverte.
- [8] Ángel Franco García, La curva catenaria [en línea], [Fecha de consulta: Enero de 2011], disponible en: [http://www.sc.ehu.es/sbweb/fisica/solido/din\\_rotacion/catenaria/catenaria.htm#Referencias](http://www.sc.ehu.es/sbweb/fisica/solido/din_rotacion/catenaria/catenaria.htm#Referencias)
- [9] Paul Kunkel, Hanging With Galileo [en línea], [Fecha de consulta: Febrero de 2011], disponible en: <http://whistleralley.com/hanging/hanging.htm>.
- [10] ROBOTIS INC. [en línea], [Fecha de consulta: Septiembre de 2010], disponible en: [http://www.robotis.com/x/dynamixel\\_en](http://www.robotis.com/x/dynamixel_en).
- [11] Dan B. Marghitu. “Mechanisms and Robots Analysis with MATLAB”. Auburn University. Estados Unidos.
- [12] John J. Craig, “Introduction to robotics: Mechanics and control” Segunda Edición. Editorial Addison - Wesley.
- [13] Sean B. Andersson, “Discrete Approximations to Continuous Curves” Harvard University. USA
- [14] Hideki Kajima, Masahiro Doi, Yasuhisa Hasegawa y Toshio Fukuda, Abril 2005, “Energy Based Swing Control of a Brachiating Robot”, IEEE International Conference on Robotics and Automation, Barcelona: España.

- [15] Jun Nakanishi, Toshio Fukuda y Daniel E. Koditschek, 1999 ,“Brachiation on a Ladder with Irregular Intervals”, IEEE International Conference on Robotics and Automation, Estados Unidos: Detroit.
- [16] Toshido Fukuda, Yasuhisa Hasegawa, Masahiro Doi y Yoshihiko Asano, 2005 ,“Multi- Locomotion Robot - Energy-based Motion Control for Dexterous Brachiation -” IEEE International Conference on Robotics and Biomimetics.
- [17] Mireia Pascual, Jorge Traver, José Gibergans y Pablo Buenestado, Sept. 2002 - Enero 2003, “Regulación Automática con MATLAB”, Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona, Barcelona, < <http://usuarios.multimania.es/automatica/> > .
- [18] Ivette Morazzani, Dennis Hong, Derek Lahr, and Ping Ren, “Novel Tripedal Mobile Robot and Considerations for Gait Planning Strategies Based on Kinematics”, Robotics & Mechanisms Laboratory ,Virginia Tech, USA.
- [19] MathWorks [en línea], [Fecha de consulta: Agosto de 2010], disponible en: <http://www.mathworks.com/help/toolbox/phymod/mech/g5/f5-12147.html#brcjlig>

## 6. ANEXOS 1: ALGORITMOS DE MATLAB®

**Ecu. A1-1** Algoritmo de la cadena articulada de 2 módulos

```
clear all; clc; close all
syms t L1 L2 L3 m1 m2 m3 g D I1 I2 I3 H

q1 = sym('q1(t)');
q2 = sym('q2(t)');
q3 = sym('q3(t)');

%H es la misma que se uso en las especificaciones físicas

IC1 = m1*((H)^2+(L1)^2)/12;
I1 = IC1 + m1*(L1/2)^2;
I2 = m2*L2^2/12;

c1 = cos(q1); s1 = sin(q1);
c2 = cos(q2); s2 = sin(q2);
c3 = cos(q3); s3 = sin(q3);
c12 = cos(q1 + q2); s12 = sin(q1 + q2);
c123 = cos(q1 + q2 + q3); s123 = sin(q1 + q2 + q3);

lc1 = L1/2; lc2 = L2/2
diffq = [diff(q1,t);diff(q2,t)];

jVc1 = [-lc1*s1 0;lc1*c1 0;0 0];
Vc1 = jVc1*diffq;

jVc2 = [-L1*s1-lc2*s12 -lc2*s12;L1*c1+lc2*c12 lc2*c12;0 0];
Vc2 = jVc2*diffq;

omega1 = [0 0 diff(q1,t)];
omega2 = [0 0 diff(q1,t) + diff(q2,t)];

%Matriz de inercia

D = m1*jVc1.'*jVc1 +m2*jVc2.'*jVc2 + [I1+I2 I2;I2 I2];

Dsim = simple (D);

%Símbolos de Christoffel
%cij = 1/2*(deriv(dkj,qi)+deriv(dki,qj)-deriv(dij,qk));

for i=1:2
    for j=1:2
        for k=1:2
            d1=num2str(i);
            v1=num2str(j);
            b1=num2str(k);
            zi=horzcat('q',d1,'(t)');
```

```

        zj=horzcat('q',v1,'(t)');
        zk=horzcat('q',b1,'(t)');
        r1= 1/2*(deriv(Dsim(k,j),zi) + deriv(Dsim(k,i),zj)-
        (deriv(Dsim(i,j),zk)));
        g1=horzcat(d1,v1,b1);
        assignin('base',horzcat('c', g1), r1);
    end
end
end

P1 = m1*g*lc1*s1;
P2 = m2*g*(L1*s1 + lc2*s12);
P = P1 + P2;

phi1 = (deriv(P,q1));
phi2 = (deriv(P,q2));

%Ecuaciones de Euler-Lagrange
%Sum(dkj(q)*diff(qj,t,2),i) + Sum ( cijk(q)*diff(qi,t)*diff(qj,t)
,(i,j))
% + phik(q) = torque k, para k = 1, ..., n

sum1=0;
sum2=0;
for k=1:2
    for j=1:2
        v2=num2str(j);
        b2=num2str(k);
        zj2=horzcat('q',v2,'(t)');
        zk2=horzcat('q',b2,'(t)');
        sum1=Dsim(k,j)*diff(zj2,t,2)+sum1;
        for i=1:2
            d2=num2str(i);
            zi2=horzcat('q',d2,'(t)');
            r2= 1/2*(deriv(Dsim(k,j),zi2) + deriv(Dsim(k,i),zj2)-
            (deriv(Dsim(i,j),zk2)));
            sum2=r2* diff(zi2,t)*diff(zj2,t)+sum2;
        end
        end
        assignin('base',horzcat('torque',num2str(k)),
sum1+sum2+deriv(P,zk2));
        sum1=0;
        sum2=0;
    end
end

%El peso de cada módulo es 55 g
%la longitud de cada módulo es 50 mm

data = {L1 L2 m1 m2 g};
datn = {0.05 , 0.05, 0.055, 0.055, 9.81};

torqu1 = subs(torque1, data, datn);
torqu2 = subs(torque2, data, datn);

```



```

q1 = {diff(q1,t,2), diff(q2,t,2),...
diff(q1,t), diff(q2,t), q1, q2};
qf = {'ddq1', 'ddq2', 'x(2)', 'x(4)',...
      'x(1)', 'x(3)'};

torq1 = subs(torqu1, q1, qf);
torq2 = subs(torqu2, q1, qf);

sol = solve(torq1,torq2,'ddq1, ddq2');
tor1 = sol.ddq1;
tor2 = sol.ddq2;

dx2dt = char(tor1);
dx4dt = char(tor2);

fid = fopen('RR_RMCB2.m','w+');
fprintf(fid,'function dx = RR_RMCB2(t,x)\n');
fprintf(fid,'dx = zeros(4,1);\n');
fprintf(fid,'dx(1) = x(2);\n');
fprintf(fid,'dx(2) = ');
fprintf(fid,dx2dt);
fprintf(fid,';\n');
fprintf(fid,'dx(3) = x(4);\n');
fprintf(fid,'dx(4) = ');
fprintf(fid,dx4dt);
fprintf(fid,';');
fclose(fid); cd(pwd);

t0 = 0; tf = 5; time = [0 tf];
x0 = [0 0 0 0];

[t,xs] = ode45(@RR_RMCB2,time,x0);
x1 = xs(:,1);
x2 = xs(:,2);
x3 = xs(:,3);
x4 = xs(:,4);

subplot(2,1,1),plot(t,x1*180/pi,'r'),...
xlabel('t (s)'),ylabel('q1 (deg)'),grid,...
subplot(2,1,2),plot(t,x3*180/pi,'b'),...
xlabel('t (s)'),ylabel('q2 (deg)'),grid

```

### **Ecu. A1-2** Algoritmo de la cadena articulada de 5 módulos

```

clear all; clc; close all
syms t L1 L2 L3 L4 L5 L6 m1 m2 m3 m4 m5 m6 g D I1 I2 I3 I4 I5 I6 H

q1 = sym('q1(t)');
q2 = sym('q2(t)');
q3 = sym('q3(t)');
q4 = sym('q4(t)');
q5 = sym('q5(t)');
%H es la misma que se uso en las especificaciones físicas

```

```

IC1 = m1*((H)^2+(L1)^2)/12;
I1 = IC1 + m1*(L1/2)^2;
I2 = m2*L2^2/12;
I3 = m3*L3^2/12;
I4 = m4*L4^2/12;
I5 = m5*L5^2/12;

c1 = cos(q1); s1 = sin(q1);
c2 = cos(q2); s2 = sin(q2);
c3 = cos(q3); s3 = sin(q3);
c12 = cos(q1 + q2);s12 = sin(q1 + q2);
c123 = cos(q1 + q2 + q3);s123 = sin(q1 + q2 + q3);
c1234 = cos(q1 + q2 + q3 + q4);s1234 = sin(q1 + q2 + q3 + q4);
c12345 = cos(q1 + q2 + q3 + q4 + q5);s12345 = sin(q1 + q2 + q3 + q4 +
q5);

lc1 = L1/2;lc2 = L2/2;lc3 = L3/2; lc4 = L4/2;lc5 = L5/2;lc6 = L6/2;
diffq = [diff(q1,t);diff(q2,t);diff(q3,t);diff(q4,t);diff(q5,t)];

jVc1 = [-lc1*s1 0 0 0 0;lc1*c1 0 0 0 0;0 0 0 0 0];
Vc1 = jVc1*diffq;

jVc2 = [-L1*s1-lc2*s12 -lc2*s12 0 0 0;L1*c1+lc2*c12 lc2*c12 0 0 0;0 0 0
0 0];
Vc2 = jVc2*diffq;

jVc3 = [-L1*s1-L2*s12-lc3*s123 -L2*s12-lc3*s123 -lc3*s123 0
0;L1*c1+L2*c12+lc3*c123 L2*c12+lc3*c123 lc3*c123 0 0;0 0 0 0 0];
Vc3 = jVc3*diffq;

jVc4 = [-L1*s1-L2*s12-L3*s123-lc4*s1234 -L2*s12-L3*s123-lc4*s1234 -
L3*s123-lc4*s1234 -lc4*s1234 0;L1*c1+L2*c12+L3*c123+lc4*c1234
L2*c12+L3*c123+lc4*c1234 L3*c123+lc4*c1234 lc4*c1234 0;0 0 0 0 0];
Vc4 = jVc4*diffq;

jVc5 = [-L1*s1-L2*s12-L3*s123-L4*s1234-lc5*s12345 -L2*s12-L3*s123-
L4*s1234-lc5*s12345 -L3*s123-L4*s1234-lc5*s12345 -L4*s1234-lc5*s12345 -
lc5*s12345;L1*c1+L2*c12+L3*c123+L4*c1234+lc5*c12345
L2*c12+L3*c123+L4*c1234+lc5*c12345 L3*c123+L4*c1234+lc5*c12345
L4*c1234+lc5*c12345 lc5*c12345;0 0 0 0 0];
Vc5 = jVc5*diffq;

omega1 = [0 0 diff(q1,t)];
omega2 = [0 0 diff(q1,t) + diff(q2,t)];
omega3 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t)];
omega4 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t) + diff(q4,t)];
omega5 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t) + diff(q4,t) +
diff(q5,t)];

%Matriz de inercia

D = m1*jVc1.*jVc1 +m2*jVc2.*jVc2 + m3*jVc3.*jVc3 + m4*jVc4.*jVc4 +
m5*jVc5.*jVc5 + [I1+I2+I3+I4+I5 I2+I3+I4+I5 I3+I4+I5 I4+I5

```

```
I5;I2+I3+I4+I5 I2+I3+I4+I5 I3+I4+I5 I4+I5 I5;I3+I4+I5 I3+I4+I5 I3+I4+I5
I4+I5 I5;I4+I5 I4+I5 I4+I5 I4+I5 I5;I5 I5 I5 I5 I5];
```

```
Dsim = simple (D);
```

```
%Símbolos de Christoffel
```

```
%cij = 1/2*(deriv(dkj,qi)+deriv(dki,qj)-deriv(dij,qk));
```

```
for i=1:5
```

```
    for j=1:5
```

```
        for k=1:5
```

```
            d1=num2str(i);
```

```
            v1=num2str(j);
```

```
            b1=num2str(k);
```

```
            zi=horzcat('q',d1,'(t)');
```

```
            zj=horzcat('q',v1,'(t)');
```

```
            zk=horzcat('q',b1,'(t)');
```

```
            r1= 1/2*(deriv(Dsim(k,j),zi) + deriv(Dsim(k,i),zj)-
(deriv(Dsim(i,j),zk)));
```

```
            g1=horzcat(d1,v1,b1);
```

```
            assignin('base',horzcat('c', g1), r1);
```

```
        end
```

```
    end
```

```
end
```

```
P1 = m1*g*lc1*s1;
```

```
P2 = m2*g*(L1*s1 + lc2*s12);
```

```
P3 = m3*g*(L1*s1 + L2*s12 + lc3*s123);
```

```
P4 = m4*g*(L1*s1 + L2*s12 + L3*s123 + lc4*s1234);
```

```
P5 = m5*g*(L1*s1 + L2*s12 + L3*s123 + L4*s1234 + lc5*s12345);
```

```
P = P1 + P2 + P3 + P4 + P5;
```

```
%Ecuaciones de Euler-Lagrange
```

```
%Sum(dkj(q)*diff(qj,t,2),i) + Sum ( cij(k,q)*diff(qi,t)*diff(qj,t)
,(i,j))
```

```
% + phik(q) = torque k, para k = 1, ..., n
```

```
sum1=0;
```

```
sum2=0;
```

```
for k=1:5
```

```
    for j=1:5
```

```
        v2=num2str(j);
```

```
        b2=num2str(k);
```

```
        zj2=horzcat('q',v2,'(t)');
```

```
        zk2=horzcat('q',b2,'(t)');
```

```
        sum1=Dsim (k,j)*diff(zj2,t,2)+sum1;
```

```
            for i=1:5
```

```
                d2=num2str(i);
```

```
                zi2=horzcat('q',d2,'(t)');
```

```
                r2= 1/2*(deriv(Dsim(k,j),zi2) + deriv(Dsim(k,i),zj2)-
(deriv(Dsim(i,j),zk2)));
```

```

        sum2=r2* diff(zi2,t)*diff(zj2,t)+sum2;
    end
    end
    assignin('base',horzcat('torque',num2str(k)),
    sum1+sum2+deriv(P,zk2));
    sum1=0;
    sum2=0;
end

data = {L1 L2 L3 L4 L5 m1 m2 m3 m4 m5 g};
datn = {0.05, 0.05, 0.05, 0.05, 0.05, 0.055, 0.055, 0.055, 0.055,
0.055, 9.81};

torqu1 = subs(torque1, data, datn);
torqu2 = subs(torque2, data, datn);
torqu3 = subs(torque3, data, datn);
torqu4 = subs(torque4, data, datn);
torqu5 = subs(torque5, data, datn);

q1 = {diff(q1,t,2), diff(q2,t,2), diff(q3,t,2), diff(q4,t,2),
diff(q5,t,2), ...
diff(q1,t), diff(q2,t),diff(q3,t), diff(q4,t),diff(q5,t), q1, q2, q3,
q4, q5};
qf = {'ddq1', 'ddq2', 'ddq3', 'ddq4', 'ddq5', 'x(2)', 'x(4)', 'x(6)',
'x(8)', 'x(10)', ...
'x(1)', 'x(3)', 'x(5)', 'x(7)', 'x(9)'};

torq1 = subs(torqu1, q1, qf);
torq2 = subs(torqu2, q1, qf);
torq3 = subs(torqu3, q1, qf);
torq4 = subs(torqu4, q1, qf);
torq5 = subs(torqu5, q1, qf);

sol = solve(torq1,torq2,torq3,torq4,torq5,'ddq1, ddq2, ddq3, ddq4,
ddq5');
tor1 = sol.ddq1;
tor2 = sol.ddq2;
tor3 = sol.ddq3;
tor4 = sol.ddq4;
tor5 = sol.ddq5;

dx2dt = char(tor1);
dx4dt = char(tor2);
dx6dt = char(tor3);
dx8dt = char(tor4);
dx10dt = char(tor5);

fid = fopen('RRRRR_RMCB5.m', 'w+');
fprintf(fid, 'function dx = RRRRR_RMCB5(t,x)\n');
fprintf(fid, 'dx = zeros(10,1);\n');
fprintf(fid, 'PI = pi;\n');
fprintf(fid, 'dx(1) = x(2);\n');
fprintf(fid, 'dx(2) = ');
fprintf(fid, dx2dt);

```

```

fprintf(fid, '\n');
fprintf(fid, 'dx(3) = x(4); \n');
fprintf(fid, 'dx(4) = ');
fprintf(fid, dx4dt);
fprintf(fid, '\n');
fprintf(fid, 'dx(5) = x(6); \n');
fprintf(fid, 'dx(6) = ');
fprintf(fid, dx6dt);
fprintf(fid, '\n');
fprintf(fid, 'dx(7) = x(8); \n');
fprintf(fid, 'dx(8) = ');
fprintf(fid, dx8dt);
fprintf(fid, '\n');
fprintf(fid, 'dx(9) = x(10); \n');
fprintf(fid, 'dx(10) = ');
fprintf(fid, dx10dt);
fprintf(fid, '\n');
fclose(fid); cd(pwd);

t0 = 0; tf = 5; time = [0 tf];
x0 = [0 0 0 0 0 0 0 0 0 0];

[t, xs] = ode45(@RRRRR_RMCS5, time, x0);
x1 = xs(:,1);
x2 = xs(:,2);
x3 = xs(:,3);
x4 = xs(:,4);
x5 = xs(:,5);
x6 = xs(:,6);
x7 = xs(:,7);
x8 = xs(:,8);
x9 = xs(:,9);
x10 = xs(:,10);

subplot(5,1,1), plot(t, x1*180/pi, 'r'), ...
xlabel('t (s)'), ylabel('q1 (deg)'), grid, ...
subplot(5,1,2), plot(t, x3*180/pi, 'b'), ...
xlabel('t (s)'), ylabel('q2 (deg)'), grid
subplot(5,1,3), plot(t, x5*180/pi, 'r'), ...
xlabel('t (s)'), ylabel('q3 (deg)'), grid, ...
subplot(5,1,4), plot(t, x7*180/pi, 'b'), ...
xlabel('t (s)'), ylabel('q4 (deg)'), grid
subplot(5,1,5), plot(t, x9*180/pi, 'r'), ...
xlabel('t (s)'), ylabel('q5 (deg)'), grid

```

**Ecu. A1-3** Algoritmo cadena articulada de 6 módulos

```

clear all; clc; close all
syms t L1 L2 L3 L4 L5 L6 m1 m2 m3 m4 m5 m6 g D I1 I2 I3 I4 I5 I6 H

q1 = sym('q1(t)');
q2 = sym('q2(t)');
q3 = sym('q3(t)');

```

```

q4 = sym('q4(t)');
q5 = sym('q5(t)');
q6 = sym('q6(t)');
%H es la misma que se puso en las especificaciones físicas

IC1 = m1*((H)^2+(L1)^2)/12;
I1 = IC1 + m1*(L1/2)^2;
I2 = m2*L2^2/12;
I3 = m3*L3^2/12;
I4 = m4*L4^2/12;
I5 = m5*L5^2/12;
I6 = m6*L6^2/12;

c1 = cos(q1); s1 = sin(q1);
c2 = cos(q2); s2 = sin(q2);
c3 = cos(q3); s3 = sin(q3);
c12 = cos(q1 + q2);s12 = sin(q1 + q2);
c123 = cos(q1 + q2 + q3);s123 = sin(q1 + q2 + q3);
c1234 = cos(q1 + q2 + q3 + q4);s1234 = sin(q1 + q2 + q3 + q4);
c12345 = cos(q1 + q2 + q3 + q4 + q5);s12345 = sin(q1 + q2 + q3 + q4 +
q5);
c123456 = cos(q1 + q2 + q3 + q4 + q5 + q6);s123456 = sin(q1 + q2 + q3 +
q4 + q5 + q6);

lc1 = L1/2;lc2 = L2/2;lc3 = L3/2; lc4 = L4/2;lc5 = L5/2;lc6 = L6/2;
diffq =
[diff(q1,t);diff(q2,t);diff(q3,t);diff(q4,t);diff(q5,t);diff(q6,t)];

jVc1 = [-lc1*s1 0 0 0 0 0;lc1*c1 0 0 0 0 0;0 0 0 0 0 0];
Vc1 = jVc1*diffq;

jVc2 = [-L1*s1-lc2*s12 -lc2*s12 0 0 0 0;L1*c1+lc2*c12 lc2*c12 0 0 0 0;0
0 0 0 0];
Vc2 = jVc2*diffq;

jVc3 = [-L1*s1-L2*s12-lc3*s123 -L2*s12-lc3*s123 -lc3*s123 0 0
0;L1*c1+L2*c12+lc3*c123 L2*c12+lc3*c123 lc3*c123 0 0 0;0 0 0 0 0 0];
Vc3 = jVc3*diffq;

jVc4 = [-L1*s1-L2*s12-L3*s123-lc4*s1234 -L2*s12-L3*s123-lc4*s1234 -
L3*s123-lc4*s1234 -lc4*s1234 0 0;L1*c1+L2*c12+L3*c123+lc4*c1234
L2*c12+L3*c123+lc4*c1234 L3*c123+lc4*c1234 lc4*c1234 0 0;0 0 0 0 0 0];
Vc4 = jVc4*diffq;

jVc5 = [-L1*s1-L2*s12-L3*s123-L4*s1234-lc5*s12345 -L2*s12-L3*s123-
L4*s1234-lc5*s12345 -L3*s123-L4*s1234-lc5*s12345 -L4*s1234-lc5*s12345 -
lc5*s12345 0;L1*c1+L2*c12+L3*c123+L4*c1234+lc5*c12345
L2*c12+L3*c123+L4*c1234+lc5*c12345 L3*c123+L4*c1234+lc5*c12345
L4*c1234+lc5*c12345 lc5*c12345 0;0 0 0 0 0 0];
Vc5 = jVc5*diffq;

jVc6 = [-L1*s1-L2*s12-L3*s123-L4*s1234-L5*s12345-lc6*s123456 -L2*s12-
L3*s123-L4*s1234-L5*s12345-lc6*s123456 -L3*s123-L4*s1234-L5*s12345-
lc6*s123456 -L4*s1234-L5*s12345-lc6*s123456 -L5*s12345-lc6*s123456 -

```

```

lc6*s123456;L1*c1+L2*c12+L3*c123+L4*c1234+L5*c12345+lc6*c123456
L2*c12+L3*c123+L4*c1234+L5*c12345+lc6*c123456
L3*c123+L4*c1234+L5*c12345+lc6*c123456 L4*c1234+L5*c12345+lc6*c123456
L5*c12345+lc6*c123456 lc6*c123456;0 0 0 0 0 0];
Vc6 = jVc6*diffq;

omega1 = [0 0 diff(q1,t)];
omega2 = [0 0 diff(q1,t) + diff(q2,t)];
omega3 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t)];
omega4 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t) + diff(q4,t)];
omega5 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t) + diff(q4,t) +
diff(q5,t)];
omega6 = [0 0 diff(q1,t) + diff(q2,t) + diff(q3,t) + diff(q4,t) +
diff(q5,t) + diff(q6,t)];

D = m1*jVc1.*jVc1 +m2*jVc2.*jVc2 + m3*jVc3.*jVc3 + m4*jVc4.*jVc4 +
m5*jVc5.*jVc5 + m6*jVc6.*jVc6 + [I1+I2+I3+I4+I5+I6 I2+I3+I4+I5+I6
I3+I4+I5+I6 I4+I5+I6 I5+I6 I6;I2+I3+I4+I5+I6 I2+I3+I4+I5+I6 I3+I4+I5+I6
I4+I5+I6 I5+I6 I6;I3+I4+I5+I6 I3+I4+I5+I6 I3+I4+I5+I6 I4+I5+I6 I5+I6
I6;I4+I5+I6 I4+I5+I6 I4+I5+I6 I4+I5+I6 I5+I6 I6;I5+I6 I5+I6 I5+I6 I5+I6
I5+I6 I6;I6 I6 I6 I6 I6 I6];

Dsim = simple (D);

%Símbolos de Christoffel
%cij_k = 1/2*(deriv(dkj,qi)+deriv(dki,qj)-deriv(dij,qk));

for i=1:6
    for j=1:6
        for k=1:6
            d1=num2str(i);
            v1=num2str(j);
            b1=num2str(k);
            zi=horzcat('q',d1,'(t)');
            zj=horzcat('q',v1,'(t)');
            zk=horzcat('q',b1,'(t)');
            r1= 1/2*(deriv(Dsim(k,j),zi) + deriv(Dsim(k,i),zj)-
(deriv(Dsim(i,j),zk)));
            g1=horzcat(d1,v1,b1);
            assignin('base',horzcat('c', g1), r1);

                end
            end
        end
    end
end

P1 = m1*g*lc1*s1;
P2 = m2*g*(L1*s1 + lc2*s12);
P3 = m3*g*(L1*s1 + L2*s12 + lc3*s123);
P4 = m4*g*(L1*s1 + L2*s12 + L3*s123 + lc4*s1234);
P5 = m5*g*(L1*s1 + L2*s12 + L3*s123 + L4*s1234 + lc5*s12345);
P6 = m6*g*(L1*s1 + L2*s12 + L3*s123 + L4*s1234 + L5*s12345 +
lc6*s123456);

```

```

P = P1 + P2 + P3 + P4 + P5 + P6;

%Ecuaciones de Euler-Lagrange
%Sum(dkj(q)*diff(qj,t,2),i) + Sum ( cijk(q)*diff(qi,t)*diff(qj,t)
,(i,j))
% + phik(q) = torque k, para k = 1, ..., n

sum1=0;
sum2=0;
for k=1:6
    for j=1:6
        v2=num2str(j);
        b2=num2str(k);
        zj2=horzcat('q',v2,'(t)');
        zk2=horzcat('q',b2,'(t)');
        sum1=Dsim(k,j)*diff(zj2,t,2)+sum1;
        for i=1:6
            d2=num2str(i);
            zi2=horzcat('q',d2,'(t)');
            r2= 1/2*(deriv(Dsim(k,j),zi2) + deriv(Dsim(k,i),zj2)-
(deriv(Dsim(i,j),zk2)));
            sum2=r2* diff(zi2,t)*diff(zj2,t)+sum2;
        end
        end
        assignin('base',horzcat('torque',num2str(k)),
sum1+sum2+deriv(P,zk2));
        sum1=0;
        sum2=0;
end

data = {L1 L2 L3 L4 L5 L6 m1 m2 m3 m4 m5 m6 g};
datn = {0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.055, 0.055, 0.055, 0.055,
0.055, 0.055, 9.81};

torqu1 = subs(torque1, data, datn);
torqu2 = subs(torque2, data, datn);
torqu3 = subs(torque3, data, datn);
torqu4 = subs(torque4, data, datn);
torqu5 = subs(torque5, data, datn);
torqu6 = subs(torque6, data, datn);

q1 = {diff(q1,t,2), diff(q2,t,2), diff(q3,t,2), diff(q4,t,2),
diff(q5,t,2), diff(q6,t,2) ...
diff(q1,t), diff(q2,t),diff(q3,t), diff(q4,t),diff(q5,t), diff(q6,t),
q1, q2, q3, q4, q5, q6};
qf = {'ddq1', 'ddq2', 'ddq3', 'ddq4', 'ddq5', 'ddq6', 'x(2)',
'x(4)', 'x(6)', 'x(8)', 'x(10)', 'x(12)', ...
'x(1)', 'x(3)', 'x(5)', 'x(7)', 'x(9)', 'x(11)'};

torq1 = subs(torqu1, q1, qf);
torq2 = subs(torqu2, q1, qf);
torq3 = subs(torqu3, q1, qf);
torq4 = subs(torqu4, q1, qf);
torq5 = subs(torqu5, q1, qf);

```



```

torq6 = subs(torqu6, q1, qf);

sol = solve(torq1,torq2,torq3,torq4,torq5,torq6,'ddq1, ddq2, ddq3,
ddq4, ddq5, ddq6');
tor1 = sol.ddq1;
tor2 = sol.ddq2;
tor3 = sol.ddq3;
tor4 = sol.ddq4;
tor5 = sol.ddq5;
tor6 = sol.ddq6;

dx2dt = char(tor1);
dx4dt = char(tor2);
dx6dt = char(tor3);
dx8dt = char(tor4);
dx10dt = char(tor5);
dx12dt = char(tor6);

fid = fopen('RRRRRR_RMCB6.m','w+');
fprintf(fid,'function dx = RRRRRR_RMCB6(t,x)\n');
fprintf(fid,'dx = zeros(12,1);\n');
fprintf(fid,'PI = pi;\n');
fprintf(fid,'dx(1) = x(2);\n');
fprintf(fid,'dx(2) = ');
fprintf(fid,dx2dt);
fprintf(fid,';\n');
fprintf(fid,'dx(3) = x(4);\n');
fprintf(fid,'dx(4) = ');
fprintf(fid,dx4dt);
fprintf(fid,';\n');
fprintf(fid,'dx(5) = x(6);\n');
fprintf(fid,'dx(6) = ');
fprintf(fid,dx6dt);
fprintf(fid,';\n');
fprintf(fid,'dx(7) = x(8);\n');
fprintf(fid,'dx(8) = ');
fprintf(fid,dx8dt);
fprintf(fid,';\n');
fprintf(fid,'dx(9) = x(10);\n');
fprintf(fid,'dx(10) = ');
fprintf(fid,dx10dt);
fprintf(fid,';\n');
fprintf(fid,'dx(11) = x(12);\n');
fprintf(fid,'dx(12) = ');
fprintf(fid,dx12dt);
fprintf(fid,';\n');
fclose(fid); cd(pwd);

t0 = 0; tf = 10; time = [0 tf];
x0 = [0 0 0 0 0 0 0 0 0 0 0 0];

[t,xs] = ode45(@RRRRRR_RMCB6,time,x0);
x1 = xs(:,1);
x2 = xs(:,2);

```

```

x3 = xs(:,3);
x4 = xs(:,4);
x5 = xs(:,5);
x6 = xs(:,6);
x7 = xs(:,7);
x8 = xs(:,8);
x9 = xs(:,9);
x10 = xs(:,10);
x11 = xs(:,11);
x12 = xs(:,12);

subplot(6,1,1),plot(t,x1*180/pi,'r'),...
xlabel('t (s)'),ylabel('q1 (deg)'),grid,...
subplot(6,1,2),plot(t,x3*180/pi,'b'),...
xlabel('t (s)'),ylabel('q2 (deg)'),grid
subplot(6,1,3),plot(t,x5*180/pi,'r'),...
xlabel('t (s)'),ylabel('q3 (deg)'),grid,...
subplot(6,1,4),plot(t,x7*180/pi,'b'),...
xlabel('t (s)'),ylabel('q4 (deg)'),grid
subplot(6,1,5),plot(t,x9*180/pi,'r'),...
xlabel('t (s)'),ylabel('q5 (deg)'),grid,...
subplot(6,1,6),plot(t,x11*180/pi,'b'),...
xlabel('t (s)'),ylabel('q6 (deg)'),grid

```

## 7. ANEXOS 2: GRÁFICAS Y TABLAS

### Cadena Articulada de tres Módulos

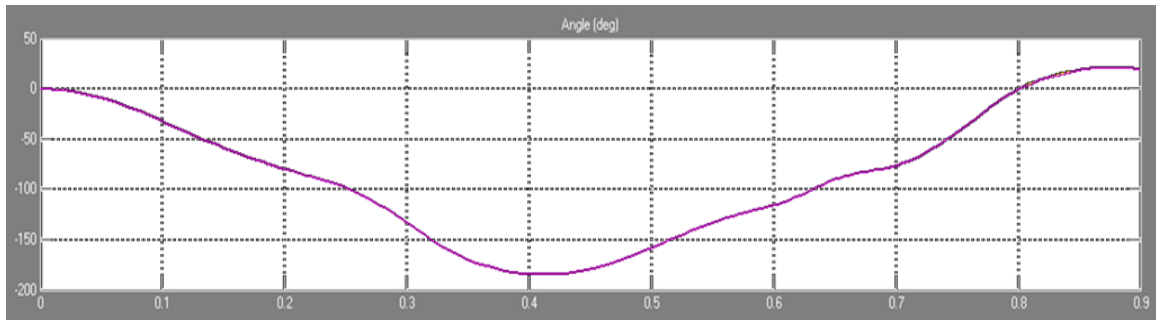


Figura 41. Comparación de  $q_1$  para la cadena articulada de 3 módulos

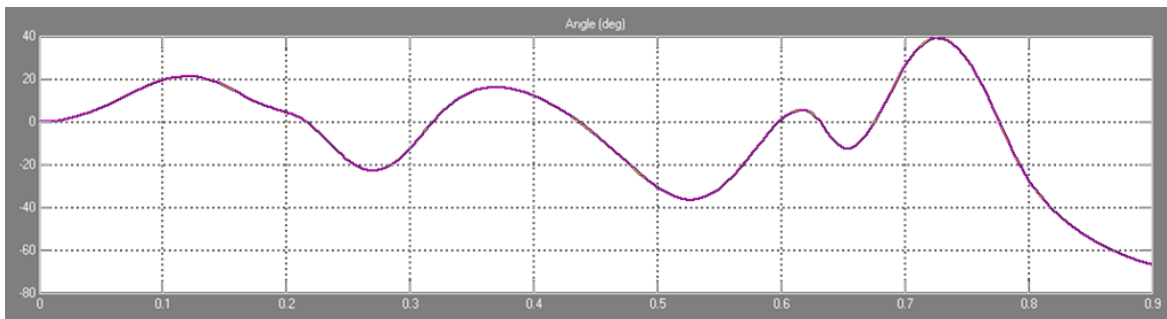


Figura 42. Comparación de  $q_2$  para la cadena articulada de 3 módulos

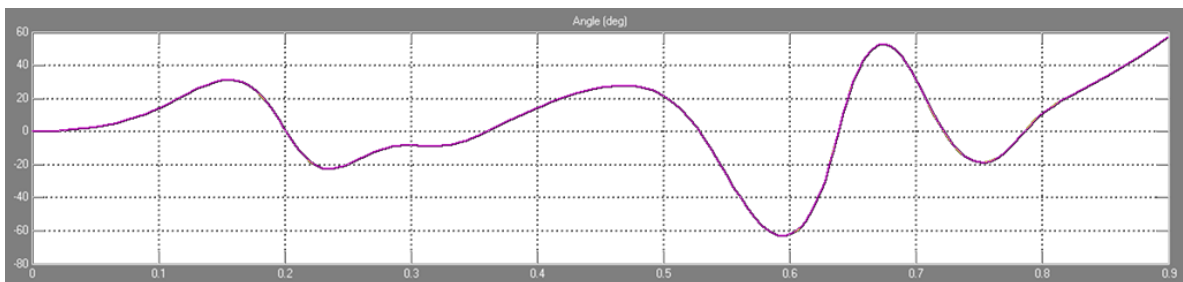


Figura 43. Comparación de  $q_3$  para la cadena articulada de 3 módulos

### Cadena Articulada de cuatro Módulos

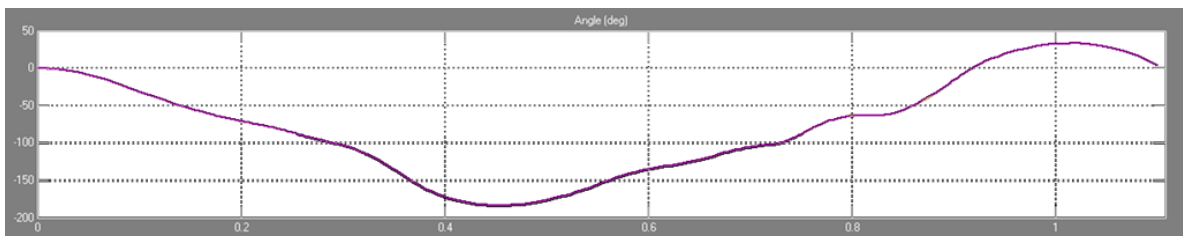
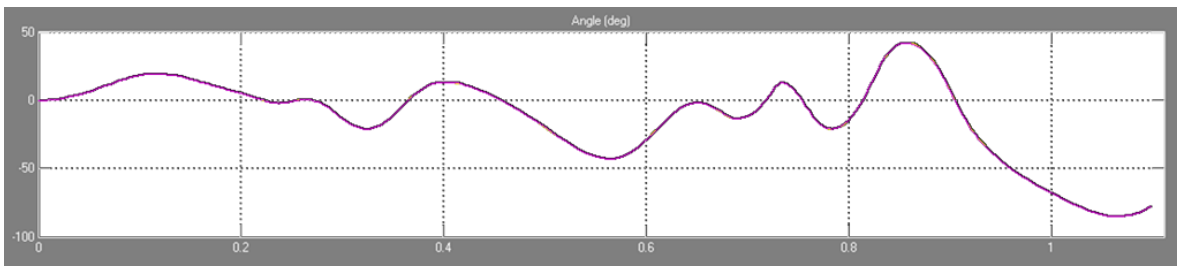
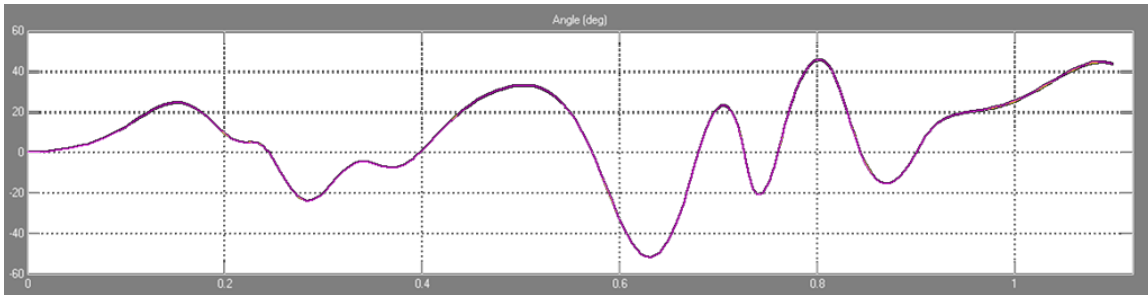


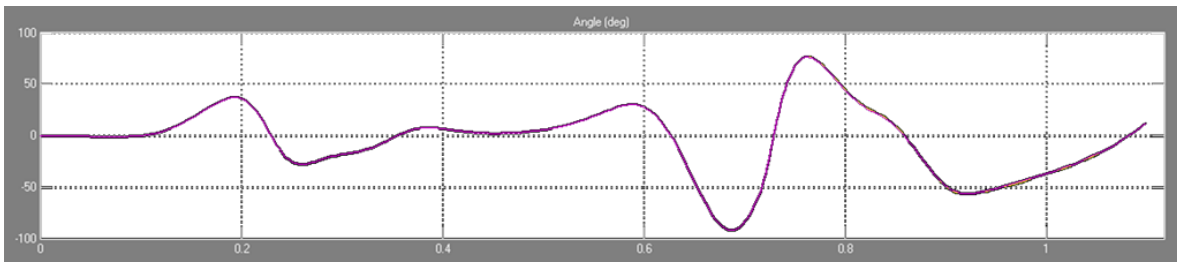
Figura 44. Comparación de  $q_1$  para la cadena articulada de 4 módulos



**Figura 45.** Comparación de  $q_2$  para la cadena articulada de 4 módulos

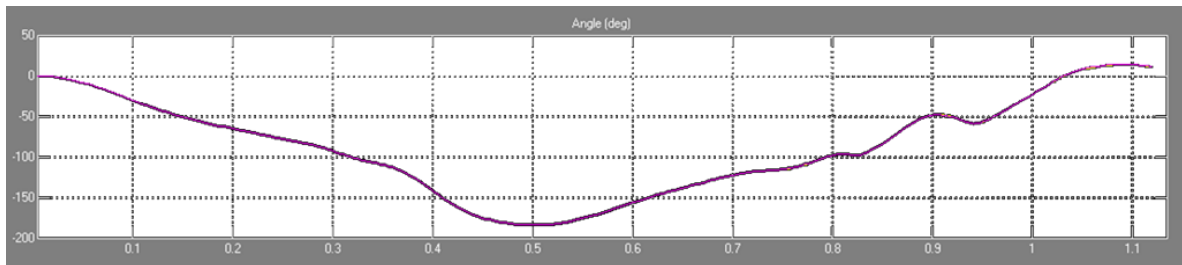


**Figura 46.** Comparación de  $q_3$  para la cadena articulada de 4 módulos

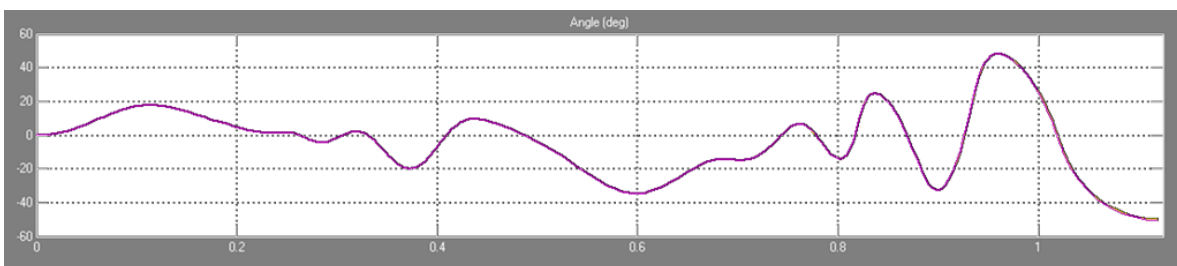


**Figura 47.** Comparación de  $q_4$  para la cadena articulada de 4 módulos

### Cadena Articulada de cinco Módulos



**Figura 48.** Comparación de  $q_1$  para la cadena articulada de 5 módulos



**Figura 49.** Comparación de  $q_2$  para la cadena articulada de 5 módulos

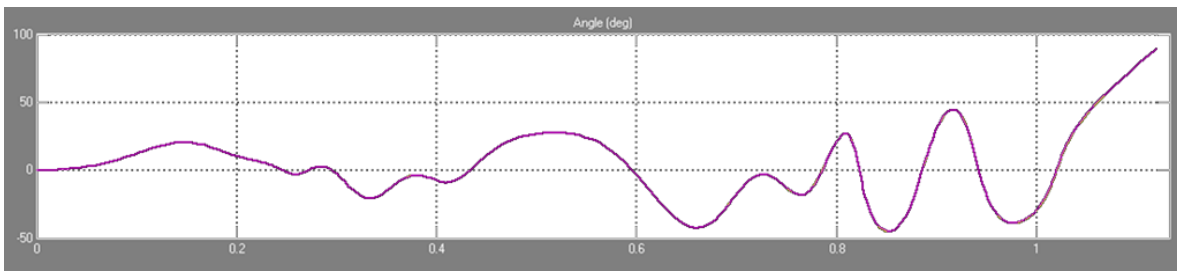


Figura 50.Comparación de  $q_3$  para la cadena articulada de 5 módulos

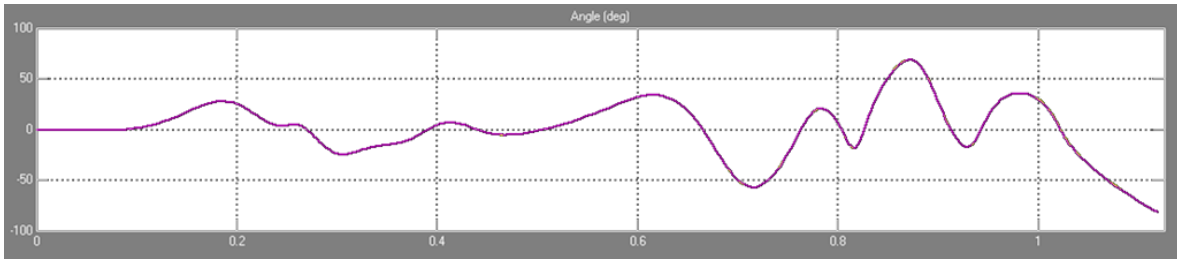


Figura 51.Comparación de  $q_4$  para la cadena articulada de 5 módulos

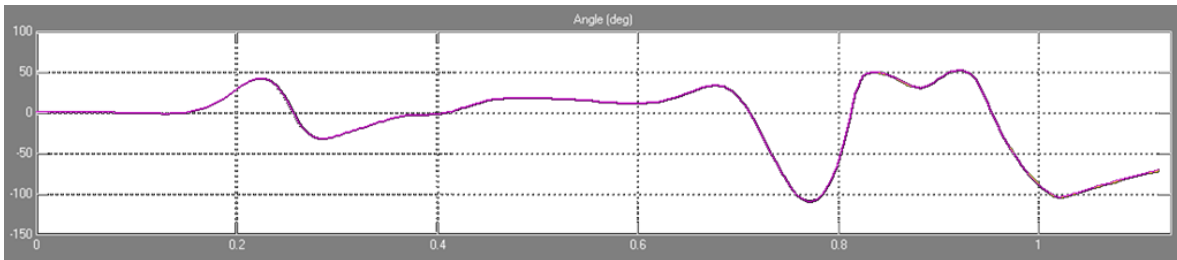


Figura 52.Comparación de  $q_5$  para la cadena articulada de 5 módulos

Tabla con diferentes posiciones iniciales

C	Tiempo (s)	Último módulo	Centro de gravedad	Distancia d (m)
		Posición en y (m)	Posición en y (m)	
3	0.4200	-0.0931	-0.1312	0.1572
4	0.4200	-0.1692	-0.1409	0.1866
5	0.4200	-0.2091	-0.1483	0.2113
6	0.4200	-0.2362	-0.1532	0.2322
7	0.4200	-0.2560	-0.1557	0.2501
8	0.4200	-0.2702	-0.1556	0.2656
9	0.4200	-0.2796	-0.1542	0.2791
10	0.4200	-0.2855	-0.1523	0.2908
11	0.4200	-0.2889	-0.1505	0.3012
12	0.4200	-0.2905	-0.1488	0.3103
13	0.4200	-0.2909	-0.1472	0.3183
14	0.4200	-0.2906	-0.1459	0.3255
15	0.4200	-0.2897	-0.1447	0.3318
16	0.4200	-0.2884	-0.1436	0.3374
17	0.4200	-0.2869	-0.1426	0.3425
18	0.4200	-0.2853	-0.1418	0.3470
19	0.4200	-0.2835	-0.1410	0.3511
20	0.4200	-0.2818	-0.1403	0.3548
21	0.4200	-0.2800	-0.1397	0.3581

C	Tiempo (s)	Último módulo	Centro de gravedad	Distancia d (m)
		Posición en y (m)	Posición en y (m)	
22	0.4200	-0.2782	-0.1391	0.3611
23	0.4200	-0.2765	-0.1385	0.3638
24	0.4200	-0.2748	-0.1380	0.3663
25	0.4200	-0.2731	-0.1375	0.3685
26	0.4200	-0.2715	-0.1370	0.3706
27	0.4200	-0.2700	-0.1366	0.3725
28	0.4200	-0.2685	-0.1362	0.3742
29	0.4200	-0.2670	-0.1358	0.3758
30	0.4200	-0.2657	-0.1354	0.3772
31	0.4200	-0.2643	-0.1351	0.3786
32	0.4200	-0.2630	-0.1347	0.3798
33	0.4200	-0.2618	-0.1344	0.3810
34	0.4200	-0.2606	-0.1341	0.3821
35	0.4200	-0.2595	-0.1338	0.3830
36	0.4200	-0.2584	-0.1336	0.3840
37	0.4200	-0.2574	-0.1333	0.3848
38	0.4200	-0.2564	-0.1331	0.3856
39	0.4200	-0.2554	-0.1328	0.3864
40	0.4200	-0.2545	-0.1326	0.3871
41	0.4200	-0.2537	-0.1324	0.3877
42	0.4200	-0.2528	-0.1322	0.3883
43	0.4200	-0.2528	-0.1320	0.3889
44	0.4200	-0.2512	-0.1318	0.3895
45	0.4200	-0.2505	-0.1316	0.3900
46	0.4200	-0.2498	-0.1315	0.3904
47	0.4200	-0.2491	-0.1313	0.3909
48	0.4200	-0.2484	-0.1311	0.3913
49	0.4200	-0.2478	-0.1310	0.3917
50	0.4200	-0.2471	-0.1308	0.3921
51	0.4200	-0.2465	-0.1307	0.3925
52	0.4200	-0.2460	-0.1305	0.3928
53	0.4200	-0.2454	-0.1304	0.3931
54	0.4200	-0.2449	-0.1303	0.3934
55	0.4200	-0.2444	-0.1301	0.3937
60	0.4200	-0.2420	-0.1296	0.3950
65	0.4200	-0.2401	-0.1291	0.3960
70	0.4200	-0.2384	-0.1287	0.3968

Tabla de anexos I. Posiciones iniciales de la cadena articulada

## VARIABLES DE JUNTURA PARA LA CADENA ARTICULADA DE 6 MÓDULOS

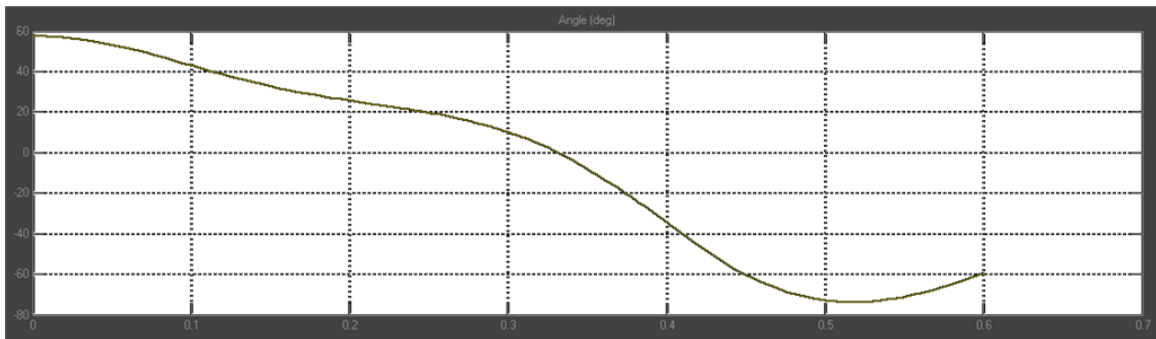


Figura 53. Variable de junta  $q_1$  para la cadena articulada de 6 módulos

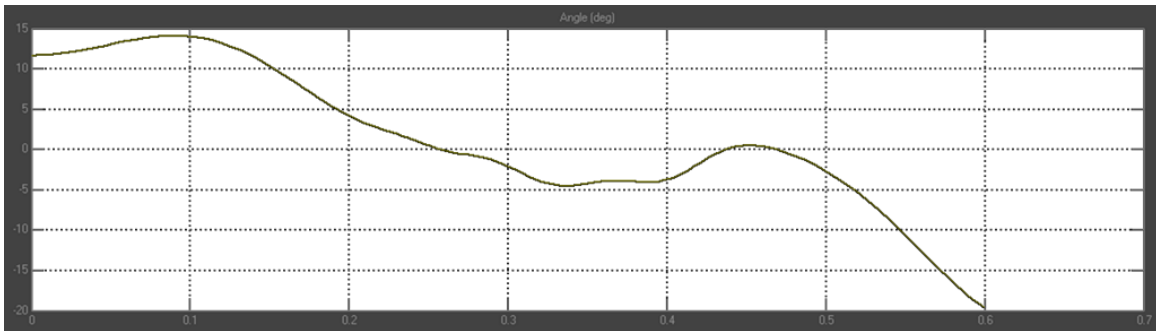


Figura 54. Variable de junta  $q_2$  para la cadena articulada de 6 módulos

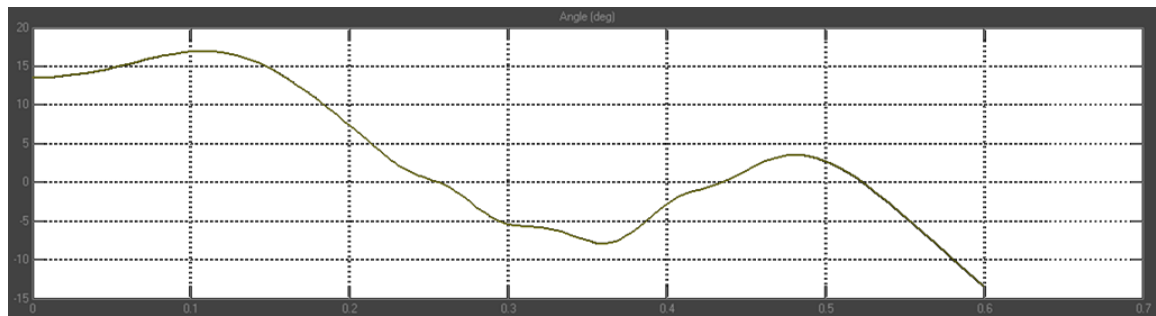


Figura 55. Variable de junta  $q_3$  para la cadena articulada de 6 módulos

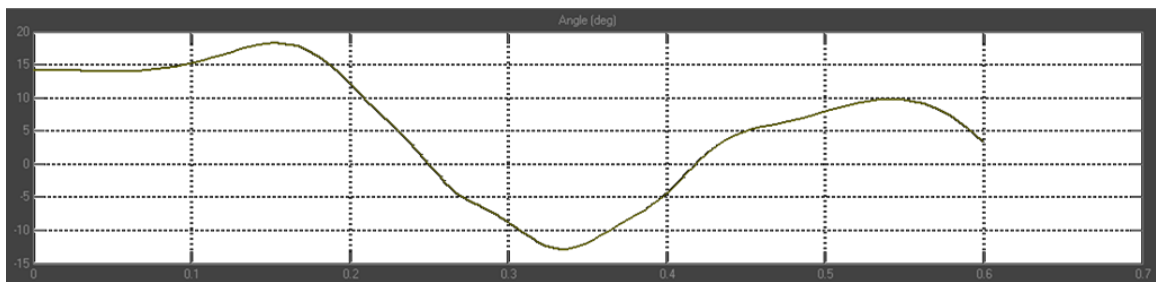


Figura 56. Variable de junta  $q_4$  para la cadena articulada de 6 módulos

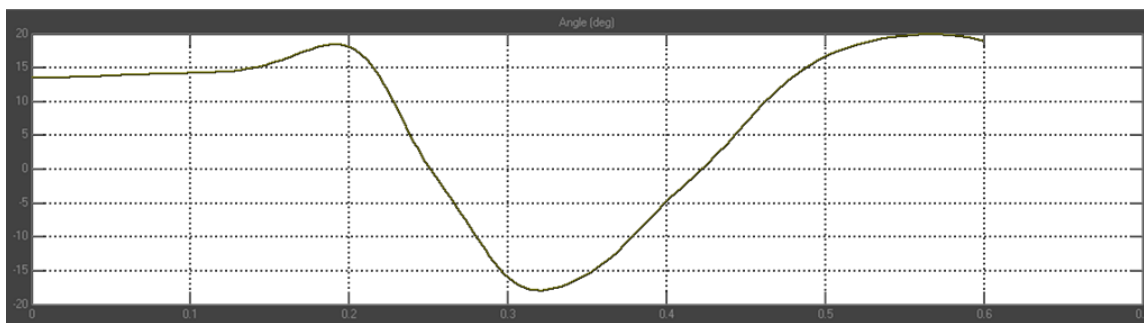


Figura 57. Variable de junta  $q_5$  para la cadena articulada de 6 módulos

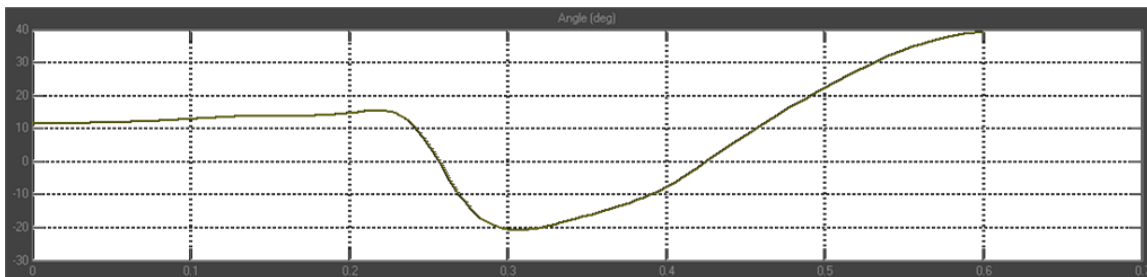


Figura 58. Variable de junta  $q_6$  para la cadena articulada de 6 módulos

### Velocidades angulares para la cadena articulada de 6 módulos

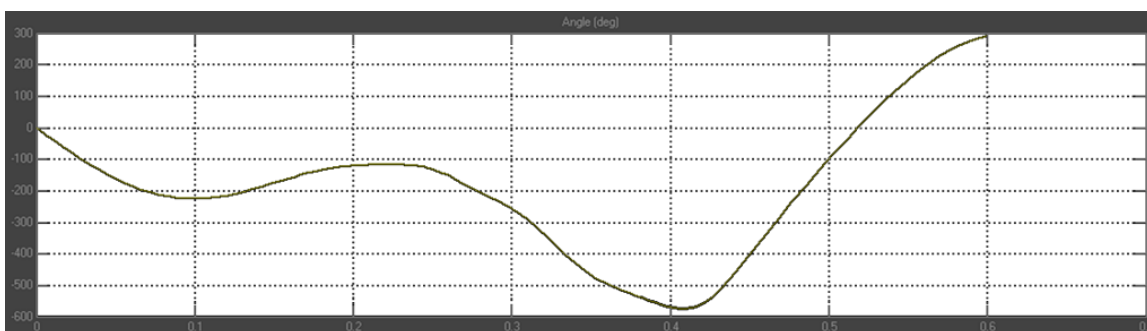


Figura 59. Velocidad angular  $\dot{q}_1$  para la cadena articulada de 6 módulos

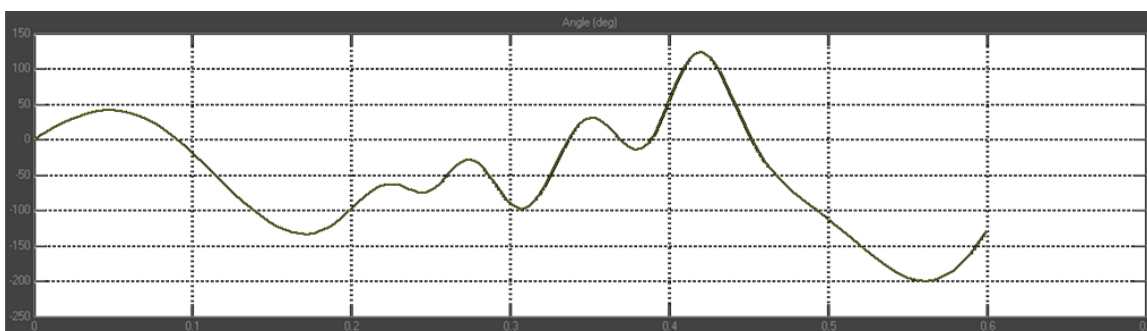


Figura 60. Velocidad angular  $\dot{q}_2$  para la cadena articulada de 6 módulos



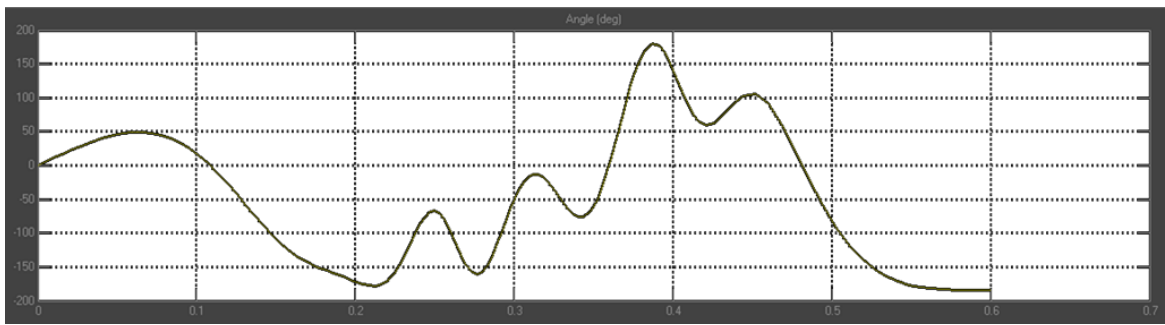


Figura 61. Velocidad angular  $\dot{q}_3$  para la cadena articulada de 6 módulos

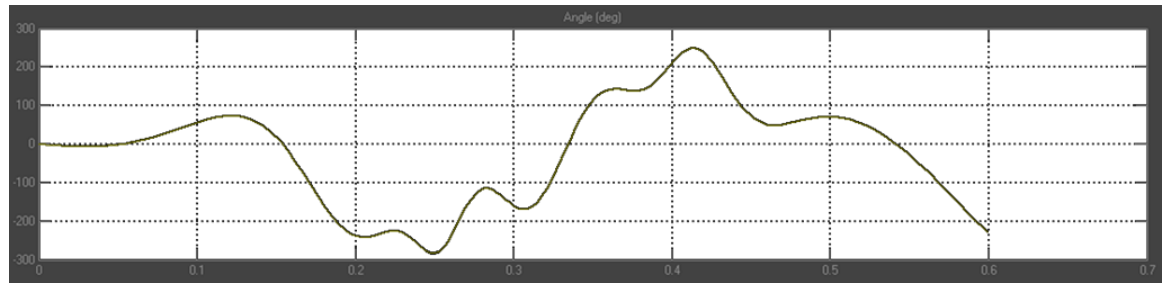


Figura 62. Velocidad angular  $\dot{q}_4$  para la cadena articulada de 6 módulos

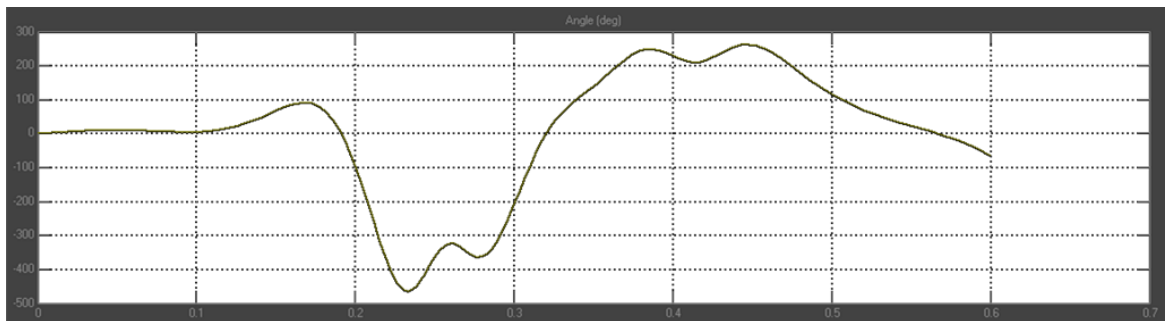


Figura 63. Velocidad angular  $\dot{q}_5$  para la cadena articulada de 6 módulos

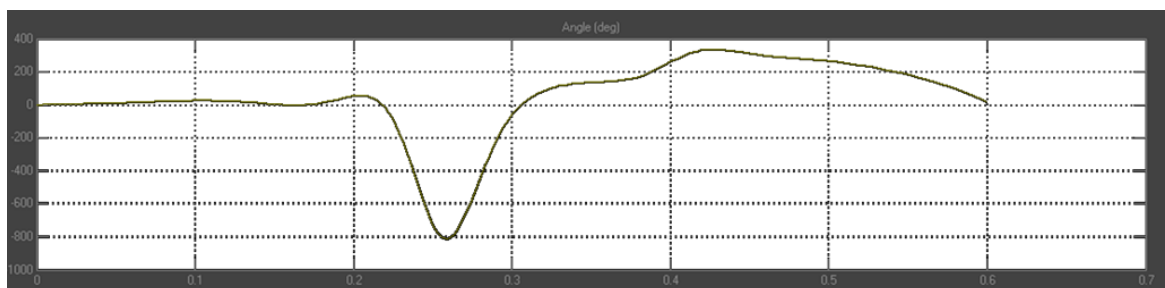


Figura 64. Velocidad angular  $\dot{q}_6$  para la cadena articulada de 6 módulos

## Acercaciones angulares para la cadena articulada de 6 módulos

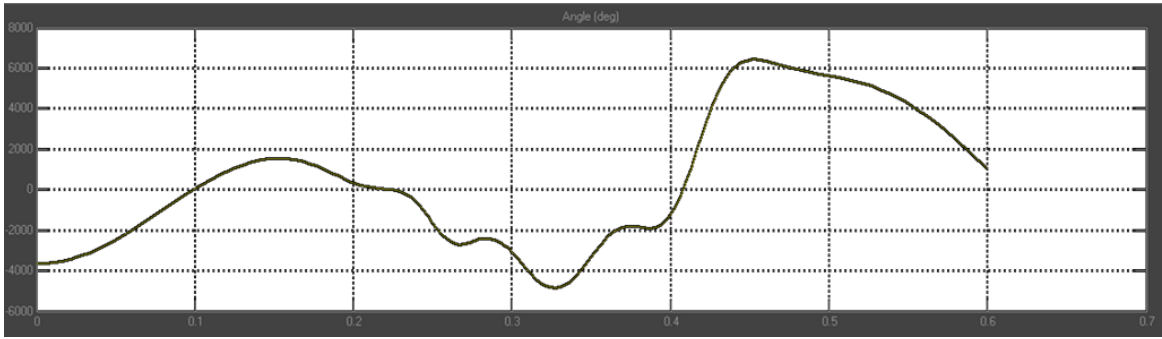


Figura 65. Aceleración angular  $\ddot{q}_1$  para la cadena articulada de 6 módulos

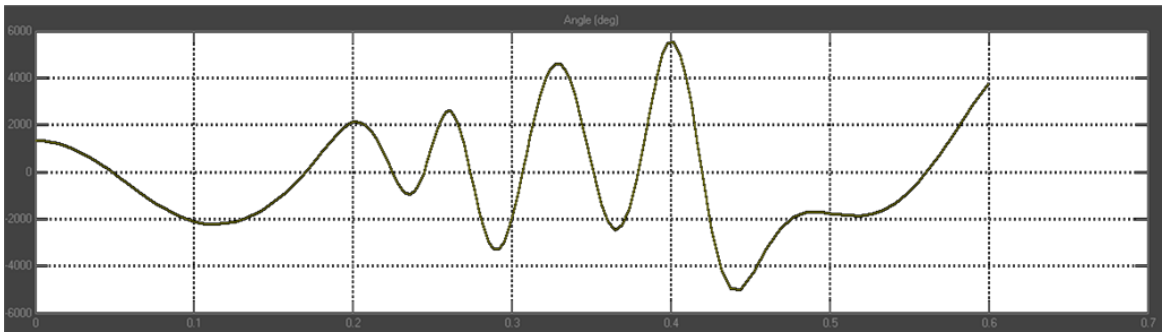


Figura 66. Aceleración angular  $\ddot{q}_2$  para la cadena articulada de 6 módulos

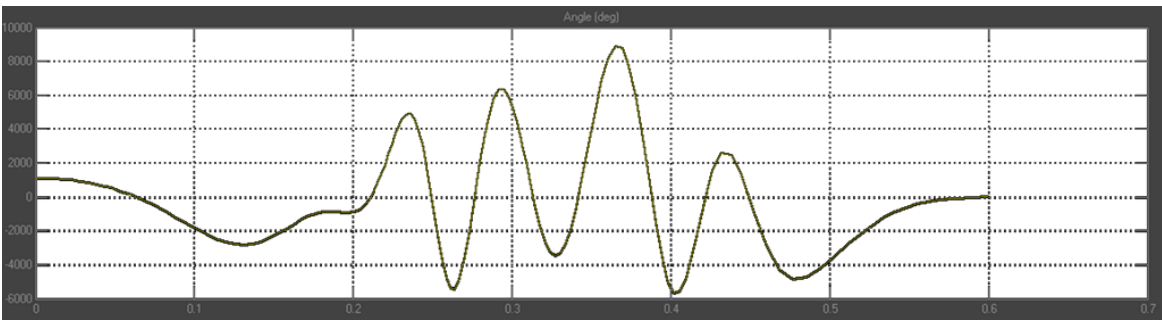


Figura 67. Aceleración angular  $\ddot{q}_3$  para la cadena articulada de 6 módulos

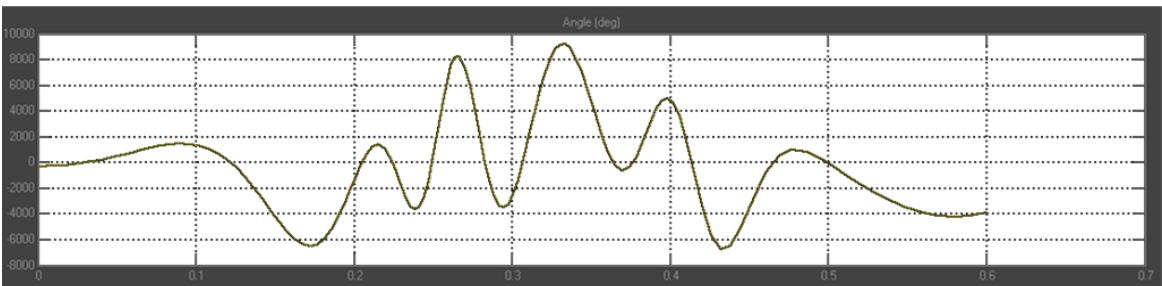
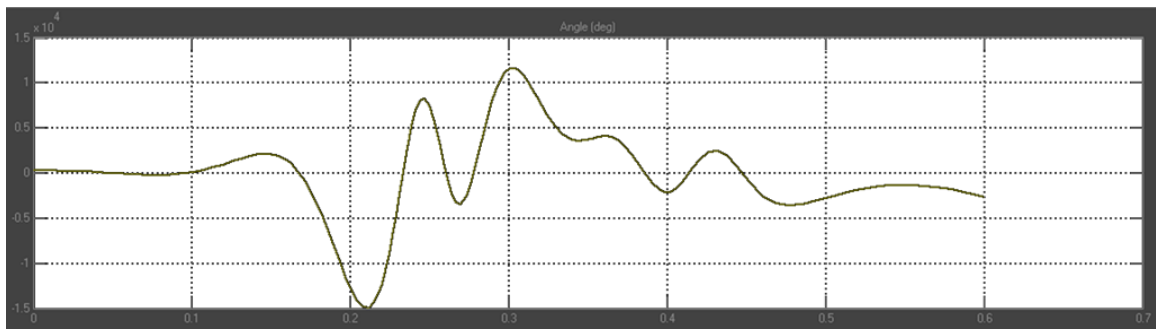
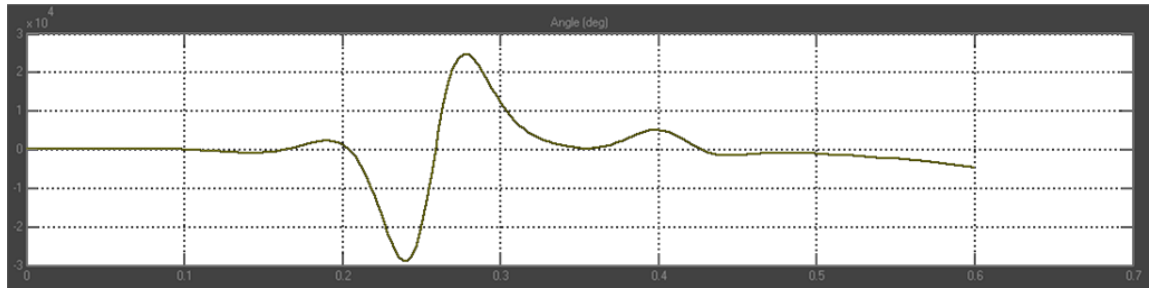


Figura 68. Aceleración angular  $\ddot{q}_4$  para la cadena articulada de 6 módulos



**Figura 69.** Aceleración angular  $\ddot{q}_5$  para la cadena articulada de 6 módulos



**Figura 70.** Aceleración angular  $\ddot{q}_6$  para la cadena articulada de 6 módulos