



**Instituto Tecnológico y de Estudios
Superiores de Monterrey**

TE3001B.101

Fundamentación de robótica (Gpo 101)

Semestre: Febrero - Junio 2022

Actividad 4 (Modelo de Energía y Lagrangiano)

Alumnos:

Frida Lizett Zavala Pérez	A01275226
José Jezarel Sánchez Mijares	A01735226
Antonio Silva Martínez	A01173663

Marzo del 2023

Actividad 4 (Modelo de Energía y Lagrangiano)

La mecánica de Lagrange, emplea energías cinéticas y potenciales que son cantidades escalares. La aplicación de la mecánica de Lagrange da lugar a n ecuaciones diferenciales correspondientes a n coordenadas generalizadas.

Suponiendo que la energía potencial está representada por T y la energía cinética por V , entonces el lagrangiano es igual a $L = T - V$.

La ecuación de Lagrange correspondiente a la coordenada generalizada q_i es:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad i = 1, 2, 3, \dots$$

A continuación se muestra el código en Matlab del **modelo de la Energía total** y el **lagrangiano** para cada una de las siguientes configuraciones de robots manipuladores.

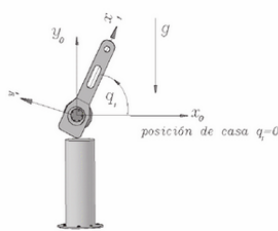
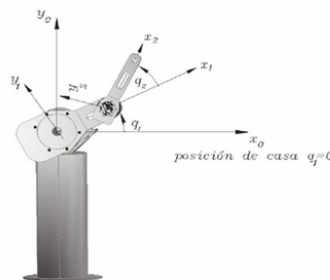
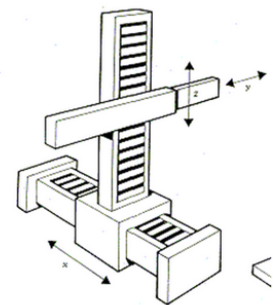


Figura 4.10 Péndulo robot.

Robot Péndulo (1gdl)



Robot Rotacional (2gdl)



Robot Cartesiano (3gdl)

Robot Péndulo

Primero se realiza la inicialización de variables, la inicialización de los vectores y la asignación de cada variable en los vectores que se usarán.

```
%Limpieza de pantalla
clear all
close all
clc

tic
%Declaración de variables simbólicas
syms th1(t) t %Ángulos de cada articulación
```

```

syms th1p(t) %Velocidades de cada articulación
syms th1pp(t) %Aceleraciones de cada articulación

syms m1 Ixx1 Iyy1 Izz1 %Masas y matrices de Inercia
syms t1 %Tiempos
syms l1 lc1 %l=longitud de eslabones y lc=distancia al centro de
masa de cada eslabón
syms pi g

%Creamos el vector de coordenadas articulares
Q= [th1];
%disp('Coordenadas generalizadas');
%pretty (Q);

%Creamos el vector de velocidades articulares
Qp= [th1p];
%disp('Velocidades generalizadas');
%pretty (Qp);
%Creamos el vector de aceleraciones articulares
Qpp= [th1pp];
%disp('Aceleraciones generalizadas');
%pretty (Qpp);

%Configuración del robot, 0 para junta rotacional, 1 para junta
prismática
RP=[0];

%Número de grado de libertad del robot
GDL= size(RP,2);
GDL_str= num2str(GDL);

```

Se usan las matrices de rotación para posteriormente guardar los valores dentro de las matrices de transformación que se inicializan en esta parte.

```

%Creamos un vector de ceros
Vector_Zeros= zeros(1, 3);

%Inicializamos las matrices de transformación Homogénea locales
A(:, :, GDL)=simplify([R(:, :, GDL) P(:, :, GDL); Vector_Zeros 1]);
%Inicializamos las matrices de transformación Homogénea globales

```

```

T(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
%Inicializamos las posiciones vistas desde el marco de referencia
inercial
PO(:,:,GDL)= P(:,:,GDL);
%Inicializamos las matrices de rotación vistas desde el marco de
referencia inercial
RO(:,:,GDL)= R(:,:,GDL);

```

El ciclo for guardará los valores obtenidos de la matriz en una matriz local y una global.

```

for i = 1:GDL
    i_str= num2str(i);
    %disp(strcat('Matriz de Transformación local A', i_str));
    A(:,:,i)=simplify([R(:,:,i) P(:,:,i); Vector_Zeros 1]);
    %pretty (A(:,:,i));

    %Globales
    try
        T(:,:,i)= T(:,:,i-1)*A(:,:,i);
    catch
        T(:,:,i)= A(:,:,i);
    end
    % disp(strcat('Matriz de Transformación global T', i_str));
    T(:,:,i)= simplify(T(:,:,i));
    % pretty(T(:,:,i))

    RO(:,:,i)= T(1:3,1:3,i);
    PO(:,:,i)= T(1:3,4,i);
    %pretty(RO(:,:,i));
    %pretty(PO(:,:,i));
end

```

Se calculan los Jacobianos a través de la derivación respecto a los ángulos theta que se usan en la configuración, creando una matriz de jacobianos para calcular el jacobiano lineal.

```

%Calculamos el jacobiano lineal de forma diferencial
%disp('Jacobiano lineal obtenido de forma diferencial');
%Derivadas parciales de x respecto a th1 y th2
Jv11= functionalDerivative(PO(1,1,GDL), th1);

```

```

%Derivadas parciales de y respecto a th1 y th2
Jv21= funcionalDerivative(PO(2,1,GDL), th1);
%Derivadas parciales de z respecto a th1 y th2
Jv31= funcionalDerivative(PO(3,1,GDL), th1);

%Creamos la matriz del Jacobiano lineal
jv_d=simplify([Jv11;
               Jv21;
               Jv31]);
%pretty(jv_d);

%Calculamos el jacobiano lineal de forma analítica
Jv_a(:,GDL)=PO(:, :,GDL);
Jw_a(:,GDL)=PO(:, :,GDL);

```

A partir de los jacobianos obtenidos se obtienen las submatrices de los jacobianos lineales y angulares y con ello sacar las velocidades lineal y angular.

```

for k= 1:GDL
    if RP(k)==0
        %Para las juntas de revolución
        try
            Jv_a(:,k)= cross(RO(:,3,k-1), PO(:, :,GDL)-PO(:, :,k-1));
            Jw_a(:,k)= RO(:,3,k-1);
        catch
            Jv_a(:,k)= cross([0,0,1], PO(:, :,GDL));%Matriz de
            rotación de 0 con respecto a 0 es la Matriz Identidad, la posición
            previa también será 0
            Jw_a(:,k)=[0,0,1];%Si no hay matriz de rotación previa
            se obtiene la Matriz identidad
        end
    else
        %Para las juntas prismáticas
        try
            Jv_a(:,k)= RO(:,3,k-1);
        catch
            Jv_a(:,k)=[0,0,1];
        end
        Jw_a(:,k)=[0,0,0];
    end
end

```

```

end

%Obtenemos SubMatrices de Jacobianos
Jv_a= simplify (Jv_a);
Jw_a= simplify (Jw_a);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano angular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac= [Jv_a;
      Jw_a];
Jacobiano= simplify(Jac);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares
% disp('Velocidad lineal obtenida mediante el Jacobiano lineal');
V=simplify (Jv_a*Qp);
% pretty(V);
% disp('Velocidad angular obtenida mediante el Jacobiano
angular');
W=simplify (Jw_a*Qp);
%    pretty(W);

```

Enseguida se obtiene la Energía Cinética considerando la distancia de los eslabones a través de los vectores y la matriz de inercia de cada eslabón, y debido a que solo se cuenta con una articulación únicamente se cuenta con una matriz de inercia para el eslabón, posteriormente se extrae la velocidad lineal de cada eje y se obtienen las velocidades angulares de euler.

```

%Energía Cinética

%Distancia del origen del eslabón a su centro de masa
%Vectores de posición respecto al centro de masa
P01=subs(P(:, :, 1)/2, l1, lc1);

%Creamos matrices de inercia para cada eslabón

I1=[Ixx1 0 0;

```

```

0 Iyy1 0;
0 0 Izz1];

%Función de energía cinética

%Extraemos las velocidades lineales en cada eje
V=V(t);
Vx= V(1,1);
Vy= V(2,1);
Vz= V(3,1);

%Extraemos las velocidades angular en cada ángulo de Euler
W=W(t);
W_pitch= W(1,1);
W_roll= W(2,1);
W_yaw= W(3,1);

```

Se calculan las velocidades de cada eslabón y se construye la matriz jacobiana de cada articulación, para obtener los vectores de las velocidades angular y lineal, finalmente se hace una suma de ellas para obtener una velocidad neta.

```

%Calculamos las velocidades para cada eslabón

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac1= [Jv_a;
       Jw_a];
Jacobiano1= simplify(Jac1);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares
%disp('Velocidad lineal obtenida mediante el Jacobiano lineal del
Eslabón 1');
Qp=Qp(t);
V1=simplify (Jv_a*Qp(1));
%pretty(V1);
% disp('Velocidad angular obtenida mediante el Jacobiano angular
del Eslabón 1');
W1=simplify (Jw_a*Qp(1));
% pretty(W1);

```

```

%Eslabón 1
V1_Total= V1+cross(W1,P01);
K1= (1/2*m1*(V1_Total))'*(1/2*m1*(V1_Total)) + (1/2*W1)'*(I1*W1);
%disp('Energía Cinética en el Eslabón 1');
K1= simplify (K1);
%pretty (K1);

K_Total= simplify (K1);
%pretty(K_Total)

```

Por último para completar el modelo de energía se calcula la energía potencial por cada uno de los eslabones, para obtener el lagrangiano se hace la resta de la energía cinética total, menos la energía potencial total. Mientras que el modelo de energía es la suma de las energías.

```

%Calculamos la energía potencial para cada uno de los eslabones

%Obtenemos las alturas respecto a la gravedad
h1= P01(2); %Tomo la altura paralela al eje y

U1=m1*g*h1;

%Calculamos la energía potencial total
U_Total= U1

%Obtenemos el Lagrangiano
Lagrangiano= simplify (K_Total-U_Total);
pretty (Lagrangiano);

H= simplify (K_Total+U_Total);
pretty (H)

```


Resultados

```

U_Total =
(g*lc1*m1*sin(th1(t)))/2

Izz1 |th1p(t)|2 - g lc1 m1 sin(th1(t)) |th1p(t)|2 cos(th1(t) - th1(t)) |m1|2 (l1 |lc1|2 + 2 lc1 |l1|2) (2 l1 + lc1)
-----
2 2 2 16 l1 lc1

Izz1 |th1p(t)|2 - g lc1 m1 sin(th1(t)) |th1p(t)|2 cos(th1(t) - th1(t)) |m1|2 (l1 |lc1|2 + 2 lc1 |l1|2) (2 l1 + lc1)
-----
2 2 2 16 l1 lc1

Elapsed time is 2.487330 seconds.

```

El desarrollo de los códigos de las configuraciones de robots siguientes es muy similar a la anterior, con la diferencia de que se deben considerar más eslabones con diversas posiciones, para o cuál se llevan a cabo algunas transformaciones en las juntas. En el robot de dos grados de libertad se toma en cuenta el marco de referencia, mientras que en el de tres grados de libertad se incluye el desplazamiento de los ejes. También cambia la cantidad de vectores para calcular los centros de masa, ya que se incluyen más elementos, del mismo modo los ciclos for para calcular los jacobianos se incrementan.

Robot Rotacional

Se consideran las alturas de cada uno de los eslabones.

```

%Calculamos la energía potencial para cada uno de los eslabones

%Obtenemos las alturas respecto a la gravedad
h1= P01(2); %Tomo la altura paralela al eje z
h2= P12(2); %Tomo la altura paralela al eje y

U1=m1*g*h1;
U2=m2*g*h2;

%Calculamos la energía potencial total
U_Total= U1 + U2

%Obtenemos el Lagrangiano
Lagrangiano= simplify (K_Total-U_Total);
%pretty (Lagrangiano);

H= simplify (K_Total+U_Total);
pretty (H)

toc

```

Resultados

U_Total =

$$(g \cdot l_{c1} \cdot m_1 \cdot \sin(\theta_1(t)))/2 + (g \cdot l_{c2} \cdot m_2 \cdot \sin(\theta_2(t)))/2$$

$$I_{zz1} \#1 = \frac{g \cdot l_{c1} \cdot m_1 \cdot \sin(\theta_1(t))}{2} - \frac{g \cdot l_{c2} \cdot m_2 \cdot \sin(\theta_2(t))}{2} + \frac{\#1 \cos(\overline{\theta_1(t)} - \theta_1(t)) |m_1|^2 (l_1 |l_{c1}|^2 + 2 l_{c1} |l_1|^2) (2 l_1 + l_{c1})}{8 l_1 l_{c1}}$$

where

$$\#1 == |\dot{\theta}_1(t)|^2$$

$$I_{zz1} \#1 + \frac{g \cdot l_{c1} \cdot m_1 \cdot \sin(\theta_1(t))}{2} + \frac{g \cdot l_{c2} \cdot m_2 \cdot \sin(\theta_2(t))}{2} + \frac{\#1 \cos(\overline{\theta_1(t)} - \theta_1(t)) |m_1|^2 (l_1 |l_{c1}|^2 + 2 l_{c1} |l_1|^2) (2 l_1 + l_{c1})}{8 l_1 l_{c1}}$$

where

$$\#1 == |\dot{\theta}_1(t)|^2$$

U_Total =

$$(g \cdot l_{c1} \cdot m_1 \cdot \sin(\theta_1(t)))/2 + (g \cdot l_{c2} \cdot m_2 \cdot \sin(\theta_2(t)))/2$$

$$I_{zz1} \#1 + \frac{g \cdot l_{c1} \cdot m_1 \cdot \sin(\theta_1(t))}{2} + \frac{g \cdot l_{c2} \cdot m_2 \cdot \sin(\theta_2(t))}{2} + \frac{\#1 \cos(\overline{\theta_1(t)} - \theta_1(t)) |m_1|^2 (l_1 |l_{c1}|^2 + 2 l_{c1} |l_1|^2) (2 l_1 + l_{c1})}{8 l_1 l_{c1}}$$

where

$$\#1 == |\dot{\theta}_1(t)|^2$$

Elapsed time is 4.791623 seconds.

Robót Cartesiano

Se considera la altura del eslabón dos y la longitud de la junta tres.

```
%Calculamos la energía potencial para cada uno de los eslabones
%Obtenemos las alturas respecto a la gravedad
```

```
U1=m1*g*h1;
```

```
U2=m2*g*l2;
```

```
%Calculamos la energía potencial total
```

```
U_Total= U1 + U2
```

```
%Obtenemos el Lagrangiano
```

```
Lagrangiano= simplify (K_Total-U_Total);
```

```
%pretty (Lagrangiano);
```

```
H= simplify (K_Total+U_Total);
```

```
pretty (H)
```

```
toc
```

Resultados

```
|l1p(t)|2 |m1|2 |m2|2 (|l1p(t)|2 + |l2p(t)|2) |m3|2 (|l1p(t)|2 + |l2p(t)|2 + |l3p(t)|2)
----- + ----- + -----
4 4 4

U_Total(t) =
g*m1*h1(t) + g*m2*l2(t)
|l1p(t)|2 |m1|2 |m2|2 (|l1p(t)|2 + |l2p(t)|2) |m3|2 (|l1p(t)|2 + |l2p(t)|2 + |l3p(t)|2)
----- + ----- + ----- + g m1 h1(t) + g m2 l2(t)
4 4 4

Elapsed time is 86.496659 seconds.
```

Conclusiones

Comprender el modelo de energía de un robot es de suma importancia para predecir el comportamiento del mismo, ya que para poder completar este modelo es necesario comprender previamente el movimiento específico del robót y las implicaciones de dichos movimientos. Realizar este modelo paso a paso fue la clave para comprender con claridad cada uno de los pasos y la finalidad de cada uno de ellos. Del mismo modo el análisis de los resultados es lo que termina de consolidar el entendimiento de cada movimiento y sus implicaciones.