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

tic
%Declaración de variables simbólicas
syms th1(t) th2(t) l3(t) th4(t) th5(t) th6(t)  t   %Angulos de cada
articulación
syms th1p(t) th2p(t) l3p(t) th4p(t) th5p(t) th6p(t)   %Velocidades de cada
articulación
syms th1pp(t) th2pp(t) l3pp(t) th4pp(t) th5pp(t) th6pp(t) %Aceleraciones de
cada articulación
syms m1 m2 m3 m4 m5 m6  Ixx1 Iyy1 Izz1 Ixx2 Iyy2 Izz2 Ixx3 Iyy3 Izz3 Ixx4
Iyy4 Izz4 Ixx5 Iyy5 Izz5 Ixx6 Iyy6 Izz6 %Masas y matrices de Inercia
syms l1 l2 d3 l4 l5 l6 lc1 lc2 lc3 lc4 lc5 lc6 %l=longitud de eslabones y
lc=distancia al centro de masa de cada eslabón
syms pi g a cero

 %Creamos el vector de coordenadas articulares
  Q= [th1; th2; l3; th4; th5; th6];
 %disp('Coordenadas generalizadas');
 %pretty (Q);

 %Creamos el vector de velocidades articulares
  Qp= [th1p; th2p; l3p; th4p; th5p; th6p];
 %disp('Velocidades generalizadas');
 %pretty (Qp);
 %Creamos el vector de aceleraciones articulares
  Qpp= [th1pp; th2pp; l3pp; th4pp; th5pp; th6pp];
 %disp('Aceleraciones generalizadas');
 %pretty (Qpp);

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

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

 rotacion_z= [cos(th1)  -sin(th1)     0 ;
              sin(th1)   cos(th1)     0 ;
              0             0         1];

 rotacion_y= [cos(th1)    0      sin(th1) ;
                 0        1          0    ;
             -sin(th1)    0      cos(th1)];

 rotacion_x= [1           0          0   ;
              0        cos(th1)   -sin(th1);
```

```
              0        sin(th1)   cos(th1)];

  x_transf= [1  0  0;   %x +90
             0  0 -1;
             0  1  0];

            [1  0  0;   %x -90
             0  0  1;
             0 -1  0];

  y_transf= [0  0  1;   %y +90
             0  1  0;
            -1  0  0];

%Articulación 1
%Posición de la articulación 1 respecto a 0
P(:,:,1)= [0;0;l1];
%Matriz de rotación de la junta 1 respecto a 0....
R(:,:,1)= [-cos(th1) 0  -sin(th1);
           -sin(th1) 0   cos(th1);
           0        -1         0];

%Articulación 2
%Posición de la articulación 2 respecto a 1
P(:,:,2)= [0;0;l2];
%Matriz de rotación de la junta 1 respecto a 0
R(:,:,2)= [cos(th2) 0   sin(th2);
           sin(th2) 0  -cos(th2);
           0        1         0];

%Articulación 3
%Posición de la articulación 2 respecto a 1
P(:,:,3)= [0;0;l3+d3];
%Matriz de rotación de la junta 1 respecto a 0
R(:,:,3)=   [1 0 0;
             0 1 0;
             0 0 1];

%Posición de la articulación 4 a 3
P(:,:,4)= [0;0;l4];
%Matriz de rotación de la junta 4 a 3
R(:,:,4)= [-cos(th4) 0  -sin(th4);
           -sin(th4) 0   cos(th4);
           0        -1         0];

%Articulación 5
%Articulación 4 a Articulación 5
P(:,:,5)= [l5*sin(th5); -l5*cos(th5);0];
%Matriz de rotación de la junta 4 a 5
R(:,:,5)= [cos(th5) 0   sin(th5);
```

```
           sin(th5) 0  -cos(th5);
           0        1         0];
%Articulación 6
%Articulación 6 a Extremo Final
%Posición de la articulación 6 a Extremo Final
P(:,:,6)= [0; 0; l6];
%Matriz de rotación de la junta 5 a 6---Transformación= Rot z(th6)
R(:,:,6)= [cos(th6) -sin(th6)  0;
           sin(th6)  cos(th6)  0;
           0         0         1];
%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);


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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CALCULAMOS LAS VELOCIDADES PARA CADA ESLABÓN%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%  VELOCIDADES PARA ESLABÓN 6  %%%%%%%%%%
```

```
%Calculamos el jacobiano lineal de forma analítica
Jv_a6(:,GDL)=PO(:,:,GDL);
Jw_a6(:,GDL)=PO(:,:,GDL);

for k= 1:GDL
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a6(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL)-PO(:,:,k-1));
            Jw_a6(:,k)= RO(:,3,k-1);
        catch
            Jv_a6(:,k)= cross([0,0,1], PO(:,:,GDL));%Matriz de rotación de 0
con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a6(:,k)=[0,0,1];%Si no hay matriz de rotación previa se
obtiene la Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a6(:,k)= RO(:,3,k-1);
        catch
            Jv_a6(:,k)=[0,0,1];
        end
            Jw_a6(:,k)=[0,0,0];
     end
 end

%Obtenemos SubMatrices de Jacobianos
Jv_a6= simplify (Jv_a6);
Jw_a6= simplify (Jw_a6);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac6= [Jv_a6;
      Jw_a6];
Jacobiano6= simplify(Jac6);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares para el eslabón 6
 disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón
6');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 6
```

```
V6=simplify (Jv_a6*Qp);
```

```
pretty(V6)
```

```
/ th1p(t) (l6 (sin(th5(t)) #2 + cos(th5(t)) sin(th1(t)) sin(th2(t))) - l2 cos(th1(t)) + l4 sin(th1(t)) sin
|
| sin(th1(t)) th2p(t) #1 - th1p(t) (l2 sin(th1(t)) - l6 (sin(th5(t)) #3 - cos(th1(t)) cos(th5(t)) sin(th2(
|
\                                                                                     cos(th2(t)) l3p(t) + th2p(t)

where

   #1 == l4 cos(th2(t)) + cos(th2(t)) (d3 + l3(t)) + l6 (cos(th2(t)) cos(th5(t)) + cos(th4(t)) sin(th2(t))

   #2 == cos(th1(t)) sin(th4(t)) - cos(th2(t)) cos(th4(t)) sin(th1(t))

   #3 == sin(th1(t)) sin(th4(t)) + cos(th1(t)) cos(th2(t)) cos(th4(t))
```

```
 disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón
6');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 6
```

```
W6=simplify (Jw_a6*Qp);
pretty(W6)
```

```
/ th6p(t) (sin(th5(t)) (sin(th1(t)) sin(th4(t)) + cos(th1(t)) cos(th2(t)) cos(th4(t))) - cos(th1(t)) cos(t
|
| th5p(t) (cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))) - th6p(t) (sin(th5(t)) (cos(th1(
|
\                                                           th1p(t) + th6p(t) (cos(th2(t)) cos(th5(t)) + c
```

```
%%%%%%%%%%  VELOCIDADES PARA ESLABÓN 5  %%%%%%%%%%

%Calculamos el jacobiano lineal y angular de forma analítica
Jv_a5(:,GDL-1)=PO(:,:,GDL-1);
Jw_a5(:,GDL-1)=PO(:,:,GDL-1);

for k= 1:GDL-1
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a5(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-1)-PO(:,:,k-1));
            Jw_a5(:,k)= RO(:,3,k-1);
        catch
            Jv_a5(:,k)= cross([0,0,1], PO(:,:,GDL-1));%Matriz de rotación de
0 con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a5(:,k)=[0,0,1];%Si no hay matriz de rotación previa se
obtiene la Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a5(:,k)= RO(:,3,k-1);
```

```
        catch
            Jv_a5(:,k)=[0,0,1];
        end
            Jw_a5(:,k)=[0,0,0];
     end
 end

%Obtenemos SubMatrices de Jacobianos
Jv_a5= simplify (Jv_a5);
Jw_a5= simplify (Jw_a5);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac5= [Jv_a5;
      Jw_a5];
Jacobiano5= simplify(Jac5);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares para el eslabón 5
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 5');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 5
```

```
Qp=Qp(t);
V5=simplify (Jv_a5*Qp(1:5));
pretty(V5)
```

```
/ th5p(t) (l5 cos(th1(t)) sin(th2(t)) sin(th5(t)) + l5 cos(th5(t)) sin(th1(t)) sin(th4(t)) + l5 cos(th1(t)
|
| th5p(t) (l5 sin(th1(t)) sin(th2(t)) sin(th5(t)) - l5 cos(th1(t)) cos(th5(t)) sin(th4(t)) + l5 cos(th2(t)
|
\

where

   #1 == l4 cos(th2(t)) + cos(th2(t)) (d3 + l3(t)) + l5 cos(th2(t)) cos(th5(t)) + l5 cos(th4(t)) sin(th2(t
```

```
 disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 
5');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 5
```

```
W5=simplify (Jw_a5*Qp(1:5));
pretty(W5)
```

```
/ - th5p(t) (cos(th4(t)) sin(th1(t)) - cos(th1(t)) cos(th2(t)) sin(th4(t))) - sin(th1(t)) th2p(t) - cos(th
|
|  th5p(t) (cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))) + cos(th1(t)) th2p(t) - sin(th1
|
\                                  th1p(t) + cos(th2(t)) th4p(t) + sin(th2(t)) sin(th4(t)) th5p(t)
```

```
%%%%%%%%%%  VELOCIDADES PARA ESLABÓN 4  %%%%%%%%%%%%%%%%%%%%%%%%%%

%Calculamos el jacobiano lineal y angular de forma analítica
Jv_a4(:,GDL-2)=PO(:,:,GDL-2);
Jw_a4(:,GDL-2)=PO(:,:,GDL-2);

for k= 1:GDL-2
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a4(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-2)-PO(:,:,k-1));
            Jw_a4(:,k)= RO(:,3,k-1);
        catch
            Jv_a4(:,k)= cross([0,0,1], PO(:,:,GDL-2));%Matriz de rotación de
0 con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a4(:,k)=[0,0,1];%Si no hay matriz de rotación previa se
obtiene la Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a4(:,k)= RO(:,3,k-1);
        catch
            Jv_a4(:,k)=[0,0,1];
        end
            Jw_a4(:,k)=[0,0,0];
     end
 end

%Obtenemos SubMatrices de Jacobianos
Jv_a4= simplify (Jv_a4);
Jw_a4= simplify (Jw_a4);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac4= [Jv_a4;
      Jw_a4];
Jacobiano4= simplify(Jac4);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares para el eslabón 4
```

```
 disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 4');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 4
```

```
V4=simplify (Jv_a4*Qp(1:4));
pretty(V4)
```

```
/ th1p(t) (l4 sin(th1(t)) sin(th2(t)) - l2 cos(th1(t)) + sin(th1(t)) sin(th2(t)) (d3 + l3(t))) - cos(th1(t
|
| cos(th2(t)) sin(th1(t)) th2p(t) #1 - sin(th1(t)) sin(th2(t)) l3p(t) - th1p(t) (l2 sin(th1(t)) + l4 cos(t
|
\                                                             cos(th2(t)) l3p(t) + sin(th2(t)) th2p(t) #1

where

   #1 == d3 + l4 + l3(t)
```

```
 disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 4');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 4
```

```
W4=simplify (Jw_a4*Qp(1:4));
pretty(W4)
```

```
/ - sin(th1(t)) th2p(t) - cos(th1(t)) sin(th2(t)) th4p(t) \
|                                                          |
|  cos(th1(t)) th2p(t) - sin(th1(t)) sin(th2(t)) th4p(t)  |
|                                                          |
\              th1p(t) + cos(th2(t)) th4p(t)              /
```

```

%%%%%%%%%%  VELOCIDADES PARA ESLABÓN 3  %%%%%%%%%%

%Calculamos el jacobiano lineal y angular de forma analítica
Jv_a3(:,GDL-3)=PO(:,:,GDL-3);
Jw_a3(:,GDL-3)=PO(:,:,GDL-3);

for k= 1:GDL-3
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a3(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-3)-PO(:,:,k-1));
            Jw_a3(:,k)= RO(:,3,k-1);
        catch
            Jv_a3(:,k)= cross([0,0,1], PO(:,:,GDL-3));%Matriz de rotación de 0 con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a3(:,k)=[0,0,1];%Si no hay matriz de rotación previa se obtiene la Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a3(:,k)= RO(:,3,k-1);
```

```
        catch
            Jv_a3(:,k)=[0,0,1];
        end
            Jw_a3(:,k)=[0,0,0];
     end
 end

%Obtenemos SubMatrices de Jacobianos
Jv_a3= simplify (Jv_a3);
Jw_a3= simplify (Jw_a3);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac3= [Jv_a3;
      Jw_a3];
Jacobiano3= simplify(Jac3);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares para el eslabón 3
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 3');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 3
```

```
V3=simplify (Jv_a3*Qp(1:3));
pretty(V3)
```

```
/ cos(th1(t)) cos(th2(t)) th2p(t) (d3 + l3(t)) - cos(th1(t)) sin(th2(t)) l3p(t) - th1p(t) (l2 cos(th1(t))
|
| cos(th2(t)) sin(th1(t)) th2p(t) (d3 + l3(t)) - sin(th1(t)) sin(th2(t)) l3p(t) - th1p(t) (l2 sin(th1(t))
|
\                                              cos(th2(t)) l3p(t) + sin(th2(t)) th2p(t) (d3 + l3(t))
```

```
 disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón
3');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 3
```

```
W3=simplify (Jw_a3*Qp(1:3));
pretty(W3)
```

```
/ -sin(th1(t)) th2p(t) \
|                      |
|  cos(th1(t)) th2p(t) |
|                      |
\        th1p(t)       /
```

```



%%%%%%%%%%  VELOCIDADES PARA ESLABÓN 2  %%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Calculamos el jacobiano lineal y angular de forma analítica
Jv_a2(:,GDL-4)=PO(:,:,GDL-4);
Jw_a2(:,GDL-4)=PO(:,:,GDL-4);

for k= 1:GDL-4
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a2(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-4)-PO(:,:,k-1));
            Jw_a2(:,k)= RO(:,3,k-1);
        catch
            Jv_a2(:,k)= cross([0,0,1], PO(:,:,GDL-4));%Matriz de rotación de
0 con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a2(:,k)=[0,0,1];%Si no hay matriz de rotación previa se
obtiene la Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a2(:,k)= RO(:,3,k-1);
        catch
            Jv_a2(:,k)=[0,0,1];
        end
            Jw_a2(:,k)=[0,0,0];
     end
 end

%Obtenemos SubMatrices de Jacobianos
Jv_a2= simplify (Jv_a2);
Jw_a2= simplify (Jw_a2);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac2= [Jv_a2;
      Jw_a2];
Jacobiano2= simplify(Jac2);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares para el eslabón 2
 disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón
2');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 2
```

```
V2=simplify (Jv_a2*Qp(1:2));
pretty(V2)
```

```
/ -l2 cos(th1(t)) th1p(t) \
|                         |
| -l2 sin(th1(t)) th1p(t) |
|                         |
\            0            /
```

```
 disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 
2');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 2
```

```
W2=simplify (Jw_a2*Qp(1:2));
pretty(W2)
```

```
/ -sin(th1(t)) th2p(t) \
|                      |
|  cos(th1(t)) th2p(t) |
|                      |
\        th1p(t)       /
```

```


%%%%%%%%%%  VELOCIDADES PARA ESLABÓN 1  %%%%%%%%%%%%%%%%%%%%%%%%%%

%Calculamos el jacobiano lineal y angular de forma analítica
Jv_a1(:,GDL-5)=PO(:,:,GDL-5);
Jw_a1(:,GDL-5)=PO(:,:,GDL-5);

for k= 1:GDL-5
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a1(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-5)-PO(:,:,k-1));
            Jw_a1(:,k)= RO(:,3,k-1);
        catch
            Jv_a1(:,k)= cross([0,0,1], PO(:,:,GDL-5));%Matriz de rotación de 
0 con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a1(:,k)=[0,0,1];%Si no hay matriz de rotación previa se 
obtiene la Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a1(:,k)= RO(:,3,k-1);
        catch
            Jv_a1(:,k)=[0,0,1];
        end
            Jw_a1(:,k)=[0,0,0];
     end
 end

%Obtenemos SubMatrices de Jacobianos
Jv_a1= simplify (Jv_a1);
Jw_a1= simplify (Jw_a1);
```

```
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac1= [Jv_a1;
      Jw_a1];
Jacobiano1= simplify(Jac1);
% pretty(Jacobiano);

%Obtenemos vectores de Velocidades Lineales y Angulares para el eslabón 1
 disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 1');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 1
```

```
V1=simplify (Jv_a1*Qp(1:1));
pretty(V1)
```

```
/ 0 \
|   |
| 0 |
|   |
\ 0 /
```

```
 disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 1');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 1
```

```
W1=simplify (Jw_a1*Qp(1:1));
pretty(W1)
```

```
/    0    \
|         |
|    0    |
|         |
\ th1p(t) /
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Energía Cinética
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Omitimos la división de cada lc%%%%%%%%%%%%%%%%
%Distancia del origen del eslabón a su centro de masa
%Vectores de posición respecto al centro de masa
 P01=subs(P(:,:,1), l1, lc1);%La función subs sustituye l1 por lc1 en
 P12=subs(P(:,:,2), l2, lc2); %la expresión P(:,:,1)/2
 P23=subs(P(:,:,3), d3, lc3);
 P34=subs(P(:,:,4), l4, lc4);%La función subs sustituye l1 por lc1 en
 P45=subs(P(:,:,5), l5, lc5); %la expresión P(:,:,1)/2
 P56=subs(P(:,:,6), l6, lc6);
```

```
%Creamos matrices de inercia para cada eslabón

I1=[Ixx1 0 0;
    0 Iyy1 0;
    0 0 Izz1];

I2=[Ixx2 0 0;
    0 Iyy2 0;
    0 0 Izz2];

I3=[Ixx3 0 0;
    0 Iyy3 0;
    0 0 Izz3];

I4=[Ixx4 0 0;
    0 Iyy4 0;
    0 0 Izz4];

I5=[Ixx5 0 0;
    0 Iyy5 0;
    0 0 Izz5];

I6=[Ixx6 0 0;
    0 Iyy6 0;
    0 0 Izz6];


%Función de energía cinética

%Calculamos la energía cinética para cada uno de los eslabones%%%%%%%%%

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

```
Energía Cinética en el Eslabón 1
```

```
K1= simplify (K1);
pretty (K1);
```

```
                2
Izz1 |th1p(t)|
---------------
      2
```

```



%Eslabón 2
V2_Total= V2+cross(W2,P12);
K2= (1/2*m2*(V2_Total))'*((V2_Total)) + (1/2*W2)'*(I2*W2);
```

```
disp('Energía Cinética en el Eslabón 2');
```

```
Energía Cinética en el Eslabón 2
```

```
K2= simplify (K2);
pretty (K2);
```

```
                                                                                   __                                  __
Izz2 #7   Iyy2 #6 cos(th1(t)) #4   Ixx2 #6 sin(th1(t)) #3   cos(th1(t)) #4 m2 #5 #1   sin(th1(t)) #3 m2 #5
------- + --------------------- + --------------------- - ---------------------- - --------------------
   2                2                        2                        #2                         #2

where

   #1 == l2 th1p(t) - lc2 th2p(t)

   #2 == 2 l2 lc2 th1p(t) th2p(t)

                ______
   #3 == sin(th1(t))

                ______
   #4 == cos(th1(t))

                       2                       2
   #5 == l2 #6 |lc2|  th1p(t) - lc2 #7 |l2|  th2p(t)

                     2
   #6 == |th2p(t)|

                     2
   #7 == |th1p(t)|
```

```

%Eslabón 3
V3_Total= V3+cross(W3,P23);
K3= (1/2*m3*(V3_Total))'*((V3_Total)) + (1/2*W3)'*(I3*W3);
disp('Energía Cinética en el Eslabón 3');
```

```
Energía Cinética en el Eslabón 3
```

```
K3= simplify (K3);
pretty (K3);
```

```
__
m3 (th1p(t) (l2 cos(th1(t)) - sin(th1(t)) sin(th2(t)) (d3 + l3(t))) - cos(th1(t)) th2p(t) (lc3 + l3(t)) +
-----------------------------------------------------------------------------------------------------------


where

                ______
   #1 == sin(th1(t))

                ______
   #2 == cos(th1(t))

            _____   __
   #3 == l3(t) + d3

                ______
```

```
  #4 == sin(th2(t))

         _______
  #5 == cos(th2(t))

       ______   ___
  #6 == l3(t) + lc3
```

```
%Eslabón 4
V4_Total= V4+cross(W4,P34);
K4= (1/2*m4*(V4_Total))'*((V4_Total)) + (1/2*W4)'*(I4*W4);
disp('Energía Cinética en el Eslabón 4');
```

```
Energía Cinética en el Eslabón 4
```

```
K4= simplify (K4);
pretty (K4);
```

```
__   ______       ________
m4 (l3p(t) #4 + th2p(t) #11 #3) (cos(th2(t)) l3p(t) + sin(th2(t)) th2p(t) #5)        / #10   #7 \
---------------------------------------------------------------------------- + Ixx4 | --- + -- | #2 + Iyy
                                      2                                             \  2     2 /

where

   #1 == cos(th1(t)) th2p(t) - sin(th1(t)) sin(th2(t)) th4p(t)

   #2 == sin(th1(t)) th2p(t) + cos(th1(t)) sin(th2(t)) th4p(t)

         ______   __   __
   #3 == l3(t) + d3 + l4

             _______
   #4 == cos(th2(t))

   #5 == d3 + l4 + l3(t)

         ______   __
   #6 == l3(t) + d3

         ________
   #7 == th4p(t) #12 #11

         ________
   #8 == th4p(t) #13 #11

         ________
   #9 == th2p(t) #12

          ________
   #10 == th2p(t) #13

              _______
   #11 == sin(th2(t))

              _______
   #12 == cos(th1(t))

              _______
   #13 == sin(th1(t))
```

```
%Eslabón 5
V5_Total= V5+cross(W5,P45);
K5= (1/2*m5*(V5_Total))'*((V5_Total)) + (1/2*W5)'*(I5*W5);
disp('Energía Cinética en el Eslabón 5');
```

```
Energía Cinética en el Eslabón 5
```

```
K5= simplify (K5);
pretty (K5);
```

```
     / _______               \       __
     | th1p(t)   #19   #18 |       m5 (cos(th2(t)) l3p(t) + th2p(t) (d3 sin(th2(t)) + l4 sin(th2(t)) + sin(
Izz5 | ------- + --- + --- | #1 + ------------------------------------------------------------------------
     \    2       2     2  /

where

   #1 == th1p(t) + cos(th2(t)) th4p(t) + sin(th2(t)) sin(th4(t)) th5p(t)

   #2 == l4 cos(th2(t)) + cos(th2(t)) (d3 + l3(t)) + l5 cos(th2(t)) cos(th5(t)) + l5 cos(th4(t)) sin(th2(t

   #3 == th5p(t) #13 + cos(th1(t)) th2p(t) - sin(th1(t)) sin(th2(t)) th4p(t)

   #4 == th5p(t) #14 + sin(th1(t)) th2p(t) + cos(th1(t)) sin(th2(t)) th4p(t)

                             __            __                __
   #5 == #25 #17 + #25 l4 + #25 #16 l5 + #26 #22 #15 l5

         _______
   #6 == th1p(t) + #19 + #18

         _______
   #7 == th5p(t) #20

         _______
   #8 == th5p(t) #21

         _______
   #9 == th4p(t) #24 #22

          _______
   #10 == th4p(t) #23 #22

          _______
   #11 == th2p(t) #24

          _______
   #12 == th2p(t) #23

   #13 == cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))

   #14 == cos(th4(t)) sin(th1(t)) - cos(th1(t)) cos(th2(t)) sin(th4(t))

              ______
   #15 == sin(th5(t))

              ______
   #16 == cos(th5(t))
```

```
          _____   __
   #17 == l3(t) + d3

          _______
   #18 == th5p(t) #22 #27

          _______
   #19 == th4p(t) #25

   #20 == #26 #23 - #24 #25 #27

   #21 == #24 #26 + #25 #23 #27

              ______
   #22 == sin(th2(t))

              ______
   #23 == sin(th1(t))

              ______
   #24 == cos(th1(t))

              ______
   #25 == cos(th2(t))

              ______
   #26 == cos(th4(t))

              ______
   #27 == sin(th4(t))
```

```
%Eslabón 6
V6_Total= V6+cross(W6,P56);
K6= (1/2*m6*(V6_Total))'*((V6_Total)) + (1/2*W6)'*(I6*W6);
disp('Energía Cinética en el Eslabón 6');
```

```
Energía Cinética en el Eslabón 6
```

```
K3= simplify (K6);
pretty (K6);
```

```
__   ___                                _______        __     __                        __                   __
m6 (lc6 (#7 - #5 + #12 + #9) - th1p(t) (#31 l2 - l6 #20 + #32 #27 #17 - #28 l5 #26 + #32 #27 l4 + #32 #29
----------------------------------------------------------------------------------------------------------


where

   #1 == th5p(t) #14 - th6p(t) #13 + cos(th1(t)) th2p(t) - sin(th1(t)) sin(th2(t)) th4p(t)

   #2 == th5p(t) #16 - th6p(t) #15 + sin(th1(t)) th2p(t) + cos(th1(t)) sin(th2(t)) th4p(t)

                     __             __             __                __
   #3 == #33 #17 + l6 #18 + #33 l4 + #33 #29 l5 + #34 #27 #28 l5

   #4 == l4 cos(th2(t)) + cos(th2(t)) (d3 + l3(t)) + l6 #19 + l5 cos(th2(t)) cos(th5(t)) + l5 cos(th4(t))

         _______
   #5 == th6p(t) #20
```

#6 == $\overline{\mathtt{th6p(t)}}$ #21

#7 == $\overline{\mathtt{th5p(t)}}$ #22

#8 == $\overline{\mathtt{th5p(t)}}$ #23

#9 == $\overline{\mathtt{th4p(t)}}$ #32 #27

#10 == $\overline{\mathtt{th4p(t)}}$ #31 #27

#11 == $\overline{\mathtt{th2p(t)}}$ #32

#12 == $\overline{\mathtt{th2p(t)}}$ #31

#13 == sin(th5(t)) #24 + cos(th5(t)) sin(th1(t)) sin(th2(t))

#14 == cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))

#15 == sin(th5(t)) #25 - cos(th1(t)) cos(th5(t)) sin(th2(t))

#16 == cos(th4(t)) sin(th1(t)) - cos(th1(t)) cos(th2(t)) sin(th4(t))

#17 == $\overline{\mathtt{l3(t)}}$ + $\overline{\mathtt{d3}}$

#18 == #33 #29 + #34 #27 #28

#19 == cos(th2(t)) cos(th5(t)) + cos(th4(t)) sin(th2(t)) sin(th5(t))

#20 == #28 #26 - #32 #29 #27

#21 == #28 #30 + #29 #31 #27

#22 == #34 #31 - #32 #33 #35

#23 == #32 #34 + #33 #31 #35

#24 == cos(th1(t)) sin(th4(t)) - cos(th2(t)) cos(th4(t)) sin(th1(t))

#25 == sin(th1(t)) sin(th4(t)) + cos(th1(t)) cos(th2(t)) cos(th4(t))

#26 == #31 #35 + #32 #33 #34

#27 == sin($\overline{\mathtt{th2(t)}}$)

#28 == sin($\overline{\mathtt{th5(t)}}$)

#29 == cos($\overline{\mathtt{th5(t)}}$)

#30 == #32 #35 - #33 #34 #31

#31 == sin($\overline{\mathtt{th1(t)}}$)

```
              ______
   #32 == cos(th1(t))

              ______
   #33 == cos(th2(t))

              ______
   #34 == cos(th4(t))

              ______
   #35 == sin(th4(t))
```

```
K_Total= simplify (K1+K2+K3+K4+K5+K6);
disp('Energía Cinética Total');
```

```
Energía Cinética Total
```

```
pretty (K_Total);
```

```
                                                              __          _______
Izz1 #68   Izz2 #68        /       #76   #75 \       m4 (#45 + th2p(t) #83 #15) (#11 + sin(th2(t)) th2p(t)
-------- + -------- + Izz5 | #10 + --- + --- | #28 + -----------------------------------------------------
    2          2           \        2     2  /                                   2

where

   #1 == sin(th1(t)) sin(th2(t)) (d3 + l3(t))

   #2 == cos(th1(t)) sin(th2(t)) (d3 + l3(t))

   #3 == sin(th1(t)) sin(th2(t)) l3p(t)

   #4 == cos(th1(t)) sin(th2(t)) l3p(t)

   #5 == l4 sin(th1(t)) sin(th2(t))

   #6 == cos(th2(t)) sin(th5(t)) - cos(th4(t)) cos(th5(t)) sin(th2(t))

   #7 == l4 cos(th1(t)) sin(th2(t))

   #8 == l5 cos(th2(t)) cos(th4(t)) sin(th5(t))

   #9 == l5 cos(th1(t)) cos(th5(t)) sin(th2(t))

          _______
          th1p(t)
   #10 == -------
             2

   #11 == cos(th2(t)) l3p(t)

   #12 == l2 sin(th1(t))

   #13 == l2 cos(th1(t))

   #14 == d3 + l4 + l3(t)

          _____   __   __
   #15 == l3(t) + d3 + l4

   #16 == l5 cos(th5(t)) sin(th1(t)) sin(th2(t))
```

#17 == sin(th2(t)) l3(t)

#18 == 2 l2 lc2 th1p(t) th2p(t)

#19 == l5 cos(th5(t)) sin(th2(t))

#20 == l2 th1p(t) - lc2 th2p(t)

#21 == l4 sin(th2(t))

#22 == d3 sin(th2(t))

#23 == th5p(t) #58 - th6p(t) #51 + #60 - #59

#24 == th5p(t) #61 - th6p(t) #52 + #63 + #62

#25 == #57 + #56 + l6 #53 + #55 + #54

#26 == th5p(t) #58 + #60 - #59

#27 == th5p(t) #61 + #63 + #62

#28 == th1p(t) + #64 + #65

#29 == #86 #71 + $\overline{l6}$ #66 + #86 $\overline{l4}$ + #86 #81 $\overline{l5}$ + #72

#30 == l5 sin(th5(t)) #77

#31 == l5 sin(th5(t)) #78

#32 == l2 #67 $|lc2|^2$ th1p(t) - lc2 #68 $|l2|^2$ th2p(t)

#33 == $\overline{th5p(t)}$ #69

#34 == $\overline{th5p(t)}$ #70

#35 == #86 #71 + #86 $\overline{l4}$ + #86 #81 $\overline{l5}$ + #72

#36 == $\overline{th6p(t)}$ #73

#37 == $\overline{th6p(t)}$ #74

#38 == $\overline{th4p(t)}$ #85 #83

#39 == $\overline{th4p(t)}$ #84 #83

#40 == $\overline{th2p(t)}$ #85

#41 == $\overline{th2p(t)}$ #84

#42 == $\overline{th1p(t)}$ + #76 + #75

$\#43 == \overline{l3p(t)}\ \#85\ \#83$

$\#44 == \overline{l3p(t)}\ \#84\ \#83$

$\#45 == \overline{l3p(t)}\ \#86$

$\#46 == \#86\ \#87\ \#80\ \overline{l5}$

$\#47 == \#85\ \#81\ \#83\ \overline{l5}$

$\#48 == \#81\ \#84\ \#83\ \overline{l5}$

#49 == #86 #80 - #87 #81 #83

$\#50 == \overline{l3(t)}\ \#83$

#51 == sin(th5(t)) #77 + cos(th5(t)) sin(th1(t)) sin(th2(t))

#52 == sin(th5(t)) #78 - cos(th1(t)) cos(th5(t)) sin(th2(t))

#53 == cos(th2(t)) cos(th5(t)) + cos(th4(t)) sin(th2(t)) sin(th5(t))

#54 == l5 cos(th4(t)) sin(th2(t)) sin(th5(t))

#55 == l5 cos(th2(t)) cos(th5(t))

#56 == cos(th2(t)) (d3 + l3(t))

#57 == l4 cos(th2(t))

#58 == cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))

#59 == sin(th1(t)) sin(th2(t)) th4p(t)

#60 == cos(th1(t)) th2p(t)

#61 == cos(th4(t)) sin(th1(t)) - cos(th1(t)) cos(th2(t)) sin(th4(t))

#62 == cos(th1(t)) sin(th2(t)) th4p(t)

#63 == sin(th1(t)) th2p(t)

#64 == cos(th2(t)) th4p(t)

#65 == sin(th2(t)) sin(th4(t)) th5p(t)

#66 == #86 #81 + #87 #83 #80

$\#67 == |th2p(t)|^2$

$\#68 == |th1p(t)|^2$

#69 == #87 #84 - #85 #86 #88

```
#70 == #85 #87 + #86 #84 #88

#71 == l3(t) + d3

#72 == #87 #83 #80 l5

#73 == #80 #79 - #85 #81 #83

#74 == #80 #82 + #81 #84 #83

#75 == th5p(t) #83 #88

#76 == th4p(t) #86

#77 == cos(th1(t)) sin(th4(t)) - cos(th2(t)) cos(th4(t)) sin(th1(t))

#78 == sin(th1(t)) sin(th4(t)) + cos(th1(t)) cos(th2(t)) cos(th4(t))

#79 == #84 #88 + #85 #86 #87

#80 == sin(th5(t))

#81 == cos(th5(t))

#82 == #85 #88 - #86 #87 #84

#83 == sin(th2(t))

#84 == sin(th1(t))

#85 == cos(th1(t))

#86 == cos(th2(t))

#87 == cos(th4(t))

#88 == sin(th4(t))
```

```
%Energia Potencial  p=mgh

%Obtenemos las alturas respecto a la gravedad
 h1= P01(3); %Tomo la altura paralela al eje Z
 h2= P12(2); %Tomo la altura paralela al eje y
 h3= P23(3); %Tomo la altura paralela al eje Z
 h4= P34(3); %Tomo la altura paralela al eje Z
 h5= P45(2); %Tomo la altura paralela al eje y
 h6= P56(3); %Tomo la altura paralela al eje Z
```

```
%Obtenemos las alturas respecto a la gravedad
U1=m1*g*h1
```

U1 = $g\,\mathrm{lc}_1\,m_1$

```
U2=m2*g*h2
```

U2 = 0

```
U3=m3*g*h3
```

U3 = $g\,m_3\,(\mathrm{lc}_3 + l_3(t))$

```
U4=m4*g*h4
```

U4 = $g\,\mathrm{lc}_4\,m_4$

```
U5=m5*g*h5
```

U5 = $-g\,\mathrm{lc}_5\,m_5\cos(\mathrm{th}_5(t))$

```
U6=m6*g*h6
```

U6 = $g\,\mathrm{lc}_6\,m_6$

```
%Calculamos la energía potencial total
U_Total= U1 + U2 +U3 +U4 +U5 +U6;

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

```
                                                      __          ________
Izz1 #68    Izz2 #68         /       #76    #75 \       m4 (#45 + th2p(t) #83 #15) (#11 + sin(th2(t)) th2p(t)
-------- + -------- + Izz5 | #10 + --- + --- | #28 + -----------------------------------------------------
    2           2            \        2     2  /                                    2

where

   #1 == sin(th1(t)) sin(th2(t)) (d3 + l3(t))

   #2 == cos(th1(t)) sin(th2(t)) (d3 + l3(t))

   #3 == sin(th1(t)) sin(th2(t)) l3p(t)

   #4 == cos(th1(t)) sin(th2(t)) l3p(t)

   #5 == cos(th2(t)) sin(th5(t)) - cos(th4(t)) cos(th5(t)) sin(th2(t))

   #6 == l4 sin(th1(t)) sin(th2(t))

   #7 == l4 cos(th1(t)) sin(th2(t))

   #8 == l5 cos(th2(t)) cos(th4(t)) sin(th5(t))
```

```
#9 == l5 cos(th1(t)) cos(th5(t)) sin(th2(t))

       _______
       th1p(t)
#10 == -------
          2

#11 == cos(th2(t)) l3p(t)

#12 == l2 sin(th1(t))

#13 == l2 cos(th1(t))

#14 == d3 + l4 + l3(t)

       _____   __   __
#15 == l3(t) + d3 + l4

#16 == l5 cos(th5(t)) sin(th1(t)) sin(th2(t))

#17 == sin(th2(t)) l3(t)

#18 == 2 l2 lc2 th1p(t) th2p(t)

#19 == l5 cos(th5(t)) sin(th2(t))

#20 == l2 th1p(t) - lc2 th2p(t)

#21 == l4 sin(th2(t))

#22 == d3 sin(th2(t))

#23 == th5p(t) #58 - th6p(t) #51 + #60 - #59

#24 == th5p(t) #61 - th6p(t) #52 + #63 + #62

#25 == #57 + #56 + l6 #53 + #55 + #54

#26 == th5p(t) #58 + #60 - #59

#27 == th5p(t) #61 + #63 + #62

#28 == th1p(t) + #64 + #65

                 __                __             __
#29 == #86 #71 + l6 #66 + #86 l4 + #86 #81 l5 + #72

#30 == l5 sin(th5(t)) #77

#31 == l5 sin(th5(t)) #78

                   2                       2
#32 == l2 #67 |lc2|  th1p(t) - lc2 #68 |l2|  th2p(t)

       _______
#33 == th5p(t) #69

       _______
#34 == th5p(t) #70

                     __             __
#35 == #86 #71 + #86 l4 + #86 #81 l5 + #72

       _______
```

#36 == th6p(t) #73

#37 == $\overline{\mathtt{th6p(t)}}$ #74

#38 == $\overline{\mathtt{th4p(t)}}$ #85 #83

#39 == $\overline{\mathtt{th4p(t)}}$ #84 #83

#40 == $\overline{\mathtt{th2p(t)}}$ #85

#41 == $\overline{\mathtt{th2p(t)}}$ #84

#42 == $\overline{\mathtt{th1p(t)}}$ + #76 + #75

#43 == $\overline{\mathtt{l3p(t)}}$ #85 #83

#44 == $\overline{\mathtt{l3p(t)}}$ #84 #83

#45 == $\overline{\mathtt{l3p(t)}}$ #86

#46 == #86 #87 #80 $\overline{\mathtt{l5}}$

#47 == #85 #81 #83 $\overline{\mathtt{l5}}$

#48 == #81 #84 #83 $\overline{\mathtt{l5}}$

#49 == #86 #80 - #87 #81 #83

#50 == $\overline{\mathtt{l3(t)}}$ #83

#51 == sin(th5(t)) #77 + cos(th5(t)) sin(th1(t)) sin(th2(t))

#52 == sin(th5(t)) #78 - cos(th1(t)) cos(th5(t)) sin(th2(t))

#53 == cos(th2(t)) cos(th5(t)) + cos(th4(t)) sin(th2(t)) sin(th5(t))

#54 == l5 cos(th4(t)) sin(th2(t)) sin(th5(t))

#55 == l5 cos(th2(t)) cos(th5(t))

#56 == cos(th2(t)) (d3 + l3(t))

#57 == l4 cos(th2(t))

#58 == cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))

#59 == sin(th1(t)) sin(th2(t)) th4p(t)

#60 == cos(th1(t)) th2p(t)

#61 == cos(th4(t)) sin(th1(t)) - cos(th1(t)) cos(th2(t)) sin(th4(t))

#62 == cos(th1(t)) sin(th2(t)) th4p(t)

#63 == sin(th1(t)) th2p(t)

#64 == cos(th2(t)) th4p(t)

#65 == sin(th2(t)) sin(th4(t)) th5p(t)

#66 == #86 #81 + #87 #83 #80

#67 == $|th2p(t)|^{2}$

#68 == $|th1p(t)|^{2}$

#69 == #87 #84 - #85 #86 #88

#70 == #85 #87 + #86 #84 #88

#71 == $\overline{l3(t)} + \overline{d3}$

#72 == #87 #83 #80 $\overline{l5}$

#73 == #80 #79 - #85 #81 #83

#74 == #80 #82 + #81 #84 #83

#75 == $\overline{th5p(t)}$ #83 #88

#76 == $\overline{th4p(t)}$ #86

#77 == cos(th1(t)) sin(th4(t)) - cos(th2(t)) cos(th4(t)) sin(th1(t))

#78 == sin(th1(t)) sin(th4(t)) + cos(th1(t)) cos(th2(t)) cos(th4(t))

#79 == #84 #88 + #85 #86 #87

#80 == sin($\overline{th5(t)}$)

#81 == cos($\overline{th5(t)}$)

#82 == #85 #88 - #86 #87 #84

#83 == sin($\overline{th2(t)}$)

#84 == sin($\overline{th1(t)}$)

#85 == cos($\overline{th1(t)}$)

#86 == cos($\overline{th2(t)}$)

```
   #87 == cos(th4(t))

              ______
   #88 == sin(th4(t))
```

```matlab
%Modelo de Energía
 H= simplify (K_Total+U_Total);
  pretty (H);
```

```
                                                     __        _______
Izz1 #68   Izz2 #68        /       #76   #75 \       m4 (#45 + th2p(t) #83 #15) (#11 + sin(th2(t)) th2p(t)
-------- + -------- + Izz5 | #10 + --- + --- | #28 + -----------------------------------------------------
    2          2           \        2     2  /                            2

where

   #1 == sin(th1(t)) sin(th2(t)) (d3 + l3(t))

   #2 == cos(th1(t)) sin(th2(t)) (d3 + l3(t))

   #3 == sin(th1(t)) sin(th2(t)) l3p(t)

   #4 == cos(th1(t)) sin(th2(t)) l3p(t)

   #5 == cos(th2(t)) sin(th5(t)) - cos(th4(t)) cos(th5(t)) sin(th2(t))

   #6 == l4 sin(th1(t)) sin(th2(t))

   #7 == l4 cos(th1(t)) sin(th2(t))

   #8 == l5 cos(th2(t)) cos(th4(t)) sin(th5(t))

   #9 == l5 cos(th1(t)) cos(th5(t)) sin(th2(t))

          _______
          th1p(t)
   #10 == -------
             2

   #11 == cos(th2(t)) l3p(t)

   #12 == l2 sin(th1(t))

   #13 == l2 cos(th1(t))

   #14 == d3 + l4 + l3(t)

          _____   __   __
   #15 == l3(t) + d3 + l4

   #16 == l5 cos(th5(t)) sin(th1(t)) sin(th2(t))

   #17 == sin(th2(t)) l3(t)

   #18 == 2 l2 lc2 th1p(t) th2p(t)

   #19 == l5 cos(th5(t)) sin(th2(t))

   #20 == l2 th1p(t) - lc2 th2p(t)

   #21 == l4 sin(th2(t))
```

#22 == d3 sin(th2(t))

#23 == th5p(t) #58 - th6p(t) #51 + #60 - #59

#24 == th5p(t) #61 - th6p(t) #52 + #63 + #62

#25 == #57 + #56 + l6 #53 + #55 + #54

#26 == th5p(t) #58 + #60 - #59

#27 == th5p(t) #61 + #63 + #62

#28 == th1p(t) + #64 + #65

#29 == #86 #71 + $\overline{\mathrm{l6}}$ #66 + #86 $\overline{\mathrm{l4}}$ + #86 #81 $\overline{\mathrm{l5}}$ + #72

#30 == l5 sin(th5(t)) #77

#31 == l5 sin(th5(t)) #78

#32 == l2 #67 $|\mathrm{lc2}|^2$ th1p(t) - lc2 #68 $|\mathrm{l2}|^2$ th2p(t)

#33 == $\overline{\mathrm{th5p(t)}}$ #69

#34 == $\overline{\mathrm{th5p(t)}}$ #70

#35 == #86 #71 + #86 $\overline{\mathrm{l4}}$ + #86 #81 $\overline{\mathrm{l5}}$ + #72

#36 == $\overline{\mathrm{th6p(t)}}$ #73

#37 == $\overline{\mathrm{th6p(t)}}$ #74

#38 == $\overline{\mathrm{th4p(t)}}$ #85 #83

#39 == $\overline{\mathrm{th4p(t)}}$ #84 #83

#40 == $\overline{\mathrm{th2p(t)}}$ #85

#41 == $\overline{\mathrm{th2p(t)}}$ #84

#42 == $\overline{\mathrm{th1p(t)}}$ + #76 + #75

#43 == $\overline{\mathrm{l3p(t)}}$ #85 #83

#44 == $\overline{\mathrm{l3p(t)}}$ #84 #83

#45 == $\overline{\mathrm{l3p(t)}}$ #86

#46 == #86 #87 #80 l5

#47 == #85 #81 #83 $\overline{l5}$

#48 == #81 #84 #83 $\overline{l5}$

#49 == #86 #80 - #87 #81 #83

#50 == $\overline{l3(t)}$ #83

#51 == sin(th5(t)) #77 + cos(th5(t)) sin(th1(t)) sin(th2(t))

#52 == sin(th5(t)) #78 - cos(th1(t)) cos(th5(t)) sin(th2(t))

#53 == cos(th2(t)) cos(th5(t)) + cos(th4(t)) sin(th2(t)) sin(th5(t))

#54 == l5 cos(th4(t)) sin(th2(t)) sin(th5(t))

#55 == l5 cos(th2(t)) cos(th5(t))

#56 == cos(th2(t)) (d3 + l3(t))

#57 == l4 cos(th2(t))

#58 == cos(th1(t)) cos(th4(t)) + cos(th2(t)) sin(th1(t)) sin(th4(t))

#59 == sin(th1(t)) sin(th2(t)) th4p(t)

#60 == cos(th1(t)) th2p(t)

#61 == cos(th4(t)) sin(th1(t)) - cos(th1(t)) cos(th2(t)) sin(th4(t))

#62 == cos(th1(t)) sin(th2(t)) th4p(t)

#63 == sin(th1(t)) th2p(t)

#64 == cos(th2(t)) th4p(t)

#65 == sin(th2(t)) sin(th4(t)) th5p(t)

#66 == #86 #81 + #87 #83 #80

#67 == $|th2p(t)|^2$

#68 == $|th1p(t)|^2$

#69 == #87 #84 - #85 #86 #88

#70 == #85 #87 + #86 #84 #88

#71 == $\overline{l3(t)} + \overline{d3}$

#72 == #87 #83 #80 $\overline{l5}$

#73 == #80 #79 - #85 #81 #83

#74 == #80 #82 + #81 #84 #83

#75 == $\overline{\mathrm{th5p}(t)}$ #83 #88

#76 == $\overline{\mathrm{th4p}(t)}$ #86

#77 == cos(th1(t)) sin(th4(t)) - cos(th2(t)) cos(th4(t)) sin(th1(t))

#78 == sin(th1(t)) sin(th4(t)) + cos(th1(t)) cos(th2(t)) cos(th4(t))

#79 == #84 #88 + #85 #86 #87

#80 == sin($\overline{\mathrm{th5}(t)}$)

#81 == cos($\overline{\mathrm{th5}(t)}$)

#82 == #85 #88 - #86 #87 #84

#83 == sin($\overline{\mathrm{th2}(t)}$)

#84 == sin($\overline{\mathrm{th1}(t)}$)

#85 == cos($\overline{\mathrm{th1}(t)}$)

#86 == cos($\overline{\mathrm{th2}(t)}$)

#87 == cos($\overline{\mathrm{th4}(t)}$)

#88 == sin($\overline{\mathrm{th4}(t)}$)