Design, modeling and control of a Scara manipulator using PID controller

Marco Baltazar, Branco Palma, Yerack Paredes
Faculty of Engineering, School of Electronic Engineering
Universidad Peruana de Ciencias Aplicadas (UPC)
U201213835@upc.edu.pe
U201417983@upc.edu.pe
U201512654@upc.edu.pe

Abstract: this paper describes the design, modeling and control of a Scara manipulator using inverse dynamics following desired trajectories. To do this, the robot model is analyzed and the necessary equations are obtained to model the system and its corresponding control.

I. INTRODUCTION

The main objective of this paper is to design a position controller for a robot based on the analysis of its behavior in space. A Scara manipulator is a robot that has four degrees of freedom, that is, four points of movement. The first three are of the rotary type and the fourth is displacement type.

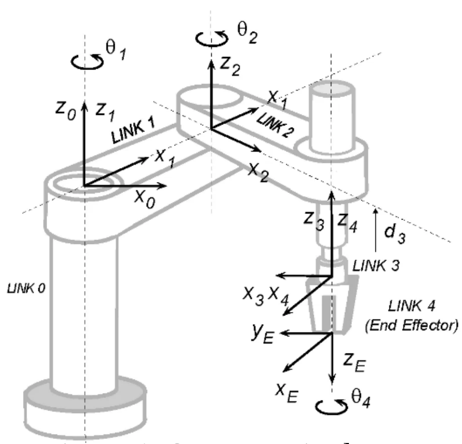

Figure 1: Scara manipulator

In order to achieve the design of a controller, it is necessary to have the basic concepts of geometry, transformation matrices, dynamic model and controller design. The controller will be in charge of governing the manipulator so that the desired position is achieved in a certain time from an input and the error measurement.

The modeling of the Scara manipulator in the coordinate axes is useful to understand not only how the behavior of the robot would be affected with respect to the input and output variables, but also, to understand the behavior of the system inside, analyze the internal variables and how these can provide more information about a real robot.

During the process, it will be necessary to consider possible non-linear behaviors of the system. This can cause unwanted behavior in the system and creating singularities which cannot be solved by the designed driver.

The behavior and design process can be represented through figure 2, which shows how the desired value goes through several stages until the robot is positioned where it is desired. In order to calculate each of the stages described in said figure, complex mathematical equations must be used, which must be detailed in detail in this paper.

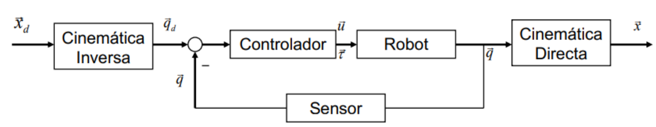

Figure 2: robot control system

The Denavit-Hartenberg (D-H) method is used to be able to determine the position of the end point of the robot with respect to the axis that is defined in the first joint of the manipulator robot. This method is also useful to establish the rotation and translation matrices of the system.

Starting from the D-H, the coordinate transformation matrix for each of the joints can be determined. This results from the multiplication of the corresponding matrices for each of the joints.

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

Figure 3: general matrix for each of the joints.

In order to design the robot controller, it is first necessary to know the equation of its dynamics, which will have the following general form:

$$M_{(q)}\ddot{q} + C_{(q,\dot{q})}\dot{q} + F_{(\dot{q})} + G_{(q)} + \tau_d = \tau$$

Where:

- M is the inertia matrix
- C is the matrix of centripetal force and Coriolis
- F is the friction vector
- G is the gravity vector
- τ_d represents the disturbances
- τ is the control input vector and contains the torques required at each joint

However, friction and disturbances can be neglected for mathematical analysis.

In order to control the Scara manipulator, it is necessary to use a PID controller, since it is probable that there is an error in steady state and a PD controller would not be sufficient in such a condition. An integrator is added in each of the joints. That is why touch control is represented by the equation:

$$\tau = M\left(\ddot{q}_d + k_v\dot{e} + k_p e + k_i\varepsilon\right) + N$$

Where:

- k_v is the derivative gain
- k_p is the proportional gain
- k_i is the integral gain
- e is the error
- $\dot{\varepsilon} = e$
- $N = \hat{C}\dot{q} + \hat{G}$

This controller adds stability to the PD type as long as the constant Ki is adequate.

II. Work Development and Analysis

In order to start the development process, it is first necessary to analyze the robot architecture from figure 4.

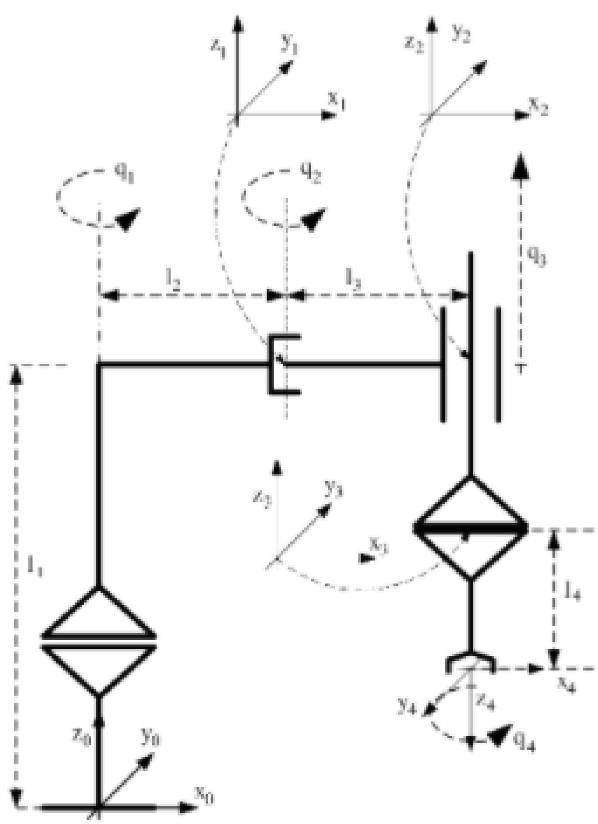

Figure 4: Scara robot architecture

Where four joint values are identified. In addition, you can also see the displacements to achieve that the point on the x4, y4 and z4 axis is located with reference to x0, y0 and z0.

Starting from a graphic analysis, it is possible to determine the D-H of the system.

	θ	d	a	α
1	q1	l1	l2	0
2	q2	0	l3	0
3	0	q3	0	0
4	q4	-l4	0	π

Figure 5: D-H Scara robot

This table explains the behavior in the joints and how the coordinates Xx, Yx, Zx are with respect to a fixed point or a mobile one as appropriate.

Starting from D-H it is possible to determine the Ai matrices for each of the joints using the general matrix.

$$A_1 = \begin{matrix} \cos(q1) & -sen(q1) & 0 & l2 \times \cos(q1) \\ sen(q1) & \cos(q1) & 0 & l2 \times sen(q1) \\ 0 & 0 & 1 & l1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$A_2 = \begin{matrix} \cos(q2) & -sen(q2) & 0 & l3 \times \cos(q2) \\ sen(q2) & \cos(q2) & 0 & l3 \times sen(q2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$A_3 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q3 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$A_4 = \begin{matrix} \cos(q4) & sen(q4) & 0 & 0 \\ sen(q4) & -\cos(q4) & 0 & 0 \\ 0 & 0 & -1 & -l4 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Later it is necessary to carry out the multiplication of the matrices in order to determine the transformation matrix.

$$T = A_1 \times A_2 \times A_3 \times A_4$$

$$T = \begin{matrix} C_{1+2+4} & S_{1+2+4} & 0 & l3 \times C_{1+2} + l2 \times C_1 \\ S_{1+2+4} & -C_{1+2+4} & 0 & l3 \times S_{1+2} + l2 \times S_1 \\ 0 & 0 & -1 & l1 - l4 + q3 \\ 0 & 0 & 0 & 1 \end{matrix}$$

We start from this point to determine the dynamics of the system, for which it is necessary to find the value of each of the variables within the following equation.

$$M_{(q)}\ddot{q} + C_{(q,\dot{q})}\dot{q} + G_{(q)} = \tau$$

From the matrix T the equations for Xi, Yi and Zi can be obtained. These equations will be derived and squared in order to determine the velocity.

$$dx_i = \frac{dx_i}{dq_1} \times dq_1 + \frac{dx_i}{dq_2} \times dq_2 + \frac{dx_i}{dq_3} \times dq_1$$

$$dy_i = \frac{dy_i}{dq_1} \times dq_1 + \frac{dy_i}{dq_2} \times dq_2 + \frac{dy_i}{dq_3} \times dq_1$$

$$dz_i = \frac{dz_i}{dq_1} \times dq_1 + \frac{dz_i}{dq_2} \times dq_2 + \frac{dz_i}{dq_3} \times dq_1$$

$$v^2 = (dx)^2 + (dy)^2 + (dz)^2$$

By finding the velocity it is possible to determine the kinetic energy with respect to each more in the robot.

$$K_i = \frac{1}{2} \times m_i \times v_i^2$$

$$K = \sum_{n=i}^{N} K_n$$

N = degrees of freedom

To calculate the inertia matrix, the equation of kinetic energy with respect to qi is derived twice.

$$M_{11} = \frac{\left(m2 \times (2 \times l2^2 + 4 \times C_2 \times l2 \times l3 + 2 \times l3^2)\right)}{2} + \frac{(m3 \times (2 \times l2^2 + 4 \times C_2 \times l2 \times l3 + 2 \times l3^2))}{2} + (m4 \times \frac{(2 \times l2^2 + 4 \times C_2 \times l2 \times l3 + 2 \times l3^2))}{2} + l2^2 \times m1$$

$$M_{21} = l3 \times (l3 + l2 \times C_2) \times (m2 + m3 + m4)$$

$$M_{31} = 0$$

$$M_{41} = 0$$

$$M_{12} = l3 \times (l3 + l2 \times C_2) \times (m2 + m3 + m4)$$

$$M_{22} = l3^2 \times (m2 + m3 + m4)$$

$$M_{32} = 0$$

$$M_{42} = 0$$

$$M_{13} = 0 \quad M_{23} = 0$$

$$M_{33} = \frac{m3}{4} + \frac{m4}{4} \quad M_{43} = 0$$

$$M_{14} = 0 \quad M_{24} = 0$$

$$M_{34} = 0 \quad M_{44} = 0$$

Similarly, using partial derivatives on each of the elements of the inertia matrix, it is possible to obtain the matrix of Centripetal Force and Coreolis.

$$C_{11} = -l2 \times l3 \times q2p \times S_2 \times (m2 + m3 + m4)$$

$$C_{21} = -l2 \times l3 \times q1p \times S_2 \times (m2 + m3 + m4)$$

$$C_{11} = 0 \quad C_{11} = 0$$

$$C_{12} = -l2 \times l3 \times S_2 \times (\dot{q}1 + \dot{q}2) \times (m2 + m3 + m4)$$

$$C_{13} = 0 \quad C_{14} = 0$$

$$\begin{matrix} C_{22} = 0 & C_{23} = 0 & C_{24} = 0 \\ C_{32} = 0 & C_{33} = 0 & C_{34} = 0 \\ C_{42} = 0 & C_{43} = 0 & C_{44} = 0 \end{matrix}$$

To determine the gravity vector, it is required to use the power energy formula, which depends only on the position in zi.

$$E_{Pi} = m \times g \times z_i$$

$$E_p = \sum_{n=i}^{N} E_{Pi}$$

$$G = \begin{matrix} 0 \\ 0 \\ \dfrac{g \times (m3 + m4)}{2} \\ 0 \end{matrix}$$

For the design of the controller it is necessary to take into account the following equation:

$$\tau = M\left(\ddot{q}_d + k_v\dot{e} + k_p e + k_i\varepsilon\right) + N$$

Where N has been calculated based on the equation:

$$N = C \times \begin{matrix} \dot{q}1 \\ \dot{q}2 \\ \dot{q}3 \end{matrix}$$

Finally, the error obtained is:

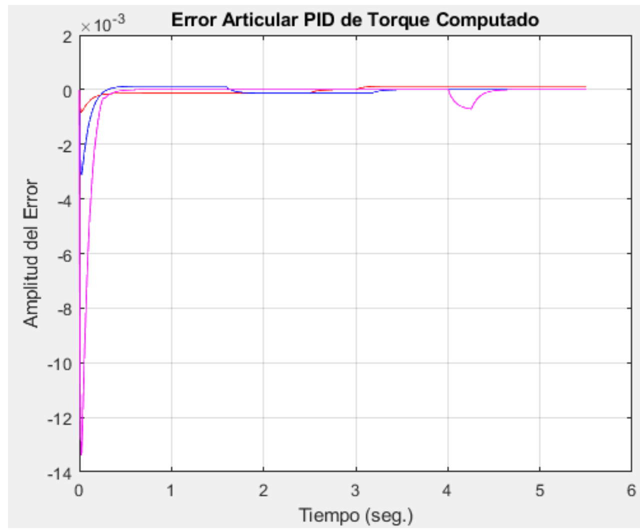

Figure 6: Joint torque error

In addition, the behavior of the position, velocity and corresponding acceleration for a given time can be determined.

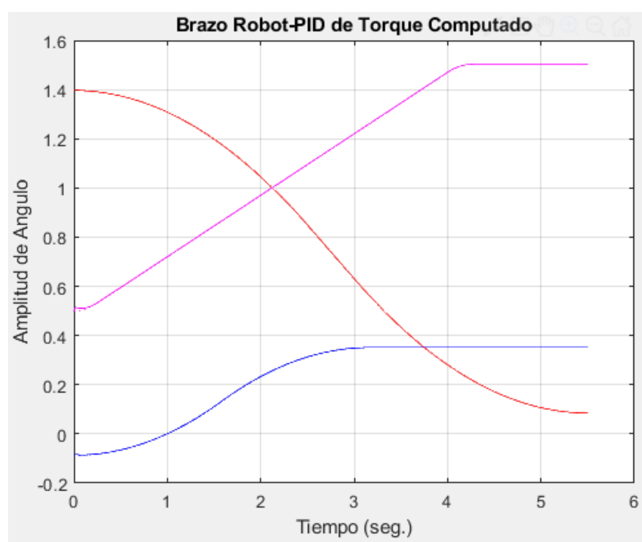

Figure 6: angle representation for each rotary joint

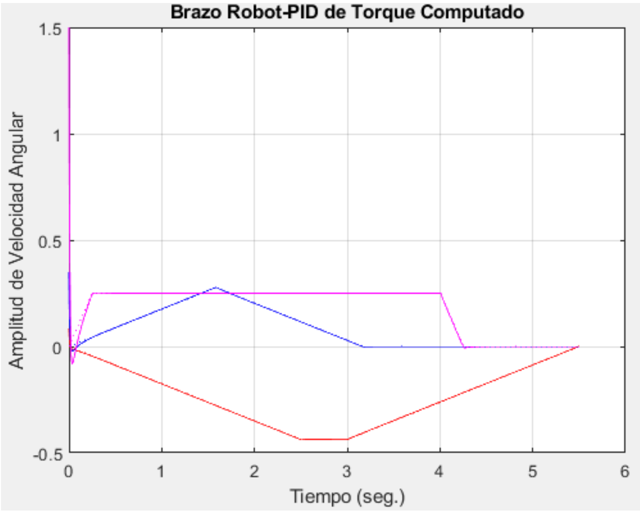

Figure 7: angular velocity for each rotary joint

III. Conclusions

By using the PID controller to manipulate the final position of the Scara robotic arm, it is possible to find its position with a low margin of error.

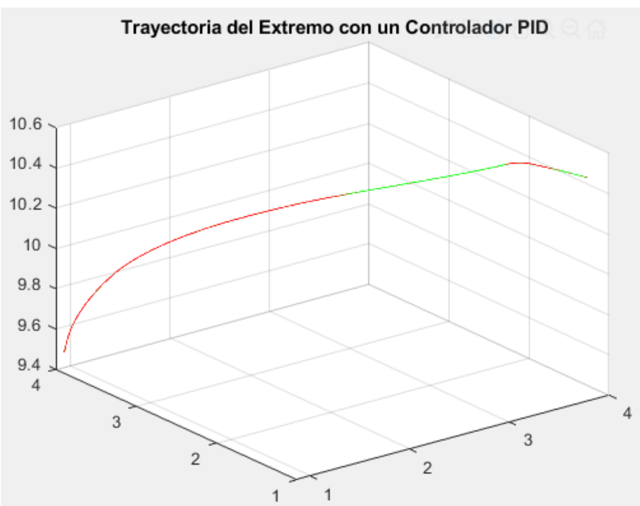

Figure 8: Position of the final point with PID controller

The result is more optimal when using an integrator, thus achieving a lower steady-state error.

IV. BIBLIOGRAPHY

[1] Enrique Arnáez Braschi, Enfoque práctico del control moderno, Primera edición. Lima: Metrocolor S.A., 2014.

[2] Antonio Barrientos, Fundamentos de Robótica 2da ed., 2017

[3] Enrique Arnaez Braschi; Enfoque práctico de la teoría de robots: con aplicaciones de Matlab. Lima: Editorial UPC, 2015.

V. ANEXO 1

```
% Cinemática Robot
clc;close all;clear all
syms q1 q2 q3 q4 l1 l2 l3 l4 l5
% a d al th
%DH = [ l2 l1 0 q1
 %      l3 0 0 q2
  %     0 q3 0 0
   %    0 -l4 pi q4];

   DH = [ q1 l1 l2 0
       q2 0 l3 0
       0 q3 0 0
       q4 -l4 0 pi];



A1 =
matra(DH(1,1),DH(1,2),DH(1,3),DH(1,4))
A2 =
matra(DH(2,1),DH(2,2),DH(2,3),DH(2,4))
A3 =
matra(DH(3,1),DH(3,2),DH(3,3),DH(3,4))
A4 =
matra(DH(4,1),DH(4,2),DH(4,3),DH(4,4))

T1 = A1
T2 = T1*A2
T3 = T2*A3
T4 = simplify(T3*A4)

% Calculando el Modelo Dinámico.
clear all; close all; clc
syms q1 q2 q3 q4 l1 l2 l3 l4 l5 q1p q2p
q3p q4p pi
% a d al th
DH = [ q1 l1 l2 0
       q2 0 l3 0
       0 q3 0 0
       q4 -l4 0 pi];
% masa 1
% a d al th
A1m =
matra(DH(1,1),DH(1,2)/2,DH(1,3),DH(1,4))
;
T1 = A1m;
x1 = T1(1,4)
y1 = T1(2,4)
z1 = T1(3,4)



% Derivamos con respecto al tiempo
x1p =
diff(x1,'q1')*q1p+diff(x1,'q2')*q2p+diff
(x1,'q3')*q3p +diff(x1,'q4')*q4p;
y1p =
diff(y1,'q1')*q1p+diff(y1,'q2')*q2p+diff
(y1,'q3')*q3p +diff(y1,'q4')*q4p;
z1p =
diff(z1,'q1')*q1p+diff(z1,'q2')*q2p+diff
(z1,'q3')*q3p +diff(z1,'q4')*q4p;
% Elevamos al cuadrado
x1p2 = x1p^2; y1p2 = y1p^2; z1p2 =
z1p^2;
% Vel. cuadrado v^2 = xp^2 + yp^2 + zp^2
v12 = simplify(x1p2 + y1p2 + z1p2)

% masa 2
A1 =
matra(DH(1,1),DH(1,2),DH(1,3),DH(1,4));
A2 =
matra(DH(2,1),DH(2,2),DH(2,3),DH(2,4));

T2 = A1*A2;
x2 = T2(1,4);
y2 = T2(2,4);
z2 = T2(3,4);
% Derivamos con respecto al tiempo
x2p =
diff(x2,'q1')*q1p+diff(x2,'q2')*q2p+diff
(x2,'q3')*q3p +diff(x2,'q4')*q4p;
y2p =
diff(y2,'q1')*q1p+diff(y2,'q2')*q2p+diff
(y2,'q3')*q3p +diff(y2,'q4')*q4p;
z2p =
diff(z2,'q1')*q1p+diff(z2,'q2')*q2p+diff
(z2,'q3')*q3p +diff(z2,'q4')*q4p;
% Elevamos al cuadrado
x2p2 = x2p^2; y2p2 = y2p^2; z2p2 =
z2p^2;
% Vel. cuadrado v^2 = xp^2 + yp^2 + zp^2
v22 = simplify(x2p2 + y2p2 + z2p2);



% masa 3
A1 =
matra(DH(1,1),DH(1,2),DH(1,3),DH(1,4));
A2 =
matra(DH(2,1),DH(2,2),DH(2,3),DH(2,4));
A3 =
matra(DH(3,1),DH(3,2)/2,DH(3,3),DH(3,4))
;
T3 = A1*A2*A3;
x3 = T3(1,4);
y3 = T3(2,4);
z3 = T3(3,4);
% Derivamos con respecto al tiempo
x3p =
diff(x3,'q1')*q1p+diff(x3,'q2')*q2p+diff
(x3,'q3')*q3p +diff(x3,'q4')*q4p;
y3p =
diff(y3,'q1')*q1p+diff(y3,'q2')*q2p+diff
(y3,'q3')*q3p +diff(y3,'q4')*q4p;
z3p =
diff(z3,'q1')*q1p+diff(z3,'q2')*q2p+diff
(z3,'q3')*q3p +diff(z3,'q4')*q4p;
% Elevamos al cuadrado
```

```
x3p2 = x3p^2; y3p2 = y3p^2; z3p2 =
z3p^2;
% Vel. cuadrado v^2 = xp^2 + yp^2 + zp^2
v32 = simplify(x3p2 + y3p2 + z3p2);
```

```
% masa 4
A1 =
matra(DH(1,1),DH(1,2),DH(1,3),DH(1,4));
A2 =
matra(DH(2,1),DH(2,2),DH(2,3),DH(2,4));
A3 =
matra(DH(3,1),DH(3,2)/2,DH(3,3),DH(3,4))
;
A4 =
matra(DH(4,1),DH(4,2)/2,DH(4,3),DH(4,4))
;
T4 = A1*A2*A3*A4;
x4 = T4(1,4);
y4 = T4(2,4);
z4 = T4(3,4);
% Derivamos con respecto al tiempo
x4p =
diff(x4,'q1')*q1p+diff(x4,'q2')*q2p+diff
(x4,'q3')*q3p +diff(x4,'q4')*q4p;
y4p =
diff(y4,'q1')*q1p+diff(y4,'q2')*q2p+diff
(y4,'q3')*q3p +diff(y4,'q4')*q4p;
z4p =
diff(z4,'q1')*q1p+diff(z4,'q2')*q2p+diff
(z4,'q3')*q3p +diff(z4,'q4')*q4p;
% Elevamos al cuadrado
x4p2 = x4p^2; y4p2 = y4p^2; z4p2 =
z4p^2;
% Vel. cuadrado v^2 = xp^2 + yp^2 + zp^2
v42 = simplify(x4p2 + y4p2 + z4p2);
```

```
syms m1 m2 m3 m4% Masas
% Energía Cinética del Sistema
% K = K1 + K2 ---> Ki = 1/2*mi*vi^2
K1 = 1/2*m1*v12; K2 = 1/2*m2*v22;
K3 = 1/2*m3*v32; K4 = 1/2*m4*v42;
K = K1+K2+K3+K4;
% Matriz de Inercias
m11 =
simplify(diff(diff(K,'q1p'),'q1p'));
m12 =
simplify(diff(diff(K,'q1p'),'q2p'));
m13 =
simplify(diff(diff(K,'q1p'),'q3p'));
m14 =
simplify(diff(diff(K,'q1p'),'q4p'));
m21 = m12;
m22 =
simplify(diff(diff(K,'q2p'),'q2p'));
m23 =
simplify(diff(diff(K,'q2p'),'q3p'));
m24 =
simplify(diff(diff(K,'q2p'),'q4p'));
m31 = m13;
m32 = m23;
m33 =
simplify(diff(diff(K,'q3p'),'q3p'));
m34 =
simplify(diff(diff(K,'q3p'),'q4p'));
m41 = m14;
m42 = m24;
m43 = m34;
m44 =
simplify(diff(diff(K,'q4p'),'q4p'));
```

```
M =[ m11 m12 m13 m14
m21 m22 m23 m24
m31 m32 m33 m34
m41 m42 m43 m44]
```

```
% Matriz de Fuerzas Centrípetas y de
Coriolis
% Empleamos los términos de Christoffel
% c11
c11 =
1/2*(diff(m11,'q1')+diff(m11,'q1')-
diff(m11,'q1'))*q1p;
c11 = c11 +
1/2*(diff(m11,'q2')+diff(m12,'q1')-
diff(m21,'q1'))*q2p;
c11 = c11 +
1/2*(diff(m11,'q3')+diff(m13,'q1')-
diff(m31,'q1'))*q3p;
c11 = c11 +
1/2*(diff(m11,'q4')+diff(m14,'q1')-
diff(m41,'q1'))*q4p;

c11 = simplify(c11);
% c12
c12 =
1/2*(diff(m12,'q1')+diff(m11,'q2')-
diff(m12,'q1'))*q1p;
c12 = c12 +
1/2*(diff(m12,'q2')+diff(m12,'q2')-
diff(m22,'q1'))*q2p;
c12 = c12 +
1/2*(diff(m12,'q3')+diff(m13,'q2')-
diff(m32,'q1'))*q3p;
c12 = c12 +
1/2*(diff(m12,'q4')+diff(m14,'q2')-
diff(m42,'q1'))*q4p;
c12 = simplify(c12);
```

```
% c13
c13 =
1/2*(diff(m13,'q1')+diff(m11,'q3')-
diff(m13,'q1'))*q1p;
c13 = c13 +
1/2*(diff(m13,'q2')+diff(m12,'q3')-
diff(m23,'q1'))*q2p;
c13 = c13 +
1/2*(diff(m13,'q3')+diff(m13,'q3')-
diff(m33,'q1'))*q3p;
c13 = c13 +
1/2*(diff(m13,'q4')+diff(m14,'q3')-
diff(m43,'q1'))*q4p;
c13 = simplify(c13);

% c14
c14 =
1/2*(diff(m14,'q1')+diff(m11,'q4')-
diff(m14,'q1'))*q1p;
c14 = c14 +
1/2*(diff(m14,'q2')+diff(m12,'q4')-
diff(m24,'q1'))*q2p;
c14 = c14 +
1/2*(diff(m14,'q3')+diff(m13,'q4')-
diff(m34,'q1'))*q3p;
c14 = c14 +
1/2*(diff(m14,'q4')+diff(m14,'q4')-
diff(m44,'q1'))*q4p;
c14 = simplify(c14);

% c21
c21 =
1/2*(diff(m21,'q1')+diff(m21,'q1')-
diff(m11,'q2'))*q1p;
c21 = c21 +
1/2*(diff(m21,'q2')+diff(m22,'q1')-
diff(m21,'q2'))*q2p;
c21 = c21 +
1/2*(diff(m21,'q3')+diff(m23,'q1')-
diff(m31,'q2'))*q3p;
c21 = c21 +
1/2*(diff(m21,'q4')+diff(m24,'q1')-
diff(m41,'q2'))*q4p;
c21 = simplify(c21);
% c22
c22 =
1/2*(diff(m22,'q1')+diff(m21,'q2')-
diff(m12,'q2'))*q1p;
c22 = c22 +
1/2*(diff(m22,'q2')+diff(m22,'q2')-
diff(m22,'q2'))*q2p;
c22 = c22 +
1/2*(diff(m22,'q3')+diff(m23,'q2')-
diff(m32,'q2'))*q3p;
c22 = c22 +
1/2*(diff(m22,'q4')+diff(m24,'q2')-
diff(m42,'q2'))*q4p;
c22 = simplify(c22);
% c23
c23 =
1/2*(diff(m23,'q1')+diff(m21,'q3')-
diff(m13,'q2'))*q1p;
c23 = c23 +
1/2*(diff(m23,'q2')+diff(m22,'q3')-
diff(m23,'q2'))*q2p;
c23 = c23 +
1/2*(diff(m23,'q3')+diff(m23,'q3')-
diff(m33,'q2'))*q3p;
c23 = c23 +
1/2*(diff(m23,'q4')+diff(m24,'q3')-
diff(m43,'q2'))*q4p;

c23 = simplify(c23);

% c24
c24 =
1/2*(diff(m24,'q1')+diff(m21,'q4')-
diff(m14,'q2'))*q1p;
c24 = c24 +
1/2*(diff(m24,'q2')+diff(m22,'q4')-
diff(m24,'q2'))*q2p;
c24 = c24 +
1/2*(diff(m24,'q3')+diff(m23,'q4')-
diff(m34,'q2'))*q3p;
c24 = c24 +
1/2*(diff(m24,'q4')+diff(m24,'q4')-
diff(m44,'q2'))*q4p;
c24 = simplify(c24);

% c31
c31 =
1/2*(diff(m31,'q1')+diff(m31,'q1')-
diff(m11,'q3'))*q1p;
c31 = c31 +
1/2*(diff(m31,'q2')+diff(m32,'q1')-
diff(m21,'q3'))*q2p;
c31 = c31 +
1/2*(diff(m31,'q3')+diff(m33,'q1')-
diff(m31,'q3'))*q3p;
c31 = c31 +
1/2*(diff(m31,'q4')+diff(m34,'q1')-
diff(m41,'q3'))*q4p;
c31 = simplify(c31);

% c32
c32 =
1/2*(diff(m32,'q1')+diff(m31,'q2')-
diff(m12,'q3'))*q1p;
c32 = c32 +
1/2*(diff(m32,'q2')+diff(m32,'q2')-
diff(m22,'q3'))*q2p;
c32 = c32 +
1/2*(diff(m32,'q3')+diff(m33,'q2')-
diff(m32,'q3'))*q3p;
```

```
c32 = c32 +
1/2*(diff(m32,'q4')+diff(m34,'q2')-
diff(m42,'q3'))*q4p;
c32 = simplify(c32);
% c33
c33 =
1/2*(diff(m33,'q1')+diff(m31,'q3')-
diff(m13,'q3'))*q1p;
c33 = c33 +
1/2*(diff(m33,'q2')+diff(m32,'q3')-
diff(m23,'q3'))*q2p;
c33 = c33 +
1/2*(diff(m33,'q3')+diff(m33,'q3')-
diff(m33,'q3'))*q3p;
c33 = c33 +
1/2*(diff(m33,'q4')+diff(m34,'q3')-
diff(m43,'q3'))*q4p;
c33 = simplify(c33);

% c34
c34 =
1/2*(diff(m34,'q1')+diff(m31,'q4')-
diff(m14,'q3'))*q1p;
c34 = c34 +
1/2*(diff(m34,'q2')+diff(m32,'q4')-
diff(m24,'q3'))*q2p;
c34 = c34 +
1/2*(diff(m34,'q3')+diff(m33,'q4')-
diff(m34,'q3'))*q3p;
c34 = c34 +
1/2*(diff(m34,'q3')+diff(m34,'q4')-
diff(m44,'q3'))*q4p;
c34 = simplify(c34);


% c41
c41 =
1/2*(diff(m41,'q1')+diff(m41,'q1')-
diff(m11,'q4'))*q1p;
c41 = c41 +
1/2*(diff(m41,'q2')+diff(m42,'q1')-
diff(m21,'q4'))*q2p;
c41 = c41 +
1/2*(diff(m41,'q3')+diff(m43,'q1')-
diff(m31,'q4'))*q3p;
c41 = c41 +
1/2*(diff(m41,'q4')+diff(m44,'q1')-
diff(m41,'q4'))*q4p;
c41 = simplify(c41);

% c42
c42 =
1/2*(diff(m42,'q1')+diff(m41,'q2')-
diff(m12,'q4'))*q1p;
c42 = c42 +
1/2*(diff(m42,'q2')+diff(m42,'q2')-
diff(m22,'q4'))*q2p;
c42 = c42 +
1/2*(diff(m42,'q3')+diff(m43,'q2')-
diff(m32,'q4'))*q3p;
c42 = c42 +
1/2*(diff(m42,'q4')+diff(m44,'q2')-
diff(m42,'q4'))*q4p;
c42 = simplify(c42);
% c43
c43 =
1/2*(diff(m43,'q1')+diff(m41,'q3')-
diff(m13,'q4'))*q1p;
c43 = c43 +
1/2*(diff(m43,'q2')+diff(m42,'q3')-
diff(m23,'q4'))*q2p;
c43 = c43 +
1/2*(diff(m43,'q3')+diff(m43,'q3')-
diff(m33,'q4'))*q3p;
c43 = c43 +
1/2*(diff(m43,'q4')+diff(m44,'q3')-
diff(m43,'q4'))*q4p;
c43 = simplify(c43);

% c44
c44 =
1/2*(diff(m44,'q1')+diff(m41,'q3')-
diff(m14,'q4'))*q1p;
c44 = c44 +
1/2*(diff(m44,'q2')+diff(m42,'q3')-
diff(m24,'q4'))*q2p;
c44 = c44 +
1/2*(diff(m44,'q3')+diff(m43,'q3')-
diff(m34,'q4'))*q3p;
c44 = c44 +
1/2*(diff(m44,'q4')+diff(m44,'q2')-
diff(m44,'q4'))*q4p;
c44 = simplify(c44);

C = [ c11 c12 c13 c14
c21 c22 c23 c24
c31 c32 c33 c34
c41 c42 c43 c44]


% Cálculo de la Energía Potencial
% P = P1 + P2 ---> Pi = mi*g*zi
syms g
P1 = m1*g*z1;
P2 = m2*g*z2;
P3 = m3*g*z3;
P4 = m4*g*z4;
```

```
P = P1 + P2 + P3 + P4;
% Determinación del Vector de Gravedad
g1 = simplify(diff(P,'q1'));
g2 = simplify(diff(P,'q2'));
g3 = simplify(diff(P,'q3'));
g4 = simplify(diff(P,'q4'));
G = conj([ g1 g2 g3 g4 ]')


% Robot Plotter con PDT
clear all; close all; clc
% Condiciones Iniciales
ff = pi/180; % Factor de Conv. Sex a Rad
amax(1) = 10*ff; % rad/s^2
vmax(1) = 25*ff; % rad/s
amax(2) = 10*ff; % rad/s^2
vmax(2) = 18*ff; % rad/s
amax(3) = 1.00; % mts/s^2
vmax(3) = 0.25; % mts/s
amax(4) = 10*ff; % rad/s^2
vmax(4) = 25*ff; % rad/s

amax = amax';
vmax = vmax';
% Constantes de la Trayectoria
p0 = [ 80*ff -5*ff 0.5]';
pf = [ 5*ff 20*ff 1.5]';

t0 = 0;
dt = 0.005; % Pequeño para simular
continuidad
x = [ p0' pf' ]';
k = 1;


[R1,V1,A1] =
trayrobot2(amax,vmax,p0,pf,dt);
QD = [R1];
QDP = [V1];
QDPP = [A1];
T = 0:(max(size(QD))-1);
T = T'*dt;
qd = QD(:,1:3)';
qdp = QDP(:,1:3)';
qdpp = QDPP(:,1:3)';


m1 = 20; m2 = 10; l1 =4; l2 = 2; l3 =2;
l4=-5; % masas y longitudes
m3 =20; m4=15;
g = 9.81; % Gravedad (m/s^2)
% Parámetros del Controlador
Kp = 1000*eye(3); Kv = 100*eye(3);
Ki=1*eye(3);
x = [ x(1:3); x(4:6); [0 0 0]'];
for t=0:dt:(max(T))
X1(k,1) = x(1); X2(k,1) = x(2); X3(k,1)
= x(3);
X4(k,1) = x(4); X5(k,1) = x(5); X6(k,1)
= x(6);
X7(k,1) = x(7); X8(k,1) = x(8); X9(k,1)
= x(9);
q1 = x(1); q2 = x(2); q3 = x(3);
q1p = x(4); q2p = x(5); q3p = x(6);

% Errores de Seguimiento
e = qd(:,k) - x(1:3); ep = qdp(:,k) -
x(4:6);
E(k,1:3) = e'; EP(k,1:3) = ep';


% Variable Intermedia para reducir las
dimesiones
% Cálculo de las Matrices del
Manipulador
% Matriz de Inercias

M= [(m2*(2*l2^2 + 4*cos(q2)*l2*l3 +
2*l3^2))/2 + (m3*(2*l2^2 +
4*cos(q2)*l2*l3 + 2*l3^2))/2 +
(m4*(2*l2^2 + 4*cos(q2)*l2*l3 +
2*l3^2))/2 + l2^2*m1, l3*(l3 +
l2*cos(q2))*(m2 + m3 + m4),          0

l3*(l3 + l2*cos(q2))*(m2 + m3 + m4),
l3^2*(m2 + m3 + m4),           0

0,                                    0,
m3/4 + m4/4 ];


% Inversa de M(q)
MI = inv(M);
% Matriz de Fuerzas Centrípetas y de
Coriolis


C = [-l2*l3*q2p*sin(q2)*(m2 + m3 + m4),
-l2*l3*sin(q2)*(q1p + q2p)*(m2 + m3 +
m4), 0
  l2*l3*q1p*sin(q2)*(m2 + m3 + m4),
0, 0
                                    0,
0, 0 ];


N = C*[ q1p q2p q3p]';
% Vector de Gravedad
```

```
G= [            0
                0
 (g*(m3 + m4))/2];



% Torque Computado (Señal de Control)
S = qdpp(:,k) + Kv*ep + Kp*e+ Ki*x(7:9);
tau = M*S+N+G;

% Ecuación de Estado
xp = [ x(4:6)
MI*(-N-G+tau)
e];
% Integración
x = x + xp*dt;
k = k + 1;
end



%Graficando
figure(1)
plot(T,E(:,1),'r',T,E(:,2),'b',T,E(:,3),
'm')
title('Error Articular PID de Torque
Computado')
xlabel('Tiempo (seg.)')
ylabel('Amplitud del Error')
grid on; zoom on

figure(2)
plot(T,X1,'r-',T,X2,'b-',T,X3,'m-
',T,QD(:,1),'r:',T,QD(:,2),'b:',T,QD(:,3
),'m:')
title('Brazo Robot-PID de Torque
Computado')
xlabel('Tiempo (seg.)');
ylabel('Amplitud de Angulo')
grid on; zoom on


figure(3)
plot(T,X4,'r-',T,X5,'b-',T,X6,'m-
',T,QDP(:,1),'r:',T,QDP(:,2),'b:',T,QDP(
:,3),'m:')
title('Brazo Robot-PID de Torque
Computado')
xlabel('Tiempo (seg.)')
ylabel('Amplitud de Velocidad Angular')
grid on; zoom on




% Trajectoria (PID) en espacio
cartesiano
x_cart =  l3*cos(X1 + X2) + l2*cos(X1);
y_cart =  l3*sin(X1 + X2) + l2*sin(X1);
z_cart = l1 - l4 + X3;
xd_cart = l3*cos(QD(:,1) +QD(:,2)) +
l2*cos(QD(:,1));
yd_cart = l3*sin(QD(:,1) + QD(:,2)) +
l2*sin(QD(:,1));
zd_cart = l1 - l4 + QD(:,3);
figure(4)
plot3(x_cart,y_cart,z_cart,'r-
',xd_cart,yd_cart,zd_cart,'g-');
title(' Trayectoria del Extremo con un
Controlador PID')
rotate3d on; grid on
```