es la distancia que se encuentra la articulaci´on de la base con respecto al origen del sistema fijo $\Sigma_0(x_0, y_0, z_0)$; l_2 y l_3 son las longitudes de los eslabones del hombro y codo, respectivamente.

Las matrices de transformaciones homog´eneas para la configuraci´on antropom´orfica adquieren la siguiente forma:

$$H_0^{'} = H_{Rz_0}(q_1)\, H_{Tz_0}(l_1 + \beta_1)\, H_{Tx_0}(0)\, H_{Rx_0}\left(\frac{\pi}{2}\right) \tag{5.19}$$

$$= \begin{bmatrix} \cos(q_1) & 0 & \text{sen}(q_1) & 0 \\ \text{sen}(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & l_1 + \beta_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.20}$$

$$H_1^2 = H_{Rz_1}(q_2)\, H_{Tz_1}(\beta_2)\, H_{Tx_1}(l_2)\, H_{Rx_1}(0) \tag{5.21}$$

$$= \begin{bmatrix} \cos(q_2) & -\text{sen}(q_2) & 0 & l_2\cos(q_2) \\ \text{sen}(q_2) & \cos(q_2) & 0 & l_2\,\text{sen}(q_2) \\ 0 & 0 & 1 & \beta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.22}$$

$$H_2^3 = H_{Rz_2}(q_3)\, H_{Tz_2}(\beta_2)\, H_{Tx_2}(l_3)\, H_{Rx_2}(0) \tag{5.23}$$

$$= \begin{bmatrix} \cos(q_3) & -\text{sen}(q_3) & 0 & l_3\cos(q_3) \\ \text{sen}(q_3) & \cos(q_3) & 0 & l_3\,\text{sen}(q_3) \\ 0 & 0 & 1 & \beta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.24}$$

Por lo tanto, la matriz homog´enea del robot antropom´orfico de 3 gdl movi´endose en su espacio tridimensional es:

$$H_0^3 = H_0^1 H_1^2 H_2^3 \tag{5.25}$$

$$= \begin{bmatrix} \cos(q_1)\cos(q_2+q_3) & -\cos(q_1)\,\text{sen}(q_2+q_3) & \text{sen}(q_1) & [\beta_2+\beta_3]\text{sen}(q_1) + \cos(q_1)[l_2\cos(q_2) + l_3\cos(q_2+q_3)] \\ \text{sen}(q_1)\cos(q_2+q_3) & -\text{sen}(q_1)\,\text{sen}(q_2+q_3) & -\cos(q_1) & -[\beta_2+\beta_3]\cos(q_1) + \text{sen}(q_1)[l_2\cos(q_2) + l_3\cos(q_2+q_3)] \\ \text{sen}(q_2+q_3) & \cos(q_2+q_3) & 0 & l_1 + \beta_1 + l_2\,\text{sen}(q_2) + l_3\,\text{sen}(q_2+q_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Las coordenadas cartesianas del extremo final del robot manipulador antropom´orfico de 3 gdl se encuentran relacionadas con las coordenadas articulares de la siguiente forma:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} [\beta_2+\beta_3]\,\text{sen}(q_1) + \cos(q_1)[l_2\cos(q_2) + l_3\cos(q_2+q_3)] \\ -[\beta_2+\beta_3]\cos(q_1) + \text{sen}(q_1)[l_2\cos(q_2) + l_3\cos(q_2+q_3)] \\ l_1 + \beta_1 + l_2\,\text{sen}(q_2) + l_3\,\text{sen}(q_2+q_3) \end{bmatrix} \tag{5.26}$$

Cinem´atica inversa

La cinem´atica inversa del robot antropom´orfico de 3 gdl consiste en obtener las posiciones articulares en funci´on de las coordenadas cartesianas de la ecuaci´on (5.26). Una forma para resolver dicho planteamiento es utilizar el m´etodo geom´etrico, con esta intenci´on considere la proyecci´on de los eslabones del hombro y del codo sobre el plano $x_0 - y_0$ (ver figura 5.8); dicha proyecci´on se ampl´ıa en la figura 5.9 para llevar a cabo el an´alisis geom´etrico correspondiente a la variable q_1.

Observe que existe proyecci´on geom´etrica sobre el plano $x_0 - y_0$ de los par´ametros β_2 y β_3 de los servomotores, debido a que los ejes z_1 y z_2 quedan paralelos a dicho plano, lo que no sucede para los casos del p´endulo y robot de 2 gdl.

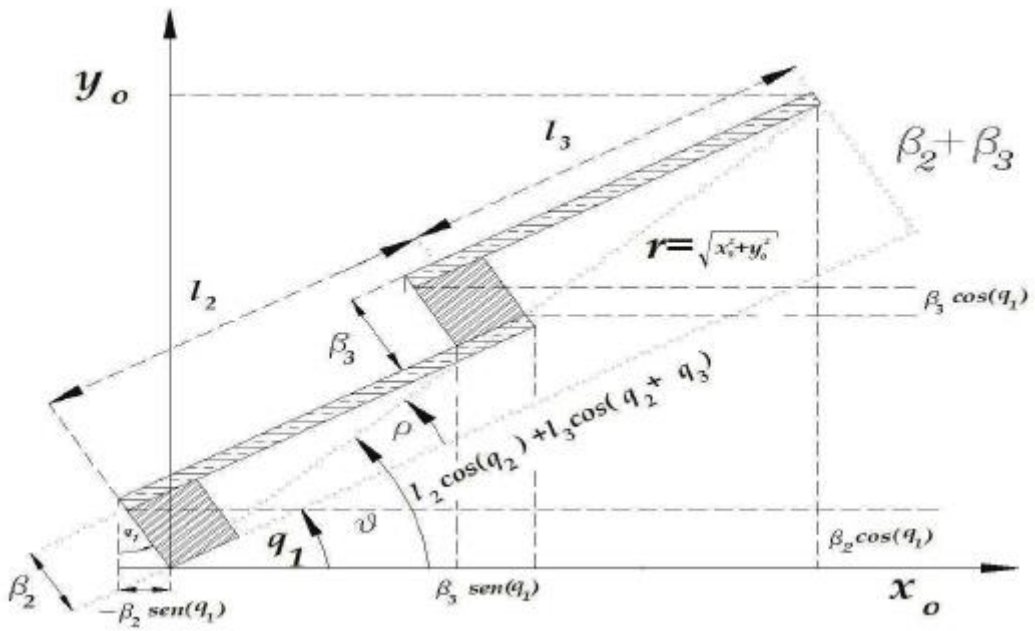


Figura 5.9 M´etodo geom´etrico de la cinem´atica inversa 3 gdl para obtener q_1.

De la figura 5.9 tenemos que $\vartheta = \rho + q_1 \Rightarrow q_1 = \vartheta - \rho$. Observe que el ángulo ϑ se obtiene en funci´on de las coordenadas cartesianas (x_0, y_0) satisface $\vartheta = \text{atan}\ \frac{y_0}{x}$. Adem´as, tomando en cuenta el tri´angulo con l´ınea punteada, el ángulo ρ queda en funci´on del cateto opuesto (formado por el ancho de los servomotores $\beta_2 + \beta_3$) y el cateto adyacente formado por la diferencia de la hipotenusa $(r = \sqrt{x_0^2 + y_0^2})$ y del

cateto opuesto obtiene la siguiente forma:

$$\rho = \text{atan}\left(\frac{\beta_2 + \beta_3}{\sqrt{x_0^2 + y_0^2 - (\beta_2 + \beta_3)^2}}\right)$$

Por lo tanto, la variable articular q_1 adquiere la siguiente estructura:

$$q_1 = \vartheta - \rho = \text{atan}\left(\frac{y_0}{x_0}\right) - \text{atan}\left(\frac{\beta_2 + \beta_3}{\sqrt{x_0^2 + y_0^2 - (\beta_2 + \beta_3)^2}}\right) \qquad (5.27)$$

Para obtener la variable articular q_3 consid´erese como referencia el ´angulo ϑ del tri´angulo formado por el cateto adyacente $l_2 + l_3\cos(q_3)$, cateto opuesto $l_3\,\text{sen}(q_3)$ y por la hipotenusa $\sqrt{x_0^2 + y_0^2 + z_0^2}$ (ver figura 5.10). Es importante recalcar que los ´angulos ϑ y q_2 se miden en direcci´on positiva con respecto al plano horizontal $x_0 - y_0$ hacia el eje z_0.

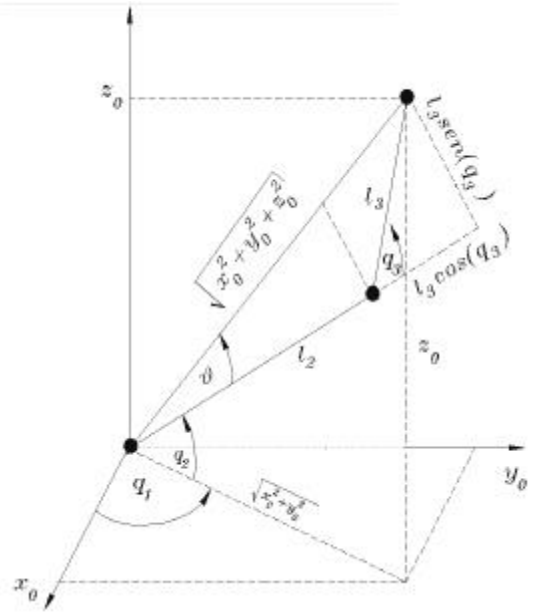


Figura 5.10 M´etodo geom´etrico para obtener q_2 y q_3 de la cinem´atica inversa 3 gdl.

Empleando el teorema de Pit´agoras se obtiene una posible soluci´on para q_3 en t´erminos de la funci´on arco-tangente:

$$x_0^2 + y_0^2 + z_0^2 = [l_2 + l_3\cos(q_3)]^2 + l_3^2\text{sen}^2(q_3)$$

$$= l_2^2 + l_3^2[\cos^2(q_3) + \operatorname{sen}^2(q_3)] + 2l_2l_3\cos(q_3)$$

$$= l_2^2 + l_3^2 + 2l_2l_3\cos(q_3)$$

$$\cos(q_3) = \frac{x_{20}^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}{2l_2l_3}$$

$$q_3 = \operatorname{atan}\left(\frac{\sqrt{1 - \left(\frac{x_{20}^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}{2l_2l_3}\right)^2}}{\frac{x_{20}^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}{2l_2l_3}}\right)$$

En referencia a la figura 5.10, obs´ervese que el tri´angulo formado por los lados $\sqrt{x_{20}^2 + y_0^2}$, z_0 y la hipotenusa $\sqrt{x_{20}^2 + y_0^2 + z_0^2}$, tomando en cuenta que la tangente de los ´angulos $\vartheta + q_2$ satisface: $\tan(\vartheta + q_2) = \frac{z_0}{\sqrt{x_{20}^2 + y_0^2}}$, además $\tan(\vartheta) = \frac{l_3\operatorname{sen}(q_3)}{l_2 + l_3\cos(q_3)}$, usando identidades trigonom´etricas de la funci´on tangente se obtiene:

$$\tan(\vartheta + q_2) = \frac{\tan(q_2) + \tan(\vartheta)}{1 - \tan(q_2)\tan(\vartheta)} = \frac{z_0}{\sqrt{x_{20}^2 + y_0^2}}$$

Realizando sencillos pasos algebraicos, para determinar la funci´on tangente de q_2:

$$\tan(q_2)\left(1 + \frac{z_0}{\sqrt{x_{20}^2 + y_0^2}}\tan(\vartheta)\right) = \frac{z_0}{\sqrt{x_{20}^2 + y_0^2}} - \tan(\vartheta)$$

$$\tan(q_2) = \frac{\frac{z_0}{\sqrt{x_{20}^2 + y_0^2}} - \tan(\vartheta)}{1 + \frac{z_0}{\sqrt{x_{20}^2 + y_0^2}}\tan(\vartheta)} = \frac{\frac{z_0}{\sqrt{x_{20}^2 + y_0^2}} - \frac{l_3\operatorname{sen}(q_3)}{l_2 + l_3\cos(q_3)}}{1 + \frac{z_0}{\sqrt{x_{20}^2 + y_0^2}}\frac{l_3\operatorname{sen}(q_3)}{l_2 + l_3\cos(q_3)}}$$

$$= \frac{[l_2 + l_3\cos(q_3)]z_0 - l_3\operatorname{sen}(q_3)\sqrt{x_{20}^2 + y_0^2}}{\sqrt{x_{20}^2 + y_0^2}[l_2 + l_3\cos(q_3)] + z_0l_3\operatorname{sen}(q_3)}$$

Por lo tanto, la cinem´atica inversa que relaciona las coordenadas articulares q_1, q_2, q_3 en funci´on de las coórdenadas cartesianas x_0, y_0, z_0 del extremo final de un robot

antropom´orfico de 3 gdl que se mueve en su espacio tridimensional, considerando el espesor de los servomotores tiene la siguiente forma:

$$q_1 = \text{atan}\frac{y_0}{x_0} - \text{atan}\left(\frac{\beta_2 + \beta_3}{\sqrt{x_0^2 + y_0^2 - (\beta_2 + \beta_3)^2}}\right) \tag{5.28}$$

$$q_2 = \text{atan}\frac{[l_2 + l_3\cos(q_3)]z_0 - l_3\,\text{sen}(q_3)\sqrt{x_0^2 + y_0^2}}{\sqrt{x_0^2 + y_0^2}[l_2 + l_3\cos(q_3)] + z_0 l_3\,\text{sen}(q_3)} \tag{5.29}$$

$$\left(\frac{}{(2l_2l_3)x_0^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}\right)^2 \tag{5.30}$$

El jacobiano de un robot antropom´orfico de tres grados de libertad est´a dado por:

$$J(\mathbf{q}) = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix} \tag{5.31}$$

$$j_{11} = [\beta_2 + \beta_3]\cos(q_1) - \text{sen}(q_1)[l_2\cos(q_2) + l_3\cos(q_2 + q_3)]$$
$$j_{12} = -\cos(q_1)[l_2\,\text{sen}(q_2) + l_3\,\text{sen}(q_2 + q_3)]$$
$$j_{13} = -l_3\cos(q_1)\,\text{sen}(q_2 + q_3)$$
$$j_{21} = [\beta_2 + \beta_3]\,\text{sen}(q_1) + \cos(q_1)[l_2\cos(q_2) + l_3\cos(q_2 + q_3)]$$
$$j_{22} = -\,\text{sen}(q_1)[l_2\,\text{sen}(q_2) + l_3\,\text{sen}(q_2 + q_3)]$$
$$j_{23} = -l_3\,\text{sen}(q_1)\,\text{sen}(q_2 + q_3)$$
$$j_{31} = 0, \quad j_{32} = l_2\cos(q_2) + l_3\cos(q_2 + q_3), \quad j_{33} = l_3\cos(q_2 + q_3)$$

Jacobiano del brazo robot de 3 gdl

El determinante del jacobiano del brazo robot de 3 gdl est´a dado por:

$$\det[J(\mathbf{q})] = -l_2l_3\cos(q_2)\,\text{sen}(q_3) - l_2l_3\,\text{sen}(q_2)\cos(q_3)^2 - l_2l_3\cos(q_2)\,\text{sen}(q_3)\cos(q3) + l_2l_3\,\text{sen}(q_2) \tag{5.32}$$

existen un n´umero infinito de configuraciones singulares para $q_2 = 0 \pm n\pi$ y $q_3 = 0 \pm n\pi$. Sin embargo, el ángulo q_1 no interviene con los puntos singulares.

Funci´on transformaci´on homog´enea del robot antropom´orfico de 3 gdl H_0^3

La funci´on transformaci´on homog´enea H_0^3 del robot antropom´orfico de 3 gdl tiene la siguiente sintaxis:

```
H0^3=H_r3gdl()
```

esta funci´on retorna H_0^3 la matriz de transformaci´on homog´enea del brazo robot con articulaciones rotacionales de 3 gdl. El c´odigo en lenguaje **MATLAB** de la funci´on transformaci´on homog´enea H_0^3 del robot de 3 gdl est´a en el cuadro 5.9.

C´odigo Fuente 5.9 H_r3gdl.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

H_r3gdl.m

```
function H=H_r3gdl()
    syms q1 q2 q3 beta1 beta2 beta3 l1 l2 l3 alpha1 alpha2 alpha3 real
    disp('Par´ametros Denavit-Hartenberg del robot planar vertical de 3 gdl')
    disp('[ l alpha d q]')
    dh=[l1, pi/2, l1+beta1, q1; l2, 0, beta2, q2; l3, 0, beta3, q3]; disp(dh)
    % H_0^1 = HRz0(q1)HTz0(l1 + β1)HTx0(0)HRx0(π/2)
    H10=simplify(HRz(q1)*HTz(l1+beta1)*HTx(0)*HRx(pi/2));
    % H_1^2 = HRz1(q2)HTz1(β2)HTx1(l2)HRx1(0)
    H21=simplify(HRz(q2)*HTz(beta2)*HTx(l2)*HRx(0)) ;
    % H_2^3 = HRz2(q3)HTz2(β3)HTx2(l3)HRx2(0)
    H32=simplify(HRz(q3)*HTz(beta3)*HTx(l3)*HRx(0));
    H30=simplify(H10*H21*H32,3); % H_0^3 = H_0^1 H_1^2 H_2^3
    [R30, cinemat_r3gdl, cero, c]=H_DH(H30); % R30, f_R(q1, q2, q3)
    H=[R30, cinemat_r3gdl;
        cero, c];
end
```

Funci´on cinem´atica directa del brazo robot de 3 gdl

La funci´on de cinem´atica directa para las coordenadas cartesianas del extremo final del brazo robot de 3 gdl se encuentra dada por:

$$[x_0, y_0, z_0]=\text{cinematica\_r3gdl}(\beta_1, l_1, q_1, \beta_2, l_2, q_2, \beta_3, l_3, q_3)$$

donde l_1 es la distancia sobre el eje z_0 donde se encuentra localizado el servomotor de la articulaci´on de la base; l_2, l_3 son las longitudes del hombro y codo, respectivamente; Ancho de los servomotores $\beta_1, \beta_2, \beta_3$; q_1, q_2, q_3 son las posiciones articulares. Esta funci´on retorna las coordenadas cartesianas (x_0, y_0, z_0).

C´odigo Fuente 5.10 cinematica_r3gdl.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinematica_r3gdl.m

```
function [x0, y0, z0]=cinematica_r3gdl(beta1,l1,q1,beta2,l2,q2,beta3,l3,q3)
    dato1=whos('beta1'); dato2=whos('l1'); dato3=whos('q1');
    dato4=whos('beta2'); dato5=whos('l2'); dato6=whos('q2');
    dato7=whos('beta3'); dato8=whos('l3'); dato9=whos('q3');
    v1=strcmp(dato1.class, 'sym');v2=strcmp(dato2.class, 'sym');
    v3=strcmp(dato3.class, 'sym');v4=strcmp(dato4.class, 'sym');
    v5=strcmp(dato5.class, 'sym');v6=strcmp(dato6.class, 'sym');
    v7=strcmp(dato7.class, 'sym');v8=strcmp(dato8.class, 'sym');
    v9=strcmp(dato9.class, 'sym'); digits(3);
    if (v1 & v2 & v3 & v4 & v5 & v6 & v7 & v8 & v9) %caso simb´olico
        x0=simplify(vpa((beta2+beta3)*sin(q1)+cos(q1)*(l3*cos(q2 + q3) + l2*cos(q2)),3));
        y0=simplify(vpa(-(beta2+beta3)*cos(q1)+sin(q1)*(l3*cos(q2 + q3) + l2*cos(q2)),3));
        z0=simplify(vpa( l1 +beta1+ l3*sin(q2 + q3) + l2*sin(q2) ,3));
        x0=vpa(x0); y0=vpa(y0);
        else %caso num´erico
        x0=(beta2+beta3)*sin(q1)+cos(q1).*(l3*cos(q2 + q3) + l2*cos(q2));
        y0=-(beta2+beta3)*cos(q1)+sin(q1).*(l3*cos(q2 + q3) + l2*cos(q2));
        z0=l1 +beta1+ l3*sin(q2 + q3) + l2*sin(q2);
    end
end
```

Funci´on cinem´atica inversa del brazo robot de 3 gdl

La funci´on de cinem´atica inversa del brazo robot de 3 gdl tiene la siguiente forma sint´actica:

$$[q_1, q_2, q_3]=\text{cinv\_r3gdl}(\beta_2, \beta_3, l_2, l_3, x_0, y_0, z_0)$$

donde los argumentos de entrada son los par´ametros geom´etricos del robot β_1, β_2, β_3 representan espesor de los servomotores de la base, hombro y codo, respectivamente; l_1 es la distancia sobre el eje z_0 donde se ubica la articulaci´on de la base; l_2 y l_3 longitudes del hombro y codo, respectivamente. Las coordenadas cartesianas del extremo final del robot x_0, y_0, z_0 se encuentran expresadas en el sistema $\Sigma_0(x_0, y_0, z_0)$. Esta funci´on retorna las coordenadas articulares (q_1, q_2, q_3) de la base, hombro y codo, respectivamente.

C´odigo Fuente 5.11 cinv_r3gdl.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinv_r3gdl.m

```
1  function [q1, q2, q3]=cinv_r3gdl(beta2,beta3,l2,l3,x0,y0,z0)
2      v1=sqrt(x0.*x0+y0.*y0-(beta2+beta3)^2);
3      %articulaci´on de la base q1
4      q1=atan(y0./x0)-atan((beta2+beta3)./v1);
5      c3=(x0.*x0+y0.*y0+z0.*z0-l2^2-l3^2)./(2*l2*l3);
6      s3=sqrt(1-c3.*c3);
7      %articulaci´on del codo q3
8      q3=atan(s3./c3);
9      2=(z.*(l2+l3.*c3)-l3*s3.*sqrt(x0.*x0+y0.*y0))./(x0.*x0+y0.*y0+z0.*z0);
10     2=((l2+l3.*c3).*sqrt(x0.*x0+y0.*y0)+l3*s3.*z)./(x0.*x0+y0.*y0+z0.*z0);
11     q2=atan(s2./c2); %articulaci´on del hombro
12 end
```

♣♣♣ Ejemplo 5.3

Diseñar un programa en lenguaje **MATLAB** que permita desplegar en forma simb´olica los par´ametros DH, cinem´atica cartesiana y el jacobiano del robot antropom´orfico de 3 gdl. Asimismo programar una aplicaci´on donde el extremo final del robot trace trayectorias circulares sobre el eje z_0.

Soluci´on

El programa 5.12 presenta el c´odigo fuente en lenguaje **MATLAB** que despliega en forma simb´olica los par´ametros DH, cinem´atica cartesiana, matriz jacobiana y su determinante (ver l´ıneas 6 a la 16).

De la l´ınea 23 a la 25 se encuentra la implementaci´on de la trayectoria circular con barrido de 1 mm sobre el eje z_0: $[x \quad y \quad z]^T = [x_c + r\ \text{sen}(t) \quad y_c + r\ \cos(t) \quad t]$, siendo x_c, y_c el centro del c´ırculo, r es el radio, t es la evoluci´on del tiempo. En la l´ınea 26 la trayectoria circular es convertida a coordenadas articulares usando la cinem´atica inversa y en la l´ınea 27 se obtiene las coordenadas que el extremo final del robot traza como se muestra en la figura 5.11.

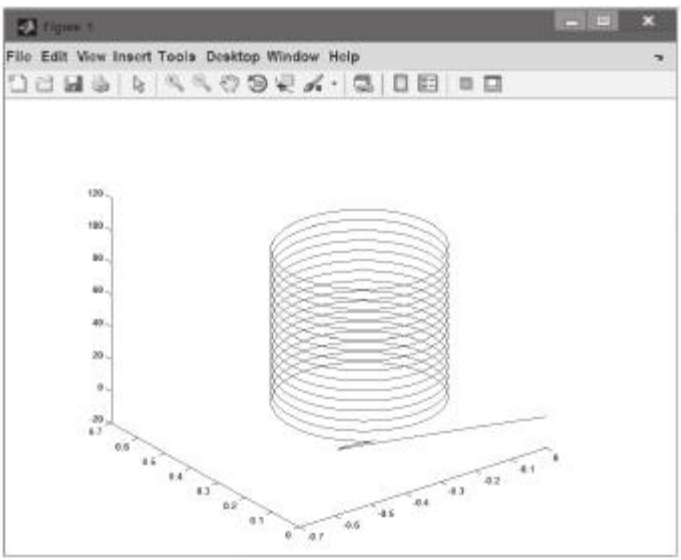


Figura 5.11 Trazo que realiza el extremo final del brazo robot de 3 gdl.

C´odigo Fuente 5.12 cap5_r3gdl.m

%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.

cap5_r3gdl.m

```
clc;
clear all;
close all;
format short
syms q1 q2 q3 beta1 beta2 beta3 l1 l2 l3 alpha1 alpha2 alpha3 real
H30=H_r3gdl() ;
disp('Tansformaci´on homog´enea del robot antropom´orfico de 3 gdl');
disp(H30);
[R30, cinemat_r3gdl,cero, c]=H_DH(H30) ;
disp('Matriz de rotaci´on');
disp(R30);
disp('cinem´atica directa');
disp(cinemat_r3gdl);
[x0, y0,z0]= cinematica_r3gdl(beta1,l1,q1,beta2,l2,q2,beta3,l3,q3); disp([x0; y0;z0])
jac_r3gdl=jacobian([x0; y0;z0], [q1;q2;q3])
det_r3gdl=simplify(vpa(det(jac_r3gdl),3))
%ejemplo num´erico
l1=0; l2=0.45; l3=0.45;
t=0:0.001:100;
xc=0.3; yc=-0.3; r=0.20;
beta1=0.12; beta2=0.01; beta3=0.01;
q1=[]; q2=[]; q3=[];
x=xc+r*sin(t) ;
y=yc+r*cos(t);
z=t;
[q1, q2, q3]=cinv_r3gdl(beta2,beta3,l2,l3,x,y,z);
[x0, y0,z0]=cinematica_r3gdl(beta1,l1,q1,beta2,l2,q2,beta3,l3,q3);
plot3(x0,y0,z0)
```

5.3 Configuraci´on SCARA (RRP)

La configuraci´on **SCARA** (Selective Compliance Assembly Robot Arm) representa una geometr´ıa especial de robots industriales; es un brazo planar antropom´orfico con dos articulaciones rotacionales y la tercera articulaci´on es prism´atica o lineal para manipular objetos. La figura 5.12 muestra un ejemplo de un robot industrial en la configuraci´on SCARA. Esta configuraci´on aprovecha las ventajas que proporciona el brazo robot antropom´orfico de 2 gdl movi´endose en el plano horizontal; en este caso la energ´ıa potencial es constante (par gravitacional cero), la estructura mec´anica es de alta rigidez para soportar cargas en forma vertical y para control de fuerza. La configuraci´on SCARA es adecuada para tareas de ensamble con pequeños objetos.

Figura 5.12 Robot KR 10 SCARA R600 (Compan´ıa KUKA).

Cinem´atica directa cartesiana

El espacio de trabajo del robot SCARA es un cilindro que se describe en la figura 5.13, el radio del cilindro es igual a la suma de las longitudes de los eslabones de las articulaciones rotacionales $l_2 + l_3$. El origen del sistema de referencia fijo $\Sigma_0(x_0, y_0, z_0)$ se coloca r´ıgidamente en el piso, de tal forma que sobre el eje z_0 la articulaci´on de la base del robot queda a una distancia l_1 m´as el ancho del servomotor y espesor de la barra met´alica representado por β_1. El eje de rotaci´on de la primera articulaci´on rotacional q_1 (base) coincide con el eje z_0. El sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ se encuentra colocado r´ıgidamente en el extremo final del primer eslab´on, donde se intercepta al eje de rotaci´on de la segunda articulaci´on rotacional q_2 con el eje x_0; las coordenadas del origen del sistema $\Sigma_1(x_1, y_1, z_1)$ con respecto

a $\Sigma_0(x_0, y_0, z_0)$ son $[l_2 \cos(q_1) \quad l_2 \operatorname{sen}(q_1), l1 + \beta_1]^T$. Cuando $q_1 = 0$ significa que el plano $x_1 - y_1$ mantiene una rotaci´on con referencia al plano fijo $x_0 - y_0$, entonces la matriz $R_{1\,0} = R_{z0}(q_1)$ describe la rotaci´on del sistema $\Sigma_1(x_1, y_1, z_1)$ con respecto al sistema fijo $\Sigma_0(x_0, y_0, z_0)$. Los ejes z_0 y z_1 son paralelos entre s´ı y mantienen la misma direcci´on, por lo tanto el ´angulo que existe entre ellos es $\alpha_1 = 0$. El sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ mide la variable articular q_2 alrededor del eje z_1. El plano $x_1 - y_1$ tiene una rotaci´on por un ´angulo q_1 con respecto al plano $x_0 - y_0$.

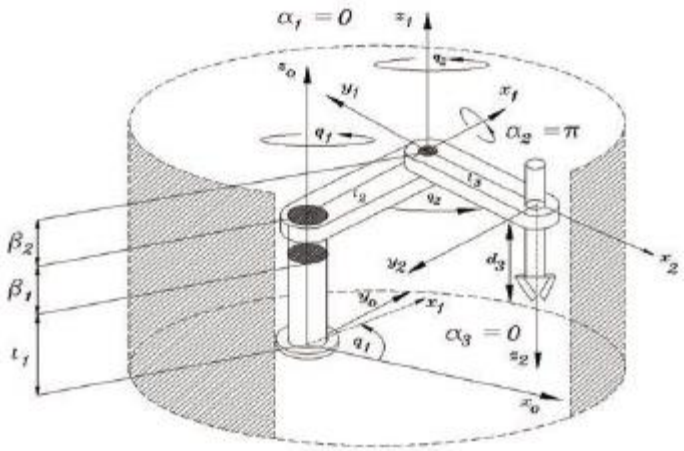


Figura 5.13 Espacio de trabajo del robot SCARA.

El sistema de referencia $\Sigma_2(x_2, y_2, z_2)$ se coloca en la parte final del segundo eslab´on, justo en la intersecci´on del eje x_1 con el desplazamiento lineal d_3. Las coordenadas del origen del sistema $\Sigma_2(x_2, y_2, z_2)$ con respecto a $\Sigma_1(x_1, y_1, z_1)$ son $[l_3 \cos(q_2) \quad l_3 \operatorname{sen}(q_2) \quad l1 + \beta_1 + \beta_2]^T$, siendo β_2 el ancho del segundo servomotor m´as el espesor de la segunda barra met´alica. El ´angulo que hay entre los ejes x_1 y x_2 es q_2. Cuando $q_2 = 0$ se encuentran alineados dichos ejes, para el caso donde $q_2 = 0$, la rotaci´on que hay entre los sistemas de referencia $\Sigma_1(x_1, y_1, z_1)$ y $\Sigma_2(x_2, y_2, z_2)$ es descrita por la matriz $R_{2\,1} = R_{z1}(q_2)$. La medici´on de la tercera articulaci´on lineal d_3 se realiza sobre el eje z_2, por lo tanto este eje est´a alineado sobre su desplazamiento lineal teniendo movimiento positivo en direcci´on hacia abajo, que corresponde a una direcci´on negativa del eje z_1 o z_0 como se muestra en la figura 5.13. El sistema de referencia $\Sigma_2(x_2, y_2, z_2)$ se obtiene a partir del sistema $\Sigma_1(x_1, y_1, z_1)$, rotando un ´angulo $\alpha_2 = \pi$ alrededor del eje x_1, por lo tanto los ejes z_2 y z_1 son paralelos entre s´ı, pero con direcci´on contraria.

En la configuraci´on SCARA todos los ejes z_i con $i = 0, 1, 2$ son paralelos entre s´ı, pero debe notarse que el eje z_2 mantiene una rotaci´on de 180 grados con respecto al eje z_1 o al eje z_0. La raz´on por la cual el eje z_2 mantiene un direcci´on positiva hacia abajo es por aspectos t´ecnicos, es decir los objetos a manipular se encuentran frente de la herramienta de trabajo. Se ha considerado que el eje z_2 es paralelo al eje de la herramienta ($\alpha_3 = 0$).

Con la anterior descripci´on de los sistemas de referencia $\Sigma_0(x_0, y_0, z_0)$ al $\Sigma_2(x_2, y_2, z_2)$ para la configuraci´on SCARA produce el siguiente conjunto de par´ametros Denavit-Hartenberg que se presenta en la tabla 5.4:

Tabla 5.4 Par´ametros DH del robot SCARA

Eslab´on	l_i	α_i	d_i	θ_i
1	0	0	$l_1 + \beta_1$	q_1
2	l_2	π	β_2	q_2
3	l_3	0	d_3	0

De acuerdo con la tabla 5.4 el robot SCARA tiene las siguientes matrices de transformaci´on homog´enea:

$$H_0^1 = H_{Rz_0}(q_1)\,H_{Tz_0}(l_1+\beta_1)\,H_{Tx}(l_2)\,H_{Rx_1}(0)$$

$$= \begin{bmatrix} \cos(q_1) & -\operatorname{sen}(q_1) & 0 & l_2\cos(q_1) \\ \operatorname{sen}(q_1) & \cos(q_1) & 0 & l_2\operatorname{sen}(q_1) \\ 0 & 0 & 1 & l_1+\beta_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.33}$$

$$H_1^2 = H_{Rz_1}(q_2)\,H_{Tz_1}(\beta_2)\,H_{Tx_1}(l_3)\,H_{Rx_1}(\pi)$$

$$= \begin{bmatrix} \cos(q_2) & \operatorname{sen}(q_2) & 0 & l_3\cos(q_2) \\ \operatorname{sen}(q_2) & -\cos(q_2) & 0 & l_3\operatorname{sen}(q_2) \\ 0 & 0 & -1 & \beta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.34}$$

$$H_2^3 = H_{Rz_2}(0)\,H_{Tz_2}(d_3)\,H_{Tx_2}(0)\,H_{Rx_2}(0)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.35}$$

$$H_0^3 = H_0^1 H_1^2 H_2^3 = \begin{bmatrix} \cos(q_1+q_2) & \operatorname{sen}(q_1+q_2) & 0 & l_2\cos(q_1)+l_3\cos(q_1+q_2) \\ \operatorname{sen}(q_1+q_2) & -\cos(q_1+q_2) & 0 & l_2\operatorname{sen}(q_1)+l_3\operatorname{sen}(q_1+q_2) \\ 0 & 0 & -1 & l_1+\beta_1+\beta_2-d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.36}$$

El modelo de cinem´atica directa para el robot SCARA est´a dada por la siguiente expresi´on:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \mathbf{f}_R(\mathbf{q}) = \begin{bmatrix} l_2\cos(q_1)+l_3\cos(q_1+q_2) \\ l_2\operatorname{sen}(q_1)+l_3\operatorname{sen}(q_1+q_2) \\ l_1+\beta_1+\beta_2-d_3 \end{bmatrix} \tag{5.37}$$

Funci´on transformaci´on homog´enea SCARA

El conjunto de ecuaciones que describen la rotaci´on y coordenadas cartesianas del extremo final del robot SCARA (5.33)-(5.37) permiten desarrollar una librer´ıa con variables simb´olicas para **MATLAB** H_SCARA() con la siguiente sintaxis:

```
H03= H_SCARA()
```

esta funci´on retorna la matriz de transformaci´on homog´enea H_0^3 la cual est´a formada con la matriz de rotaci´on R_{30} que relaciona la rotaci´on del extremo final del robot con respecto al sistema fijo $\Sigma_0(x_0, y_0, z_0)$ y la cinem´atica directa que relaciona las coordenadas articulares con las coordenadas cartesianas $\mathbf{f}_R(\mathbf{q})$. Adem´as, tambi´en despliega la tabla 5.4 con los par´ametros Denavit-Hartenberg del robot SCARA.

El programa 5.13 contiene el c´odigo **MATLAB** de la funci´on transformaci´on homog´enea del robot SCARA:

C´odigo Fuente 5.13 H_SCARA.m

```
%Transformaci´on homog´enea del robot SCARA H_0^3
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

H_SCARA.m

```
function H=H_SCARA()
    syms q1 q2 q3 beta1 beta2 l1 l2 l3 d1 d2 d3 alpha1 alpha2 alpha3 real
    disp('Transformaci´on Homog´enea H_0^3del robot SCARA')
    disp('Par´ametros Denavit-Hartenberg del robot SCARA')
    disp('[ l alpha d q]')
    %tabla de par´ametros DH del robot SCARA
    dh=[0, 0,l1+beta1, q1; l2, pi, beta2, q2; l3, 0, d3, 0]; %despliega par´ametros
    DH del robot SCARA
    disp(dh)
    %c´alculo de las matrices de transformaci´on homog´enea de cada articulaci´on
    %H_0^1 = H_Rz0(q1) H_Tz0(l1 + β1) H_Tx(l2) H_Rx(0)
    H10=HRz(q1)*HTz(l1+beta1)*HTx(l2)*HRx(0);
    %H_1^2 = H_Rz1(q2) H_Tz1(β2) H_Tx1(l3) H_Rx1(π)
    H21=HRz(q2)*HTz(beta2)*HTx(l3)*HRx(pi);
    %H_2^3 = H_Rz2(0) H_Tz2(d3) H_Tx2(0) H_Rx2(0)
    H32=HRz(0)*HTz(d3)*HTx(0)*HRx(0);
    %transformaci´on homog´enea del robot SCARA
    %H_0^3 = H_0^1 H_1^2 H_2^3
    H30=simplify(H10*H21*H32);
    %deducci´on de la matriz de rotaci´on R_30
    %as´ı como la cinem´atica directa cartesiana f_R(q1, q2, d3) del robot SCARA
    [R30, cinemat_scara, cero, c]=H_DH(H30);
    %estructura de la matriz homog´enea
    H=[R30, cinemat_scara;
        cero, c];
end
```

Funci´on cinem´atica directa

La funci´on cinem´atica directa del robot SCARA se define como:

```
[x0, y0, z0]=cinematica_SCARA(β1, β2, l1, l2, l3, q1, q2, d3)
```

donde los argumentos de entrada son los par´ametros geom´etricos: β_1, β_2, l_1, l_2, l_3 y las variables articulares q_1, q_2, d_3. Esta funci´on retorna las coordenadas x_0, y_0, z_0 en espacio cartesiano del sistema de referencia fijo $\Sigma_0(x_0, y_0, z_0)$.

En el cuadro 5.14 se presenta el c´odigo en lenguaje **MATLAB** de la cinem´atica directa del robot SCARA.

C´odigo Fuente 5.14 cinematica_SCARA.m

```
%cinem´atica directa robot SCARA
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinematica_SCARA.m

```
1 function [x0, y0, z0]=cinematica_SCARA(beta1,beta2,l1,l2,l3,q1,q2,d3)
2     x0=l2*cos(q1)+l3*cos(q1+q2);
3     y0=l2*sin(q1)+l3*sin(q1+q2);
4     z0=l1+beta1+beta2-d3;
5 end
```

Jacobiano del robot SCARA

El jacobiano del robot SCARA se encuentra de la siguiente forma:

$$J(q_1, q_2, d_3) = \frac{\partial \mathbf{f}_R(\mathbf{q})}{\partial \mathbf{q}} \tag{5.38}$$

$$= \begin{bmatrix} -l_2\,\mathrm{sen}(q_1) - l_3\,\mathrm{sen}(q_1+q_2) & -l_3\,\mathrm{sen}(q_1+q_2) & 0 \\ l_2\cos(q_1) + l_3\cos(q_1+q_2) & l_3\cos(q_1+q_2) & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

el determinante del jacobiano del robot SCARA tiene la siguiente forma:

$$\det[\mathbf{J}(q_1, q_2, d_3)] = -l_2 l_3 \operatorname{sen}(q_2) \tag{5.39}$$

lo que significa que se presentan problemas de singularidad cuando $q_2 = 0, \pm n\pi$.

Cinem´atica inversa del robot SCARA

El ancho de los servomotores y espesor de las barras met´alicas (β_1, β_2) no tienen proyecci´on sobre el plano $x_0 - y_0$; s´olo contribuyen en la coordenada z_0, como en el caso del p´endulo y robot antropom´orfico de 2 gdl, entonces la cinem´atica inversa del robot SCARA toma la siguiente expresi´on:

$$q_2 = \operatorname{acos}\left(\frac{x_{20}^2 + y_0^2 - l_2^2 - l_3^2}{2 l_2 l_3}\right) \tag{5.40}$$

$$q_1 = \operatorname{atan}\frac{y_0}{x_0} - \operatorname{atan}\frac{l_3 \operatorname{sen}(q_2)}{l_2 + l_3 \cos(q_2)} \tag{5.41}$$

$$d_3 = l_1 + \beta_1 + \beta_2 - z_0 \tag{5.42}$$

Funci´on cinem´atica inversa del robot SCARA

La funci´on cinem´atica inversa del robot SCARA tiene la siguiente sintaxis:

```
if  [d3, q2, q1]=cinv_SCARA(β1, β2, l1, l2, l3, x0, y0, z0)
```

donde $\beta_1, \beta_2, l_1, l_2, l_3$ son los par´ametros geom´etricos del robot SCARA, x_0, y_0, z_0 son las coordenadas en espacio cartesiano en el sistema fijo $\Sigma_0(x_0, y_0, z_0)$; esta funci´on retorna las coordenadas articulares d_3, q_2, q_1.

El programa 5.15 contiene el c´odigo fuente **MATLAB** de la cinem´atica inversa del robot SCARA, para facilitar la implementaci´on de procedimientos recursivos con pase de par´ametros escalares o vectoriales, la cinem´atica inversa es programada con operaciones entre arreglos.

Código Fuente 5.15 cinv_SCARA.m

```
%cinemática inversa robot SCARA
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega, Fernando Reyes Cortés.
%Capítulo 5 Cinemática directa cartesiana.
```

cinv_SCARA.m

```
1 function [d3 q2 q1]=cinv_SCARA(beta1,beta2,l1,l2,l3,x0,y0,z0)
2     q2=acos((x0.*x0+y0.*y0-l2*l2-l3*l3)/(2*l2*l3));
3     q1=atan(y0./x0)-atan((l3*sin(q2))./(l2+l3*cos(q2)));
4     d3=l1+beta1+beta2-z0;
5 end
```

♣ ♣ Ejemplo 5.4

Realizar una aplicación numérica del robot SCARA para que el extremo final lleve a cabo el trazo de una rosa polar de radio $r = 0.1$ m y con centro en las coordenadas cartesianas $[x_0 \quad y_0 \quad z_0]^T = [0.3 \quad -0.3 \quad -0.5]^T$ m. Asimismo presentar en forma simbólica la matriz jacobiana, determinante y cinemática directa.

Solución

En el cuadro 5.16 se presenta el código en lenguaje **MATLAB** en variables simbólicas de la matriz de transformación homogénea H_0^3 del robot SCARA usando la función H30=H_SCARA() (ver línea 4). En la línea 5 empleando la función H_DH(H30) se obtiene la matriz de rotación R_{30} a través de la función $[R_{30}, \mathbf{f}_R(\mathbf{q}), 0^T, 1]$=H_DH(H30). La matriz jacobiano se obtiene en la línea 6 y su determinante en la línea 7. La aplicación numérica consiste en que el extremo final del robot SCARA trace una rosa polar de radio $r = 0.1$ m, y con centro en las coordenadas cartesianas $[x_0 \quad y_0 \quad z_0]^T = [0.3 \quad -0.3 \quad -0.5]^T$ m. La ecuación en coordenadas cartesianas de la rosa polar (lineas 19 a la 21) está dada por:

$$r = 0.1\,\mathrm{sen}\left(\frac{6t}{7}\right) \quad x = x_c + r\,\mathrm{sen}^3(t)$$

$y = y_c + r\cos^3(t) \quad z = -0.5$

donde x_c, y_c representan el centro de la flor, r es el radio de los p´etalos y t es el tiempo.

El c´alculo de las coordenadas cartesianas de la rosa polar se realiza en forma iterativa usando la instrucci´on `for···end` (l´ıneas 16 a la 24). En la l´ınea 24 se convierten las coordenadas cartesianas de la rosa polar a coordenadas articulares del robot empleando la funci´on `cinv_SCARA(···)`. En la l´ınea 27 se usa la funci´on `cinematica_SCARA(···)` para convertir las coordenadas articulares del robot a coordenadas cartesianas, entonces el extremo final del robot realiza el trazo de la figura 5.14. El programa que se presenta en el cuadro 5.17 realiza la misma funci´on del programa 5.16; la finalidad de mostrarlo es ilustrar el c´alculo de las coordenadas cartesianas de la rosa polar sin utilizar la forma recursiva de la instrucci´on `for`, es decir se emplean operaciones con arreglos en lugar del proceso iterativo con escalares.

Observe que en las funciones `cinv_SCARA` y `cinematica_SCARA` se encuentran implementadas para soportar ambos tipos de pase de par´ametros (recursivo y por arreglos), lo que facilita la implementaci´on de varias aplicaciones en control de robots manipuladores.

Es muy importante aclarar que el dibujo realizado por el robot SCARA (figura 5.14) no implica control del robot o una simulaci´on de la din´amica del mismo. Estos programas ilustran s´olo el empleo de la cinem´atica inversa y su conversi´on a coordenadas cartesianas del extremo final del robot para trazar una figura en su espacio de trabajo.

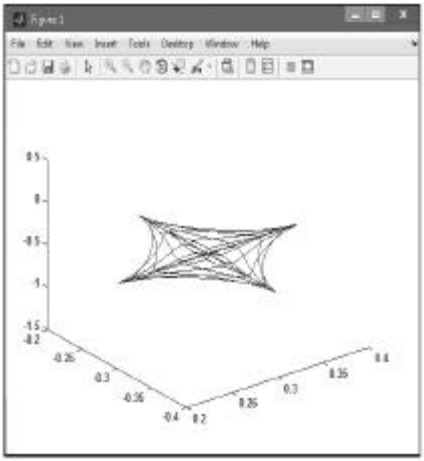


Figura 5.14 Trayectoria del robot SCARA.

Cuando las coordenadas articulares del robot provienen de un sistema din´amico como puede ser la ecuaci´on en lazo cerrado (modelo din´amico del robot y algoritmo de control), entonces se trata de una simulaci´on de un esquema de control. La cinem´atica no reproduce los fen´omenos f´ısicos del robot.

C´odigo Fuente 5.16 SCARA.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

SCARA.m

```
clc; clear all; close all; format short
syms theta l d alpha q real
syms q1 q2 d3 beta1 beta2 l1 l2 l3 alpha1 alpha2 alpha3 real
H30=H_SCARA() ;
[R30, frq_scara, cero, c]=H_DH(H30);
jac_scara=jacobian(frq_scara, [q1;q2;d3]);
det_scara=simplify(det(jac_scara))
%ejemplo num´erico
%par´ametros geom´etricos del robot SCARA
l1=0.45; l2=0.45; l3=0.45; beta1=0.1; beta2=0.1;
t=0:0.001:100;
%ecuaci´on de la figura centro en xc,yc y radio r
xc=0.3; yc=-0.3;
q1=[]; q2=[];z0=[]; d3=[];
[n m]=size(t); % dimensi´on del vector de tiempo
for k=1:m%forma recursiva
    %ecuaci´on cartesiana de la figura rosa polar
    r=0.1*sin(6*t(k)/7); % radio de la figura
    x=xc+r*(sin(t(k)))^3 ;
    y=yc+r*(cos(t(k)))^ 3;
    z=-0.5 ;
    % conversi´on de coordenadas cartesianas de la rosa polar
    %a coordenadas articulares del robot SCARA
    [d3(k), q2(k), q1(k)]=cinv_SCARA(beta1,beta2,l1,l2,l3,x,y,z) ;
end
%trayectoria del robot SCARA en el sistema Σ0(x0, y0, z0)
[x0, y0, z0]=cinematica_SCARA(beta1,beta2,l1,l2,l3,q1,q2,d3);
plot3(x0,y0,z0)
```

C´odigo Fuente 5.17 SCARA1.m

%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.

SCARA1.m

```
1  clc; clear all; close all;
2  format short
3  syms theta l d alpha q real
4  syms q1 q2 d3 beta1 beta2 l1 l2 l3 alpha1 alpha2 alpha3 real
5  H30=H_SCARA();
6  [R30, frq_scara, cero, c]=H_DH(H30);
7  jac_scara=jacobian(frq_scara, [q1;q2;d3]);
8  det_scara=simplify(det(jac_scara))
9  %ejemplo num´erico
10 %par´ametros geom´etricos del robot SCARA
11 l1=0.45; l2=0.45; l3=0.45; beta1=0.1; beta2=0.1;
12 t=0:0.001:100;
13 %ecuaci´on de la rosa polar centro en xc,yc y radio r
14 xc=0.3; yc=-0.3;
15 q1=[]; q2=[];z0=[]; d3=[];
16 [n m]=size(t); % dimensi´on del vector de tiempo
17 r=0.1*sin(6*t/7); %radio de la rosa polar
18 %ecuaci´on cartesiana de la rosa polar
19 x=xc+r.*(sin(t).^3);
20 y=yc+r.*(cos(t).^3);
21 z(1:m)=-0.5; %coordenada sobre el eje z0
22 % conversi´on de coordenadas cartesianas de la rosa polar
23 %a coordenadas articulares del robot SCARA
24 [d3, q2, q1]=cinv_SCARA(beta1,beta2,l1,l2,l3,x,y,z);
25 %trayectoria del robot SCARA en el sistema Σ0(x0, y0, z0)
26 [x0, y0, z0]=cinematica_SCARA(beta1,beta2,l1,l2,l3,q1,q2,d3);
27 plot3(x0,y0,z0)
```

5.4 Robot esf´erico (RRP)

La **configuraci´on esf´erica** de robots manipuladores presenta dos articulaciones rotacionales (base y hombro) y la articulaci´on del codo corresponde al tipo prism´atica o lineal como se muestra en la figura 5.15. Los ejes de movimiento de las articulaciones son mutuamente perpendiculares entre s´ı. Dentro de los ejemplares de esta configuraci´on se encuentra el robot Standford, cuya principal aplicaci´on es el mecanizado de piezas automotrices y la manipulaci´on de objetos sobre piso.

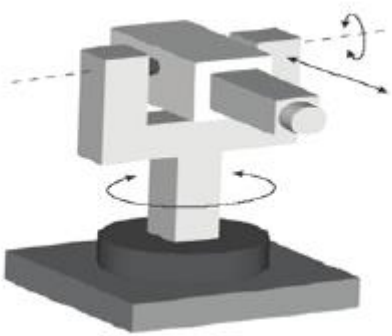


Figura 5.15 Robot manipulador en configuraci´on esf´erica.

Cinem´atica directa cartesiana

El espacio de trabajo de la configuraci´on esf´erica corresponde a una esfera hueca, cuyo radio se encuentra en funci´on del desplazamiento lineal d_3 de la articulaci´on prism´atica del codo. El eje z_0 se alinea con el eje de giro de la articulaci´on rotacional de la base q_1 y el origen del sistema de referencia fijo $\Sigma_0(x_0, y_0, z_0)$ se ubica en el piso de tal forma que la articulaci´on de la base est´a a una altura l_1 sobre el eje z_0. El ancho del servomotor de la base y el espesor de la placa met´alica est´an representados por β_1. Por otro lado, el eje z_1 determina la medici´on de la variable articular q_2 del hombro y se encuentra alineado con el eje de rotaci´on de esta articulaci´on. Los ejes z_1 y z_0 son perpendiculares entre s´ı. El sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ se coloca en el dorso del servomotor de la articulaci´on del hombro y las coordenadas del origen de este sistema respecto al sistema fijo $\Sigma_0(x_0, y_0, z_0)$ est´an en: $[\beta_1 \cos(q_1) \quad \beta_1 \operatorname{sen}(q_1) \quad l_1 + \beta_1]^T$. El eje z_2 se alinea con el desplazamiento lineal de

la articulaci´on prism´atica d_3 y las coordenadas del origen del sistema $\Sigma_2(x_2, y_2, z_2)$ con referencia al sistema $\Sigma_1(x_1, y_1, z_1)$ se encuentran localizadas en: $[0 \quad 0 \quad \beta_2]^T$. Los ejes z_0, z_1 y z_2 son perpendiculares entre s´ı.

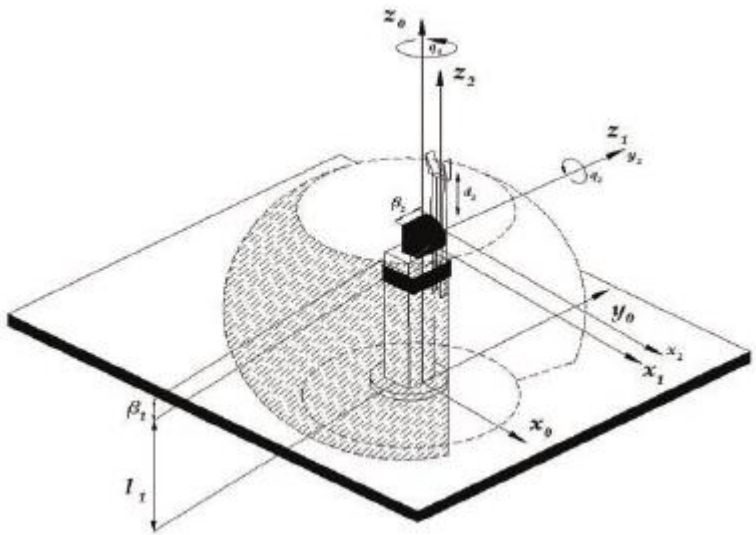


Figura 5.16 Configuraci´on esf´erica.

La tabla 5.5 presenta los par´ametros de la convenci´on Denavit-Hartenberg para el robot en la configuraci´on esf´erica.

Tabla 5.5 DH del robot esf´erico

Eslab´on	l_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	$l_1 + \beta_1$	q_1
2	0	$\frac{\pi}{2}$	β_2	q_2
3	0	0	d_3	0

Las matrices de transformaci´on homog´enea para el robot en la configuraci´on esf´erica tienen la siguiente estructura:

$$H_0 = H_{R_{z_0}}(q_1)\, H_{T_{z_0}}(l_1 + \beta_1)\, H_{T_{x_0}}(0)\, H_{R_{x_0}}(-\tfrac{pi}{2}$$

$$= \begin{bmatrix} \cos(q_1) & 0 & -\operatorname{sen}(q_1) & 0 \\ \operatorname{sen}(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & l_1+\beta_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.43}$$

$$H_1^2 = H_{Rz_1}(q_2)\,H_{Tz_1}(\beta_2)\,H_{Tx_0}(0)\,H_{Rx_1}(\alpha_2)$$

$$= \begin{bmatrix} \cos(q_2) & 0 & \operatorname{sen}(q_2) & 0 \\ \operatorname{sen}(q_2) & 0 & -\cos(q_2) & 0 \\ 0 & 1 & 0 & \beta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.44}$$

$$H_2^3 = H_{Rz_2}(0)\,H_{Tz_2}(d_3)\,H_{Tx_2}(0)\,H_{Rx_2}(0)$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.45}$$

$$H_0^3 = H_0^1 H_1^2 H_2^3 \tag{5.46}$$

$$= \begin{bmatrix} \cos(q_1)\cos(q_2) & -\operatorname{sen}(q_1) & \cos(q_1)\operatorname{sen}(q_2) & -\beta_2\operatorname{sen}(q_1)+d_3\operatorname{sen}(q_2)\cos(q_1) \\ \operatorname{sen}(q_1)\cos(q_2) & \cos(q_1) & \operatorname{sen}(q_1)\operatorname{sen}(q_2) & \beta_2\cos(q_1)+d_3\operatorname{sen}(q_2)\operatorname{sen}(q_1) \\ -\operatorname{sen}(q_2) & 0 & \cos(q_2) & l_1+\beta_1+d_3\cos(q_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.47}$$

La cinem´atica directa de un robot manipulador en configuraci´on esf´erica es:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\beta_2\operatorname{sen}(q_1)+d_3\operatorname{sen}(q_2)\cos(q_1) \\ \beta_2\cos(q_1)+d_3\operatorname{sen}(q_2)\operatorname{sen}(q_1) \\ l_1+\beta_1+d_3\cos(q_2) \end{bmatrix} \tag{5.48}$$

El jacobiano del robot esf´erico tiene la siguiente forma:

$$J(q_1, q_2, d_3) = \begin{matrix} -\beta_2\cos(q_1)-d_3\operatorname{sen}(q_1)\operatorname{sen}(q_2) & d_3\cos(q_1)\cos(q_2) & \cos(q_1)\operatorname{sen}(q_2) \\ d_3\cos(q_1)\operatorname{sen}(q_2)-\beta_2\operatorname{sen}(q_1) & d_3\cos(q_2)\operatorname{sen}(q_1) & \operatorname{sen}(q_1)\operatorname{sen}(q_2) \\ 0 & -d_3\operatorname{sen}(q_2) & \cos(q_2) \end{matrix} \tag{5.49}$$

Determinante $\det[J(q_1, q_2, d_3)] = -d_3^2\operatorname{sen}(q_2)$

Funci´on transformaci´on homog´enea robot esf´erico

La funci´on matriz de transformaci´on homog´enea `H_esferico()` del brazo robot en la configuraci´on esf´erica tiene la siguiente sintaxis:

```
if    H03 = H_esferico()
```

esta funci´on retorna la matriz de transformaci´on homog´enea H_0^3 integrada por la matriz de rotaci´on R_3^0 que describe la rotaci´on del extremo final del robot con respecto al sistema fijo $\Sigma_0(x_0, y_0, z_0)$ y la cinem´atica directa cartesiana $\mathbf{f}_R(\mathbf{q})$ que relaciona las coordenadas articulares. Adem´as, tambi´en despliega en forma simb´olica la tabla 5.5 con los par´ametros Denavit-Hartenberg. El programa 5.18 contiene el c´odigo **MATLAB** de la funci´on transformaci´on homog´enea del robot esf´erico.

C´odigo Fuente 5.18 H_esferico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

H_esferico.m

```
1  function H=H_esferico()
2      syms q1 q2 q3 beta1 beta2 l1 l2 l3 d1 d2 d3 alpha1 alpha2 alpha3 real
3      disp('Par´ametros Denavit-Hartenberg del robot SCARA') disp('[ l alpha d q]')
4      dh=[0, -pi/2, l1+beta1, q1; 0, pi/2, beta2, q2; 0, 0, d3, 0]; disp(dh)
5      %H_0^ = H_Rz0(q1)H_Tz0(l1 + β1)H_Tx0(0)H_Rx0(−π)2
6      H10=HRz(q1)*HTz(l1+beta1)*HTx(0)*HRx(-pi/2);
7      %H_1^ = H_Rz1(q2)H_Tz1(β2)H_Tx1(0)H_Rx1(π)2
8      H21=HRz(q2)*HTz(beta2)*HTx(0)*HRx(pi/2);
9      H32=HRz(0)*HTz(d3)*HTx(0)*HRx(0); %H_2 3 H_Rz2(0)H_Tz2(d3)H_Tx2(0)H_Rx2(0)
10     H30=simplify(H10*H21*H32);%H_0 3= H_0^1 H_1^2 H_2 3
11     [R30, cinemat_esferico, cero, c]=H_DH(H30);
12     H=[R30, cinemat_esferico; cero, c];
13 end
```

Funci´on cinem´atica directa del robot esf´erico

La funci´on de cinem´atica directa que relaciona las coordenadas cartesianas del extremo final con las coordenadas articulares del brazo robot en configuraci´on esf´erica est´a dada por la siguiente sintaxis (el c´odigo en lenguaje **MATLAB** se encuentra en el cuadro 5.19):

$$[x_0, y_0, z_0]=\text{cinematica\_esferico}(\beta_1, \beta_2, l_1, q_1, q_2, d_3)$$

C´odigo Fuente 5.19 cinematica_esferico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinematica_esferico.m

```
1  function [x0, y0,z0]=cinematica_esferico(beta1,beta2,l1,q1,q2,d3)
2      dato1=whos('beta1'); dato2=whos('beta2'); dato3=whos('l1');
3      dato4=whos('q1'); dato5=whos('q2'); dato6=whos('d3');
4      v1=strcmp(dato1.class, 'sym'); v2=strcmp(dato2.class, 'sym');
5      v3=strcmp(dato3.class, 'sym'); v4=strcmp(dato4.class, 'sym');
6      v5=strcmp(dato5.class, 'sym'); v6=strcmp(dato6.class, 'sym');
7      digits(3);
8      if ( v1 & v2 & v3 & v4 & v5 &v6)% caso simb´olico
9          x0=simplify(vpa(-beta2*sin(q1)+d3*cos(q1)*sin(q2),3));
10         y0=simplify(vpa(beta2*cos(q1) + d3*sin(q1)*sin(q2),3));
11         z0=simplify(vpa(beta1 + l1 + d3.*cos(q2),3));
12         x0=vpa(x0); y0=vpa(y0); z0=vpa(z0);
13         else %caso num´erico
14         x0=-beta2*sin(q1)+d3.*cos(q1).*sin(q2);
15         y0= beta2*cos(q1) + d3.*sin(q1).*sin(q2);
16         z0=beta1 + l1 + d3.*cos(q2);
17     end
18 end
```

donde l_1 es la distancia sobre el eje z_0 donde está ubicada la articulación de la base; q_1, q_2, d_2, d_3 son las posiciones articulares de la base, hombro y codo, respectivamente. Retorna las coordenadas cartesianas (x_0, y_0, z_0) en el sistema $\Sigma_0(x_0, y_0, z_0)$.

Cinemática inversa

Para obtener la cinemática inversa de la configuración esférica se emplea el método geométrico que se presenta a detalle en la figura 5.17. Los ángulos $\vartheta + q_1$ satisfacen $\tan(\vartheta + q_1) = \frac{y_0}{x_0}$. Note que el ángulo ϑ queda ubicado en el interior del triángulo formado por el cateto adyacente $\sqrt{x_0^2 + y_0^2 - \beta_2}$, cateto opuesto β_2 y la hipotenusa $\sqrt{x_0^2 + y_0^2}$, entonces $\tan(\vartheta) = \frac{\beta_2}{\sqrt{x_0^2 + y_0^2 - \beta_2^2}}$. Ahora, tomando en cuenta la siguiente identidad trigonométrica $\tan(\vartheta + q_1) = \frac{\tan(\vartheta) + \tan(q_1)}{1 - \tan(\vartheta)\tan(q_1)}$, se obtiene la expresión $\tan(q_1) = \frac{-x_0\beta_2 + y_0\sqrt{x_0^2 + y_0^2 - \beta_2^2}}{y_0\beta_2 + x_0\sqrt{x_0^2 + y_0^2 - \beta_2^2}}$. Por otro lado se tiene que $\tan(q_2 - \frac{\pi}{2}) = \frac{x_0 + y_0 - \beta_2}{z_0 - (l_1 + \beta_1)}$ y para movimiento de la articulación d_3 sobre el plano $x_0 - y_0$ es $d_3\cos(q_2 - \frac{\pi}{2})$ del $d_3\,\mathrm{sen}(q_2) = \sqrt{x_0^2 + y_0^2 - \beta_2^2}$. Ahora, tomando en cuenta la siguiente identidad trigonométrica $\tan(\vartheta + q_1) = \frac{\tan(\vartheta) + \tan(q_1)}{1 - \tan(\vartheta)\tan(q_1)}$, se obtiene la expresión $\tan(q_1) = \frac{-x_0\beta_2 + y_0\sqrt{x_0^2 + y_0^2 - \beta_2^2}}{y_0\beta_2 + x_0\sqrt{x_0^2 + y_0^2 - \beta_2^2}}$. Por otro lado se tiene que $\tan(q_2 - \frac{\pi}{2}) = \frac{x_0 + y_0 - \beta_2}{z_0 - (l_1 + \beta_1)}$ y para $d_3 = \sqrt{x_0^2 + y_0^2 - \beta_2^2 + [z_0 - (l_1 + \beta_1)]^2}$.

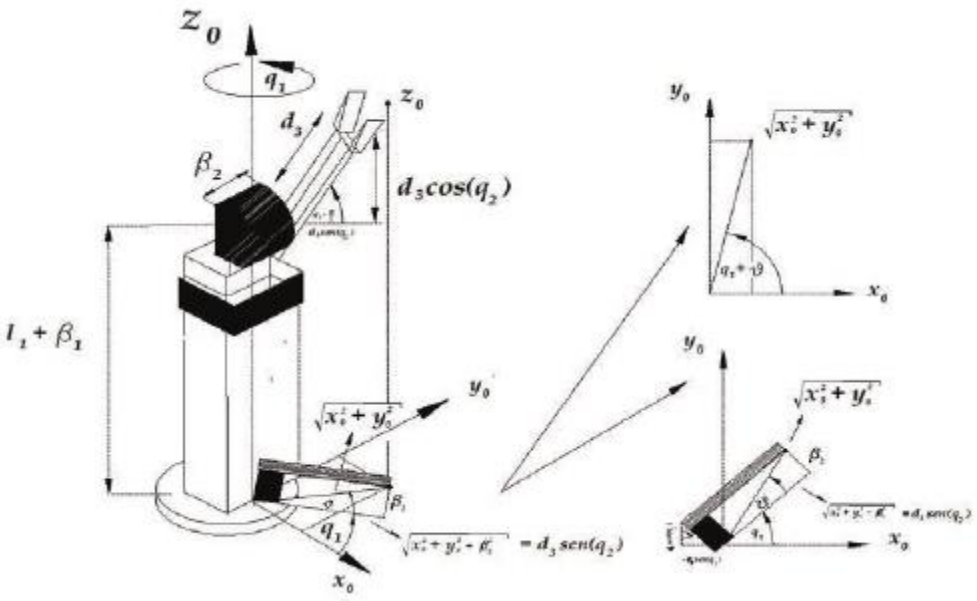


Figura 5.17 Cinemática inversa de la configuración esférica.

La cinemática inversa de la configuración esférica toma la siguiente forma:

$$q_1 = \operatorname{atan}\frac{-x_0\beta_2 + y_0\sqrt{x_{20} + y_0^2 - \beta_2^2}}{y_0\beta_2 + x_0\sqrt{x_{20} + y_0^2 - \beta_2^2}} \tag{5.50}$$

$$q_2 = \frac{\pi}{2} + \operatorname{atan}\frac{z_0 - (l_1 + \beta_1)}{\sqrt{x_{20} + y_0^2 - \beta_2^2}} \tag{5.51}$$

$$d_3 = \sqrt{x_{20} + y_0^2 - \beta_2^2 + [z_0 - (l_1 + \beta_1)]^2} \tag{5.52}$$

Funci´on cinem´atica inversa robot esf´erico

La sintaxis de la funci´on de cinem´atica inversa de la configuraci´on esf´erica es (ver c´odigo en el cuadro 5.20); esta funci´on retorna las coordenadas articulares.

```
[q1, q2, d3]=cinv_esferico(β1, β2, l1, x0, y0, z0)
```

C´odigo Fuente 5.20 cinv_esferico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinv_esferico.m

```
function [q1, q2, d3]=cinv_esferico(beta1,beta2,l1,x0,y0,z0)
    w1=y0.*sqrt(x0.*x0+y0.*y0-beta2*beta2)-x0*beta2;
    w2=x0.* sqrt(x0.*x0+y0.*y0-beta2*beta2)+y0*beta2;
    q1=atan(w1./w2);
    w3=z0-(l1+beta1);
    w4=sqrt(x0.*x0+y0.*y0-beta2*beta2);
    q2=pi/2+atan(w3./w4);
    d3=sqrt(x0.*x0+y0.*y0-beta2*beta2+(z0-(l1+beta1)).*(z0-(l1+beta1)));
end
```

♣♣♣ Ejemplo 5.5

Escribir un programa en lenguaje **MATLAB** que permita desplegar en forma simb´olica los par´ametros DH, matriz homog´enea, cinem´atica cartesiana, jacobiano y determinante del robot esf´erico. Adem´as, programar una aplicaci´on donde el extremo final del robot trace un conjunto de figuras circulares. El radio de cada c´ırculo es $r = 0.25$ m, con centro en el plano $(x_0, y_0) = (0.3, -0.3)$ m, cada c´ırculo estar´a desplazado un mil´ımetro sobre el eje z_0 (de 0 a 100 mm).

Soluci´on

El programa que permite resolver el problema planteado se encuentra en el cuadro 5.21. La informaci´on simb´olica del robot esf´erico se encuentra en las l´ıneas 4 a la 15 (par´ametros DH, matriz homog´enea, cinem´atica directa, jacobiano y determinante del robot cil´ındrico).

La figura circular se encuentra implementada en la l´ınea 18 a la 24, con la siguiente ecuaci´on:

$$\begin{aligned} x &= x_c + r\ \text{sen}(t) \\ y &= y_c + r\cos(t) \\ z &= kt \end{aligned}$$

donde r es el radio del c´ırculo, t es la evoluci´on del tiempo, k [mm/seg] es una constante positiva que convierte el tiempo a mil´ımetros, las coordenadas $[x_c, y_c]^T = [0.3, -0.3]^T$ m representan el centro del c´ırculo en el plano x_0, y_0 y el desplazamiento variable est´a dado por z. La l´ınea 25 convierte las coordenadas cartesianas del c´ırculo a coordenadas articulares del robot usando la funci´on `cinv_esferico`. La l´ınea 26 realiza la conversi´on de coordenadas articulares a coordenadas cartesianas del extremo final del robot (`cinematica_esferico`). La figura 5.18 muestra el trazo realizado por el extremo final del robot esf´erico (ver l´ınea 27).

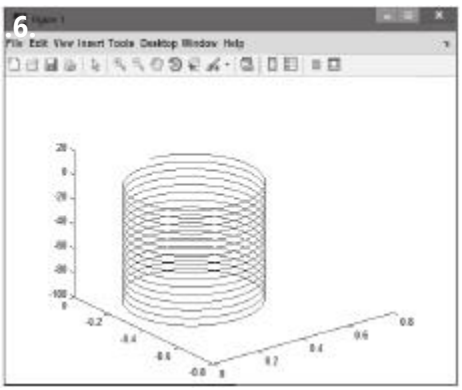


Figura 5.18 Aplicaci´on del robot esf´erico.

C´odigo Fuente 5.21 cap5_esferico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cap5_esferico.m

```
1  clc; clear all; close all;
2  format short
3  syms q1 q2 d3 beta1 beta2 l1 alpha1 alpha2 alpha3 real
4  H30=H_esferico();
5  disp('Transformaci´on homog´enea del robot esf´erico');
6  disp(H30);
7  [R30, cinemat_esferico,cero, c]=H_DH(H30);
8  disp('Matriz de rotaci´on');
9  disp(R30);
10 disp('cinem´atica directa');
11 disp(cinemat_esferico);
12 [x0, y0,z0]=cinematica_esferico(beta1,beta2,l1,q1,q2,d3);
13 disp([x0; y0;z0])
14 jac_esferico=jacobian([x0; y0;z0], [q1;q2;d3])
15 det_esferico=simplify(vpa(det(jac_esferico),3))
16 %ejemplo num´erico
17 l1=0.5; beta1=0.10; beta2=0.05;
18 t=0:0.001:100; [n m]=size(t);
19 %ecuaci´on del c´ırculo con centro en xc,yc y radio r
20 xc=0.3; yc=-0.3; r=0.25;
21 q1=[]; q2=[]; d3=[]; k=1;
22 x=xc+r.*sin(t);
23 y=yc+r.*cos(t);
24 z=k*t;
25 [q1, q2, d3]=cinv_esferico(beta1,beta2,l1,x,y,z);
26 [x0, y0,z0]=cinematica_esferico(beta1,beta2,l1,q1,q2,d3);
27 plot3(x0,y0,z0)
```

5.5 Manipulador cil´ındrico (RPP)

La **configuraci´on cil´ındrica** tiene la articulaci´on de la base rotacional, mientras que las articulaciones del hombro y codo son lineales o prism´aticas. Entre las aplicaciones m´as importantes de los robots manipuladores en esta configuraci´on se encuentran procesos para desbastar moldes, traslado de objetos, ensamble de piezas en espacios horizontales y transporte de objetos. En la figura 5.19 se muestra la configuraci´on cil´ındrica.

Figura 5.19 Robot en configuraci´on cil´ındrica.

Cinem´atica directa cartesiana

El espacio de trabajo de la configuraci´on cil´ındrica que se muestra en la figura 5.21 tiene la forma de un cilindro hueco con radio directamente proporcional al desplazamiento lineal d_3 y altura variable d_2. El origen del sistema de referencia cartesiano fijo $\Sigma_0(x_0, y_0, z_0)$ se ubica convenientemente en el piso, de tal forma que la articulaci´on rotacional de la base se encuentra a una distancia l_1 sobre el eje z_0. Este eje se alinea con el eje de rotaci´on del servomotor, de esta forma la variable articular q_1 gira alrededor del eje z_0. El ancho del servomotor y espesor de las placas met´alicas que se utilizan para acoplar mec´anicamente la siguiente articulaci´on se encuentra expresado por el par´ametro geom´etrico β_1. Note que el par´ametro l_1 produce s´olo coordenadas positivas para z_0, cuando este par´ametro es cero o el origen de $\Sigma_0(x_0, y_0, z_0)$ se coloca sobre la articulaci´on de la base, entonces puede haber coordenadas negativas en z_0.

El sistema $\Sigma_1(x_1, y_1, z_1)$ sirve para medir la coordenada lineal d_2 de la articulaci´on

del hombro, las coordenadas de $\Sigma_1(x_1, y_1, z_1)$ con respecto al sistema $\Sigma_0(x_0, y_0, z_0)$ son: $[0, 0, l_1 + \beta_1]^T$. El eje z_1 se alinea con el desplazamiento lineal de la variable d_2 y los ejes z_0 y z_1 son paralelos entre s´ı. El sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ se encuentra rotado un ángulo q_1 con respecto al sistema fijo $\Sigma_0(x_0, y_0, z_0)$, la matriz de rotaci´on que relaciona dicha rotaci´on entre estos dos sistemas de referencia es: $R_{10} = R_{z0}(q_1)$.

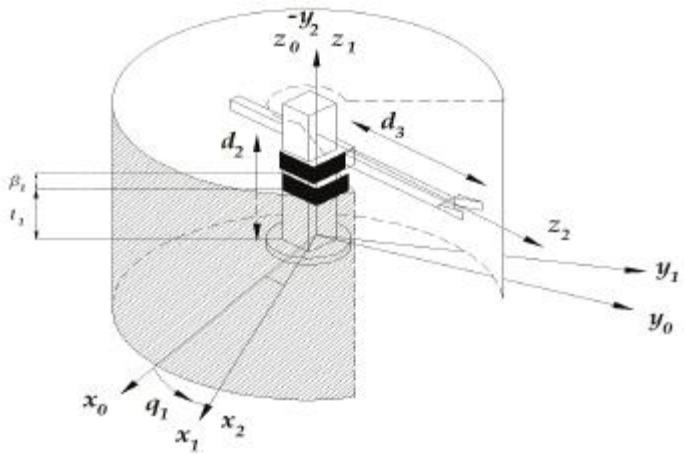


Figura 5.20 Espacio de trabajo del robot cil´ındrico.

La articulaci´on del codo tiene asociado el sistema de referencia $\Sigma_2(x_2, y_2, z_2)$ para medir el desplazamiento lineal de la variable d_3. Este sistema se genera por rotar un angulo $-\frac{\pi}{2}$ radianes alrededor del eje x_1, de tal forma que el eje z_2 determina el desplazamiento lineal d_2. Note que una consecuencia de esta rotaci´on alrededor del eje x_1, es que el eje z_2 queda alineado sobre el eje y_1, de tal forma que para $q_1 = 0$, el desplazamiento lineal de d_3 coincide sobre el eje y_0. Debido a esto, para la articulaci´on lineal del codo el ´angulo alrededor del eje z_0 es $q_1 + \frac{\pi}{2}$Observe tambi´en la direcci´on para coordenadas negativas del eje y_2 coincide con la direcci´on positiva de los ejes z_0 y z_1.

Las coordenadas del origen del sistema $\Sigma_2(x_2, y_2, z_2)$ con respecto al sistema $\Sigma_1(x_1, y_1, z_1)$ son: $[0, 0, l_1 + \beta_1]^T$. El eje z_2 es perpendicular a los ejes z_0 y z_1. La forma en que est´an indicados los sistemas de referencia $\Sigma_0(x_0, y_0, z_0)$, $\Sigma_1(x_1, y_1, z_1)$ y $\Sigma_2(x_2, y_2, z_2)$ en la figura 5.21 determinan la posici´on de casa del brazo robot en configuraci´on cil´ındrica, es decir para esa configuraci´on de sistemas de referencia las

variables articulares tienen un valor de $[q_1, d_2, d_3]^T = [0, 0, l_1 + \beta_1]^T$. La variable lineal de codo normalmente es calibrada para dar valor cero en $z_0 = l_1 + \beta_1$, desplazamientos mayores a esta longitud son positivos, o en caso contrario negativos. Para otro tipo de posici´on de casa, la cinem´atica directa del robot cil´ındrico adquiere una forma cartesiana diferente.

Los par´ametros geom´etricos de espesor y ancho β_2, β_3 de los servomotores de las articulaciones del hombro y codo son absorbidos como parte de las caracter´ısticas mec´anicas de sus articulaciones, por lo que no influyen directamente en la expresi´on anal´ıtica de la cinem´atica directa del brazo robot en configuraci´on cil´ındrica.

En la tabla 5.6 se muestran los par´ametros del robot manipulador en la configuraci´on cil´ındrica.

Tabla 5.6 DH del robot cil´ındrico

Eslab´on	l_i	α_i	d_i	θ_i
1	0	0	$l_1 + \beta_1$	q_1
2	0	$-\frac{\pi}{2}$	d_2	0
3	0	0	d_3	0

Con la tabla 5.6 el robot en la configuraci´on cil´ındrica tiene las siguientes matrices de transformaci´on homog´enea:

$$H_0^1 = H_{Rz_0}(q_1)\,H_{Tz_0}(\beta_1 + l_1)\,H_{Tx_0}(0)\,H_{Rx_0}(0)$$

$$= \begin{bmatrix} \cos(q_1) & -\operatorname{sen}(q_1) & 0 & 0 \\ \operatorname{sen}(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & l_1 + \beta_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.53}$$

$$H_1^2 = H_{Rz_0}(0)\,H_{Tz_0}(d_2)\,H_{Tx_0}(0)\,H_{Rx_0}(-\pi/2)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.54}$$

$$H_2^3 = H_{Rz_0}(0)\,H_{Tz_0}(d_3)\,H_{Tx_0}(0)\,H_{Rx_0}(0)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.55}$$

$$H_0^3 = H_0^1 H_1^2 H_2^3 = \begin{bmatrix} \cos(q_1) & 0 & -\operatorname{sen}(q_1) & -\operatorname{sen}(q_1) d_3 \\ \operatorname{sen}(q_1) & 0 & \cos(q_1) & d_3 \cos(q_1) \\ 0 & -1 & 0 & l_1 + \beta_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.56}$$

El modelo de cinem´atica directa para el robot manipulador cil´ındrico est´a dado de la siguiente forma:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{f}_R(\mathbf{q}) = \begin{bmatrix} -d_3 \operatorname{sen}(q_1) \\ d_3 \cos(q_1) \\ l_1 + \beta_1 + d_2 \end{bmatrix} \tag{5.57}$$

Jacobiano del robot cil´ındrico

El jacobiano del brazo robot en configuraci´on cil´ındrica se expresa de la siguiente forma:

$$J(q_1, d_2, d_3) = \frac{\partial \mathbf{f}_R(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} -\cos(q_1) d_3 & 0 & -\operatorname{sen}(q_1) \\ -d_3 \operatorname{sen}(q_1) & 0 & \cos(q_1) \\ 0 & 1 & 0 \end{bmatrix} \tag{5.58}$$

cuyo determinante est´a dado por:

$$\det[J(q_1, d_2, d_3)] = d_3 \tag{5.59}$$

la única singularidad que presenta el brazo robot en configuraci´on cil´ındrica es cuando la articulaci´on lineal del codo es $d_3 = 0$, mientras que las variables $q_1, d_2 \in \mathbb{R}$ pueden tener cualquier valor. De ah´ı que es recomendable para algunas aplicaciones donde se involucra la inversa de la matriz jacobiana que la variable articular lineal $d_3 = 0$. En algunos robots este problema se evita insertando sensores, interruptores o por programaci´on para deshabilitar el movimiento del robot cuando la articulaci´on lineal d_3 pasa por el origen del sistema $\Sigma_0(x_0, y_0, z_0)$.

Funci´on transformaci´on homog´enea robot cil´ındrico

La funci´on matriz de transformaci´on homog´enea del brazo robot en configuraci´on cil´ındrica tiene la siguiente sintaxis:

```
H03= H_cilindrico()
```

retorna la matriz de transformaci´on homog´enea H_0^3 compuesta por la matriz de rotaci´on R_{30} y la cinem´atica directa cartesiana $\mathbf{f}_R(\mathbf{q})$ que relaciona las coordenadas cartesianas del extremo final con las coordenadas articulares. Tambi´en despliega los par´ametros Denavit-Hartenberg en forma simb´olica que se muestran en la tabla 5.6. El programa 5.22 contiene el c´odigo de la funci´on transformaci´on homog´enea.

C´odigo Fuente 5.22 H_cilindrico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

H_cilindrico.m

```
1  function H=H_cilindrico()
2      syms q1 d2 d3 beta1 l1 alpha1 alpha2 alpha3 real
3      disp('Par´ametros Denavit-Hartenberg del robot cil´ındrico');
4      disp('[ l alpha d q]');
5      dh=[0, 0, l1+beta1, q1; 0, -pi/2, d2, 0; 0, 0, d3, 0];
6      disp(dh)
7      %H_0^1 = HRz0(q1)HTz0(l1 + β1)HTx0(0)HRx0(0)
8      H10=HRz(q1)*HTz(l1+beta1)*HTx(0)*HRx(0);
9      %H_1^2 = HRz1(0)HTz1(d2)HTx1(0)HRx1(-π/2)
10     H21=HRz(0)*HTz(d2)*HTx(0)*HRx(-pi/2);
11     %H_2^3 = HRz2(0)HTz2(d3)HTx2(0)HRx2(0)
12     H32=HRz(0)*HTz(d3)*HTx(0)*HRx(0);
13     H30=simplify(H10*H21*H32); % H_0^3 = H_0^1 H_1^2 H_2^3
14     [R30, cinemat_cilindrico, cero, c]=H_DH(H30);
15     H=[R30, cinemat_cilindrico;
16         cero, c];
17 end
```

Funci´on cinem´atica directa del robot cil´ındrico

La funci´on de cinem´atica directa del brazo robot cil´ındrico est´a por:

$$[x_0, y_0, z_0] = \text{cinematica\_cilindrico}(\beta_1, l_1, q_1, d_2, d_3)$$

donde q_1, d_2, d_3 son las posiciones articulares de la base, hombro y codo, respectivamente. Retorna las coordenadas cartesianas (x_0, y_0, z_0) en el sistema $\Sigma_0(x_0, y_0, z_0)$. El cuadro 5.23 contiene el c´odigo en lenguaje **MATLAB** para la funci´on cinem´atica del robot cil´ındrico:

C´odigo Fuente 5.23 cinematica_cilindrico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinematica_cilindrico.m

```
1  function [x0, y0,z0]=cinematica_cilindrico(beta1,l1,q1,d2,d3)
2      dato1=whos('beta1'); dato2=whos('l1'); dato3=whos('q1');
3      dato4=whos('d2'); dato5=whos('d3'); v1=strcmp(dato1.class, 'sym');
4      v2=strcmp(dato2.class, 'sym'); v3=strcmp(dato3.class, 'sym');
5      v4=strcmp(dato4.class, 'sym'); v5=strcmp(dato5.class, 'sym'); digits(3);
6      if( v1 & v2 & v3 & v4 & v5)% caso simb´olico
7          x0= -d3*sin(q1);
8          y0= d3*cos(q1);
9          z0=l1+beta1+d2;
10         x0=vpa(x0); y0=vpa(y0); z0=vpa(z0);
11         else %caso num´erico
12         x0= -d3.*sin(q1);
13         y0= d3.*cos(q1);
14         z0=l1+beta1+d2;
15     end
16 end
```

Cinem´atica inversa

La cinem´atica inversa del brazo robot en configuraci´on cil´ındrica se obtiene de la cinem´atica directa (5.57) de la siguiente forma: primero que nada, debe considerarse que debido a que la posici´on de casa $[q_1, d_2, d_3]^T = [0, 0, 0]^T$ seleccionada para este brazo robot se caracteriza por la articulaci´on del codo d_3 est´a alineada sobre el eje y_0 (los eje y_0 y z_2 coinciden cuando $q_1 = 0$), entonces el desplazamiento rotacional de la base alrededor del eje z_0 es por un angulo $q_1 + \pi_2$ (ver figura 5.21). Por lo tanto, se cumple: $q_1 + \frac{\pi}{2} = \text{atan} -\frac{x}{y_0}$.

La articulaci´on del hombro tiene un desplazamiento lineal sobre el eje z_0, entonces $d_2 = z_0 - (l_1 + \beta_1)$. La articulaci´on del codo d_3 proyecta su movimiento sobre el plano $x_0 - y_0$, por lo que puede tomar dos posibles soluciones: $d_3 = \pm \sqrt{x_{20} + y_0^2}$.

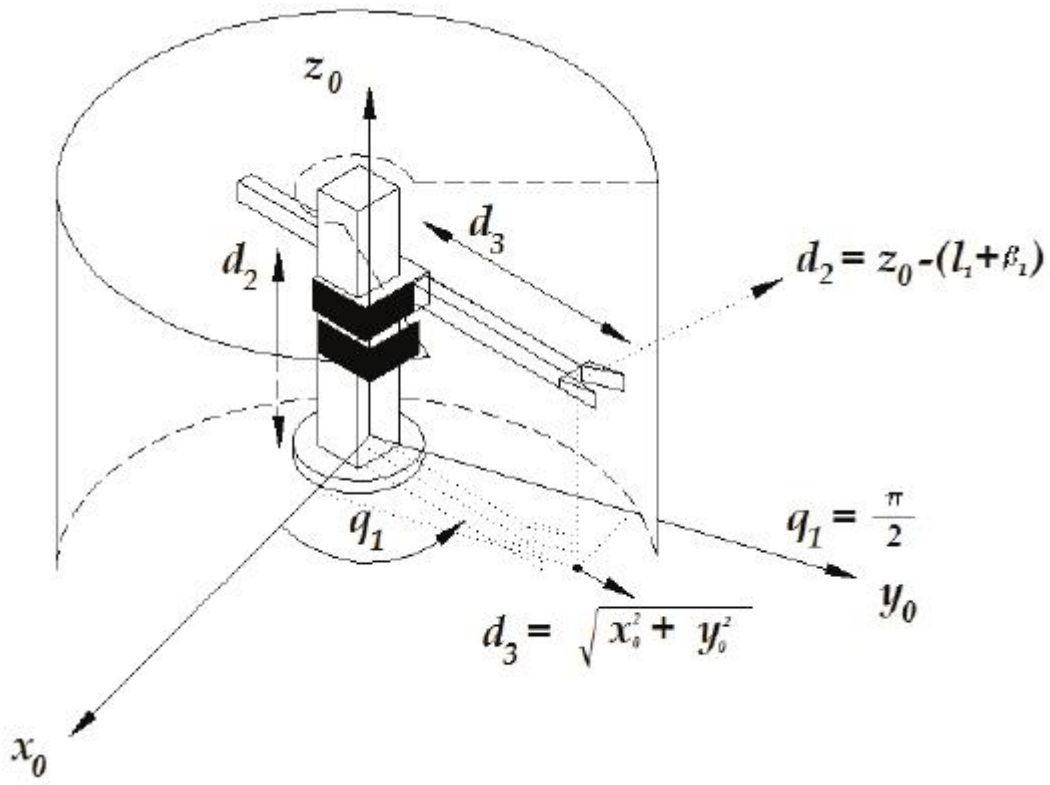


Figura 5.21 Cinem´atica inversa del robot cil´ındrico.

La cinem´atica inversa del brazo robot en configuraci´on cil´ındrica est´a dada como:

$$q_1 = -\frac{\pi}{} \qquad \frac{x_0}{y_0} \tag{5.60}$$

$$d_2 = z_0 - (l_1 + \beta_1) \tag{5.61}$$

$$d_3 = \pm \sqrt{x_0^2 + y_0^2} \tag{5.62}$$

Funci´on cinem´atica inversa robot cil´ındrico

La funci´on de cinem´atica inversa del robot cil´ındrico convierte las coordenadas cartesianas del extremo final (x_0, y_0, z_0) en funci´on de las coordenadas articulares (q_1, d_2, d_3) con la siguiente sintaxis:

```
[q1, d2, d3]=cinv_cilindrico(beta1, l1, x0, y0, z0)
```

donde β_1, l_1 son los par´ametros geom´etricos, x_0, y_0, z_0 son las coordenadas cartesianas del extremo final. Esta funci´on retorna las coordenadas articulares q_1, d_2, d_3 de la base, hombro y codo, respectivamente. El cuadro 5.24 contiene la implementaci´on de la funci´on cinem´atica inversa del brazo robot cil´ındrico.

C´odigo Fuente 5.24 cinv_cilindrico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinv_cilindrico.m

```
1 function [q1, d2,d3]=cinv_cilindrico(beta1,l1,x0,y0,z0)
2     q1= -pi/2+atan(-x0./y0);
3     d2=z0-(l1+beta1);
4     d3= sqrt(x0.*x0+y0.*y0);
5 end
```

♣♣♣ Ejemplo 5.6

Escribir un programa en lenguaje **MATLAB** que permita desplegar en forma simbólica los parámetros DH, cinemática cartesiana, jacobiano y determinante del robot cilíndrico. Asimismo programar una aplicación donde el extremo final del robot realice el trazo de una rosa polar con centro en $[0.3, -0.3, 0.5]$ y radio $r = 0.15$.

Solución

El desplegado simbólico de los parámetros DH, cinemática directa y jacobiano del robot cilíndrico se encuentra en el cuadro 5.25. En la línea 4 se obtiene la matriz de transformación homogénea H_0^3 utilizando la función `H_cilindrico()`. La matriz de transformación homogénea H_0^3 se utiliza como argumento de entrada en la función `D_HD(H_0^3)` para obtener la matriz de rotación R_{30} y la cinemática directa (línea 5). El jacobiano y su determinante se despliegan en las líneas 11 y 12, respectivamente.

La figura "rosa polar" tiene la siguiente ecuación:

$$\begin{aligned} r &= 0.05 + 0.1\,\text{sen}(t) \\ x &= x_c + r\,\text{sen}(t) \\ y &= y_c + r\cos(t) \\ z &= l_1 + \beta_1 + 0.05 \end{aligned}$$

donde las coordendas $[x_c, y_c, z]^T$ son el centro de la figura (ver líneas 19-22), $l_1 + \beta_1 = 0.45$ El vector tiempo es de 0 a 100 segundos, con incrementos de un milisegundo. Las coordenadas cartesianas de la rosa polar son convertidas a coordenadas articulares usando la función `cinv_cilindrico(β1, l1, x, y, z)` (ver línea 24). La línea 26 contiene la conversión de coordenadas cartesianas del extremo final del robot a coordenadas articulares usando la función `cinematica_cilindrico(q1, d2, d3)`. La línea 27 exhibe la figura 5.22 trazada por el robot cilíndrico.

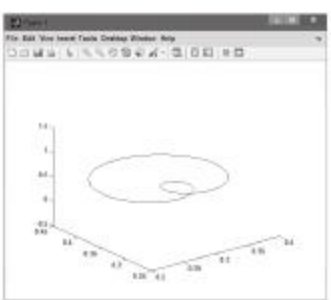


Figura 5.22 Aplicación del robot cilíndrico.

C´odigo Fuente 5.25 cap5_cil´ındrico.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cap5_cil´ındrico.m

```
1  clc; clear all; close all;
2  format short
3  syms q1 d2 d3 beta1 l1 alpha1 alpha2 alpha3 real
4  H30=H_cilindrico(); disp('Transformaci´on homog´enea del robot cil´ındrico');
5  [R30, cinemat_cilindrico,cero, c]=H_DH(H30);
6  disp('Matriz de rotaci´on');
7  disp(R30);
8  disp('cinem´atica directa'); disp(cinemat_cilindrico);
9  [x0, y0,z0]=cinematica_cilindrico(beta1,l1,q1,d2,d3);
10 disp([x0; y0;z0])
11 jac_cilindrico=jacobian([x0;y0;z0], [q1;d2;d3])
12 det_cilindrico=simplify(vpa(det(jac_cilindrico),3))
13 %ejemplo num´erico
14 t=0:0.001:100;
15 %ecuaci´on de la rosa polar con centro en (xc,yc) y radio r
16 xc=0.3; yc=-0.3; beta1=0.05; l1=0.40;
17 q1=[]; d2=[]; d3=[]; r=0.2;
18 [n m]=size(t);
19 r=0.05+0.1*sin(t);
20 x=xc+r.*sin(t);
21 y=yc+r.*cos(t);
22 z(1:m)=l1+beta1+0.05;
23 %conversi´on de coordenadas cartesianas a coordenadas articulares del robot
24 [q1, d2, d3]=cinv_cilindrico(beta1,l1,x,y,z);
25 %coordenadas cartesianas del extremo final del robot
26 [x0, y0,z0]=cinematica_cilindrico(beta1,l1,q1,d2,d3);
27 plot3(x0,y0,z0)%trayectoria trazada por el robot
```

5.6 Configuraci´on cartesiana (PPP)

Los robots manipuladores que incluyen sus tres primeras articulaciones del tipo prism´atico o lineales se denominan **robots cartesianos**, tambi´en conocidos como robots p´orticos o lineales, por lo que no tienen articulaciones rotacionales. Existen varias aplicaciones de los robots cartesianos, entre ellas: corte de mascarillas, graficadores o plotters, taladros autom´aticos, mesas de medici´on de coordenadas como la que se muestra la figura 5.23, impresoras l´aser o de matriz son otros ejemplos de robots cartesianos con 2 gdl.

Figura 5.23 Robot cartesiano.

Particularmente, la configuraci´on cartesiana facilita la programaci´on del extremo final del robot y sus potenciales aplicaciones, ya que la cinem´atica directa es un mapa lineal entre coordenadas articulares y cartesianas. Sin embargo, la estructura mec´anica del robot cartesiano presenta baja destreza de movilidad comparado con la configuraci´on antropom´orfica debido a que todas sus articulaciones son prism´aticas.

Cinem´atica cartesiana

En la figura 5.24 se muestra el espacio de trabajo del robot cartesiano, tiene forma de un paralelep´ıpedo recto. El sistema de referencia cartesiano fijo $\Sigma_0(x_0, y_0, z_0)$ se selecciona de manera conveniente. El eje z_0 determina el desplazamiento lineal de la primera articulaci´on prism´atica d_1, el origen del sistema $\Sigma_0(x_0, y_0, z_0)$ se ubica en $[0, 0, l_1]$ siendo l_1 una longitud con respecto al nivel del piso, l_1 representa la longitud de las barras del robot cartesiano. Los ejes x_0 y y_0 quedan alineados por la

regla de la mano derecha. El siguiente sistema de referencia $\Sigma_1(x_1, y_1, z_1)$ se obtiene rotando el eje x_0 un angulo $\alpha_1 = -\frac{\pi}{2}$ quedando el eje z_1 en direcci´on positiva de la variable d_2. Observe que el eje y_1 apunta en direcci´on negativa del eje z_0 y el eje x_1 es paralelo al eje x_0, separado evidentemente por la dimensi´on f´ısica de la mesa cartesiana denotada por la longitud l_2. El sistema $\Sigma_2(x_2, y_2, z_2)$ sirve para medir el desplazamiento lineal de la variable articular d_3. Sin embargo, la obtenci´on de este tercer sistema de referencia no es trivial, ya que no se puede deducir directamente del sistema $\Sigma_1(x_1, y_1, z_1)$ debido a que no existe un ´angulo α_2 alrededor del eje x_1 que genere directamente el sistema de referencia $\Sigma_2(x_2, y_2, z_2)$.

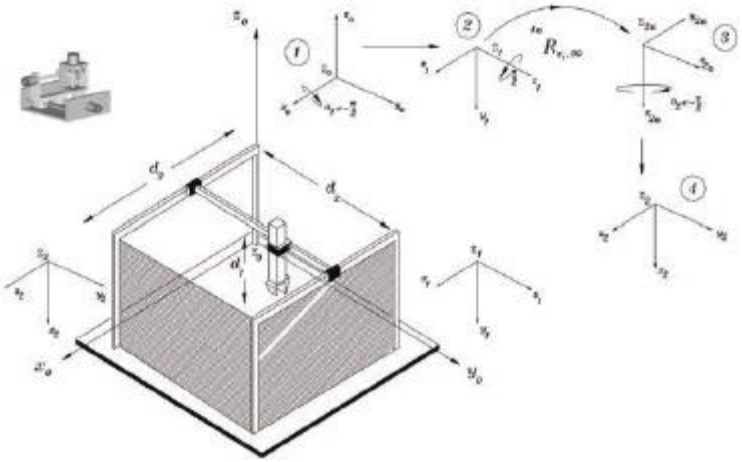


Figura 5.24 Espacio de trabajo del robot cartesiano.

Para obtener el sistema de referencia $\Sigma_2(x_2, y_2, z_2)$, primero se emplea un sistema de referencia auxiliar denominado $\Sigma_{2a}(x_{2a}, y_{2a}, z_{2a})$ el cual se obtiene realizando una rotaci´on de 90 grados con respecto al eje z_1, esta rotaci´on hace que el eje x_{2a} permanezca paralelo al eje y_1. Debe observarse que los ejes z_1 y z_{2a} son paralelos entre s´ı de tal manera que sus coordenadas (d_2) entre estos ejes sean id´enticas. El origen del sistema de referencia auxiliar $\Sigma_{2a}(x_{2a}, y_{2a}, z_{2a})$ es el mismo que el sistema de referencia $\Sigma_1(x_1, y_1, z_1)$, pero el sistema $\Sigma_{2a}(x_{2a}, y_{2a}, z_{2a})$ mantiene una rotaci´on relativa al sistema $\Sigma_1(x_1, y_1, z_1)$ expresada por la matriz rotaci´on $R_{z1}(\frac{\pi}{2})$, es decir:

$$\begin{bmatrix} x_{2a} \\ y_{2a} \\ z_{2a} \end{bmatrix} = R_{z1}^T\left(\frac{\pi}{2}\right) \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

Posteriormente, girando un angulo $\alpha_2 = \frac{\pi}{2}$ alrededor del eje x_{2a} se obtiene el sistema

de referencia $\Sigma_2(x_2, y_2, z_2)$ donde la articulaci´on prism´atica d_3 se mueve linealmente sobre el eje z_2. Esta última rotaci´on α_2 alrededor del eje x_2 determina el ´angulo entre los ejes z_2 y z_1. Los ejes z_0, z_1 y z_2 son mutuamente perpendiculares entre s´ı.

En la figura 5.25 se muestra la secuencia de rotaciones. El paso 1 corresponde al sistema fijo $\Sigma_0(x_0, y_0, z_0)$, rotando un angulo $\alpha_1 = -\frac{\pi}{2}$ alrededor del eje x_0 genera el sistema $\Sigma_1(x_1, y_1, z_1)$. El paso 2 consiste en rotar por un angulo $\frac{\pi}{2}$ alrededor del eje z_1 para generar el sistema de referencia $\Sigma_{2a}(x_{2a}, y_{2a}, z_{2a})$, este sistema es auxiliar y sirve para generar el sistema $\Sigma_2(x_2, y_2, z_2)$. Para eso el paso 3 consiste en rotar un ´angulo $\alpha_2 = -\frac{\pi}{2}$ alrededor del eje x_{2a}.

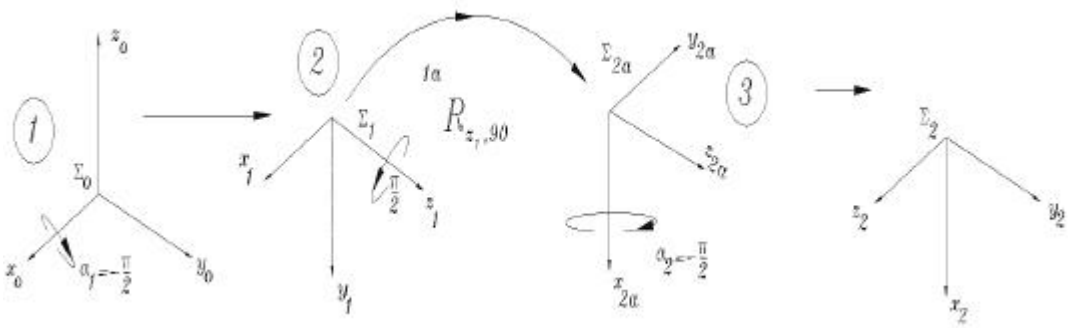


Figura 5.25 Sistemas de referencia del robot cartesiano.

La tabla de par´ametros del robot manipulador de 3 gdl en configuraci´on cartesiana queda de la siguiente forma:

Tabla 5.7 Par´ametros Denavit-Hartenberg para el robot cartesiano de 3 gdl

Eslab´on	l_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	d_1	0
2	0	$-\frac{\pi}{2}$	d_2	0
3	0	0	d_3	0

La matriz de transformación homogénea para la primera articulación toma la siguiente forma:

$$H_0^1 = H_{Rz_0}(0)\,H_{Tz_0}(d_1)\,H_{Tx_0}(0)\,H_{Rx_0}\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.63}$$

$$H_1^2 = H_{Rz_1}(0)\,H_{Tz_1}(d_2)\,H_{Tx_1}(0)\,H_{Tx_1}\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.64}$$

$$H_{2a} = H_{Rz_1}\left(\frac{\pi}{2}\right)^T H_1 = H_{Rz_1}\left(\frac{\pi}{2}\right)^T \underbrace{H_{Rz_1}(0)\,H_{Tz_2}(d_2)\,H_{Tx_1}(0)\,H_{Tx_1}\left(-\frac{\pi}{2}\right)}_{H_1^2}$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.65}$$

$$H_2^3 = H_{Rz_2}(0)\,H_{Tz_2}(d_3)\,H_{Tx_2}(0)\,H_{Tx_2}(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.66}$$

$$H_0^3 = H_0\,\underbrace{R_{Tz,\frac{\pi}{2}}\,H_1}_{H_{2a}}\,H_2 = \begin{bmatrix} 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La cinemática directa del robot cartesiano está dado por:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \end{bmatrix} \tag{5.67}$$

Jacobiano del robot cartesiano

La matriz jacobiano del robot cartesiano est´a dado por:

$$\mathbf{J}(d_3, d_2, d_1) = \frac{\partial \mathbf{f}_R(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.68}$$

cuyo determinante es unitario $\det[\mathbf{J}(d_3, d_2, d_1)] = 1$.

La configuraci´on cartesiana ofrece ventajas debido que no tiene singularidades.

Funci´on transformaci´on homog´enea robot cartesiano

La sintaxis de la funci´on transformaci´on homog´enea del brazo robot cartesiano es:

```
if    H_0^3=H_cartesiano()
```

esta funci´on retorna la matriz de transformaci´on homog´enea H_0^3 compuesta por la matriz de rotaci´on R_{30} que describe la rotaci´on del extremo final del robot con respecto al sistema fijo $\Sigma_0(x_0, y_0, z_0)$ y la cinem´atica directa cartesiana $\mathbf{f}_R(\mathbf{q})$ que relaciona las coordenadas articulares. Adem´as, tambi´en despliega los par´ametros Denavit-Hartenberg en forma simb´olica de la tabla 5.7.

El programa 5.26 contiene el c´odigo **MATLAB** de la funci´on transformaci´on homog´enea del robot cartesiano. Utiliza las matrices de transformaci´on homog´eneas de rotaci´on $H_{R_z}(q_i)$ y traslaci´on $H_{T_z}(d)$ para el c´alculo de cada una de las matrices homog´eneas de las articulaciones: H_0^1, H_1^2, H_2^3. Tambi´en emplea la funci´on H_DH(H30) para obtener la matriz de rotaci´on R_{30} y la cinem´atica directa cartesiana $\mathbf{f}_R(d_1, d_2, d_3)$.

C´odigo Fuente 5.26 H_cartesiano.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

H_cartesiano.m

```
function H=H_cartesiano()
    syms l1 l2 l3 d1 d2 d3 alpha1 alpha2 alpha3 real
    disp('Par´ametros Denavit-Hartenberg del robot cartesiano')
    disp('[ l alpha d q]')
    dh=[0, -pi/2, d1, 0; 0, -pi/2, d2, 0; 0, 0, d3, 0];
    %despliega tabla de par´ametros DH
    disp(dh)
    %c´alculo de las matrices de transformaci´on homog´enea de cada articulaci´on
    %H_0^1 = H_Rz0(0) H_Tz0(d1) H_Tx0(0) H_Rx0(-pi/2)
    H10=HRz(0)*HTz(d1)*HTx(0)*HRx(-pi/2);
    %H_1^2 = H_Rz1(0) H_Tz1(d2) H_Tx1(0) H_Tx1(-pi/2)
    H21=HRz(0)*HTz(d2)*HTx(0)*HRx(-pi/2);
    %H_1^2a = H_Rz1(pi/2) H_1^2
    H21a=simplify((HRz(pi/2))'*H21);
    %H_2^3 = H_Rz2(0) H_Tz2(d3) H_Tx2(0) H_Tx2(0)
    H32=HRz(0)*HTz(d3)*HTx(0)*HRx(0);
    %H_0^3 = H_0^1 H_1a^2 H_2^3
    H30=simplify(H10*H21a*H32);
    %matriz de rotaci´on R_30 y cinem´atica directa f_R(d1, d2, d3)
    [R30, cinemat_cartesiano, cero, c]=H_DH(H30);
    %Forma la matriz de transformaci´on homog´enea del robot cartesiano
                    f_R(d1, d2, d3)
    θ^T
                    1
    H=[R30, cinemat_cartesiano;
       cero, c];
end
```

Funci´on cinem´atica directa del robot cartesiano

La sintaxis de la funci´on de cinem´atica directa de la configuraci´on cartesiana es:

```
[x0, y0, z0]=cinematica_cartesiano(d3, d2, d1)
```

donde d_3, d_2, 1_3 son las posiciones articulares del robot en configuraci´on cartesiana. Esta funci´on retorna las coordenadas cartesianas (x_0, y_0, z_0) en el sistema $\Sigma_0(x_0, y_0, z_0)$. El cuadro 5.27 contiene funci´on cinem´atica del robot cartesiano.

C´odigo Fuente 5.27 cinematica_cartesiano.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinematica_cartesiano.m

```
1  function [x0, y0, z0]=cinematica_cartesiano(d3,d2,d1)
2      dato1=whos('d1'); dato2=whos('d2'); dato3=whos('d3');
3      v1=strcmp(dato1.class, 'sym'); v2=strcmp(dato2.class, 'sym');
4      v3=strcmp(dato3.class, 'sym'); digits(3);
5      if ( v1 & v2 & v3) %caso simb´olico
6          x0=d3; y0=d2; z0=d1; x0=vpa(x0); y0=vpa(y0); z0=vpa(z0);
7          else %caso num´erico
8          x0=d3; y0=d2; z0=d1;
9      end
10 end
```

Funci´on cinem´atica inversa robot cartesiano

La funci´on de cinem´atica inversa del robot cartesiano convierte las coordenadas cartesianas del extremo final (x_0, y_0, z_0) en el sistema $\Sigma_0(x_0, y_0, z_0)$ a las coordenadas articulares (d_1, d_2, d_3), bajo la siguiente sintaxis:

```
[d3, d2, d1]=cinv_cartesiano(x0, y0, z0)
```

Esta funci´on retorna las coordenadas articulares. El cuadro 5.28 contiene el c´odigo de la cinem´atica inversa del robot cartesiano:

C´odigo Fuente 5.28 cinv_cartesiano.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.
```

cinv_cartesiano.m

```
1 function [d3, d2, d1]=cinv cartesiano(x0,y0,z0)
2 |   d3=x0; d2=y0; d1=z0;;
3 end
```

♣♣♣ Ejemplo 5.7

Escribir un programa en lenguaje **MATLAB** que permita desplegar en forma simb´olica los par´ametros DH, cinem´atica directa y el jacobiano del robot cartesiano. Adem´as, implementar una aplicaci´on donde el extremo final del robot trace una rosa polar, con centro en $[0.3, -0.3, 0.5]^T$ m y radio $r = 0.1$ m.

Soluci´on

El programa que se presenta en el cuadro 5.29 permite desplegar en forma simb´olica los par´ametros DH, cinem´atica directa, jacobiano del robot cartesiano y matriz de transformaci´on homog´enea H_0^3. De la l´ınea 19 a la 25 se obtienen las coordenadas cartesianas de la figura rosa polar, dichas coordenadas son trazadas por el extremo final del robot como se indica en la figura 5.26.

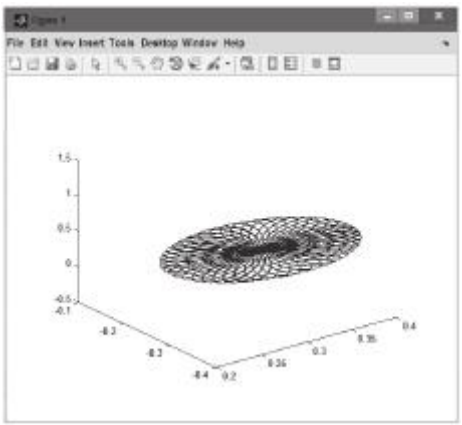


Figura 5.26 Rosa polar.

C´odigo Fuente 5.29 cap5_cartesiano.m

%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 5 Cinem´atica directa cartesiana.

cap5_cartesiano.m

```
1  clc;
2  clear all;
3  close all;
4  format short
5  syms d1 d2 d3 alpha1 alpha2 alpha3 real
6  H30=H_cartesiano();
7  disp('Transformaci´on homog´enea del robot cartesiano'); disp(H30);
8  [R30, cinemat_cartesiano,cero, c]=H_DH(H30);
9  disp('Matriz de rotaci´on'); disp(R30);
10 disp('cinem´atica directa');
11 disp(cinemat_cartesiano);
12 [x0, y0,z0]=cinematica_cartesiano(d3,d2,d1);
13 disp([x0; y0;z0])
14 jac_cartesiano=jacobian([x0;y0;z0], [d3;d2;d1])
15 det_cartesiano=simplify(vpa(det(jac_cartesiano),3))
16 %ejemplo num´erico
17 t=0:0.001:100;
18 %ecuaci´on de la flor con 8 p´etalos centro en xc,yc y radio r
19 xc=0.3; yc=-0.3 ;
20 d1=[]; d2=[]; d3=[]; r=0.2;
21 [n m]=size(t);
22 r=0.1*cos(pi*t);
23 x=xc+r.*sin(t);
24 y=yc+r.*cos(t);
25 z(1:m)=0.5 ;
26 [d3, d2, d1]=cinv_cartesiano(x,y,z);
27 [x0, y0,z0]=cinematica_cartesiano(d3,d2,d1);
28 plot3(x0,y0,z0)
```

5.7 Resumen

Cinemática directa cartesiana de robots manipuladores permite una descripci´on entre las coordenadas cartesianas del extremo final del robot manipulador y las coordenadas articulares. La convenci´on Denavit-Hartenberg es una herramienta de la ingenier´ıa de suma utilidad para la obtenci´on del modelo de cinem´atica directa de robots manipuladores la cual establece que la matriz de transformaci´on homog´enea de la ¡-´esima articulaci´on se compone de 4 par´ametros: l_i es la longitud de los eslabones, α_i el ´angulo que hay entre el eje z_{i-1} y z_i, d_i es el desplazamiento lineal sobre el eje z_{i-1}; cuando la ¡-´esima articulaci´on no es prism´atica o lineal, entonces d_i representa el ancho del servomotor y espesor de la placa met´alica, en este caso se especifica por β_i. Para articulaciones rotacionales alrededor del eje z_{i-1} se denota por q_i.

De acuerdo al an´alisis de cinem´atica directa cartesiana para cada configuraci´on de robots industriales (antropom´orfica, SCARA, esf´erico, cil´ındrico y cartesiano), se han desarrollado un conjunto de librer´ıas de cinem´atica directa de robots manipuladores para **MATLAB** (toolbox) que permiten obtener los par´ametros Denavit-Hartenberg, la matriz homog´enea, cinem´atica directa, matriz jacobiano y su determinante en funci´on de los par´ametros geom´etricos como longitudes l_i y tambi´en tomando en cuenta el ancho de la ¡-´esima articulaci´on y espesor de la placa met´alica (β_i) que se emplea para acoplar mec´anicamente la siguiente articulaci´on del robot.

Este conjunto de librer´ıas tienen la ventaja de trabajar en forma simb´olica, ya que despliegan no s´olo las ecuaciones de cinem´atica, tambi´en permiten realizar aplicaciones num´ericas como manejo de coordenadas con valores espec´ıficos o trayectorias de curvas para que el extremo final del robot pueda trazarlas.

La programaci´on indistinta en lenguaje **MATLAB** de variables simb´olicas y num´ericas resulta de gran utilidad sobre todo en el an´alisis cinem´atico, planeaci´on de trayectorias, guiado del robot en su espacio de trabajo y detecci´on de puntos singulares en la cinem´atica diferencial o en procesos donde involucra la matriz inversa del jacobiano del robot.

En el sitio WEB del libro se encuentran disponibles todos los programas fuentes en **MATLAB** de las funciones de cinemática cartesiana, matrices de transformación homogénea, cinemática inversa, jacobianos y determinantes, así como diversas aplicaciones de las configuraciones analizadas de robots manipuladores.

Para una rápida identificación de las librerías desarrolladas, a continuación se resume la sintaxis del conjunto de funciones para cinemática directa cartesiana con pase de parámetros de entrada y datos que retorna de cada las configuraciones de robots manipuladores analizados.

La tabla 5.8 contiene la descripción de funciones de cinemática para la configuración antropomórfica considerando los casos de estudio del péndulo robot, robot planar vertical de 2 gdl y robot con movimiento tridimensional en su espacio de trabajo (3 gdl).

Tabla 5.8 Configuración antropomórfico

Función	Sintaxis
Péndulo	H_0^1=H_pendulo() [x0, y0, z0]=cinematica_pendulo(β_1, l1, q1) q1=cinv_pendulo(x0, y0)
Robot antropomórfico vertical planar de 2 gdl	H_0^2=H_r2gdl() [x0, y0, z0]=cinematica_r2gdl(β_1, l1, q1, β_2, l2, q2) J(q1, q2)=jacobiano_r2gdl(β_1, l1, q1, β_2, l2, q2) det[J(q1, q2)]=detjac_r2gdl(β_1, l1, q1, β_2, l2, q2) [q1, q2]=cinv_r2gdl(l1, l2, x0, y0)
Robot antropomórfico de 3 gdl	H_0^3=H_r3gdl() [x0, y0, z0]=cinematica_r3gdl(β_1, l1, q1, β_2, l2, q2, β_3, l3, q3) J(q1, q2, q3)=jacobiano_r3gdl(β_1, l1, q1, β_2, l2, q2, q3) det[J(q1, q2, q3)]=detjac_r3gdl(β_1, l1, q1, β_2, l2, q2, q3) [q1, q2, q3]=cinv_r3gdl(β_2, β_3, l2, l3, x0, y0, z0)

En la tabla 5.9 se resume las funciones de transformaci´on homog´enea H_0^3, cinem´atica directa, cinem´atica inversa, matriz jacobiano y su determinante del brazo robot en configuraci´on SCARA.

Tabla 5.9 Configuraci´on SCARA

Funci´on	Sintaxis
Matriz de transformaci´on homog´enea	H_0^3=H_SCARA()
Cinem´atica directa	[x0, y0, z0]=cinematica_SCARA(β1, β2, l1, l2, l3, q1, q2, d3)
Jacobiano	J(q1, q2, d3)=jacobiano_SCARA(β1, l1, q1, β2, l2, q2)
Determinante del jacobiano	det[J(q1, q2, d3)]=detjac_SCARA(β1, l1, q1, β2, l2, q2)
Cinem´atica inversa	[d3, q2, q1]=cinv_SCARA(β1, β2, l1, l2, l3, x0, y0, z0)

El conjunto de librer´ıas para la configuraci´on esf´erica se presenta en la tabla 5.10.

Tabla 5.10 Configuraci´on esf´erica

Funci´on	Sintaxis
Matriz de transformaci´on homog´enea	H_0^3= H_esferico()
Cinem´atica directa	[x0, y0, z0]=cinematica_esferico(β1, β2, l1, q1, q2, d3)
Jacobiano	J(q1, q2, d3)=jacobiano_esferico(β1, l1, q1, β2, l2, q2)
Determinante del jacobiano	det[J(q1, q2, d3)]=detjac_esferico(β1, l1, q1, β2, l2, q2)
Cinem´atica inversa	[q1, q2, d3]=cinv_esferico(β1, β2, l1, x0, y0, z0)

La tabla 5.11 contiene las librer´ıas del brazo robot en configuraci´on cil´ındrica.

Tabla 5.11 Configuraci´on cil´ındrica

Funci´on	Sintaxis
Matriz de transformaci´on homog´enea	H_0^3= H_cilindrico()
Cinem´atica directa	[x0, y0, z0]=cinematica_cilindrico(β1, l1, q1, d2, d3)
Jacobiano	J(q1, q2, d3)=jacobiano_cilindrico(β1, l1, q1, β2, l2, q2)
Determinante del jacobiano	det[J(q1, q2, d3)]=detjac_cilindrico(β1, l1, q1, β2, l2, q2)
Cinem´atica inversa	[q1, d2, d3]=cinv_cilindrico(β1, l1, x0, y0, z0)

Para el caso del brazo robot cartesiano, las librer´ıas que corresponde a esta configuraci´on se encuentran en la tabla 5.12. La cinem´atica directa resulta muy simple debido a que su cinem´atica es lineal: $[x_0 \quad y_0 \quad z_0]^T = [d_1 \quad d_2 \quad d_3]^T$, por lo que el jacobiano es la matriz identidad y su determinante es unitario, es decir: det[J(d1, d2, d3)] = 1. La cinem´atica inversa tambi´en es lineal $[d_1 \quad d_2 \quad d_3]^T = [x_0 \quad y_0 \quad z_0]^T$.

Tabla 5.12 Configuraci´on cartesiana

Funci´on	Sintaxis
Matriz de transformaci´on homog´enea	H_0^3= H_cartesiano()
Cinem´atica directa	[x0, y0, z0]=cinematica_cartesiano(d3, d2, d1)
Cinem´atica inversa	[d3, d2, d1]=cinv_cartesiano(x0, y0, z0)

Parte II: Referencias selectas

Existe una extensa literatura sobre la cinem´atica directa de robots manipuladores. Particularmente se recomienda al lector la siguiente bibliograf´ıa:

J. Denavit & R.S. Hartenberg.
"A kinematic notation for lower-pair mechanisms based on matrices".
Trans ASME J. Appl. Mech, 23:215-221,1955

R.S. Hartenberg & J. Denavit. "Kinematic synthesis of linkages".
McGraw-Hill, New York, NY, 1964.

R.M. Murray, Z. Li and S.S. Sastry.
"A mathematical introduction to robotic manipulation". CRC Press (1994).

Lorenzo Sciavicco & Bruno Siciliano.
"Modeling and control for robot manipulators". McGraw Hill International Editions. 1996.

Mark W. Spong and Seth Hutchinson, M. Vidyasagar.
"Robot modeling and control". John Wiley and Sons, Inc. 2006.

John J. Craig. "Rob´otica". Tercera edici´on. Pearson Prentice-Hall. 2006.

Fernando Reyes. "Rob´otica: control de robots manipuladores ".
Editorial Alfaomega, 2011.

Parte II: Problemas propuestos

En esta secci´on se presenta una serie de ejercicios con la finalidad de que el lector mejore sus conocimientos sobre m´etodos num´ericos.

Cap´ıtulo 3 Preliminares matem´aticos

3.1 Considere los siguientes vectores:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \qquad y = \begin{bmatrix} 8 \\ 10.2 \\ 3.12 \end{bmatrix}$$

Realizar un programa en **MATLAB** para obtener:

(a) La norma euclidiana de cada vector.

(b) El producto punto: $\mathbf{x} \cdot \mathbf{y}$.

(c) Calcular el ´angulo θ que existe entre los vectores.

Realice la programaci´on considerando variables simb´olicas y aspectos num´ericos.

3.2 Considere las siguientes coordenadas en el sistema de referencia $\Sigma_0(x_0, y_0, z_0)$

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 8 \\ 10.2 \\ 3.12 \end{bmatrix}$$

Realizar un programa en **MATLAB** que realice las siguientes rotaciones:

(a) Alrededor del eje z_0 por un ´angulo $\frac{\pi}{3}$.

(b) Alrededor del eje x_0 por un angulo $-\pi$.

(c) Alrededor del eje y_0 por un angulo $\frac{\pi}{4}$

Analice y discuta sus resultados.

3.3 Sea $R_y(\theta) \in SO(3)$ una matriz ortogonal donde $\theta \in \mathbb{R}$ es el ángulo de rotación alrededor del eje y. Implementar el código **MATLAB** correspondiente para comprobar las siguientes propiedades (usar programación simbólica y numérica):

(a) $R_y(0) = \mathbf{I}$, $\mathbf{I} \in \mathbb{R}^{3\times 3}$ es la matriz identidad.

(b) $R_y(\theta)R_y(\beta) = R_y(\beta)R_y(\theta) = R_y(\theta + \beta) = R_y(\beta + \theta)$.

(c) $R_y^{-1}(\theta) = R_y(-\theta)$

(d) $R_{Ty}(\theta) = R_y^{-1}(\theta)$.

(e) $R_y(\theta)R_{Ty}(\theta) = R_{Ty}(\theta)R_y(\theta) = \mathbf{I}$.

(f) $\det[R_y(\theta)] = 1$ si el sistema de referencia $\Sigma_0(x_0, y_0, z_0)$ es seleccionado por la regla de la mano derecha, en otro caso $\det[R_y(\theta)] = -1$

3.4 Describir el significado de una matriz de rotación.

3.5 Describir el significado de una matriz homogénea.

3.6 ¿Cuáles son las principales propiedades de una matriz homogénea?

Capítulo 4 Cinemática

4.1 ¿Cómo define cinemática?

4.2 ¿Qué es cinemática directa?

4.3 ¿Cuáles son las características principales de la cinemática directa?

4.4 ¿Qué es cinemática inversa? Explicar claramente su función principal.

4.5 Explicar el concepto de cinemática diferencial.

4.6 ¿Qué es el jacobiano del robot?

4.7 ¿Cual es la diferencia que existe entre el jacobiano analítico con el geométrico?

4.8 Explicar el concepto de posición y rotación de robots manipuladores.

4.9 Explicar el significado de una singularidad.

4.10 Explicar el concepto de posici´on de casa de un robot manipulador.

4.11 ¿Cu´al es la diferencia entre rotaci´on y orientaci´on?

4.12 ¿Por qu´e la expresi´on anal´ıtica de la cinem´atica directa depende de la posici´on de casa?

4.13 ¿Cu´ales son los tipos de robots industriales que existen?

4.14 Explicar cinco aplicaciones para cada una de las configuraciones: antropom´orfica, SCARA, esf´erica, cil´ındrica y cartesiana.

4.15 ¿Qu´e robot representa mayores ventajas?

4.16 Desde el punto de vista mec´anico, ¿qu´e tipo de robot presenta mayor complejidad?

4.17 Explicar las ventajas que presenta la convenci´on Denavit-Hartenberg.

4.18 Describir los par´ametros de an´alisis que presenta la convenci´on Denavit-Hartenberg.

4.19 De manera general, se requieren seis coordenadas para posicionar y orientar el extremo final del robot en el espacio tridimensional: ¿Por qu´e la convenci´on Denavit-Hartenberg s´olo utiliza cuatro par´ametros?

4.20 ¿Cuales son las principales hip´otesis que debe satisfacer la convenci´on Denavit-Hartenberg?

Cap´ıtulo 5 Cinem´atica directa cartesiana

5.1 Considere un sistema mecatr´onico denominado centr´ıfuga como el que se muestra en en la figura 5.27.

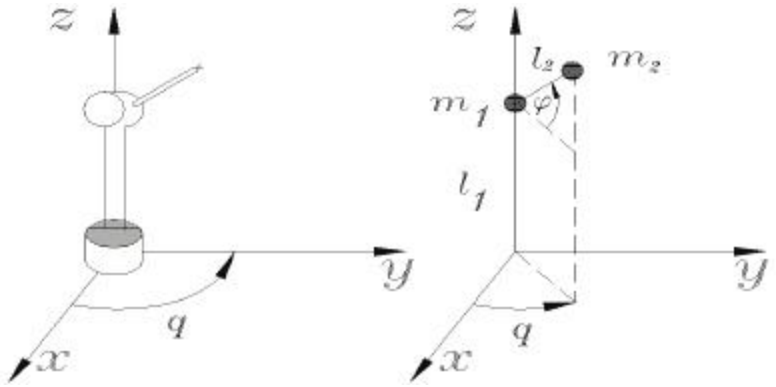


Figura 5.27 Centr´ıfuga.

Obtener:

(a) La tabla de par´ametros DH.

(b) La matriz homog´enea.

(c) Cinem´atica directa.

(d) Cinem´atica inversa.

Asimismo, realizar un programa que despliegue en forma simb´olica los anteriores incisos.

5.2 Considere el robot manipulador en configuraci´on cartesiana. Suponga que se selecciona el sistema de referencia fijo $\Sigma_0(x_0, y_0, z_0)$ de tal forma que el eje z_0 determina el desplazamiento de la variable d_1 como se muestra en la figura 5.28.

Llevar a cabo los pasos necesarios en los respectivos sistemas de referencia $\Sigma_0(x_0, y_0, z_0)$, $\Sigma_1(x_1, y_1, z_1)$, $\Sigma_2(x_2, y_2, z_2)$ para obtener el modelo de cinem´atica directa. ¿En este caso se requiere un sistema auxiliar $\Sigma_{2a}(x_{2a}, y_{2a}, z_{2a})$?

¿Este procedimiento presenta alguna ventaja con el desarrollado en la secci´on 5.6 del robot cartesiano?

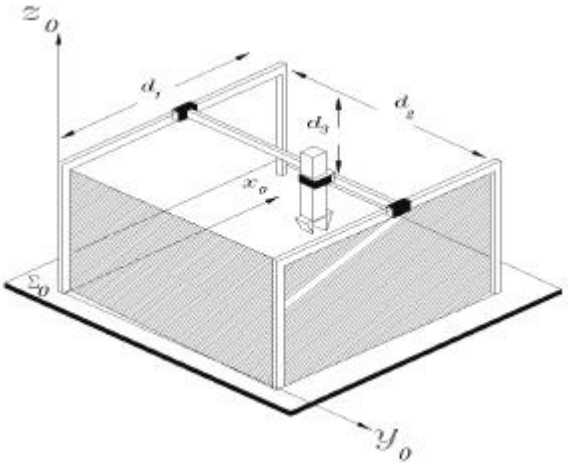


Figura 5.28 Robot cartesiano.

Del procedimiento obtenido, deducir el jacobiano del robot.

5.3 Considere un robot industrial en la configuraci´on antropom´orfica; desarrolle un programa en **MATLAB** para que el extremo final del robot trace una rosa polar de 12 p´etalos con centro en [0.3, – 0.3, 0.5]m.

5.4 Desarrollar un programa en **MATLAB** para que el extremo final de un brazo robot en la configuraci´on SCARA trace la palabra ROBOT en estilo caligr´afico. Dicha palabra deber´a iniciar en las siguientes coordenadas: $[x_0, y_0, z_0]^T$ = [0.3, 0.5, 0.2] m. La longitud y altura de la palabra es propuesta por el usuario.

5.5 Desarrollar un programa en **MATLAB** para que el extremo final de un brazo robot en la configuraci´on cil´ındrica trace la palabra HOLA en estilo italizada. Dicho trazo deber´a empezar en las siguientes coordenadas: $[x_0, y_0, z_0]^T$ = [0.4, 0.7, 0.8] m.

5.6 Desarrollar un programa en **MATLAB** para que el extremo final de un brazo robot en la configuraci´on esf´erica trace la palabra HOLA con tipo de letra romana. Dicha palabra deber´a comenzar en las siguientes coordenadas: $[x_0, y_0, z_0]^T$ = [0.35, 0.6, 0.2] m.

Parte III

Din´amica

La Din´amica como ´area de las ciencias exactas permite explicar todos los fen´omenos f´ısicos de un sistema mecatr´onico o robot manipulador y llevar a cabo procesos de simulaci´on. Bajo esta tem´atica se ubican los objetivos de la **Parte III** compuesta por dos cap´ıtulos: **Cap´ıtulo 6 Din´amica**, presenta la t´ecnica de simulaci´on para sistemas mecatr´onicos y robots manipuladores en base a una ecuaci´on diferencial ordinaria de primer orden descrita por variables de estado; modelos din´amicos de varios sistemas mecatr´onicos y robots manipuladores se desarrollan en c´odigo fuente para **MATLAB**. La t´ecnica de m´ınimos cuadrados para obtener el valor num´erico de los par´ametros del modelo din´amico se detalla en el **Cap´ıtulo 7 Identificaci´on param´etrica**.

Cap´ıtulo 6 Din´amica presenta la forma de simular modelos din´amicos descritos en variables de estado. Generalmente la mayor´ıa de los sistemas f´ısicos tienen su modelo din´amico con ecuaciones diferenciales de orden mayor o igual a dos en variables f´ısicas o generalizadas. La t´ecnica de simulaci´on consiste en convertir el modelo din´amico a una ecuaci´on diferencial de primer orden mediante un adecuado cambio de variables de estado. Se desarrollan librer´ıas en lenguaje **MATLAB** para simular la din´amica de sistemas mecatr´onicos y robots manipuladores.

Cap´ıtulo 7 Identificaci´on param´etrica presenta cincos esquemas de regresi´on lineal para realizar identificaci´on param´etrica de sistemas mecatr´onicos y robots manipuladores. Los esquemas de regresi´on presentados son los modelos: din´amico, din´amico filtrado, energ´ıa, potencia y potencia filtrada. La t´ecnica de m´ınimos cuadrados recursivo es implementada como librer´ıa para **MATLAB** cubriendo los casos: escalar y multivariable de sistemas din´amicos.

La Parte III tambi´en incluye:

Referencias selectas

Problemas propuestos

Capítulo 6

Dinámica

Objetivos

Presentar la t´ecnica de simulaci´on de modelado de robots manipuladores y sistemas mecatr´onicos a trav´es de la estructura de ecuaci´on diferencial ordinaria en variables fase.

Objetivos particulares:

- Estructura matem´atica fundamental de simulaci´on.
- Sistema masa resorte amortiguador.
- Centr´ıfuga.
- Robot antropom´orfico: p´endulo, robot de 2 y 3 gdl.
- Robot cartesiano de 3 gdl.
- Aplicaciones en control de posici´on.

6.1 Introducci´on

Robots manipuladores son sistemas mec´anicos muy complejos cuya descripci´on anal´ıtica requiere de ecuaciones diferenciales. La naturaleza no lineal, multivariable y fuerte acoplamiento en su comportamiento din´amico ofrece un amplio espectro en la formulaci´on de problemas de control te´oricos y pr´acticos. El modelo din´amico del robot manipulador permite explicar todos los fen´omenos f´ısicos que se encuentran en su estructura mec´anica, tales como efectos inerciales, fuerzas centr´ıpetas y de Coriolis, par gravitacional y fricci´on, los cuales son fen´omenos f´ısicos propios de la naturaleza din´amica del robot. Hay varios m´etodos de modelado de la f´ısica como el de Newton o el de Hamilton. Sin embargo, la mejor opci´on como metodolog´ıa de modelado la representa las ecuaciones de movimiento de Euler-Lagrange debido a las propiedades matem´aticas que se deducen de manera natural, ya que facilitan el an´alisis y diseño de algoritmos de control.

El modelo din´amico de robots manipuladores es fundamental en aplicaciones de simulaci´on, diseño y construcci´on del sistema mec´anico, as´ı como en an´alisis y diseño de algoritmos de control. En el a´rea de **simulaci´on** el modelo din´amico es estrat´egico debido a que puede reproducir todos los fen´omenos f´ısicos del robot sin la necesidad de usar un robot real (realidad virtual), y esta caracter´ıstica resulta clave para evaluar algoritmos de control, t´ecnicas de planeaci´on de trayectorias, programaci´on de aplicaciones industriales, etc. La simulaci´on es el empleo del modelo din´amico para analizar y describir su comportamiento din´amico en una computadora o sistema m´ınimo digital y de ah´ı inferir aplicaciones. Es importante no confundir simulaci´on con animaci´on, ya que son procesos diferentes; la animaci´on no requiere incorporar efectos din´amicos en el movimiento del robot, generalmente son ecuaciones est´aticas como la cinem´atica directa.

La construcci´on de un robot se fundamenta en el modelo din´amico; los esquemas y planos de ingenier´ıa de los eslabones se deducen directamente del modelo din´amico y se trasladan a un programa CAD para su maquinado y construcci´on mec´anica. De esta forma, un robot industrial puede ser estudiado y se pueden hacer las adecuaciones pertinentes antes de llegar a la etapa de construcci´on f´ısica.

6.2 Estructura matemática para simulación

La gran mayoría de los sistemas mecánicos contienen como parte del modelo dinámico ecuaciones diferenciales de orden mayor o igual a 2. Sin embargo, para propósitos de simulación es recomendable transformar ese modelo a un sistema dinámico compuesto por una ecuación diferencial ordinaria de primer orden debido a que este tipo de ecuaciones es conocida, está ampliamente documentada y es fácil de programar.

La estructura de una ecuación diferencial ordinaria (ode) de primer orden tiene la siguiente forma:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{6.1}$$

donde $\mathbf{x} \in \mathbb{R}^n$ se conoce como variable de estado fase, la cual proporciona información interna sobre la dinámica del sistema, es una función continua en el tiempo $\mathbf{x} = \mathbf{x}(t)$, $n \in \mathbb{N}$ es un número natural que indica la dimensión euclidiana, la derivada temporal de la variable de estado $\dot{\mathbf{x}} \in \mathbb{R}^n$ existe y también es continua en el tiempo $\dot{\mathbf{x}} = \dot{\mathbf{x}}(t)$. La notación $\dot{\mathbf{x}}$ significa $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$. La función $\mathbf{f} \in \mathbb{R}^n$ es un mapa vectorial continua en la variable de estado $\mathbf{x}(t)$.

La estructura matemática $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ es para ambos sistemas: lineales y no lineales. La característica de ser lineal o no lineal es referente a la variable de estado. El modelo (6.1) se conoce como sistema dinámico autónomo, debido a que la variable tiempo t se encuentra presente de manera implícita, es decir no aparece como parte de la estructura matemática, se encuentra incluida como parte de las propiedades de la variable de estado $\mathbf{x}(t)$. La gran mayoría de los sistemas mecánicos corresponden a esta forma, sobre todo considerando que no hay variación temporal de sus parámetros, los cuales se consideran constantes al menos en un tiempo suficientemente grande.

De otra manera se tendría como parte de la ecuación (6.1) dependencia explícita del tiempo, es decir $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$; a este tipo de sistemas se les conoce como sistema dinámicos no autónomos, los cuales no non objetos de estudio en la presente obra.

♣ Ejemplo 6.1

Convertir a la forma $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ el siguiente sistema lineal expresado por la funci´on de transferencia:

$$\frac{y}{u} = \frac{2}{s^2 + 3s + 1}$$

donde $y \in$ IR es la respuesta del sistema, $u \in$ IR es la entrada.

Soluci´on

El concepto de funci´on de transferencia es exclusivo para sistemas lineales, se define como la relaci´on de la salida y la entrada cuando las condiciones iniciales del sistema son cero.

Es necesario aclarar que el t´ermino s que aparece en una funci´on de transferencia significa frecuencia y es un n´umero imaginario formado como: $s = jw$, siendo w una variable que indica frecuencia, cuyas unidades son rad/seg. Cuando transformamos la funci´on de transferencia a una ecuaci´on diferencial, entonces s adquiere otra interpretaci´on, en este caso funciona como un operador $s = \frac{d}{dt}$. Bajo este escenario, debe interpretarse que $s^2 = \frac{d}{dt}\frac{d}{dt} = \frac{d^2}{dt^2}$ (teniendo en mente condiciones iniciales cero).

Por lo tanto, el procedimiento para convertir la funci´on de transferencia del sistema a una ecuaci´on diferencial ordinaria de primer orden es el siguiente:

$$\frac{y}{u} = \frac{2}{s^2 + 3s + 1} \Rightarrow s^2 y + 3sy + y = 2u$$
$$\Rightarrow \ddot{y} + 3\dot{y} + y = 2u$$

donde se ha empleado $s^2 y = \ddot{y}$ $sy = \dot{y}$. La notaci´on $\dot{y} = \frac{dy(t)}{dt}$, $\ddot{y} = \frac{d^2 y(t)}{dt^2}$

Ahora, se requiere un cambio de variable de estado, sea $x_1 = y$, $x_2 = x_1$, $\dot{x}_2 = -3x_2 - x_1 + 2u$ entonces:

$$\ddot{y} + 3\dot{y} + y = 2u \Rightarrow \underset{\ddot{y}}{\dot{x}_2} + 3\underset{\dot{y}}{x_2} + \underset{y}{x_1} = 2u$$

$$\underbrace{\frac{d}{dt}\begin{matrix}x_1\\x_2\end{matrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{matrix}0 & 1\\-1 & -3\end{matrix}\begin{matrix}x_1\\x_2\end{matrix}}_{\mathbf{f}(\mathbf{x})} + \begin{matrix}0\\2\end{matrix} u$$

donde $\mathbf{x} = [x_1, x_2]^T$.

Cuando la ecuación diferencial original se encuentra en la forma de variables fase $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, entonces el sistema dinámico (convertido) ya tiene la estructura matemática adecuada para realizar simulación usando la función ode45(···).

♣ Ejemplo 6.2

Convertir a la forma $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ el siguiente sistema no lineal:

$$\alpha_2\ddot{y} + \alpha_1 \operatorname{sen}(\dot{y}) + \alpha_0 \cos(y) = \beta u$$

donde $y \in \mathbb{R}$ es la respuesta del sistema, $u \in \mathbb{R}$ es la entrada, $\alpha_2, \alpha_1, \alpha_0, \beta \in \mathbb{R}^+$.

Solución

La función de transferencia es exclusiva de sistemas lineales, para un sistema no lineal no existe dicho concepto.

La conversión a la forma de ecuación diferencial ordinaria de primer orden en variables fase se realiza de la siguiente forma. Sea $x_1 = y$, $x_2 = \dot{x}_1$, $\dot{x}_2 = (-\alpha_1 \operatorname{sen}(x_2) - \alpha_0 \cos(x_1) + \beta u)/\alpha_2$. Por lo tanto, la estructura matemática solicitada es:

$$\underbrace{\frac{d}{dt}\begin{matrix}x_1\\x_2\end{matrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{matrix}x_2\\\frac{1}{\alpha_0}\left[\beta u - \alpha_1 \operatorname{sen}(x_2) - \alpha_0 \cos(x_1)\right]\end{matrix}}_{\mathbf{f}(\mathbf{x})}$$

6.3 Sistema masa-resorte-amortiguador

Uno de los sistema mec´anicos t´ıpicos que ha sido estudiado ampliamente en cursos de control autom´atico de la carreras de ingenier´ıa mecatr´onica y electr´onica es el sistema masa resorte amortiguador. La importancia de este sistema radica en comprender su comportamiento din´amico para entender el funcionamiento cualitativo de otros sistemas, por ejemplo: circuitos el´ectricos (resistencia, capacitor, inductancia), esquemas de control (algoritmo proporcional derivativo de robots manipuladores).

Considere un sistema din´amico como el que se muestra en la figura 6.1, formado por una masa m que se encuentra unida a un resorte de rigidez k y constante de amortiguamiento b debido a la viscosidad del medio ambiente. La entrada est´a especificada por una fuerza f y el desplazamiento de la masa por x.

Figura 6.1 Sistema masa resorte amortiguador.

El modelo din´amico que describe al sistema masa resorte amortiguador se encuentra dado por:

$$f = m\ddot{x} + b\dot{x} + kx$$

Fuerza externa — Fuerza inercial — Fricci´on viscosa — Ley de Hooke

La funci´on de transferencia del sistema masa resorte amortiguador est´a dada por:

$$\frac{x}{f} = \frac{1}{ms^2 + bs + k} \tag{6.2}$$

la ecuaci´on caracter´ıstica de los polos del sistema est´a dada por

$$\frac{-b \pm \sqrt{b^2 - 4mk}}{2k} \tag{6.3}$$

El sistema masa resorte amortiguador oscilará cuando se cumpla $(b^2 - 4mk) < 0$, y será amortiguado si $(b^2 - 4mk) > 0$.

La conversión de la ecuación (6.2) a una ecuación diferencial ordinaria de primer orden en variables fase $\mathbf{x} = [x, \dot{x}]^T$ se realiza de la siguiente forma:

$$\underbrace{\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \dot{x} \\ m^{-1}[f - b\dot{x} - kx] \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} \tag{6.4}$$

♣ Ejemplo 6.3

Considere el modelo dinámico del sistema masa resorte amortiguador (6.4) utilizando los valores numéricos de los parámetros del sistema de la tabla 6.1. Realizar un programa en **MATLAB** para graficar el desplazamiento lineal x y la velocidad de la masa cuando se le aplica una fuerza constante $f = 50$N durante 5 segundos y después de este tiempo desaparece la fuerza aplicada.

Tabla 6.1 Parámetros del sistema masa resorte amortiguador

Masa	Fricción viscosa	Rigidez
m=5 kg	b =0.16 kg/seg	k =0.6 kg/seg^2

Solución

El programa en código **MATLAB** que contiene el modelo dinámico del sistema masa resorte amortiguador (6.4) se describe en el cuadro 6.2. En este programa la variable x_mra representa el desplazamiento lineal que tiene de la masa al aplicarse una fuerza constante de 5N durante 5 segundos. Por medio del programa 6.1 se realiza la simulación del sistema.

En la figura 6.2 (modo oscilatorio) se muestra el desplazamiento lineal x_mra y la velocidad de movimiento xp_mra que tiene la masa. Por los valores que presentan en la tabla 6.1 el sistema masa resorte amortiguador satisface la condición $(b^2 - 4mk) < 0$ de la ecuación (6.3) que corresponde a un oscilador, por eso cuando

desaparece la fuerza constante despu´es de 5 segundos el sistema se queda oscilando permanentemente (el tiempo de simulaci´on es de 100 segundos).

En realidad el sistema masa resorte amortiguador cuando se encuentra en la configuraci´on de oscilador debido a los valores num´ericos de sus par´ametros, el sistema empezar´a a oscilar con una condici´on diferente a cero, independientemente de la magnitud de la fuerza aplicada.

Ahora, sup´ongase que los valores de los par´ametros son: m=5 kg, la constante de rigidez del resorte k = 2 kg/seg^2 y el coeficiente de fricci´on viscosa b = 4 Kg/seg; con estos valores se configura el sistema masa resorte amortiguador en modo amortiguado de tal forma que en la ecuaci´on (6.3) satisface ($b^2 - 4mk$) > 0, es decir el factor de amortiguamiento es adecuado para evitar oscilaciones y grandes sobre impulsos en la respuesta transitoria.

La figura 6.2 muestra la simulaci´on en modo amortiguado. Un´icamente, durante los primeros 5 segundos se le aplica la fuerza de f =5 N. Despu´es de este transitorio la masa regresa a su posici´on inicial, restableci´endose la elongaci´on del resorte y desapareciendo el efecto de amortiguamiento. Obs´ervese que cuando desaparece la fuerza constante, la posici´on del sistema es atra´ıda por el punto de equilibrio, quedando en reposo.

C´odigo Fuente 6.1 Simulador de smr.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa smr simu.m
```

Simulador de smr.m

```
1 clc; clear all; close all; format short
2 %par´ametros de simulaci´on
3 ti=0; h=0.0025; tf = 100; ts=ti:h:tf;
4 opciones=odeset('RelTol',1e-3,'InitialStep', 2.5e-3, 'MaxStep',2.5e-3);
5 [t,x]=ode45('mra',ts,[0; 0],opciones);
6 plot(t,x(:,1),t,x(:,2))
```

C´odigo Fuente 6.2 Sistema masa resorte amortiguador

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa mra.m
%requiere del programa mra simu.m
```

Sistema masa resorte amortiguador

```
function xp =mra(t,x)
    %vector de estados
    %posici´on articular
    x_mra=x(1);
    %velocidad articular
    xp_mra=x(2);
    %par´ametros del sistema masa resorte amortiguador
    %masa
    m=5;
    %constante de rigidez del resorte
    k=0.6;
    %amortiguamiento: coeficiente de fricci´on viscosa
    b=0.16;
    %fuerza aplicada durante 5 segundos
    if (t<5)
        f=5;
        else
        f=0;
    end
    %aceleraci´on del sistema masa resorte amortiguador
    xpp_mra=(f-b*xp_mra-k*x_mra)/m;
    %vector de salida
    xp=[ xp_mra ; %velocidad articular
          xpp_mra] ; %aceleraci´on articular
end
```

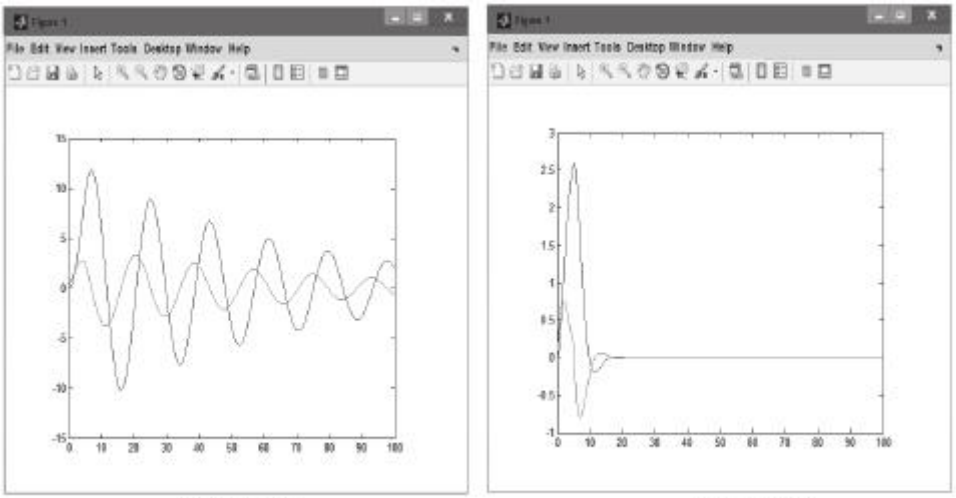


Figura 6.2 Respuesta del sistema masa resorte amortiguador.

6.4 Sistema lineal escalar

Sistemas lineales se utilizan ampliamente como estimadores de velocidad, filtrado en robots manipuladores y sistemas mecatr´onicos. La estructura matem´atica b´asica de un sistema lineal escalar de primer orden en variables fase est´a dada por:

$$\dot{x} = -ax + bu \tag{6.5}$$

$$y = cx \tag{6.6}$$

donde $a, b, c \in \mathbb{R}_+$ son los par´ametros del sistema, $x, \dot{x} \in \mathbb{R}$ son la variable de estado y la derivada temporal de la variable de estado, respectivamente; la entrada es $u \in \mathbb{R}$ y la salida del sistema est´a dada por $y \in \mathbb{R}$.

La funci´on de transferencia del sistema lineal (6.5)-(6.6) tiene la siguiente forma:

$$\frac{Y}{U} = \frac{cb}{s+a} = c\,\frac{b}{a}\,\frac{1}{\frac{1}{a}s+1} = c\,\frac{b}{a}\,\frac{1}{Ts+1} \tag{6.7}$$

donde $T = \frac{1}{a}$, se conoce como la constante de tiempo. Para una constante de tiempo

pequeña la respuesta del sistema es r´apida, de otra manera si la constante de tiempo es grande, la respuesta del sistema es lenta.

Por otro lado, la funci´on de transferencia representa un filtro pasa-bajas, donde a significa la frecuencia de corte, tambi´en determina el ancho de banda del filtro y $\frac{b}{a}$ es la ganancia (frecuencia cero).

Estimador de velocidad y filtrado

Entre las aplicaciones del sistema lineal (6.5)-(6.6) se encuentra como estimador de velocidad y filtrado para robots manipuladores y sistemas mecatr´onicos. Por ejemplo, si q_i representa la ¡-´esima posici´on articular de un robot, la manera de estimar la ¡-´esima velocidad $\dot{q}_i$ es por medio de la siguiente forma:

$$\dot{q}_i \approx \dot{F}_i = -\lambda F_i + \lambda q_i \tag{6.8}$$

donde λ es la frecuencia de corte del filtro de ganancia unitaria, F_i es ¡-´esimo estado del filtro, representa la señal filtrada de la entrada (q_i), y $\dot{F}_i$ es la estimaci´on de la velocidad $\dot{q}_i$.

Para comprobar la afirmaci´on que F_i representa la señal filtrada de q_i, considere la funci´on de transferencia de (6.8):

$$F_i = \frac{\lambda}{\lambda + s} q_i \tag{6.9}$$

observe que la señal de entrada q_i es procesada por el filtro pasa bajas $\frac{\lambda}{s+\lambda}$.

Adem´as, la derivada temporal de la señal filtrada de la entrada q_i aproxima a la derivada temporal de $\dot{q}_i$ (velocidad), es decir $\dot{q}_i \approx \dot{F}_i$; este procedimiento es v´alido para cualquier articulaci´on del robot.

♣ Ejemplo 6.4

Escribir un programa en **MATLAB** para analizar la respuesta del sistema lineal (6.5)-(6.6) para una entrada unitaria $u = 1$.

Solución

El programa que realiza la implementación del sistema lineal escalar de primer orden (6.5-6.6) se encuentra en el cuadro 6.3. La señal de entrada u es una constante de magnitud unitaria. El programa principal que realiza la simulación se encuentra en el cuadro 6.4. El tiempo de simulación es de 5 segundos y la respuesta a la entrada unitaria no contiene sobre impulsos en régimen transitorio tal y como se describe en la figura 6.3. Dentro de las características en la respuesta a un escalón o entrada constante de un sistema lineal, el transitorio no presentaría sobre impulsos y de manera suave convergería al valor de la entrada. En la primera constante de tiempo T, la respuesta ha alcanzado el 63.2% del valor de la magnitud de la entrada u. Para este ejemplo la constante de tiempo es $T = \frac{1}{3} = 0.3333$ segundos, en la segunda constante de tiempo $2T$ la respuesta del sistema ha alcanzado el 86.5% del valor final.

Código Fuente 6.3 Sistema lineal escalar

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega,  Fernando Reyes Cortés
%Capítulo 4 Dinámica. Programa sle.m
Sistema lineal escalar
function xp =sle(t,x)
    a=3;
    b=3;
    u=1;
    xp=-a*x+b*u;
end
```

Para las constantes de tiempo 3τ, 4τ y 5τ la respuesta tiene 95%, 98.2% y 99.3% del valor final, respectivamente. Para finalidades pr´acticas despu´es de 5τ, se puede decir que el sistema ha entrado en estado estacionario.

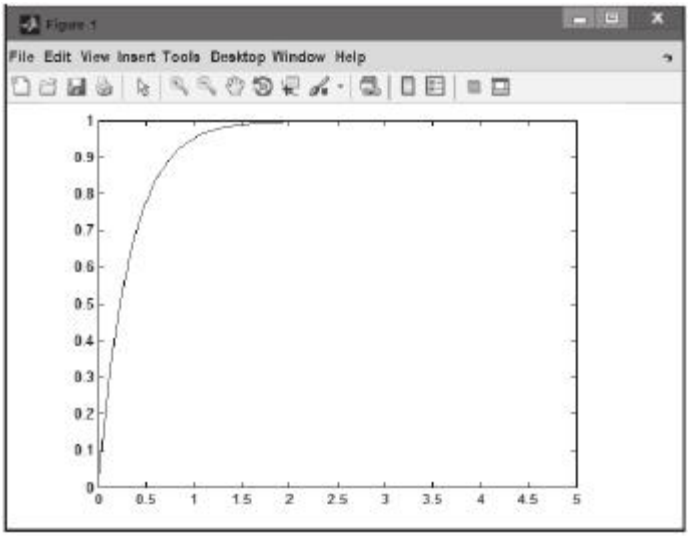


Figura 6.3 Respuesta a un escal´on del sistema lineal escalar.

C´odigo Fuente 6.4 sle_simu

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa sle_simu.m
% este programa requiere del archivo: sle.m
```

sle_simu

```
1 clc; clear all; close all;
2 format short
3 %tiempo de simulaci´on (segundos)
4 ti=0; h=0.0025; tf = 5;
5 ts=ti:h:tf; %vector tiempo
6 opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
7 [t,x]=ode45('sle',ts,0,opciones);
8 plot(t,x)
```

♣ Ejemplo 6.5

Escribir un programa en **MATLAB** para que un sistema lineal escalar realice el filtrado de una señal contaminada con ruido $u = 5\,\text{sen}(t) + \text{ruido}(t)$.

Solución

Considere una señal $u(t)$ con una envolvente de ruido de la siguiente forma: $u = 5\,\text{sen}(t) + \text{random('Normal',t,t)}$, se empleará el sistema lineal $\dot{x} = -3x + 3u(t)$ para obtener la señal filtrada de la entrada $u(t)$. El programa que implementa la función filtro a través de un sistema lineal se encuentra en el cuadro 6.5. Como generador de ruido se emplea la función de datos aleatorios random utilizando distribución 'Normal' de datos, la cual se añade a la función senoidal $5\,\text{sen}(t)$.

El programa principal que realiza la simulación se encuentra en el cuadro 6.6, en la línea 7 se resuelve numéricamente el filtro, en la línea 8 se reproduce la señal de entrada contaminada, y se compara con la señal libre de ruido contenida en la variable $x(t)$ (líneas 8 y 9). La figura 6.4 muestra la comparación entre la señal original u con ruido y filtrada (figura izquierda).

Código Fuente 6.5 Aplicación filtrado

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega,  Fernando Reyes Cortés
%Capítulo 4 Dinámica. Programa filtro.m
```

Aplicación filtrado

```
1 function xp=filtro(t,x)
2     u=5*sin(t)+random('Normal',t,t);
3     a=3;
4     b=3;
5     xp=-a*x+b*u;
6 end
```

En la gr´afica derecha de la figura 6.4 se presentan los detalles de la señ1 filtrada; observe que esta señal est´a restaurada lo mejor posible despu´es de haber sido contaminada con ruido. El filtro tiene un par´ametro de sinton´ıa llamado frecuencia de corte, que en este caso es $a = 3$rad/seg. Este par´ametro puede adquirir otro valor para mejorar la calidad de filtrado de la señ1 de entrada u.

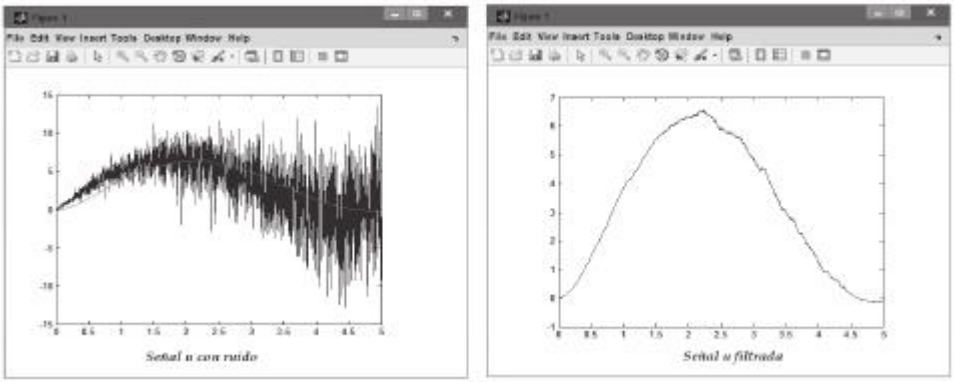


Figura 6.4 Filtrado de señ1es.

C´odigo Fuente 6.6 sle_simu

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa sle_simu.m
% este programa requiere del archivo: filtro.m
```

sle_simu

```
1  clc; clear all; close all;
2  format short
3  %tiempo de simulaci´on (segundos)
4  ti=0; h=0.0025; tf = 5;
5  ts=ti:h:tf; %vector tiempo
6  opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
7  [t,x]=ode45('filtro',ts,0,opciones);
8  u=5*sin(t)+random('Normal',t,t);
9  subplot(2,2,1); plot(t,u)
10 subplot(2,2,2); plot(t,x)
```

6.5 Centr´ıfuga

La centr´ıfuga es un sistema mecatr´onico que se describe en la figura 6.5, el cual consta de un servomotor que gira alrededor del eje z_0, el a´ngulo de rotaci´on est´a descrito por la variable q_1. La distancia del origen del sistema de referencia fijo $\Sigma_0(x_0, y_0, z_0)$ al extremo final est´a representado por $l_1 + l_2 \text{ sen}(\delta)$, donde l_1 es la longitud del servomotor m´as la longitud del rotor, l_2 es la longitud de la barra met´alica que se encuentra acoplada al rotor, y δ es un ´angulo constante que significa la inclinaci´on de la barra l_2 con respecto a la horizontal, l_{c2} representa el centro de masa y el momento de inercia $\mathbf{I}$ del servomotor de la centr´ıfuga. El movimiento rotatorio de la centr´ıfuga describe un c´ırculo sobre el plano $x_0 - y_0$. La acci´on de la gravedad g se encuentra en direcci´on contraria al eje z_0.

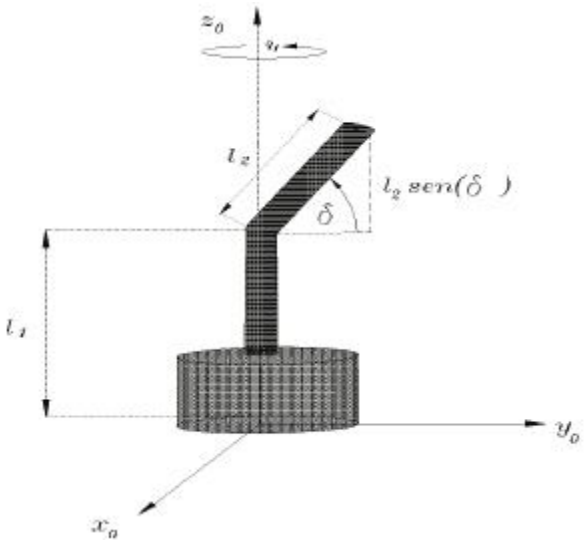


Figura 6.5 Centr´ıfuga.

El modelo din´amico de una centr´ıfuga es una ecuaci´on diferencial de segundo orden expresada de la siguiente manera:

$$\tau = [ml_{c2}^2 \text{ sen}(\delta) + \mathbf{I}_{sm}]\ddot{q} + b\dot{q} + \mathbf{f}_c \text{signo}(\dot{q}) + \mathbf{f}_e[1 - |\text{signo}(\dot{q})|] \qquad (6.10)$$

La ecuaci´on (6.10) puede ser convertida a una ecuaci´on diferencial ordinaria de

primer orden de la forma $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, a trav´es de la siguiente expresi´on:

$$\underbrace{\frac{d}{dt}\begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \dot{q}_1 \\ [m_1 l_{c2}^2 \operatorname{sen}(\delta) + I_{sm}]^{-1} [\tau_1 - b_1 \dot{q}_1 - f_{c1} \operatorname{signo}(\dot{q}_1) - f_{e1} [1 - |\operatorname{signo}(\dot{q}_1)|]] \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} \tag{6.11}$$

6.2 Ejemplo 6.6

Considere el modelo din´amico de la centr´ıfuga expresado como una ecuaci´on diferencial de primer orden (6.11). Tomar en cuenta los valores num´ericos de los par´ametros de la centr´ıfuga que aparecen en la tabla 6.2. Realizar un programa en **MATLAB** para que la centr´ıfuga gire a una revoluci´on por segundo en ambos sentidos (positivo/negativo).

Tabla 6.2 Par´ametros de la centr´ıfuga

Masa de la centr´ıfuga	Fricci´on viscosa	Fricci´on de Coulomb	Fricci´on est´atica
m_1=5 kg	b_1 =0.12 Nm-seg/rad	f_{c1} =0.11 Nm	f_{e1} =0.09 Nm
Centro de masa de la barra	Longitud de la barra	Momento de inercia del rotor	Ángulo de inclinaci´on de la barra
l_{c2}=0.01m	l_2=0.15m	I_{sm}=0.08 Nm-seg^2/rad	$\delta = \frac{45\pi}{180}$ rad

Soluci´on

El modelo num´erico se obtiene combinando el modelo din´amico (6.11) con los valores num´ericos de los par´ametros de la centr´ıfuga contenidos en la tabla 6.2.

$$\frac{d}{dt}\begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \frac{1}{} \end{bmatrix} \tag{6.12}$$

para lograr que el sistema mecatr´onico oscile en ambos sentidos a una revoluci´on por segundo (360 grados/seg), se requiere que su movimiento sea generado por una

energía tipo senoidal. Es decir, al servomotor se le aplica una energía de la forma $\tau_1 = 2\,\mathrm{sen}(t)$, para generar un movimiento cíclico. Con este tipo de energía la centrífuga funciona como una lavadora; cuando se requiera implementar la función centrifugado, entonces el tipo de energía aplicada al servomotor es a través de un par constante.

El programa centrifuga.m 6.7 contiene el modelo dinámico de la centrífuga con los parámetros de la tabla 6.2. La ecuación programada corresponde a la estructura de variables de estados de los modelos 6.11 o 6.12. El programa principal centrifuga_simu 6.8 realiza la solución numérica del modelo dinámico de la centrífuga usando el método de integración numérica de Runge-Kutta de la función ode45(···).

Código Fuente 6.7 centrifuga

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega, Fernando Reyes Cortés
%Capítulo 4 Dinámica. Programa centrifuga.m
```

centrifuga

```
1  function xp = centrifuga( t,x)
2      m1=5;%masa de la centrífuga
3      lc2=0.01;% centro de masa de la barra inclinada
4      delta=45*pi/180; %ángulo de inclinación de la barra l2
5      I_sm=0.08;% momento de inercia del rotor del servomotor
6      b1=0.12; fc1=0.11; fe1=0.09;%coeficientes de fricción
7      q1=x(1);% posición articular
8      qp1=x(2);% velocidad articular
9      tau=2*sin(t);% energía aplicada al servomotor
10     %aceleración rotacional de la centrífuga
11     qpp1=(tau1-b1*qp1-fc1*sign(qp1)-fe1*(1-abs(sign(qp1))))/
       (m1*lc2*lc2*sin(delta)+I_sm);
12     xp=[ qp1; qpp1 ];% vector de salida
13 end
```

C´odigo Fuente 6.8 centrifuga_simu

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa centrifuga_simu.m
% este programa requiere del archivo: centrifuga.m
```

centrifuga_simu

```
1  clc; clear all; close all;
2  format short
3  %tiempo de simulaci´on (segundos)
4  ti=0; h=0.0025; tf = 10;
5  ts=ti:h:tf; %vector tiempo
6  opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
7  %integraci´on num´erica del modelo din´amico de la centr´ıfuga
8  [t,x]=ode45('centrifuga',ts,[0; 0],opciones);
9  %graficar variable articular de la posici´on q1 =x(:,1)
10 %la velocidad de movimiento est´a dada por q1' =x(:,2)
11 plot(t,x(:,1),t,x(:,2))
```

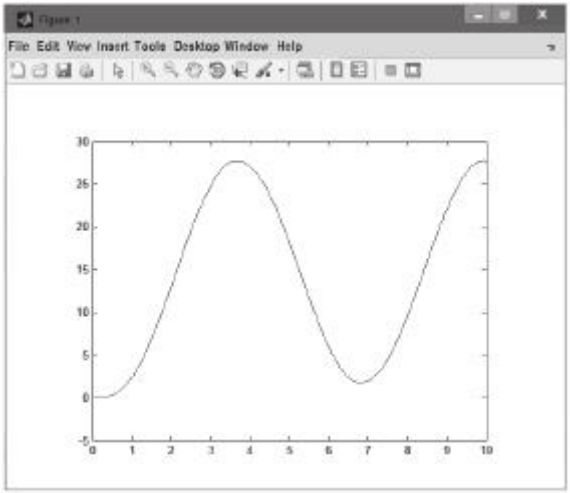


Figura 6.6 Movimiento de la centr´ıfuga.

La figura 6.6 describe la forma de desplazamiento rotacional que tiene la variable articular q_1 cuando se aplica energ´ıa de la forma $\tau_1 = 2\,\mathrm{sen}(t)$, el movimiento de la centr´ıfuga es oscilatorio; este es el principio b´asico de funcionamiento de las lavadoras. Con el tipo de energ´ıa senoidal aplicada al servomotor, la centr´ıfuga gira 360 grados por segundo alrededor del eje z_0, cambiando su sentido de giro cada periodo $t = 6.28$ segundos.

6.6 P´endulo

Actualmente el p´endulo es una planta de estudio vigente debido a su din´amica no lineal y sus aplicaciones pr´acticas, lo que lo convierte en un sistema clave para prop´ositos de investigaci´on cient´ıfica, y en docencia representa una herramienta pedag´ogica muy útil para la enseñanza de la din´amica de sistemas. La figura 6.7 muestra un p´endulo robot el cual est´a formado por un servomotor y una barra met´alica de longitud l_1. El sistema de referencia se elige de tal manera que el eje z_0 coincida con el eje de rotaci´on del servomotor (perpendicular al plano de la hoja). La acci´on de la gravedad g est´a dirigida en direcci´on del eje y_0 negativo. El momento de inercia se denota por I_{sm}, el centro de masa se representa como l_{c1} y la masa del p´endulo por m_1.

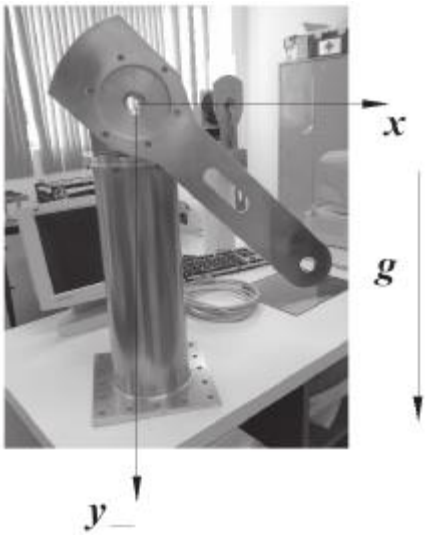


Figura 6.7 P´endulo robot.

El modelo din´amico de un p´endulo-robot est´a dado por:

$$\tau = [ml_c^2 + I]\ddot{q} + mgl_c \operatorname{sen}(q) + b\dot{q} + f_c \operatorname{signo}(\dot{q}) + f_e[1 - |\operatorname{signo}(\dot{q})|] \quad (6.13)$$

♣ Ejemplo 6.7

Considere el modelo din´amico del p´endulo (6.13) y realice un programa en lenguaje **MATLAB** para describir los fen´omenos f´ısicos del sistema. Para llevar a cabo la simulaci´on, considere los valores num´ericos de los par´ametros del p´endulo que presenta la tabla 6.3.

Soluci´on

El p´endulo robot prototipo de la figura 6.7 tiene un servomotor de transmisi´on directa con un torque m´aximo de 15Nm ($\pm$ 15 Nm para el primer y tercer cuadrante, respectivamente). La longitud de la barra l_1 es de 0.45m, los valores num´ericos de los par´ametros que forman el modelo din´amico como el momento de inercia, centro de gravedad, coeficientes de fricci´on viscosa, Coulomb y est´atica est´an concentrados en la tabla 6.3.

Tabla 6.3 Par´ametros del p´endulo robot

Masa	Fricci´on viscosa	Fricci´on de Coulomb	Fricci´on est´atica
m_1=3.88 kg	b_1 =0.16 Nm-seg/rad	f_{c1} =0.19 Nm	f_{e1} =0.20 Nm
Centro de masa	**Longitud**	**Mom. de inercia del rotor**	**Cap. del servomotor**
l_{c1}=0.081 m	l_1=0.45 m	I_{r1}=0.16 Nm-seg^2/rad	$\pm$ 15Nm

La masa m_1 del p´endulo incluye la masa de la barra, rotor del servomotor y tornillos. El centro de masa l_{c1} del p´endulo se calcula de la siguiente manera:

$$l_{c1} = \frac{l_{crotor} m_{rotor} + l_{cbarra} m_{barra} + l_{ctornillos} m_{tornillos}}{m_{rotor} + m_{barra} + m_{tornillos}}$$

El centro de masa del rotor l_{Crotor} es cero debido a las propiedades geom´etricas del servomotor, y adem´as el origen del sistema de referencia $\Sigma_0(x_0, y_0, z_0)$ se coloca en el centro de masa l_{Crotor}. Los tornillos sujetan la barra met´alica de aluminio al rotor del servomotor; estos tornillos se encuentran igualmente espaciados sobre una circunferencia cuyo centro pasa el eje de rotaci´on, el cual est´a alineado con el eje z_0. Por lo tanto, el centro de masa de los tornillos $l_{Ctornillos}$ es cero. El momento de

inercia del p´endulo I_1 es igual a la contribuci´on del momento de inercia del rotor del servomotor I_{r1} m´as el momento de inercia de la barra de aluminio $m_1 l^2_{1_c}$ (teorema de ejes paralelos de inercias).

Los programas 6.9 y 6.10 realizan la simulaci´on del p´endulo. La energ´ıa aplicada al servomotor del p´endulo es tipo senoidal, por ejemplo de la forma $\tau_1 = \text{sin}(t)$, donde t es la evoluci´on del tiempo. Este tipo de torque produce un movimiento oscilatorio en su espacio de trabajo sobre el plano $x_0 - y_0$, entonces se genera desplazamiento rotacional q_1 alrededor del eje z_0 y velocidad de movimiento articular $\dot{q}_1$; con estas variables es posible obtener la din´amica del p´endulo tales como efector inercial $(I_1 \ddot{q}_1)$, fricci´on $(b_1\dot{q}_1 + f_{c1}\ \text{signo}(\dot{q}_1))$ y par gravitacional $(m_1 g l_{c1}\ \text{sen}(q_1))$.

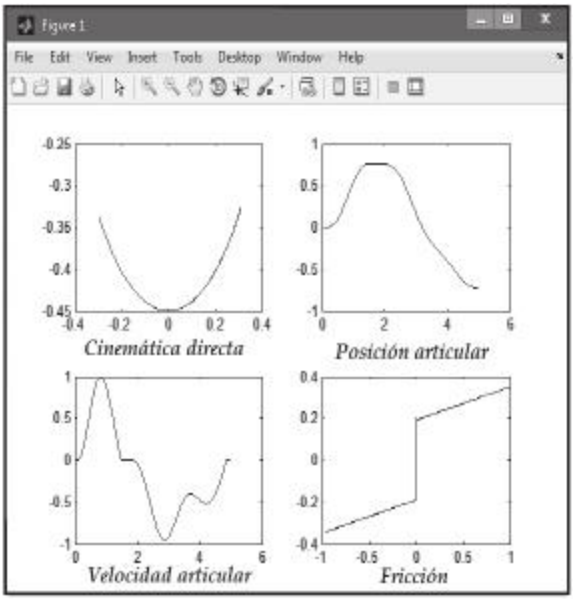


Figura 6.8 Respuesta del p´endulo.

La figura 6.8 muestra la respuesta del p´endulo robot a una se˜nal de entrada $\tau 1 = \text{sen}(t)$; se emplea la cinem´atica directa del p´endulo (y_0 vs x_0) para graficar el desplazamiento del extremo final del robot en su espacio de trabajo. En este caso, la posici´on de casa del p´endulo se ubica sobre el eje y_0. La posici´on articular en funci´on del tiempo $q_1(t)$ toma un perfil parecido a la se˜nal de entrada $\tau_1(t)$; la variable articular $\dot{q}_1(t)$ representa la velocidad de movimiento, y adquiere magnitudes de 0.5 rad/seg (90 grados/seg) en ambas direcciones (positivo/negativo).

El fen´omeno de fricci´on viscosa y de Coulomb se grafica en funci´on de la velocidad de movimiento (fricci´on vs velocidad $\dot{q}_1$); note que dicho fen´omeno disipativo se encuentra ubicado dentro del primer y tercer cuadrante. La magnitud que alcanza el fen´omeno de fricci´on es de 0.4 Nm, que representa el 2.6 % de la capacidad m´axima del servomotor. El fen´omeno de fricci´on convierte la energ´ıa cin´etica en energ´ıa t´ermica.

C´odigo Fuente 6.9 P´endulo

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa pendulo.m
% este programa requiere del archivo pendulo_simu.m
```

P´endulo

```
1  function xp =pendulo(t,x)
2      %vector de estados
3      q1=x(1); %posici´on articular
4      qp1=x(2); %velocidad articular
5      %par´ametros del p´endulo
6      m1=3.88; %masa
7      lc1=0.081; %centro de masa
8      g=9.81; %constante de aceleraci´on gravitacional
9      I_r1=0.16; %momento de inercia del rotor
10     %momento de inercial total del p´endulo
11     I1=m1*lc1*lc1+I_r1;
12     %fricci´on del p´endulo
13     b1=0.16; %coeficiente de fricci´on viscosa
14     fc1=0.19; %fricci´on de Coulomb
15     fe1=0.2; %fricci´on de est´atica
16     %par aplicado al servomotor del p´endulo
17     tau1=sin(t);
18     %aceleraci´on articular del p´endulo
19     qpp1=(tau1-b1*qp1-fc1*sign(qp1)- fe1*(1-abs(sign(qp1)))-m1*g*lc1*sin(q1))
       /gamma1;
20     %vector de salida
21     xp=[ qp1 ;   %xp(1)=qp1=x(2) velocidad articular
22           qpp1] ; %xp(2)=qpp1 aceleraci´on articular
23 end
```

C´odigo Fuente 6.10 p´endulo_simu

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.
```

p´endulo_simu

```
1  clear; close all;
2  clc;
3  format short g
4  x0=[0; 0]; %condici´on inicial
5  ti=0; h=0.0025; tf = 5; %tiempo de simulaci´on
6  ts=ti:h:tf; %vector de tiempo
7  opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
8  %simulaci´on del p´endulo robot
9  [t, x]=ode45('pendulo',[ti:0.01:tf], x0,opciones);
10 %asignaci´on de posiciones y velocidades articulares
11 q1 =x(:,1); %Posici´on articular
12 qp1=x(:,2); %Velocidad articular
13 %cinem´atica directa del p´endulo
14 %posici´on de casa sobre el eje y0_
15 x_ef=0.45*sin(q1);
16 y_ef=-0.45*cos(q1);
17 %fen´omeno de fricci´on
18 b1=0.16; %coeficiente de fricci´on viscosa
19 fc1=0.19; %fricci´on de Coulomb
20 fe1=0.20; %fricci´on est´atica
21 friccion=b1*qp1+fc1*sign(qp1)+fe1*(1-abs(sign(qp1)));
22 %gr´aficas de cinem´atica directa, posici´on, velocidad, y fricci´on, respectivamente.
23 subplot(2,2,1); plot(x_ef,y_ef)
24 subplot(2,2,2); plot(t,q1)
25 subplot(2,2,3); plot(t,qp1)
26 subplot(2,2,3); plot(qp,friccion)
```

6.7 Robot de 2 gdl

Robots manipuladores de 2 gdl tienen amplias aplicaciones sobre el plano $x_0 - y_0$: pintado de objetos, ensamble, estibado, etc. Considere el robot manipulador de 2 gdl que se muestra en la figura 6.9.

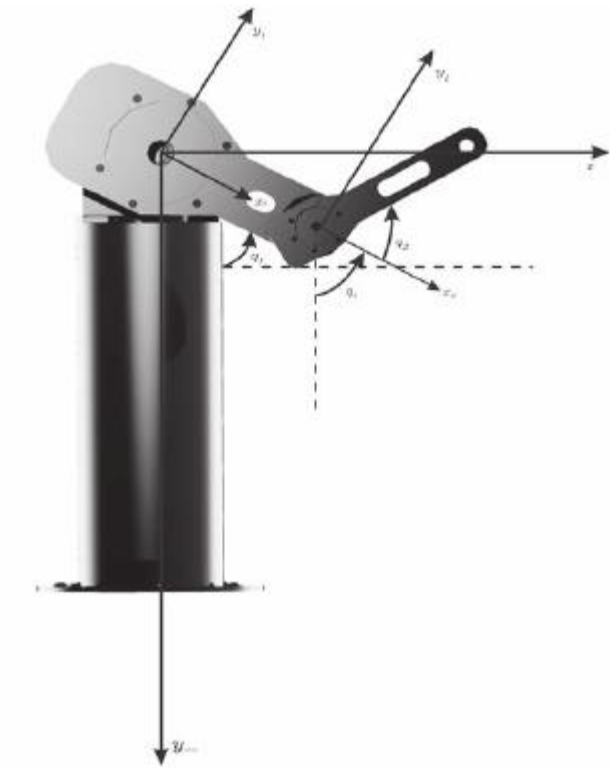


Figura 6.9 Robot manipulador de 2 gdl.

El modelo din´amico de un robot manipulador antropom´orfico de 2 gdl est´a dado por la siguiente expresi´on:

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}_f(\dot{\mathbf{q}}) + \mathbf{f}_e$$

$$= \underbrace{\begin{bmatrix} m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2}\cos(q_2) + I_1 + I_2 & m_2 l_{c2}^2 + m_2 l_1 l_{c2}\cos(q_2) + I_2 \\ m_2 l_{c2}^2 + m_2 l_1 l_{c2}\cos(q_2) + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix}}_{M(\mathbf{q})} \ddot{\mathbf{q}}$$

$$+ \underbrace{\begin{bmatrix} -2m_2 l_1 l_{c2} \operatorname{sen}(q_2)\dot{q}_2 & -m_2 l_1 l_{c2} \operatorname{sen}(q_2)\dot{q}_2 \\ m_2 l_1 l_{c2} \operatorname{sen}(q_2)\dot{q}_1 & 0 \end{bmatrix}}_{C(\mathbf{q},\dot{\mathbf{q}})} \dot{\mathbf{q}} +$$

$$g \underbrace{\begin{bmatrix} l_{c1} m_1 \operatorname{sen}(q_1) + m_2 l_1 \operatorname{sen}(q_1) + m_2 l_{c2} \operatorname{sen}(q_1 + q_2) \\ m_2 l_{c2} \operatorname{sen}(q_1 + q_2) \end{bmatrix}}_{\mathbf{g}(\mathbf{q})} +$$

$$\underbrace{\begin{bmatrix} b_1\dot{q}_1 + f_{c1}\operatorname{signo}(\dot{q}_1) + [1 - |\operatorname{signo}(\dot{q}_1)|]\operatorname{sat}(\tau_1; f_1) \\ b_2\dot{q}_2 + f_{c2}\operatorname{signo}(\dot{q}_2) + [1 - |\operatorname{signo}(\dot{q}_2)|]\operatorname{sat}(\tau_2; f_2) \end{bmatrix}}_{\mathbf{f}_f(\dot{\mathbf{q}},\mathbf{f}_e)}$$

La terminolog´ıa y significado de los par´ametros del robot de 2 gdl se encuentran en la tabla 6.4.

Tabla 6.4 Par´ametros del robot planar de 2 gdl

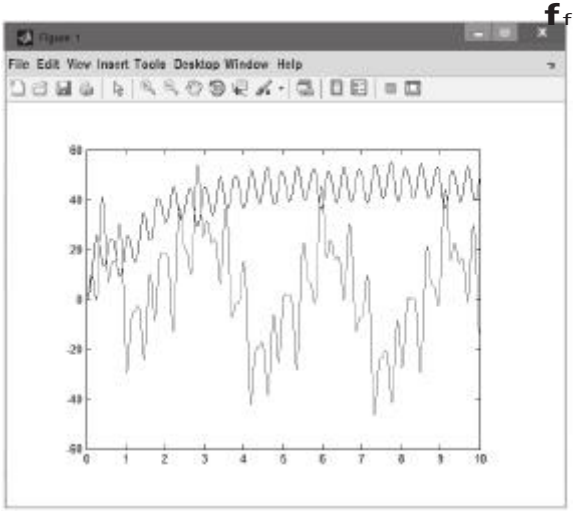
Eslab´on	Significado	Notaci´on	Valor
Hombro	Masa del eslab´on 1	m_1	23.902 kg
	Longitud del eslab´on 1	l_1	0.45 m
	Inercia del eslab´on 1	$\mathbf{I}_1$	1.266 Nm seg^2/rad
	Centro de masa del eslab´on 1	l_{c1}	0.091 m
	Coeficiente de fricci´on viscosa	b_1	2.288 Nm seg/rad
	Coeficiente de fricci´on de Coulomb	f_{c1}	7.17 Nm
	Coeficiente de fricci´on est´atica	f_{e1}	8.8 Nm
Codo	Masa del eslab´on 2	m_2	3.88 kg
	Longitud del eslab´on 2	l_2	0.45 m
	Inercia del eslab´on 2	$\mathbf{I}_2$	0.093 Nm seg^2/rad
	Centro de masa del eslab´on 2	l_{c2}	0.048
	Coeficiente de fricci´on viscosa	b_2	0.175 Nm seg/rad
	Coeficiente de fricci´on de Coulomb	f_{c2}	1.734 Nm
	Coeficiente de fricci´on est´atica	f_{e2}	1.87 Nm
	Aceleraci´on debida a la gravedad	g	9.81 m/seg^2

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \underbrace{\begin{bmatrix}2.351 + 0.1676\cos(q_2) & 0.102 + 0.0838\cos(q_2)\\ 0.102 + 0.0838\cos(q_2) & 0.102\end{bmatrix}}_{M(\mathbf{q})}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} +$$

$$\underbrace{\begin{bmatrix}-0.1676\,\mathrm{sen}(q_2)\dot{q}_2 & -0.0838\,\mathrm{sen}(q_2)\dot{q}_2\\ 0.084\,\mathrm{sen}(q_2)\dot{q}_1 & 0.0\end{bmatrix}}_{C(\mathbf{q},\dot{\mathbf{q}})}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} +$$

$$g\underbrace{\begin{bmatrix}3.9211\,\mathrm{sen}(q_1) + 0.1862\,\mathrm{sen}(q_1+q_2)\\ 0.1862\,\mathrm{sen}(q_1+q_2)\end{bmatrix}}_{\mathbf{g}(\mathbf{q})} +$$

$$\underbrace{\begin{bmatrix}2.288\dot{q}_1 + 7.17\,\mathrm{signo}(\dot{q}_1) + 8.8[1 - |\mathrm{signo}(\dot{q}_2)|]\\ 0.175\dot{q}_2 + 1.734\,\mathrm{signo}(\dot{q}_2)) + 1.87[1 - |\mathrm{signo}(\dot{q}_1)|]\end{bmatrix}}_{\mathbf{f}_f(\dot{\mathbf{q}},\mathbf{f}_e)}$$



La figura 6.10 muestra la respuesta de las posiciones articulares (q_1, q_2) del robot manipulador de 2 gdl en configuración antropomórfica cuando se le aplica senales de prueba del par aplicado (6.14). La sintonía de las amplitudes de estas senales para τ_1 y τ_2 se deben seleccionar de manera conveniente de tal forma que no se sature los servoamplificadores, y no producir velocidades articulares a 720 grados/seg.

Figura 6.10 Respuesta del robot de 2 gdl.

C´odigo Fuente 6.11 robot2gdl.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.
```

robot2gdl.m

```
function xp =robot2gdl(t,x)
    q1=x(1); q2=x(2); q = [q1; q2];%vector de posici´on articular
    qp1=x(3); qp2=x(4); qp = [qp1; qp2];%vector de velocidad articular
    m1=23.902; l1=0.45; lc1=0.091; I1=1.266;
    b1=2.288; fc1=7.17; fe1=8.8;
    m2=3.880; l2=0.45;
    lc2=0.048; I2=0.093;
    b2=0.175; fc2=1.734; fe2=1.87; g=9.81;
    m11=m1*lc1*lc1+m2*l1*l1+m2*lc2*lc2+2*m2*l1*lc2*cos(q2)+I1+I2;
    m12=m2*lc2*lc2+m2*l1*lc2*cos(q2)+I2;
    m21=m12;
    m22=m2*lc2*lc2+I2;
    M=[m11, m12; m21, m22];
    c11=-2*m2*l1*lc2*sin(q2)*qp2;
    c12=-m2*l1*lc2*sin(q2)*qp2;
    c21=m2*l1*lc2*sin(q2)*qp1;
    c22=0;
    C=[ c11, c12; c21, c22];
    gq11=(lc1*m1++m2*l1)*sin(q1)+m2*lc2*sin(q1+q2);
    gq21=m2*lc2*sin(q1+q2);
    gq=g*[gq11; gq21];
    fr=[b1*qp1+fc1*sign(qp1)+fe1*(1-abs(sign(qp1)));
        b2*qp2+fc2*sign(qp2)+fe2*(1-abs(sign(qp2)))];
    tau=[(1-exp(-0.8*t))*29.0+ 68*sin(16*t+0.1) + 9*sin(20*t+0.15);
       (1-exp(-1.8*t))*1.2+ 8*sin(26*t+0.08)+2*sin(12*t+0.34) ];
    qpp = M^(-1)*(tau-C*qp-gq-fr);
    xp = [qp1; qp2; qpp(1); qpp(2)];
end
```

Las señales de prueba para el par aplicado se encuentran expresadas por:

$$\tau_1 = [1 - e^{-0.8t}]\,29.0 + 68\,\text{sen}(16t + 0.1) + 9\,\text{sen}(20t + 0.15) \quad (6.14)$$

$$\tau_2 = [1 - e^{-1.8t}]\,1.2 + 8\,\text{sen}(26t + 0.08) + 2\,\text{sen}(12t + 0.34) \quad (6.15)$$

C´odigo Fuente 6.12 cap6_robot2gdlsimu.m

%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.

cap6_robot2gdlsimu.m

```
1  clc; clear all; close all;
2  format short
3  ti=0; h=0.0025; tf = 10; ts=ti:h:tf; %vector tiempo
4  %configuraci´on de la funci´on ode45()
5  opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
6  %condiciones iniciales [q(0), q'(0)] = [0, 0, 0, 0] T. %integraci´on num´erica de la
   din´amica del robot de 2 gdl
7  [t,x]=ode45('robot2gdl',ts,[0; 0; 0; 0],opciones);
8  %retorna x = [q1(t), q2(t), q1'(t), q2'](t)
9  %vector de posici´on articular
10 % q = [q1, q2]^T
11 q1=x(:,1);
12 q2=x(:,2);
13 %vector de velocidad articular
14 % q' = [q1', q2']^T
15 qp1=x(:,3);
16 qp2=x(:,4);
17 %grafica posiciones articulares en funci´on del tiempo
18 %conversi´on de radianes a grados
19 plot(t,180*q1/pi,t,180*q2/pi)
```

6.8 Robot de 3 gdl

El modelo din´amico de un robot de 3 gdl en configuraci´on antropom´orfica es una ecuaci´on diferencial muy complicada, el proceso de simulaci´on a trav´es de la funci´on ode45(···) puede consumir tiempo de c´omputo de manera significativa si se introducen funciones discontinuas como el fen´omeno de fricci´on de Coulomb. En este sentido, se recomienda al lector eliminar la fricci´on de Coulomb y la fricci´on est´atica, conservando la fricci´on viscosa para una r´apida simulaci´on. Evidentemente, la simulaci´on estar´ıa incompleta por falta de estos fen´omenos. Sin embargo, en un robot de transmisi´on directa las magnitudes de fricci´on son bajas.

La figura 6.11 muestra un robot industrial de la compan˜´ıa FANUC en configuraci´on antropom´orfica; los 3 gdl se refieren a las articulaciones de la base, hombro y codo, sin tomar en cuenta la orientaci´on del robot.

Figura 6.11 Robot manipulador FANUC.

La gran mayor´ıa de los robots que se encuentran operativos en el sector industrial tienen la configuraci´on antropom´orfica por ser la que mejor destreza presenta en sus movimientos y por lo tanto la que mayor n´umero de aplicaciones tiene. Este tipo de robots no s´olo se emplea en la industria, sino tambi´en es importante en

la automatizaci´on de quir´ofanos robotizados, asistencia personalizada a un sector vulnerable de la poblaci´on con capacidades diferenciadas, etc´etera.

La utilidad del modelo din´amico resulta importante cuando se dispone de los valores num´ericos de todos los par´ametros del robot; bajo este enfoque la tabla 6.5 muestra la terminolog´ıa, significado y valores de un robot prototipo de 3 gdl con motores de transmisi´on directa.

Tabla 6.5 Par´ametros del robot antropom´orfico de 3 gdl

Eslab´on	Significado	Notaci´on	Valor
Base	Masa del eslab´on 1	m_1	26.9 kg
	Longitud del eslab´on 1	l_1	0.45 m
	Inercia del eslab´on 1	$\mathbf{I}_{z1}$	1.266 Nm-seg^2/rad
	Inercia del eslab´on 1	$\mathbf{I}_{y1}$	0.089 Nm-seg^2/rad
	Inercia del eslab´on 1	$\mathbf{I}_{x1}$	0.03 Nm-seg^2/rad
	Centro de masa del eslab´on 1	l_{c1}	0.091 m
	Coeficiente de fricci´on viscosa	b_1	2.288 Nm-seg/rad
	Coeficiente de fricci´on de Coulomb	$\mathbf{f}_{c1}$	7.17 Nm
	Coeficiente de fricci´on est´atica	$\mathbf{f}_{e1}$	8.8 Nm
Hombro	Masa del eslab´on 2	m_2	30 kg
	Longitud del eslab´on 2	l_2	0.45 m
	Inercia del eslab´on 1	$\mathbf{I}_{z2}$	0.084 Nm-seg^2/rad
	Inercia del eslab´on 1	$\mathbf{I}_{y2}$	0.003 Nm-seg^2/rad
	Inercia del eslab´on 1	$\mathbf{I}_{x2}$	0.05 Nm-seg^2/rad
	Centro de masa del eslab´on 2	l_{c2}	0.038 m
	Coeficiente de fricci´on viscosa	b_2	0.2 Nm-seg/rad
	Coeficiente de fricci´on de Coulomb	$\mathbf{f}_{c2}$	1.9 Nm
	Coeficiente de fricci´on est´atica	$\mathbf{f}_{e2}$	2.1 Nm
Codo	Masa del eslab´on 3	m_3	3.88
	Longitud del eslab´on 2	l_3	0.45 m
	Inercia del eslab´on 1	$\mathbf{I}_{z3}$	0.056 Nm-seg^2/rad
	Inercia del eslab´on 1	$\mathbf{I}_{y3}$	0.0012 Nm-seg^2/rad
	Inercia del eslab´on 1	$\mathbf{I}_{x3}$	0.009 Nm-seg^2/rad
	Centro de masa del eslab´on 3	l_{c3}	0.048
	Coeficiente de fricci´on viscosa	b_3	0.175 Nm-seg/rad
	Coeficiente de fricci´on de Coulomb	$\mathbf{f}_{c3}$	1.734
	Coeficiente de fricci´on est´atica	$\mathbf{f}_{e3}$	1.87
	Aceleraci´on debida a la gravedad	g	9.81 m/seg^2

El modelo din´amico del robot manipulador en configuraci´on antropom´orfica se detalla en la ecuaci´on (6.16); de acuerdo con los datos de la tabla 6.5 se obtiene un modelo din´amico para prop´ositos de simulaci´on.

(6.16)

♣♣♣ Ejemplo 6.8

Emplear el modelo dinámico de un robot antropomórfico (6.16) de 3 gdl y los valores numéricos de los parámetros contenidos en la tabla 6.5 para realizar un programa en **MATLAB** que controle las articulaciones de la base, hombro y codo en las posiciones deseadas $[q_{d1}, q_{d2}, q_{d3}]^T = [45, 45, 90]^T$ grados, respectivamente. Presentar la evolución en el tiempo de las posiciones articulares.

Solución

Una de las aplicaciones que tiene el modelo dinámico de robots manipuladores es en control de posición; a través de la simulación es posible estudiar en detalle la respuesta que presenta el robot con una determinada estructura de control, sintonizar de manera adecuada las ganancias proporcional y derivativa, para obtener un transitorio pequeño sin sobre tiro y oscilaciones y reducir significativamente el error de posición en régimen estacionario.

El modelo dinámico del robot antropomórfico de 3 gdl junto con el algoritmo de control proporcional derivativo (PD) está implementado con la estructura de una ecuación diferencial ordinaria de primer orden $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ en la función `robot3gdl(t,x)` descrita en el cuadro 6.13; el programa principal para realizar la simulación se presenta en el cuadro 6.14.

La sintaxis de la función del modelo dinámico del robot antropomórfico de 3 gdl está dada como:

```
if   xp = robot3gdl(t, x)
```

donde t es la evolución del tiempo, $\mathbf{x}$ es la variable de estado, contiene las posiciones y velocidades articulares, $\dot{\mathbf{x}}$ es la derivada temporal de la variable de estado $\mathbf{x}$, contiene las velocidades y aceleraciones articulares del robot, es decir:

$$\mathbf{x} = [q_1(t) \quad q_2(t) \quad q_3(t) \quad \dot{q}_1(t) \quad \dot{q}_2(t) \quad \dot{q}_3(t)]^T$$
$$\dot{\mathbf{x}} = [q_1(t) \quad q_2(t) \quad q_3(t) \quad \ddot{q}_1(t) \quad \ddot{q}_2(t) \quad \ddot{q}_3(t)]^T$$

C´odigo Fuente 6.13 robot3gdl

%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Funci´on robot3gdl.m (robot3gdl simu.m)

robot3gdl

```
function xp = robot3gdl(t,x)
    q1=x(1); q2=x(2); q3=x(3); q = [q1; q2; q3];%vector de posici´on articular
    qp1=x(4); qp2=x(5); qp3=x(6); qp = [qp1; qp2; qp3];%vector de velocidad articular
    Iz1=1.26; Iz2=0.084; Iz3=0.056; Iy1=0.089; Iy2=0.003; Iy3=0.0012; Ix1=0.03; Ix2=0.05;
    Ix3=0.009; m1=26.902; l1=0.45; b1=2.288; fc1=7.17; fe1=8.8; m2=30; l2=0.45; lc2=0.038;
    b2=0.2; fc2=1.9; fe2=2.1; m3=3.880; l3=0.45; lc3=0.048; b3=0.175; fc3=1.734; fe3=1.87;
    m11=Iy2*sin(q2)*sin(q2)+Iy3*sin(q2+q3)*sin(q2+q3)+Iz1+Iz2*cos(q2)*cos(q2)
    +Iz3*cos(q2+q3)*cos(q2+q3)+m2*lc2*lc2*cos(q2)*cos(q2)+
    m3*(l2*cos(q2)+lc3*cos(q2+q3))*(l2*cos(q2)+lc3*cos(q2+q3));
    m12=0; m13=0; m21=0; m22=Ix2+Ix3+m3*l2*l2+m2*lc2*lc2+m3*lc3*lc3+
    2*m3*l2*lc3*cos(q3); m23=Ix3+m3*lc3*lc3+m3*l2*lc3*cos(q3);
    m31=0; m32=Ix3+m3*lc3*lc3+m3*l2*lc3*cos(q3); m33=Ix3+m3*lc3*lc3;
    M=[m11, m12, m13; m21, m22, m23; m31, m32, m33];
    gamma112=(Iy2-Ix2-m2*lc2*lc2)*cos(q2)*sin(q2)+(Iy3-Iz3)*cos(q2+q3)*sin(q2+q3)
    -m3*(l2*cos(q2)+lc3*cos(q2+q3))*(l2*sin(q2)+lc3*sin(q2+q3));
    gamma113=(Iy3-Iz3)*cos(q2+q3)*sin(q2+q3)-m3*lc3*sin(q2+q3)*
    (l2*cos(q2)+lc3*cos(q2+q3));
    gamma121=(Iy2-Iz2-m2*lc2*lc2)*cos(q2)*sin(q2)+(Iy3-Iz3)*cos(q2+q3)*sin(q2+q3)-
    m3*(l2*cos(q2)+lc3*cos(q2+q3))*(l2*sin(q2)+lc3*sin(q2+q3));
    gamma131=(Iy3-Iz3)*cos(q2+q3)*sin(q2+q3)-m3*lc3*sin(q2+q3)*
    (l2*cos(q2)+lc3*cos(q2+q3)); gamma211=(Ix2-Iy2+m2*lc2*lc2)*cos(q2)*sin(q2)+
    (Iz3-Iy3)*cos(q2+q3)*sin(q2+q3)+m3*(l2*cos(q2)+lc3*cos(q2+q3))*
    (l2*sin(q2)+lc3*sin(q2+q3)); gamma223=-l2*m3*lc3*sin(q3);
    gamma232=-l2*m3*lc3*sin(q3); gamma233=-l2*m3*lc3*sin(q3) ;
    gamma311=(Iz3-Iy3)*cos(q2+q3)*sin(q2+q3)+m3*lc3*sin(q2+q3)*(l2*cos(q2)+
    lc3*cos(q2+q3)); gamma322=l2*m3*lc3*sin(q3);
    c11=gamma112*qp2+gamma113*qp3; c12=gamma121*qp1; c13=gamma131*qp1;
    c21=gamma211*qp1; c22=gamma223*qp3; c23=gamma232*qp2+gamma233*qp3;
    c31=gamma311*qp1; c32=gamma322*qp2; c33=0; C=[c11, c12, c13; c21, c22, c23;
    c31, c32, c33]; gq11=0; gq21=(lc2*m1++m2*l2)*sin(q1)+m2*lc3*sin(q1+q2);
    gq31=m2*lc3*sin(q1+q2);g=9.81; gq=g*[gq11; gq21; gq31]; fr=[b1*qp1; b2*qp2; b3*qp3];
    qd=[45*pi/180; 45*pi/180; 90*pi/180]; qt=[qd(1)-q1;qd(2)-q2; qd(3)-q3];
    Kp=[5, 0,0;0, 5,0; 0,0,5]; Kv=[3, 0,0;0, 3,0;0,0,3]; tau=Kp*qt-Kv*qp+gq;
    qpp = inv(M)*(tau-C*qp-gq-fr);%aceleraci´on articular
    xp = [qp1; qp2; qp3; qpp(1); qpp(2); qpp(3)];%vector de salida
end
```

C´odigo Fuente 6.14 robot3gdl_simu

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa robot3gdl_simu.m
%requiere la funci´on robot3gdl.m
```

robot3gdl_simu

```
1  clear; close all; clc;
2  format short g
3  %par´ametros de simulaci´on:
4  ti=0; h=0.0025; tf = 10;%tiempo de simulaci´on 0 a 10 segundos
5  ts=ti:h:tf; %vector tiempo
6  %configuraci´on de la funci´on ode45
7  opciones=odeset('RelTol',1e-3,'InitialStep', 2.5e-3, 'MaxStep', 2.5e-3);
8  %soluci´on num´erica del sistema, con condici´on iniciales cero
9  [t,x]=ode45('robot3gdl',ts,[0; 0; 0; 0; 0; 0], opciones);
10 %gr´aficas de las 3 articulaciones; q1=x(:,1), q2 =x(:,2), q3 =x(:,3)
11 plot(t,180*x(:,1)/pi,t,180*x(:,2)/pi, t,180*x(:,3)/pi)
```

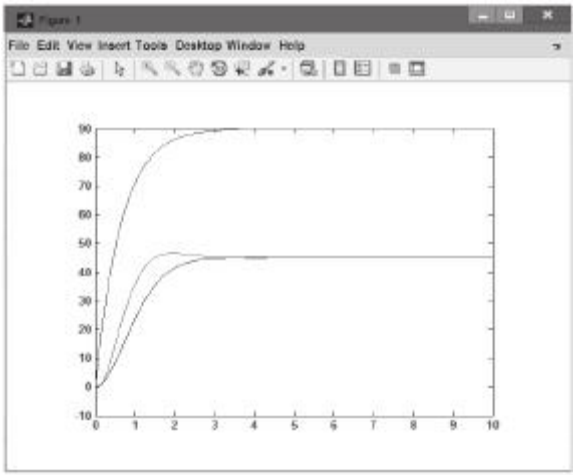


Figura 6.12 Respuesta del robot 3 gdl.

La figura 6.12 presenta la respuesta del robot a una entrada τ de la ley de control PD. Las posiciones de las articulaciones de la base, hombro y codo convergen a [45, 45, 90] grados, respectivamente. La sinton´ıa de las ganancias se hizo para que el transitorio no tuviera sobre impulsos y error en estado estacionario menor a 0.001 grados en las 3 articulaciones.

6.9 Robot cartesiano

El robot cartesiano de 3 gdl tiene los siguientes fen´omenos: inercial, fricci´on (viscosa, Coulomb y est´atica) y par gravitacional, debido a que la articulaci´on d_1 tiene desplazamiento en el eje z_0, su energ´ıa potencial es variable. Para las variables d_2 y d_3 tienen movimiento en el plano $x_0 - y_0$, entonces la energ´ıa potencial de esas variables es constante. Por lo tanto el par gravitacional para aquellas componentes es cero. Por otro lado, el robot cartesiano no tiene variables rotacionales q_i, mientras que las fuerzas centr´ıpetas y de Coriolis son cero.

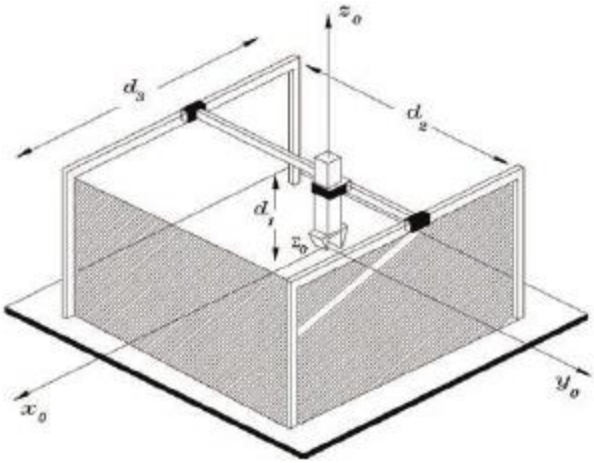


Figura 6.13 Robot cartesiano.

El modelo din´amico de un robot cartesiano de 3 gdl est´a dado por:

$$\boldsymbol{\tau} = \begin{bmatrix} m_1+m_2+m_3 & 0 & 0 \\ 0 & m_1+m_2 & 0 \\ 0 & 0 & m_1 \end{bmatrix} \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + \begin{bmatrix} m_1+m_2+m_3 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + g + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} signo(\dot{d}_1) \\ signo(\dot{d}_2) \\ signo(\dot{d}_3) \end{bmatrix} + \begin{bmatrix} f_{e1}[1-|signo(\dot{d}_1)|] \\ f_{e2}[1-|signo(\dot{d}_2)|] \\ f_{e3}[1-|signo(\dot{d}_3)|] \end{bmatrix} \tag{6.16}$$

donde m_1, m_2, m_3 son las masas de los servomotores e incluyen las barras y piezas met´alicas de acoplamiento mec´anico, g es la constante debido a la aceleraci´on gravitacional (g=9.81 m/seg^2), b_1, b_2, b_2, f_{c1}, f_{c2}, f_{c3} y f_{e1}, f_{e2}, f_{e3} representan los coeficientes de fricci´on viscosa, Coulomb y est´atica, respectivamente.

Los fen´omenos de fricci´on pueden ser de magnitud considerable si los servomotores emplean reductores de velocidad por medio de sistemas de engranes, los cuales introducen juego mec´anico que repercute directamente en errores de posici´on. Adem´as, debido al efecto disipativo de la fricci´on (conversi´on de energ´ıa mec´anica en energ´ıa t´ermica), se degradan las piezas o componentes del robot, lo que produce un pobre desempeño. Cuando la tecnolog´ıa de los servomotores del robot son de transmisi´on directa (direct drive), entonces el fen´omeno de fricci´on se reduce considerablemente, de ah´ı que la mayor´ıa de la bibliograf´ıa especializada de rob´otica presenta el modelo din´amico del robot cartesiano sin el fen´omeno de fricci´on; debido a este hecho el modelo din´amico resultante del robot cartesiano es lineal con respecto a las variables de estado d_1, d_2, d_3. Tambi´en debe recordarse, como se demostr´o en la secci´on 5.6 que el modelo cinem´atico cartesiano es lineal. Por ambas caracter´ıstica, al robot cartesiano tambi´en se le denomina **robot lineal**.

A continuaci´on se presenta el modelo din´amico en forma num´erica de un prototipo robot cartesiano; los valores num´ericos de los par´ametros se encuentran descritos en la tabla 6.6.

Tabla 6.6 Par´ametros del robot cartesiano

Articulaci´on	Masa	Fricci´on viscosa	Fricci´on de Coulomb	Fricci´on est´atica
τ_1^{max} =50 N, d_1	m_1=0.7 kg	b_1 =0.02 kg/seg	f_{c1} =0.01 N	f_{e1} =0.015 N
τ_2^{max} =4 N, d_2	m_2=0.28 kg	b_2 =0.08 kg/seg	f_{c2} =0.07 N	f_{e2} =0.076 N
τ_3^{max} =4 N, d_3	m_3=0.28 kg	b_3 =0.02 kg/seg	f_{c3} =0.02 N	f_{e3} =0.022 N

donde τ_i^{max} (con $i = 1, 2, 3$) representa las capacidades m´aximas de los servomotores de las articulaciones (d_1, d_2, d_3), respectivamente.

La mesa del robot cartesiano tiene una longitud de 1m por cada lado. La posici´on de casa $[d_1, d_2, d_3]^T = [0, 0, 0]^T$ est´a ubicada de manera conveniente sobre el piso, en el centro geom´etrico de la mesa. De esta forma, la variable articular d_1 que se desplaza sobre el eje z_0 tiene coordenadas positivas y acotada en el intervalo $0 \leq d_1 \leq 1$, para las variables d_2, d_3 que se mueven en el plano $x_0 - y_0$ pueden tener coordenadas positivas y negativas; ambas variables se encuentran restringidas al intervalo: $-0.5 \leq d_i \leq 0.5$, para $i = 2, 3$. Tambi´en es posible ubicar otro lugar

para la posici´on de casa, lo cual depende finalmente de la facilidad que permita programar la aplicaci´on que va a realizar el robot.

$$\boldsymbol{\tau} = \begin{bmatrix} 1.26 & 0 & 0 \\ 0 & 0.98 & 0 \\ 0 & 0 & 0.7 \end{bmatrix} \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + \begin{bmatrix} 0.02 & 0 & 0 \\ 0 & 0.08 & 0 \\ 0 & 0 & 0.02 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + \begin{bmatrix} 12.36 \\ 0 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.07 & 0 \\ 0 & 0 & 0.02 \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{d}_1) \\ \text{signo}(\dot{d}_2) \\ \text{signo}(\dot{d}_3) \end{bmatrix} + \begin{bmatrix} 0.015[1 - |\text{signo}(\dot{d}_1)|] \\ 0.076[1 - |\text{signo}(\dot{d}_2)|] \\ 0.022[1 - |\text{signo}(\dot{d}_3)|] \end{bmatrix} \tag{6.17}$$

En el caso del robot cartesiano debe notarse que en el modelo din´amico (6.17) la entrada $\boldsymbol{\tau}$ representa una fuerza cartesiana y no un par aplicado o torque.

Cuando reci´en se construye o se adquiere un prototipo nuevo es recomendable aplicarle una señal de energ´ıa apropiada para moverlo en su espacio de trabajo por un periodo de una semana (d´ıa y noche). Lo anterior tiene la finalidad de "asentar" los servomotores y bajar los niveles de fricci´on a sus estados normales.

Es conveniente aplicar una señal de entrada al robot como la que se presenta a continuaci´on:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 12.36 + 0.05\,\text{sen}(t) \\ 0.13[1 - e^{-0.5t}]\,\text{sen}(t) \\ 0.06\,\text{sen}(t) \end{bmatrix} \tag{6.18}$$

donde t es la variable del tiempo; las amplitudes y par´ametros son seleccionados de manera conveniente para no saturar a los servoamplificadores del robot.

El modelo din´amico (6.17) es una ecuaci´on diferencial de segundo orden; para poder simular dicho modelo es necesario convertirlo al formato $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, es decir a una ecuaci´on diferencial ordinaria de primer orden y de esta forma estar en condiciones de utilizar la funci´on de **MATLAB** ode45(···). Con esta finalidad, observe que $\mathbf{x} = [d_1, d_2, d_3, \dot{d}_1, \dot{d}_2, \dot{d}_3]$, entonces la ecuaci´on (6.17) adquiere la forma (6.19):

Una vez que el modelo din´amico ya se encuentra correctamente expresado como una ecuaci´on diferencial ordinaria de primer orden, puede ser implementado en lenguaje **MATLAB** a trav´es de la funci´on ode45(···).

En el cuadro 6.15 se encuentra el c´odigo fuente del modelo din´amico del robot cartesiano. Se ha tomado la entrada de prueba (6.18) para estudiar su comportamiento din´amico. El cuadro 6.16 contiene el programa principal para realizar la simulaci´on con el robot cartesiano. La figura 6.14 muestra el desplazamiento articular de las tres variables d_1, d_2, d_3. La oscilaci´on que presenta cada articulaci´on es intencional con la finalidad de mover a cada servo en ambas direcciones y al mismo tiempo recorran la mayor parte de su espacio de trabajo. Es recomendable este procedimiento antes de utilizar el robot, al menos durante una semana dejarlo con la trayectoria de prueba ayuda a normalizar los niveles de fricci´on. Posteriormente se puede realizar identificaci´on param´etrica (ver secci´on 7.7) o realizar experimentos con el robot.

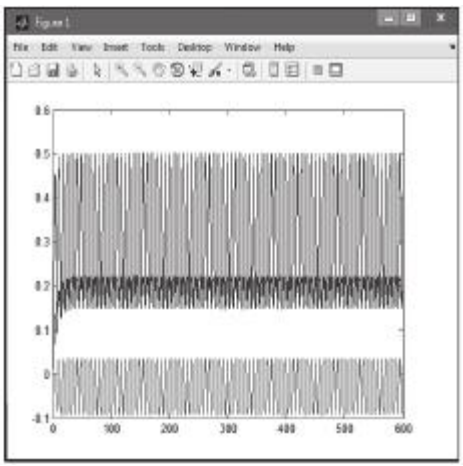


Figura 6.14 Robot cartesiano.

(6.19)

C´odigo Fuente 6.15 robot_cartesiano3gdl

%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa robot_cartesiano3gdl.m

robot_cartesiano3gdl

```
function xp = robot_cartesiano3gdl(t,x)
    d = [x(1); x(2); x(3)]; % Vector de posici´on articular
    dp = [x(4); x(5); x(6)]; % vector de velocidad articular
    %par´ametros del robot cartesiano
    m1=0.7; m2=0.28; m3=0.28;% masas de los servomotores y partes mec´anicas
    b1=0.02; b2=0.08; b3=.02;% coeficientes de fricci´on viscosa
    fc1=0.01; fc2=0.07; fc3=.02;% coeficientes de la fricci´on de Coulomb
    fe1=0.015; fe2=0.076; fe3=0.022;% coeficientes de la fricci´on est´atica
    g=9.81;% constante de aceleraci´on gravitacional
    B=[b1, 0, 0;
          0, b2,0;
          0, 0, b3];
    Fc=[fc1, 0, 0;
          0, fc2,0;
          0, 0, fc3];
    % matriz de masas
    M = [m1+m2+m3, 0, 0;
          0 m1+m2, 0;
          0 0 m3];
    par_grav = g*[m1+m2+m3; 0;0];%vector de par de gravitacional
    %fricci´on est´atica
    fe=[ fe1*(1-abs(sign(dp(1)))); fe2*(1-abs(sign(dp(2))));
    fe3*(1-abs(sign(dp(3))))];
    fr= B*dp+Fc*sign(dp)+fe;
    tau=[ g*(m1+m2+m3)+ 0.05*sin(t); (1-exp(-0.5*t))*0.13*sin(t);0.06*sin(t)];
    dpp = inv(M)*(tau- par_grav-fr);
    xp = [dp(1); dp(2); dp(3); dpp(1); dpp(2); dpp(3)];
end
```

C´odigo Fuente 6.16 simu_robotcartesiano3gdl

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es
%Cap´ıtulo 4 Din´amica. Programa simu_robotcartesiano3gdl.m
% este programa requiere del archivo: robot_cartesiano3gdl.m
```

simu_robotcartesiano3gdl

```
1  clc;
2  clear all;
3  close all;
4  format short
5  %par´ametros de simulaci´on:
6  ti=0; %tiempo inicial
7  h=0.0025; %incremento del tiempo
8  tf = 600; % tiempo de simulaci´on (segundos)
9  ts=ti:h:tf; %vector de tiempo de simulaci´on
10 %para un correcto desempeño de integraci´on de la funci´on ode45(...) se requieren:
11 opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
12 % soluci´on num´erica del sistema x˙ = f(x):
13 %las condiciones iniciales del robot cartesiano han sido puestas en cero
14 %[d1(0), d2(0), d3(0), d˙1(0), d˙2(0), d˙3(0)]^T = [0, 0, 0, 0, 0, 0]^T
15 %advertencia: debido al uso de funciones discontinuas en el modelo din´amico del
16 %robot cartesiano como la funci´on signo(...), el proceso de simulaci´on puede
   tardar
17 [t,x]=ode45('robot_cartesiano3gdl',ts,[0; 0; 0; 0; 0; 0],opciones);
18 % t representa el tiempo, x representa la soluci´on de la ecuaci´on (6.19)
19 % Las componentes del vector x representan
20 % componentes de posiciones
21 %x(:, 1) = d1  x(:, 2) = d2  x(:, 3) = d3
22 % componentes de velocidades
23 %x(:, 4) = d˙1  x(:, 5) = d˙2  x(:, 6) = d˙3
24 %presenta gr´afica de posiciones en funci´on del tiempo
25 plot(t,x(:,1),t,x(:,2),t,x(:,3))
```

6.10 Resumen

Dinámica es la parte de la f´ısica que explica el movimiento de un sistema mec´anico tomando en cuenta las fuerzas que lo producen. El modelo din´amico est´a formado por ecuaciones diferenciales no lineales que reproducen ´ıntegramente todos los fen´omenos f´ısicos del robot.

En este cap´ıtulo se ha presentado la t´ecnica para simular diversos sistemas din´amicos, lo que facilita el estudio, an´alisis y aplicaciones del sistema o prototipo. Para un adecuado entendimiento de la din´amica del sistema la mejor herramienta es la simulaci´on. El paso previo a una etapa experimental es la simulaci´on, ya que conocer a detalle todos los aspectos din´amicos del robot o del sistema mecatr´onico facilita su control, as´ı como llevar a cabo en forma pr´actica aplicaciones potenciales del prototipo. El m´etodo de integraci´on num´erica es un factor fundamental para una adecuada soluci´on num´erica del modelo din´amico, por lo tanto es recomendable utilizar el m´etodo de Runge Kutta 4/5, el cual lo tiene implementado **MATLAB** a trav´es de la funci´on ode45(···).

Se han presentado diversos ejemplos para ilustrar la t´ecnica de simulaci´on de sistemas din´amicos: sistemas lineales y no lineales simples, sistema masa resorte amortiguador, centr´ıfuga y p´endulo, hasta sistemas din´amicos de robots manipuladores m´as complejos como el robot antropom´orfico de 2 y 3 gdl, y robot cartesiano de 3 gdl. Dichos modelos se encuentran estructurados de acuerdo con la forma: function $\dot{\mathbf{x}}$=modelo_dinamico(t, $\mathbf{x}$) para f´acil implementaci´on y uso en diversas aplicaciones de rob´otica y mecatr´onica.

Es importante resaltar que la estructura matem´atica del modelo din´amico debe ser de la forma tradicional de variables fase como una ecuaci´on diferencial ordinaria de primer orden:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

si el modelo original del sistema es una ecuaci´on diferencial de orden superior, entonces mediante un adecuado cambio de variables de estados siempre es posible transformarlo a la estructura requerida.

A continuación se resume la sintaxis de los modelos dinámicos desarrollados para **MATLAB**:

```
xp = centrifuga( t,x)
```

```
xp =pendulo(t,x)
```

```
xp =mra(t,x)
```

```
xp =robot2gdl(t,x)
```

```
xp=robot3gdl(t, x)
```

```
xp = robot_cartesiano3gdl(t,x)
```

Para un adecuado funcionamiento de las librerías es recomendable utilizar en todos los casos un código parecido al siguiente:

```
ti=0; h=0.0025; tf =5; ts=ti:h:tf;
```

```
opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
```

Condiciones iniciales (dependiendo del robot):

ci=[0;0] (centrífuga); ci=[0;0] (péndulo); ci=[0;0;0;0] robot de 2 gdl; ci=[0;0;0;0;0;0] robot de 3 gdl (antropomórfico y cartesiano).

```
[t,x]=ode45('nombre funcion',ts,ci,opciones);
```

Capítulo 7

Identificación paramétrica

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \frac{P(k-1)\,\Psi(k)\,[\mathbf{y}(k) - \Psi(k)^{T}\,\hat{\boldsymbol{\theta}}(k-1)]}{1 + \Psi(k-1)^{T} P(k-1)\Psi(k)}$$

$$P(k) = P(k-1) - \frac{P(k-1)\Psi(k)\Psi(k)^{T} P(k-1)}{1 + \Psi(k-1)^{T} P(k-1)\Psi(k)}$$

Objetivos

Presentar la t´ecnica de estimaci´on de m´ınimos cuadrados y su aplicaci´on en identificaci´on param´etrica de sistemas mecatr´onicos y robots manipuladores.

Objetivos particulares:

- Esquemas de regresi´on escalar y multivariable.
- Regresi´on lineal del modelo din´amico.
- Regresi´on del modelo de energ´ıa.
- Esquema de regresi´on del modelo de potencia.

7.1 Introducci´on

El modelo din´amico de sistemas mecatr´onicos y robots manipuladores contiene en su estructura matem´atica par´ametros tales como centros de gravedad, masas, momentos de inercia y coeficientes de fricci´on. Estos par´ametros generalmente son desconocidos; este es el caso de la mayor´ıa de los robots comerciales donde el fabricante no proporciona sus valores nominales. Si bien existen herramientas de la teor´ıa de control como esquemas adaptables y controladores robustos que permiten tolerar errores en los par´ametros din´amicos, el conocimiento de ´estos es crucial para la mayor´ıa de los esquemas basados en el modelo din´amico del robot manipulador.

Identificaci´on param´etrica es una herramienta atractiva para determinar los par´ametros din´amicos de robots manipuladores, sobre todo cuando existe dificultad para medirlos directamente. Sin embargo, la naturaleza no lineal del modelo din´amico de robots manipuladores hace que la tarea de identificaci´on param´etrica no sea trivial.

El modelo din´amico de robots manipuladores posee propiedades que permiten generar varios esquemas de regresi´on lineal como los siguientes: modelo din´amico se le denomina as´ı debido a que emplea la estructura del modelo din´amico para expresarlo como un regresor lineal vectorial. Sin embargo, este modelo requiere medir la aceleraci´on articular lo que representa una desventaja pr´actica. La estimaci´on de la velocidad puede realizarse mediante t´ecnicas de filtrado, lo que da origen al modelo de regresi´on din´amico filtrado. Por otro lado, existen esquemas de regresi´on escalares como los modelos de energ´ıa, potencia y potencia filtrada; estos modelos permiten identificar los mismos par´ametros de los esquemas vectoriales din´amico y din´amico filtrado.

Para poder utilizar el m´etodo de m´ınimos cuadrados se requiere que el modelo matem´atico del sistema a identificar pueda ser expresado como un regresor lineal en los par´ametros desconocidos. La t´ecnica de m´ınimos cuadrados recursivo permite obtener el valor num´erico de los par´ametros del sistema sin la necesidad de desarmar al sistema mecatr´onico o robot manipulador. Se requieren de contar con las mediciones u observaciones de las posiciones y velocidades articulares, as´ı como

la señal del par aplicado para que el sistema pueda moverse en su espacio de trabajo.

7.2 M´etodo de m´ınimos cuadrados

De particular relevancia para los esquemas de identificaci´on param´etrica y de control adaptable es expresar el modelo din´amico no lineal del robot manipulador como el producto de una matriz de regresi´on compuesta de funciones no lineales (dependientes de la posici´on, velocidad y aceleraci´on articular) y un vector de par´ametros constantes dependiente de masas, momentos de inercias, distancias a centros de masa y coeficientes de fricci´on.

Los esquemas de identificaci´on que se describir´an en esta secci´on son sistemas de identificaci´on que pertenecen a la filosof´ıa de identificaci´on h´ıbrida, es decir el modelo de regresi´on es formulado en tiempo continuo mientras que la identificaci´on se realiza a trav´es de un estimador recursivo llamado m´ınimos cuadrados, el cual se utiliza ampliamente en la literatura debido a su sencillez y a su propiedad de recursividad, atributo que lo hace atractivo para su implementaci´on.

Esta t´ecnica es particularmente simple si el modelo tiene la propiedad de linealidad en los par´ametros del modelo. El m´etodo de m´ınimos cuadrados es un esquema est´andar que aproxima la soluci´on de sistemas sobre determinados, por ejemplo cuando hay m´as variables inc´ognitas que ecuaciones. M´ınimos cuadrados minimiza la suma de cuadrados de los errores; un error es la diferencia entre un valor observado y el valor proporcionado por el modelo matem´atico (robot).

El algoritmo recursivo de m´ınimos cuadrados se describe a continuaci´on. Consid´erese el siguiente modelo de regresi´on:

Linealidad en los par´ametros

Considere el modelo matem´atico que est´a expresado como un regresor lineal de la siguiente forma:

$$\mathbf{y}(k) = \Psi(k)^{T}\boldsymbol{\theta} \tag{7.1}$$

donde $\mathbf{y}(k) \in IR^r$ es un vector de mediciones (entradas o salidas) del sistema, $\Psi(k) \in IR^{m\times r}$ es la matriz de regresi´on compuesta por observaciones de funciones conocidas y el vector de par´ametros desconocidos est´a representado por $\boldsymbol{\theta} \in IR^m$. El modelo (7.1) est´a indexado por la variable k, la cual denota el tiempo discreto; se asume que el conjunto de ´indices $\boldsymbol{\tau}(k)$, $\Psi(k)$ forman un conjunto discreto.

El algoritmo de m´ınimos cuadrados para el caso vectorial tiene la siguiente forma:

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + P(k-1)\,\Psi(k)[\mathbf{I} + \Psi(k-1)^T P(k-1)\Psi(k)]^{-1}\,\mathbf{e}(k) \tag{7.2}$$

$$P(k) = P(k-1) - P(k-1)\Psi(k)[\mathbf{I} + \Psi(k)^T P(k-1)\Psi(k)]^{-1}\Psi(k)^T P(k-1) \tag{7.3}$$

$$\mathbf{e}(k) = \mathbf{y}(k) - \Psi(k)^T\hat{\boldsymbol{\theta}}(k-1) \tag{7.4}$$

donde $P(k) \in IR^{m\times m}$ es la matriz de covarianza, la cual es una matriz definida positiva, $\mathbf{e}(k) \in IR^r$ es el error de predicci´on, $\hat{\boldsymbol{\theta}}(k)$ es el vector de par´ametros estimados; $\boldsymbol{\theta}$ es el vector de par´ametros reales.

El algoritmo de m´ınimos cuadrados para el caso escalar tiene la siguiente forma:

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \frac{P(k-1)\,\Psi(k)\mathbf{e}(k)}{}$$

$$P(k) = P(k-1) - \frac{P(k-1)\Psi(k)\Psi(k)^T P(k-1)}{1+\Psi(k-1)^T P(k-1)\Psi(k)} \tag{7.6}$$

$$\mathbf{e}(k) = \mathbf{y}(k) - \Psi(k)^T\hat{\boldsymbol{\theta}}(k-1) \tag{7.7}$$

Propiedades: El algoritmo de m´ınimos cuadrados puede identificar los par´ametros de cualquier modelo matem´atico lineal, no lineal, din´amico, est´atico, continuo o discreto (no depende del periodo de muestreo, inclusive puede ser aperi´odico). La estructura matem´atica del modelo debe cumplir con la condici´on de linealidad en los par´ametros, es decir que pueda ser expresado como un regresor lineal manteniendo la

estructura de una matriz de observaciones o mediciones y un vector de par´ametros.

7.3 Librer´ıa de m´ınimos cuadrados

La sintaxis de la funci´on de m´ınimos cuadrados var´ıa dependiendo de que el modelo sea expresado en forma escalar o multivariable (vectorial) de la siguiente forma:

Caso escalar

Para el caso de modelos matem´aticos con estructura escalar

$$\mathbf{y}(k) = \Psi(k)^T \hat{\boldsymbol{\theta}}(k-1)$$

la sintaxis de la funci´on de m´ınimos cuadrados tiene la siguiente forma:

```
function [r, θ̂] =mincuad(y,fi)
```

donde $\mathbf{y}(k) \in IR$ es un escalar definido como arreglo con n renglones con el registro de las correspondientes n mediciones del sistema; indexado por el pivote k para $k = 1, 2, \cdots, n$; $fi \in IR^{n \times m}$ es la matriz de regresi´on compuesta por n renglones de observaciones de funciones conocidas y m columnas (m indica el n´umero de par´ametros). Esta funci´on retorna el vector de par´ametros estimados $\hat{\boldsymbol{\theta}} \in IR^m$ y $\mathbf{r}$ es una matriz de dimensi´on $n \times m$ que contiene el registro temporal de las componentes $\hat{\theta}_i$ del vector de par´ametros estimados $\hat{\boldsymbol{\theta}}$ que tuvieron durante el proceso de identificaci´on.

En el cuadro 7.1 se presenta el c´odigo fuente del algoritmo de m´ınimos cuadrados en su versi´on escalar. En la l´ınea 2 se obtiene el n´umero de observaciones o mediciones (n= n´umero de renglones) que contiene cada columna de la matriz de regresi´on **fi** (m= n´umero de columnas). La l´ınea 3 genera el vector de par´ametros estimados $\hat{\boldsymbol{\theta}}$, con condici´on inicial cero $\hat{\boldsymbol{\theta}}(0)$ (l´ınea 4). En realidad, la condici´on inicial puede ser puesta en cualquier valor. La l´ınea 5 contiene la formaci´on del vector columna $\boldsymbol{\psi}$ con

las observaciones o mediciones. La matriz de covarianza $P \in \mathbb{R}^{m \times m}$ se genera en la l´ınea 6; observe que la condici´on inicial $P(0)$ es una matriz diagonal con valores $p_{ii} = 10^{20}$, $i = 1, 2 \cdots, m$.

De las l´ıneas 8 -14 se realiza el algoritmo recursivo de identificaci´on param´etrica de m´ınimos cuadrados de acuerdo a las ecuaciones (7.5)-(7.7). El error de predicci´on $e(k)$ se calcula en la l´ınea 12, el vector de par´ametros $\hat{\boldsymbol{\theta}}$ en la l´ınea 13, mientras que la matriz de covarianza P en la l´ınea 14. La forma como var´ıan las componentes $\hat{\theta}_i$ del vector de par´ametros estimados $\hat{\boldsymbol{\theta}}$ se registra en la l´ınea 16. Esta informaci´on es muy útil para analizar a trav´es de una gr´afica la convergencia param´etrica de cada componente $\hat{\theta}_i$ en funci´on del tiempo t.

C´odigo Fuente 7.1 mincuad.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.
```

mincuad.m

```
1  function [r, theta] =mincuad(y,fi)
2      [n,m]=size(fi); %n=n´umero de renglones, m=n´umero de columnas
3      theta=[1:m]'; %vector columna de par´ametros
4      theta(1)=0;%Condici´on inicial del vector de par´ametros
5      psi=[1:m]'; %vector columna de observaciones
6      P=eye(m,m)*10e20; %matriz de covarianza P
7      r=eye(n,m);%registro para los par´ametros estimados
8      for k=1:n % algoritmo recursivo de m´ınimos cuadrados
9          for i=1:m %se forma el regresor
10             psi(i,1)=fi(k,i);
11         end
12         e=y(k)-theta'*psi;%error de regresi´on
13         theta= theta+(P*psi*e)/(1+psi'*P*psi); %vector estimado
14         P=P-(P*psi*(psi')*P)/(1+psi'*P*psi); %matriz de covariancia
15         for i=1:m
16             r(k,i)=theta(i,1);%registro por cada iteraci´on de par´ametros estimados
17         end
18     end
19 end
```

Caso multivariable

Cuando el sistema est´a expresado como multivariable, es decir tiene n entradas o salidas $y_1, y_2, \cdots, y_n$ bajo la siguiente estructura matem´atica:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_n(k) \end{bmatrix} = \Psi(k)^T \hat{\boldsymbol{\theta}}(k-1)$$

entonces la librer´ıa del algoritmo de m´ınimos cuadrados recursivo correspondiente se denomina mincuadm teniendo la sintaxis que a continuaci´on se describe:

```
function θ̂ =mincuadm(y,fi,Nob,p,n)
```

donde y es un vector de dimensi´on $n \times Nob$ con las mediciones u observaciones de las entradas (salidas) del sistema a identificar; n indica el n´umero de variables $y_1, y_2, \cdots, y_n$; Nob representa el n´umero total de observaciones. Cada variable y_i tiene un conjunto de Nob observaciones o renglones, para $i = 1, \cdots n$; **fi** es el regresor de observaciones compuesto por p columnas (p= n´umero de par´ametros). Cada columna del regresor **fi** tiene un n´umero Nob de observaciones; cada observaci´on evidentemente est´a registrada en su respectivo renglón.

En la l´ınea 2 se genera el vector de par´ametros estimados $\hat{\boldsymbol{\theta}}$. La condici´on inicial de este vector es cero. Sin embargo, puede ser inicializado con cualquier valor (l´ınea 3). La matriz de covarianza $P \in IR^{p \times p}$ se define en la l´ınea 21. Note que el valor inicial de esta matriz $P(0)$ corresponde a una matriz diagonal con valores 10^{20} (ver l´ınea 5). La matriz identidad con las dimensiones adecuadas se define en la l´ınea 6. El vector de observaciones $\mathbf{y} \in IR^n$ para entradas (salidas) del sistema se define en la l´ınea 7.

El algoritmo de m´ınimos cuadrados recursivo multivariable de las ecuaciones (7.2)-(7.4) se implementa de las l´ıneas 8 a la 21. Observe que en la l´ınea 14 se forma el vector $\mathbf{y} \in IR^n$, de manera recursiva a cada renglón se le asigna la correspondiente medici´on u observaci´on.

En la l´ınea 17 se calcula el vector de error de predicci´on $\mathbf{e}(k)$, en la l´ınea 18 el vector de par´ametros estimados $\hat{\boldsymbol{\theta}}$ y en la l´ınea 19 la matriz de covarianza P.

La funci´on mincuadm retorna el vector de par´ametros estimados $\hat{\boldsymbol{\theta}} \in IR^p$. En el cuadro 7.2 se presenta el c´odigo fuente de la funci´on minicuad.m (caso multivariable).

C´odigo Fuente 7.2 mincuadm.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.
```

mincuadm.m

```
1  function theta =mincuadm(y,fi,Nob,p,n)
2      theta=[1:p]'; %vector columna de par´ametros
3      theta(1)=0; %condici´on inicial del vector de par´ametros
4      psi=zeros(p,n); %vector columna de observaciones
5      P=eye(p,p)*10e20; %matriz de covarianza P
6      I=eye(n,n); %matriz identidad
7      ys=zeros(n,1);
8      for k=1:Nob
9          for j=1:n
10             for i=1:p %se forma el regresor
11                 psi(i,j)=fi(k+Nob*(j-1),i);
12             end
13             for i=1:n
14                 ys(i,1)=y(k+Nob*(i-1));
15             end
16         end
17         e=ys-psi'*theta; %error de regresi´on
18         theta= theta+P*psi*(I+psi'*P*psi)^(-1)*e; %vector estimado
19         P=P-(P*psi*(I+psi'*P*psi)^(-1)*(psi')*P); %Matriz de covarianza
20     end
21 end
```

7.4 Ejemplos

En esta secci´on se presentan ejemplos sencillos que ilustran la forma de utilizar las librer´ıas de m´ınimos cuadrados recursivo para el caso escalar mincuad y multivariable mincuadm.

♣ Ejemplo 7.1

Identificar el par´ametro a_1 del siguiente modelo:

$$y = a_1 t^3$$

donde el valor te´orico de a_1=8.034.

Soluci´on

El sistema $y = a_1 t^3$ corresponde al caso escalar. Para prop´ositos de simulaci´on la variable t estar´a definida en el intervalo de 0 a 10 segundos, con incrementos de un milisegundo.

En este caso el regresor est´a formado por

$$y(k) = \Psi(k)^T \theta$$

donde $y = y(k)$, $\Psi(k) = t^3$, $\theta = a_1$.

Por lo tanto, el error de predicci´on $e(k)$ est´a dado por:

$$e(k) = \underset{\Psi(k)}{t^3}\ \hat{\theta}$$

En el cuadro 7.3 se muestra el c´odigo del programa cap7_ejemplo1.m; en la l´ınea 7 se forma la ecuaci´on del sistema a identificar, el vector de mediciones o regresor est´a definido en la l´ınea 8. El algoritmo de m´ınimos cuadrados se emplea en la l´ınea 10; el resultado se presenta en la l´ınea 11.

El resultado del programa 7.3 es:

theta=

8.034

C´odigo Fuente 7.3 cap7_ejemplo1.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.
```

cap7_ejemplo1.m

```
1  clc;
2  clear all;
3  close all;
4  format short g
5  t=[0:0.001:10]';%vector columna de tiempo
6  a1=8.034;%par´ametro del sistema
7  y=a1*t.*t.*t; %sistema a identificar
8  fi=[t.*t.*t ] ;
9  [m, p]=size(fi);
10 [~ theta] =mincuad(y,fi);
11 theta %resultado de la identificaci´on
```

♣ Ejemplo 7.2

Identificar los par´ametros del siguiente sistema:

$$y = a_1 t + a_2 \operatorname{sen}(t) + a_3 \cos(t) + a_4 \tanh(t)$$

donde los valores de los par´ametros son: $[a_1, a_2, a_3, a_4]^T = [4.456, 7.456, -0.089, 12.37]^T$

Soluci´on

El error de predicci´on se encuentra dado por

$$e(k) = y(k) - \underbrace{[t \quad \text{sen}(t) \quad \cos(t) \quad \tanh(t)]}_{\Psi(k)} \hat{\boldsymbol{\theta}}(k-1)$$

donde $\boldsymbol{\theta} = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{a}_3 \quad \hat{a}_4]^T$.

Para prop´ositos de simulaci´on, la variable tiempo t estar´a definida en un intervalo de 0 a 10 segundos, con incrementos de un milisegundo. El cuadro 7.4 contiene el programa cap7_ejemplo2.m; en la l´ınea 5 se define el sistema a identificar. El vector de observaciones o regresor se encuentra construido en la l´ınea 6 y en la l´ınea 7 se emplea la funci´on de m´ınimos cuadrados para obtener el vector de par´ametros $\hat{\boldsymbol{\theta}}$ en la l´ınea 8.

El resultado del programa cap7_ejemplo2.m es el siguiente:

theta=

$$[4.456 \quad 7.456 \quad -0.089 \quad 12.37]^T$$

C´odigo Fuente 7.4 cap7_ejemplo2.m

```
%MATLAB Aplicado a Rob´otica y Mecatr´onica.
%Editorial Alfaomega, Fernando Reyes Cort´es.
%Cap´ıtulo 7 Identificaci´on param´etrica.
```

cap7_ejemplo2.m

```
1 clc; clear all; close all; format short g
2 t=[0:0.001:10]';%vector columna de tiempo
3 %par´ametros del sistema
4 a1=4.456; a2=7.456; a3=-0.089; a4=12.37;%sistema a identificar
5 y=a1*t+a2*sin(t)+a3*cos(t)+a4*tanh(t);
6 fi=[t, sin(t), cos(t), tanh(t)];
7 [~ theta] =mincuad(y,fi);
8 theta %resultado de la identificaci´on
```

♣ Ejemplo 7.3

Identificar los par´ametros del siguiente sistema:

$$y(t) = a_1 \cosh(t) + a_2 \operatorname{senh}(t) + a_3 \cos(t)$$

donde los valores de los par´ametros son:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0.001 \\ 456 \\ -345 \end{bmatrix}$$

Soluci´on

El error de regresi´on est´a formado de la siguiente forma:

$$e(k) = y(k) - \underbrace{[\cosh(k) \quad \operatorname{senh}(k) \quad \cos(k)]}_{\psi(k)} \underbrace{\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix}}_{\hat{\theta}(k-1)} \tag{7.8}$$

El cuadro 7.5 contiene el programa cap7_ejemplo3.m con el c´odigo fuente en **MATLAB** para llevar a cabo el proceso de identificaci´on param´etrica del sistema.

Se emplea un vector tiempo definido de 0 a 10 segundos con intervalos de un milisegundo. El sistema a identificar se encuentra definido en la l´ınea 8; el regresor **fi** se forma en la l´ınea 9. La funci´on de m´ınimos cuadrados recursivo mincuad se emplea en la l´ınea 10 para obtener el vector de par´ametros estimados $\hat{\theta}$, cuyo resultado se presenta en la l´ınea 11:

```
theta=
    0.001
    456
   -345
```