



### Código Fuente 7.5 cap7\_ejemplo3.m

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega, Fernando Reyes Cortés.
%Capítulo 7 Identificación paramétrica.
```

```
cap7_ejemplo3.m
```

```
1 clc;
2 clear all;
3 close all;
4 format short g
5 t=[0:0.001:10]';%vector columna de tiempo
6 %parámetros del sistema
7 a1=0.001; a2=456; a3=-345;%sistema a identificar
8 y=a1*cosh(t)+a2*sinh(t)+a3*cos(t);
9 fi=[cosh(t), sinh(t), cos(t)];
10 [~, theta] =mincuad(y,fi);
11 theta %resultado de la identificación
```

### ♣ Ejemplo 7.4

Identificar los parámetros del siguiente sistema dinámico lineal:

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

donde los valores de los parámetros son:  $[a, b]^T = [6, 2]^T$

### Solución

El error de predicción está definido por:

$$e(k) = y(k) - \underbrace{[-x \quad u]}_{\psi(k)} \underbrace{\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}}_{\hat{\theta}(k-1)}$$

El cuadro 7.7 contiene el código del programa cap7 ejemplo4.m. Debido a que el sistema a identificar es dinámico, se emplearía la función ode45() para llevar a cabo el proceso de integración numérica. El programa cap7 se1.m tiene implementado al sistema dinámico  $\dot{x} = -ax + bu$ , cuyo código se encuentra en el cuadro 7.6. Esta declaración también se encuentra en el programa cap7 se1.m.

Es importante resaltar la utilidad de la variable global u que registra la entrada del sistema dinámico  $u = \sin(t)$  (ver línea 4 ). La línea 12 del programa cap7\_ejemplo4.m realiza la solución numérica del sistema  $\dot{x} = -ax + bu$  para un intervalo de tiempo de 0 a 5 segundos, con incrementos de un milisegundo. En esta fase se obtiene la variable de estado x, la cual se emplea en la línea 13 para obtener la variable velocidad  $\dot{x}$ . El regresor de observaciones **fi** se forma en la línea 14. El vector de parámetros estimados  $\hat{\theta}$  se obtiene en la línea 16; usando el algoritmo de mínimos cuadrados implementado en la función mincuad y el resultado se presenta en la línea 17:

theta=

6

2



### Código Fuente 7.6 cap7\_se1.m

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega, Fernando Reyes Cortés.
%Capítulo 7 Identificación paramétrica.
```

cap7\_se1.m

```
1 function xp =se1(t,x)
2     global qpp u
3     u=sin(t); a=6; b=2;
4     xp=-a*x+b*u;
5     qpp=xp;
6 end
```



### Código Fuente 7.7 cap7\_ejemplo4.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

---

cap7\_ejemplo4.m

---

```

1 clc;
2 clear all;
3 close all;
4 global u %variable global almacena la entrada del sistema
5 format short g
6 ti=0;
7 h=0.001;
8 tf = 5;
9 ts=ti:h:tf;%vector tiempo de simulación
10 opciones=odeset('RelTol',1e-3,'InitialStep',2.5e-3,'MaxStep',2.5e-3);
11 %solución numérica del sistema dinámico lineal
12 [t,x]=ode45('cap7 se1',ts,0,opciones);
13 xp=se1(t,x); %se obtiene la velocidad
14 fi=[-x, u]; %regresor
15 %algoritmo de mínimos cuadrados
16 [r,theta] =mincuad(xp,fi);
17 theta %resultado de la identificación

```

---

### ♣♣♣ Ejemplo 7.5

Identificar los parámetros del siguiente sistema multivariable:

$$\begin{aligned}
 y_1 &= 8.7 \operatorname{sen}(t) + 0.001 \operatorname{cos}(t) \\
 y_2 &= 109.789 \operatorname{atan}(t) - 0.23 * \operatorname{tanh} t
 \end{aligned}$$

**Solución**

En el cuadro 7.8 se muestra el programa cap7 ejemplo5.m; observe que en la línea 3 se forma el vector columna, en la línea 4 se obtiene el número de observaciones o muestras Nob. En las líneas 5-6 se define el sistema a identificar; en la línea 7 se forma el vector de salida del sistema  $\mathbf{y} = [y_1; y_2]$ , el regresor o vector de observaciones se forma en la línea 8 y 9. La función mincuadm es utilizada en la línea 10 para obtener el vector de estimación paramétrica cuyo resultado es:  $\hat{\theta} = [8.7, 0.001, 109.79, -0.23]$ .

**Código Fuente 7.8 cap7\_ejemplo5.m**

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega, Fernando Reyes Cortés.
%Capítulo 7 Identificación paramétrica.
```

---

```
cap7_ejemplo5.m
```

---

```
1 clc; clear all; close all;
2 format short g
3 t=(0:0.001:10)';
4 [Nob, ~]=size(t);
5 y1=8.7*sin(t)+0.001*cos(t);
6 y2=109.789*atan(t)-0.23*tanh(t);
7 y=[y1; y2];
8 fi=[sin(t), cos(t), zeros(Nob,1), zeros(Nob,1);
9     zeros(Nob,1), zeros(Nob,1), atan(t), tanh(t)];
10 theta=mincuadm(y,fi,Nob,4,2);
11 theta
```

A continuación se presentan aplicaciones de identificación paramétrica de sistemas mecánicos y robots manipuladores. Particularmente se explotan las propiedades matemáticas del modelo dinámico de los robots manipuladores para expresarlo como un regresor lineal teniendo casos escalares (modelo de energía y potencia) y multivariable (modelo dinámico).



## 7.5 Modelos de regresión del péndulo

La metodología de Euler-Lagrange permite obtener el modelo dinámico de robots manipuladores con propiedades importantes para diseñar y analizar esquemas de control e identificación paramétrica.

Dentro de las propiedades del modelo dinámico de robots manipuladores se encuentra la de linealidad en los parámetros, es decir: a pesar de que la dinámica del robot es no lineal con respecto a las variables de estado de la posición y velocidad, es posible expresar dicha dinámica como un regresor lineal de una matriz de observaciones y un vector columna de parámetros desconocidos. Además, las propiedades de linealidad también se satisfacen en los modelos de energía y potencia que seleccionando de manera conveniente el regresor de observaciones, resultan los mismos parámetros que los modelos de energía y potencia.

Como caso particular de robots manipuladores de 1 gdl se encuentra el péndulo que tiene las mismas ventajas de linealidad en los parámetros. En esta sección se presenta la implementación y análisis comparativo de cinco esquemas de regresión paramétrica: modelo dinámico, dinámico filtrado, energía, potencia y potencia filtrada.



### Modelo dinámico del péndulo

El modelo dinámico del péndulo robot se encuentra dado por la siguiente ecuación:

$$\tau_1 = [I_{r1} + m_1 l_{c1}^2] \ddot{q}_1 + b_1 \dot{q}_1 + f_{c1} \text{signo}(\dot{q}_1) + m_1 g l_{c1} \text{sen}(q_1) \quad (7.9)$$

donde  $I_{r1}$  es el momento de inercia del rotor,  $m_1$  es la masa del péndulo,  $l_{c1}$  representa el centro de masa,  $b_1$ ,  $f_{c1}$  son los coeficientes de fricción viscosa y de Coulomb, respectivamente; las variables articulares son la aceleración  $\ddot{q}_1$  velocidad  $\dot{q}_1$  y posición  $q_1$ ; la constante de la aceleración debido a la gravedad es  $g$  y  $\tau_1$  es el par aplicado al servomotor.

♣ ♣ Ejemplo 7.6

Obtener la identificación paramétrica del modelo dinámico del péndulo-robot. Tomar en cuenta los valores de los parámetros del péndulo que se encuentran contenidos en la tabla 6.3.

Solución

El esquema de regresión del modelo dinámico del péndulo está dado por:

$$e(k) = \underbrace{\ddot{y}(k)}_{\tau_1(k)} - \underbrace{[q_1(k) \quad \dot{q}_1(k) \quad \text{signo}(\dot{q}_1(k)) \quad \text{sen}(q_1(k))]}_{\psi(k)} \underbrace{\begin{bmatrix} \hat{I}_{r1} + \hat{m}_1 l_c^2 \\ b_1 \\ \hat{f}_c \\ m_1 \hat{g} l_c \end{bmatrix}}_{\hat{\theta}(k-1)}$$

donde  $\hat{I}_{r1} + \hat{m}_1 l_c^2$ ,  $b_1$ ,  $\hat{f}_c$  y  $m_1 \hat{g} l_c$  son los parámetros estimados de  $I_{r1} + m_1 g l_c^2$ ,  $b_1$ ,  $f_c$ , y  $m_1 g l_c$ , respectivamente.

El modelo dinámico del péndulo está contenido en el programa cap7 pendulo.m (ver cuadro 7.10). Para los propósitos de identificación paramétrica es conveniente registrar la señal del par aplicado  $\tau_1$  al servomotor y la aceleración del péndulo  $\ddot{q}_1$  a través de variables globales tal y como se define en la línea 2. En la declaración de la función del péndulo (línea 1) se ha incluido la variable bandera para poder obtener la aceleración articular  $\ddot{q}_1$  empleando las variables de estado  $q_1$  y  $\dot{q}_1$  (ver líneas 4 y 5), o realizar integración numérica usando el algoritmo de Runge-Kutta ode45(...), que en este caso se emplean en las líneas 7 y 8.

Los parámetros del péndulo se encuentran declarados en la línea 11 que corresponden a la tabla 6.3. De acuerdo con esos valores numéricos, el vector de parámetros  $\theta = [0.18546 \quad 0.16 \quad 0.19 \quad 3.0831]^T$ . La aceleración articular del péndulo se obtiene en la línea 20 y el vector de salida en la líneas 22 y 23.

El programa principal cap7 pendulosimu.m permite realizar la identificación paramétrica del modelo dinámico y se presenta en el cuadro 7.9; en la línea 2 se declaran las variables globales tau1 qpp1 que también están definidas en el programa

del modelo dinámico del péndulo 7.10. El tiempo de simulación es de 0 a 5 segundos con pasos de 0.001 milisegundo (línea 3); las condiciones iniciales del péndulo son:  $[q(0)_1, \dot{q}_1(0)]^T = [0, 0]^T$  (línea 4).

En la línea 7 se realiza la integración numérica del modelo dinámico del péndulo. En esta fase se obtienen las variables de estado  $q_1(t), \dot{q}_1(t)$  contenidas en el vector  $x=[q_1(t), \dot{q}_1(t)]^T$ . En la línea 8 se emplea la función `cap7_pendulo(t,x,9)`; note que el valor del tercer argumento tiene asignado el valor 9 para no realizar el proceso de integración numérica, solamente obtener la aceleración articular del péndulo usando la variable de estado  $x$ ; la información de la aceleración articular se registra en la variable global `qpp1` y la energía aplicada al servomotor en la variable global `tau1`. En las líneas 9 y 10 se realiza un cambio de notación con la finalidad presentar más claro el vector de observaciones que se forma en la línea 11 y el algoritmo de mínimos cuadrados es llamado en la línea 12).



### Código Fuente 7.9 cap7\_pendulosimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

---

cap7\_pendulosimu.m

---

```

1 clc; clear all; close all; format short g
2 global tau1 qpp1 %variables globales
3 ti=0; h=0.001; tf = 5 ;
4 ci=[0; 0]; %condiciones iniciales
5 opciones=odeset('RelTol',1e-3,'InitialStep', 1e-3,'MaxStep',1e-3);
6 %solución numérica del modelo dinámico del péndulo
7 [t,x]=ode45('cap7_pendulo',t,ci,opciones);
8 xp=pendulo(t,x,9);
9 q1=x(:,1) ; %posición articular
10 qp1=x(:,2); %velocidad articular
11 fi=[qpp1, qp1, sign(qp1), sin(q1)]; %regresor  $\psi$ 
12 [r,theta] =mincuad(tau1,fi); %mínimos cuadrados
13 theta
```

---

**Código Fuente 7.10 cap7\_pendolo.m**

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

**cap7\_pendolo.m**

```

1 function xp =cap7_pendolo(t,x, bandera)
2     global tau1 qpp1
3     if (bandera==9)%evalúa la aceleración qpp1 en función de q1 y qp1
4         q1=x(:,1);
5         qp1=x(:,2);
6     else %integración numérica
7         q1=x(1);%posición articular
8         qp1=x(2);%velocidad articular
9     end
10    %parámetros del péndulo
11    m1=3.88; lc1=0.081; ;
12    %coeficientes de fricción viscosa y de Coulomb
13    b1=0.16; fc1=0.19;
14    lr1=0.16;
15    lp=lr1+m1*lc1*lc1;
16    g=9.81;
17    %par aplicado al servomotor
18    tau1=sin(t);
19    %aceleración articular del péndulo
20    qpp1=(tau1-b1*qp1-fc1*sign(qp1)-m1*g*lc1*sin(q1))/lp;
21    %vector de salida
22    xp=[ qp1 ; %xp(1)=x(2) %velocidad articular
23        qpp1] ; %xp(2)=qpp %aceleración articular
24 end

```

Los resultados de la identificación paramétrica del modelo dinámico del péndulo se presentan en la tabla 7.1.





### Modelo dinámico filtrado del péndulo

El modelo dinámico presupone que el sistema cuenta con los sensores necesarios para medir la posición  $q_1$ , velocidad  $\dot{q}_1$  y aceleración  $\ddot{q}_1$ . Sin embargo, generalmente los robots industriales sólo cuentan con el encoder o sensor de posición. Esto representa una desventaja para medir la aceleración. Por lo tanto, el problema consiste en usar la información de la posición articular para generar las señales de velocidad y aceleración.

Para resolver este problema práctico se puede usar un conjunto de filtros  $F_u = \frac{\lambda}{s + \lambda} u$  para obtener la señal filtrada de  $u$  (representada por  $F_u$ ) y la velocidad  $\dot{u}$  es aproximada por:  $\dot{u} \approx F_{\dot{u}}$ , entonces el modelo dinámico filtrado del péndulo se obtiene por aplicar el operador filtro  $\frac{\lambda}{s + \lambda}$  a la ecuación (7.9) obteniendo:

$$\begin{aligned} \frac{\lambda \tau_1}{s + \lambda} &= [I_{r1} + m_1 l_{c1}^2] \frac{\lambda \ddot{q}_1}{s^2 + \lambda s} + b_1 \frac{\lambda \dot{q}_1}{s + \lambda} + f_{r1} \frac{\lambda \text{signo}(q_1)}{s + \lambda} + m_1 l_{c1} \frac{\lambda \text{sen}(q_1)}{s + \lambda} \\ F_{\tau} &= [I_{r1} + m_1 l_{c1}^2] F_{F_{q_1}} + b_1 F_{F_{\dot{q}_1}} + f_{r1} F_{F_{\text{signo}(q_1)}} + m_1 l_{c1} F_{F_{\text{sen}(q_1)}} \end{aligned}$$

donde  $s = \frac{\sigma}{\tau}$  y  $\lambda$  es la frecuencia de corte del filtro, y las señales filtradas están dadas en forma recursiva como:

$$\begin{aligned} F_{\tau_1}(k) &= e^{-\lambda h} F_{\tau_1}(k-1) + [1 - e^{-\lambda h}] \tau_1(k-1) \\ \dot{F}_{\tau_1}(k) &= -\lambda F_{\tau_1}(k) + \lambda \tau_1(k) \\ F_{q_1}(k) &= e^{-\lambda h} F_{q_1}(k-1) + [1 - e^{-\lambda h}] q_1(k-1) \\ \dot{F}_{q_1}(k) &= -\lambda F_{q_1}(k) + \lambda q_1(k) \\ F_{\dot{q}_1}(k) &= e^{-\lambda h} F_{\dot{q}_1}(k-1) + [1 - e^{-\lambda h}] \dot{q}_1(k-1) \\ F_{F_{q_1}}(k) &= -\lambda F_{F_{q_1}}(k) + \lambda F_{q_1}(k) \\ F_{\text{sen}(q_1)}(k) &= e^{-\lambda h} F_{\text{sen}(q_1)}(k-1) + [1 - e^{-\lambda h}] \text{sen}(q_1)(k-1) \\ \dot{F}_{\text{sen}(q_1)}(k) &= -\lambda F_{\text{sen}(q_1)}(k) + \lambda \text{sen}(q_1)(k) \\ F_{\text{signo}(q_1)}(k) &= e^{-\lambda h} F_{\text{signo}(q_1)}(k-1) + [1 - e^{-\lambda h}] \text{signo}(q_1)(k-1) \\ \dot{F}_{\text{signo}(q_1)}(k) &= -\lambda F_{\text{signo}(q_1)}(k) + \lambda \text{signo}(q_1)(k) \end{aligned}$$

$F_{q_1}$  representa la señal filtrada de la posición  $q_1$ ; la velocidad  $\dot{q}_1$  es aproximada por

el filtro  $F_{q_1}$  y la aproximación de la aceleración es  $\ddot{q}_1 \approx F_{F_{q_1}}$ .

♣♣♣ Ejemplo 7.7

Obtener la identificación paramétrica del modelo dinámico filtrado del péndulo-robot. Para propósitos de simulación del modelo dinámico tomar en cuenta los valores de los parámetros del péndulo que se encuentran contenidos en la tabla 6.3.

Solución

El esquema de regresión del modelo dinámico filtrado del péndulo está dado por:

$$e(k) = \frac{\lambda \tau_1(k)}{\underline{s} + \lambda} - \left[ \frac{\lambda \hat{g}_1(k)}{\underline{s} + \lambda} \quad \frac{\lambda \hat{q}_1(k)}{\underline{s} + \lambda} \quad \frac{\lambda \text{signo}(\hat{q}_1(k))}{\lambda \text{sen}(\hat{m}_1(k))} \right] \underbrace{\begin{bmatrix} \hat{I}_r + m_1 l_c \\ b_1 \\ f_c \\ \hat{m}_1 g \\ \hat{l}_c \end{bmatrix}}_{\hat{\theta}(k-1)}$$

El programa `cap7_pendulodinafiltsimu.m` tiene el código **MATLAB** que permite realizar identificación paramétrica del modelo dinámico filtrado del péndulo (ver cuadro 7.11). El vector tiempo varía de 0 a 5 segundos con incrementos de 0.001.

En la línea 8 se emplea la función `ode45` para obtener la solución numérica  $q_1$ ,  $\dot{q}_1$  del péndulo. Con estas variables de estado en la línea 9 se llama a la función `cap7_pendolo.m` para registrar la señal del par aplicado  $\tau_1 = \text{sen}(t)$  la cual fue declarada como variable global (línea 3). De la línea 11 a la 14 se generan los vectores correspondientes para implementar los filtros. La frecuencia de corte  $\lambda$  ha sido puesta de manera empírica en 25 Hertz. Los filtros se realizan en las líneas 16-24. El regresor de observaciones se forma en la línea 25 y se emplea como parámetro de entrada con el registro del par aplicado filtrado en la función de mínimos cuadrados (línea 26). El vector de parámetros estimado  $\theta$  se encuentra indicado en la tabla 7.1. **Nota:** el tiempo de ejecución del programa `cap7_pendulodinafiltsimu.m` puede demorar varios segundos por utilizar la función discontinua `signo`.



### Código Fuente 7.11 cap7\_pendulodinafiltsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

---

cap7\_pendulodinafiltsimu.m

---

```

1  clc; clear all; close all;
2  format short g
3  global tau1 %variable global almacena la entrada del sistema
4  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
5  ci=[0; 0]; %condiciones iniciales
6  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
7  %solución numérica del modelo dinámico del péndulo
8  [t,x]=ode45('cap7_pendolo',t,ci,opciones);
9  xp=pendulo(t,x,9);
10 [m n]=size(t); q1=x(:,1); qp1=x(:,2);
11 Ftau1=zeros(m,1);
12 Fq1=zeros(m,1); sFq1=zeros(m,1);
13 Fqp1=zeros(m,1); sFqp1=zeros(m,1);
14 Fsign=zeros(m,1); Fsin=zeros(m,1);
15 lambda=25;
16 for k=2:m %filtros recursivos
17     Ftau1(k)=exp(-h*lambda)*Ftau1(k-1)+(1-exp(-h*lambda))*tau1(k-1);
18     Fq1(k)=exp(-h*lambda)*Fq1(k-1)+(1-exp(-h*lambda))*q1(k-1);
19     sFq1(k)=-lambda*Fq1(k)+lambda*q1(k);
20     Fqp1(k)=exp(-h*lambda)*Fqp1(k-1)+(1-exp(-h*lambda))*sFq1(k-1);
21     sFqp1(k)=-lambda*Fqp1(k)+lambda*sFq1(k);
22     Fsign(k)=exp(-h*lambda)*Fsign(k-1)+(1-exp(-h*lambda))*sign(sFq1(k-1));
23     Fsin(k)=exp(-h*lambda)*Fsin(k-1)+(1-exp(-h*lambda))*sin(q1(k-1));
24 end
25 fi=[sFqp1, sFq1, Fsign, Fsin];
26 [r,theta] =mincuad(Ftau1,fi);
27 theta

```

---



**Modelo de energía del péndulo**

Un enfoque particularmente atractivo para el diseño de esquemas de identificación paramétrica de robots manipuladores se basa en el modelo de energía aplicada al robot manipulador, el cual puede ser descrito como un modelo de regresión lineal en términos de los parámetros dinámicos.

El modelo de energía del péndulo consiste de la suma de la energía cinética  $K(q_1, \dot{q}_1)$ , la energía potencial  $U(q_1)$  más la energía disipativa de la siguiente manera:

$$e(k) = \int_0^t \tau_1 dt - \left[ \frac{1}{2} m_1 \dot{q}_1^2 + m_1 g l_c (1 - \cos(q_1)) \right] - \int_0^t b_1 \dot{q}_1 dt - \int_0^t f_{c1} |q_1| dt$$

**♣ ♣ Ejemplo 7.8**

Realizar el proceso de identificación paramétrica del modelo de energía del péndulo-robot. Emplear los valores numéricos de los parámetros del péndulo indicados en la tabla 6.3.

**Solución**

El modelo de regresión lineal de la energía para el péndulo se encuentra dado por:

$$e(k) = \underbrace{\int_0^t \tau_1 dt}_{y(k)} - \underbrace{\left[ \frac{1}{2} \hat{m}_1 \dot{q}_1^2 + \left[ \hat{m}_1 g l_c (1 - \cos(q_1)) \right] \right]}_{\psi(k)} - \underbrace{\left[ \int_0^t b_1 \dot{q}_1 dt + \int_0^t f_{c1} |q_1| dt \right]}_{\hat{\theta}(k-1)}$$

En el cuadro 7.12 se muestra el programa cap7\_penduloenergysimu.m; en la línea 7 se obtiene la solución numérica de la dinámica del péndulo. Por medio de las variables de estado de posición y velocidad en la línea 8 se emplea el modelo dinámico para registrar la señal de par aplicado  $\tau_1 = \sin(t)$ . La integral de la energía, fricción viscosa y de Coulomb se calcula de la línea 15 a la 19. La técnica de integración

numérica que se utiliza es la del método de Euler. El regresor del modelo de energía se forma en la línea 20 y el vector de parámetros estimados  $\theta$  se obtiene en la línea 21 por medio de la función mincuad; el valor de los parámetros identificados se encuentran en la tabla 7.1.



### Código Fuente 7.12 cap7\_penduloenersimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.  
 %Editorial Alfaomega, Fernando Reyes Cortés.  
 %Capítulo 7 Identificación paramétrica.

#### cap7\_penduloenersimu.m

```

1  clc; clear all; close all; format short g
2  global tau1 %variable global almacena la entrada del sistema
3  ti=0; h=0.001; tf = 5; t=ti:h:tf; %vector tiempo
4  ci=[0; 0]; %condición inicial
5  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
6  %solución numérica del modelo dinámico del péndulo
7  [t,x]=ode45('cap7_pendolo',t,ci,opciones);
8  xp=pendulo(t,x,9);
9  [m n]=size(t);
10 q1=x(:,1);
11 qp1=x(:,2);
12 Int1=zeros(m,1);
13 Int2=zeros(m,1);
14 Int3=zeros(m,1);
15 for k=2:m
16     Int1(k,1)=Int1(k-1,1)+h*qp1(k,1)*tau1(k,1);
17     Int2(k,1)=Int2(k-1,1)+h*qp1(k,1)*qp1(k,1);
18     Int3(k,1)=Int3(k-1,1)+h*abs(qp1(k,1));
19 end
20 fi=[1/2*qp1.*qp1, Int2, Int3, (1-cos(q1))];
21 [r,theta] =mincuad(Int1,fi);
22 theta

```



### Modelo de potencia del péndulo

El modelo de potencia del péndulo está dado por la siguiente ecuación:

$$\tau_1 \dot{q}_1 = [\hat{I}_{r1} + m_1 l_c^2] \dot{q}_1 \ddot{q}_1 + b_1 \dot{q}_1 + f_c |q_1| + m_1 g l_c \sin(q_1) \dot{q}_1 \quad (7.10)$$

El modelo de la potencia involucra la medición de la aceleración articular  $\ddot{q}_1$ .

#### ♣ ♣ Ejemplo 7.10

Realizar el proceso de identificación paramétrica para el modelo de potencia del péndulo. Utilice los parámetros del péndulo indicados en la tabla 6.3.

#### Solución

El modelo de potencia del péndulo tiene la propiedad de linealidad en los parámetros; por lo que el regresor lineal adquiere la siguiente forma:

$$e(k) = \tau_1 \dot{q}_1 - \underbrace{[\dot{q}_1 \ddot{q}_1 \quad \dot{q}_1^2 \quad |q_1| \quad \sin(q_1) \dot{q}_1]}_{\Psi(k)} \underbrace{\begin{bmatrix} \hat{I}_{r1} + m_1 l_c^2 \\ b_1 \\ f_c \\ \hat{m}_1 g l_c \end{bmatrix}}_{\hat{\theta}(k-1)} \quad (7.11)$$

El programa en **MATLAB** cap7\_pendulopotsimu.m se encuentra descrito en el cuadro 7.13. De manera análoga a los anteriores programas, el uso de las variables globales es adecuado para registrar señales importantes en la formación del regresor.

En la línea 3 se encuentra la definición de las variables tau\_1 y qpp1 para registrar la información en el modelo dinámico del péndulo (cap7\_pendolo.m); la línea 8 resuelve numéricamente la dinámica del péndulo y en la línea 9 se registran las señales tau\_1 y qpp1 para formar el regresor de observaciones (línea 12). La función mincuad es empleada para obtener la estimación de los parámetros (línea 13).

El resultado del vector de parámetros  $\hat{\theta}$  se encuentra indicado en la tabla 7.1.



### Código Fuente 7.13 cap7\_pendulopotsimu.m

```
%MATLAB Aplicado a Robótica y Mecatrónica.
%Editorial Alfaomega, Fernando Reyes Cortés.
%Capítulo 7 Identificación paramétrica.
```

```
cap7_pendulopotsimu.m
```

```
1 clc; clear all; close all;
2 format short g
3 global tau1 qpp1 %variables globales para almacenar par aplicado y aceleración
4 ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector de tiempo
5 ci=[0; 0];%condiciones iniciales
6 opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
7 %solución numérica de la dinámica del péndulo
8 [t,x]=ode45('cap7_pendolo',t,ci,opciones);
9 xp=pendulo(t,x,9);
10 q1=x(:,1);
11 qp1=x(:,2);
12 fi=[qpp1.*qp1, qp1.*qp1, abs(qp1), sin(q1).*qp1];
13 [r,theta] =mincuad(tau1.*qp1,fi);
14 theta
```



## 7.5.5 Modelo de potencia filtrada

El modelo de la potencia filtrada del péndulo se obtiene aplicando el operador filtro  $\frac{\lambda}{s+\lambda}$  al modelo de potencia (7.10) obteniendo la siguiente expresión

$$\begin{aligned} \frac{\lambda \tau_1 \ddot{q}_1}{s + \lambda} &= \left[ I_{r1} + m l_c^2 \right] \frac{\lambda \ddot{q}_1}{s + \lambda} + b_1 \frac{\dot{q}_1}{s + \lambda} + f_{c1} \frac{\lambda |q_1|}{s + \lambda} + m_1 a_{l_1} \frac{\lambda \sin(q_1) q_1}{s + \lambda} \\ &= \frac{1}{\left[ I_{r1} + m l_c^2 \right] s + b_1 + f_{c1} \lambda |q_1| + m_1 a_{l_1} \lambda \sin(q_1) q_1} \lambda \ddot{q}_1 \end{aligned} \quad (7.12)$$

El modelo de potencia del péndulo tiene la desventaja de requerir la medición de la aceleración  $\ddot{q}_1$  como en el caso del modelo dinámico. Sin embargo, la aceleración

puede ser obtenida por aproximación de técnicas de filtrado. La aceleración puede ser aproximada de la siguiente forma:  $\hat{a}(k) = \frac{\lambda q_1^2}{s + \lambda} - \frac{\lambda \sin(q_1) q_1}{s + \lambda} - \frac{\lambda q_1^2}{\lambda(1 + \lambda)}$

♣♣♣ Ejemplo 7.10

Obtener el valor numérico de los parámetros del péndulo empleando el modelo de potencia filtrada. Utilice los parámetros del péndulo indicados en la tabla 6.3.

Solución

El modelo de regresión de la potencia filtrada para el péndulo es:

$$e(k) = \frac{\lambda q_1^2}{s + \lambda} - \left[ \frac{\lambda \sin(q_1) q_1}{s + \lambda} - \frac{\lambda q_1^2}{\lambda(1 + \lambda)} \right] \begin{bmatrix} \hat{r}_1 + m_1 l_c \\ b_1 \\ f_c \\ \hat{m}_1 g \hat{l}_c \end{bmatrix} \hat{\theta}(k-1)$$

El proceso de identificación paramétrica del modelo dinámico filtrado del péndulo se implementó en el programa cap7\_pendulopotfilsimu.m (cuadro 7.14). A través de la señal filtrada de la posición (líneas 13-21) se puede obtener la estimación de la velocidad y aceleración. Por ejemplo, la señal filtrada de la posición se obtiene en la línea 15 y la aproximación de la velocidad articular se realiza en la línea 16. La señal filtrada de la velocidad articular se genera en la línea 17, la cual se emplea para aproximar a la aceleración en la línea 18. Evidentemente la señal de la aceleración es de baja calidad debido a que fue obtenida de la aproximación de la velocidad; lo más adecuado es obtener la aceleración por aproximación de la señal de velocidad que entrega el sensor del robot. No siempre es posible realizar eso, ya que los robots sólo están equipados con el sensor del encoder para proporcionar información de la posición. Sin embargo, este es un escenario real que corresponde al ambiente industrial donde la contaminación de ruido es alto en las variables de estado. En este contexto, la sintonía de la frecuencia de corte  $\lambda$  presenta un problema práctico el cual se determina de manera empírica. El resultado del vector  $\hat{\theta}$  se encuentra en la tabla 7.1.




**Código Fuente 7.14 cap7\_pendulopotfilsimu.m**

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

---

**cap7\_pendulopotfilsimu.m**


---

```

1  clc; clear all; close all;
2  format short g
3  global tau1 qpp1 %variables global para almacenar par y aceleración
4  ti=0; h=0.001; tf = 5; t=ti:h:tf; %vector tiempo
5  ci=[0; 0]; %condiciones iniciales
6  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
7  %solución numérica de la dinámica del péndulo
8  [t,x]=ode45('cap7_pendolo',t,ci,opciones);
9  xp=pendulo(t,x,9); [m n]=size(t);
10 q1=x(:,1); qp1=x(:,2);
11 Ftau1qp1=zeros(m,1); Fq1=zeros(m,1); sFq1=zeros(m,1); Fqp1qp1=zeros(m,1);
12 sFqp1=zeros(m,1); Fabs=zeros(m,1); Fsinqp1=zeros(m,1); lambda=25;
13 for k=2:m %filtros
14     Ftau1qp1(k)=exp(-h*lambda)*Ftau1qp1(k-1)+(1-exp(-h*lambda))*tau1(k-1)*qp1(k-1);
15     Fq1(k)=exp(-h*lambda)*Fq1(k-1)+(1-exp(-h*lambda))*q1(k-1);
16     sFq1(k)=-lambda*Fq1(k)+lambda*q1(k);
17     Fqp1qp1(k)=exp(-h*lambda)*Fqp1qp1(k-1)+(1-exp(-h*lambda))*sFq1(k-1)*sFq1(k-1);
18     sFqp1(k)=-lambda*Fqp1qp1(k)+lambda*sFq1(k)*sFq1(k);
19     Fabs(k)=exp(-h*lambda)*Fabs(k-1)+(1-exp(-h*lambda))*abs(sFq1(k-1));
20     Fsinqp1(k)=exp(-h*lambda)*Fsinqp1(k-1)+(1-exp(-h*lambda))*sin(q1(k-1,1))*sFq1(k-1,1);
21 end
22 fi=[(1/2)*sFqp1, Fqp1qp1, Fabs, Fsinqp1];
23 [r,theta] =mincuad(Ftau1qp1,fi);
24 theta

```



**Análisis comparativo de esquemas de regresión**

La tabla 7.1 contiene el resumen de resultados de los cinco esquemas de regresión evaluados para el péndulo. Para propósitos de comparación se presentan los valores reales de los parámetros  $\theta_i$ , con  $i = 1, \dots, 4$ .

Los modelos de regresión dinámico y de potencia generan una estimación exacta, libre de error, en contraste el modelo de regresión de la energía tiene un 99.956% de exactitud en el parámetro del momento de inercia, tiene una deriva del 3.43% en referencia al valor del coeficiente de fricción viscosa, la exactitud de estimación en el coeficiente de fricción de Coulomb es del 99.91%, mientras que para el parámetro del par gravitacional es del 100%. En este procedimiento hay que tomar en cuenta que el método de integración numérica seleccionado fue el de Euler y también influye el espaciamiento que existe entre las muestras ( $h = 0.001$ ). Los modelos de regresión filtrado presentan pobre exactitud en la identificación paramétrica; errores grandes alrededor del 40%. En este punto es conveniente resaltar que la frecuencia de corte de los filtros  $\lambda$  influye notablemente en la convergencia paramétrica, su valor se determina en forma empírica; además la función de transferencia del filtro  $\frac{\lambda}{\lambda + s}$  distorsiona la fase, lo que trae como consecuencia errores de aproximación en la aceleración articular.

**Tabla 7.1 Comparación de resultados de identificación paramétrica**

Parámetro	$\frac{J + m_1 l^2}{\text{rad}}$	$\frac{b_1}{\text{Nm seg rad}}$	$f_{c1}$ Nm	$m_1 g l c_1$ Nm
Esquema de regresión	$\hat{J} r_1 + \hat{m}_1 l^2 c_1$	$\hat{b}_1$	$\hat{f}_{c1}$	$\hat{m}_1 \hat{g} l c_1$
Valor real	0.18546	0.16	0.19	3.0831
Modelo dinámico	0.18546	0.16	0.19	3.0831
Modelo dinámico filtrado	0.17682	0.25021	0.15068	3.0384
Modelo de energía	0.18538	0.16055	0.18983	3.0831
Modelo de potencia	0.18546	0.16	0.19	3.0831
Modelo de potencia filtrada	0.24657	0.26366	0.11265	3.1154



## 7.6 Modelos de regresión del robot de 2 gdl

El modelo dinámico de un robot antropomórfico planar de 2 gdl está determinado por la matriz de inercia, fuerzas centrípetas y de Coriolis, par o torque gravitacional y fenómeno de fricción. Una parte clave del procedimiento de identificación paramétrica consiste en seleccionar un adecuado conjunto de parámetros; para el caso particular del robot de 2 gdl se pueden seleccionar 11 parámetros:

$$\begin{aligned}
 \mathbf{M}(\mathbf{q}) &= \begin{bmatrix} \theta_1 + 2\theta_2 \cos(q_2) & \theta_3 + \theta_2 \cos(q_2) \\ \theta_3 + \theta_2 \cos(q_2) & \theta_3 \end{bmatrix}, \\
 \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} -2\theta_2 \sin(q_2) \dot{q}_2 & -\theta_2 \sin(q_2) \dot{q}_2 \\ \theta_2 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}, \\
 \mathbf{g}(\mathbf{q}) &= \begin{bmatrix} \theta_4 \sin(q_1) + \theta_5 \sin(q_1 + q_2) \\ \theta_5 \sin(q_1 + q_2) \end{bmatrix}, \\
 \mathbf{f}(\dot{\mathbf{q}}) &= \begin{bmatrix} \theta_6 \dot{q}_1 + \theta_8 \operatorname{sgn}(\dot{q}_1) + \theta_{10} [1 - |\operatorname{signo}(\dot{q}_1)|] \\ \theta_7 \dot{q}_2 + \theta_9 \operatorname{sgn}(\dot{q}_2) + \theta_{11} [1 - |\operatorname{signo}(\dot{q}_2)|] \end{bmatrix}
 \end{aligned}$$

donde

$$\begin{aligned}
 \theta_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_2^2 + I_1 + I_2, \\
 \theta_2 &= l_1 m_2 l_{c2}, \\
 \theta_3 &= m_2 l_2^2 + I_2, \\
 \theta_4 &= g(l_{c1} m_1 + m_2 l_1), \\
 \theta_5 &= g m_2 l_{c2}, \\
 \theta_6 &= b_1, \\
 \theta_7 &= b_2, \\
 \theta_8 &= \mathbf{f}_{c1}, \\
 \theta_9 &= \mathbf{f}_{c2}, \\
 \theta_{10} &= \mathbf{f}_{e1}, \\
 \theta_{11} &= \mathbf{f}_{e2}.
 \end{aligned}$$



### Modelo de regresión dinámico del robot de 2 gdl

El modelo dinámico del robot antropomórfico de 2 gdl puede ser expresado de manera conveniente como el producto de una matriz de regresión y un vector de parámetros. Observe que los componentes del vector de parámetros  $\theta = [\theta_1 \dots \theta_{11}]^T$  dependen de una combinación de los parámetros físicos de los eslabones como masas, centros de masas, momentos de inercia y coeficientes de fricción, los cuales se consideran invariantes en el tiempo. En este caso, el orden de la matriz de regresión  $p$  es  $11 \times 11$ .

$$\begin{matrix} \tau_1 \\ \tau_2 \end{matrix} = \begin{matrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} & \psi_{15} & \psi_{16} & \psi_{17} & \psi_{18} & \psi_{19} & \psi_{110} & \psi_{111} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} & \psi_{25} & \psi_{26} & \psi_{27} & \psi_{28} & \psi_{29} & \psi_{210} & \psi_{211} \end{matrix} \begin{matrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \theta_{10} \\ \theta_{11} \end{matrix} \quad (7.13)$$

donde los elementos del regresor  $\psi$  están dados como:

$$\begin{aligned} \psi_{11} &= \ddot{q}_1 \\ \psi_{12} &= 2 \cos(q_2) \ddot{q}_1 + \cos(q_2) \ddot{q}_2 - 2 \sin(q_2) \dot{q}_2 \dot{q}_1 - \sin(q_2) \dot{q}_2^2 \\ \psi_{13} &= \ddot{q}_2 \\ \psi_{14} &= \sin(q_1) \\ \psi_{15} &= \sin(q_1 + q_2) \\ \psi_{16} &= \dot{q}_1 \\ \psi_{17} &= 0 \\ \psi_{18} &= \text{signo}(\dot{q}_1) \\ \psi_{19} &= 0 \\ \psi_{110} &= 1 - |\text{signo}(\dot{q}_1)| \\ \psi_{111} &= 0 \\ \psi_{21} &= 0 \end{aligned}$$

$$\begin{aligned} \psi_{22} &= \cos(q_2) \ddot{q}_1 + \sin(q_2) \dot{q}_1^2 \\ \psi_{23} &= \ddot{q}_1 + \ddot{q}_2 \\ \psi_{24} &= 0 \\ \psi_{25} &= \sin(q_1 + q_2) \\ \psi_{26} &= 0 \\ \psi_{27} &= \dot{q}_2 \\ \psi_{28} &= 0 \\ \psi_{29} &= \text{signo}(\dot{q}_2) \\ \psi_{210} &= 0 \\ \psi_{211} &= 1 - |\text{signo}(\dot{q}_2)| \end{aligned}$$

♣♣♣ **Ejemplo 7.12**

Realizar el procedimiento de identificación paramétrica de un robot manipulador de 2 gdl empleando el modelo de regresión dinámico. Emplear los valores numéricos de los parámetros del robot de 2 gdl indicados en la tabla 6.4.

**Solución**

El modelo de regresión del modelo dinámico del robot antropomórfico de 2 gdl está dado por la siguiente expresión:

$$\mathbf{e}(k) = \begin{matrix} \tau_1 \\ \tau_2 \end{matrix} \mathbf{y}^{(k)} = \begin{matrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} & \psi_{15} & \psi_{16} & \psi_{17} & \psi_{18} & \psi_{19} & \psi_{110} & \psi_{111} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} & \psi_{25} & \psi_{26} & \psi_{27} & \psi_{28} & \psi_{29} & \psi_{210} & \psi_{211} \end{matrix} \mathbf{z}$$

Note que el problema de identificación paramétrica para un robot de 2 gdl es complejo, ahora el modelo de regresión es multivariable y no escalar como los esquemas de regresión del péndulo, por lo que deberá usarse la función `mincuadm`.

Una fase importante del procedimiento consiste en elegir una adecuada trayectoria de entrada al robot manipulador que permita obtener cierto grado de excitación en la matriz de regresión debido a que la calidad de estimación del vector de parámetros depende de los niveles de excitación del regresor; por ejemplo el algoritmo de mínimos cuadrados recursivo usualmente requiere satisfacer ciertas condiciones de excitación persistente. Sin embargo, la selección de la trayectoria no es una tarea trivial, debido a que la matriz de regresión está compuesta por una combinación no lineal de las variables dinámicas del robot tal como  $\text{sen}(q_2)q_1^2$ ,  $\text{cos}(q_2)q_1$ . Para el diseño de la trayectoria de excitación persistente se recomienda superposición de funciones senoidales como los siguientes pares aplicados:

$$\tau_1 = [1 - e^{-0.8t}] 29.0 + 68 \text{sen}(16t + 0.1) + 9 \text{sen}(20t + 0.15) \quad (7.14)$$

$$\tau_2 = [1 - e^{-1.8t}] 1.2 + 8 \text{sen}(26t + 0.08) + 2 \text{sen}(12t + 0.34) \quad (7.15)$$

El cuadro 7.15 contiene implementado el modelo dinámico del robot de 2 gdl en lenguaje **MATLAB** (programa `cap7_iderobot2gdl.m`); se han utilizado los valores numéricos de los parámetros del robot mostrados en la tabla 6.4, así como las ecuaciones del par aplicado (7.14)-(7.15).

El programa principal `cap7_iderobot2gdlsimu.m` permite realizar el procedimiento de identificación paramétrica del robot de 2 gdl, y está descrito en el cuadro 7.16. El registro de las señales de pares aplicados  $\tau_1, \tau_2$  se realiza a través de variables globales (línea 2). La solución numérica del modelo dinámico del robot de 2 gdl se encuentra en la línea 7; empleando dicha solución numérica  $q_1(t), q_2(t), \dot{q}_1(t), \dot{q}_2(t)$  en el modelo dinámico del robot se registran la aceleración  $\mathbf{q} = [\ddot{q} \ddot{q}]$  y pares aplicados  $\tau_1, \tau_2$  (ver líneas 11 a la 16).

Las componentes  $\psi_{ij}$ , con  $i = 1, 2, \dots, 11$  y  $j = 1, 2$  del regresor de observaciones se forman entre las líneas 18 y 24. El regresor  $\boldsymbol{\psi}$  queda formado en las líneas 25 y 26. La función de mínimos cuadrados recursivo para el caso multivariable `mincuadm` se usa en la línea 27 para obtener el vector de parámetros estimado  $\boldsymbol{\theta} \in \mathbb{R}^{11}$ . Los resultados se encuentran en la tabla 7.2.



### Código Fuente 7.15 cap7\_iderobot2gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

cap7\_iderobot2gdl.m

```

1 function xp = cap7_iderobot2gdl(t,x)
2     global tau1 tau2
3     q1=x(1); q2=x(2); q = [q1; q2]; %vector de posición articular
4     qp1=x(3); qp2=x(4); qp = [qp1; qp2]; % vector de velocidad articular
5     m1=23.902; l1=0.45; lc1=0.091; %parámetros del robot
6     l1=1.266; b1=2.288; fc1=7.17; fe1=8.8; m2=3.880; l2=0.45; lc2=0.048;
7     l2=0.093; b2=0.175; fc2=1.734; fe2=1.87; g=9.81;
8     theta1=m1*lc1*lc1+m2*l1*l1+m2*lc2*lc2+l1+l2;
9     theta2=l1*m2*lc2; theta3=m2*lc2*lc2+l2; theta4=g*(lc1*m1+m2*l1);
10    theta5=g*m2*lc2; theta6=b1; theta7=b2; theta8=fc1; theta9=fc2;
11    theta10=fe1; theta11=fe2;
12    %matriz de inercia
13    M=[theta1+2*theta2*cos(q2), theta3+theta2*cos(q2);
14        theta3+theta2*cos(q2), theta3];
15    %matriz de Coriolis y fuerzas centrífugas
16    C=[ -2*theta2*sin(q2)*qp2, -theta2*sin(q2)*qp2;
17        theta2*sin(q2)*qp1, 0];
18    gq11=theta4*sin(q1)+theta5*sin(q1+q2); gq21=theta5*sin(q1+q2);
19    gq=[gq11; gq21]; %par gravitacional
20    %par de fricción viscosa, Coulomb y estática
21    fr=[theta6*qp1+theta8*sign(qp1)+theta10*(1-abs(sign(qp1)))];
22        theta7*qp2+theta9*sign(qp2)+theta11*(1-abs(sign(qp2)))]];
23    tau1=(1-exp(-0.8*t))*29.0+ 68*sin(16*t+0.1) + 9*sin(20*t+0.15);
24    tau2=(1-exp(-1.8*t))*1.2+ 8*sin(26*t+0.08)+2*sin(12*t+0.34);
25    tau=[tau1; tau2]; %señal de excitación persistente
26    qpp = M^(-1)*(tau-C*qp-gq-fr); %aceleración articular
27    xp = [qp1; qp2; qpp(1); qpp(2)]; %vector de salida
28 end

```



### Código Fuente 7.16 cap7\_iderobot2gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

cap7\_iderobot2gdlsimu.m

```

1  clc; clear all; close all; format short g
2  global tau1 tau2
3  ti=0; h=0.001; tf = 5; t=ti:h:tf; %tiempo de simulación
4  ci=[0; 0; 0; 0]; %condiciones iniciales
5  opciones=odeset('RelTol',1e-3,'InitialStep', 1e-3, 'MaxStep', 1e-3);
6  %solución numérica del robot antropomórfico de 2 gdl
7  [t,x]=ode45('cap7_iderobot2gdl',t,ci,opciones);
8  %vectores de posición  $\mathbf{q} = [q_1, q_2]^T$  y velocidad  $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2]^T$ 
9  q1=x(:,1); q2=x(:,2); qp1=x(:,3); qp2=x(:,4);
10 [m n]=size(t); qpp1=zeros(m,1); qpp2=zeros(m,1); u1=zeros(m,1); u2=zeros(m,1);
11 for k=1:m
12     xp=cap7_iderobot2gdl(t(k),[x(k,1),x(k,2),x(k,3), x(k,4)]);
13     qpp1(k,1)=xp(3,1);
14     qpp2(k,1)=xp(4,1);
15     u1(k,1) =tau1; u2(k,1) =tau2;
16 end
17 tau=[u1; u2];
18 fi11=qpp1;
19 fi12=2*cos(q2).*qpp1+cos(q2).*qpp2-2*sin(q2).*qp2.*qp1-sin(q2).*qp2.*qp2;
20 fi13=qpp2; fi14=sin(q1); fi15=sin(q1+q2); fi16=qp1; fi17=zeros(m,1);
21 fi18=sign(qp1); fi19=zeros(m,1); fi110=(1-abs(sign(qp1))); fi111=zeros(m,1);
22 fi21=zeros(m,1); fi22=cos(q2).*qpp1+sin(q2).*qp1.*qp1; fi23=qpp1+qpp2;
23 fi24=zeros(m,1); fi25=sin(q1+q2); fi26=zeros(m,1); fi27=qp2;
24 fi28=zeros(m,1); fi29=sign(qp2); fi210=zeros(m,1); fi211=(1-abs(sign(qp2)));
25 fi=[fi11, fi12, fi13, fi14, fi15, fi16, fi17, fi18, fi19, fi110, fi111 ;
26     fi21, fi22, fi23, fi24, fi25, fi26, fi27, fi28, fi29, fi210, fi211];
27 theta=mincuadm(tau,fi,m,11,2);
28 theta

```





### Modelo de energía del robot de 2 gdl

El modelo de energía de un robot de 2 gdl con condiciones iniciales cero está dado por la siguiente ecuación:

$$\begin{aligned}
 \int_0^t [\tau_1 \quad \tau_2] \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} dt &= \frac{1}{2} \begin{bmatrix} \theta_1 & 0 \\ \theta_2 \cos(q_2) & \theta_3 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}^2 + \theta_3 \int_0^t |q_1| dt + \theta_{10} \int_0^t q_1 [1 - \text{signo}(q_1^2)] dt \\
 &+ \theta_8 \int_0^t |q_1| dt + \theta_{10} \int_0^t q_1 [1 - \text{signo}(q_1^2)] dt + \theta_3 \int_0^t q_2^2 dt + \theta_3 \int_0^t q_2 \cos(q_2) dt + \theta_3 \int_0^t q_2 dt
 \end{aligned}$$

El modelo de energía del robot manipulador de 2 gdl no requiere de la medición de la aceleración articular, y a demás se convierte en un modelo de regresión escalar.

Llevar a cabo la identificación paramétrica de un robot manipulador de 2 gdl empleando el modelo de regresión de la energía. Emplear los valores numéricos de los parámetros del robot de 2 gdl indicados en la tabla 6.4.

El error de predicción del modelo de regresión de la energía del robot antropomórfico de 2 gdl se encuentra expresado de la siguiente manera:

$$e(k) = \begin{bmatrix} R \\ 0 \end{bmatrix} \hat{y}(k) - \psi(k)$$

Alfaomega

Matlab Aplicado a Robotica y Mecatronica • Fernando Reyes Cortés.

$$\int^t (\tau_1 \dot{q}_1 + \tau_2 \dot{q}_2) dt - [\psi_{11} \ \psi_{12} \ \psi_{13} \ \psi_{14} \ \psi_{15} \ \psi_{16} \ \psi_{17} \ \psi_{18} \ \psi_{19} \ \psi_{110} \ \psi_{111}] \theta$$

siendo las componentes  $\psi_{ij}$  del vector de regresión:

$$\psi_{11} = \frac{1}{2} \ddot{q}_1$$

$$\psi_{12} = \cos(q_2) \dot{q}_1 (q_1 + q_2)$$

$$\psi_{13} = \dot{q}_2 [q_2 + q_1]$$

$$\psi_{14} = 1 - \cos(q_1)$$

$$\psi_{15} = 1 - \cos(q_1 + q_2)$$

$$\psi_{16} = \int_0^t \dot{q}_1^2 dt$$

$$\psi_{17} = \int_0^t \dot{q}_2 dt$$

$$\psi_{18} = \int_0^t |\dot{q}_1| dt$$

$$\psi_{19} = \int_0^t |\dot{q}_2| dt$$

$$\psi_{110} = \int_0^t q_1 [1 - |\text{signo}(q_1)|] dt$$

$$\psi_{111} = \int_0^t q_2 [1 - |\text{signo}(q_2)|] dt$$

El programa `cap7_energiarobot2gdlsimu.m` que se encuentra en el cuadro 7.17 lleva a cabo la implementación de identificación paramétrica para el robot de 2 gdl. De manera análoga al anterior ejemplo se emplea el modelo dinámico del robot de 2 gdl para obtener la solución numérica (línea 10) y el registro de la señal del par aplicado en la línea 15.

El conjunto de integrales del regresor de observaciones se encuentra implementado de la línea 20 a la 28 a través de la técnica de integración numérica de Euler. Las componentes individuales  $\psi_{ij}$  del regresor se forman en las líneas 30-32, mientras que el regresor completo está declarado en la línea 33. La función de mínimos cuadrados para la versión escalar `mincuad` se emplea en la línea 34. El resultado del vector de parámetros estimados  $\hat{\theta}$  está contenido en la tabla 7.2.



### Código Fuente 7.17 cap7\_energiarobot2gdsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

cap7\_energiarobot2gdsimu.m

```

1  clc; clear all; close all;
2  format short g
3  global tau1 tau2
4  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
5  ci=[0; 0; 0; 0]; %condiciones iniciales
6  opciones=odeset('RelTol',1e-3, 'InitialStep',1e-3, 'MaxStep',1e-3);
7  %solución numérica del robot de 2 gdl
8  disp('Advertencia: la simulación puede durar varios segundos.....')
9  disp('Espere por favor..... ')
10 [t,x]=ode45('cap7_iderobot2gdl',t,ci,opciones);
11 q1=x(:,1); q2=x(:,2);%vector de posición
12 qp1=x(:,3); qp2=x(:,4);%vector de velocidad
13 [m n]=size(t); Int1=zeros(m,1); Int2=zeros(m,1); Int3=zeros(m,1); Int4=zeros(m,1);
14 Int5=zeros(m,1); Int6=zeros(m,1); Int7=zeros(m,1); u1=zeros(m,1); u2=zeros(m,1);
15 for k=1:m      %registro de la se~nal del par aplicado τ
16     xp=cap7_iderobot2gdl(t(k),x(k,1),x(k,2),x(k,3), x(k,4));
17     u1(k,1) =tau1;
18     u2(k,1) =tau2;
19 end
20 for k=2:m      %integrales del regresor
21     Int1(k,1)=Int1(k-1,1)+h*(qp1(k,1)*u1(k,1)+qp2(k,1)*u2(k,1));
22     Int2(k,1)=Int2(k-1,1)+h*qp1(k,1)*qp1(k,1);
23     Int3(k,1)=Int3(k-1,1)+h*qp2(k,1)*qp2(k,1);
24     Int4(k,1)=Int4(k-1,1)+h*abs(qp1(k,1));
25     Int5(k,1)=Int5(k-1,1)+h*abs(qp2(k,1));
26     Int6(k,1)=Int6(k-1,1)+h*qp1(k,1)*(1-abs(sign(qp1(k,1))));
27     Int7(k,1)=Int7(k-1,1)+h*qp2(k,1)*(1-abs(sign(qp2(k,1))));
28 end
29 %regresor ψ
30 fi11=(1/2)*qp1.*qp1 ; fi12=cos(q2).*qp1.*(qp1+qp2); fi13=qp2.*((1/2)*qp2+qp1);
31 fi14=1-cos(q1); fi15=1-cos(q1+q2); fi16=Int2; fi17=Int3;
32 fi18=Int4; fi19=Int5; fi110=Int6; fi111=Int7;
33 fi=[fi11, fi12, fi13, fi14, fi15, fi16, fi17, fi18, fi19, fi110, fi111];
34 [r,theta] =mincuad(Int1,fi) ;
35 theta

```



**Modelo de potencia del robot de 2 gdl**

El modelo de potencia del robot de 2 gdl representa la derivada temporal del modelo de energía. La siguiente ecuación describe el modelo de potencia:

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} &= \begin{bmatrix} \theta_1 + 2\theta_2 \cos(q_2) & \theta_3 + \theta_2 \cos(q_2) \\ \theta_3 + \theta_2 \cos(q_2) & \theta_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
 &+ \frac{1}{1} \ddot{q}_1 \begin{bmatrix} -2\theta_2 \sin(q_2) \dot{q}_2 & -\theta_2 \sin(q_2) \dot{q}_2 \\ -\theta_2 \sin(q_2) \dot{q}_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
 &+ \begin{bmatrix} \theta_4 \sin(q_1) \dot{q}_1 + \theta_5 \sin(q_1 + q_2) [\dot{q}_1 + \dot{q}_2] \\ \theta_5 \sin(q_1 + q_2) [\dot{q}_1 + \dot{q}_2] \end{bmatrix} + \begin{bmatrix} \theta_6 \dot{q}_1^2 \\ \theta_7 \dot{q}_2 \end{bmatrix} \\
 &+ \begin{bmatrix} 2 \theta_8 |\dot{q}_1| & \theta_{10} \dot{q}_1 [1 - \text{signo}(|\dot{q}_1|)] \\ \theta_9 |q_2| & \theta_{11} q_2 [1 - \text{signo}(q_2)] \end{bmatrix}
 \end{aligned}$$

(7.17)

El modelo de potencia del robot manipulador de 2 gdl requiere la medición de la aceleración articular, y pertenece a la clase de modelos de regresión escalar.

Obtener el valor numérico de los parámetros de un robot de 2 gdl empleando el modelo de potencia. Utilice los parámetros del robot de 2 gdl indicados en la tabla 6.4.

Para llevar a cabo la identificación paramétrica del robot de 2 gdl usando el modelo de potencia es necesario obtener el modelo de regresión. Con esta finalidad considere

el siguiente regresor:

$$e(k) = \underbrace{[\tau_1 \quad \tau_2]}_{y(k)} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} - [\psi_{11} \quad \psi_{12} \quad \psi_{13} \quad \psi_{14} \quad \psi_{15} \quad \psi_{16} \quad \psi_{17} \quad \psi_{18} \quad \psi_{19} \quad \psi_{110} \quad \psi_{111}] \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \theta_{10} \\ \theta_{11} \end{bmatrix}$$

donde los elementos del regresor  $\psi$  están dados como:

$$\begin{aligned} \psi_{11} &= \dot{q}_1 \ddot{q}_1 \\ \psi_{12} &= -\text{sen}(q_2) q_2 \dot{q}_1 [q_1 + q_2] + \text{cos}(q_2) \ddot{q} [q_1 + q_2] + \text{cos}(q_2) q_1 [\ddot{q}_1 + \ddot{q}_2] \\ \psi_{13} &= \ddot{q}_2 [q_1 \dot{q}_2 + \dot{q}_1] + \dot{q}_2 [q_1 \ddot{q}_2 + \dot{q}_1] + \dot{q}_2 [q_2 \dot{q}_2 + \ddot{q}_1] \\ \psi_{14} &= \text{sen}(q_1) \dot{q}_1 \\ \psi_{15} &= \text{sen}(q_1 + q_2) [q_1 + q_2] \\ \psi_{16} &= \dot{q}_1^2 \\ \psi_{17} &= \dot{q}_2^2 \\ \psi_{18} &= |\dot{q}_1| \\ \psi_{19} &= |\dot{q}_2| \\ \psi_{110} &= \dot{q}_1 [1 - |\text{signo}(\dot{q}_1)|] \\ \psi_{111} &= \dot{q}_2 [1 - |\text{signo}(\dot{q}_2)|] \end{aligned}$$

El programa `cap7_potenciarobot2gdlsimu.m` contiene el código para realizar la identificación paramétrica de un robot de 2 gdl a través del modelo de potencia (ver cuadro 7.18). En forma similar a los anteriores ejemplos, primero se realiza la integración numérica sobre la dinámica del robot, con las variables de estado se obtiene el registro de aceleración y señales con los pares aplicados a los servomotores (líneas 18-24). Las componentes del vector de observaciones se indican en las líneas 25 a la 31. El regresor se forma en la línea 32 y la función de mínimos cuadrados escalar se emplea en la línea 33. El vector de predicción  $\hat{\theta}$  se encuentra indicado en la tabla 7.2.



### Código Fuente 7.18 cap7\_potenciarobot2gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

#### cap7\_potenciarobot2gdlsimu.m

```

1  clc; clear all;
2  close all;
3  format short g
4  global tau1 tau2
5  ti=0; h=0.001; tf = 5;
6  t=ti:h:tf;%vector tiempo
7  ci=[0; 0; 0; 0];%condiciones iniciales
8  opciones=odeset('RelTol', 1e-3, 'InitialStep',1e-3, 'MaxStep',1e-3);
9  %solución numérica del robot de 2 gdl
10 disp('Advertencia: la simulación puede durar varios segundos.....')
11 disp('Espere por favor..... ')
12 [t,x]=ode45('cap7_iderobot2gdl',t,ci,opciones);
13 q1=x(:,1); q2=x(:,2);%vector de posición
14 qp1=x(:,3); qp2=x(:,4);%vector de velocidad
15 [m n]=size(t);
16 qpp1=zeros(m,1); qpp2=zeros(m,1);%vectores de aceleración
17 u1=zeros(m,1); u2=zeros(m,1);%vectores de pares aplicados
18 for k=1:m %registro de la aceleración q'' y par aplicado tau
19     xp=cap7_iderobot2gdl(t(k),[x(k,1),x(k,2),x(k,3), x(k,4)]);
20     qpp1(k,1)=xp(3,1);%información de aceleración q''_1(t)
21     qpp2(k,1)=xp(4,1);%información de aceleración q''_2(t)
22     u1(k,1) =tau1;%par aplicado tau_1(t)
23     u2(k,1) =tau2;%par aplicado tau_2(t)
24 end
25 y=qp1.*u1+qp2.*u2; %regresor y
26 %regresor psi
27 fi11=qp1.*qpp1;
28 I2=-sin(q2).*qp2.*qp1.*(qp1+qp2)+cos(q2).*qpp1.*(qp1+qp2)+cos(q2).*qp1.*(qpp1+qpp2);
29 fi13=qpp2.*((1/2)*qp2+qp1)+qp2.*((1/2)*qpp2+qp1)+qp2.*((1/2)*qp2+qpp1);
30 fi14=sin(q1).*qp1; fi15=sin(q1+q2).(qp1+qp2); fi16=qp1.*qp1; fi17=qp2.*qp2;
31 fi18=abs(qp1); fi19=abs(qp2); fi110=qp1.*(1-abs(sign(qp1))); fi111=qp2.*(1-abs(sign(qp2)));
32 fi=[fi11, fi12, fi13, fi14, fi15, fi16, fi17, fi18, fi19, fi110, fi111];
33 [r,theta] =mincuad(y,fi);
34 theta

```





### Análisis comparativo de resultados de regresión

La tabla 7.2 muestra los resultados de identificación paramétrica obtenidos a través de los esquemas de regresión dinámico, energía y potencia para el robot manipulador de 2 gdl en configuración antropomórfica.

De acuerdo a los resultados, el modelo dinámico es el que mejor estimación realiza, inclusive identifica de manera exacta los parámetros de fricción estática. Estos parámetros son particularmente difíciles de estimar debido a que la trayectoria de excitación persistente produce movimiento en todo momento y por lo tanto el fenómeno de fricción estática desaparece, es decir el término  $[1 - |\text{signo}(q_i)|] = 0$  cuando el robot está en movimiento ( $\dot{q}_i > 0$  o  $\dot{q}_i < 0$ ), entonces esta componente no mostraría variación temporal de tal forma que el algoritmo de mínimos cuadrados tendría divergencia en estos parámetros. Uno de los motivos por lo cual se obtiene convergencia paramétrica en los parámetros de fricción estática es debido al fuerte acoplamiento de la dinámica no lineal que se ve reflejado en el regresor multivariable. En contraste con los métodos escalares este acoplamiento es más débil. De ahí que la deriva en dichos parámetros es más pronunciada. El método de la potencia también en

**Tabla 7.2 Resultados de identificación paramétrica del robot de 2 gdl**

Parámetro	Valor real	Dinámico	Energía	Potencia
$\hat{\theta}_1$	2.351	2.351	2.358	2.353
$\hat{\theta}_2$	0.0838	0.0838	0.0841	0.0843
$\hat{\theta}_3$	0.1019	0.1019	0.1025	0.1017
$\hat{\theta}_4$	38.466	38.466	38.461	38.412
$\hat{\theta}_5$	1.827	1.827	1.8134	1.812
$\hat{\theta}_6$	2.288	2.288	2.282	2.522
$\hat{\theta}_7$	0.175	0.175	0.1616	0.140
$\hat{\theta}_8$	7.17	7.17	7.1082	6.865
$\hat{\theta}_9$	1.734	1.734	1.8289	1.761
$\hat{\theta}_{10}$	8.8	8.8	10	10
$\hat{\theta}_{11}$	1.87	1.87	11	11

proporciona buena exactitud en los resultados de identificación paramétrica, debido que no se realiza ningún tipo de aproximación de la aceleración. No obstante, el

requerir medir la aceleración representa una desventaja práctica. Por otro lado, el método de regresión de energía tiene pobre desempeño, realiza la identificación con mayor margen de error. Note que los dos esquemas escalares divergen en los parámetros de fricción estática

Otro aspecto importante que afecta la calidad en el proceso de identificación paramétrica es la forma de diseñar la señal de excitación persistente. Preferentemente debe tener estructuras senoidales cuyas amplitudes no saturan a los servoamplificadores.

## 7.7 Robot cartesiano de 3 gdl



El modelo dinámico del robot manipulador en configuración cartesiana de 3 gdl está formado por la matriz de masas, para gravitacional y fenómeno de fricción:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \theta_1 & 0 & 0 \\ 0 & \theta_2 & 0 \\ 0 & 0 & \theta_3 \end{bmatrix} \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + \begin{bmatrix} \theta_4 & 0 & 0 \\ 0 & \theta_5 & 0 \\ 0 & 0 & \theta_6 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + \begin{bmatrix} \theta_7 & 0 & 0 \\ 0 & \theta_8 & 0 \\ 0 & 0 & \theta_9 \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{d}_1) \\ \text{signo}(\dot{d}_2) \\ \text{signo}(\dot{d}_3) \end{bmatrix} + \begin{bmatrix} \theta_{10} \\ 0 \\ 0 \end{bmatrix}$$

donde

- $\theta_1 = m_1 + m_2 + m_3$
- $\theta_2 = m_1 + m_2$
- $\theta_3 = m_3$
- $\theta_4 = b_1$
- $\theta_5 = b_2$
- $\theta_6 = b_3$
- $\theta_7 = f_{c1}$
- $\theta_8 = f_{c2}$
- $\theta_9 = f_{c3}$
- $\theta_{10} = g(m_1 + m_2 + m_3)$





Las componentes del vector de observaciones se realiza de la línea 22 a la línea 30; el regresor completo se obtiene en las líneas 31-33. La función mínimos cuadrados multivariable se emplea en la línea 34 para obtener el vector de parámetros estimados  $\hat{\theta}$  cuyo resultado está indicado en la tabla 7.3.



### Código Fuente 7.19 cap7\_idecartesiano3gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

cap7\_idecartesiano3gdl.m

```

1 function xp = cap7_idecartesiano3gdl(t,x)
2     global tau1 tau2 tau3
3     d = [x(1); x(2); x(3)]; %vector de posición articular
4     dp = [x(4); x(5); x(6)]; %vector de velocidad articular
5     m1=0.7; m2=0.28; m3=0.28;
6     b1=0.02; b2=0.08; b3=.02;
7     fc1=0.01; fc2=0.07; fc3=.02; g=9.81;
8     theta1=m1+m2+m3; theta2=m1+m2; theta3=m3;
9     theta4=b1; theta5=b2; theta6=b3; theta7=fc1;
10    theta8=fc2; theta9=fc3; theta10=g*(m1+m2+m3);
11    %modelo dinámico del robot
12    M = [theta1, 0, 0; 0 theta2, 0; 0 0 theta3]; %matriz de inercia
13    B=[ theta4, 0, 0; 0, theta5, 0; 0, 0, theta6]; %fricción viscosa
14    Fc=[theta7, 0, 0; 0, theta8, 0; 0, 0, theta9]; %fricción de Coulomb
15    par_grav = [theta10; 0; 0]; %vector de pares de gravitacionales
16    fr= B*dp+Fc*sign(dp); %vector de fricción
17    tau1=12.36+ 0.05*sin(t);
18    tau2=(1-exp(-0.5*t))*0.13*sin(t);
19    tau3=0.06*sin(t);
20    tau=[tau1; tau2; tau3];
21    dpp = inv(M)*(tau- par_grav-fr); %vector de aceleración articular
22    xp = [dp(1); dp(2); dp(3); dpp(1); dpp(2); dpp(3)]; %vector de salida
23 end

```



### Código Fuente 7.20 cap7\_jdecartesiano3gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

#### cap7\_jdecartesiano3gdlsimu.m

```

1  clc; clear all; close all; format short g
2  global tau1 tau2 tau3
3  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
4  ci=[0; 0; 0; 0; 0; 0];%condiciones iniciales
5  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
6  %solución numérica del robot cartesiano de 3 gdl
7  [t,x]=ode45('cap7_jdecartesiano3gdl',t,ci,opciones);
8  d1=x(:,1); d2=x(:,2); d3=x(:,3);%posiciones articulares
9  dp1=x(:,4); dp2=x(:,5); dp3=x(:,6);%velocidades articulares
10 [m n]=size(t);% dimensión del vector tiempo
11 dpp1=zeros(m,1); dpp2=zeros(m,1); dpp3=zeros(m,1);%registro para d
12 u1=zeros(m,1); u2=zeros(m,1); u3=zeros(m,1);%registros para pares aplicados τ
13 for k=1:m
14     xp=cap7_jdecartesiano3gdl(t(k),[x(k,1),x(k,2),x(k,3), x(k,4), x(k,5), x(k,6)]);
15     dpp1(k,1)=xp(4,1);
16     dpp2(k,1)=xp(5,1);
17     dpp3(k,1)=xp(6,1);
18     u1(k,1) =tau1;
19     u2(k,1) =tau2;
20     u3(k,1) =tau3;
21 end
22 tau=[u1; u2; u3]; %vector de pares aplicados
23 fi11=dpp1; fi12=zeros(m,1); fi13=zeros(m,1); fi14=dp1;
24 fi15=zeros(m,1); fi16=zeros(m,1); fi17=sign(dp1);
25 fi18=zeros(m,1); fi19=zeros(m,1); fi110=ones(m,1);
26 fi21=zeros(m,1); fi22=dpp2; fi23=zeros(m,1); fi24=zeros(m,1); fi25=dp2;
27 fi26=zeros(m,1); fi27=zeros(m,1); fi28=sign(dp2); fi29=zeros(m,1); fi210=zeros(m,1);
28 fi31=zeros(m,1); fi32=zeros(m,1); fi33=dpp3;
29 fi34=zeros(m,1); fi35=zeros(m,1); fi36=dp3;
30 fi37=zeros(m,1); fi38=zeros(m,1); fi39=sign(dp3); fi310=zeros(m,1) ;
31 fi=[fi11, fi12, fi13, fi14, fi15, fi16, fi17, fi18, fi19, fi110;
32 fi21, fi22, fi23, fi24, fi25, fi26, fi27, fi28, fi29, fi210;
33 fi31, fi32, fi33, fi34, fi35, fi36, fi37, fi38, fi39, fi310];
34 theta=mincuadm(tau,fi,m,13,3);
35 theta

```



**Modelo de potencia del robot cartesiano de 3 gdl**

El modelo dinámico de potencia del robot manipulador cartesiano de 3 gdl está dado por:

$$[\tau_1 \dot{d}_1 + \tau_2 \dot{d}_2 + \tau_3 \dot{d}_3] = \begin{bmatrix} \theta_1 & 0 & 0 & \theta_4 & \theta_1 & 0 \\ 0 & \theta_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + \begin{bmatrix} \theta_7 & \theta_8 & \theta_9 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} |d_1| \\ |d_2| \\ |d_3| \end{bmatrix} + \theta_{10} \dot{d}_1$$

el modelo de potencia del robot cartesiano de 3 gdl pertenece al tipo de regresores escalares.

♣♣♣ Ejemplo 7.15

Realizar el procedimiento de identificación paramétrica de un robot cartesiano de 3 gdl usando el modelo de potencia. Utilice los parámetros del robot de 3 gdl indicados en la tabla 6.6.

El error de predicción del regresor lineal del modelo de potencia del robot cartesiano de 3 gdl está dado por:

$$e(k) = [\tau_1 \dot{d}_1 + \tau_2 \dot{d}_2 + \tau_3 \dot{d}_3] - [\psi_{11} \psi_{12} \psi_{13} \psi_{14} \psi_{15} \psi_{16} \psi_{17} \psi_{18} \psi_{19} \psi_{110}] \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \theta_{10} \end{bmatrix}$$

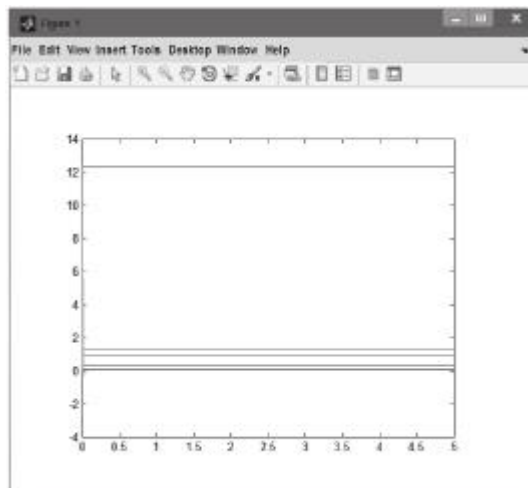




donde los componentes del regresor están compuestas por:

$$\begin{aligned}\psi_{11} &= \dot{d}_1 \ddot{d}_1 \\ \psi_{12} &= \dot{d}_2 \ddot{d}_2 \\ \psi_{13} &= \dot{d}_3 \ddot{d}_3 \\ \psi_{14} &= d_1^2 \\ \psi_{15} &= \dot{d}_2 \\ \psi_{16} &= \dot{d}_3 \\ \psi_{17} &= |\dot{d}_1| \\ \psi_{18} &= |\dot{d}_2| \\ \psi_{19} &= |\dot{d}_3| \\ \psi_{110} &= \dot{d}_1\end{aligned}$$

El programa `cap7_potenciapcartesiano3gdlsimu.m` muestra el código para realizar la identificación paramétrica de un robot cartesiano de 3 gdl empleando su modelo de potencia (ver cuadro 7.21). De la línea 15 a 22 se forman las componentes del regresor o vector de observaciones. De tal forma que dicho regresor está definido en la línea 23. La función de mínimos cuadrados versión escalar se emplea para obtener el vector de parámetros estimados  $\hat{\theta}$  (línea 24). En la línea 26 se grafica la variación temporal que tienen las componentes del vector  $\hat{\theta}$  (ver figura 7.1).



**Figura 7.1** Comportamiento de los componentes  $\hat{\theta}$


**Código Fuente 7.21 cap7\_potenciartesiano3gdlsimu.m**

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

---

**cap7\_potenciartesiano3gdlsimu.m**


---

```

1  clc; clear all; close all; format short g
2  global tau1 tau2 tau3
3  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
4  ci=[0; 0; 0; 0;0; 0]';
   opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
5  %solución numérica de la dinámica del robot cartesiano de 3 gdl
6  [t,x]=ode45('cap7_idcartesiano3gdl',t,ci,opciones);
7  d1=x(:,1); d2=x(:,2); d3=x(:,3); dp1=x(:,4); dp2=x(:,5); dp3=x(:,6);
8  [m n]=size(t); dpp1=zeros(m,1); dpp2=zeros(m,1); dpp3=zeros(m,1);
9  u1=zeros(m,1); u2=zeros(m,1); u3=zeros(m,1);
10 for k=1:m
11     xp=cap7_idcartesiano3gdl(t(k),[x(k,1),x(k,2),x(k,3), x(k,4), x(k,5), x(k,6)]);
12     dpp1(k,1)=xp(4,1); dpp2(k,1)=xp(5,1); dpp3(k,1)=xp(6,1);
13     u1(k,1) =tau1; u2(k,1) =tau2; u3(k,1) =tau3;
14 end
15 y=dp1.*u1+dp2.*u2+dp3.*u3 ;
16 fi11=dp1.*dpp1;
17 fi12=dp2.*dpp2;
18 fi13=dp3.*dpp3;
19 fi14=dp1.*dp1;
20 fi15=dp2.*dp2;
21 fi16=dp3.*dp3;
22 fi17=abs(dp1); fi18=abs(dp2); fi19=abs(dp3); fi110=dp1;
23 fi=[fi11, fi12, fi13, fi14, fi15, fi16, fi17, fi18, fi19, fi110];
24 [r,theta] =mincuad(y,fi);
25 theta %vector estimado  $\hat{\theta}$ 
26 plot(t,r) %grafica la variación temporal de los componentes del vector  $\hat{\theta}$ 

```

---



### Análisis comparativo de identificación

Los resultados de identificación paramétrica del robot cartesiano de 3 gdl de los modelos de regresión dinámico y potencia se encuentran contenidos en la tabla 7.3. Para propósitos de comparación se proporcionan los valores reales de cada uno de los componentes del vector  $\theta$ .

El modelo de regresión dinámico tiene una identificación exacta de los diez parámetros, este modelo de regresión es multivariable ya que toma como parte del regresor las señales de pares  $\tau_1, \tau_2, \tau_3$ , la matriz de regresión con las observaciones de posición, velocidad y aceleración tiene dimensión:  $\Psi \in \mathbb{R}^{3 \times 10}$ .

Por otro lado, el modelo de regresión de la potencia resulta un esquema escalar, en este caso las señales del par no forman parte del vector de mediciones de la entrada del robot, más bien, forman parte de la potencia de entrada al robot en forma escalar  $\dot{q}_1\tau_1 + \dot{q}_2\tau_2 + \dot{q}_3\tau_3$ ; se incorpora un vector de observaciones en lugar de la matriz de mediciones. Dicho vector de observaciones tiene dimensión:  $\Psi \in \mathbb{R}^{10 \times 1}$ . El esquema escalar es mucho más atractivo que la versión vectorial o multivariable debido a que su implementación es mucho más fácil.

**Tabla 7.3 Parámetros identificados del robot cartesiano de 3 gdl**

Parámetro	Valor real	Dinámico	Potencia
$\hat{\theta}_1$	1.26	1.26	1.26
$\hat{\theta}_2$	0.98	0.98	0.98
$\hat{\theta}_3$	0.28	0.28	0.28
$\hat{\theta}_4$	0.02	0.02	0.02
$\hat{\theta}_5$	0.08	0.08	0.08
$\hat{\theta}_6$	0.02	0.02	0.02
$\hat{\theta}_7$	0.01	0.01	0.01
$\hat{\theta}_8$	0.07	0.07	0.07
$\hat{\theta}_9$	0.02	0.02	0.02
$\hat{\theta}_{10}$	12.361	12.361	12.361

Ambos esquemas de regresión dinámico (multivariable) y potencia (escalar) requieren la medición de la aceleración, lo que representa una desventaja cuando el robot no dispone de los sensores adecuados para medir esa señal.



## 7.8 Resumen

**E**l método de mínimos cuadrados es una técnica de identificación paramétrica que permite obtener el valor numérico de los parámetros del sistema empleando mediciones experimentales de posiciones y velocidades articulares.

La naturaleza del modelo matemático a identificar puede ser diversa, por ejemplo: dinámico (lineal y no-lineal), estático, continuo o discreto. Sin embargo, una condición necesaria del método de mínimos cuadrados es que el modelo matemático del sistema puede ser expresado como un regresor lineal de los parámetros a identificar, es decir como el producto de una matriz de observaciones y un vector columna de parámetros desconocidos; esta estructura genera un regresor vectorial o multivariable. Ejemplos de esta categoría son los modelos dinámico y dinámico filtrado de robots manipuladores.

Cuando el regresor lineal se puede expresar como el producto de un vector transpuesto de observaciones y un vector columna de parámetros desconocidos, entonces el regresor lineal se denomina escalar. Los modelos de regresión de la energía, potencia y potencia filtrada son ejemplos del regresor lineal escalar.

Se han desarrollado dos librerías en código fuente **MATLAB** para resolver el problema de identificación paramétrica:

if

$$[r, \hat{\theta}] = \text{mincuad}(y, fi)$$

if

$$\hat{\theta} = \text{mincuadm}(y, fi, Nob, p, n)$$

Estas funciones permiten obtener la identificación paramétrica de sistemas mecatrónicos y robots manipuladores reales. La calidad de estimación paramétrica está en función del tipo de señal aplicada (excitación persistente), la exactitud de aproximar las señales de velocidad y aceleración y en la medida de lo posible no incluir en la matriz o vector de observaciones funciones discontinuas.

## Parte III Referencias selectas

Dentro de la extensa literatura sobre dinámica de sistemas mecánicos, se recomiendan ampliamente los siguientes textos para que el lector pueda profundizar en dinámica de sistemas mecatrónicos y robots manipuladores.



Dare A. Wells. "Lagrangian dynamics". Schaum's outlines 1967.



Leonard Meirovitch. "Methods of analytical dynamics". McGraw-Hill. 1970.



A. P. French. "Mecánica newtoniana". Editorial Reverté, S. A. 1978.



Wolfram Stadler. "Analytical robotics and mechatronics". McGraw Hill International Editions. 1995.



Donald T. Greenwood. "Classical dynamics". Dover Publications Inc. 1997.










Jerry B. Marion. "Dinámica clásica de las partículas y sistemas". Editorial Reverté, S. A. 2000.



H. Goldstein. "Mecánica clásica". Editorial Reverté, S. A. 2000.



G. C. Goodwin & K. S. Sin. "Adaptive filtering prediction and control". Prentice-Hall, NY. 1984.

-  M. Gautier & W. Khalil. "On the identification of the inertial parameters of robot ". Proceedings of the 27th Conference on Decision and Control. Austin, Texas, pp. 2264–2269. 1988.
-  C. Canudas and A. Aubin. "Parameters identification of robot manipulators via sequential hybrid estimation algorithms ". Proc. IFAC'90 Congress, Tallin, pp. 178–183, (1990).
-  M. Prüfer, C. Schmidt & F. Wahl. "Identification of robot dynamics with differential and integral models: a comparison ". Proceedings IEEE International Conference on Robotics and Automation. San Diego, California, Vol 1, pp. 340–345. 1994.
-  F. Reyes F. & R. Kelly. "A direct drive robot for control research". Proc. IASTED International Conference, Applications of Control and Robotics. Orlando, FL. pp. 181–184. 1996.
-  F. Reyes & R. Kelly. "Experimental evaluation of identification schemes on a direct drive robot ". Robotica. Cambridge University Press. 15, pp. 563-571. 1997.
-  F. Reyes F. & R. Kelly. "On parameter identification of robot manipulators ". Proceedings of the 1997 IEEE International Conference on Robotics and Automation. Albuquerque, New Mexico. April. pp. 1910-1915. 1997.
-  F. Reyes, & E. Lopez. "A new hamiltonian-based identification scheme for robot manipulators ". Proceedings of the 5th WSEAS-IMMCCAS. Cancun, Mexico, May 11-14. pp. 126-131. 2005.

## Parte III Problemas propuestos



**E**n esta sección se presenta una serie de ejercicios con la finalidad de que el lector mejore sus conocimientos sobre dinámica e identificación paramétrica.



### Capítulo 6 Dinámica

6.1 Considere la siguiente función de transferencia

$$Y = \frac{U}{c s^2 + 2\rho\omega_n s + \omega_n^2}$$

donde  $y \in \mathbb{R}$  es la salida del sistema,  $u \in \mathbb{R}$  es la entrada,  $\rho \in \mathbb{R}_+$  es el factor de amortiguamiento,  $\omega_n \in \mathbb{R}_+$  es la frecuencia natural de resonancia:

- (a) Obtener el modelo dinámico.
- (b) Realizar la conversión a variables de estado fase.
- (c) Escribir un programa en **MATLAB** para simular el sistema con los siguientes valores:

$$\omega_n = 1, \rho = 0,1, c = 1.$$

$$\omega_n = 1, \rho = 0,9, c = 1.$$

$$\omega_n = 1, \rho = 1,5, c = 1.$$

6.2 Considere el sistema masa resorte amortiguador, en referencia a los programas 6.1 y 6.2 (modo oscilador), modifique el valor de la entrada de la fuerza a cero Newtons, además considere condiciones iniciales diferentes a cero.

Realizar la simulación del sistema masa resorte amortiguador y analizar:

- (a) La respuesta del sistema (posición y velocidad).
- (b) Explicar el comportamiento del sistema.

Fundamente sus respuestas.

6.3 Modificar los programas 6.9 y 6.10 para simular el péndulo robot usando la siguiente entrada:

$$\tau = 5 \sin(t) + 0.33 \cos(t + 0.1 \text{random}('Normal', t, t))$$

Obtener:

- (a) Filtrado de la posición  $q_1(t)$ .
- (b) Estimación de la velocidad.

6.4 Implementar un control tipo proporcional derivativo

$$\tau_1 = \begin{bmatrix} 0 & \tilde{q}_1 & 0 \\ k_{p1} & k_{p2} & 0 \\ 0 & 0 & k_{v2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \tilde{q}_2 \\ \dot{q}_2 \end{bmatrix} + \mathbf{g}(\mathbf{q})$$

para posicionar al robot antropomórfico de 2 gdl en las siguientes configuraciones:

- (a)  $[q_{d1}, q_{d2}]^T = [45, 90]^T$  grados.
- (b)  $[q_{d1}, q_{d2}]^T = [180, 0]^T$  grados.

¿Qué método de sintonía en las ganancias proporcional derivativa puede proponer?

6.5 Implementar un control tipo proporcional derivativo saturado dado por

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_{p1} & 0 & 0 \\ 0 & k_{p2} & 0 \\ 0 & 0 & k_{p3} \end{bmatrix} \begin{bmatrix} \tanh(\tilde{q}_1) \\ \tanh(\tilde{q}_2) \\ \tanh(\tilde{q}_3) \end{bmatrix} - \begin{bmatrix} k_{v1} & 0 & 0 \\ 0 & k_{v2} & 0 \\ 0 & 0 & k_{v3} \end{bmatrix} \begin{bmatrix} \tanh(\dot{q}_1) \\ \tanh(\dot{q}_2) \\ \tanh(\dot{q}_3) \end{bmatrix} + \mathbf{g}(\mathbf{q})$$

para controlar las posiciones articulares de la base, hombro y codo del robot antropomórfico de 3 gdl en las siguientes configuraciones:

- (a)  $[q_{d1}, q_{d2}, q_{d3}]^T = [45, 90, 90]^T$  grados.
- (b)  $[q_{d1}, q_{d2}, q_{d3}]^T = [45, 180, 0]^T$  grados.

¿Qué método de sintonía en las ganancias proporcional derivativa puede proponer, para el robot de 3 gdl?



6.6 Realizar un programa en **MATLAB** para controlar al robot cartesiano de 3

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_{p1} & 0 & 0 \\ 0 & k_{p2} & 0 \\ 0 & 0 & k_{p3} \end{bmatrix} \begin{bmatrix} \text{atan}(\tilde{q}_1) \\ \text{atan}(\tilde{q}_2) \\ \text{atan}(\tilde{q}_3) \end{bmatrix} - \begin{bmatrix} k_{v1} & 0 & 0 \\ 0 & k_{v2} & 0 \\ 0 & 0 & k_{v3} \end{bmatrix} \begin{bmatrix} \text{atan}(\dot{q}_1) \\ \text{atan}(\dot{q}_2) \\ \text{atan}(\dot{q}_3) \end{bmatrix} + \mathbf{g}(\mathbf{q})$$

en las siguientes configuraciones:

(a)  $[d_{d1}, d_{d2}, d_{d3}]^T = [0.5, 0.3, .65]^T$  m.

(b)  $[d_{d1}, d_{d2}, d_{d3}]^T = [0.8, 0.2, 0.34]^T$  m.

¿Qué método de sintonía en las ganancias proporcional derivativa puede proponer, para el robot cartesiano?



### Capítulo 7 Identificación paramétrica

7.1 Sean los siguientes sistemas dinámicos escalares:

$$\begin{aligned} \dot{x} &= -3x + 8 \sin(x) + 8.98x^3 \\ \dot{y} &= 6.66y + 5.61e^{-|y|} + 1.23 \sinh(y) \\ \dot{z} &= 2.345z^3 + 6.1234z^4 - 2.31z^5 \end{aligned}$$

Para cada uno de los sistemas propuestos implementar un esquema de identificación paramétrica para obtener el valor de sus respectivos parámetros. Tome un intervalo para  $y \in [-10, 10]$ .

7.2 Sean los siguientes sistemas multivariables:

$$\begin{aligned} \dot{x}_1 &= 0 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3 + \text{sen}(t) \\ \begin{bmatrix} \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} -3 & -4 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \end{bmatrix} \text{sen}(t) \\ \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} 1 & 4 & -6 \\ 0.01 & 3.1 & 4.1 \\ 2.2 & 1.11 & -8.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 9 \\ 3.2 \text{sen}(t) \\ 2.1 \text{sen}(t) \cos(t) \end{bmatrix} \\ \dot{x}_4 &= -0.09 \cdot x_4 + 2.33 \cdot x_1 + 1.21 \cdot x_2 + 4.55 \cdot x_3 + \text{sen}^3(t) \end{aligned}$$

para un intervalo de  $t \in [0, 5]$ .

7.3 Considere el siguiente modelo matemático:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 0.33t^3 + 0.24 \sin(t) + \cos(t) + 0.35t^2 - 4.56 \sinh(t) \\ 0.33t^1 + 0.48 \sin^3(t) + 0.66 \cos(t) + 0.70t^2 + 0.33 \cos(t) \\ 0.66t^4 + 0.24 \sin^4(t) + 0.99 \cos(t) + 0.35t^2 - 0.66 \tan(t) \\ 0.99t^5 + 0.72 \sin^5(t) + 0.48 \cos(t) + 0.24t^2 + 0.33 \operatorname{atan}(t) \\ 0.33t^3 + 0.24 \sin(t) + \cos(t) + 0.48t^2 - 0.24 \sin(t) \end{bmatrix}$$

Seleccione un vector de parámetros  $\theta$  conveniente y diseñe un esquema de regresión para llevar a cabo el proceso de identificación paramétrica para un intervalo de  $t \in [0, 5]$ .

7.4 Considere los esquemas de regresión modelo dinámico filtrado y potencia filtrada del péndulo

- En el regresor del modelo dinámico filtrado sustituir la técnica de filtrado para obtener la aceleración  $\ddot{q}_1$  por el método de Euler:  $\ddot{q}_1 \approx \frac{\dot{q}_1(k) - \dot{q}_1(k-1)}{h}$  y  $\dot{q}_1 \approx \frac{q_1(k) - q_1(k-1)}{h}$ . Llevar a cabo el proceso de identificación paramétrica y comparar los resultados con el método de filtrado.
- En el regresor del modelo de potencia filtrada sustituir la técnica de filtrado para obtener la aceleración  $\ddot{q}_1$  por el método de Euler:  $\ddot{q}_1 \approx \frac{\dot{q}_1(k) - \dot{q}_1(k-1)}{h}$  y  $\dot{q}_1 \approx \frac{q_1(k) - q_1(k-1)}{h}$ . Llevar a cabo el proceso de identificación paramétrica y comparar los resultados con el método de filtrado.

7.5 Considere un brazo robot antropomórfico de 2 gdl:

- Implementar el esquema de regresión lineal del modelo dinámico filtrado.
- Implementar el esquema de regresión lineal del modelo de la potencia filtrada.

7.6 Considere un robot manipulador en configuración cartesiana de 3 gdl:

- Implementar el esquema de regresión lineal dinámico filtrado.
- Implementar el esquema de regresión lineal del modelo de energía.
- Implementar el esquema de regresión lineal del modelo de la potencia filtrada.

# Parte IV

## Control

Control de robots manipuladores es un tema vigente en la actualidad y de interés para la comunidad científica. El diseño de sistemas de control en aplicaciones industriales representa no sólo un reto teórico, también resuelve sustancialmente problemas prácticos para mejorar deficiencias de algoritmos de control tradicionales. La **Parte IV** está dedicada a presentar aplicaciones de la metodología de **moldeo de energía** la cual hoy en día representa una técnica moderna de diseño de algoritmos de control para robots manipuladores.

La **Parte IV** se compone del **Capítulo 8 Control**



**Capítulo 8 Control de posición** contiene aplicaciones de algoritmos de control diseñados directamente por la técnica de moldeo de energía, y además presenta una explicación cualitativa de esquemas clásicos como lo son el control proporcional derivativo (PD) y el control proporcional integral derivativo (PID); también se incluye nuevas estrategias de control como el control tangente hiperbólico y arco tangente.

La Parte IV finaliza con:



**Referencias selectas**



**Problemas propuestos**

# 8

## Capítulo

# Control de posición

$$\tau = \nabla U_a(K_p, \tilde{\mathbf{q}}) - \mathbf{f}_v(K_v, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

**8.1 Introducción**

**8.2 Control proporcional derivativo (PD)**

**8.3 Control proporcional integral derivativo (PID)**

**8.4 Control punto a punto**

**8.5 Resumen**

## Objetivos

Presentar aplicaciones de la técnica de moldeo de energía de robots manipuladores.

### Objetivos particulares:



Control proporcional derivativo.



Control proporcional integral derivativo.



Control tangente hiperbólico.



Control arcotangente.



Control punto a punto.

## 8.1 Introducción

**C**ontrol de robots manipuladores es un tema de control automático vigente y de gran interés para la comunidad científica de robótica debido a los retos teóricos y prácticos que involucra el diseño de nuevas estrategias de control con alto desempeño y exactitud en aplicaciones industriales tales como estibado de cajas, ensamble, traslado, pintado de objetos, etc. El diseño de nuevos esquemas de control implica mejorar sustancialmente el desempeño de algoritmos de control tradicionales.

El problema más simple de control de robots manipuladores se denomina control de posición o regulación el cual se refiere a ubicar o posicionar al extremo final del robot en un punto constante en el tiempo al que se le denomina posición deseada o set point. El objetivo de control es posicionar al extremo final del robot en el punto deseado y que permanezca ahí de manera indefinida.

El problema de control de posición o regulación consiste en mover el extremo final del robot manipulador hacia una posición deseada constante  $\mathbf{q}_d$ ; independientemente de su posición inicial  $\mathbf{q}(0)$ . Los puntos intermedios entre la posición inicial y la referencia deseada no son controlados como tal, forman parte de la etapa transitoria de la respuesta del robot incluido en la señal de error  $\mathbf{q}(\tilde{t})$  definida como la diferencia entre la posición deseada y la posición actual del robot  $\tilde{\mathbf{q}}(t) = \mathbf{q}_d - \mathbf{q}(t)$ . El punto final  $\mathbf{q}_d$  es la posición clave donde se realiza el control.

El objetivo del problema de control de posición está determinado por encontrar una ley de control  $\tau$  que proporcione los pares aplicados a las articulaciones o servomotores del robot, de tal forma que la posición actual del robot  $\mathbf{q}(t)$  y la velocidad articular de movimiento  $\dot{\mathbf{q}}(t)$  tiendan asintóticamente hacia la posición deseada  $\mathbf{q}_d$  y velocidad cero, respectivamente, sin importar las condiciones iniciales. Es decir,

$$\lim_{t \rightarrow \infty} \mathbf{q}(t) = \mathbf{q}_d \quad (8.1)$$

El objetivo de control se cumple cuando:

$$\lim_{t \rightarrow \infty} \begin{matrix} \dot{\mathbf{q}}(t) \\ \tilde{\mathbf{q}}(t) \end{matrix} = \begin{matrix} \mathbf{0} \\ \mathbf{0} \end{matrix}$$



**Figura 8.1** Control de posición

Nótese que en la figura 8.1 el robot se encuentra en su punto de equilibrio, lo que significa que el objetivo de control (8.1) se cumplió sin depender de las condiciones iniciales, entonces la posición deseada se alcanza, por lo que la posición del extremo final del robot permanece constante ( $\mathbf{q}(t) = \mathbf{q}_d$ ) y por lo tanto la velocidad de movimiento es cero ( $\dot{\mathbf{q}} = \mathbf{0}$ ).

Evidentemente, para propósitos industriales, una vez que el extremo final del robot ha alcanzado el punto deseado, en el siguiente periodo de muestreo el punto deseado cambiaría su valor, entonces el robot se movería hacia al siguiente punto; el anterior punto deseado representa la condición inicial y el extremo final del robot se estaría posicionando en la siguiente coordenada; así sucesivamente el robot sigue una curva compuesta de puntos cercanos entre sí. Esta forma de controlar a un robot se denomina **control punto a punto**. Para realizar esta aplicación es necesario que el esquema de control forme parte de la ecuación en lazo cerrado y que genere un punto de equilibrio asintóticamente estable, ya que no dependería de las condiciones iniciales. Es importante aclarar que en control punto a punto no se controla la



velocidad de movimiento como en el caso de control de trayectoria, donde el error de posición y error de velocidad son controlados simultáneamente. La gran mayoría de las aplicaciones actuales de la industria son realizadas a través de control punto a punto como en los casos de traslado y pintado de objetos, estibado y pintura de carrocerías, etcétera.

El **desempeño** de un algoritmo de control se refiere a realizar de manera correcta y exacta la tarea programada al robot, lo que lo habilita a llevar a cabo diversas aplicaciones de control punto a punto. Por lo tanto, el espectro de aplicaciones comerciales, domésticas, científicas e industriales se incrementa.

A continuación se describen los algoritmos tradicionales de control de robots manipuladores como lo son el esquema proporcional derivativo (PD) y el regulador proporcional integral derivativo (PID). Posteriormente se presentan nuevas estrategias de control usando la técnica de moldeo de energía.

## 8.2 Control proporcional-derivativo (PD)

**E**l algoritmo de control proporcional-derivativo (PD) es el esquema de controlador más simple y más popular que se puede usar en robots manipuladores. En 1981, Takegaki y Arimoto demostraron que la ecuación en lazo cerrado del modelo dinámico del robot manipulador y el control proporcional derivativo tiene un punto de equilibrio global y asintóticamente estable.

El control proporcional derivativo más compensación de gravedad (PD) está dado por la siguiente ecuación:

$$\boldsymbol{\tau} = \mathbf{K}_p \tilde{\mathbf{q}} - \mathbf{K}_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (8.2)$$

donde  $\tilde{\mathbf{q}} \in \mathbb{R}^n$  es el vector de error de posición que se define como la diferencia entre la posición deseada  $\mathbf{q}_d \in \mathbb{R}^n$  y la posición actual del robot  $\mathbf{q} \in \mathbb{R}^n$ ;  $\mathbf{K}_p \in \mathbb{R}^{n \times n}$  es una matriz definida positiva,  $\mathbf{K}_v \in \mathbb{R}^{n \times n}$  es la ganancia derivativa la cual es una matriz definida positiva.

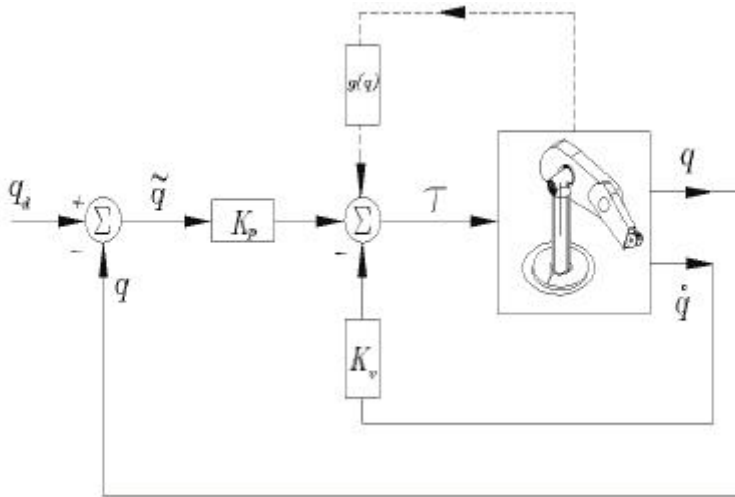
La ecuación en lazo cerrado en variables de estado que definen el problema de control

de posición es:

$$\frac{d}{dt} \ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1} [\mathbf{K}_p \mathbf{q} - \mathbf{K}_v \dot{\mathbf{q}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}] \quad (8.3)$$

la cual es una ecuación diferencial autónoma.

El diagrama a bloques del control proporcional derivativo con compensación de gravedad se muestra en la figura 8.2.



**Figura 8.2** Diagrama a bloques del control PD.

La posición articular del robot  $\mathbf{q}$  se retroalimenta para generar la señal de error de posición  $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ . La velocidad de movimiento  $\dot{\mathbf{q}}$  se emplea para inyección de amortiguamiento. Obsérvese que el signo menos de la acción de control derivativa sirve para contrarrestar la energía al control proporcional. El término de compensación de gravedad  $\mathbf{g}(\mathbf{q})$  es una componente esencial para la unicidad del punto de equilibrio. Sin embargo, esta condición no es suficiente, también se requiere que las ganancias proporcional y derivativa sean matrices definidas positivas.

Desde el punto de vista práctico el término  $\mathbf{g}(\mathbf{q})$  se requiere en la implementación del control PD, y se debe conocer como parte de la dinámica parcial del robot si el movimiento es en un plano vertical o en general en su espacio tridimensional. Para el caso en que el movimiento sea en un plano horizontal, la energía potencial es

constante y el gradiente de la energía potencial o par gravitacional  $\mathbf{g}(\mathbf{q}) = \frac{\partial U}{\partial \mathbf{q}}$  es cero.



### Control PD de un péndulo

La forma inmediata de entender el funcionamiento cualitativo del control PD es a través del péndulo.

El modelo dinámico de un péndulo está dado por:

$$\tau_1 = [I_r + m_1 l_c^2] \ddot{q}_1 + b \dot{q}_1 + m_1 g l_c \sin(q_1)$$

mientras que el control proporcional derivativo es:

$$\tau_1 = k_p \tilde{q}_1 - k_v \dot{q}_1 + m_1 g l_c \sin(q_1)$$

donde el término  $k_p \tilde{q}_1$  corresponde al regulador proporcional; siendo el error de posición  $\tilde{q}_1$  definido como la diferencia entre la posición deseada  $q_{d1}$  y la posición actual del péndulo  $q_1(t)$ . El nombre del esquema proporcional se debe al producto de la ganancia  $k_p$  y el error de posición  $\tilde{q}$ .

Por otro lado, el término  $-k_v \dot{q}_1$  se denomina acción de control derivativo y no debe ser confundido con un esquema de control; este término se incluye con la finalidad de mejorar la respuesta del control proporcional a través de la inyección de amortiguamiento que se logra restando al control proporcional el producto entre la ganancia derivativa  $k_v$  y la velocidad articular  $\dot{q}_1$ . En términos prácticos el efecto de amortiguamiento significa freno mecánico para dosificar la energía aplicada al servomotor  $\tau_1$  y de esta forma lograr que la respuesta transitoria del péndulo no tenga sobre impulsos y entre de manera suave al régimen estacionario.

Por lo tanto, la ecuación en lazo cerrado está determinada de la siguiente forma:

$$\frac{d}{dt} \tilde{q}_1 = \frac{-\dot{q}_1}{[I_r + m_1 l_c^2]} [k_p \tilde{q}_1 - k_v \dot{q}_1 - b \dot{q}_1]$$

se ha eliminado la fricción de Coulomb y estática con la finalidad que los programas de simulación no demoren dicho proceso.

### ♣ Ejemplo 8.1

Controlar la posición del extremo final del péndulo en el siguiente punto deseado:  
 $q_{d1} = 90$  grados

### Solución

La estructura matemática del algoritmo de control proporcional derivativo del péndulo está dada por la siguiente expresión:

$$\tau_1 = k_{p1}\tilde{q}_1 - k_{v1} + m_1glc_1 \sin(q_1)$$

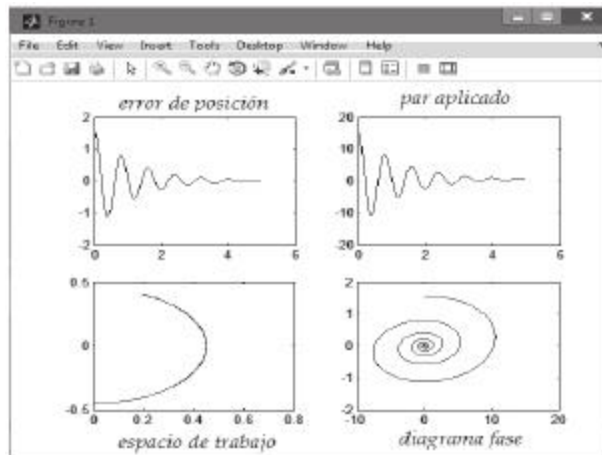
donde  $k_{p1}$  es la ganancia proporcional;  $k_{v1}$  representa la ganancia derivativa,  $\tilde{q}_1$  es el error de posición definido como la diferencia  $\tilde{q}_1 = q_{d1} - q_1(t)$  entre la posición deseada  $q_{d1}$  y la posición actual del robot  $q_1(t)$ ;  $\tau_1$  es el par aplicado al servomotor del péndulo.

El cuadro 8.1 contiene el modelo dinámico del péndulo `cap8_pendulo.m`; observe que en la línea 18 se utiliza el esquema de control proporcional derivativo  $[\tau_1, \sim] = \text{cap8\_pdpendolo}(q_1, q_{p1})$ , el cual proporciona la energía  $\tau_1$  necesaria para mover al péndulo desde la posición inicial  $q(0)$  hacia la posición deseada  $q_{d1} = 90$  grados.

El programa `cap8_pdpendolo.m` tiene el código fuente en **MATLAB** del control proporcional derivativo, el cual se encuentra en el cuadro 8.2 y a través del programa principal `cap8_pdpendulosimu.m` (ver cuadro 8.3) se puede llevar a cabo el proceso de simulación.

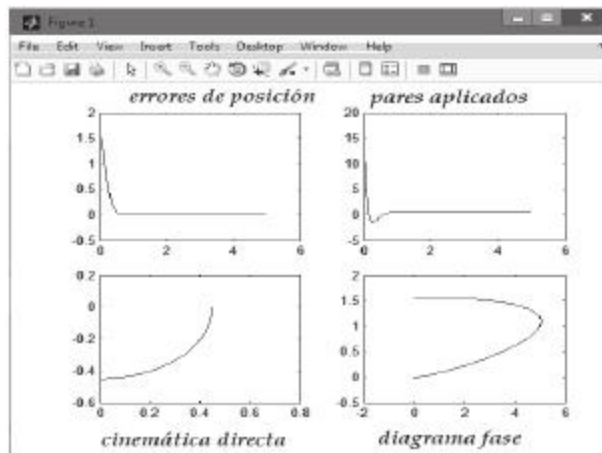
En una primera fase de la simulación las ganancias del control (PD) se han sintonizado con los siguientes valores:  $k_{p1} = 10$  y  $k_{v1} = 0.1$ ; lo que significa que la inyección de amortiguamiento del término derivativo es pobre, causa sobre impulsos en la etapa transitoria y fluctuación abrupta de la respuesta, marcando oscilaciones sucesivas y sostenidas en la etapa transitoria como se muestra en las gráficas del error de posición y par aplicado de la figura 8.3. Note que en el espacio de trabajo (cinemática directa) también en la trayectoria del extremo final del péndulo tiene oscilaciones; lo que se confirma por el diagrama fase ( $q_1$  vs  $\dot{q}_1$ ) en que el atractor del

punto de equilibrio tiene un conjunto de espirales que van disminuyendo conforme las senales  $q_1(t)$  y  $\tilde{q}(t)$  evoluciona el tiempo  $t$ .



**Figura 8.3** Péndulo: control PD,  $k_{p1} = 10$ ,  $k_{v1} = 0.1$

La segunda fase de la simulación consiste en cambiar el valor de la ganancia derivativa  $k_{v1} = 2$  (manteniendo el valor de la ganancia proporcional), es decir aumentar el efecto de amortiguamiento o freno mecánico para obtener una respuesta más suave en régimen transitorio, y en estado estacionario libre de oscilaciones. La figura muestra la respuesta del péndulo con la nueva sintonía de ganancia derivativa.



**Figura 8.4** Péndulo: control PD,  $k_p = 10$ ,  $k_v = 2$



### Código Fuente 8.1 cap8\_pendolo.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

cap8\_pendolo.m

```

1 function xp=cap8_pendolo(t,x)
2     %modelo dinámico del péndulo
3     %no se incluye el fenómeno de fricción de Coulomb, ni estática
4     %variables de estados (entradas) -----%
5     %posición articular
6     q1=x(1);
7     %velocidad articular
8     qp1=x(2);
9     %parámetros del péndulo -----%
10    m1=5;%masa
11    lc1=0.01;%centro de gravedad
12    g=9.81;%constante de aceleración gravitacional
13    b1=0.17;%coeficiente de fricción viscosa
14    lr1=0.16;%momento de inercia del rotor
15    lp=lr1+m1*lc1*lc1;%momento de inercia del péndulo
16    %en este punto se inserta el algoritmo de control
17    %par aplicado (controlador) -----%
18    [tau1, ~ ] = cap8_pdpendolo(q1,qp1) ;
19    %aceleración articular del péndulo -----%
20    qpp1=(tau1-b1*qp1-m1*g*lc1*sin(q1))/lp;
21    %vector de salida -----%
22    xp=[qp1;%xp(1)=x(2) velocidad articular
23        qpp1;%xp(2)=qpp aceleración articular
24 end

```



### Código Fuente 8.2 cap8\_pdpendolo.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

Esta función retorna:

error de posición  $\tilde{q}_1$

par aplicado  $\tau_1$

cap8\_pdpendolo.m

```

1 function [tau1, qt1] = cap8_pdpendolo(x,xp)
2     %variables de estados (entradas) -----%
3     %posición articular
4     q1=x;
5     %velocidad articular
6     qp1=xp;
7     %parámetros de la compensación de gravedad del péndulo
8     m1=5;%masa
9     lc1=0.01;%centro de gravedad
10    g=9.81;%constante de aceleración gravitacional
11    %ganancias del controlador -----%
12    kp1=10;%ganancia proporcional
13    kv1=2;%ganancia derivativa
14    %referencia (posición deseada en grados)
15    qd1=90;
16    %vector de pares de gravitacionales -----%
17    par_grav = m1*g*lc1*sin(q1);
18    %error de posición
19    qt1=pi*qd1/180-q1;
20    %control proporcional derivativo
21    tau1=kp1*qt1-kv1*qp1+par_grav;
22 end

```



### Código Fuente 8.3 cap8\_pdependulosimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 7 Identificación paramétrica.

---

cap8\_pdependulosimu.m

---

```

1  clc; clear all; close all;
2  format short g
3  %parámetros de simulación _____-%
4  ti=0;%tiempo inicial (segundos)
5  tf=5;%tiempo de final (segundos)
6  h=0.001;%incremento de tiempo (segundos)
7  ts=ti:h:tf;%vector de tiempo
8  ci=[0;0];%condiciones iniciales
9  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
10 %solución numérica del sistema _____-%
11 [t,x]=ode45('cap8_pendolo',ts,ci,opciones);
12 q1=x(:,1);
13 qp1=x(:,2);
14 [n,m]=size(t);
15 tau1=zeros(n,1);
16 qt1=zeros(n,1);
17 xef=zeros(n,1);
18 yef=zeros(n,1);
19 for k=1:n
20     [tau1(k), qt1(k)] = cap8_pdependulo(q1(k),qp1(k));
21     frq=Rz(-pi/2)*cinematica_pendolo(0.45,q1(k),0.1);
22     xef(k)=frq(1);
23     yef(k)=frq(2);
24 end
25 subplot(2,2,1); plot(t,qt1)
26 subplot(2,2,2); plot(t, tau1)
27 subplot(2,2,3); plot(xef,yef)
28 subplot(2,2,4); plot(qp1, qt1)

```

---





### Control PD de un brazo robot de 2 gdl

El control proporcional derivativo (PD) para un robot en configuración antropomórfica de 2 gdl está dado por la siguiente ecuación:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \underbrace{\begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix}}_{K_p} \begin{bmatrix} \tilde{q}_1 \\ \tilde{q}_2 \end{bmatrix} - \underbrace{\begin{bmatrix} k_{v1} & 0 \\ 0 & k_{v2} \end{bmatrix}}_{K_v} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \mathbf{g}(\mathbf{q})$$

donde  $K_p, K_v \in \mathbb{R}^{2 \times 2}$  son la ganancia proporcional y derivativa, respectivamente; la velocidad articular es representada por  $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2]^T$ ; el error de posición  $\mathbf{q}$  se define como la diferencia entre la posición deseada  $\mathbf{q}_d$  y la posición actual del robot  $\mathbf{q}$ ;  $\mathbf{g}(\mathbf{q})$  es la compensación de gravedad;  $\tau_1, \tau_2$  son los pares aplicados a las articulaciones del hombro y codo, respectivamente.

#### ♣ ♣ Ejemplo 8.2

Controlar la posición del extremo final de un robot de 2 gdl en configuración antropomórfica en el siguiente punto deseado:  $[\mathbf{q}_{d1}, \mathbf{q}_{d2}]^T = [90, 45]$  grados.

#### Solución

El cuadro 8.8 contiene el modelo dinámico del brazo robot de 2 gdl (programa cap8\_robot2gdl.m); en referencia a este programa es muy importante aclarar que la ecuación en lazo cerrado está formada por la dinámica no lineal del robot y el esquema de control (ver línea 18). Por lo tanto, en la línea 22 es el lugar adecuado para insertar el algoritmo de control, en este caso es el proporcional derivativo cuyo código **MATLAB** se presenta en el programa cap8\_PDrobot2gdl.m (ver cuadro 8.8).

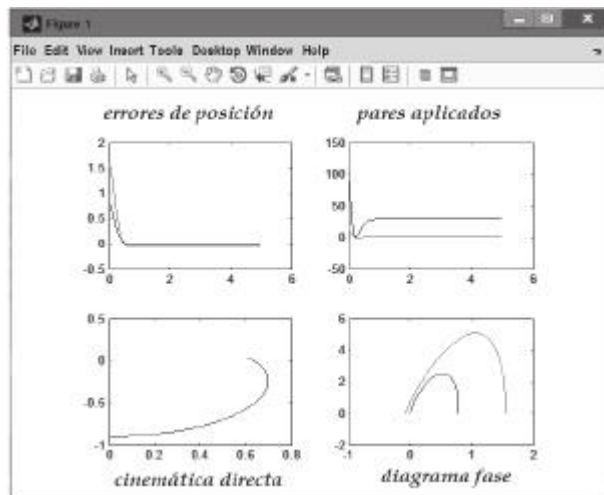
Las ganancias proporcional  $K_p$  y derivativa  $K_v$  son sintonizadas de tal forma que no produzcan sobre impulsos, ni oscilaciones en el régimen transitorio, así como una respuesta suave en estado estacionario.

El valor de la sintonía de las ganancias proporcional para las articulaciones del hombro y codo, respectivamente son:  $k_{p1}=3$  Nm/grados y  $k_{p2}=0.15$  Nm/grado;

mientras que las ganancias derivativas has sido seleccionadas con un factor de amortiguamiento del 20% del valor de las ganancias proporcional:  $k_{v1} = 0.20k_p1$  Nm-seg/grados y  $k_{v2} = 0.20k_p2$  Nm-seg/grados.

La figura 8.5 muestra un perfil suave de los errores de posición, sin sobre impulsos, no hay oscilaciones pronunciadas, teniendo una respuesta suave; ambos componentes de error de posición tienden asintóticamente a cero como el tiempo evoluciona a infinito (estabilidad asintótica).

Las señales de pares aplicados  $\tau_1$  y  $\tau_2$  se aplican directamente a las articulaciones del hombro y codo, respectivamente; el perfil de estas señales muestran de igual forma un comportamiento suave, sin oscilaciones que hace que el robot no vibre tal como se aprecia en la trayectoria que describe el extremo final del robot en su espacio de trabajo, la cual inicia desde la posición de casa ubicada en el eje  $y_-$  hasta el punto deseado. Observe también que en el diagrama fase del robot las velocidades articulares como las señales de error de posición convergen asintóticamente al punto de equilibrio. Este tipo de respuesta es consecuencia de una adecuada sintonía de las ganancias, así como del factor de amortiguamiento, que en este caso es del 20%. La sintonía de las ganancias es empírica, es decir a prueba y error; este proceso depende en gran medida de la experiencia que tenga el usuario.



**Figura 8.5** Robot de 2 gdl: control PD,  $k_p = 10$ ,  $k_v = 0.1$

**Código Fuente 8.8 cap8\_robot2gdl.m**

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

**cap8\_robot2gdl.m**

```

1 function xp = cap8_robot2gdl(t,x)
2     %vector de posición articular
3     q1=x(1); q2=x(2); q = [q1; q2];
4     %vector de velocidad articular
5     qp1=x(3); qp2=x(4); qp = [qp1; qp2];
6     m1=23.902; l1=0.45; lc1=0.091; l1=1.266; b1=2.288;
7     m2=3.880; l2=0.45; lc2=0.048; l2=0.093; b2=0.175; g=9.81;
8     theta1=m1*lc1*lc1+m2*l1*l1+m2*lc2*lc2+l1+l2;
9     theta2=l1*m2*lc2;
10    theta3=m2*lc2*lc2+l2;
11    theta4=g*(lc1*m1+m2*l1);
12    theta5=g*m2*lc2; theta6=b1; theta7=b2;
13    M=[theta1+2*theta2*cos(q2), theta3+theta2*cos(q2);
14    theta3+theta2*cos(q2), theta3];
15    C=[ -2*theta2*sin(q2)*qp2, -theta2*sin(q2)*qp2;
16    theta2*sin(q2)*qp1, 0];
17    gq11=theta4*sin(q1)+theta5*sin(q1+q2);
18    gq21=theta5*sin(q1+q2);
19    gq=[gq11; gq21];
20    fr=[theta6*qp1;
21    theta7*qp2];
22    [ ~ , tau ]=cap8_PDrobot2gdl(q,qp); %en esta línea se inserta el controlador
23    qpp = M^(-1)*(tau-C*qp-gq-fr);
24    %vector de salida
25    xp = [qp1; qp2; qpp(1); qpp(2)];
26 end

```



### Código Fuente 8.8 cap8\_PDrobot2gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_PDrobot2gdl.m

```

1 function [qtilde tau] = cap8_PDrobot2gdl(x, xp)
2     q1=x(1); q2=x(2); q = [q1; q2]; %vector de posiciones
3     qp1=xp(1); qp2=xp(2); qp = [qp1; qp2]; %vector de velocidades
4     %parámetros del robot: par gravitacional
5     m1=23.902; lc1=0.091; l1=0.45;
6     m2=3.880; lc2=0.048; g=9.81;
7     theta4=g*(lc1*m1+m2*l1);
8     theta5=g*m2*lc2;
9     %par gravitacional
10    par_grav = [theta4*sin(q1)+theta5*sin(q1+q2);
11               g*m2*lc2];
12    %ganancia proporcional
13    kp1=3; kp2=0.15;
14    Kp=[kp1, 0;
15        0, kp2]; %ganancia proporcional
16    %ganancia derivativa
17    kv1=0.20*kp1; kv2=0.20*kp2;
18    Kv=[kv1, 0;
19        0, kv2]; %ganancia derivativa
20    qd1=45; qd2=90;
21    qd=[qd1;
22        qd2]; %vector de referencias
23    qtilde=pi/180*qd-q;
24    qtgrados=(180/pi)*qtilde; %error de posición en grados
25    qpgrados=180*qp/pi; %velocidad en grados/segundo.
26    tau=Kp*qtgrados-Kv*qpgrados+par_grav;
27 end

```



### Código Fuente 8.6 cap8\_robot2gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_robot2gdlsimu.m

```

1  clc; clear all; close all;
2  format short g
3  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
4  ci=[0; 0; 0; 0];%condiciones iniciales
5  opciones=odeset('RelTol',1e-3,'InitialStep', 1e-3,'MaxStep',1e-3);
6  %solución numérica del sistema dinámico lineal
7  [t,x]=ode45('cap8_robot2gdl',t,ci,opciones);
8  q1=x(:,1); q2=x(:,2);
9  qp1=x(:,3); qp2=x(:,4);
10 [n,m]=size(t);
11 tau1=zeros(n,1); tau2=zeros(n,1);
12 qtilde1=zeros(n,1); qtilde2=zeros(n,1);
13 xef=zeros(n,1); yef=zeros(n,1);
14 beta1=0.15; beta2=0.15; l1=0.45; l2=0.45;
15 for k=1:n
16     [qt tau] = cap8_PDrobot2gdl( [q1(k);q2(k)], [qp1(k); qp2(k)]);
17     tau1(k)=tau(1); tau2(k)=tau(2);
18     qtilde1(k)=qt(1); qtilde2(k)=qt(2);
19     [xef(k), yef(k), ~ ]=cinematica_2gdl( beta1, l1, q1(k), beta2, l2, q2(k));
20     p=Rz(-pi/2)*[xef(k), yef(k), beta1+beta2]';
21     xef(k)=p(1);
22     yef(k)=p(2);
23 end
24 subplot(2,2,1); plot(t,qtilde1,t,qtilde2)
25 subplot(2,2,2); plot( t, tau1, t,tau2)
26 subplot(2,2,3); plot(xef,yef)
27 subplot(2,2,4); plot(qtilde1, qp1, qtilde2,qp2)

```



### Control PD de un brazo robot de 3 gdl

El algoritmo de control proporcional derivativo para un brazo robot de 3 gdl está dado por:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 & k_{p2} & 0 \\ 0 & 0 & k_{p3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \tilde{q}_2 \\ \tilde{q}_3 \end{bmatrix} - \begin{bmatrix} 0 & k_{v2} & 0 \\ 0 & 0 & k_{v3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_3 \end{bmatrix} + \mathbf{g}(\mathbf{q})$$

donde  $K_p, K_v \in \mathbb{R}^{3 \times 3}$  son la ganancia proporcional y derivativa, respectivamente; el error de posición  $\tilde{\mathbf{q}} \in \mathbb{R}^3$  se define como la diferencia entre la posición deseada  $\mathbf{q}_d \in \mathbb{R}^3$  y la posición actual del robot  $\mathbf{q} \in \mathbb{R}^3$ .

Por conveniencia se seleccionan las ganancias proporcional y derivativa como matrices diagonales definidas positivas con la finalidad de desacoplar la estructura de control; de esta forma la sintonía de la  $i$ -ésima ganancia sólo depende de la  $i$ -ésima articulación. Es decir, el par aplicado a la  $i$ -ésima articulación depende de:  $\tau_i = \tau(k_{pi}, \tilde{q}_i, \dot{q}_i)$ .

#### ♣ Ejemplo 8.3

Controlar la posición del extremo final de un robot de 3 gdl en configuración antropomórfica en las siguientes coordenadas:  $[q_{d1}, q_{d2}, q_{d3}]^T = [30, 45, 90]$  grados

#### Solución

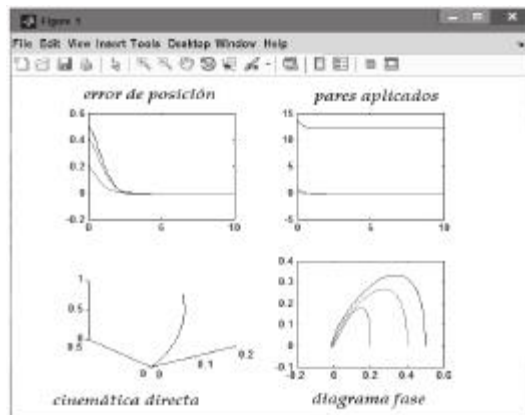
El modelo dinámico del robot antropomórfico de 3 gdl se encuentra en el programa `cap8_robot3gdl.m` como se describe en el cuadro 8.7. De particular interés representa la línea de código 23, debido a que en este lugar se inserta el algoritmo de control que envía la señal de par aplicado para que el robot se mueva a la posición indicada.

El esquema de control proporcional derivativo se encuentra implementado en el

programa cap8\_PDrobot3gdl.m del cuadro 8.8. La sintonía de las ganancias han sido pensadas en la capacidad de par aplicado que puedan suministrar las articulaciones del robot; es decir, la idea es no saturar a los servoamplificadores de tal forma que se encuentren trabajando en la parte lineal, lejos de los límites de saturación.

Supóngase que se cuenta con un prototipo de un robot manipulador de 3 gdl, cuyos servomotores corresponden a las siguientes características: 50 Nm, 150 Nm y 15 Nm para las articulaciones de la base, hombro y codo, respectivamente. El criterio para sintonizar las ganancias es el siguiente:  $k_{pi} \leq 0.80 \frac{\tau_{\text{max}}}{\tau_{\text{eqi}}}$ , donde  $\tau_{\text{max}}$  corresponde al par máximo de la  $i$ -ésima articulación. En la línea 18 del programa cap8\_PDrobot3gdl.m se encuentra la sintonía final de la ganancia proporcional; el factor de amortiguamiento de la ganancia derivativa se ha seleccionado en el 20% de la ganancia proporcional.

El código fuente que contiene la simulación del control proporcional derivativo del robot manipulador en configuración antropomórfica se encuentra en el programa cap8\_robot3gdlsimu.m (cuadro 8.9). La figura 8.6 muestra el comportamiento de los errores de posición para las tres articulaciones; presentan un perfil suave, sin ruido mecánico, ni sobre impulsos, tampoco se observan oscilaciones en régimen estacionario. Los pares aplicados evolucionan dentro de los límites de saturación y la trayectoria que describe el extremo final del robot en su espacio de trabajo es suave, tal y como se muestra en el diagrama fase.



**Figura 8.6** Control PD: robot cartesiano de 3 gdl.



### Código Fuente 8.7 cap8\_robot3gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_robot3gdl.m

```

1 function xp = cap8_robot3gdl(t,x)
2     q1=x(1); q2=x(2); q3=x(3); q = [q1; q2; q3];%vector de posición articular
3     qp1=x(4); qp2=x(5); qp3=x(6); qp = [qp1; qp2; qp3];%vector de velocidad articular
4     lz1=1.26; lz2=0.084; lz3=0.056; ly1=0.089; ly2=0.003; ly3=0.0012; lx1=0.03; lx2=0.05;
5     lx3=0.009; m1=26.902; l1=0.45; b1=2.288; m2=30; l2=0.45; lc2=0.038; b2=0.2; m3=3.880;
6     l3=0.45; lc3=0.048; b3=0.175; g=9.81;
7     m11=ly2*sin(q2)*sin(q2)+ ly3*sin(q2+q3)*sin(q2+q3)+ lz1+lz2*cos(q2)*cos(q2)+
8     lz3*cos(q2+q3)*cos(q2+q3)+ m2*lc2*lc2*cos(q2)*cos(q2)+ m3*(l2*cos(q2)+
9     lc3*cos(q2+q3))*(l2*cos(q2)+lc3*cos(q2+q3));
10    m12=0; m13=0; m21=0;
11    m22=lx2+ lx3+m3*l2*l2+ m2*lc2*lc2+m3*lc3*lc3+ 2*m3*l2*lc3*cos(q3);
12    m23=lx3+m3*lc3*lc3+m3*l2*lc3*cos(q3); m31=0;
13    m32=lx3+m3*lc3*lc3+m3*l2*lc3*cos(q3); m33=lx3+m3*lc3*lc3;
14    M=[m11, m12, m13; m21, m22, m23 ;m31, m32, m33];%matriz de inercia
15    gamma112=(ly2-lx2-m2*lc2*lc2)*cos(q2)*sin(q2)+(ly3-lz3)*cos(q2+q3)*sin(q2+q3)-
16    m3*(l2*cos(q2)+lc3*cos(q2+q3))*(l2*sin(q2)+ lc3*sin(q2+q3));
17    gamma113=(ly3-lz3)*cos(q2+q3)*sin(q2+q3)- m3*lc3*sin(q2+q3)*(l2*cos(q2)+lc3*cos(q2+q3));
18    gamma121=(ly2-lz2-m2*lc2*lc2)*cos(q2)*sin(q2)+(ly3-lz3)*cos(q2+q3)*sin(q2+q3)-
19    m3*(l2*cos(q2)+lc3*cos(q2+q3))*(l2*sin(q2)+lc3*sin(q2+q3));
20    gamma131=(ly3-lz3)*cos(q2+q3)*sin(q2+q3) -m3*lc3*sin(q2+q3)*(l2*cos(q2)+ lc3*cos(q2+q3));
21    gamma211=(lx2-ly2+m2*lc2*lc2)*cos(q2)*sin(q2)+(lz3-ly3)*cos(q2+q3)*sin(q2+q3)+
22    m3*(l2*cos(q2)+lc3*cos(q2+q3))*(l2*sin(q2)+lc3*sin(q2+q3));
23    gamma223=-l2*m3*lc3*sin(q3); gamma232=-l2*m3*lc3*sin(q3);
24    gamma233=-l2*m3*lc3*sin(q3); gamma311=(lz3-ly3)*cos(q2+q3)*sin(q2+q3)+
25    m3*lc3*sin(q2+q3)*(l2*cos(q2)+lc3*cos(q2+q3));
26    gamma322=l2*m3*lc3*sin(q3); c11=gamma112*qp2+gamma113*qp3;
27    c12=gamma112*qp1; c13=gamma113*qp1; c21=gamma211*qp1; c22=gamma223*qp3;
28    c23=gamma232*qp2+gamma233*qp3; c31=gamma311*qp1; c32=gamma322*qp2; c33=0;
29    C=[c11, c12, c13; c21, c22, c23; c31, c32, c33];
30    gq11=0; gq21=(lc2*m1++m2*l2)*sin(q1)+ m2*lc3*sin(q1+q2);
31    gq31=m2*lc3*sin(q1+q2);
32    gq=g*[gq11; gq21; gq31];
33    fr=[b1*qp1; b2*qp2; b3*qp3];
34    [tau, tau] = cap8_PDrobot3gdl(q,qp);
35    qpp = inv(M)*(tau-C*qp-gq-fr);
36    xp = [qp1; qp2; qp3; qpp(1); qpp(2); qpp(3)];%vector de salida
37 end

```



**Código Fuente 8.8 cap8\_PDrobot3gdl.m**

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

**cap8\_PDrobot3gdl.m**

```

1 function [qtildeg tau] = cap8_PDrobot3gdl(x,xp)
2     q1=x(1); q2=x(2); q3=x(3);
3     q = [q1; q2; q3];%posiciones
4     qp1=xp(1); qp2=xp(2); qp3=xp(3);
5     qp = [qp1; qp2; qp3];%velocidades
6     %parámetros del robot: par gravitacional
7     m1=26.902; l1=0.45; m2=30; l2=0.45; lc2=0.038;
8     m3=3.880; l3=0.45; lc3=0.048; g=9.81;
9     gq11=0;
10    gq21=(lc2*m1+m2*l2)*sin(q1)+m2*lc3*sin(q1+q2);
11    gq31=m2*lc3*sin(q1+q2);
12    gq=g*[gq11; gq21; gq31];%par gravitacional
13    qd=[30; 45; 90];%referencias o posiciones deseadas
14    qtilde=qd*pi/180-q;%error de posición  $q$ 
15    qtildeg=(180/pi)*qtilde; %error de posición  $q$  en grados
16    qpgrados=180*qp/pi;%velocidad  $q$  en grados/segundo.
17    %ganancia proporcional
18    Kp=[1, 0, 0 ;
19        0, 2, 0;
20        0, 0, 0.15];
21    %ganancia derivativa
22    Kv=[0.3, 0, 0;
23        0, 0.6, 0;
24        0, 0, 0.015];
25    % control proporcional derivativo PD
26    tau=Kp*qtildeg-Kv*qpgrados+gq;
27 end

```



### Código Fuente 8.9 cap8\_robot3gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

---

#### cap8\_robot3gdlsimu.m

---

```

1  clc; clear all;
2  close all;
3  format short
4  %parámetros de simulación:
5  ti=0; h=0.0025; tf = 5; ts=ti:h:tf; %vector tiempo
6  opciones=odeset('RelTol',1e-3,'InitialStep', 2.5e-3,'MaxStep',2.5e-3);
7  ci=[0; 0; 0; 0;0;0];
8  [t,x]=ode45('cap8_robot3gdl',ts,ci,opciones);
9  q1=x(:,1); q2=x(:,2); q3=x(:,3); %posiciones articulares
10 qp1=x(:,4); qp2=x(:,5); qp3=x(:,6); %velocidades articulares
11 [n,m]=size(t);
12 tau1=zeros(n,1); tau2=zeros(n,1); tau3=zeros(n,1);
13 qtilde1=zeros(n,1); qtilde2=zeros(n,1); qtilde3=zeros(n,1);
14 xef=zeros(n,1); yef=zeros(n,1); zef=zeros(n,1);
15 beta1=0.15; beta2=0.15; beta3=0.15; l1=1; l2=0.45; l3=0.45;
16 for k=1:n
17     [qt tau] = cap8_PDrobot3gdl([q1(k);q2(k);q3(k)],[qp1(k);qp2(k); qp3(k)]);
18     tau1(k)=tau(1); tau2(k)=tau(2); tau3(k)=tau(3);
19     qtilde1(k)=qt(1); qtilde2(k)=qt(2); qtilde3(k)=qt(3);
20     [xef(k), yef(k),zef(k)]=cinematica_r3gdl(beta1, l1, q1(k), beta2,
        l2,q2(k),beta3,l3,q3(k));
21 end
22 subplot(2,2,1); plot(t,qtilde1,t,qtilde2, t,qtilde3)
23 subplot(2,2,2); plot( t, tau1, t,tau2, t,tau3)
24 subplot(2,2,3); plot3(xef,yef,zef)
25 subplot(2,2,4); plot(qtilde1, qp1, qtilde2,qp2, qtilde3,qp3)

```



**Control PD de un robot cartesiano de 3 gdl**

El algoritmo de control proporcional derivativo para el robot manipulador en configuración cartesiana está dado por la siguiente estructura:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & d_1 & k_{v1} & 0 & 0 & d_1 \\ 0 & k_{p2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{p3} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \tilde{d}_3 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_{v2} & 0 \\ 0 & 0 & k_{v3} \end{bmatrix} \begin{bmatrix} \dot{\tilde{d}}_1 \\ \dot{\tilde{d}}_2 \\ \dot{\tilde{d}}_3 \end{bmatrix} + g \begin{bmatrix} m_1 + m_2 + m_3 \\ 0 \\ 0 \end{bmatrix}$$

donde  $K_p, K_v \in \mathbb{R}^{3 \times 3}$  son la ganancia proporcional y derivativa, respectivamente; el error de posición  $\tilde{\mathbf{d}} \in \mathbb{R}^3$  se define como la diferencia entre la posición deseada  $\mathbf{d}_d \in \mathbb{R}^3$  y la posición actual del robot  $\mathbf{d} \in \mathbb{R}^3$ .

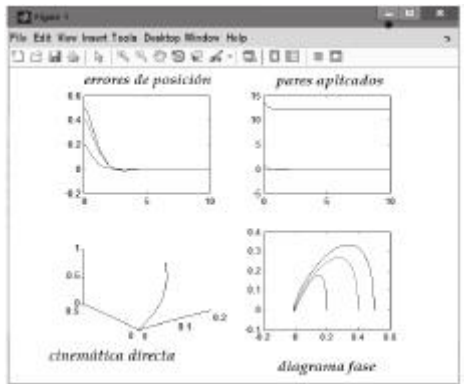
**♣ Ejemplo 8.4**

Controlar la posición del extremo final de un robot de 3 gdl en configuración cartesiana en la siguiente ubicación:  $[d_{d1}, d_{d2}, d_{d3}]^T = [0.5, 0.4, 0.2] \text{ m}$

**Solución**

El modelo dinámico del robot cartesiano se encuentra en el cuadro 8.10 que describe el programa cap8\_cartesiano3gdl.m; en la línea 23 de este programa se inserta el algoritmo de control PD definido por la función cap8\_PDcartesiano3gdl.m (ver cuadro 8.11). El programa principal para realizar la simulación del control PD en el robot cartesiano de 3 gdl se encuentra en el cuadro 8.12 que describe el contenido del programa cap8\_cartesiano3gdlsimu.m; los

resultados se encuentran en la figura 8.7.



**Figura 8.7** Control PD.



### Código Fuente 8.10 cap8\_cartesiano3gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.  
 %Editorial Alfaomega, Fernando Reyes Cortés.  
 %Capítulo 8 Control de posición.

---

#### cap8\_cartesiano3gdl.m

---

```

1 function xp = cap8_cartesiano3gdl(t,x)
2     %vector de posición articular
3     d = [x(1);
4         x(2);
5         x(3)];
6     %vector de velocidad articular
7     dp = [x(4);
8          x(5);
9          x(6)];
10    m1=0.7; m2=0.28; m3=0.28; g=9.81;
11    b1=0.02; b2=0.08; b3=.02;
12    B=[b1, 0, 0;
13      0, b2,0;
14      0, 0, b3];
15    %modelo dinámico del robot
16    M = [m1+m2+m3, 0, 0; % Matriz de inercia
17        0 m1+m2, 0;
18        0 0 m3];
19    par_grav = g*[m1+m2+m3;% vector de pares de gravitacionales
20    0;
21    0];
22    fr= B*dp;
23    [tau, tau] = cap8_PDcartesiano3gdl(d,dp);
24    dpp = inv(M)*(tau- par_grav-fr);%vector de aceleración articular
25    xp = [dp(1); dp(2); dp(3); dpp(1); dpp(2); dpp(3)];%vector de salida
26 end

```

---

**Código Fuente 8.11 cap8\_PDcartesiano3gdl.m**

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

**cap8\_PDcartesiano3gdl.m**

```

1 function [dtilde tau] = cap8_PDcartesiano3gdl (d,dp)
2     d1=d(1);
3     d2=d(2);
4     d3=d(3);
5     dp1=dp(1);
6     dp2=dp(2);
7     dp3=dp(3);
8     %parámetros del robot: par gravitacional
9     m1=0.7; m2=0.28; m3=0.28; g=9.81;
10    par_grav = g*[m1+m2+m3;
11                0;
12                0];
13    dd=[0.5;
14        0.4;
15        0.2];
16    dtilde=dd-[d1; d2;d3];
17    kp1=3; kp2=2.5;kp3=1.7;
18    kv1=3; kv2=2.5; kv3=1.4;
19    Kp=[kp1, 0,0;
20        0, kp2,0;
21        0,0,kp3];
22    Kv=[kv1, 0,0;
23        0, kv2,0;
24        0,0,kv3];
25    %Control proporcional derivativo
26    tau=Kp*dtilde-Kv*dp+par grav;
27 end

```



### Código Fuente 8.12 cap8\_cartesiano3gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

---

#### cap8\_cartesiano3gdlsimu.m

---

```

1  clc;
2  clear all;
3  close all;
4  format short
5  %parámetros de simulación:
6  ti=0; h=0.001; tf = 10; ts=ti:h:tf; %vector tiempo
7  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3, 'MaxStep',1e-3);
8  ci=[0; 0; 0; 0; 0; 0];
9  [t,x]=ode45('cap8_cartesiano3gdl',ts,ci,opciones);
10 d1=x(:,1); d2=x(:,2); d3=x(:,3);
11 dp1=x(:,4); dp2=x(:,5); dp3=x(:,6);
12 [n,m]=size(t);
13 tau1=zeros(n,1); tau2=zeros(n,1); tau3=zeros(n,1);
14 dtilde1=zeros(n,1); dtilde2=zeros(n,1); dtilde3=zeros(n,1);
15 xef=zeros(n,1); yef=zeros(n,1); zef=zeros(n,1);
16 for k=1:n
17     [dt tau] = cap8_PDcartesiano3gdl([d1(k); d2(k); d3(k)], [dp1(k); dp2(k);
        dp3(k)]);
18     tau1(k)=tau(1); tau2(k)=tau(2); tau3(k)=tau(3);
19     dtilde1(k)=dt(1); dtilde2(k)=dt(2); dtilde3(k)=dt(3);
20     [xef(k), yef(k), zef(k)]=cinematica_cartesiano(d3(k), d2(k), d1(k));
21 end
22 subplot(2,2,1); plot(t,dtilde1,t,dtilde2, t,dtilde3)
23 subplot(2,2,2); plot( t, tau1, t,tau2, t,tau3)
24 subplot(2,2,3); plot3(xef,yef,zef)
25 subplot(2,2,4); plot(dtilde1, dp1, dtilde2,dp2, dtilde3,dp3)

```

## 8.3 Control PID

El esquema de control proporcional integral derivativo (PID) tiene la siguiente estructura:

$$\tau = K_p \tilde{\mathbf{q}} - K_v \dot{\mathbf{q}} + K_i \int_0^t \tilde{\mathbf{q}}(\mu) d\mu + \mathbf{g}(\mathbf{q}) \quad (8.4)$$

donde  $K_p, K_v \in \mathbb{R}^{n \times n}$  son matrices definidas positivas y se les denomina ganancias proporcional y derivativa, respectivamente.  $K_i \in \mathbb{R}^{n \times n}$  es una matriz definida positiva denominada ganancia integral. Todas las ganancias  $K_p, K_v$  y  $K_i$  son matrices definidas positivas y generalmente son diagonales.

La acción de control PID introduce una nueva variable de estado  $\boldsymbol{\varepsilon}$ , la cual se encuentra relacionada con la derivada temporal del error de posición  $\boldsymbol{\varepsilon} = \dot{\tilde{\mathbf{q}}}$ . Por lo tanto,  $\boldsymbol{\varepsilon} = \int \dot{\tilde{\mathbf{q}}} dt + \boldsymbol{\varepsilon}(0)$ , con  $\boldsymbol{\varepsilon}(0) \in \mathbb{R}^n$  es un vector constante. La ecuación en lazo cerrado en términos de las variables de estado  $[\boldsymbol{\varepsilon}, \tilde{\mathbf{q}}]^T$  adquiere la siguiente forma:

$$\frac{d}{dt} \begin{bmatrix} \boldsymbol{\varepsilon} \\ \tilde{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{q}}} \\ -\dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{M}(\mathbf{q})^{-1} [K_p \tilde{\mathbf{q}} - K_v \dot{\tilde{\mathbf{q}}} + K_i \boldsymbol{\varepsilon} - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\tilde{\mathbf{q}}} - B \dot{\tilde{\mathbf{q}}}] & -\dot{\tilde{\mathbf{q}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon} \\ \tilde{\mathbf{q}} \end{bmatrix} \quad (8.5)$$

El algoritmo de control PID no representa una nueva estructura de control; es una versión modificada del control PD más la acción de control integral. El algoritmo de control PD tiene la desventaja que en régimen estacionario presenta un error permanente de posición denominado offset; para disminuir este error se incorpora en la estructura matemática del PD un término más que se denomina acción de control integral. El proceso de integración es la suma del área bajo la curva del error de posición, por lo tanto acumula energía la cual canalizada adecuadamente puede disminuir la magnitud del error en régimen estacionario.



## Control PID de un robot de 2 gdl

La ecuación del esquema de control proporcional derivativo para un brazo robot de 2 gdl está dado por:

$$\tau_1 = \begin{bmatrix} 0 \\ \tilde{q}_1 \end{bmatrix} + \begin{bmatrix} k_{p1} & k_{p2} \\ k_{i1} & k_{i2} \end{bmatrix} \begin{bmatrix} \tilde{q}_1 \\ \int \tilde{q}_1 dt \end{bmatrix}$$

$$g(l_1 m_1 + m_2 l_1) \sin(q_1) + g m_2 l_2 \sin(q_1 + q_2)$$

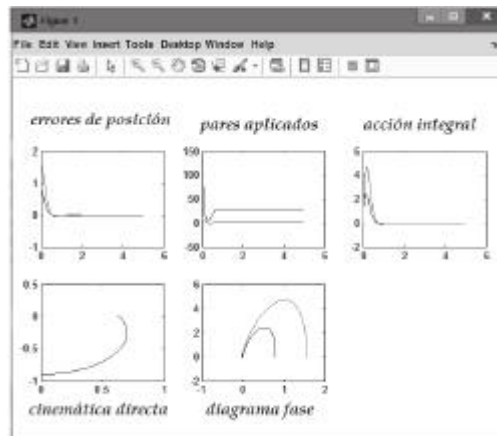
$$g m_2 l_2 \sin(q_1 + q_2)$$

### ♣ ♣ Ejemplo 8.5

Controlar la posición del extremo final de un robot de 2 gdl en configuración antropórfica en la siguiente ubicación:  $[q_{d1}, q_{d2}]^T = [45, 90] \text{ m}$

### Solución

El programa cap8\_PIDrobot2gdl.m tiene implementado el modelo dinámico del robot de 2 gdl (cuadro 8.13). La figura 8.8 muestra los resultados del algoritmo de control PID sobre el brazo robot de 2 gdl.



**Figura 8.8** Control PID de un robot de 2 gdl.



El esquema de control PID se encuentra en el programa cap8 PIDrobot2gdl.m que se muestra en el cuadro 8.13. La regla de sintonía de la ganancia derivativa e integral se ha fijado respectivamente en 20% y 10% de la ganancia proporcional. La integral del error introduce una nueva variable de estado (varepsilon) cuya integración numérica se realiza en el modelo dinámico del robot.



### Código Fuente 8.13 cap8\_PIDrobot2gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_PIDrobot2gdl.m

```

1 function [qtilde tau] = cap8 PIDrobot2gdl(x, xp, varepsilon)
2     q = [x(1); x(2)]; %posiciones
3     qp = [xp(1); xp(2)]; %velocidades
4     kp1=2.666; kp2=0.133;
5     Kp=[kp1, 0; 0, kp2]; %ganancia proporcional
6     ki1=0.1*kp1;
7     ki2=0.1*kp2;
8     Ki=[ki1, 0; 0, ki2]; %ganancia integral
9     kv1=0.2*kp1;
10    kv2=0.2*kp2;
11    Kv=[kv1, 0; 0, kv2]; %ganancia derivativa
12    %referencias
13    qd1=45; qd2=90; qd=[qd1; qd2]; %vector de posiciones deseadas
14    %vector de pares de gravitacionales
15    par_grav = [38.46*sin(q(1))+1.82*sin(q(1)+q(2));
16    1.82*sin(q(1)+q(2))];
17    qtilde=[pi*qd(1)/180-q(1); pi*qd(2)/180-q(2)]; %error de posición en grados
18    qtgrados=(180/pi)*[qtilde(1) ;qtilde(2)];
19    %velocidad en grados/segundo. qpgrados=180*qp/pi;
20    tau=Kp*qtgrados+Ki*varepsilon-Kv*qpgrados+par_grav;
21 end

```



### Código Fuente 8.14 cap8\_robot2gdIPID.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_robot2gdIPID.m

```

1 function xp = cap8_robot2gdIPID(t,x)
2     %vector de posición articular
3     q1=x(1); q2=x(2); q = [q1; q2];
4     %vector de velocidad articular
5     qp1=x(3); qp2=x(4); qp = [qp1; qp2];
6     varepsilon = [x(5); x(6)]; %integral de qtilde
7     m1=23.902; l1=0.45; lc1=0.091; l1=1.266; b1=2.288;
8     m2=3.880; l2=0.45; lc2=0.048; l2=0.093; b2=0.175; g=9.81;
9     theta1=m1*lc1*lc1+m2*l1*l1+m2*lc2*lc2+l1+l2;
10    theta2=l1*m2*lc2;
11    theta3=m2*lc2*lc2+l2;
12    theta4=g*(lc1*m1+m2*l1);
13    theta5=g*m2*lc2; theta6=b1; theta7=b2;
14    M=[theta1+2*theta2*cos(q2), theta3+theta2*cos(q2);
15    theta3+theta2*cos(q2), theta3];
16    C=[ -2*theta2*sin(q2)*qp2, -theta2*sin(q2)*qp2;
17    theta2*sin(q2)*qp1, 0];
18    gq11=theta4*sin(q1)+theta5*sin(q1+q2);
19    gq21=theta5*sin(q1+q2);
20    gq=[gq11; gq21];
21    fr=[theta6*qp1; theta7*qp2];
22    %ley de control
23    [qtilde tau] = cap8_PIDrobot2gdI(q,qp,varepsilon);
24    varepsilonp=qtilde;
25    qpp = M^(-1)*(tau-C*qp-gq-fr);
26    xp = [qp1; qp2; qpp(1); qpp(2);varepsilonp(1); varepsilonp(2)];
27 end

```

El programa `cap8_robot2gdIPIDsimu.m` que se presenta en el cuadro 8.15 permite realizar la simulación del control PID con el robot de 2 gdl



### Código Fuente 8.15 `cap8_robot2gdIPIDsimu.m`

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### `cap8_robot2gdIPIDsimu.m`

```

1  clc;
2  clear all;
3  close all;
4  format short g
5  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
6  ci=[0; 0; 0; 0;0;0]; %condiciones iniciales
7  opciones=odeset('RelTol',1e-3,'InitialStep', 1e-3, 'MaxStep',1e-3);
8  [t,x]=ode45('cap8_robot2gdIPID',t,ci,opciones);
9  q1=x(:,1); q2=x(:,2);
10 qp1=x(:,3); qp2=x(:,4);
11 int1=x(:,3); int2=x(:,4);
12 [n,m]=size(t);
13 tau1=zeros(n,1); tau2=zeros(n,1);
14 qtilde1=zeros(n,1); qtilde2=zeros(n,1);
15 xef=zeros(n,1); yef=zeros(n,1);
16 beta1=0.15; beta2=0.15; l1=0.45; l2=0.45;
17 for k=1:n
18     [qt tau] = cap8_PIDrobot2gdI([q1(k); q2(k)], [qp1(k); qp2(k)], [int1(k); int2(k)]);
19     tau1(k)=tau(1); tau2(k)=tau(2);
20     qtilde1(k)=qt(1); qtilde2(k)=qt(2);
21     [xef(k), yef(k), ~] = cinematica_robot2gdI(beta1, l1, q1(k), beta2, l2, q2(k));
22     p=Rz(-pi/2)*[xef(k), yef(k), beta1+beta2]';
23     xef(k)=p(1);
24     yef(k)=p(2);
25 end
26 subplot(3,3,1); plot(t,qtilde1,t,qtilde2)
27 subplot(3,3,2); plot( t, tau1, t,tau2)
28 subplot(3,3,3); plot( t,int1,t,int2)
29 subplot(3,3,4); plot(xef,yef)
30 subplot(3,3,5); plot(qtilde1, qp1, qtilde2,qp2)
31 subplot(3,3,6); plot(t, qp1, t,qp2)

```

## 8.4 Control punto a punto

Una aplicación inmediata de control de posición es el que se denomina control punto a punto, consiste en mover el extremo final del robot para cada una de las posiciones deseadas  $\mathbf{q}_{di}$ , con  $i = 1, 2, \dots, n$ . De esta forma es posible parametrizar líneas, círculos, elipses, figuras y superficies en función de un conjunto de puntos  $\mathbf{q}_{di}$ . Los algoritmos de control de posición de alto desempeño pueden ser empleados en control punto a punto como son los casos de los esquemas tangente hiperbólico y arcotangente.



### Control tangente hiperbólico

La estructura matemática del control tangente hiperbólico es la siguiente:

$$\boldsymbol{\tau} = \mathbf{K}_p \tanh(\tilde{\mathbf{q}}) - \mathbf{K}_v \tanh(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

#### ♣ ♣ Ejemplo 8.6

Posicionar el extremo final de un robot de 2 gdl en:  $[\mathbf{q}_{d1}, \mathbf{q}_{d2}]^T = [45, 90] \text{ m}$ .

#### Solución

El programa `cap8_robot2gdlTANH.m` que se presenta en el cuadro 8.16 tiene el modelo dinámico del robot de 2 gdl; en la línea 22 se conecta el archivo `cap8_TANHrobot2gdl.m` (ver cuadro 8.17) para realizar el control tangente hiperbólico. El programa `cap8_TANHrobot2gdlsimu.m` del cuadro 8.18 permite realizar la simulación del control tangente hiperbólico cuyos resultados se muestran en la figura 8.8.

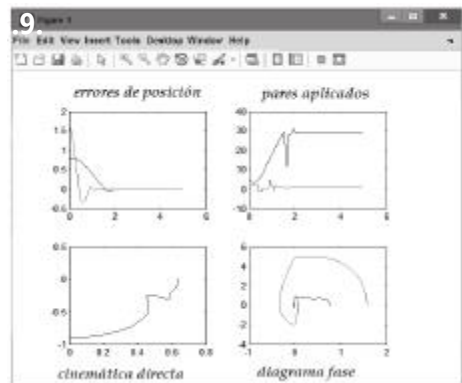


Figura 8.9 Control  $\tanh()$ .



### Código Fuente 8.16 cap8\_robot2gdITANH.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_robot2gdITANH.m

```

1 function xp = cap8_robot2gdITANH(t,x)
2     %vector de posición articular
3     q1=x(1); q2=x(2); q = [q1; q2];
4     %vector de velocidad articular
5     qp1=x(3); qp2=x(4); qp = [qp1; qp2];
6     m1=23.902; l1=0.45; lc1=0.091; l1=1.266; b1=2.288;
7     m2=3.880; l2=0.45; lc2=0.048; l2=0.093; b2=0.175; g=9.81;
8     theta1=m1*lc1*lc1+m2*l1*l1+m2*lc2*lc2+l1+l2;
9     theta2=l1*m2*lc2;
10    theta3=m2*lc2*lc2+l2;
11    theta4=g*(lc1*m1+m2*l1); theta5=g*m2*lc2;
12    theta6=b1; theta7=b2;
13    M=[theta1+2*theta2*cos(q2), theta3+theta2*cos(q2);
14    theta3+theta2*cos(q2), theta3];
15    C=[ -2*theta2*sin(q2)*qp2, -theta2*sin(q2)*qp2;
16    theta2*sin(q2)*qp1, 0];
17    gq11=theta4*sin(q1)+theta5*sin(q1+q2);
18    gq21=theta5*sin(q1+q2);
19    gq=[gq11; gq21];
20    fr=[theta6*qp1;
21    theta7*qp2];
22    [~, tau]=cap8_ITANHrobot2gdl(q,qp);
23    qpp = M^(-1)*(tau-C*qp-gq-fr);
24    %vector de salida
25    xp = [qp1; qp2; qpp(1); qpp(2)];
26 end

```



### Código Fuente 8.17 cap8\_TANHrobot2gdl.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

---

cap8\_TANHrobot2gdl.m

---

```

1 function [qtilde tau] = cap8_TANHrobot2gdl(x, xp)
2     q1=x(1); q2=x(2); q = [q1; q2]; %posiciones
3     qp1=xp(1); qp2=xp(2); qp = [qp1; qp2]; %velocidades
4     %parámetros del robot: par gravitacional
5     m1=23.902; lc1=0.091; l1=0.45;
6     m2=3.880; lc2=0.048; g=9.81;
7     theta4=g*(lc1*m1+m2*l1);
8     theta5=g*m2*lc2;
9     %par gravitacional
10    par_grav = [theta4*sin(q1)+theta5*sin(q1+q2);
11              g*m2*lc2];
12    %ganancia proporcional
13    kp1=10; kp2=2;
14    Kp=[kp1, 0; 0, kp2];
15    %ganancia derivativa
16    kv1=8; kv2=1; Kv=[kv1, 0; 0, kv2];
17    %referencias
18    qd1=45; qd2=90; qd=[qd1; qd2];
19    qtilde=pi/180*qd-q;
20    %error de posición en grados
21    qtgrados=(180/pi)*qtilde;
22    %velocidad en grados/segundo
23    qpgrados=180*qp/pi;
24    tau=Kp*tanh(qtgrados)-Kv*tanh(qpgrados)+par_grav;
25 end

```

---



### Código Fuente 8.18 cap8\_TANHrobot2gdlsimu.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

---

#### cap8\_TANHrobot2gdlsimu.m

---

```

1  clc;
2  clear all;
3  close all;
4  format short g
5  ti=0; h=0.001; tf = 5; t=ti:h:tf;%vector tiempo
6  ci=[0; 0; 0; 0];%condciones iniciales
7  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
8  [t,x]=ode45('cap8_robot2gdITANH',t,ci,opciones);
9  q1=x(:,1); q2=x(:,2);
10 qp1=x(:,3); qp2=x(:,4);
11 [n,m]=size(t);
12 tau1=zeros(n,1); tau2=zeros(n,1); qtilde1=zeros(n,1); qtilde2=zeros(n,1);
   xef=zeros(n,1); yef=zeros(n,1);
13 beta1=0.15; beta2=0.15; l1=0.45; l2=0.45;
14 for k=1:n
15     [qt tau] = cap8_JANHrobot2gdl([q1(k);q2(k)],[qp1(k);qp2(k)]);
16     tau1(k)=tau(1); tau2(k)=tau(2);
17     qtilde1(k)=qt(1); qtilde2(k)=qt(2);
18     [xef(k), yef(k), ~]=cinematica_r2gdl(beta1,l1,q1(k),beta2,l2,q2(k));
19     p=Rz(-pi/2)*[xef(k), yef(k), beta1+beta2]';
20     xef(k)=p(1); yef(k)=p(2);
21 end
22 subplot(2,2,1); plot(t,qtilde1,t,qtilde2)
23 subplot(2,2,2); plot( t, tau1, t,tau2)
24 subplot(2,2,3); plot(xef,yef)
25 subplot(2,2,4); plot(qtilde1, qp1, qtilde2,qp2)

```

---



## Control arcotangente

El control arcotangente está dado por la siguiente expresión:

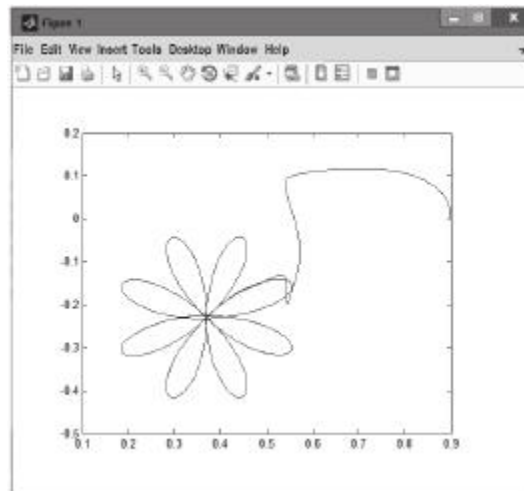
$$\tau = K_p \operatorname{atan}(\dot{\mathbf{q}}) - K_v \operatorname{atan}(\mathbf{q}) + \mathbf{g}(\mathbf{q})$$

### ♣ ♣ Ejemplo 8.7

Generar una flor de 8 pétalos con un robot de 2 gdl usando el control arcotangente.

### Solución

Dentro de las aplicaciones de control punto a punto se encuentra el trazo de figuras. El programa `cap8_robot2gdlatansimu.m` (cuadro 8.19) realiza la simulación para que un robot antropomórfico de 2 gdl trace una figura tipo flor con 8 pétalos usando el control arcotangente como se muestra en la figura 8.10.



**Figura 8.10** Control  $\operatorname{atan}()$ .

El control arcotangente es de alto desempeño y la sintonización de las ganancias proporcional y derivativa deben satisfacer ser menor al par máximo que pueda



entregar el servoamplificador. Con esta sintonía de las ganancias es posible controlar todos los puntos de la figura a trazar.

El programa `cap8_robot2gdflor8p.m` que se encuentra en el cuadro 8.20 tiene el código fuente del robot de 2 gdl; particularmente en la línea 11 se inserta el algoritmo de control arcotangente el cual se encuentra en el programa `cap8_atanflor8p.m` del cuadro 8.21. Observe que el pase de parámetros del control arcotangente son:  $t$ ,  $q$ ,  $q^*$  el tiempo  $t$  es necesario para generar la figura en forma de flor con 8 pétalos.



### Código Fuente 8.19 `cap8_robot2gdlatansimu.m`

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

---

`cap8_robot2gdlatansimu.m`

---

```

1  clc; clear all; close all;
2  format short g
3  ti=0; h=0.001; tf = 10;%vector tiempo
4  t=ti:h:tf;%tiempo de simulación
5  ci=[0; 0; 0; 0];
6  opciones=odeset('RelTol',1e-3,'InitialStep',1e-3,'MaxStep',1e-3);
7  [t,x]=ode45('cap8_robot2gdflor8p',t,ci,opciones);
8  q1=x(:,1);
9  q2=x(:,2);
10 [n,m]=size(t);
11 xef=zeros(n,1);
12 yef=zeros(n,1);
13 beta1=0.15; beta2=0.15; l1=0.45; l2=0.45;
14 [xef, yef, ~]=cinematica_2gdl(beta1,l1,q1,beta2,l2,q2);
15 plot(xef,yef)

```

---



### Código Fuente 8.20 cap8\_robot2gdlflor8p.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_robot2gdlflor8p.m

```

1 function xp = cap8_robot2gdlflor8p(t,x)
2     q = [x(1); x(2)]; %vector de posición articular
3     qp = [x(3); x(4)]; %vector de velocidad articular
4     M = [2.351+0.168*cos(q(2)) 0.102+0.084*cos(q(2)); %matriz de inercia
5          0.102+0.084*cos(q(2)) 0.102 ];
6     C = [-0.168*sin(q(2))*qp(2) -0.084*sin(q(2))*qp(2); %matriz de fuerzas centrífugas y de Coriolis
7          0.084*sin(q(2))*qp(1) 0.0 ];
8     par_grav = [38.46*sin(q(1))+1.82*sin(q(1)+q(2)); %vector de pares de gravitacionales
9                1.82*sin(q(1)+q(2)) ];
10    fr = [2.288*qp(1); 0.175*qp(2)];
11    [tau, tau] = cap8_atanflor8p(t,q,qp); %ley de control arcotangente
12    qpp = inv(M)*(tau-C*qp- par_grav-fr); %vector de aceleración articular
13    xp = [qp(1); qp(2);qpp(1); qpp(2)]; %vector de salida
14 end

```



### Código Fuente 8.21 cap8\_atanflor8p.m

%MATLAB Aplicado a Robótica y Mecatrónica.

%Editorial Alfaomega, Fernando Reyes Cortés.

%Capítulo 8 Control de posición.

#### cap8\_atanflor8p.m

```

1 function [qtilde tau] = cap8_atanflor8p(t,x,qp)
2     q = [x(1); x(2)]; %posiciones
3     qp = [xp(1); xp(2)]; %velocidades
4     kp1=100; kp2=15; kv1=kp1;kv2=kp2;
5     Kp=[kp1, 0; 0, kp2]; %ganancia proporcional
6     Kv=[kv1, 0; 0, kv2]; %ganancia derivativa
7     lambda1=180/pi; lambda2=180/pi; Lambda=[lambda1 0; 0 lambda2];
8     gamma1=1; gamma2=1; Gamma=[gamma1 0; 0 gamma2];
9     par_grav = [38.46*sin(q(1))+1.82*sin(q(1)+q(2)); %vector de pares de gravitacionales
10              1.82*sin(q(1)+q(2)) ];
11    xc=0.3; yc=0.3; r=0.2; l1=0.45; l2=0.45; alfa=0.07; beta=4; x=xc+alfa+r*(cos(t))*(sin(beta*t));
12    y=-yc+alfa+r*(sin(t))*(sin(beta*t));
13    [qd1 qd2]=cinvt2gdl(l1,l2,x,y);
14    qd=[qd1; qd2];
15    qtilde=[qd(1)-q(1); qd(2)-q(2)]; %error de posición
16    %ley de control PD
17    tau=Kp*atan(Lambda*qtilde)-Kv*atan(Gamma*qp)+par_grav;
18 end

```

## 8.5 Resumen

**C**ontrol de posición robots manipuladores permite realizar una amplia gama de aplicaciones industriales, comerciales, científicas, etc. La aplicación más popular del control de robot manipuladores es la que se denomina control punto a punto. Prácticamente todas las aplicaciones industriales pueden ser realizadas a través de esta técnica de control.

En el presente capítulo se presentó las bases para entender el funcionamiento cualitativo de los algoritmos de control, partiendo desde los tradicionales como el proporcional derivativo (PD) y control proporcional integral derivativo (PID), hasta nuevos esquemas como las funciones tangente hiperbólica y arcotangente.

El modelo dinámico de robots manipuladores permite evaluar cualitativamente el desempeño de nuevos esquemas de control como:

$$[\dot{q} \text{ tilde } \tau] = \text{pd}(x, x_p)$$

if

$$[\dot{q} \text{ tilde } \tau] = \text{pid}(x, x_p, \text{varepsilon})$$

if

$$[\dot{q} \text{ tilde } \tau] = \text{atan}(x, x_p)$$

if

$$[\dot{q} \text{ tilde } \tau] = \text{tanh}(x, x_p)$$

if

El diseño de nuevas estrategias de control es posible gracias a la técnica de moldeo de energía la cual genera una familia muy extensa de esquemas de control para robots manipuladores.

En el Web del libro se encuentran los programas fuentes de las librerías de control de posición.













## Parte IV Referencias selectas

Hay una extensa literatura para el área de control de robots manipuladores, particularmente se recomienda los siguientes textos:



### 8 Control de posición

-  M. Vidyasagar. "Nonlinear systems analysis". Prentice-Hall, Englewood Cliffs, NJ (1993).
-  R.M. Murray, Z. Li and S.S. Sastry. "A mathematical introduction to robotic manipulation". CRC Press (1994).
-  A. Isidori. "Nonlinear control systems". Springer-Verlag, London, UK (1995).
-  Karl J. Åström and Björn Wittenmark. "Computer controlled systems: theory and design". Third Edition. Prentice Hall Inc. 1997.
-  Shankar Sastry. "Nonlinear systems: analysis stability and control". Springer 1999.
-  Hassan K. Khalil. "Nonlinear systems". Third Edition. Prentice Hall Inc. 2002.

-  R. Kelly y V. Santibáñez. "Control de movimiento de robots manipuladores". Printice-Hall, Pearson. 2003.
  
-  R. Kelly, V. Santibáñez and A. Loría. "Control of robot manipulators in joint space". Springer-Verlag London 2005.
  
-  Mark W. Spong and Seth Hutchinson, M. Vidyasagar. "Robot modeling and control". John Wiley and Sons, Inc. 2006.
  
-  Fernando Reyes "Robótica: control de robots manipuladores". Alfaomega, 2011.

## Parte IV Problemas propuestos



**E**n esta sección se presenta una serie de ejercicios con la finalidad de mejorar las habilidades en simulación de control de robots manipuladores.



### Capítulo 8 Control de posición

8.1 Implementar el control PID para los siguientes robots:

- (a) Péndulo  $q_{d1} = 90$  grados.
- (b) Robot antropomórfico de 3 gdl  $\mathbf{q}_d = [30, 45, 90]$  grados.
- (c) Control cartesiano de 3 gdl  $\mathbf{d}_d = [0.3, 0.4, 0.5]$  m.

8.2 Considere la siguiente estructura de control:

$$\boldsymbol{\tau} = \mathbf{K}_p \text{sen}(\tilde{\mathbf{q}}) - \mathbf{K}_v \text{sen}(\mathbf{q}) + \mathbf{g}(\mathbf{q})$$

implementar esta estructura de control en los siguientes robots:

- (a) Péndulo  $q_{d1} = 90$  grados.
- (b) Robot antropomórfico de 2 gdl  $\mathbf{q}_d = [45, 90]^T$  grados.
- (c) Robot antropomórfico de 3 gdl  $\mathbf{q}_d = [45, 90, 135]^T$  grados.
- (d) Control cartesiano de 3 gdl  $\mathbf{d}_d = [0.5, 0.8, 0.9]^T$  m.

8.3 Considere la siguiente familia de algoritmos de control tipo PD:

$$\boldsymbol{\tau} = \mathbf{K}_p \ddot{\mathbf{q}}^{2m-1} - \mathbf{K}_v \dot{\mathbf{q}}^{2m-1} + \mathbf{g}(\mathbf{q})$$

donde  $m \in \mathbb{N}$ .

Implemente esta estructura de control en los casos para  $m = 1, 2, 3, 4, 5$  en los siguientes robots:

- (b) Robot antropomórfico de 2 gdl  $\mathbf{q}_d = [90, 135]^T$  grados.
- (c) Control cartesiano de 3 gdl  $\mathbf{d}_d = [0.5, 0.8, 0.9]^T$  m.

¿Qué reglas de sintonía puede proponer para obtener el valor de las ganancias de tal forma que no sature a los servoamplificadores cuya capacidad son de  $\tau_1^{\max} = 150$  Nm,  $\tau_2^{\max} = 15$  Nm para el robot de 2 gdl y  $\tau_1^{\max} = 50$  Nm,  $\tau_2^{\max} = 150$ ,  $\tau_3^{\max} = 15$  para el robot de 3 gdl.

8.4 Sea la siguiente familia de controladores

$$\boldsymbol{\tau} = \mathbf{K}_p \frac{\cosh^{m-1}(\tilde{\mathbf{q}}) \sinh(\tilde{\mathbf{q}})}{1 + \cosh(\tilde{\mathbf{q}})} - \mathbf{K}_v \operatorname{atan}(\tilde{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

considere los casos de estudio para  $m = 1, 2, 3, 4, 5$  en un robot cartesiano de 3 gdl para controlar la posición del extremo final del robot en las siguientes coordenadas:  $[0.4, 0.3, 0.8]$  m.

- (a) ¿Cómo varía la respuesta del robot (errores de posición) conforme se incrementa el exponente  $m$ .
- (b) ¿Describa el desempeño de este esquema de control en función de las ganancias y el exponente  $m$ .

# Índice analítico

## A

algoritmo  
    recursivo, 332  
    de control, 393

amortiguamiento, 129

antropomórfico, 408

archivos experimentales, 76

arreglos, 26, 41

## B

backward difference, 95

borrando, 35

brazo robot de 2 gdl, 403

break, 73

## C

case, 72

centrífuga, 301

clear, 17

cinemática, 140, 185  
    diferencial, 187  
    inversa, 186

columna de una matriz, 27

columnas, 26, 35

concatenación, 35

constante de tiempo, 297

constantes útiles, 18

continue, 74

control, 242

convección DH, 192  
    arcotangente, 390  
    de posición, 393  
    de robots, 390, 393  
    PD, 396  
    PD,  
        péndulo, 397  
        robot cartesiano, 413  
        robot de 2 gdl, 403  
        robot de 3 gdl, 408

- PID, 417
  - proporcional derivativo, 390, 395
  - proporcional integral derivativo, 390
  - punto a punto, 395
  - tangente hiperbólico, 390
- CRTL Q, 15
- D**
- Denavit-Hartenberg, 192
  - demos, 13
  - derivada, 93
  - desempeño, 395
  - det, 85
  - determinante simbólico, 86
  - determinantes, 85
  - diferenciación numérica, 92
  - diff, 94, 97
  - dinámica, 284
- E**
- ecuación diferencial ordinaria, 117, 288
  - else, 60
  - elseif, 60, 64
  - end, 51, 52
  - error de posición, 395, 417
  - esquema escalar, 381
  - esquemas de regresión, 332, 359
  - estimador de velocidad, 296
  - exit, 14
  - expansión de escalares, 37
  - eye, 32
  - ezmesh, 47
- F**
- fclose, 76
  - feval, 50
  - filtrado, 296, 299, 350
  - filtro, 299
  - fopen, 76
  - for, 58, 61, 64, 67, 73
  - format, 18
  - formato numérico, 18
  - forward difference, 95
  - fplot, 46
  - fprintf, 76



frecuencia de corte, 357

frecuencia natural, 129

fricción de Coulomb, 306

fricción viscosa, 306

función, 51

función

    círculo, 56

    transferencia, 289

    discontinua, 351

    genvarname, 16

    ode, 120

    ode23, 122

    anidada, 53

    archivo, 51

    cuadratura, 113

    en línea, 50

    quad, 108



ganancia

    derivativa, 417

    proporcional, 417

    integral, 417

globales, 53

gráficas, 43



help, 13, 43

    elfun, 49

    elmat, 49

    specfun, 49

helpwin, 43



identificación, 331

identificadores, 15

if, 58–60, 64

inercia del rotor, 306

inercia, 311

inercia, robot 3 gdl

instrucciones condicionales, 59

integración numérica, 100

integral, 93

inversa de matriz, 87



librerías de

    cinemática, 274

    Denavit-Hartenberg, 180

Traslación y rotación, 182  
 matrices homogéneas, 174  
 mínimos cuadrados, 334

linesolve, 89

load, 32, 76

Lobatto, 113

locales, 53

## M

matrices de rotación, 148

máximo, 69

método de Euler, 95, 114

método de primer orden, 118

método de Runge-Kutta, 118, 119

mínimo, 69

mínimos cuadrados, 332, 334

magic, 32

Mathworks, 5, 80

matrices, 26, 38

matriz de covarianza, 333

mesh, 47

modelo

- energía, 353, 366

- potencia, 355, 359, 369, 378
- regresión, 346
- regresión dinámico, 361
- dinámico, 287, 331, 346, 359
- dinámico filtrado, 350

moldeo de energía, 390, 393

multivariable, 287, 336, 381

## O

objetivo de control, 394

ode, 120, 288

ode113, 123

ode15s, 123

ode23, 122

ode23s, 123

ode23t, 123

ode23tb, 124

ode45, 120

off-line, 83

ones, 32

operaciones básicas, 38, 40

operaciones con arreglos, 41

operadores

- `[]`, 28
- `'`, 24
- `()`, 23

,, 24

:, 21

;; 23

%, 25

**MATLAB**, 5, 7

~, 25

aritméticos, 12

arreglos, 41

lógicos, 58



péndulo, 305, 346, 353, 397

parámetros, 331

PD, 390, 395

PID, 100, 390, 417

pivote, 27

plot3, 43

potencia, 381

potencia filtrada, 357

problema de control, 393

producto interno, 144

prompt, 11



quad, 113

quadgk, 114

quadl, 113

quadv, 114

quit, 14



rand, 32

randn, 32

regla

de Cramer, 91

de Simpson, 108

trapezoidal, 102

rotación, 164

regresor, 332

regulación, 393

renglón de una matriz, 27

renglones, 26, 35

return, 73

robótica, 393

robot






antropomórfico, 408

cartesiano, 373

cartesiano de 3 gdl, 413

de 3gdl, 321

de 2 gdl, 310, 360

- manipulador de 3 gdl, 315, 408
- manipuladores, 287, 393
- ruido, 299
- Runge Kutta, 327
- Runge-Kutta, 118
- 
- save, 76
- script, 54
- serie de Taylor, 118
- set point, 393, 395
- signo, 65
- Simpson, 113
- simulación, 287
- simulación de sistemas dinámicos, 124
- Simulink, 7
- sintaxis, 15
- sistema lineal escalar, 295
- sistema masa resorte amortiguador, 291
- sistemas dinámicos, 117
- sistemas dinámicos autónomos, 288
- sistemas dinámicos no autónomos, 288
- sistemas lineales, 289
- sistemas mecatrónicos, 331
- subplot, 48
- suma de matrices, 68
- switch, 58, 72
- 
- transformación
  - homogénea, 173
  - traslación, 171
- 
- variables, 15
- vector de mediciones, 333
- vector de parámetros, 333
- ventana de comandos, 11
- 
- while, 58, 71, 73
- whos, 16
- 
- zeros, 32