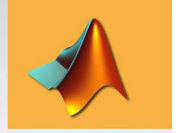


FERNANDO REYES CORTÉS

MATLAB

APLICADO A ROBÓTICA Y MECATRÓNICA



```
function xp =robot(t,x)
%vector de estados
q=x(1); %posición articular
qp=x(2); %velocidad articular
%parámetros del robot
m=5; %masa
lc=0.01; %centro de masa
g=9.81; %constante de aceleración
b=0.17; %coeficiente de fricción
fc=0.45; %coeficiente de fricción
Ir=0.16; %momento de inercia de
tau=1.5*sin(t); %par aplicado
%aceleración articular del robot
qpp=(tau-b*qp-fc*tau)/(m*g*lc*sin(t));
%vector de salida
xp=[ qp ; %xp(1)=x(2)
      qpp]; %xp(2)=qpp
end
```

Apoyo en la



 **Alfaomega**



Guía de usuario

ESTE documento pretende ser un breve manual de instalación y guía de ejecución de los programas en código abierto de la obra **MATLAB Aplicado a Robótica y Mecatrónica**. De manera inicial en esta página Web se reportan más de 140 programas en código fuente que de manera conjunta incluyen alrededor de 3000 líneas de código para simular, analizar, diseñar y desarrollar aplicaciones con sistemas mecatrónicos y robots manipuladores. Los programas desarrollados representan un conjunto de librerías técnicas denominados *toolbox* que facilitan tanto al profesor como al estudiante presentar, explicar y comprender conceptos tales como: orientación, rotación y traslación del extremo final del robot; cinemática directa e inversa, jacobianos, dinámica, identificación paramétrica y algoritmos de control.

La página Web de la presente obra continuamente estará actualizando la información con notas académicas, ejercicios e incrementando programas de simulación para realizar un mayor número de aplicaciones.

Asimismo se reporta la fe de erratas que se refiere a errores tipográficos de impresión señalando la página y lugar del libro donde ocurrió. Dentro de los principales motivos que dieron origen a las erratas se encuentran, claro está el factor humano y el cambio de formato de código **MATLAB** a lenguaje **LaTeX**. Sin embargo, la depuración de errores es una actividad que se realizará de manera constante y sistemática. Es decir, se está perfeccionando las librerías **Latex** que permiten editar, formar y compilar un libro; esto tiene la finalidad de mejorar la calidad de la obra en ediciones posteriores. No obstante, los programas con código fuente reportados en este sitio Web **no tienen ningún errores para su ejecución**, ya que todos los programas han sido probados con el ambiente de programación de **MATLAB**, versión 2011a.

Nota importante

Los programas de código fuente de esta obra han sido desarrollador para la versión **MATLAB 2011a**.

Como una primera fase y con la finalidad que el lector se familiarice con el conjunto de programas para robótica y mecatrónica (*toolbox*) se recomienda enfáticamente que la instalación de librerías sea como a continuación se sugiere en la figura 1:

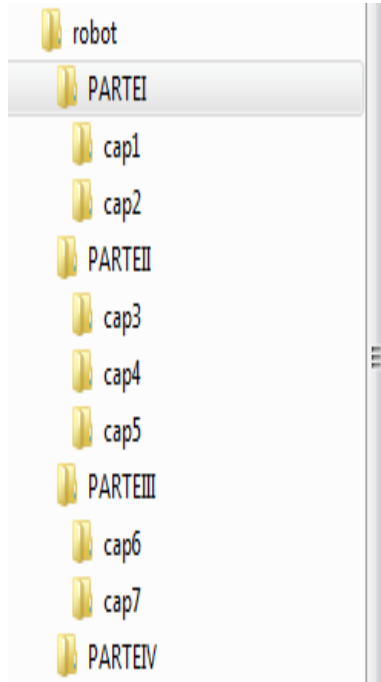


Figura 1 Instalación recomendada para las librerías de robótica y mecatrónica.

Por ejemplo, en el disco duro *c* crear el siguiente directorio `c:\robot\PARTEI` y dentro de la carpeta `PARTEI` crear las subcarpetas `cap1` y `cap2`, las cuales almacenarán las librerías correspondientes a dichos capítulos. De manera análoga para las carpetas `PARTEII` a la `PARTEIV` con sus respectivos capítulos. Lo anterior tiene la finalidad de conservar los programas de acuerdo a la organización y estructura del libro; de esta manera es mucho más fácil seguir todos los ejemplos ilustrados en la citada obra.

Una vez que el lector tenga dominio pleno sobre el *toolbox* de robótica y mecatrónica, entonces podrá instalar todas las librerías directamente sobre la raíz de la carpeta de instalación: `c:\robot`, listo para desarrollar aplicaciones.

NOTA IMPORTANTE

En el sitio Web del libro todos los programas en código fuente han sido depurados y se encuentran libres de las erratas.

La tabla 1 contiene el número de programas por capítulo. Esta estadística es inicial. No obstante, de manera sistemática se estará incrementando el número de programas en código abierto para **MATLAB**.

Tabla 1 Estadísticas de programas por capítulo

Capítulo	Nombre del capítulo	Número de programas
1	Conceptos básicos	16
2	Métodos numéricos	12
3	Preliminares matemáticos	14
4	Cinemática directa	6
5	Cinemática directa cartesiana	40
6	Dinámica	16
7	Identificación paramétrica	20
8	Control de posición	21
	Total de programas	145

A continuación se realiza una descripción de los programas de cada capítulo de la obra: **MATLAB Aplicado a Robótica y Mecatrónica**.

AlfaOmega

Parte I

Programación

 **Material Web**

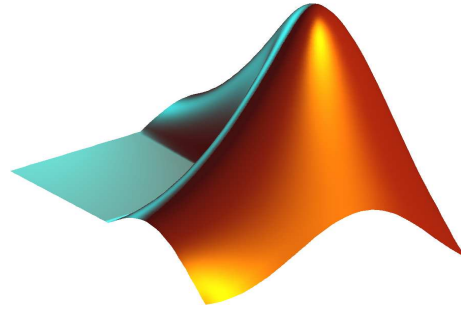
AlfaOmega

AlfaOmega

1

Capítulo

Conceptos básicos





















Material Web

AlfaOmega

1.1 Listado de programas del capítulo 1

EN esta sección se enlistan los nombres de los programas con código fuente correspondientes al capítulo 1:






-  `abre_archivos_experimentales.m` explica cómo cargar un archivo experimental con datos del robot.
-  `archivos_experimentales.m` explica cómo crear un archivo experimental con registro de variables del robot.
-  `archivos_experimentalesI.m` contiene otra opción para crear un archivo experimental con registro de variables del robot.
-  `cap1_exponencial.m` ejemplo de la función `fplot`.
-  `cap1_grafica.m` ejemplo de la función `plot`.
-  `cap1_graficas3D.m` funciones para graficar en 3D.
-  `cap1_matrices1.m` programa principal para realizar operaciones básicas con matrices.
-  `cap1_minimax.m` obtiene el mínimo y máximo de una función.
-  `cinematica_directa_robot2gd1.m` cinemática directa de un robot de 2 grados de libertad.
-  `circulo.m` función círculo.
-  `circulo_simu.m` programa principal para la función círculo.
-  `normaev.m` norma euclidiana de un vector.
-  `senoide.m` aplicación de la instrucción `for` grafica la función seno.
-  `signo.m` función signo empleando las instrucciones `if... else...`
-  `signo_simu.m` programa principal para ejecutar la función signo

-  `sumamatrices.m` programa para ejemplificar la instrucción `for...end`. Realiza en forma iterativa suma de matrices.
-  `robot.dat` archivo con datos experimentales o simulados de robots manipuladores.
-  `robot1.dat` archivo con datos experimentales o simulados de robots manipuladores.

1.2 Fe de erratas

EN el Capítulo 1 **Conceptos básicos** se reportan erratas de impresión que han sido depuradas en los respectivos programas fuentes del sitio Web de esta obra.

A continuación se describen las erratas del capítulo 1:

-  En la página 40 **Código fuente 1.1 Operaciones básicas con matrices:** en la línea 27 `r=root(p)` deberá ir `r=roots(p)`.
-  Página 47 línea 2 `fplot('9*x^5+3*sen(x^3)', [-10,10])` debe ser sustituida por `fplot('9*x^5+3*sin(x^3)', [-10,10])` se ha reemplazado (sin por sen).
-  Página 48 **Código fuente 1.4 Funciones 3D** en la línea 6
`6 subplot(2,2,2); ezmesh('x.*exp(-x.^2-y.^6)', 40);`
deberá ser
`6 subplot(2,2,2); ezmesh('x.*exp(-x.^2-y.^6)', 40); fh='x.*exp(-x.^2-y.^2)';`
-  Página 52 **Código fuente 1.5 norma euclidiana de un vector** en la línea 8 `y=norma(x,2)` debe ser: `y=norm(x,2)`
-  Página 70 **Código fuente 1.13 Mínimos y máximos** línea 11:
`11 y(i)=sin(t(i))-cos(3.1416*t(i))-2*tanh(y(i))+0.1*log(t(i)^3+1);`
debe ser:

11 $y(i)=\sin(t(i))-\cos(3.1416*t(i))-2*\tanh(t(i))+0.1*\log(t(i)^3+1)$;
se ha sustituido $\tanh(y(i))$ por $\tanh(t(i))$.



Al final de la página 76 `fprintf(fid, '%3.3f%3.3f%3.3f%3.3f', datos)`;
debe ser sustituido por `fprintf(fid, '%3.3f%3.3f%3.3f%3.3f', datos')`;
la variable `datos` debe ir transpuesta `datos'`. El mismo problema se presenta
en la estructura de código 1.16 de la página 77.



En la parte final de la página 78 se tiene:

```
datos_experimentales=load('c:\robot\experimentos\robot.dat');
```

lo correcto debe ser:

```
datos_experimentales=load('c:\robot\experimentos\robot.dat');
```

2

Capítulo

Métodos numéricos

$$I = \int_0^t f(x)dx \Rightarrow I_k = I_{k-1} + hf(x_{k-1})$$













$$\dot{f} = \lim_{\Delta t \rightarrow 0} \frac{f(t + \Delta t) - f(t)}{\Delta t} \Rightarrow \dot{f}_k = \frac{f_k - f_{k-1}}{h}$$

 **Material Web**



2.1 Listado de programas del capítulo 2


LOS nombres de los programas con código fuente correspondientes al capítulo 2 se enlistan a continuación:

-  `cap2_detsim.m` ejemplo para obtener el determinante de una matriz usando variables simbólicas.
-  `cap2_sle.m` solución de un sistema lineal de ecuaciones.
-  `cap2_diffnum.m` aplicación de la función `diff` (diferenciación numérica).
-  `cap2_trap.m` técnica de integración trapezoidal.
-  `cap2_trapezoidal.m` ejemplo de integración trapezoidal.
-  `cap2_simpson1.m` método Simpson para integración numérica
-  `cap2_simpson.m` ejemplo de integración numérica por el método de Simpson.
-  `cap2_euler.m` ejemplo de integración numérica Euler.
-  `cap2_ejemplo26.m` ejemplo de un sistema dinámico lineal escalar.
-  `cap2_simuejemplo26.m` programa principal para simular el ejemplo `cap2_ejemplo26.m`.
-  `cap2_ejemplo27.m` ejemplo de un sistema dinámico lineal de segundo orden.
-  `cap2_simuejemplo27.m` programa principal para simular el ejemplo `cap2_ejemplo27.m`.

2.2 Fe de erratas



EL Capítulo 2 denominado **Métodos numéricos** tiene los siguientes errores de impresión:

-  Al final de la página 111 se indica

Valor analítico=33.333333

Simpson=33.333333

corresponde al intervalo $a=0$ y $b=10$.

El valor correcto es:

Valor analítico=21.081851

Simpson=21.081847

AlfaOmega

Parte II

Cinemática

AlfaOmega

3

Capítulo

Preliminares matemáticos















$$H_{i-1}^i = \begin{bmatrix} R_{i-1}^i & \mathbf{d}_{i-1}^i \\ \mathbf{0}^T & 1 \end{bmatrix}$$

 **Material Web**



3.1 Listado de programas del capítulo 3

L OS nombres de los programas con código fuente correspondientes al capítulo 3 se enlistan a continuación:

-  `cap3_prodint.m` ejemplo del producto punto vectorial o producto escalar.
-  `cap3_propiedadesRz.m` propiedades de la matriz de rotación $R_z(\theta)$
-  `cap3_rotacion.m` ejemplo de rotación alrededor del eje z .
-  `cap3_rotacionRz.m` ejemplo de rotación alrededor del eje z .
-  `H_DH.m` función que realiza la estructura de una matriz de transformación homogénea.
-  `HRx.m` matriz de transformación homogénea de rotación alrededor del eje x .
-  `HRy.m` matriz de transformación homogénea de rotación alrededor del eje y .
-  `HRz.m` matriz de transformación homogénea de rotación alrededor del eje z .
-  `HTx.m` matriz de transformación homogénea de traslación sobre el eje x .
-  `HTy.m` matriz de transformación homogénea de traslación sobre el eje y .
-  `HTz.m` matriz de transformación homogénea de traslación sobre el eje z .
-  `Rx.m` matriz de rotación alrededor del eje x .
-  `Ry.m` matriz de rotación alrededor del eje y .
-  `Rz.m` matriz de rotación alrededor del eje z .



3.2 Fe de erratas

EL Capítulo 3 **Preliminares matemáticos** tiene los siguientes errores de impresión:



Página 156 el contador 3.5 de la caja de código fuente **Propiedades de la matriz de rotación** $R_z(\theta)$ debe estar numerado como 3.4.



Página 158 en el comentario de la línea 3 del código fuente 3.4 **Rotación alrededor del eje** z_0 se indica:

$$\mathbf{p}_0 = \begin{bmatrix} p_{x0} \\ p_{y0} \\ p_{x0} \end{bmatrix}$$

debe ser:

$$\mathbf{p}_0 = \begin{bmatrix} p_{x0} \\ p_{y0} \\ p_{z0} \end{bmatrix}$$

AlfaOmega

4

Capítulo

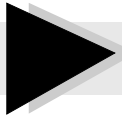
Cinemática directa

l_i	α_i	$d_i \dot{\beta}_i$	θ_i
-------	------------	---------------------	------------

$$H_{i-1}^i = H_{R_{z_i}}(\theta_i)H_{T_{z_i}}(d_i \dot{\beta}_i)H_{T_{x_i}}(l_i)H_{R_{x_i}}(\alpha_i)$$

 **Material Web**

AlfaOmega



4.1 Cinemática directa

L OS nombres de los programas con código fuente correspondientes al capítulo 4 se enlistan a continuación:



`tabla_DHCartesiano.m` despliega la tabla de parámetros Denavit Hartenberg del robot cartesiano de 3 gdl.



`tabla_DHCilindrico.m` despliega la tabla de parámetros Denavit Hartenberg del brazo robot en configuración cilíndrico de 3 gdl.



`tabla_DHEsferico.m` despliega la tabla de parámetros Denavit Hartenberg del brazo robot en configuración esférico de 3 gdl.



`tabla_DHr2gdl.m` despliega la tabla de parámetros Denavit Hartenberg del robot antropomórfico de 2 gdl.



`tabla_DHr3gdl.m` despliega la tabla de parámetros Denavit Hartenberg del robot antropomórfico 3 gdl.

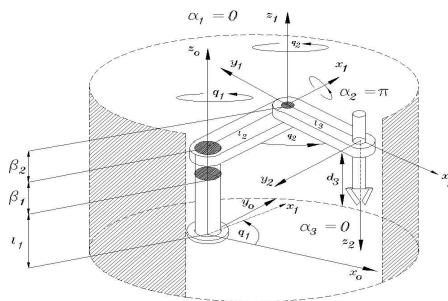


`tabla_DHpendulo.m` despliega la tabla de parámetros Denavit Hartenberg del péndulo robot.

5

Capítulo

Cinemática directa cartesiana



































Material Web











5.1 Listado de programas del capítulo 5

L OS nombres de los programas con código fuente correspondientes al capítulo 5 **Cinemática directa cartesiana** se enlistan a continuación:



-  `cap5_pendolo.m` programa principal para llevar a cabo el análisis de cinemática directa cartesiana de un péndulo robot.
-  `cap5_r2gdl.m` programa principal para llevar a cabo el análisis de cinemática directa cartesiana de un brazo robot antropomórfico de 2 gdl.
-  `cap5_r3gdl.m` programa principal para llevar a cabo el análisis de cinemática directa cartesiana de un robot antropomórfico de 3 gdl.
-  `cinematica_pendolo.m` cinemática directa cartesiana del péndulo.
-  `cinematica_r2gdl.m` cinemática directa cartesiana del brazo robot de 2 gdl.
-  `cinematica_r3gdl.m` cinemática directa cartesiana del robot antropomórfico de 3 gdl.
-  `cinv_pendolo.m` cinemática inversa del péndulo.
-  `cinv_r2gdl.m` cinemática inversa del robot de 2 gdl.
-  `cinv_r3gdl.m` cinemática inversa del robot de 3 gdl.
-  `H_pendolo.m` matriz de transformación homogénea del péndulo.
-  `H_r2gdl.m` matriz de transformación homogénea del brazo robot de 2 gdl.
-  `H_r3gdl.m` matriz de transformación homogénea del brazo robot de 3 gdl.
-  `H_SCARA.m` matriz de transformación homogénea del robot SCARA.
-  `SCARA.m` programa principal para realizar el análisis cinemático de un robot en configuración SCARA.
-  `SCARA1.m` aplicación del robot SCARA.

-  `cinv_SCARA` cinemática inversa del robot en configuración SCARA.
-  `H_esfercio.m` matriz de transformación homogénea del robot esférico.
-  `cinematica_esfercio.m` cinemática directa cartesiana del robot esférico.
-  `cinv_esferico.m` cinemática inversa del robot esférico.
-  `cap5_esferico.m` programa principal para realizar el análisis cinemático de un robot en configuración esférico.
-  `H_cilindrico.m` matriz de transformación homogénea de un robot en configuración cilíndrica.
-  `cinematica_cilindrico.m` cinemática directa cartesiana de un robot en configuración cilíndrica.
-  `cinv_cilindrico.m` cinemática inversa de un robot en configuración cilíndrica.
-  `cap5_cilindrico.m` programa principal para realizar el análisis cinemático de un robot en configuración cilíndrica.
-  `H_cartesiano.m` matriz de transformación homogénea del robot cartesiano.
-  `cinematica_cartesiano.m` cinemática directa de un robot en configuración cartesiana.
-  `cinv_cartesiano.m` cinemática inversa del robot cartesiano.
-  `cap5_cartesiano.m` programa principal para realizar el análisis cinemático de un robot en configuración cartesiano.
-  `Jacobianocartesiano.m` despliega la matriz jacobiana del robot cartesiano de 3 gdl.
-  `Jacobianoesferico.m` despliega la matriz jacobiana del robot esférico de 3 gdl.
-  `Jacobianocilindrico.m` despliega simbólicamente la matriz jacobiano del robot en configuración cilíndrica de 3 gdl.
-  `JacobianoSCARA.m` contiene el jacobiano del robot SCARA de 3 gdl.

-  `Jacobianor2gdl.m` despliega en forma simbólica el jacobiano del robot antropomórfico de 2 gdl.
-  `Jacobianor3gdl.m` despliega en forma simbólica y numérica el jacobiano del robot antropomórfico de 3 gdl.
-  `detcartesiano.m` contiene el determinante de la matriz jacobiana del robot cartesiano de 3 gdl.
-  `detesferico.m` contiene el determinante de la matriz jacobiana del robot en configuración esférica de 3 gdl.
-  `detcilindrico.m` contiene el determinante de la matriz jacobiana del robot cilíndrico de 3 gdl.
-  `detSCARA.m` contiene el determinante de la matriz jacobiana del robot SCARA de 3 gdl.
-  `detr2gdl.m` contiene el determinante de la matriz jacobiana del robot antropomórfico de 2 gdl.
-  `detr3gdl.m` contiene el determinante de la matriz jacobiana del robot antropomórfico de 3 gdl.

5.2 Fe de erratas

EL Capítulo 5 **Cinemática directa cartesiana** tiene los siguientes errores de impresión:

-  Página 207 en el cuadro de código fuente 5.1 **H_pendolo.m** en la línea 4 hay doble punto y coma (;;) sólo debe tener un operador punto y coma (;). Lo mismo sucede en la línea 9.
-  En la página 231 en el código fuente 5.11 **cinvr3gdl.m** en las líneas 9 y 11 la variable **z** debe ser sustituida por **z0**.



En la página 248 en el código fuente 5.18 **H_esferico.m** en la línea 3 `disp('Parámetros Denavit Hartenberg del robot SCARA')` debe ser `disp('Parámetros Denavit Hartenberg del robot Esférico')`.

Alfaomega

Parte III

Dinámica

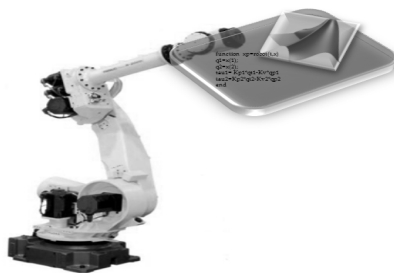


Material Web

6

Dinámica

Capítulo



















 **Material Web**

AlfaOmega



6.1 Listado de programas del capítulo 6

L OS nombres de los programas con código fuente correspondientes que incluye el capítulo 6 **Dinámica** se enlistan a continuación:

-  `mra.m` sistema masa resorte amortiguador.
-  `smr_simu.m` programa principal para simular el sistema masa resorte amortiguador.
-  `sle.m` sistema dinámico lineal escalar
-  `sle_simu.m` programa principal para simular un sistema dinámico lineal.
-  `filtro.m` implementación de un filtro pasa bajas.
-  `filtro_simu.m` programa principal para simular el filtro pasa bajas.
-  `centrifuga.m` implementa un sistema mecatrónico centrífuga.
-  `centrifuga_simu` programa principal para simular a la centrífuga.
-  `pendulo.m` sistema dinámico del péndulo.
-  `pendulo_simu.m` programa principal para simular la dinámica del péndulo.
-  `robot2gdl.m` implementa el modelo dinámico de un brazo robot de 2 gdl.
-  `cap6_robot2gdlsimu.m` programa principal para simular la dinámica de un robot de 2 gdl.
-  `robot3gdl.m` modelo dinámico de un robot antropomórfico de 3 gdl.
-  `robot3gdl_simu.m` programa principal para simular la dinámica del brazo robot de 3 gdl.
-  `robot_cartesiano3gdl.m` modelo dinámico de un robot cartesiano de 3 gdl.
-  `simu_robotcartesiano3gdl.m` programa principal para simular el modelo dinámico del robot cartesiano de 3 gdl.



6.2 Fe de erratas

EL Capítulo 5 **Cinemática directa cartesiana** tiene los siguientes errores de impresión:



En este capítulo 6 **Dinámica** el encabezado de los programas dice **capítulo 4 Dinámica**; debe decir **capítulo 6 Dinámica**.



Página 300 en el cuadro de código fuente 6.6 el nombre del archivo `sle_simu.m` ha sido sustituido en la página Web del libro por el nombre `filtro_simu.m`. Lo anterior se debe a que en la página 298, cuadro 6.4 ya existe un programa con ese nombre.



Página 303 código fuente 6.7, línea 9 la variable `tau` debe ser `tau1`.



Página 308 código fuente 6.9 del programa `pendulo.m` en la línea 19 la variable `gamma1` debe ser sustituida por `I1`.



Página 309 código fuente 6.10 del programa `pendulo_simu` línea 26 `subplot(2,2,3); plot(qp,friccion)` debe ser sustituido por `subplot(2,2,4); plot(qp1,friccion)`

AlfaOmega

7

Capítulo

Identificación paramétrica

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















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







 **Material Web**



7.1 Listado de programas del capítulo 7






L OS nombres de los programas con código fuente que incluye el capítulo 7 **Identificación paramétrica** se enlistan a continuación:

-  `mincuad.m` algoritmo recursivo de mínimos cuadrados para el caso escalar.
-  `mincuadm.m` algoritmo recursivo de mínimos cuadrados para el caso vectorial o multivariable.
-  `cap7_ejemplo1.m` identificación paramétrica de un sistema escalar
-  `cap7_ejemplo2.m` identificación paramétrica de un sistema escalar.
-  `cap7_ejemplo3.m` identificación paramétrica de un sistema escalar
-  `cap7_se1.m` implementación de un sistema dinámico lineal.
-  `cap7_ejemplo4.m` identificación paramétrica de un sistema dinámico escalar
`cap7_se1.m`.
-  `cap7_ejemplo5.m` identificación paramétrica de un sistema multivariable.
-  `cap7_pendulo.m` implementación del modelo dinámico del péndulo.
-  `cap7_pendulosimu.m` programa principal para identificación paramétrica del modelo dinámico del péndulo.
-  `cap7_pendulodinafiltsimu.m` programa principal para identificación paramétrica del modelo dinámico filtrado del péndulo.
-  `cap7_penduloenersimu.m` programa principal para identificación paramétrica del modelo de energía del péndulo.
-  `cap7_pendulopotsimu.m` programa principal para identificación paramétrica del modelo de potencia del péndulo.
-  `cap7_pendulopotfilsimu.m` programa principal para identificación paramétrica del modelo de potencia filtrada del péndulo.

-  `cap7_iderobot2gd1.m` modelo dinámico del robot de 2 gdl.
-  `cap7_iderobot2gdlsimu.m` programa principal para identificación paramétrica del modelo dinámico del robot de 2 gdl.
-  `cap7_energiarobot2gdlsimu.m` programa principal para identificación paramétrica del modelo de energía del robot de 2 gdl.
-  `cap7_potenciarobot2gdlsimu.m` programa principal para identificación paramétrica del modelo de potencia del robot de 2 gdl.
-  `cap7_idecartesiano3gd1.m` programa principal para identificación paramétrica del modelo dinámico del robot cartesiano de 3 gdl.
-  `cap7_potenciapcartesiano3gdlsimu.m` programa principal para identificación paramétrica del modelo de potencia del robot cartesiano de 3 gdl.

7.2 Fe de erratas

EL Capítulo 7 **Identificación paramétrica** tiene los siguientes errores de impresión:

-  Página 343 en el cuadro de código fuente 7.6 el archivo `cap7_se1.m` en la línea 1 `function xp =se1(t,x)` lo correcto debe ser: `function xp =cap7_se1(t,x)`
-  Página 348 en el cuadro de código 7.9 `cap7_pendulosimu.m` en la línea 3 faltó declarar: `t=ti:h:tf;`. Es decir, lo correcto debe ser `ti=0; h=0.001; tf = 5; t=ti:h:tf;`. Asimismo, en la línea 8 `xp=pendulo(t,x,9)`; debe ser `xp=cap7_pendulo(t,x,9)`;
-  Página 352 en el cuadro 7.11 línea 9 `xp=pendulo(t,x,9)`, deber ser `xp=cap7_pendulo(t,x,9)`
-  Página 354 en el cuadro 7.12 línea 8 `xp=pendulo(t,x,9)`, deber ser `xp=cap7_pendulo(t,x,9)`
-  Página 356 en el cuadro 7.13 línea 9 `xp=pendulo(t,x,9)`, deber ser `xp=cap7_pendulo(t,x,9)`



Página 358 en el cuadro 7.14 línea 9 `xp=pendulo(t,x,9)`, deber ser `xp=cap7_pendolo(t,x,9)`



Página 377 en el cuadro 7.20 línea 34 `theta=mincuadm(tau, fi,m,13,3)` debe ser `theta=mincuadm(tau, fi,m,10,3)`, debido a que son 10 parámetros y no 13.

AlfaOmega

Parte IV

Control

 **Material Web**

Alfa Romeo

AlfaOmega

8

Capítulo

Control de posición

$$\tau = \nabla \mathcal{U}_a(K_p, \tilde{q}) - f_v(K_v, \dot{q}) + g(q)$$



 **Material Web**







AlfaOmega



8.1 Listado de programas del capítulo 8




LOS nombres de los programas con código fuente correspondientes que incluye el capítulo 8 **Control de posición** se enlistan a continuación:

-  `cap8_pendolo.m` implementación del modelo dinámico del péndulo.
-  `cap8_pdpendulo.m` control proporcional derivativo para el péndulo.
-  `cap8_pdpendulosimu.m` programa principal que permite la simulación del modelo dinámico del péndulo y el control proporcional derivativo.
-  `cap8_robot2gdl.m` modelo dinámico de 2 gdl.
-  `cap8_PDrobot2gdl.m` control proporcional derivativo para el robot de 2 gdl.
-  `cap8_robot2gdlsimu.m` programa principal para simular el control proporcional derivativo sobre el robot de 2 gdl.
-  `cap8_robot3gdl.m` modelo dinámico de 3 gdl.
-  `cap8_PDrobot3gdl.m` control proporcional derivativo del robot de 3 gdl.
-  `cap8_robot3gdlsimu.m` simulación del control proporcional derivativo sobre el modelo dinámico del robot antropomórfico de 3 gdl.
-  `cap8_cartesiano3gdl.m` modelo dinámico del robot cartesiano de 3 gdl.
-  `cap8_PDcartesiano3gdl.m` control proporcional derivativo de 3 gdl.
-  `cap8_cartesiano3gdlsimu.m` simulación del control proporcional derivativo sobre el robot cartesiano de 3 gdl.
-  `cap8_PIDrobot2gdl.m` control proporcional integral derivativo de un robot de 2 gdl.
-  `cap8_robot2gdlPID.m` modelo dinámico del robot de 2 gdl acoplado al control PID.
-  `cap8_robot2gdlPIDsimu.m` simulación del control PID sobre el robot de 2 gdl.

-  `cap8_robot2gd1TANH.m` modelo dinámico del robot de 2 gdl con entrada de control tangente hiperbólica.
-  `cap8_TANHrobot2gd1.m` control tangente hiperbólica de un robot de 2 gdl.
-  `cap8_TANHrobot2gd1simu.m` simulación del control tangente hiperbólica sobre un robot de 2 gdl.
-  `cap8_atanflor8p.m` control arco tangente para el seguimiento de una trayectoria tipo flor de 8 pétalos.
-  `cap8_robot2gd1flor8p.m` modelo dinámico de un robot de 2 gdl acoplada al control arco tangente.
-  `cap8_robot2gdlatansimu` programa principal para que un robot de 2 gdl trace una flor con 8 pétalos.

▶ 8.2 Fe de erratas

EL Capítulo 8 **Control de posición** contiene los siguientes errores tipográficos de impresión:

-  Página 402 en el cuadro de código fuente 8.3 en la línea 9 `opciones=odeset('RelTol', 1e-3, ÍnitialStep', 1e-3, 'MaxStep', 1e-3);` la variable `InitialStep` se encuentra acentuada. Lo correcto deber ser: `opciones=odeset('RelTol', 1e-3, 'InitialStep', 1e-3, 'MaxStep', 1e-3);`
-  Página 412 en el cuadro de código fuente 8.9 en la línea 17 `[qt tau] = cap8_PDrobot3gd1([q1(k);q2(k)q3(k)], [qp1(k);qp2(k); qp3(k)]);` le falta un punto y coma (;) entre `q2(k)` y `q3(k)`. Lo correcto debe ser: `[qt tau] = cap8_PDrobot3gd1([q1(k);q2(k);q3(k)], [qp1(k);qp2(k); qp3(k)]);`
-  Página 428 del código fuente 8.21 en la línea 16 `%ley de control PD` debe decir: `%ley de control arco tangente`

AlfaOmega