

TUTORIAL MATLAB

COMANDOS UTILIZADOS EN TEORÍA DE LOS CIRCUITOS Y ANÁLISIS Y DISEÑO DE SISTEMAS DE CONTROL

COMANDOS DE ASIGNACIONES DE VARIABLES

<code>a = 5</code> or <code>a = 5;</code>	Con y sin salida de pantalla
<code>A = [1 2 3];</code>	Asignación de array de una dimensión
<code>B = [1 2 3; 4 5 6; 7 8 9];</code>	Asignación de array de dos dimensiones
<code>c = B(2,3);</code>	Selección de un elemento de un array
<code>c = B(1:3,2);</code>	<code>c</code> = columna 2 de <code>B</code>
<code>d = B(3,:);</code>	Última fila de un array
<code>C = [0:2:100]</code>	Asignación de todos los números pares de un array hasta 100 en el vector <code>C</code>
<code>D = linspace(5, 50, 1000);</code>	Genera un vector <code>D</code> de 1000 elementos entre 5 y 50 con igual espaciamiento
<code>D = logspace(-1, 2, 1000);</code>	Genera un vector <code>D</code> de 1000 elementos con igual espaciamiento entre 10^{-1} to 10^2

COMANDOS GENERALES DE DISPOSITIVO

<code>pwd, cd, ls</code>	Comandos de directorio
<code><up arrow></code>	Último comando
<code>abc<up arrow></code>	Último comando comenzando en <code>abc</code>
<code>help command_name</code>	Ayuda sobre el comando "NAME" mas útil que el HELP genérico del menú principal
<code>m = mean(Y);</code>	Llamado de una función
<code>save data A B C -ascii -double -tab</code>	Salva las variables <code>A, B, C</code> en el archivo "data.mat" en formato doble, ASCII.
<code>load data</code>	Carga la variable almacenada en el archivo "data.mat"
<code>load data.txt</code>	Carga un array de números almacenados en ASCII desde el archivo "data.txt"
<code>filename</code>	Ejecuta el archivo "m" llamado "filename.m" que contiene los comandos de MATLAB

PLOTEO Y GRAFICACIÓN

<code>plot(x,y)</code>	Grafica el vector <code>y</code> en función del vector <code>x</code>
<code>plot(y)</code>	Siendo <code>y</code> un vector complejo, grafica $\text{Im}\{y\}$ versus $\text{Re}\{y\}$
<code>plot(t,[x1,x2,x3])</code>	Grafica <code>x1,x2,x3</code> versus <code>t</code> sobre el mismo gráfico
<code>plot(x1,y1,'r',x2,y2,'g')</code>	Grafica <code>y1</code> versus <code>x1</code> in "rojo" y <code>y2</code> versus <code>x2</code> in "verde"
<code>subplot(r,c,p)</code>	Define el grafico actual como "p" en graficaciones múltiples de "r" por "c"
<code>title('text'), xlabel('text'), ylabel('text')</code>	Define a title and labels for the x and y axis
<code>axis([xmin, xmax, ymin, ymax])</code>	Defines la escala de los ejes
<code>semilogx(x,y), semilogy(x,y)</code>	Identico a <code>plot(x,y)</code> pero con ejes <code>x</code> o <code>y</code> semilogaritmicos
<code>grid</code>	Dibuja la grilla en el gráfico actual

Menú de edición en la figura ("edit") copia una figura al "clipboard" y se puede pegar en otra aplicación ej. "Word"

MODELOS DE FUNCIONES DE TRANSFERENCIA

Los modelos de funciones de transferencia son representados mediante dos polinomios, uno numerador y el otro denominador.

<code>n1 = [1 5];</code>	Define el polinomio $s + 5$
<code>d1 = [1 15 10];</code>	Define el polinomio $s^3 + s^2 + 15s$
<code>printsys(n1,d1);</code>	Función para un sistema TF como polinomio racional
<code>n = poly(V);</code>	Genera un polinomio "n" el cual tiene raíces dadas por el vector "V"
<code>V = roots(n);</code>	Calcula las raíces de un polinomio "n"
<code>A = tf(n1,d1)</code>	Genera una función de transferencia con numerador "n1" y denominador "d1"
<code>n3 = conv(n1,n2);</code>	Calcula el producto de dos polinomios

MODELOS DE FUNCIONES DE TRANSFERENCIA (Continuación)

`[n3,d3] = series(n1,d1,n2,d2);` Calcula el producto de dos funciones de transferencia $n3/d3 = n1/d1 * n2/d2$
`[n2,d2] = cloop(n1,d1,-1);` Calcula la función de transferencia de lazo cerrado con realimentación negativa unitaria.
`[n3,d3] = feedback(n1,d1,n2,d2,-1);` Idem con realimentación negativa $n2/d2$
`[r1,p1,k1] = residue(n1,d1);` Determina los residuos de los polos en la expansión en Frac. Parciales Simples de $n1/d1$

ANÁLISIS DE RESPUESTA EN FRECUENCIA

`bode(n1,d1)` Genera el gráfico de Bode de la función $n1/d1$
`bode(A)` Genera el gráfico de Bode de la función de transferencia "A" si se define como $A = tf(n1/d1)$
`bode(n1,d1,w)` Genera el gráfico de Bode de la función $n1/d1$ entre las frecuencias dadas por el vector w (en rps)
`[mag,phase,w] = bode(n1,d1);` Calcula módulo y fase a las frecuencias " w "
`F = freqs(n1,d1,w);` Calcula la respuesta de frecuencia compleja de $n1/d1$ a las frecuencias " w "
`[Gm,Pm,Wcg,Wcp] = margin(n1,d1)` Calcula el margen de ganancia y de fase de la función $n1/d1$
`nyquist(n1,d1)` Genera el gráfico de Nyquist de la función $n1/d1$
`nyquist(A)` Genera el gráfico de Nyquist de la función de transferencia "A" si se define como $A = tf(n1/d1)$
`nyquist(n1,d1,w)` Genera el diagrama de Nyquist de $n1/d1$ a las frecuencias " w "
`[r1,i1,w] = nyquist(n1,d1,w)` Calcula la parte real e imaginaria de respuesta a la frecuencia compleja

LUGAR DE RAICES

`rlocus(n1,d1)` Calcula y grafica el lugar de raíces de $n1/d1$
`rlocus(n1,d1,k)` Calcula y grafica el lugar de raíces de $n1/d1$ a un valor dado de ganancia " k "
`R = rlocus(n1,d1,k)` Calcula la matriz de raíces para valores de la ganancia " k "
`[R,k] = rlocus(n1,d1)` Selección automática de valores de la ganancia " k "
`[p,k] = rlocfind(n1,d1)` Encuentra el valor de la raíces (polos) en el punto del gráfico del lugar de raíces señalado por el "mouse"
`pzmap(n1,d1)` Calcula y grafica los ceros y polos de $n1/d1$
`sgrid` Genera una grilla de igual relación de aspecto " x " e " y "

RESPUESTA EN EL TIEMPO

`step(n1,d1)` Calcula y grafica la respuesta escalón de $n1/d1$
`impulse(n1,d1)` Calcula y grafica la respuesta impulsiva de $n1/d1$
`step(n1,d1,t)` or `impulse(n1,d1,t)` Calcula y grafica la respuesta escalón de $n1/d1$ a un tiempo " t " especificado
`lsim(n1,d1,u,t)` Simulación lineal para la entrada " u " sobre el tiempo " t "