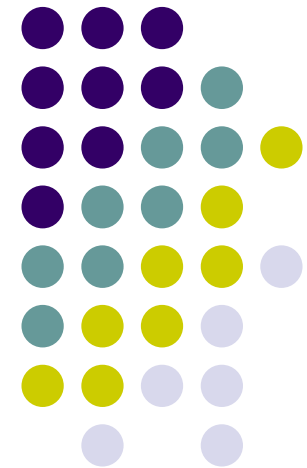


# Sistemas de Control Digital con Matlab y Labview

Ing. Juan Manuel Chaparro  
Universidad Central

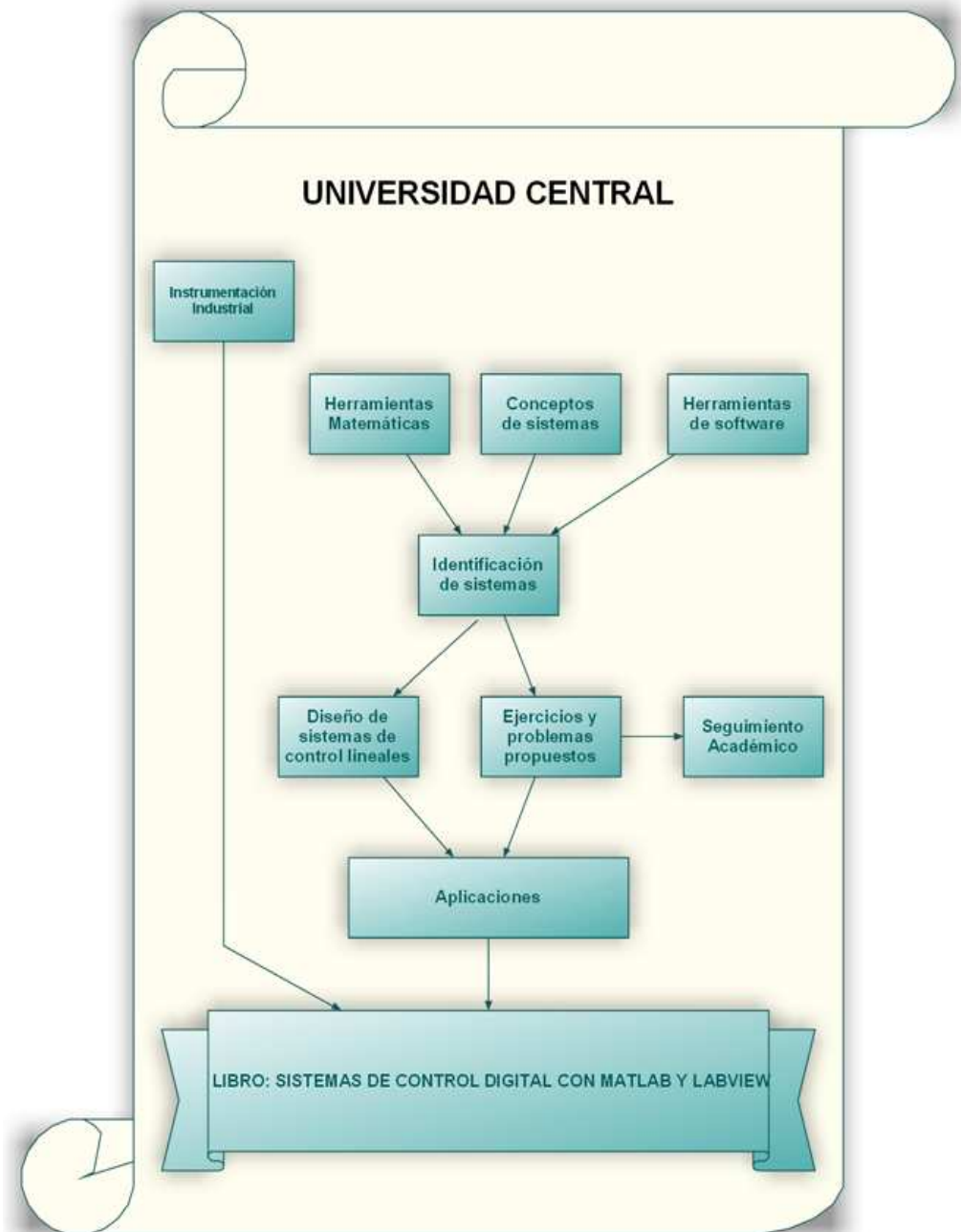


# PROBLEMÁTICA



- Proliferación de muchos textos teóricos difíciles de entender para los estudiantes y con pocos ejercicios prácticos.
- Aterrizaje de los conceptos dados teóricamente por medio de simulaciones.
- Utilización de herramientas de software de bastante uso a nivel académico como son Matlab y Labview.
- Aplicación de herramientas especializadas en Matlab como son Ident, Sisotool, Guide y Simulink.

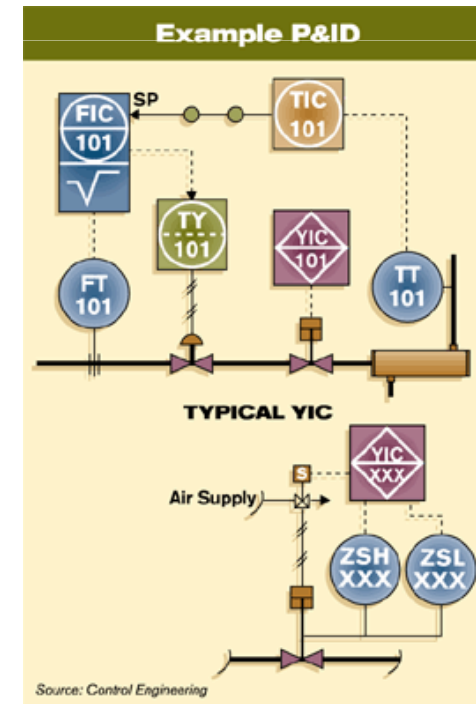
# MAPA CONCEPTUAL DEL LIBRO



# CAPITULO 0: INSTRUMENTACIÓN INDUSTRIAL



- Introducción a los procesos industriales y su simbología
- Definiciones para el análisis de instrumentos, sistemas de medición y control
- Símbolos internacionales de instrumentación
- Descripción de símbolos
- Ejemplo de diagrama P&ID
- Identificación de los instrumentos
- Procesos industriales

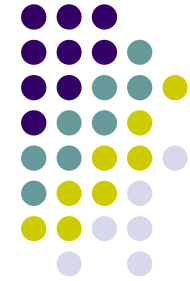


# CAPITULO 1: CONCEPTOS RELACIONADOS



- Sistemas de control en tiempo discreto
- Sistemas de control en tiempo continuo y en tiempo discreto
- Sistemas de control continuo
- Sistemas de control digital
- Señales continuas y discretas
- Sistemas de adquisición, conversión y distribución de datos.
- Implementación sistema de control digital utilizando puerto serial
- Herramientas para adquirir datos por el puerto serial utilizando matlab
- Programa para toma y envío de datos con matlab
- Ejemplos de programas utilizando labview
- Adquisición y distribución de datos por puerto serial para control de nivel
- Sistema de adquisición y distribución de datos para temperatura y nivel

# CAPITULO 1: HERRAMIENTAS DE MATLAB UTILIZADAS



- Otro programa para generar la señal pseudoaleatoria que se enviará al puerto serial. Para esto, es necesario tener la herramienta IDENT de Matlab para generar la señal PRBS:
- `s=serial('COM1');`
- `fopen(s);`
- `entrada=idinput(2000,'PRBS',[0 0.25],[10 30])` %Generación señal aleatoria . Se debe tener Ident de Matlab
- `for i=1:2000`
- `sal=entrada(i,1);`
- `f(i,1)=sal;` %salida aleatoria hacia el micro
- `fwrite(s,sal,'char','sync');` %Envío de información al puerto serial
- `fclose(s)` % Se cierra el Puerto para limpiar el buffer de datos.
- `fopen(s)`
- `a=fread(s,s.inputbuffersize,'char');` %Adquisición de información del Puerto serial
- `c(i,1)=a;` %entrada de la señal del sensor proveniente del micro
- `d(i,1)=i;`
- `pause(10)`
- `i=i+1`
- `end`
- `fclose(s);`
- `subplot(2,1,1) %Grafica la información obtenida`
- `plot(d,f,'-')`
- `subplot(2,1,2)`
- `plot(d,c,'o')`

# CAPITULO 2: HERRAMIENTAS MATEMÁTICAS UTILIZADAS EN LOS SISTEMAS DE CONTROL DIGITAL



- La transformada Z
- Transformada Z para funciones básicas
- Función escalón unitario
- Función rampa unitaria
- Función polinomial  $ak$
- Función exponencial
- Función senoidal
- Utilización de Matlab para encontrar la transformada Z de una expresión
- Transformada z inversa
- Método de la división directa
- Método computacional
- Utilización de comandos especiales de Matlab para encontrar la transformada z inversa de una expresión
- Transformada z inversa utilizando ecuaciones en diferencias
- Transformada z inversa utilizando fracciones parciales
- Método de los residuos o de la integral de inversión.

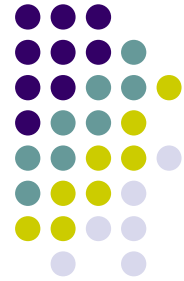
# CAPITULO 2: HERRAMIENTAS UTILIZADAS DE MATLAB



- COMANDOS ESPECIALES:
- ztrans: Transformada Z
  - - a.  $n = \text{sym}('n')$
  - - b.  $f = n^4$
  - - c. `ztrans(f)`
  - - d.  $z^*(z^3+11*z^2+11*z+1)/(z-1)^5$
- Iztrans: Transformada Z inversa
  - - a.  $z = \text{sym}('z')$
  - - b.  $f = 2*z/(z-2)^2$
  - - c. `iztrans(f)`
- Función de transferencia: `filter`
- Graficación: `plot`

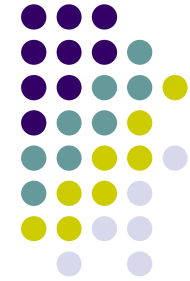


# CAPITULO 3: CONCEPTOS DE UN SISTEMA DIGITAL



- Sistema muestreador
- Circuitos para retención de datos
- Reconstrucción de señales originales a partir de señales muestreadas
- - Teorema del muestreo
- La función de transferencia pulso
- - Lazo abierto
- - Lazo cerrado
- Correspondencia entre el plano  $s$  y el plano  $z$

# CAPITULO 4: IDENTIFICACIÓN DE SISTEMAS LINEALES



- Concepto de sistema
- Modelo de un sistema
- Métodos de identificación
- Técnicas de identificación paramétrica
- Tipos de modelos parametricos
- Métodos para el ajuste de parámetros
- Consideraciones para identificación
- Obtención de datos
- Pretratamiento de datos
- Validación del modelo
- Reducción del modelo

# CAPITULO 4: HERRAMIENTAS UTILIZADAS DE MATLAB

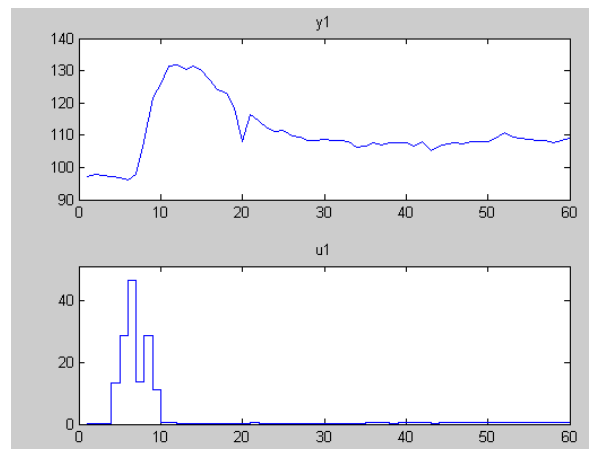


- System Identification Toolbox: Ident
- - Comandos
- - Interfaz grafica

# Identificación de sistemas utilizando IDENT



- `datos=[XT FT]` % Configuración de los datos. Se coloca primera la variable de salida XT y después la variable de entrada FT. Deben tener el mismo tamaño.
- `tam=length(FT)` % Cantidad de datos de la variable de entrada FT.
- `datos_ident=[XT(1:60) FT(1:60)]` % Cantidad de datos tomados para la validación del sistema. Para este caso, se toman los siguientes 60 datos tanto de entrada como de salida.
- `datos_val=[XT(61:tam) FT(61:tam)]` % Cantidad de datos tomados para la identificación del sistema. Para este caso, se toman los primeros 61 datos tanto de entrada como de salida.
- `idplot(datos_ident)` % Visualizar los datos tomados para identificación.



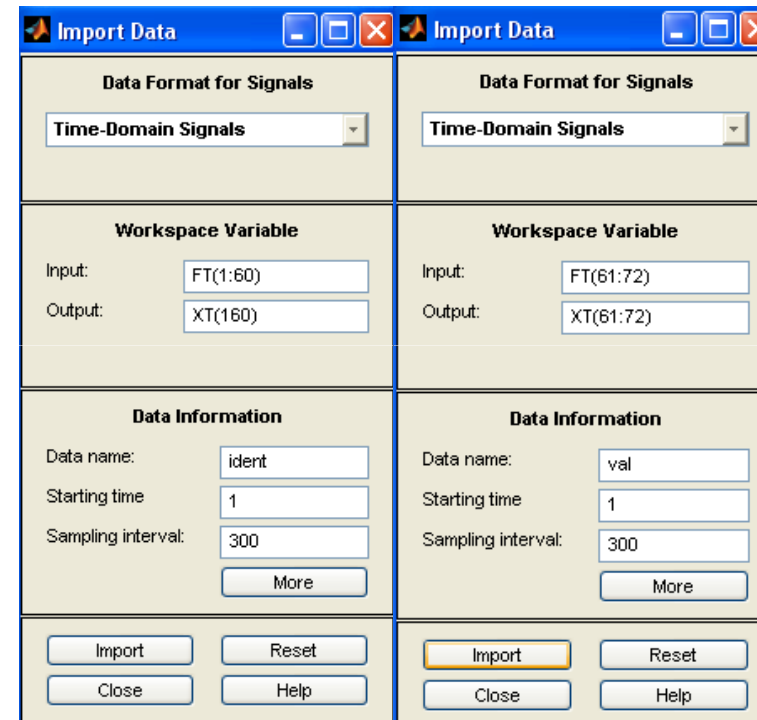
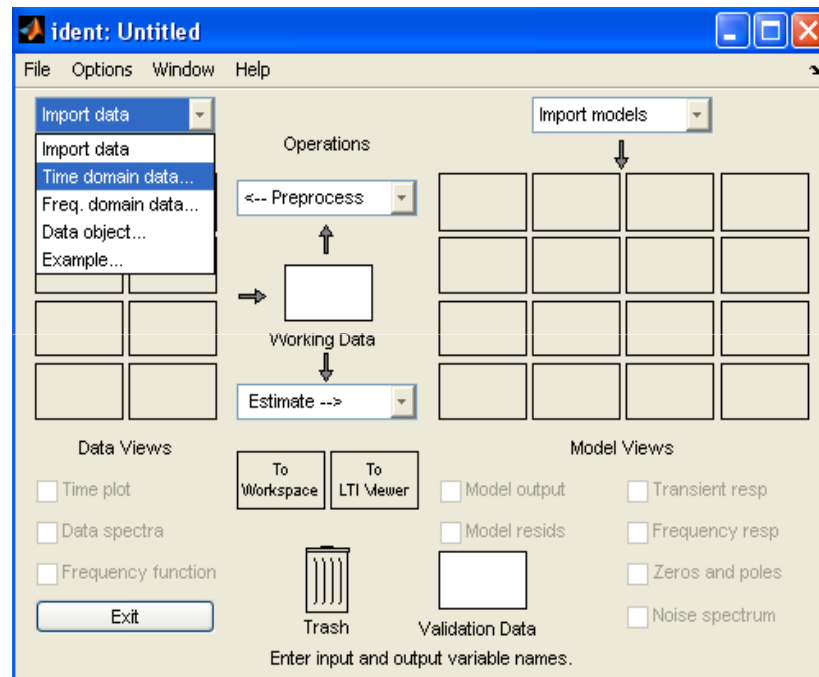
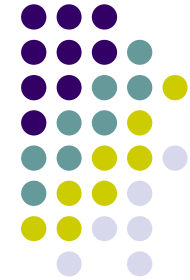
# Identificación de sistemas utilizando IDENT



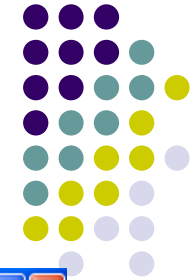
- `datos_ident=dtrend(datos_ident)` % % Remueve las tendencias lineales de los datos de identificación, manteniendo la información de la dinámica del sistema, pero no su comportamiento estático.
- `datos_val=dtrend(datos_val)` % Remueve las tendencias lineales de los datos de validación, manteniendo la información de la dinámica del sistema, pero no su comportamiento estático.
- `idplot(datos_ident)` % Visualiza los datos de identificación sin tendencia lineal.
- `idplot(datos_val)` % Visualiza los datos de validación sin tendencia lineal.
- `th=arx(datos_ident,[2 7 6])` % Aplicación del modelo posible. Para este caso es ARX. Puede ser ARMAX, OE y BJ. Se debe tener presente los parámetros que maneja cada uno.
  
- Discrete-time IDPOLY model:  $A(q)y(t) = B(q)u(t) + e(t)$
- $A(q) = 1 - 0.3144 q^{-1} - 0.3001 q^{-2}$
- $B(q) = 0.1531 q^{-6} + 0.07232 q^{-7} + 0.02384 q^{-8} + 0.05164 q^{-9} + 0.1027 q^{-10} - 0.008651 q^{-11} - 0.03379 q^{-12}$
- Estimated using ARX from data set `datos_ident`
- Loss function 2.16316 and FPE 2.92663
- Sampling interval: 1



# Identificación de sistemas utilizando el GUI de Ident



# Identificación de sistemas con el GUI de Ident



The main window of the Ident software, titled 'ident: Untitled'. It features a menu bar (File, Options, Window, Help) and several functional areas:

- Import data:** A dropdown menu and a grid of data plots (ident, val, identd, vald).
- Operations:** A central area with a 'Preprocess' dropdown and an 'Estimate' button.
- Import models:** A grid of model thumbnails (bj31416, arx276, oe617).
- Data Views:** Checkboxes for 'Time plot', 'Data spectra', and 'Frequency function'.
- Model Views:** Checkboxes for 'Model output', 'Model resid', 'Transient resp', 'Frequency resp', 'Zeros and poles', and 'Noise spectrum'.
- Workspace:** Buttons for 'To Workspace' and 'To LTI Viewer'.
- Validation Data:** A 'Validation Data' button and a 'Trash' icon.

A message at the bottom states: "The object bj21416 is now in the trash."

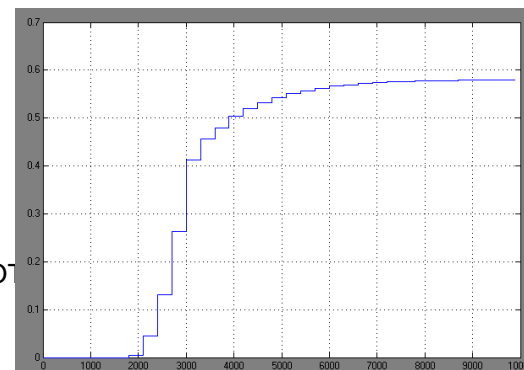
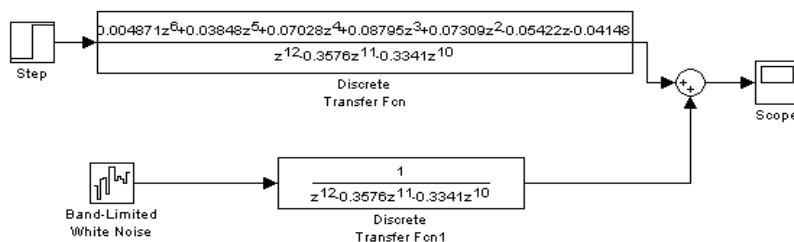
The 'Data/model Info: bj31416' window displays the following information:

- Model name:** bj31416
- Color:** [0.75,0.75,0.2]
- Discrete-time IDPOLY model:**  $y(t) = [B(q)/F(q)]u(t) + [C(q)/D(q)]e(t)$
- Transfer Functions:**
  - $B(q) = 0.1049 q^{-6} - 0.003257 q^{-7} + 0.05643 q^{-8}$
  - $C(q) = 1 + 0.7843 q^{-1}$
  - $D(q) = 1 + 0.7771 q^{-1} + 0.07291 q^{-2} + 0.06066 q^{-3} - 0.02986 q^{-4}$
  - $F(q) = 1 - 0.8437 q^{-1}$
- Estimation Info:** Estimated using PEM from data set identd. Loss function 1.78884 and FPE 3.71529. Sampling interval: 300.
- Diary And Notes:** A text area containing the following MATLAB commands:
 

```
% Import ident
      identd = dtrend(ident,1)

      bj31416 = bj(identd,[3 1 4 1 6],)
      Warning: Reference to non-existent field 'domain'.
```

Buttons for 'Present', 'Close', and 'Help' are located at the bottom.



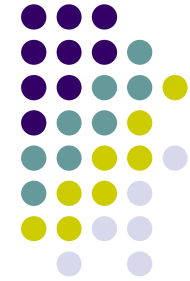


# CAPITULO 5: CONCEPTOS DE ESTABILIDAD EN UN SISTEMA DE CONTROL EN TIEMPO DISCRETO

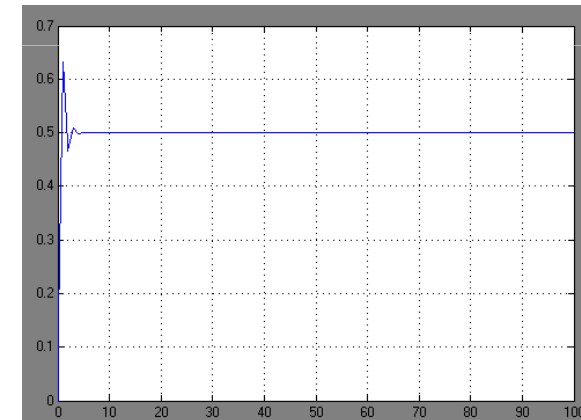
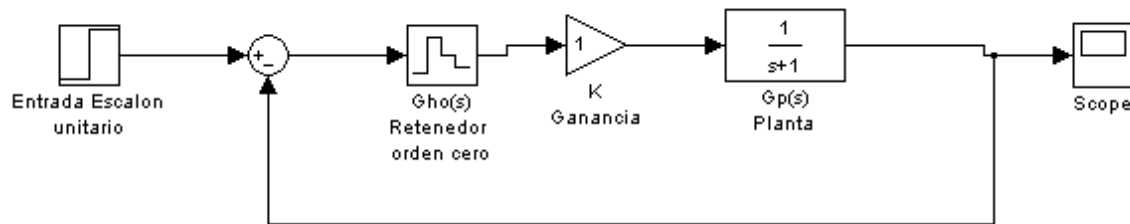


- Criterio de estabilidad de Jury
- Transformación bilineal y criterio de estabilidad de Routh-Hurwitz
- Método del lugar geométrico de las raíces
- Utilización de la herramienta SISOTOOL de Matlab para obtener el lugar geométrico de las raíces de un sistema de control discreto.

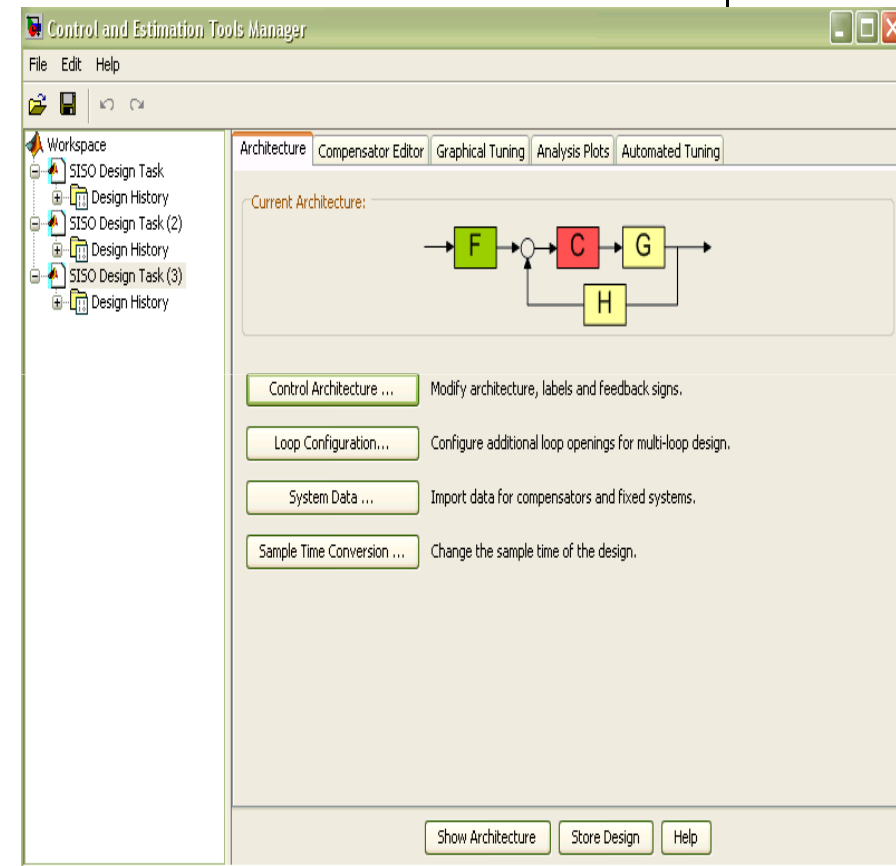
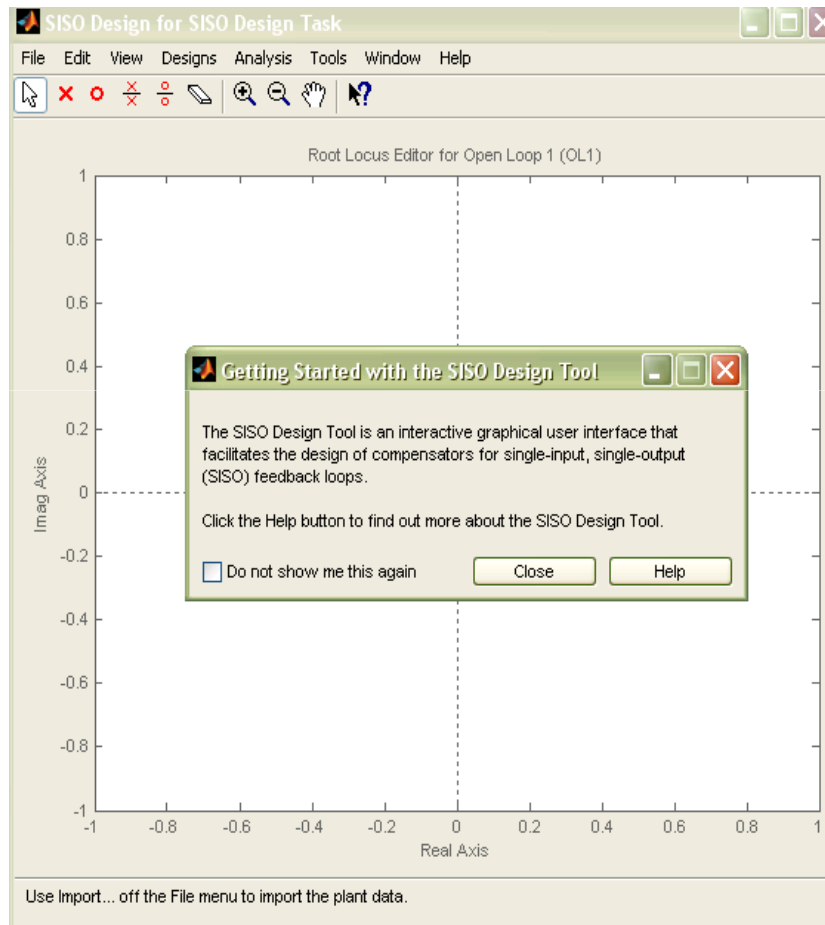
# CAPITULO 5: HERRAMIENTAS DE MATLAB UTILIZADAS



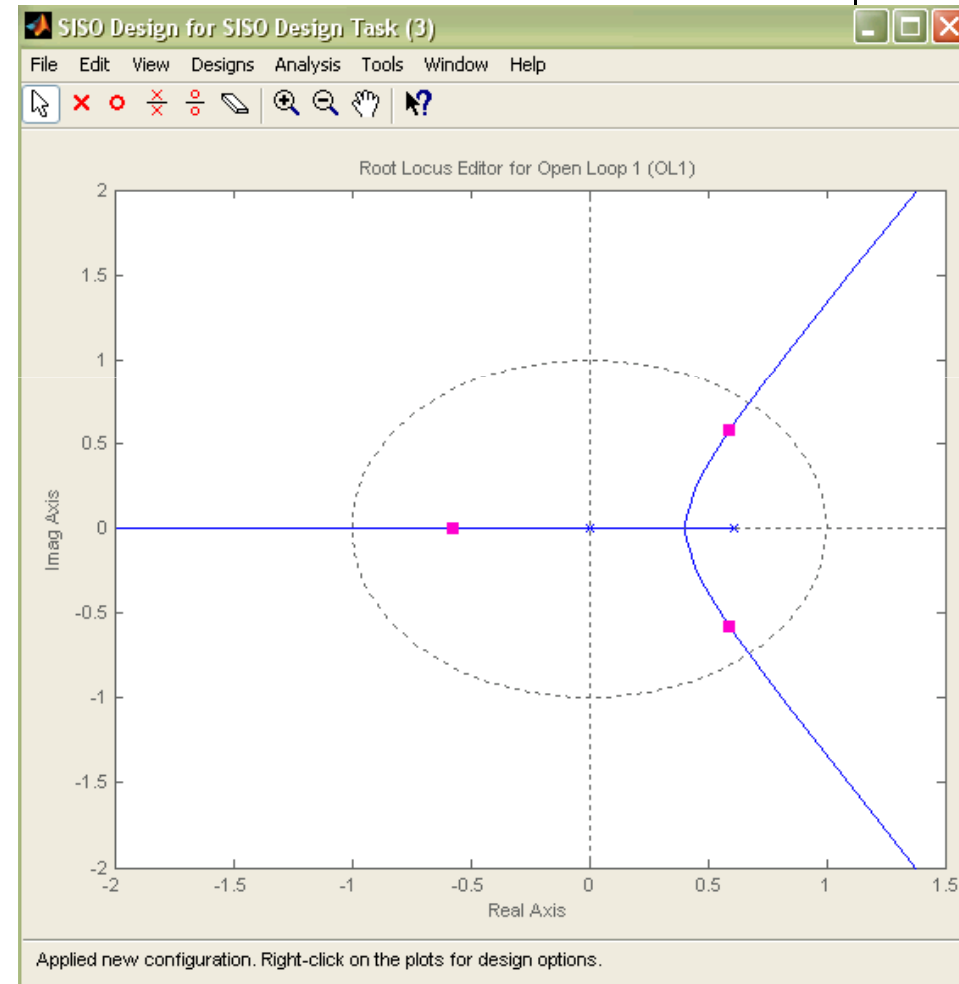
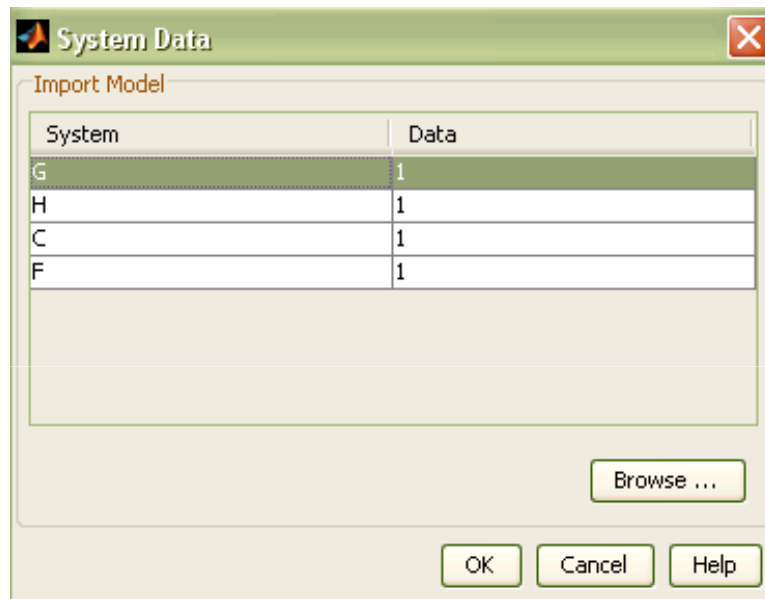
- Simulink
- Sisotool: rltool



# CAPITULO 5: HERRAMIENTAS DE MATLAB UTILIZADAS



# CAPITULO 5: HERRAMIENTAS DE MATLAB UTILIZADAS

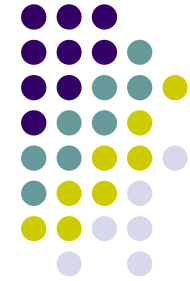


# CAPITULO 6: RESPUESTA DE LOS SISTEMAS EN TIEMPO DISCRETO

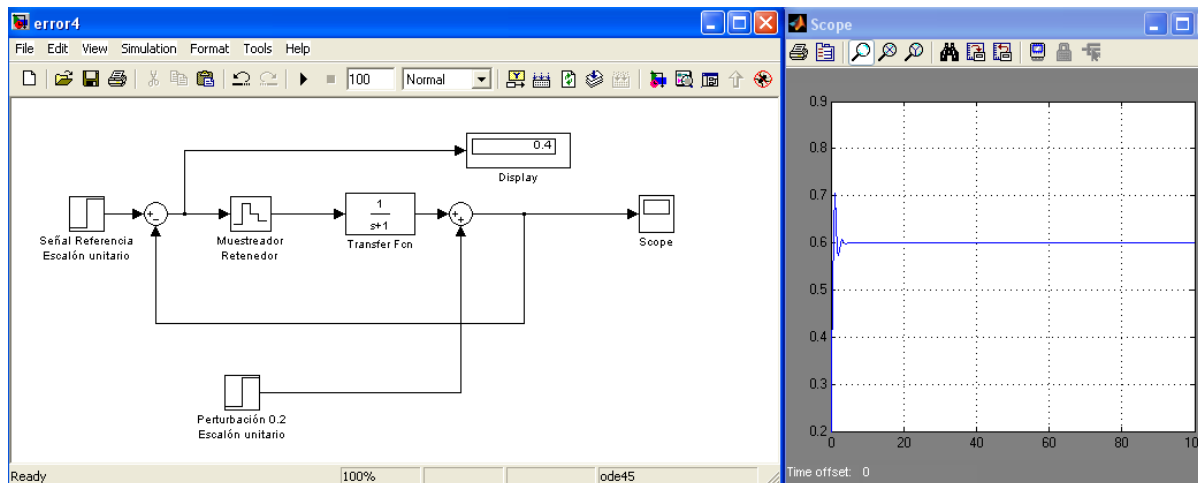
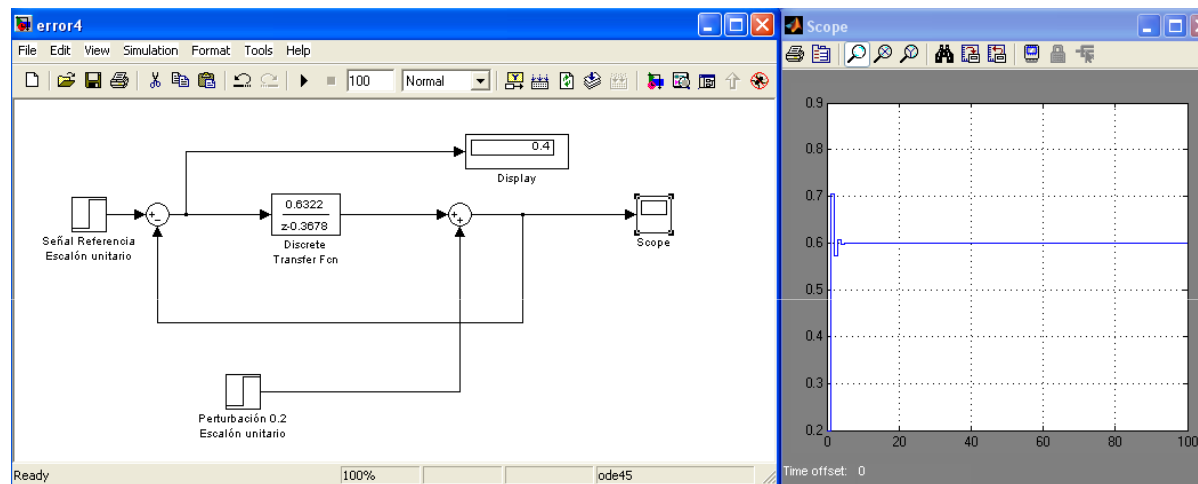


- Respuesta estacionaria de los sistemas
- Análisis de error en estado permanente
- Error a escalón unitario o señal de posición
- Error a rampa unitaria o señal de velocidad
- Error a entrada parabólica o señal de aceleración
- Robustez de un sistema discreto

# CAPITULO 6: HERRAMIENTAS UTILIZADAS DE MATLAB



- SIMULINK



# CAPITULO 7: DISEÑO DE COMPENSADORES DISCRETOS

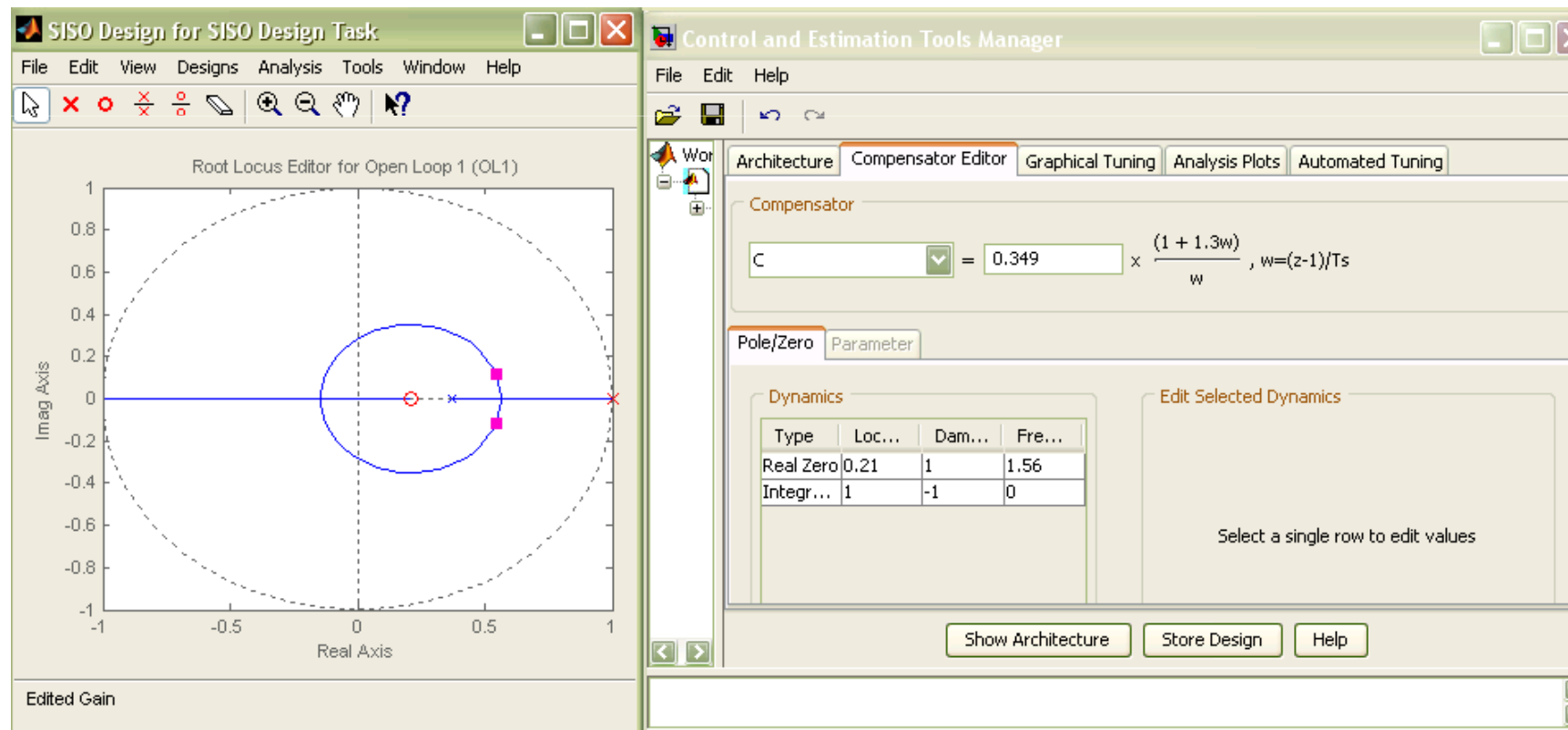


- Diseño de compensadores discretos a partir de especificaciones temporales
- Diseño de controladores discretos
- Acción proporcional
- Acción derivativa
- Acción integral
- Control PI discreto
- Control PD discreto
- Control PID discreto
- Implementación de un controlador PID discreto con LabVIEW
- Implementación de un controlador PID discreto utilizando la herramienta GUIDE de Matlab

# CAPITULO 7: HERRAMIENTAS UTILIZADAS DE MATLAB

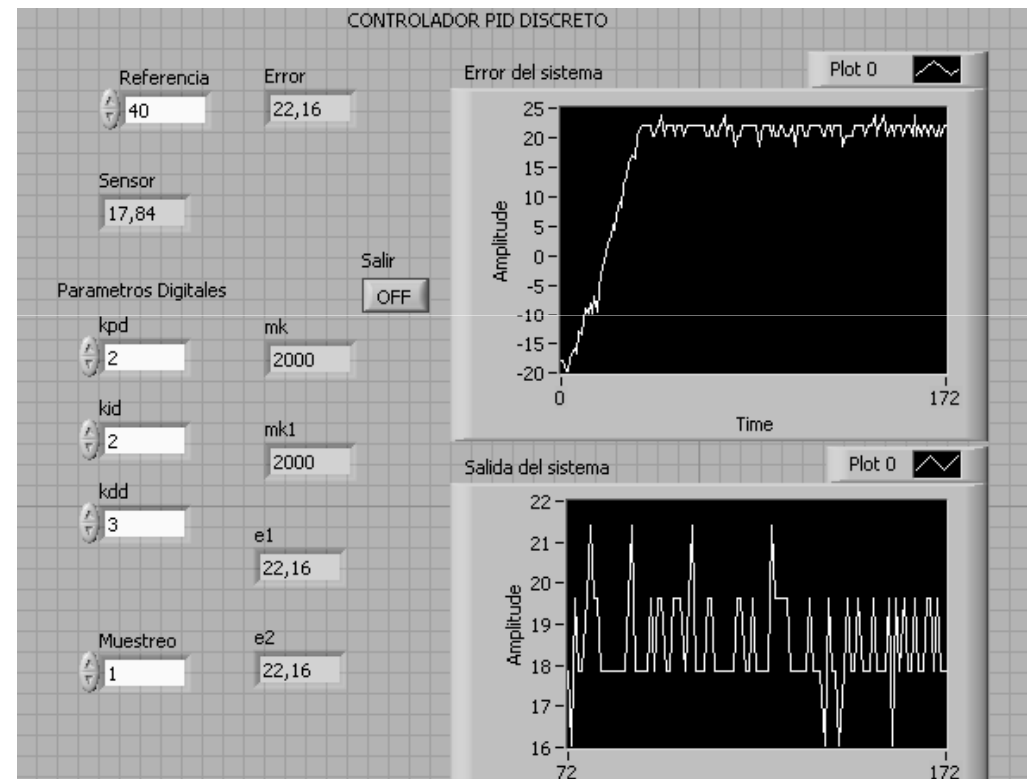
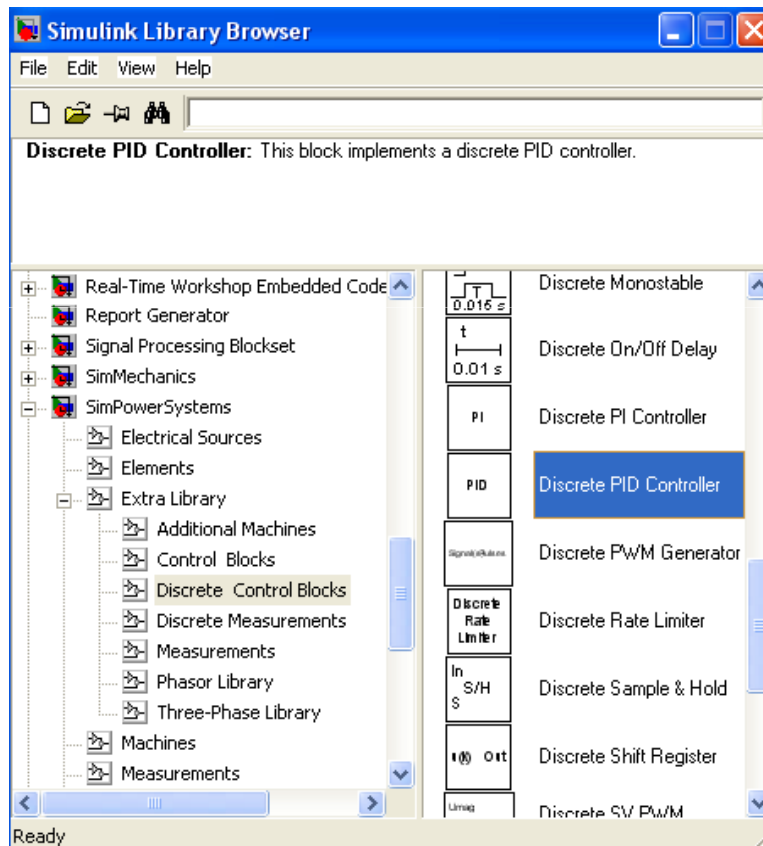


- GUIDE: Interfaz gráfica de Usuario
- Simulink
- Sisotool





# CAPITULO 7: HERRAMIENTAS UTILIZADAS DE MATLAB





**MUCHAS GRACIAS!!!**