

# Práctica 1.

## Control de un proceso industrial

---



### Informática Industrial y Robótica

Área de Ingeniería de Sistemas y Automática  
Departamento de Informática  
Universidad de Almería





# ÍNDICE



## Práctica 1. Control de un proceso industrial

### Objetivos

1. Descripción del proceso real: maqueta 4 tanques.
2. Introducción a Matlab y Simulink
3. Implementación de modelos utilizando Simulink.

- Identificar los distintos elementos de un sistema de control.
- Mostrar las dos fases fundamentales del proceso de simulación: la definición y construcción del modelo y, una segunda, correspondiente a su análisis y simulación.
- Estudiar la repuesta ante entradas en escalón de sistemas físicos (no lineales) que se pueden representar por modelos simplificados, así como observar el efecto de cada uno de los parámetros que caracterizan a estos tipos de sistemas.
- Breve introducción al entorno Simulink para el proceso de Simulación.
- Obtener modelos empíricos de procesos reales a partir de datos de entrada/salida obtenidos con el método de la curva de reacción.
- Diseñar controladores PI desde varios puntos de vista entendiendo el papel que juega cada uno de los parámetros de sintonía de este controlador.

# Práctica 1.

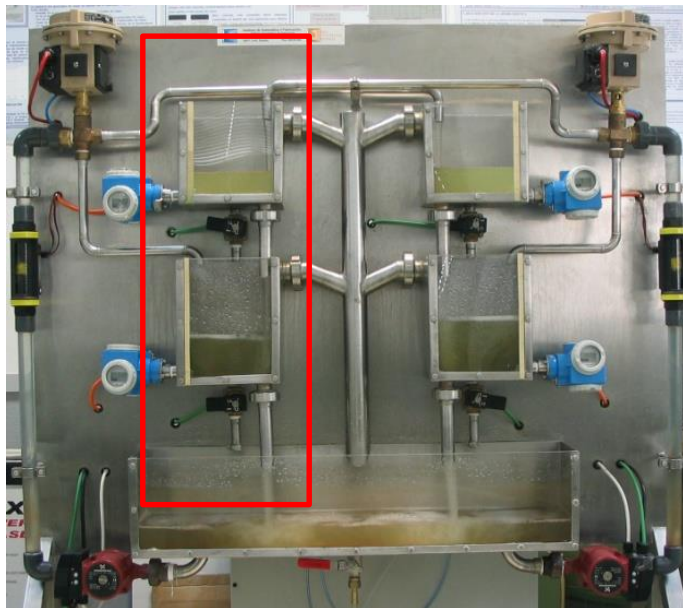
## Control de un proceso industrial



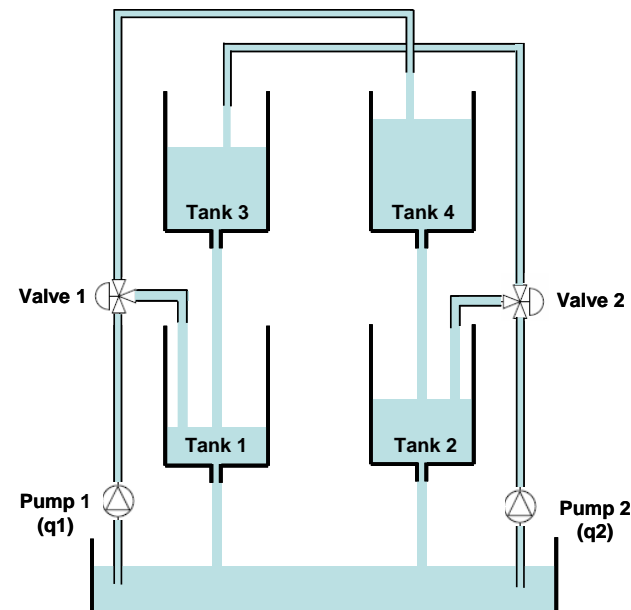
### 1. Descripción del proceso real: Maqueta de 4 tanques

# Maqueta de 4 tanques

El objetivo de esta planta es controlar la altura de los dos tanques inferiores actuando sobre las bombas.



(a) Planta Real



(b) Esquema de la planta



# Modelado de nivel de un único tanque



El modelo no lineal teórico que describe cómo varía la altura del tanque,  $h(t)$ , con respecto al caudal de entrada,  $q(t)$ , viene dado por:

$$\frac{dh(t)}{dt} = -\frac{a}{A} \sqrt{2gh(t)} + \frac{1}{A} q(t)$$

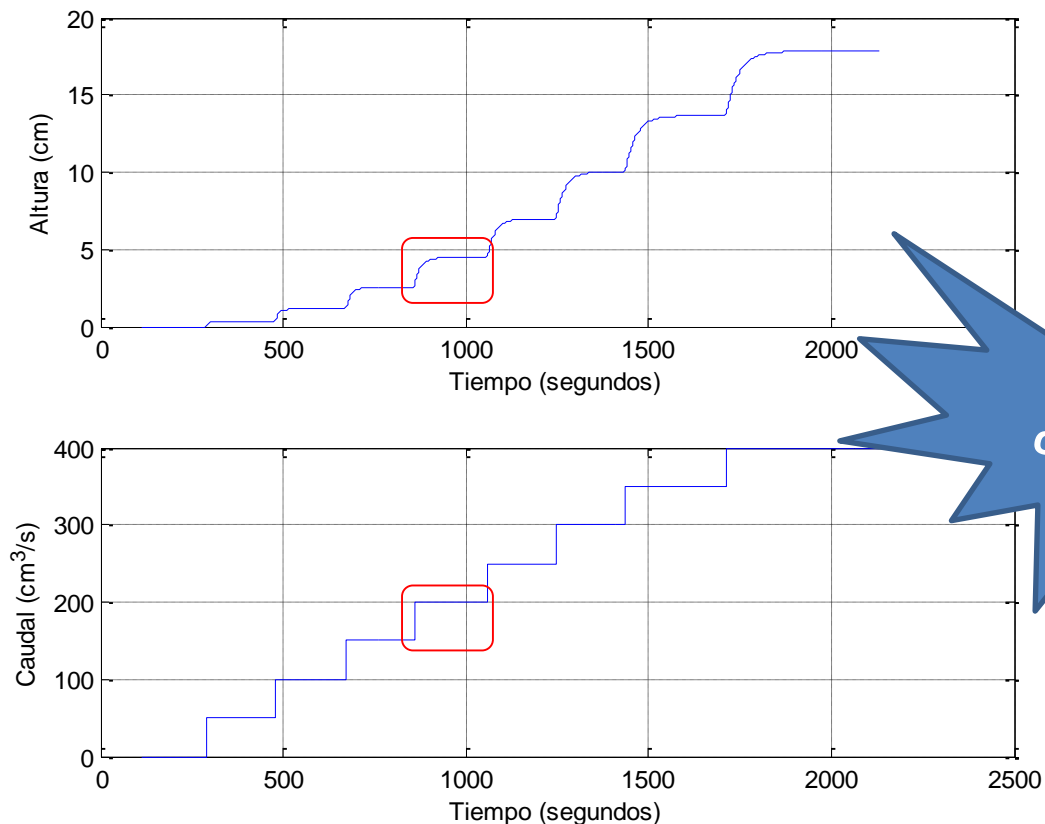
donde  $A$  es el área del tanque,  $a$  es la sección de descarga y  $g$  es la gravedad. Para el caso de la maqueta de 4 tanques que se va a utilizar en esta práctica, los parámetros toman los siguientes valores:  $A = 389.16 \text{ cm}^2$  y  $a = 2.1382 \text{ cm}^2$ .



**No linealidad del proceso**

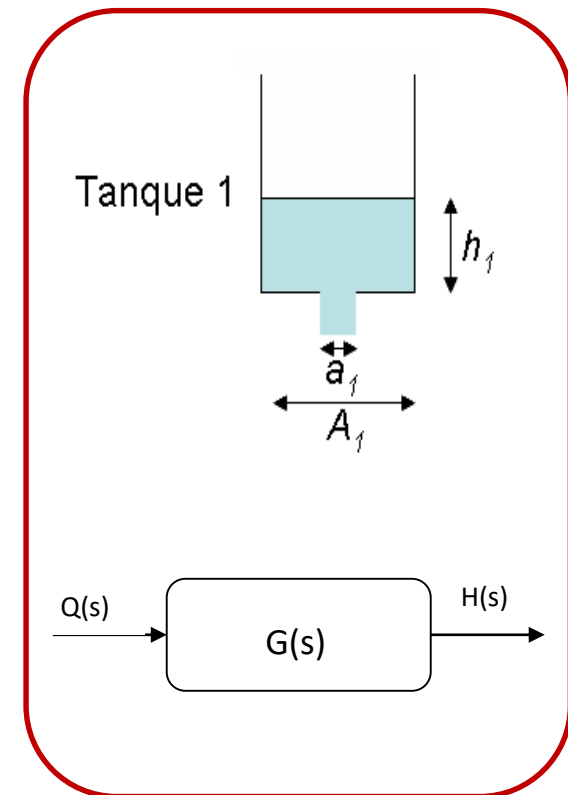
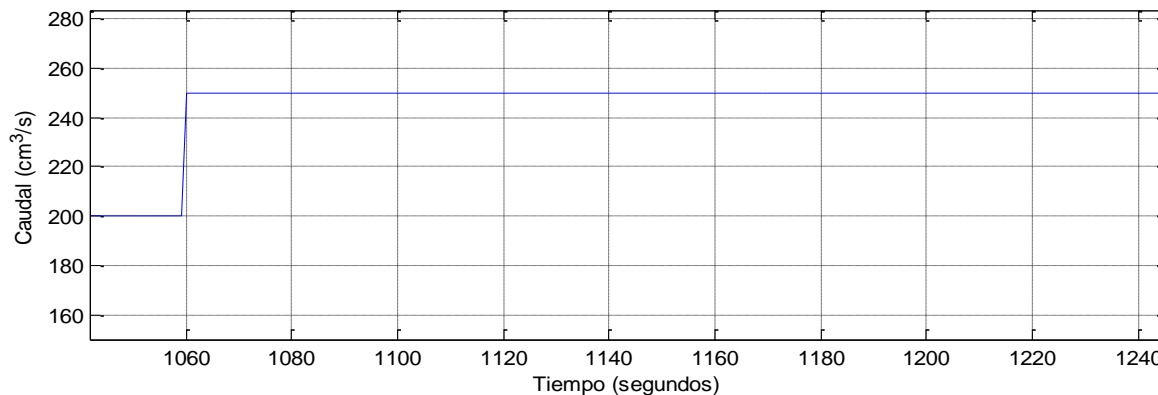
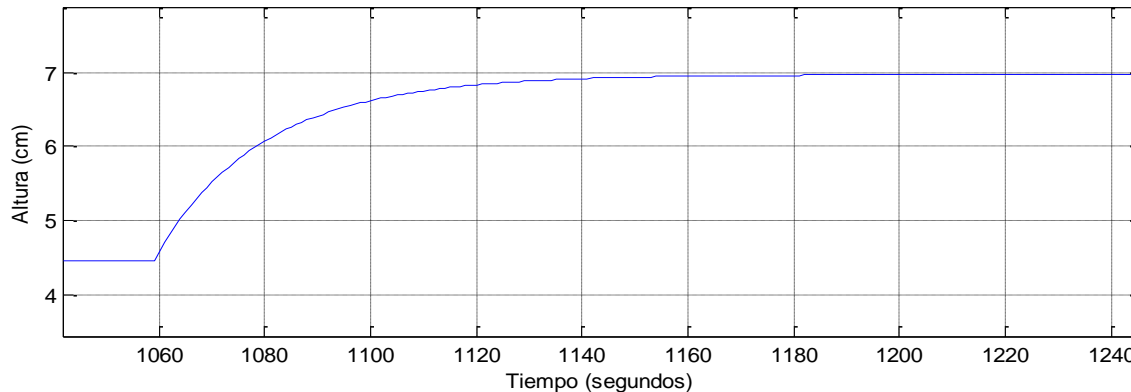
# Modelado de nivel de un único tanque

## Comportamiento no lineal del tanque



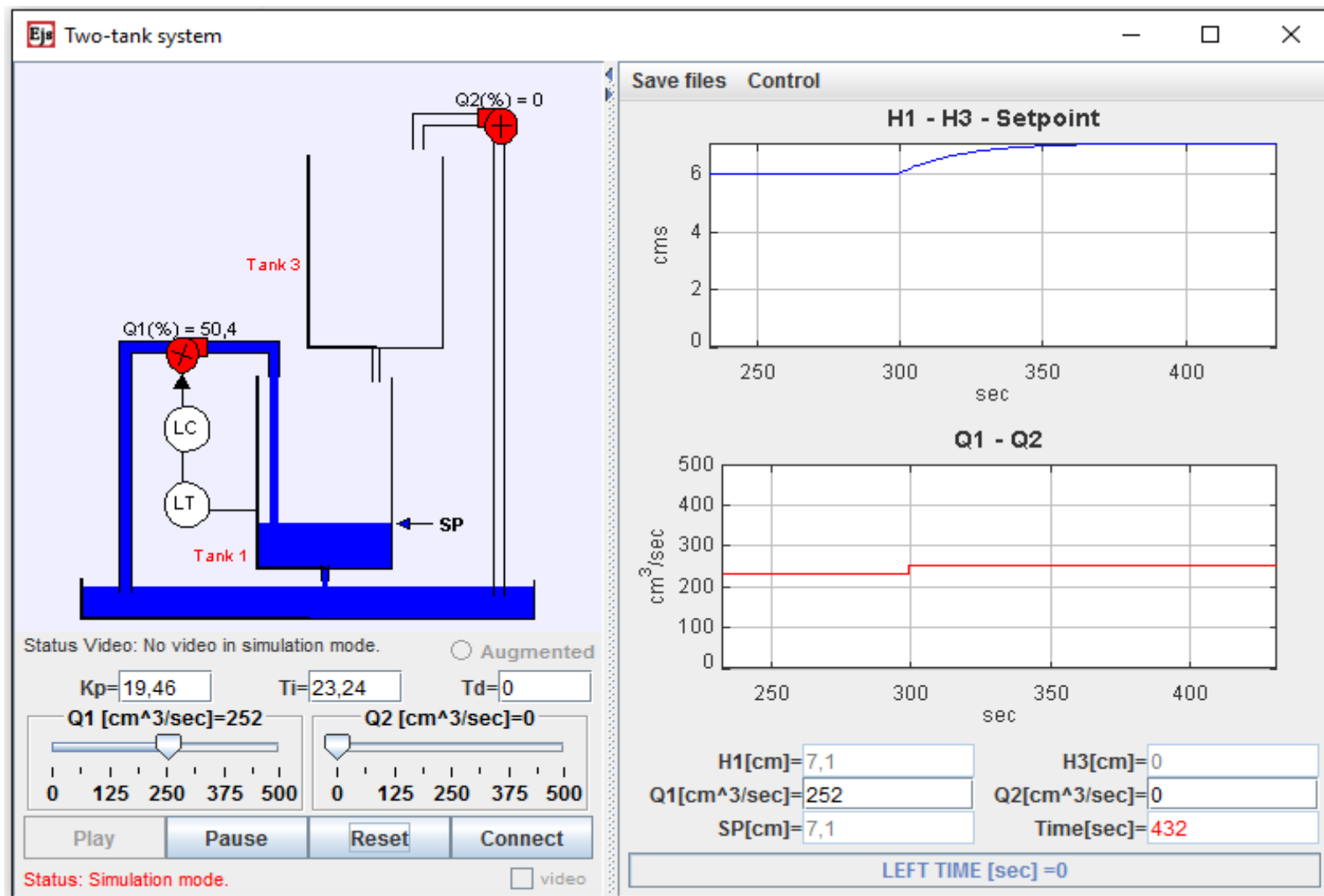
# Modelado de nivel de un único tanque

**Comportamiento lineal del tanque:**  
**Punto de operación  $h_0=4.5$  cm /  $q_0 = 200$  cm<sup>3</sup>/s**





# Simulador sistema dos tanques



# Práctica 1.

## Control de un proceso industrial



## 2. Introducción a Matlab y Simulink



# Introducción a Matlab y Simulink



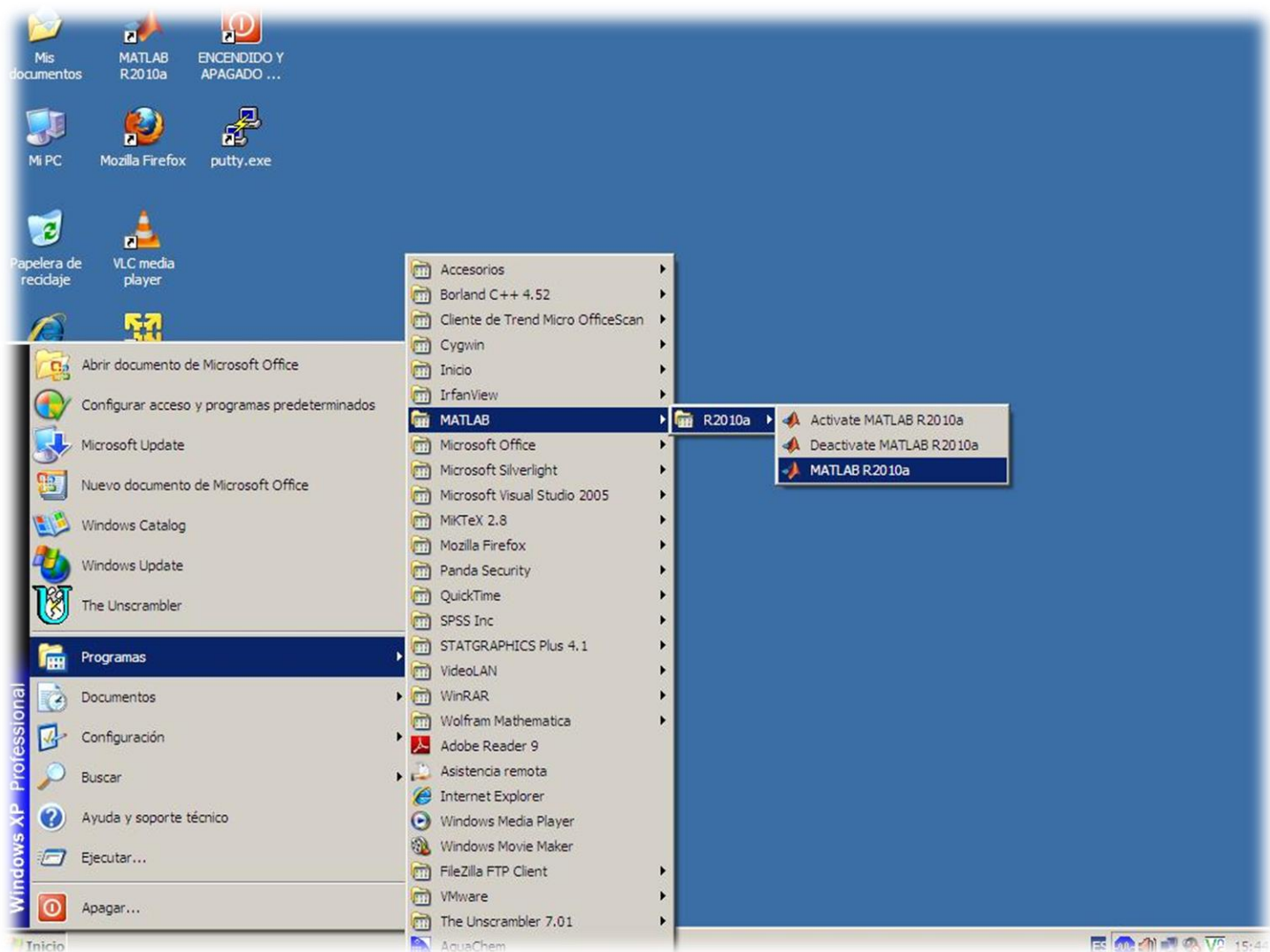
MATLAB es un lenguaje de computación técnica de alto nivel y un entorno interactivo para desarrollo de algoritmos, visualización de datos, análisis de datos y cálculo numérico.



- Existen paquetes específicos adicionales, toolboxes, de adquisición de datos, sistemas de control, procesamiento de imágenes, robótica, identificación de sistemas, redes neuronales, lógica borrosa, optimización, control robusto, control predictivo, etc.
- En el campo de la Ingeniería, MATLAB es una herramienta de gran utilidad para el análisis y representación de datos, realización de cálculos, simulación, etc. Cuenta con dos tipos básicos de funciones que se tratarán con posterioridad:
  - Las llamadas built-in functions. Son funciones que MATLAB tiene incorporadas internamente y por tanto sus códigos no son accesibles al usuario.
  - Funciones m-functions. Son funciones cuyo código es accesible y que están escritas a base de órdenes de con objeto de realizar una función determinada.

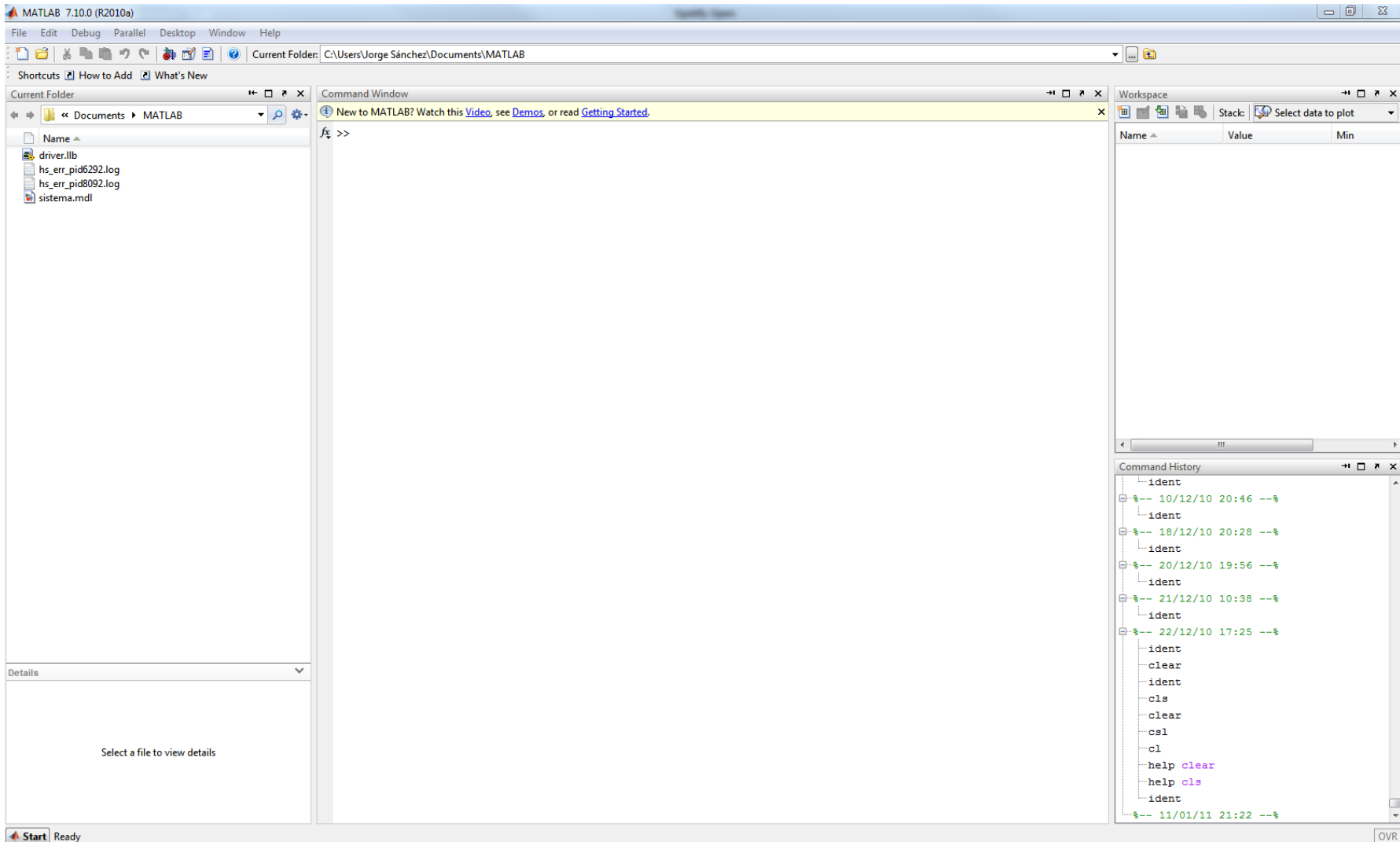


# Introducción a Matlab y Simulink





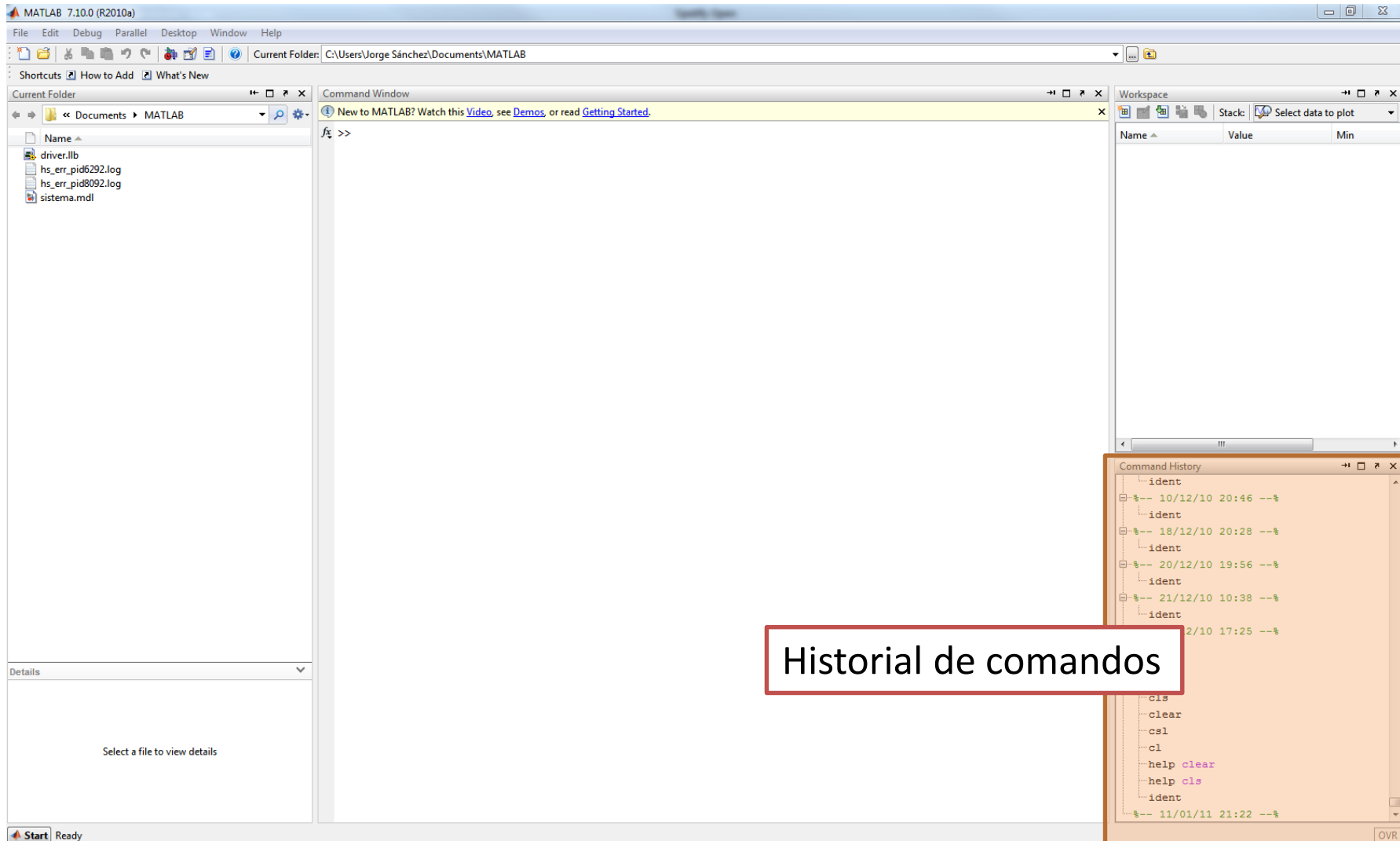
# Introducción a Matlab y Simulink





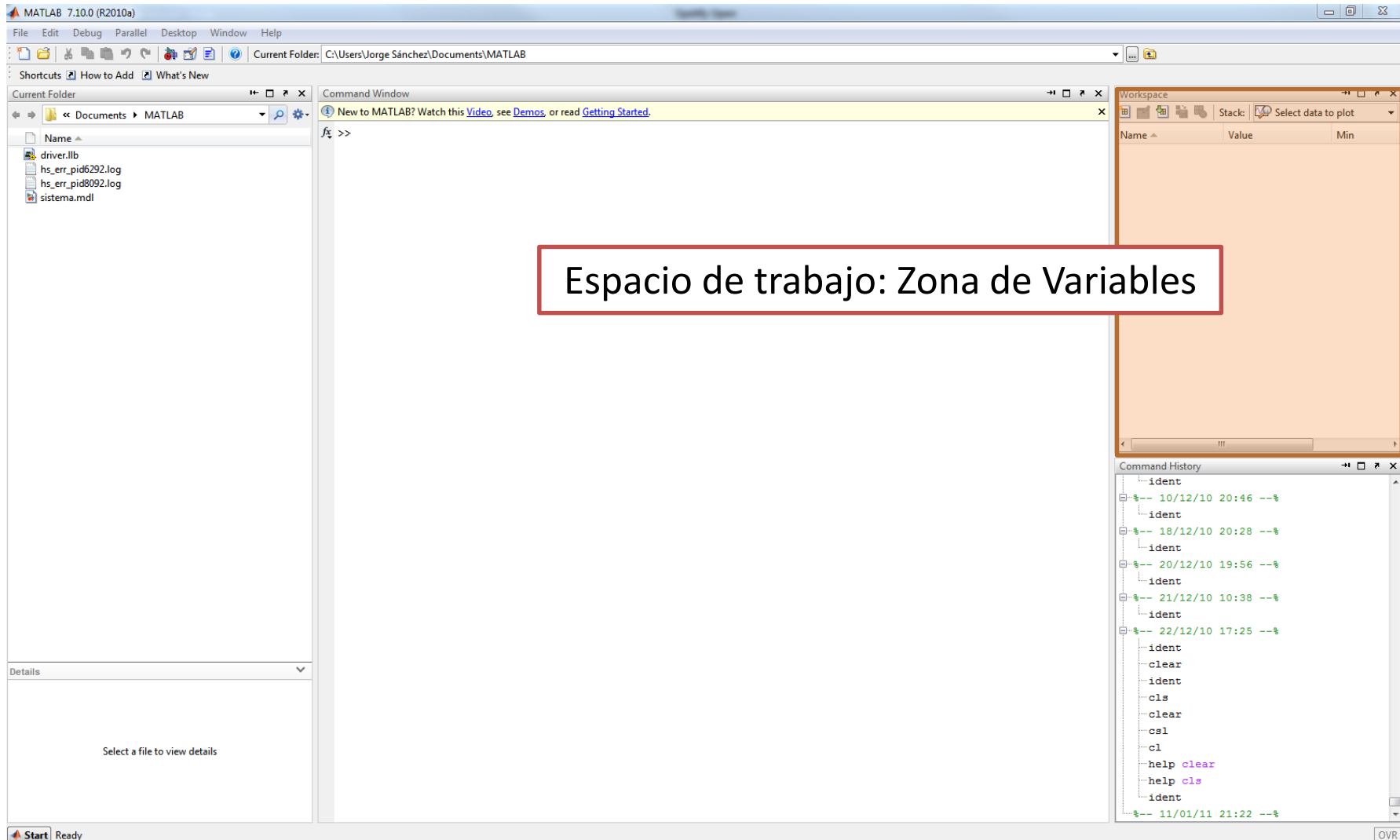


# Introducción a Matlab y Simulink





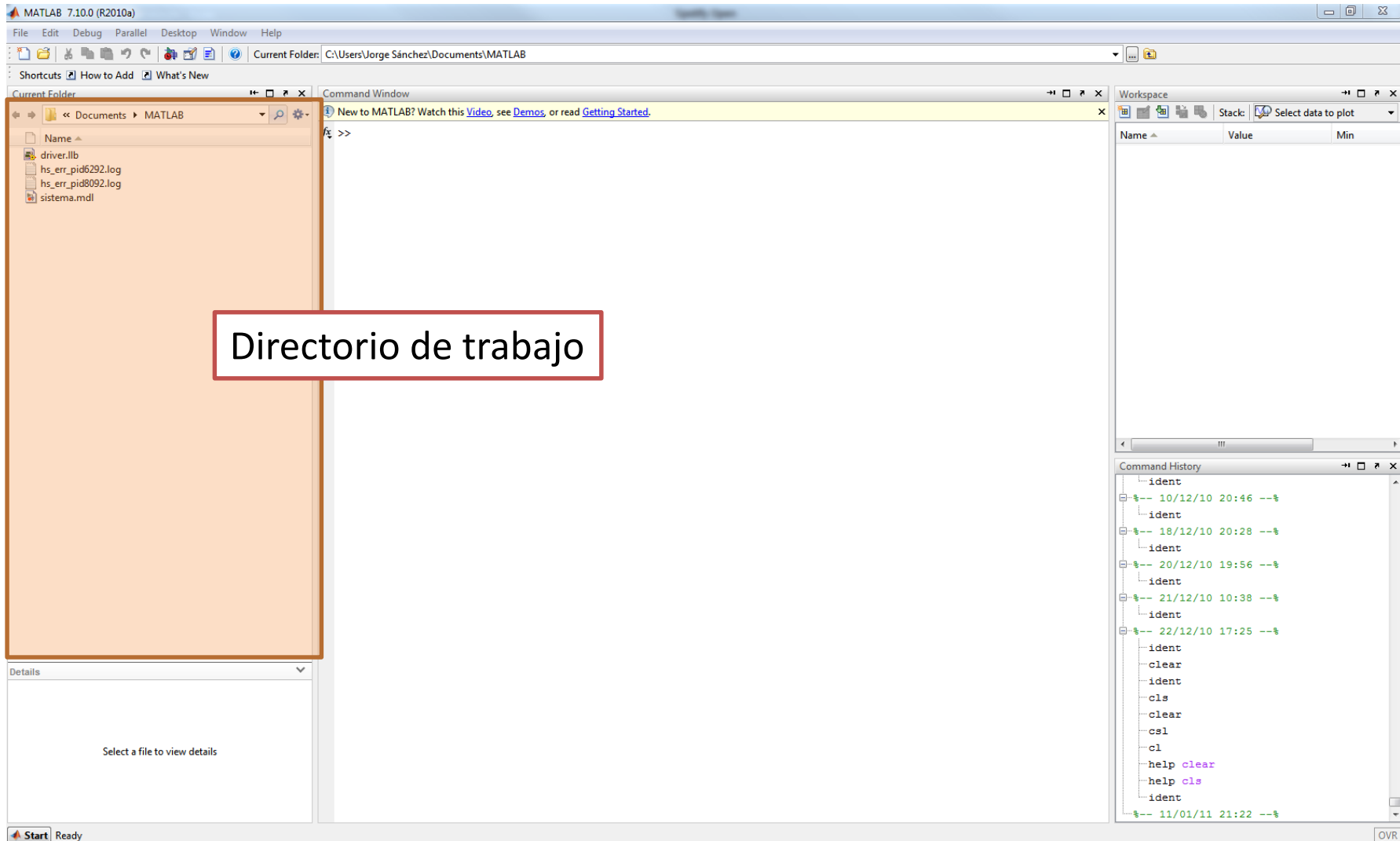
# Introducción a Matlab y Simulink





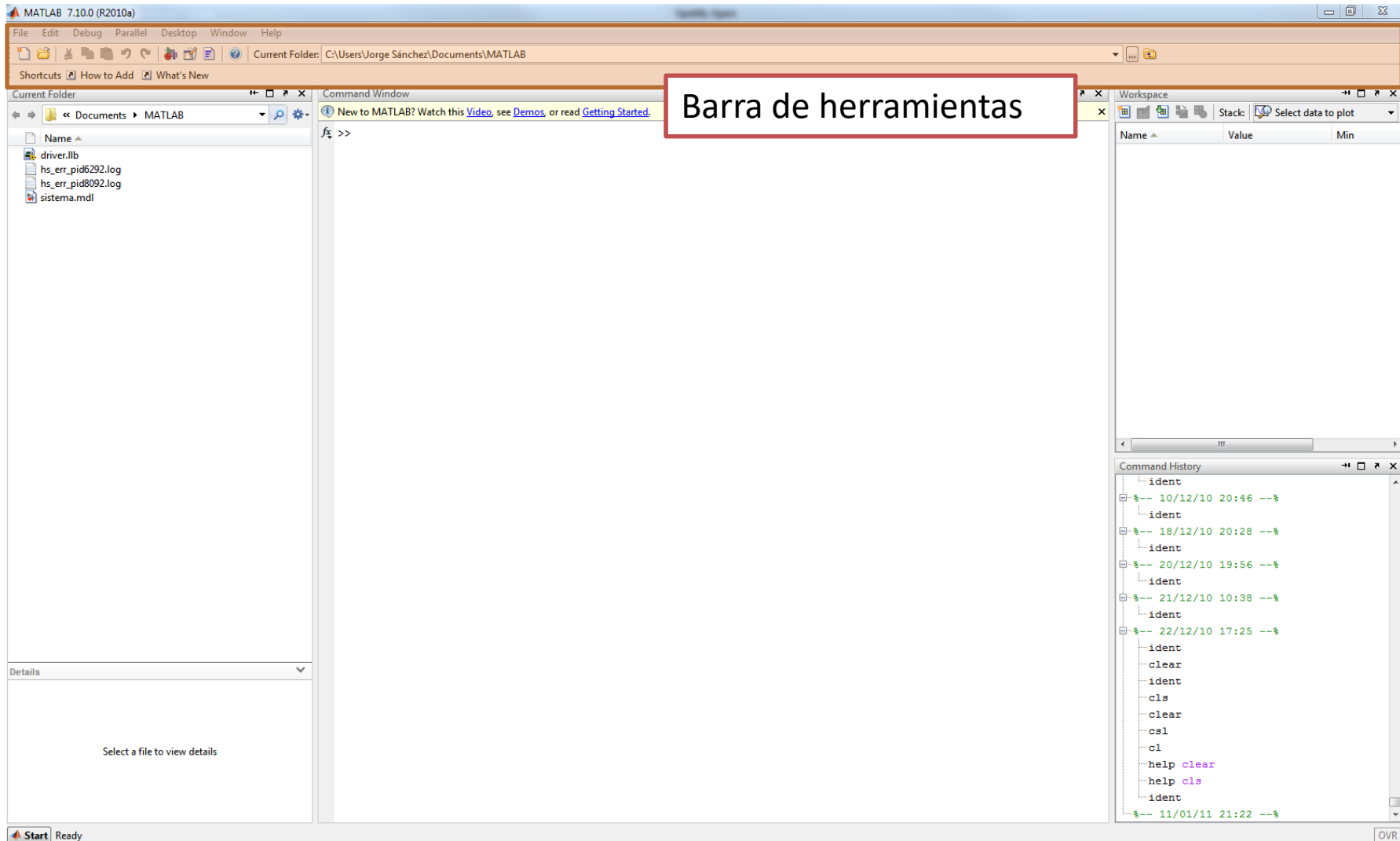


# Introducción a Matlab y Simulink





# Introducción a Matlab y Simulink





# Introducción a Matlab y Simulink



## Consideraciones prácticas

- Diferencia entre mayúsculas y minúsculas.
- No reconoce la ñ.
- Los decimales se separan con un punto.
- Acceso a la ayuda:
  - `help` + comando Ej: `help plot`
- Palabras reservadas
  - `ans`: respuesta sin asignación
  - `eps`: precisión mínima
  - `pi`: número pi
  - `inf`: Infinito
  - `NaN`: "Not a Number"
  - Etc.
- El elemento básico de Matlab son las matrices. Todos los tipos de datos se definen en función de vectores o matrices (desde números reales a complejos).



# Introducción a Matlab y Simulink

## Informática Industrial y Robótica

### Práctica 1. Control de un proceso industrial

#### Descripción de la práctica

Archivos adjuntos: Descripción.pdf (879,991 KB)



#### Memoria

Archivos adjuntos: Memoria Práctica 1.pdf (1,005 MB)



#### Tutorial simulador

Archivos adjuntos: Tutorial Lab Virtual.pdf (681,979 KB)



#### Simulador ej\_s\_twoTanks\_UAL

Archivos adjuntos: ej\_s\_twoTanks\_UAL.jar (1,19 MB)



#### Actividades propuestas

Archivos adjuntos: Actividades.docx (364,218 KB)  
Actividades.pdf (690,734 KB)



#### Extras



#### Análisis de datos utilizando Matlab

Archivos adjuntos: Análisis de datos utilizando Matlab.pdf (688,177 KB)



#### Diseño e implementación con Simulink

Archivos adjuntos: Diseño e implementación con Simulink.pdf (491,839 KB)



## Práctica 1

### ANÁLISIS DE DATOS UTILIZANDO MATLAB

#### PRÁCTICA 1. ANÁLISIS DE DATOS UTILIZANDO MATLAB

##### Justificación

En el modelado y control de sistemas, una de las principales tareas a realizar consiste en el análisis y validación de los datos que se toman del sistema, ya que si no se lleva a cabo esta labor, todas las pruebas que se realizan para construir el modelo y controlar el sistema pueden no servir cuando se realicen los ensayos en las instalaciones reales.

Como es la primera fase en el diseño del modelo y de los sistemas de control, se ha considerado conveniente tratarlo en la primera práctica. Además, como no es necesario haber impartido ninguno de los contenidos de teoría para poder llevarla a cabo, se plantea comenzar a la vez que el primer tema de teoría, lo que ayuda a coordinar los programas de práctica y teoría de la asignatura.

Por otra parte, como la herramienta que se va a utilizar a lo largo de las prácticas es *Matlab*, en esta primera se hará una introducción a la misma y al manejo de sus funciones fundamentales. Además, como en la mayoría de las prácticas es necesario diseñar e implementar programas bajo este entorno, se han incluido en esta práctica unos ejercicios donde se desarrollan distintos algoritmos útiles para la asignatura y que deberán ser implementados en el lenguaje que incorpora *Matlab*.

##### Objetivos

- Mostrar la necesidad del análisis de datos y la validación de los mismos en el proceso de diseño y construcción del modelado de sistemas dinámicos, así como en el desarrollo de sistemas de control.
- Exponer las ventajas de la utilización de herramientas informáticas para la realización de análisis de datos, utilizando *Matlab* como entorno de computación y desarrollo de aplicaciones totalmente integrado y orientado para llevar a cabo proyectos en donde se encuentren implicados cálculos matemáticos y la visualización gráfica de los mismos.
- Realizar distintos análisis de datos y representaciones gráficas de los mismos tomando como referente un sistema agrícola real, para poder tomar decisiones a partir de estos estudios.



# Introducción a Matlab y Simulink



*Simulink* es un programa orientado a la simulación de sistemas dinámicos.

La elaboración de modelos de sistemas es muy cómoda y fácil mediante la selección de iconos con el ratón y su conveniente conexionado. En su uso se pueden distinguir dos fases fundamentales:

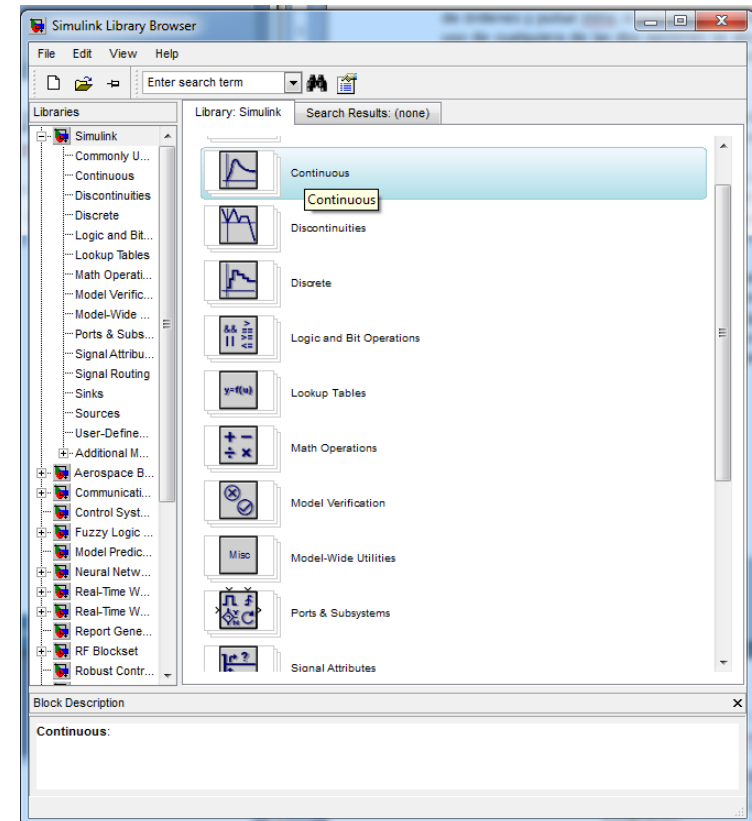
- Definición y construcción del modelo.
- Análisis y simulación del modelo.



- ¿Cómo se ejecuta Simulink?

En la línea de comandos escribir “*simulink*”

Con el siguiente icono:



# Introducción a Matlab y Simulink

## Informática Industrial y Robótica

### Práctica 1. Control de un proceso industrial

#### Descripción de la práctica

Archivos adjuntos: Descripción.pdf (879,991 KB)



#### Memoria

Archivos adjuntos: Memoria Práctica 1.pdf (1,005 MB)



#### Tutorial simulador

Archivos adjuntos: Tutorial Lab Virtual.pdf (681,979 KB)



#### Simulador ej\_s\_twoTanks\_UAL

Archivos adjuntos: ej\_s\_twoTanks\_UAL.jar (1,19 MB)



#### Actividades propuestas

Archivos adjuntos: Actividades.docx (364,218 KB)  
Actividades.pdf (690,734 KB)



#### Extras



### Práctica 1. Control de un proceso industrial

#### Análisis de datos utilizando Matlab

Archivos adjuntos: Análisis de datos utilizando Matlab.pdf (688,177 KB)



#### Diseño e implementación con Simulink

Archivos adjuntos: Diseño e implementación con Simulink.pdf (491,839 KB)



## Práctica 2

### DISEÑO E IMPLEMENTACIÓN DE MODELOS DE SISTEMAS DINÁMICOS Y SIMULACIÓN DE PROCESOS UTILIZANDO SIMULINK

#### PRÁCTICA 2. DISEÑO E IMPLEMENTACIÓN DE MODELOS DE SISTEMAS DINÁMICOS Y SIMULACIÓN DE PROCESOS UTILIZANDO SIMULINK

##### Justificación

A la hora de diseñar un sistema de control para un determinado proceso es necesario conocer el comportamiento dinámico del mismo. Para ello hay que obtener un modelo del mismo que relacione las dinámicas existentes entre las variables a controlar y las variables de entrada (tanto perturbaciones como variables de control).

Con los términos *modelado* y *simulación* se designa al conjunto de actividades asociadas con la construcción de modelos de sistemas del mundo real y su ejecución en un computador. La *simulación* digital relaciona modelos y computadores, refiriéndose al proceso de imitación de aspectos importantes del comportamiento del sistema mediante el diseño, construcción y experimentación con el modelo del sistema. Este proceso es similar al de experimentación que llevan a cabo los científicos en un laboratorio con el que pretenden aumentar la comprensión de alguna teoría para su validación y empleo posterior, y esto es precisamente lo que van a realizar los alumnos en esta práctica (cuando comienza ya se ha impartido parte del tema 2 dedicado al modelado y análisis de sistemas dinámicos, por lo que el alumno ya ha adquirido los conceptos básicos y necesarios).

El objetivo principal de la misma es que el alumno sea capaz de diseñar e implementar modelos de determinados procesos o sistemas, realizando distintas experiencias con ellos para analizarlos y obtener sus propias conclusiones.

##### Objetivos

- Mostrar las ventajas de utilizar herramientas informáticas para la simulación de sistemas dinámicos, concretamente *Simulink* basada en la representación en el dominio de *Laplace* que

# Práctica 1. Control de un proceso industrial: Nivel de tanques



## 4. Implementación de modelos utilizando Simulink



# Modelos utilizando Simulink



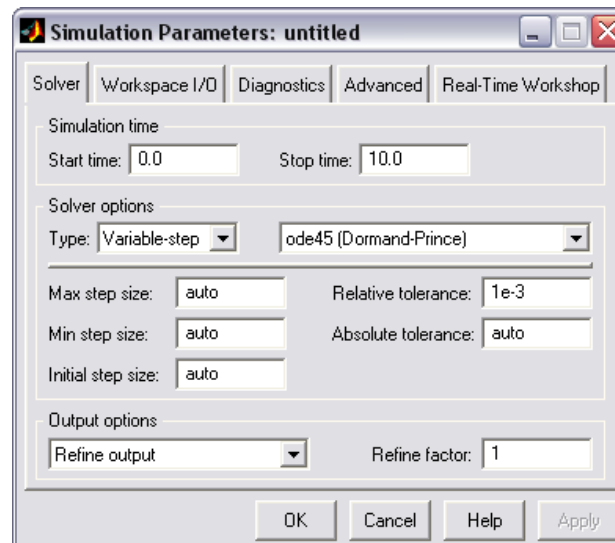
*Simulink* es un programa orientado a la simulación de sistemas dinámicos.

- Creación de un nuevo archivo

**menú File -> New -> Model**

- Forma de trabajar con los bloques
- Simulación del modelo: configuración y ejecución

**menú Simulation -> Simulation parameters**







# Modelos utilizando Simulink



## Bibliotecas de bloques de Simulink

- Señales de entrada (sources)**

**Constant.** Introduce un valor constante como entrada (independiente del tiempo), que se introduce como parámetro.



Constant

**Ramp.** Genera una señal en forma de rampa (aumenta a velocidad constante).



Ramp

**Step.** Genera una función escalón entre dos valores definidos en un momento determinado. Parámetros: el tiempo en el que comienza el escalón, y los valores inicial y final de la amplitud.



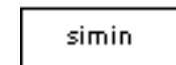
Step

**Clock.** Muestra la hora del sistema en una ventana. Puede servir también como bloque de entrada para otro bloque.



Clock

**From Workspace.** Lee datos de entrada desde una matriz. Formato y requerimientos similares a los del bloque anterior, pero cambiando filas por columnas, es decir, en este caso el vector temporal corresponde con la primera columna. Parámetro: nombre de la matriz.



From Workspace



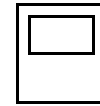
# Modelos utilizando Simulink



## Bibliotecas de bloques de Simulink

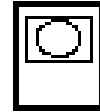
- Señales de salida (sinks)**

**Scope.** Muestra las señales durante la simulación (señales de salida del bloque al que esté conectado). Permite varias señales de entrada, que se dibujarán simultáneamente.



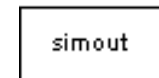
Scope

**XY Graph.** Gráfico XY donde la primera entrada se toma como eje X y las siguientes como eje Y.



XY Graph

**To Workspace.** Escribe los datos de simulación en una matriz. Cuando se seleccione esta opción en el bloque la matriz, cuyo nombre aparecerá en la caja, tendrá el mismo número de columnas que el vector de datos de entrada al bloque (cada canal de datos se escribirá en una columna). Por tanto, cada paso de simulación escribe una fila en la matriz conteniendo el estado del vector de entradas al bloque. Parámetros: el nombre de la variable (matriz) donde se va a guardar el resultado de la simulación y el máximo número de filas (periodos de muestreo) permitidos.



To Workspace



# Modelos utilizando Simulink



## Bibliotecas de bloques de Simulink

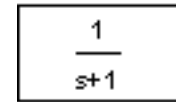
- Señales lineales continuos (Continuous)**

**Integrator.** Integra una señal. Parámetros: valor inicial del integrador (constante o variable).



Integrator

**Transfer Fcn.** Función de transferencia. Hay que especificar los vectores del numerador y denominador, que se tratan en potencias descendentes de  $s$  (como en MATLAB).



Transfer Fcn

**Derivative.** Bloque que tiene como salida la derivada respecto al tiempo de la señal de entrada. La precisión de los resultados depende del tamaño de los intervalos de tiempo tomados en la simulación. Por ello es importante fijar en los parámetros de simulación el maximum step size a un valor pequeño.



Derivative

**Transport Delay.** Retarda la señal de entrada un tiempo determinado. Parámetros: el retardo puro (Time delay) y la entrada inicial (Initial input), que será la salida del bloque hasta que la simulación supere el retardo puro.



Transport Delay



# Modelos utilizando Simulink



## Bibliotecas de bloques de Simulink

- No linealidades (Discontinuous)**

**Saturation.** Saturación de la señal de entrada. Parámetros: Lower output limit , límite más bajo de la salida; Upper output limit, límite más alto de la salida.



Saturation

**Dead zone.** Región de salida nula (zona muerta). Parámetros: comienzo y fin de la zona muerta.



Dead Zone

**Relay.** Satura la señal de salida entre dos valores.



Relay

- Tratamiento de señales (Signal Routing)**

**Mux.** Permite combinar una serie de señales de entrada, obteniéndose como resultado un único vector de salida.



Mux

**Demux.** Es el bloque opuesto al anterior. Su función es separar un conjunto de señales que han sido combinadas previamente con un bloque Mux.



Demux



# Modelos utilizando Simulink



## Bibliotecas de bloques de Simulink

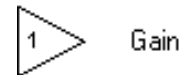
- Operaciones matemáticas (Math Operations)**

**Abs.** Bloque que tiene como salida el valor absoluto de la entrada.



Abs

**Gain.** Multiplica la entrada por una constante, que se introduce como parámetro en forma de escalar o de variable. En caso que se introduzca una variable, si se quiere ver su contenido hay que introducirla entre paréntesis.



Gain

**Sum.** Suma o resta varias salidas de otros bloques. Parámetro: lista de signos que se aplican a las entradas. Esta lista contiene un conjunto de signos + y – seguidos (sin dejar espacio en blanco). El número de signos introducidos determina la cantidad de puertos de entrada que mostrará el bloque.



Sum

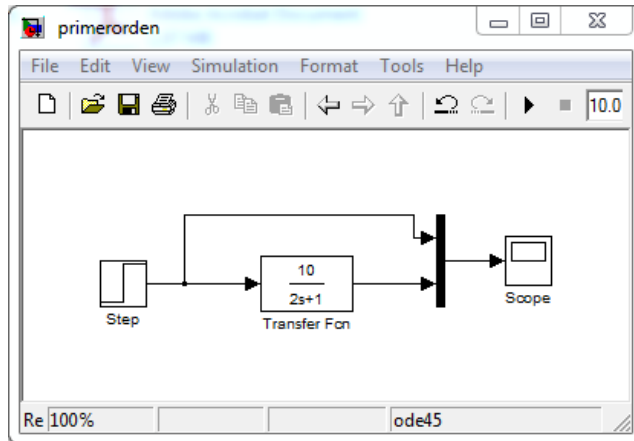
**Trigonometric Function.** Aplica una determinada función trigonométrica a la señal de entrada. La función trigonométrica a aplicar se selecciona dentro del bloque sobre el parámetro Function.



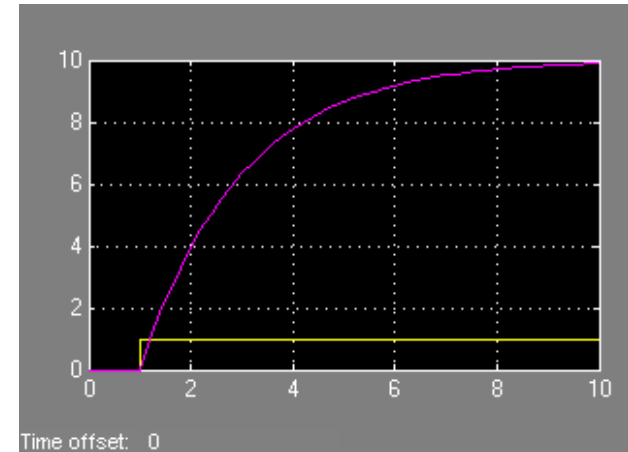
Trigonometric Function

# Modelos utilizando Simulink

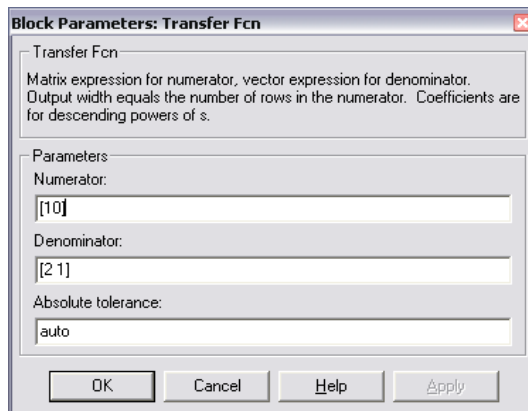
## Ejemplo de uso: Sistema de primer orden



(a) Diagrama de bloques



(b) Respuesta del sistema



Block Parameters: Transfer Fcn

Transfer Fcn

Matrix expression for numerator, vector expression for denominator. Output width equals the number of rows in the numerator. Coefficients are for descending powers of s.

Parameters

Numerator:

[10]

Denominator:

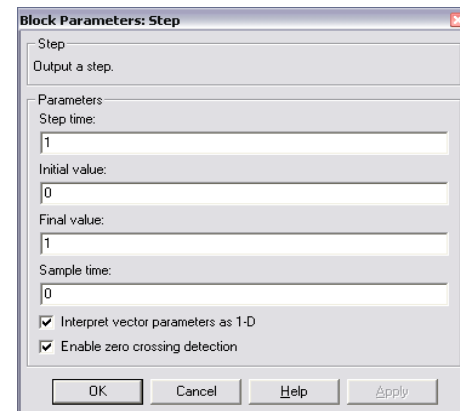
[2 1]

Absolute tolerance:

auto

OK Cancel Help Apply

(c) Valores para el bloque Transfer Fcn



Block Parameters: Step

Step

Output a step.

Parameters

Step time:

1

Initial value:

0

Final value:

1

Sample time:

0

☒ Interpret vector parameters as 1-D

☒ Enable zero crossing detection

OK Cancel Help Apply

(d) Valores para el bloque Step

# Práctica 1.

## Control de un proceso industrial

---



### Informática Industrial y Robótica

Área de Ingeniería de Sistemas y Automática  
Departamento de Informática  
Universidad de Almería

