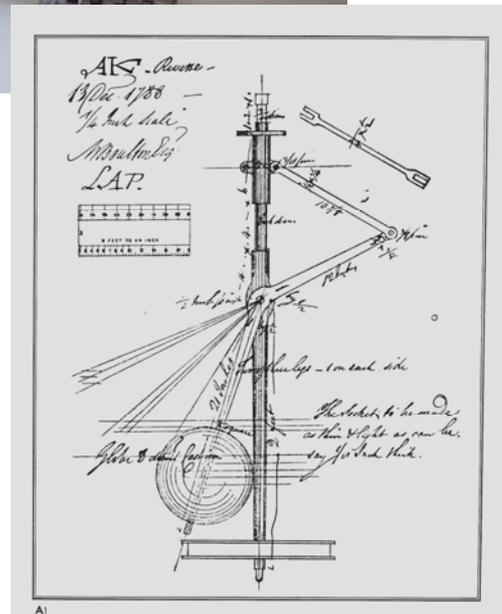
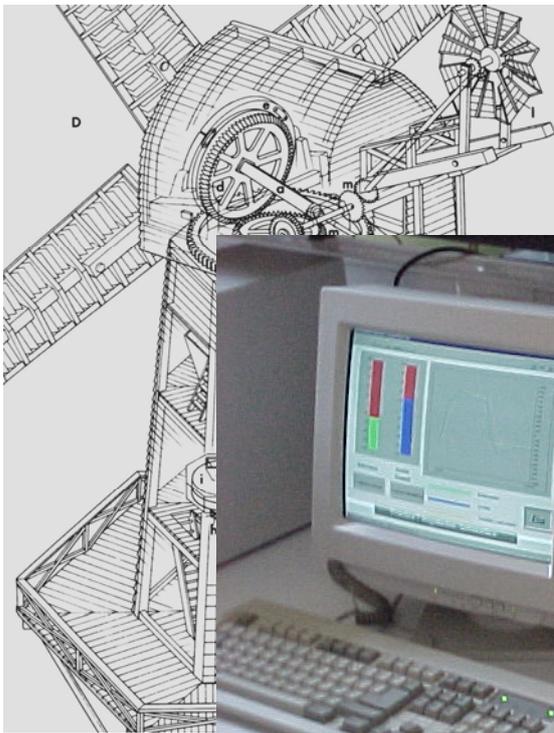


# PRÁCTICAS DE REGULACIÓN AUTOMÁTICA



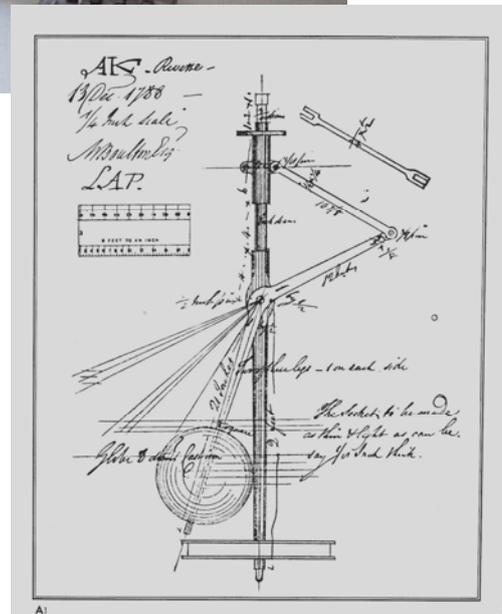
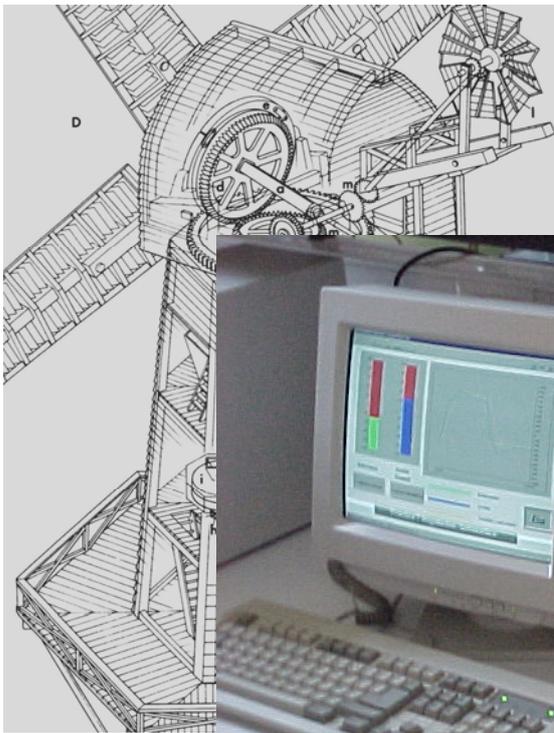
RAMÓN  
PIEDRAFITA  
MORENO

INGENIERÍA DE SISTEMAS  
Y AUTOMÁTICA

ESPECIALIDAD DE ELECTRICIDAD



# PRÁCTICAS DE REGULACIÓN AUTOMÁTICA



RAMÓN  
PIEDRAFITA  
MORENO

INGENIERÍA DE SISTEMAS  
Y AUTOMÁTICA

ESPECIALIDAD DE ELECTRICIDAD

© Ramón Piedrafita Moreno  
1ª Edición: Septiembre de 2001

ISBN 84-88502-81-8  
Depósito Legal Z-3243-99

Imprime Kronos  
Menendez Pelayo,4  
50009 ZARAGOZA

# ***Regulación Automática***

## ***NORMATIVA DE PRACTICAS***

### ***INGENIERO TECNICO EN ELECTRICIDAD***

*Se realizan cinco prácticas a lo largo del curso.*

- *2 prácticas de simulación de sistemas*
- *1 práctica de control de sistemas mediante reguladores analógicos*
- *prácticas de control por computador.*

*Las prácticas constan de:*

- *Estudio Teórico. (se recomienda su realización previa a la sesión de laboratorio)*
- *Realización Práctica.*

*Las prácticas son una parte fundamental de la asignatura por lo que se recomienda se realicen en paralelo con la asistencia a clase.*

*Previa asistencia a las prácticas la evaluación de estas se hará mediante preguntas integradas en los exámenes de la asignatura, las cuales supondrán el 25% de la calificación*

<i>Transformada de Laplace</i> $E(s)$	<i>Función en el tiempo</i> $e(t)$
$1$	$\delta(t)$
$e^{-Ts}$	$\delta(t-T)$
$\frac{1}{s}$	$u(t)$
$\frac{1}{s^2}$	$t$
$\frac{1}{s+a}$	$e^{-at}$
$\frac{1}{(s+a)^2}$	$t e^{-at}$
$\frac{1}{(s+a)^n}$	$\frac{1}{(n-1)!} t^{n-1} e^{-at}$
$\frac{1}{(s+a)(s+b)}$	$\frac{1}{b-a} (e^{-at} - e^{-bt})$
$\frac{s}{(s+a)(s+b)}$	$\frac{1}{a-b} (ae^{-at} - be^{-bt})$
$\frac{s+z}{(s+a)(s+b)}$	$\frac{1}{b-a} ((z-a)e^{-at} - (z-b)e^{-bt})$
$\frac{a}{s^2(s+a)}$	$t - \frac{1-e^{-at}}{a}$
$\frac{a}{s^2+a^2}$	$\sin(at)$
$\frac{s}{s^2+a^2}$	$\cos(at)$
$\frac{1}{(s+a)^2+b^2}$	$\frac{1}{b} e^{-at} \sin(bt)$
$\frac{s+a}{(s+a)^2+b^2}$	$e^{-at} \cos(bt)$
$\frac{1}{s(s+a)}$	$\frac{1}{a} (1 - e^{-at})$
$\frac{1}{s}$	$u(t)$ ó $1$ <i>escalón unitario</i>
$\frac{1}{s} e^{-Ts}$	$u(t-T)$ <i>escalón unitario retardado</i> $T$ segundos
$\frac{1}{s} (1 - e^{-Ts})$	$u(t) - u(t-T)$ <i>pulso rectangular</i>
$\frac{1}{s^2}$	$t$ <i>rampa unitaria</i>
$\frac{1}{s^n}$	$\frac{t^{n-1}}{(n-1)!}$
$\frac{1}{s^2 + 2\xi\omega_n s + \omega_n^2}$	$\frac{1}{\omega_d} e^{-\xi\omega_n t} \sin(\omega_d t)$ $\omega_d = \omega_n \sqrt{1-\xi^2}$

$\frac{1}{s(s^2 + 2\xi\omega_n s + \omega_n^2)}$	$\frac{1}{\omega_n^2} - \frac{1}{\omega_n \omega_d} e^{-\xi\omega_n t} \text{sen}(\omega_d t + \theta)$ $\omega_d = \omega_n \sqrt{1 - \xi^2} \quad \theta = \arccos \xi$
--	---

*Tabla de Transformadas de Laplace*

Transformada de Laplace E(s)	Función en el tiempo e(t)	Secuencia numérica e(kT) o e(k)	Transformada z E(z)
$\frac{1}{s}$	$u(t)$	$1(k)$	$\frac{z}{z-1}$
$\frac{1}{s^2}$	$t$	$kT$	$\frac{Tz}{(z-1)^2}$
$\frac{1}{s+a}$	$e^{-at}$	$e^{-akT}$	$\frac{z}{z-e^{-aT}}$
$\frac{a}{s(s+a)}$	$1-e^{-at}$	$1-e^{-akT}$	$\frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}$
$\frac{1}{(s+a)^2}$	$t e^{-at}$	$kT e^{-akT}$	$\frac{Tz e^{-aT}}{(z-e^{-aT})^2}$
$\frac{a}{s^2(s+a)}$	$t - \frac{1-e^{-at}}{a}$	$kT - \frac{1-e^{-akT}}{a}$	$\frac{Tz}{(z-1)^2} - \frac{z(1-e^{-aT})}{a(z-1)(z-e^{-aT})}$
$\frac{a}{s^2+a^2}$	$\sin(at)$	$\sin(akT)$	$\frac{z \operatorname{sen}(aT)}{z^2 - 2z \cos(aT) + 1}$
$\frac{s}{s^2+a^2}$	$\cos(at)$	$\cos(akT)$	$\frac{z(z - \cos(aT))}{z^2 - 2z \cos(aT) + 1}$
$\frac{1}{(s+a)^2 + b^2}$	$\frac{1}{b} e^{-at} \sin(bt)$	$\frac{1}{b} e^{-akT} \sin(bkT)$	$\frac{1}{b} \left[ \frac{z e^{-aT} \operatorname{sen} bT}{z^2 - 2e^{-aT} z \cos bT + e^{-2aT}} \right]$
$\frac{s+a}{(s+a)^2 + b^2}$	$e^{-at} \cos(bt)$	$e^{-akT} \cos(bkT)$	$\frac{z^2 - e^{-aT} z \cos bT}{z^2 - 2e^{-aT} z \cos bT + e^{-2aT}}$
		$a^k$	$\frac{1}{1 - az^{-1}}$
		$a^{k-1}$	$\frac{z^{-1}}{1 - az^{-1}}$
		$k = 1, 2, 3, \dots$	$\frac{1 - az^{-1}}{z^{-1}}$
		$ka^{k-1}$	$\frac{z^{-1}}{(1 - az^{-1})^2}$
		$k^2 a^{k-1}$	$\frac{z^{-1}(1 + az^{-1})}{(1 - az^{-1})^3}$
		$k^3 a^{k-1}$	$\frac{z^{-1}(1 + 4az^{-1} + a^2 z^{-2})}{(1 - az^{-1})^4}$
		$k^4 a^{k-1}$	$\frac{z^{-1}(1 + 11az^{-1} + 11a^2 z^{-2} + a^3 z^{-3})}{(1 - az^{-1})^5}$
		$a^k \cos k\pi$	$\frac{1}{1 + az^{-1}}$

Transformadas z más comunes.

---

# *Práctica 1*

## *Introducción a la simulación de Sistemas mediante Matlab.*

---

### **OBJETIVO**

En esta práctica se pretende que el alumno tome contacto con una herramienta informática de ayuda a la simulación y análisis de sistemas. Esta herramienta es el programa "MATLAB" que permite simular el comportamiento dinámico de sistemas, sirviendo de base estas simulaciones para el análisis de su comportamiento.

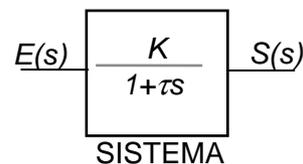
### **PROGRAMA EJEMPLO**

```
PROGRAMA MATLAB
%****Respuesta al impulso de un sistema de primer orden****
%*****
%****Se introduce el tiempo final de simulación*****
%****Se introduce la ganancia y la constante de tiempo****
%*****
tfinal=input ('tiempo final de simulación');
K=input ('Ganancia sistema Primer Orden');
T=input ('constante de tiempo');
num=[K 0];
den=[T 1];
```

```
periodo=ffinal/1000;  
t=0:periodo:ffinal;  
u=ones(length(t),1);  
[yc,xc]=lsim(num,den,u,t);  
plot(t,yc)
```

## Estudio de un sistema de primer orden

Considerese el sistema de la figura expresado mediante su función de transferencia:



Introducir dicho sistema dentro de Matlab definiendo el numerador y el denominador del sistema como variables (por ejemplo num y den).

Obtener las siguientes gráficas y detallar el proceso seguido (comandos ejecutados) para su obtención:

- 1.- Respuesta del sistema ante una entrada en impulso.
- 2.- Respuesta del sistema ante un escalón unitario. Indicar además en esta gráfica la ganancia, cte de tiempo y tiempo de respuesta del sistema.
- 3.- Respuesta ante una entrada en rampa de pendiente 2.

PROGRAMA MATLAB

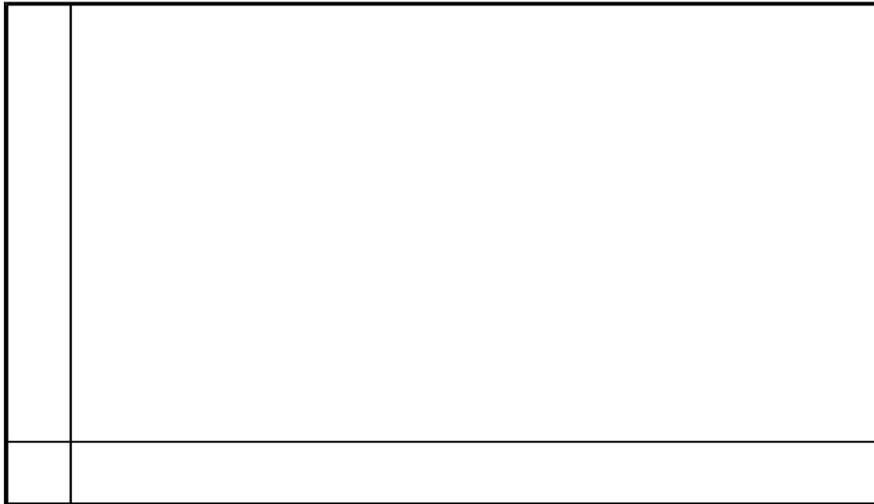
```
%****Respuesta ante un escalón unitario  
%
```

%\*\*\*\*Respuesta ante una entrada en impulso

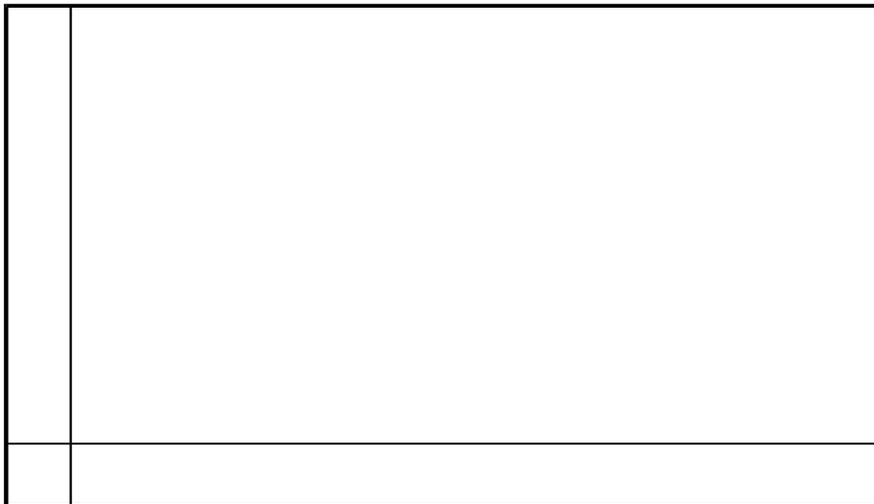
%\*\*\*\*Respuesta ante una rampa de pendiente 2



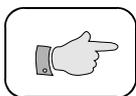
*Gráfica 1. Respuesta del sistema ante una entrada en Impulso.*



Gráfica 2. Respuesta del sistema ante una entrada en escalón unitario.

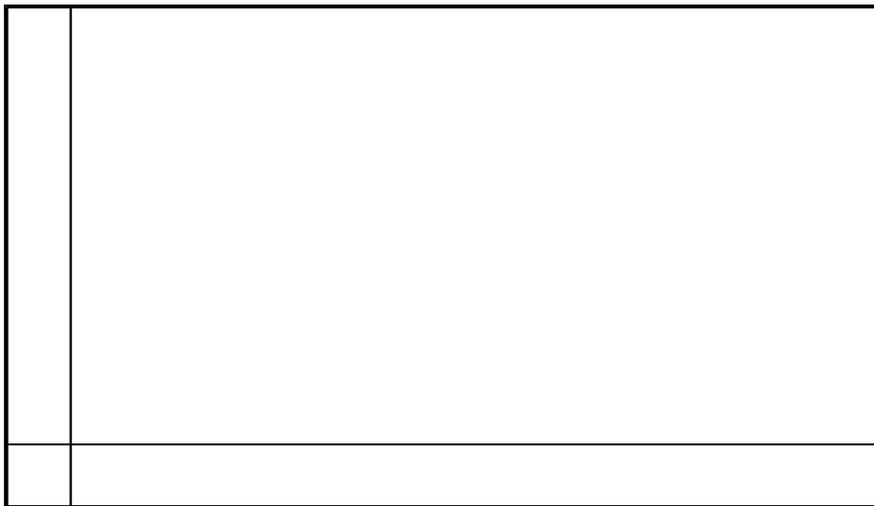


Gráfica 3. Respuesta del sistema ante una entrada en rampa de pendiente 2.



Para el mismo ejemplo anterior comentar los resultados obtenidos variando los valores de  $K$  y  $\tau$ .

%\*\*\*\*Resultados obtenidos para diferentes valores de  $K$  y  $\tau$



*Gráfica 4. Gráficas para diferentes valores de  $K$  y  $\tau$*

## **ESTUDIO DE UN SISTEMA DE SEGUNDO ORDEN**

Dado el siguiente sistema de segundo orden

$$\frac{C(s)}{R(s)} = \frac{1}{s^2 + 0.5s + 4}$$

Responder a las siguientes cuestiones:

- 1.- Considerese la respuesta a un escalón unitario y dibujense la gráfica correspondiente
- 2.- Obtengase la respuesta del sistema ante una rampa de pendiente 3.
- 3.- Obtener los diagramas de Bode utilizando la función semilogarítmica.
- 4.- Obtener además el correspondiente diagrama de Nyquist.

```
PROGRAMA MATLAB
```

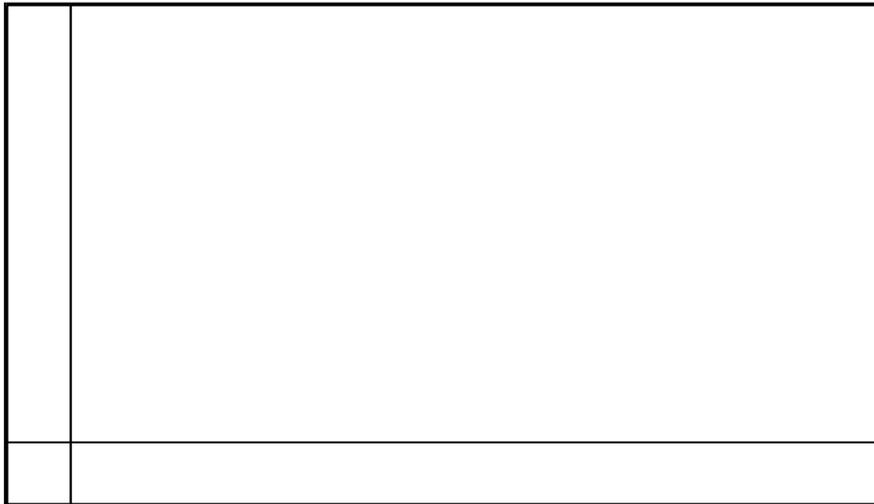
```
%****Respuesta a un escalón unitario de un sistema de segundo orden
```

```
%
```

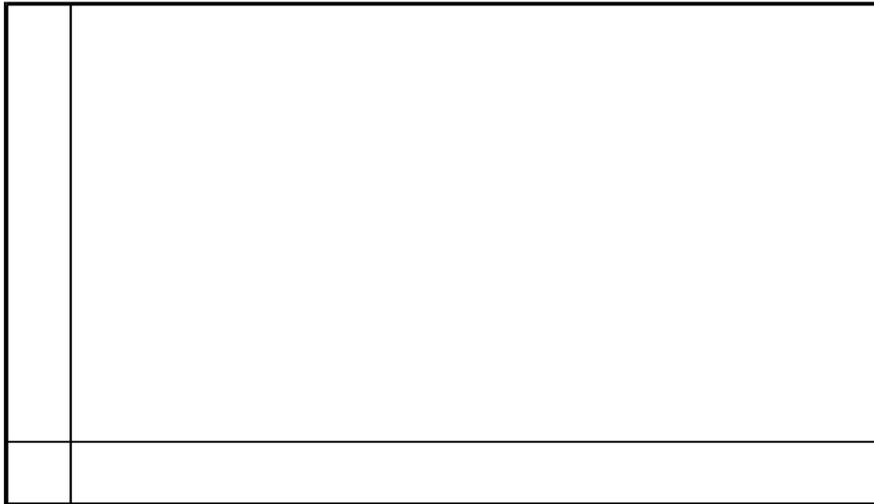
```
%****Respuesta a una rampa de pendiente 3
```

```
%****Diagrama de Bode para sistema de segundo orden
```

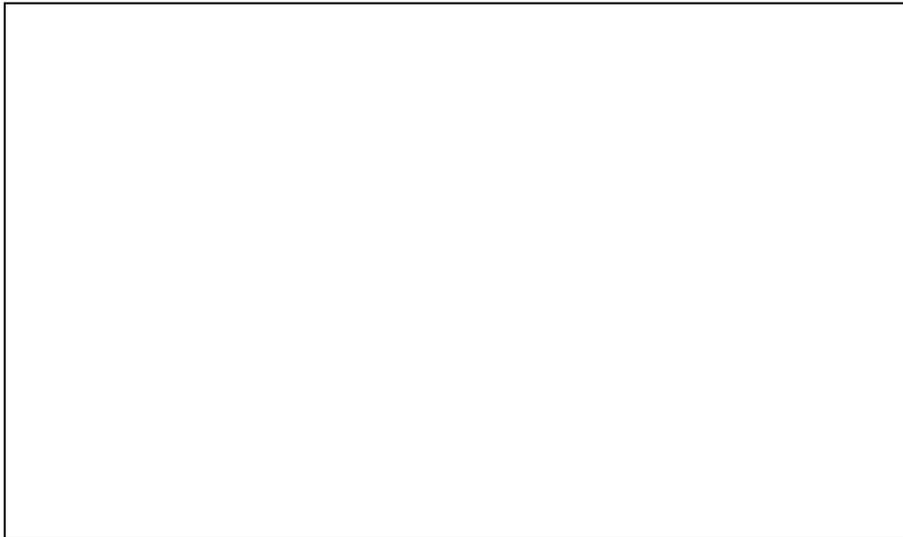
%\*\*\*\*Diagrama de Nyquist para un sistema de segundo orden



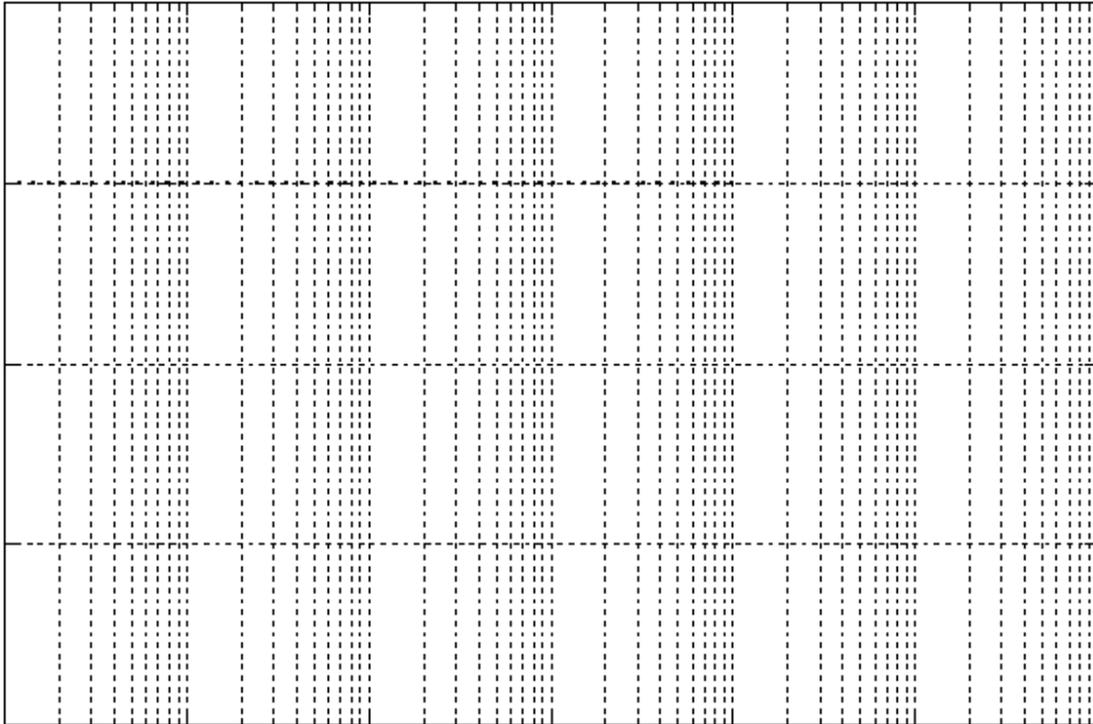
*Gráfica 5. Respuesta a un escalón unitario en segundo orden.*



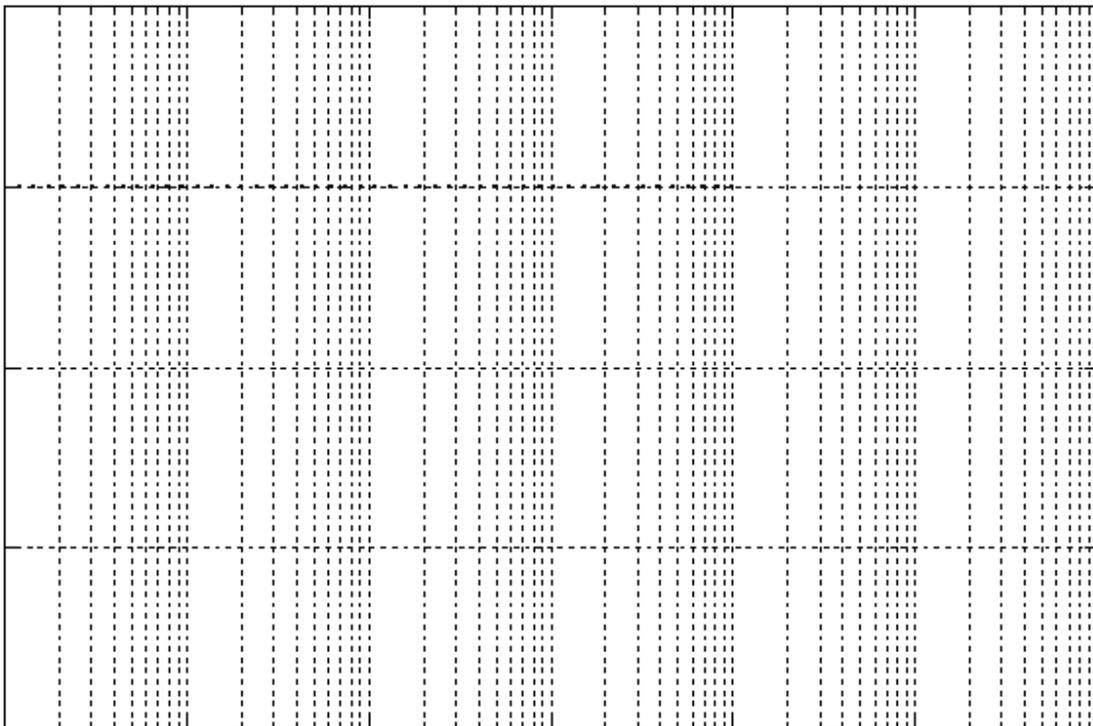
*Gráfica 6. Respuesta a una rampa de pendiente 3*



*Gráfica 7. Diagrama de Nyquist*



*Gráfica 8. Diagrama de Bode. Módulo*



*Gráfica 9. Diagrama de Bode. Fase*

## ***Comandos y Funciones de Matlab***

<b>Comando o Función</b>	<b>Explicación</b>
<b>abs</b>	Valor absoluto, magnitud compleja
<b>angle</b>	Ángulo de fase
<b>ans</b>	Respuesta cuando no se asigna una expresión
<b>atan</b>	Arcotangente
<b>axis</b>	Escalamiento de un eje en forma manual
<b>bode</b>	Diagrama de las trazas de Bode
<b>clear</b>	Borrar el área de trabajo
<b>clg</b>	Borrar las gráficas de la pantalla
<b>computer</b>	Tipo de computadora
<b>conj</b>	Compleja conjugada
<b>conv</b>	Convolución, multiplicación
<b>corrcoef</b>	Coefficientes de correlación
<b>cos</b>	Coseno
<b>cosh</b>	Coseno hiperbólico
<b>cov</b>	Covarianza
<b>deconv</b>	Deconvolución, división
<b>det</b>	Determinante
<b>diag</b>	Matriz diagonal
<b>eig</b>	Valores y vectores característicos
<b>exit</b>	Terminar sesión
<b>exp</b>	Base e exponencial
<b>expm</b>	Exponencial de una matriz
<b>eye</b>	Matriz identidad
<b>filter</b>	Implementación directa de un filtro
<b>format long</b>	Punto fijo en escala de 15 dígitos
<b>format long e</b>	Punto flotante de quince dígitos

<b>Comando o Función</b>	<b>Explicación</b>
<b>format short</b>	<b>Punto fijo en escala de 5 dígitos</b>
<b>format short e</b>	<b>Punto flotante de cinco dígitos</b>
<b>freqs</b>	<b>Respuesta en frecuencia de la transformada de Laplace</b>
<b>freqz</b>	<b>Respuesta en frecuencia de la transformada z</b>
<b>grid</b>	<b>Dibujar líneas de retícula</b>
<b>hold</b>	<b>Mantener la gráfica actual en la pantalla</b>
<b>imag</b>	<b>Parte imaginaria</b>
<b>inf</b>	<b>Infinito (<math>\infty</math>)</b>
<b>inv</b>	<b>Inverso</b>
<b>j</b>	$\sqrt{-1}$
<b>length</b>	<b>Longitud de vector</b>
<b>linspace</b>	<b>Vectores linealmente espaciados</b>
<b>Iog</b>	<b>Logaritmo natural</b>
<b>loglog</b>	<b>Gráfica logarítmica x-y</b>
<b>logrn</b>	<b>Logaritmo de una matriz</b>
<b>Iogspace</b>	<b>Vectores logarítmicamente espaciados</b>
<b>log10</b>	<b>Logaritmo en base 10</b>
<b>lqe</b>	<b>Diseño de estimador lineal cuadrático</b>
<b>lqr</b>	<b>Diseño de regulador lineal cuadrático</b>
<b>max</b>	<b>Valor máximo</b>
<b>mean</b>	<b>Valor medio</b>
<b>median</b>	<b>Valor mediano</b>
<b>mm</b>	<b>Valor mínimo</b>
<b>NaN</b>	<b>No es un número</b>
<b>nyquist</b>	<b>Diagrama polar de un sistema.</b>
<b>ones</b>	<b>Constante</b>

<b>Comando o Función</b>	<b>Explicación</b>
<b>pi</b>	$\pi$
<b>plot</b>	Gráfica lineal xy
<b>polar</b>	Gráfica polar
<b>poly</b>	Polinomio característico
<b>polyfit</b>	Ajuste de curva a un polinomio
<b>polyval</b>	Evaluación de un polinomio
<b>polyvalm</b>	Evaluación del polinomio de una matriz
<b>prod</b>	Producto de elementos
<b>quit</b>	Terminar sesión
<b>rand</b>	Generar números y matrices aleatorios
<b>rank</b>	Determinar el rango de una matriz
<b>real</b>	Parte real
<b>rem</b>	Residuo o módulo
<b>residue</b>	Expansión en fracciones parciales
<b>rlocus</b>	Gráfica del lugar geométrico de las raíces
<b>roots</b>	Raíces de un polinomio
<b>semilogx</b>	Gráfica semilogarítmica x-y (logaritmo del eje x)
<b>semilogy</b>	Gráfica semilogarítmica x-y (logaritmo del eje y)
<b>sign</b>	Función signo
<b>sin</b>	Seno
<b>sinh</b>	Seno hiperbólico
<b>size</b>	Dimensiones de renglón y columna
<b>sqrt</b>	Raíz cuadrada
<b>sqrtm</b>	Raíz cuadrada de una matriz
<b>std</b>	Desviación estándar
<b>step</b>	respuesta al escalón unitario
<b>sum</b>	Suma de elementos

<b>Comando o Función</b>	<b>Explicación</b>
<b>tan</b>	<b>Tangente</b>
<b>tanh</b>	<b>Tangente hiperbólica</b>
<b>text</b>	<b>Texto arbitrariamente colocado</b>
<b>title</b>	<b>Título de la gráfica</b>
<b>trace</b>	<b>Traza de una matriz</b>
<b>who</b>	<b>Listado de las variables en memoria</b>
<b>xlabel</b>	<b>Leyenda del eje x</b>
<b>ylabel</b>	<b>Leyenda del eje y</b>
<b>zeros</b>	<b>Cero</b>



---

## *Práctica 2*

### *Introducción a la simulación de sistemas mediante Simulink.*

### *Sistemas de primer, segundo y tercer orden.*

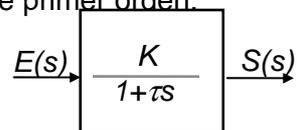
---

#### **Objetivo**

En esta práctica se pretende que el alumno tome contacto con una herramienta informática de ayuda a la simulación y análisis de sistemas. Esta herramienta es el programa "MATLAB WITH SIMULINK" que permite simular el comportamiento dinámico de sistemas, sirviendo de base estas simulaciones para el análisis de su comportamiento.

#### **Sistemas de primer orden básico**

Sea el sistema de primer orden:



#### **Estudio teórico**

Para  $K=3$  y  $\tau=0.5$  obtener la expresión temporal de la salida del sistema. Para los siguientes casos:

Entrada en impulso unitario o Delta de Dirac.

Entrada en escalón de amplitud 2.5.

Entrada en rampa de pendiente 0.5

Entrada en rampa parabólica

## **Estudio Práctico**

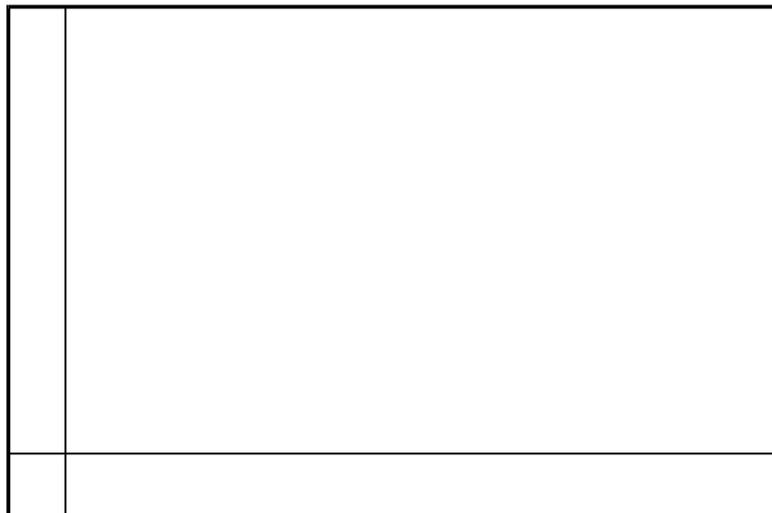
Introducir el sistema en "Simulink" y verificar su comportamiento en comparación con el estudio teórico .

Se deberá introducir un bloque de función de transferencia (paleta de linear), un bloque de entrada en escalón unitario (paleta de sources), y un visualizador del tipo "AutoEscale" (Paleta de Extras; Subpaleta de Display Devices).

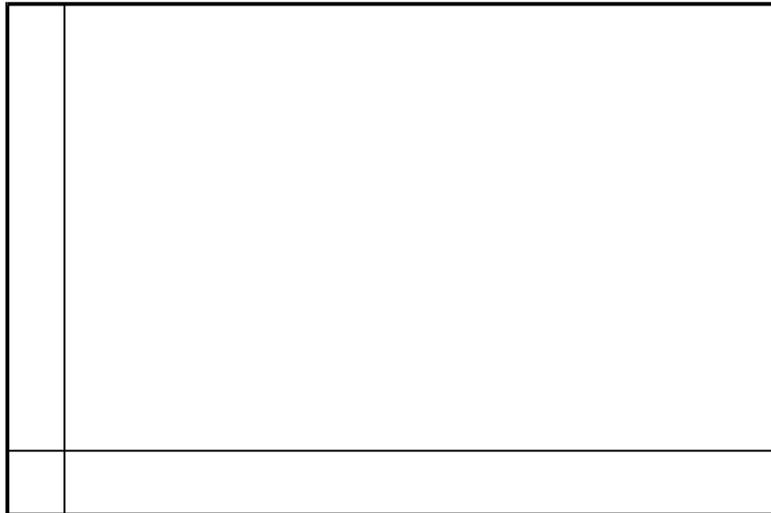
Para obtener una mejor simulación de la respuesta se recomienda:

-Dar un valor finito (por ejemplo 5 segundos) al parámetro STOP\_TIME del menú de simulación, submenú de parameters.

Obtener las siguientes gráficas:

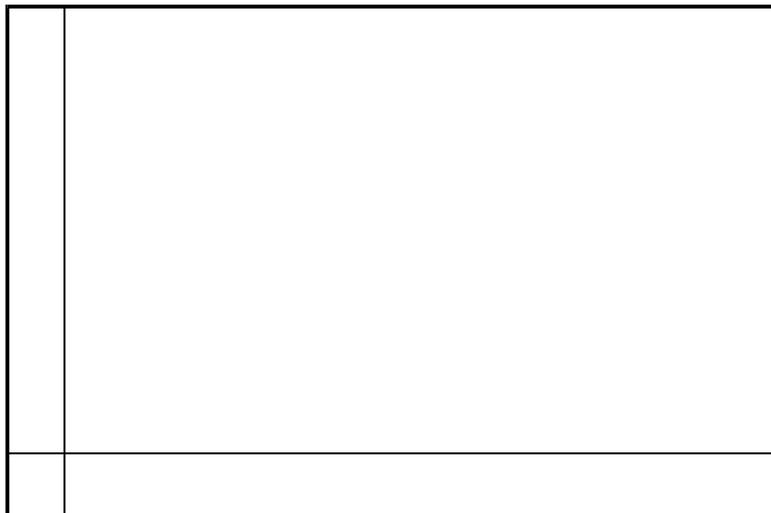


*Gráfica 1. Respuesta del sistema ante una entrada en Impulso.*



*Gráfica 2. Respuesta del sistema ante una entrada en escalón unitario.*

Indicar en la gráfica 2 la ganancia, cte de tiempo y tiempo de respuesta del sistema



*Gráfica 3. Respuesta del sistema ante una entrada en rampa de pendiente 2.*

## **Sistemas de primer orden completo**

Sea el sistema de la figura:

$$E(s) \longrightarrow \boxed{\frac{K(1+as)}{(1+\tau s)}} \longrightarrow S(s)$$

Donde  $E(s)$  es la entrada al Sistema y  $S(s)$  es la salida del Sistema

## **Estudio teórico**

Para  $K=2$ ,  $a=3$  y  $\tau=4$  obtener la expresión temporal de la salida del sistema. Para los siguientes casos:

Entrada en escalón de amplitud 5.

Entrada en rampa de pendiente 0.5.

## **Estudio practico**

Introducir el sistema en "Simulink".

Obtener las siguientes gráficas:



*Gráfica 4. Respuesta del sistema ante una entrada en escalón de amplitud 0.5.*

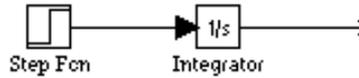


*Gráfica 5. Respuesta del sistema ante una entrada en rampa unitaria.*

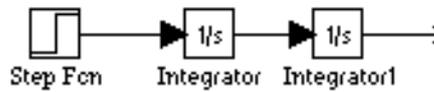


**IMPORTANTE :**

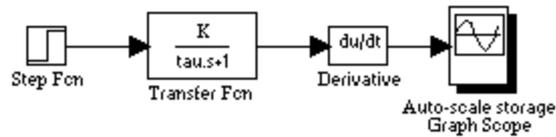
Para introducir una entrada en rampa al sistema, se tendrá que situar un integrador detrás del escalón, tal como se observa en la figura siguiente.



Para introducir una entrada en rampa parabólica al sistema, se tendrá que integrar dos veces escalón, tal como se observa en la figura siguiente.



Para obtener la salida de un sistema ante una entrada en impulso unitario o "Delta de Dirac", se tendrá que derivar la salida del sistema ante una entrada en escalón. tal como se observa en la figura siguiente:



## **Sistemas De Segundo Orden Básico**

Sea el sistema ::

$$\frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

### **Estudio Teórico**

Para  $K=3$ ,  $\omega_n=4$  y  $\xi=1$  obtener la expresión temporal de la salida del sistema. Para los siguientes casos:

Entrada en impulso unitario o Delta de Dirac.

Entrada en escalón de amplitud 3.5.

Entrada en rampa de pendiente 2.5

### **Estudio Práctico**

Introducir el sistema en Simulink y verificar su comportamiento ante las entradas en escalón unitario y rampa unitaria.

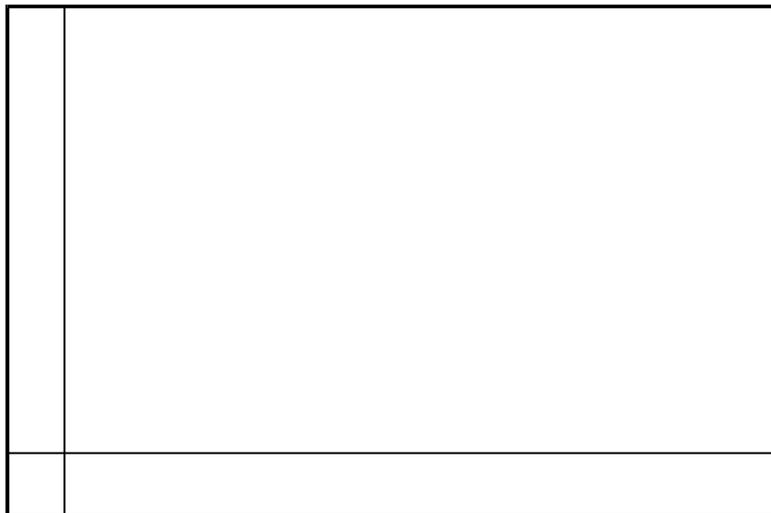
Realizar las siguientes gráficas:

Para  $K=5$ ,  $\omega_n=5$  y  $\xi=1$ .



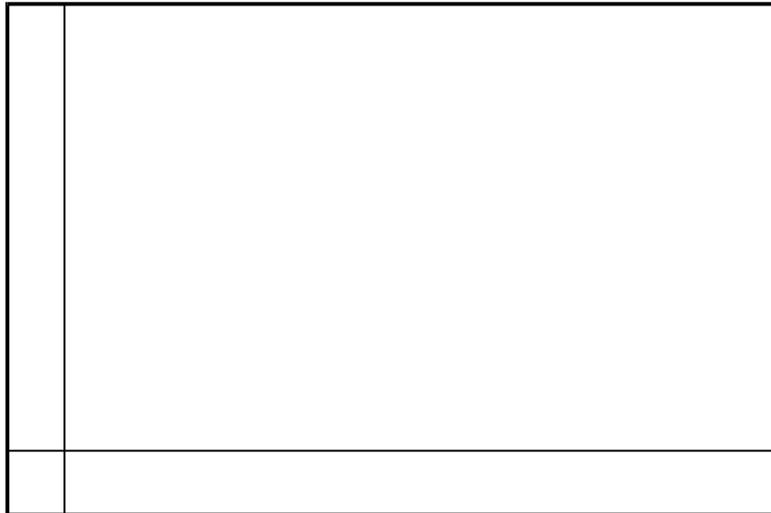
*Gráfica 6. Respuesta del sistema ante una entrada en escalón de amplitud 2.*

*Indicar en la respuesta la ganancia, sobreoscilación, tiempo de subida, tiempo de pico y el tiempo de respuesta del sistema.*



*Gráfica 7. Respuesta del sistema ante una entrada en rampa de pendiente 0.5.*

Para  $K=2$ ,  $\omega_n=1$  y  $\xi=0.2$ .



*Gráfica 8. Respuesta del sistema ante una entrada en escalón de amplitud 2.*

Indicar en la respuesta la ganancia, sobreoscilación, tiempo de subida, tiempo de pico y el tiempo de respuesta del sistema.



*Gráfica 9. Respuesta del sistema ante una entrada en rampa de pendiente 0.5.*

*Comentar los resultados obtenidos en cada gráfica.*



### **SISTEMAS DE TERCER ORDEN**

Sea el sistema:

$$\frac{K\omega_n^2}{(1+\tau s)(s^2 + 2\xi\omega_n s + \omega_n^2)}$$

### **ESTUDIO TEORICO**

Para  $K=2$  ,  $\omega_n =5$  ,  $\xi=1$  y  $\tau=1$ , obtener la expresión temporal de la salida del sistema en los siguientes casos:

Entrada en escalón unitario.

Entrada en rampa unitaria.

### **ESTUDIO PRACTICO.**

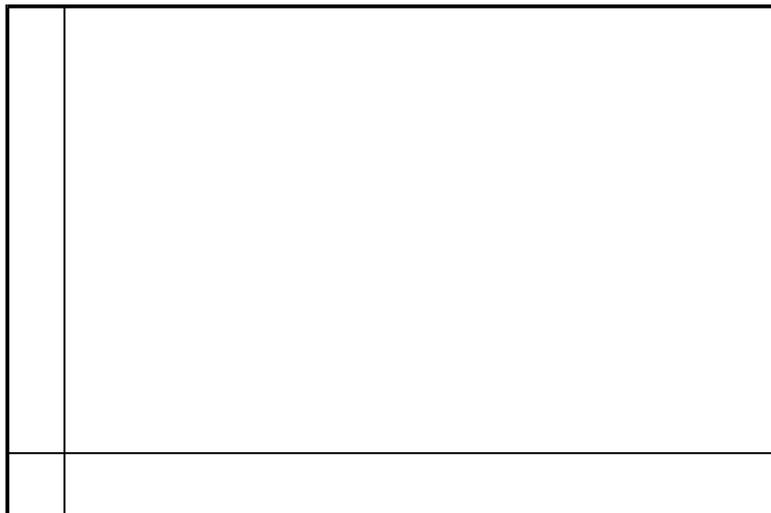
Introducir el sistema en SIMULINK y verificar su comportamiento .  
Realizar las siguientes gráficas:

Para  $K=2$  ,  $\omega_n =5$  ,  $\xi=1$  y  $\tau=1$



*Gráfica 10. respuesta del sistema ante una entrada en escalón.*

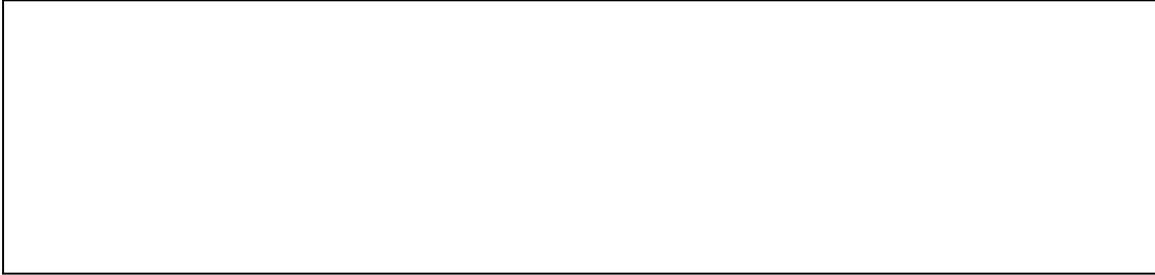
Para  $K=1$  ,  $\omega_n =5$  ,  $\xi=0.5$  y  $\tau=2.5$



*Gráfica 11. respuesta del sistema ante una entrada en escalón.*

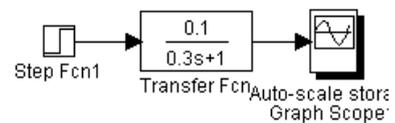
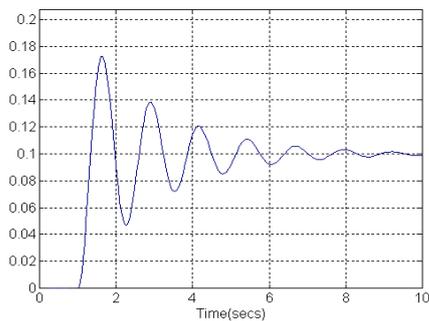
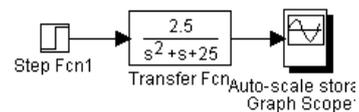
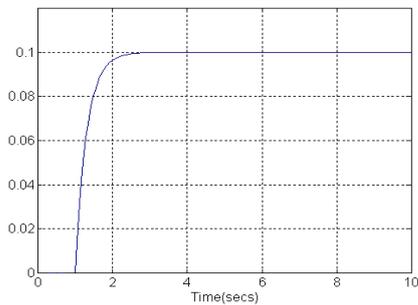
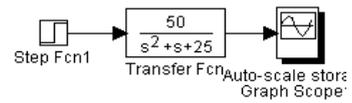
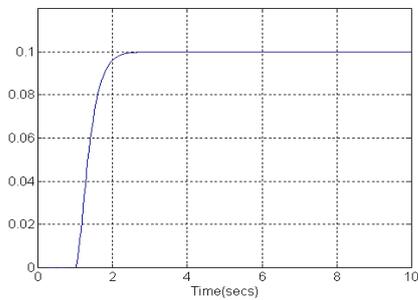
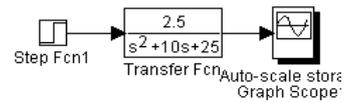
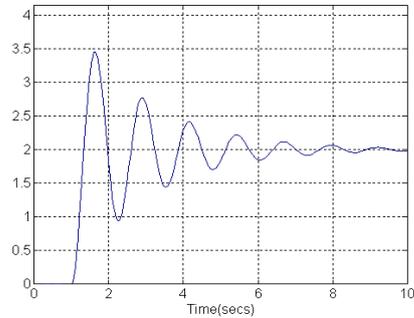


Comentar los resultados obtenidos en cada gráfica.



## Cuestiones adicionales

1. Empareja **razonadamente** las respuestas con el diagrama de bloques correspondiente ( la entrada es un escalón unitario con step-time=1seg)



2-Escribe la transformada de Laplace de las siguientes funciones.

Escalón de amplitud 2.  
Función Impulso unitario.  
Rampa de pendiente 0.5

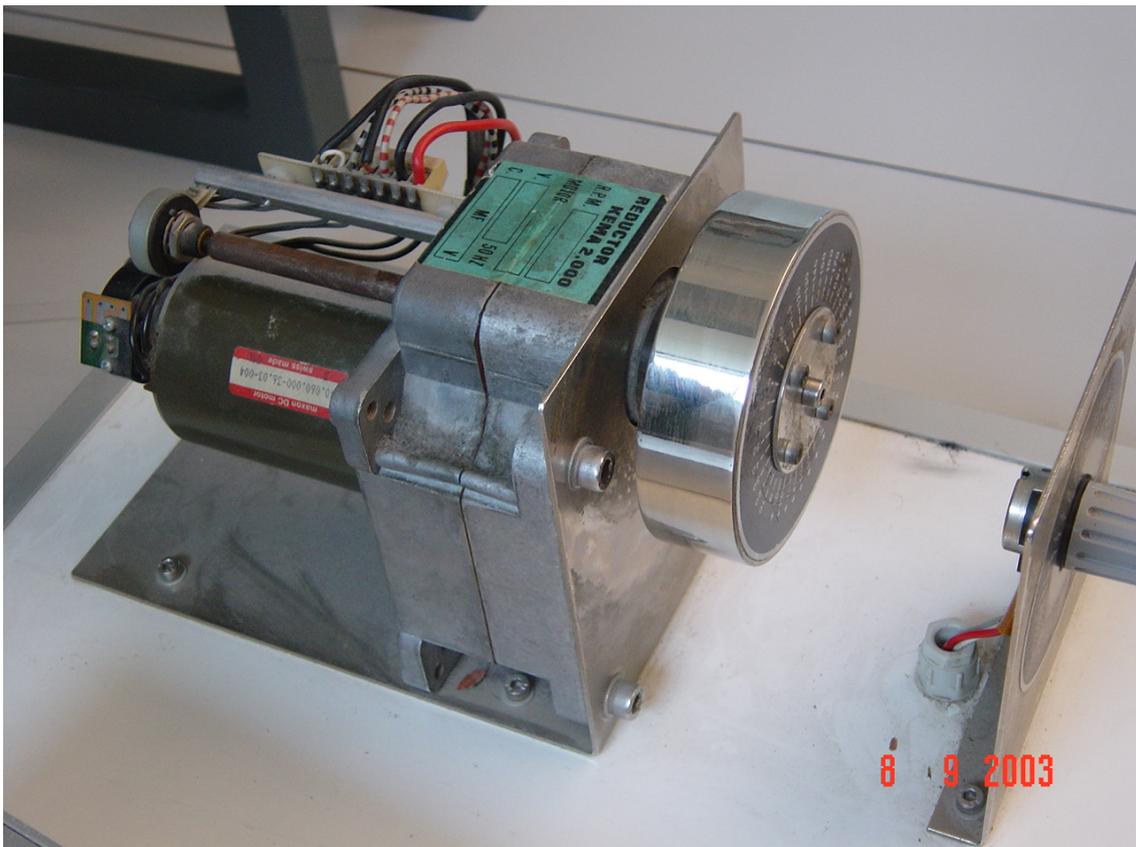
3-Parámetros característicos de la respuesta al escalón de un sistema de primer orden básico.

---

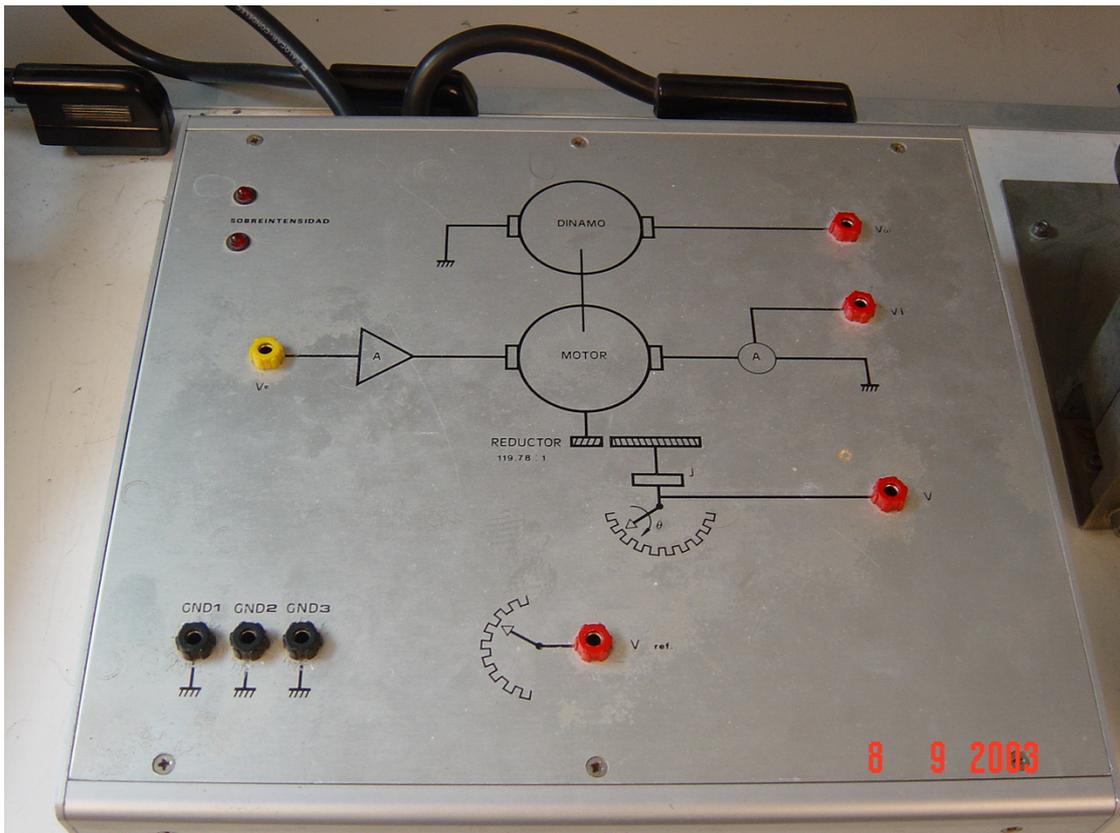
## *Práctica 3*

# *Control de posición y velocidad de un motor de corriente continua*

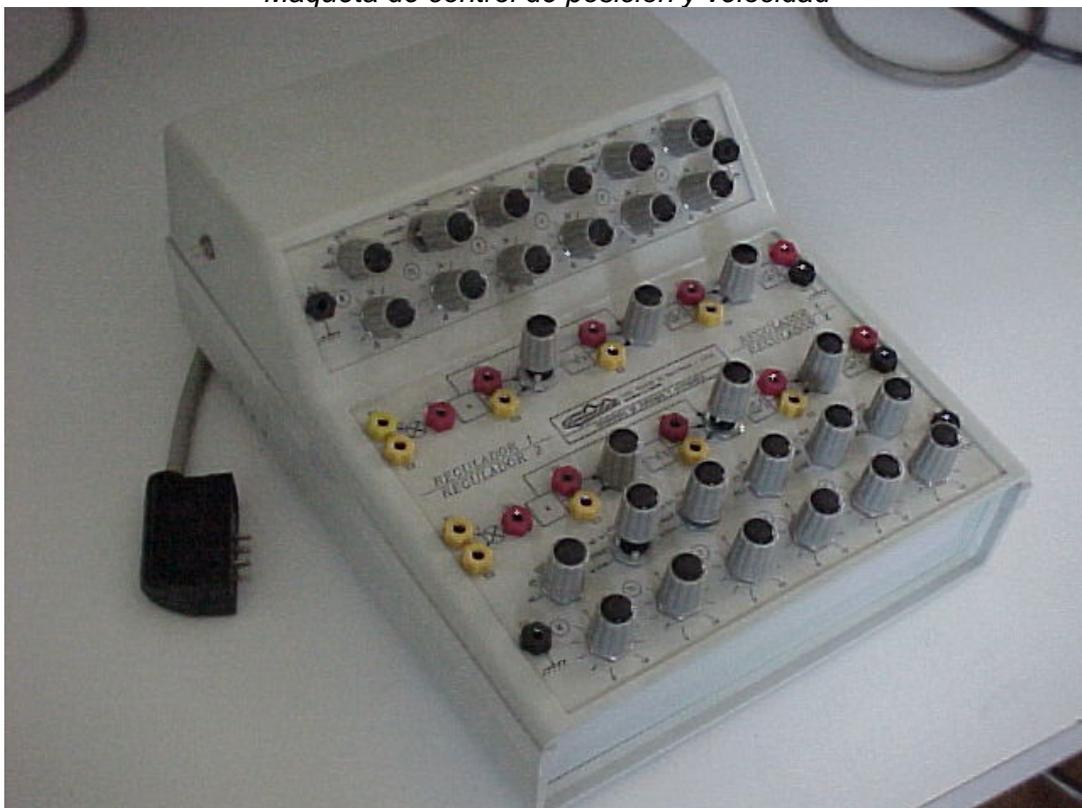
---



*Maqueta de control de posición y velocidad*



*Maqueta de control de posición y velocidad*



*Caja de Reguladores Analógicos*

## OBJETIVO:

Esta práctica pondrá en contacto al alumno con un sistema de control real. Durante su desarrollo se pondrán de manifiesto la influencia de las limitaciones de la acción sobre las especificaciones del control, el efecto de las perturbaciones sobre la salida, la clara dependencia de calidad del control en función de la calidad del sensor...

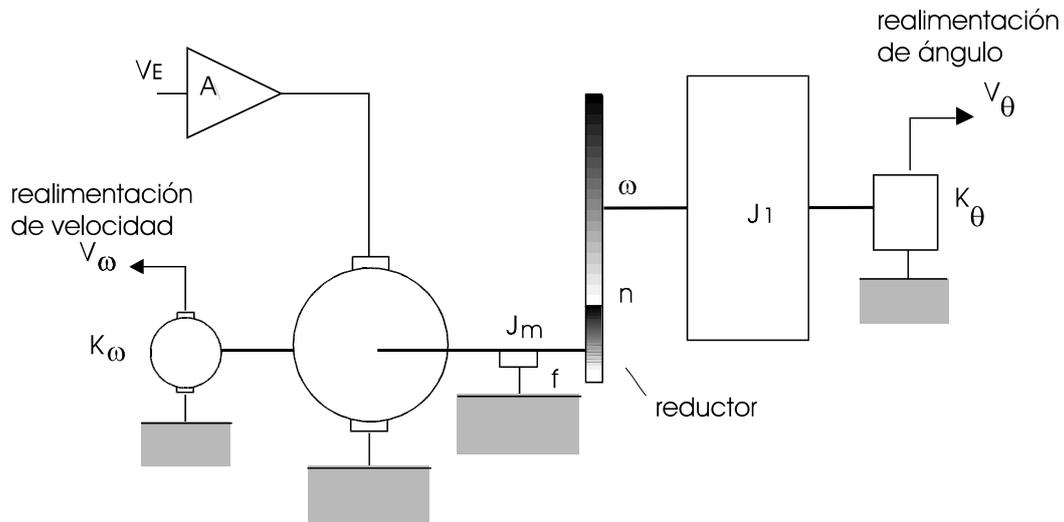


Diagrama esquemático del Servosistema.

La figura representa el esquema de un accionamiento electromecánico. Este sistema consta de un amplificador de potencia de ganancia  $A$ , un motor de corriente continua controlado por inducido, un reductor de relación de reducción  $n$  y la inercia  $J_1$ . Sobre el eje del motor actúa una fricción viscosa de valor  $f$ . Para permitir el control de dicho sistema se dispone de dos sensores, uno que mide la posición  $\theta$  del eje de la inercia de constante  $K_\theta$  V/rad y una dínamo tacométrica sobre el eje del motor de constante  $K_\omega$  V/rad\*sg<sup>-1</sup>.

Existen en el motor una fricción seca y juego en el reductor que se pueden modelar como perturbaciones aditivas sobre el par del eje.

Los valores numéricos de los parámetros del sistema son:

$$A=2'7$$

$$R=2'79\Omega$$

$$K_p=0'0729\text{Nw}\cdot\text{m}/\text{A}$$

$$K_e=0'07289\text{ V}/\text{rad}\cdot\text{sg}^{-1}$$

$$n=119'75$$

$$f=0'655\cdot 10^{-4}\text{Nw}\cdot\text{m}/\text{rad}\cdot\text{sg}^{-1}$$

$$J=1'08\cdot 10^{-4}\text{ Kg}\cdot\text{m}^2 \text{ Inercia total referida al eje del motor } (J_m+J_1/n^2)$$

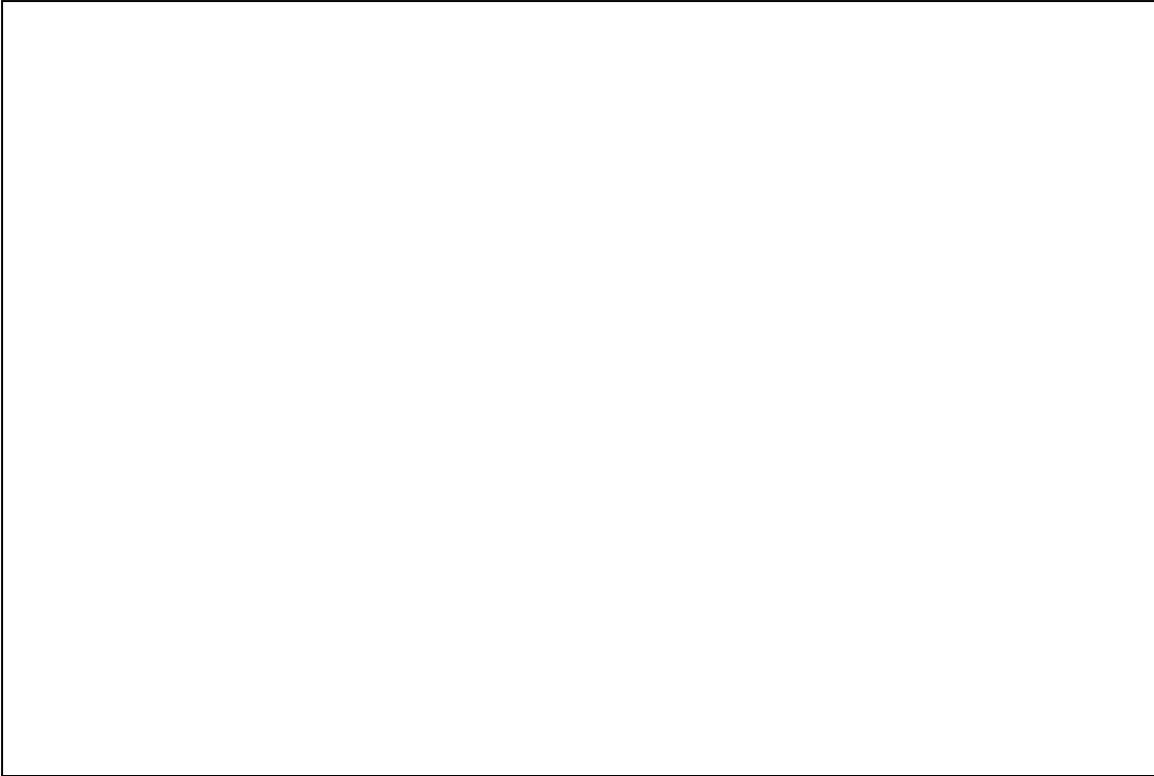
Sensores:

$$K_\theta =2'75\text{ V}/\text{rad}$$

$$K_\omega =2'792\cdot 10^{-2}\text{V}/\text{rad}\cdot\text{sg}^{-1}.$$

## **Estudio Teórico**

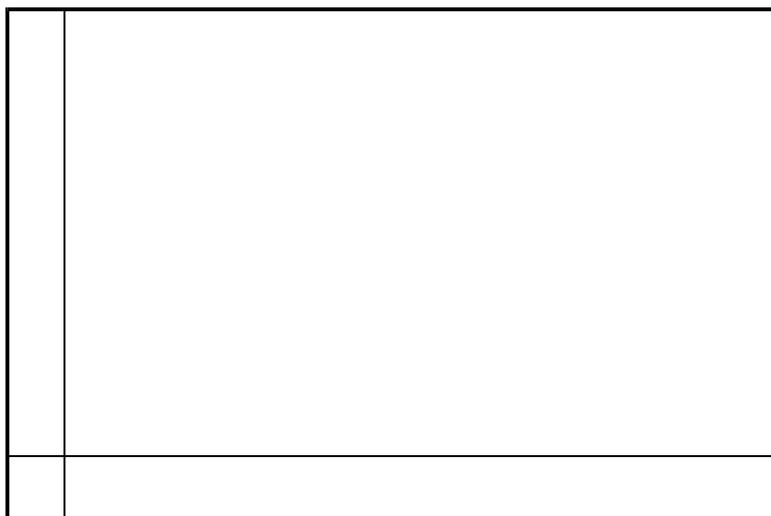
ET1-Obtener la Función de Transferencia en bucle abierto  $\theta/VE$ .



## **Estudio Práctico**

Con el sistema en bucle abierto, obtener la salida del sistema en bucle abierto ante una entrada en escalón de tensión de 5 voltios. (Utilizar el generador de señales).

GRAFICA1. Dibujar las gráficas de la señal que ofrece el sensor de posición y el sensor de velocidad



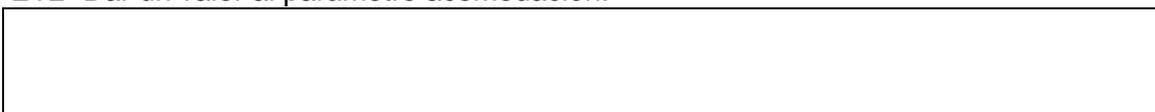
*Gráfica 1*

En base a la gráfica anterior. Realizar la identificación empírica del sistema.



## Estudio Teórico

ET2- Dar un valor al parámetro acomodación.



ET3-Diseñar un esquema de control serie utilizando el sensor disponible para controlar la posición  $\theta$ . Calcúlese el regulador más sencillo que permita obtener las siguientes especificaciones:

$$\begin{aligned} e_p &= 0 \\ SO &\leq 20\% \\ t_r &\leq 1 \text{ sg} \end{aligned}$$

En este cálculo no se tendrá en cuenta que la salida del regulador debe estar en el rango comprendido entre los  $\pm 10$  voltios.

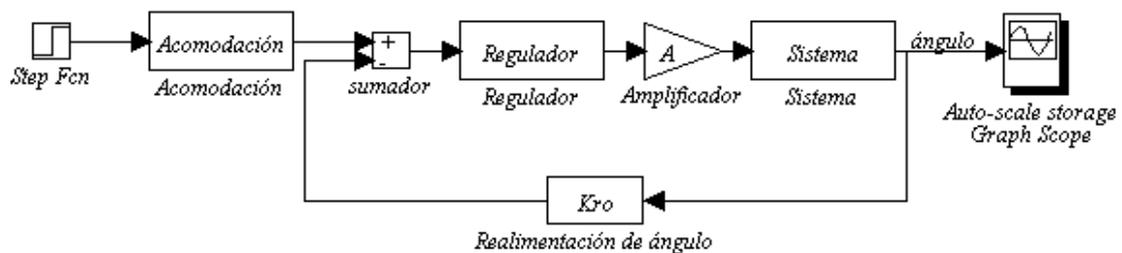


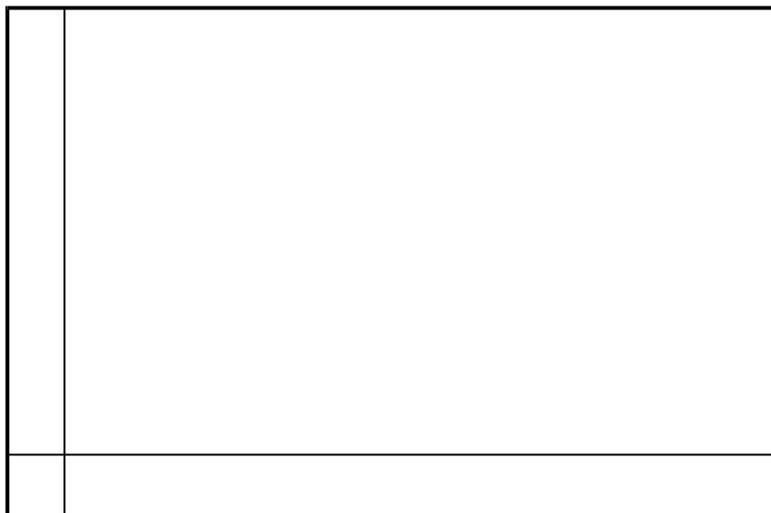
Diagrama de bloques del sistema de control de posición





## **Estudio Práctico**

*Gráfica 2. Con el regulador obtenido en el apartado ET3, obtener la salida del sistema en bucle cerrado ante una entrada en escalón de ángulo de amplitud 0.87 rad*



*Gráfica 2*

## **Estudio Teórico**

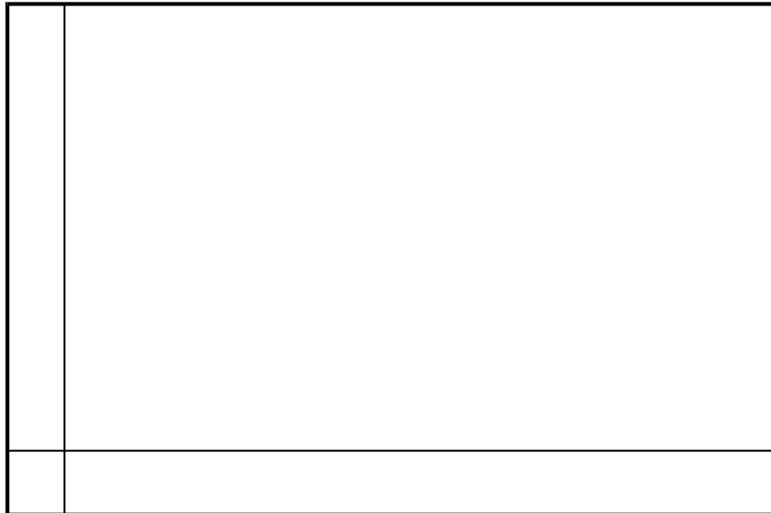
ET4-Calcular la acción de dicho regulador sobre el sistema para consignas en escalón de 0 a 0'87 rad. Teniendo en cuenta que la salida de los reguladores disponibles está limitada a 10v, determinar qué regulador podremos implementar para que no se sature su acción con la consigna deseada y las especificaciones impuestas. ¿Cuál es ahora el tiempo de respuesta y la sobreoscilación?



Montar el esquema de control con el regulador del apartado ET4 sobre el equipo de prácticas y verificar el cumplimiento de las especificaciones. Reajustar manualmente el regulador teórico calculado.

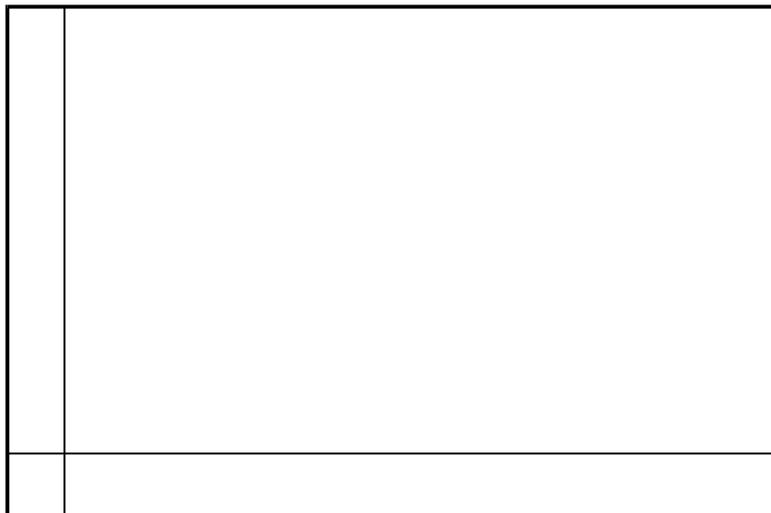
## **Estudio Práctico**

*Gráfica 3. Obtener la acción del regulador sobre el sistema en bucle cerrado ante una entrada en escalón de ángulo de amplitud 0.87 rad*



Gráfica 3

Gráfica 4. Obtener la salida del sistema en bucle cerrado ante una entrada en escalón de ángulo de amplitud 0.87 rad



Gráfica 4

## Estudio Teórico

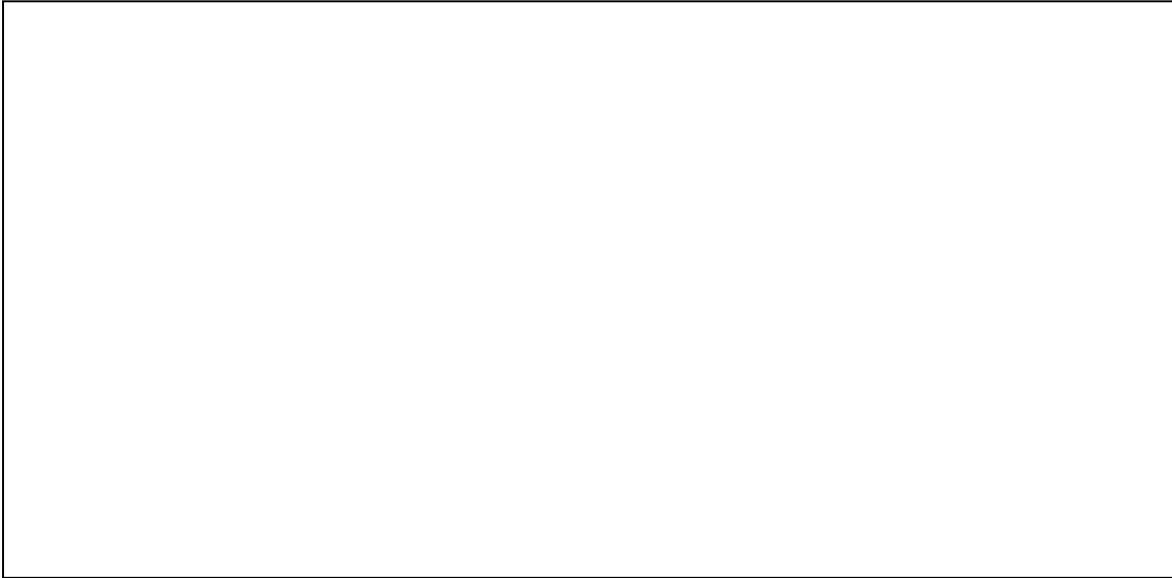
ET5-Diseñar un esquema de control de velocidad  $\omega$  usando el sensor disponible y calcular un regulador que permita obtener las siguientes especificaciones:

$$e_p = 0$$

$$SO \leq 20\%$$

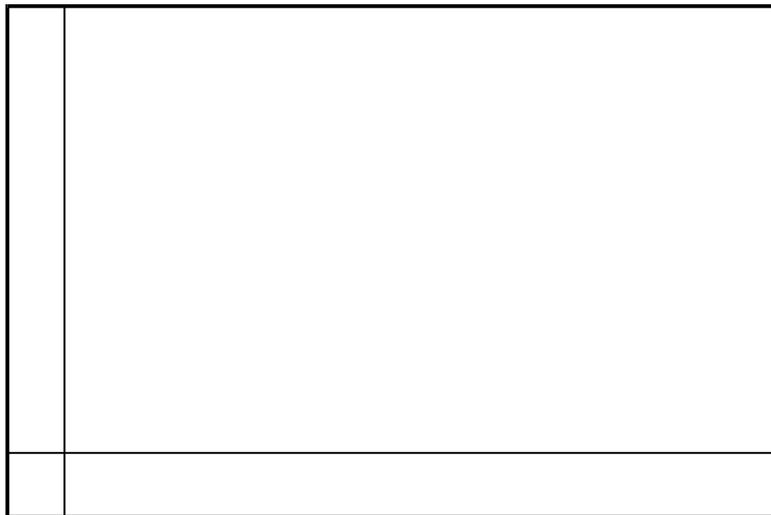
$$t_r \leq 0'6 \text{ sg.}$$

Para ello suponer que no existen restricciones sobre el valor máximo de las acciones.



## **Estudio Práctico**

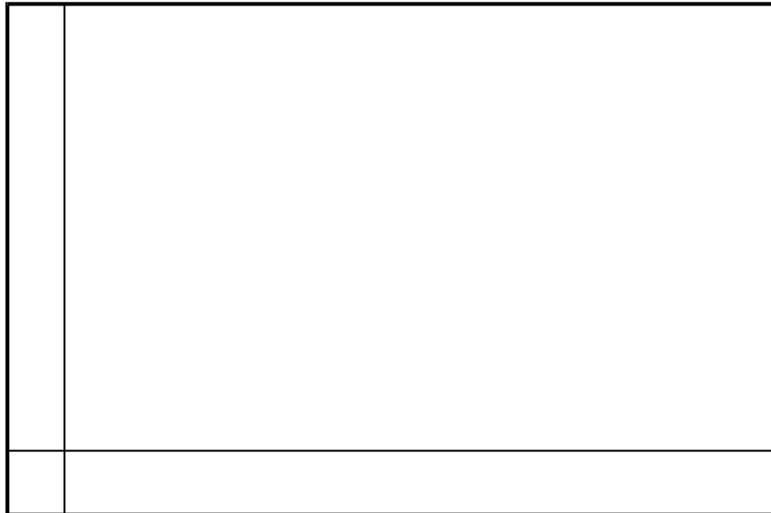
*Gráfica 5. Obtener la salida del sistema en bucle cerrado ante una entrada en escalón de velocidad angular de amplitud 0.87 rad/seg*



*Gráfica 5*

Montar el esquema de control con el regulador obtenido en el apartado ET5. Comprobar la acción de dicho regulador. Ajustar la consigna para que no se sature la acción del regulador en ningún instante.

Gráfica 6 Dibujar la respuesta medida  $V_{\omega}$  para un caso en el que no existe saturación de la acción, indicando el valor del escalón utilizado en voltios y en rad/sg.



*Gráfica 6*

¿Por qué ahora, al controlar velocidad angular, el  $e_p$  es nulo?

Introducir sobre el mismo sistema regulado una entrada en rampa (onda triangular) de pendiente unidad (atención)

Gráfica 7 Dibujar la respuesta medida indicando el valor  $e_v$ .



*Gráfica 7*

Indicar el valor del error de velocidad en voltios y en rad/seg

---

## *Práctica 4*

# *Control de sistemas continuos mediante Autómatas Programables.*



---

### **Objetivos**

*En esta práctica se realiza una introducción al control de sistemas continuos mediante Autómata Programable. Se realizarán programas en lenguaje de Autómata y se monitorizarán los procesos mediante un Sistema Scada.*

Se dispone de un equipo de prácticas dotado de:

- Un ordenador personal PC.

- Autómata Programable Tsx3722 dotado con entradas y salidas analógicas.

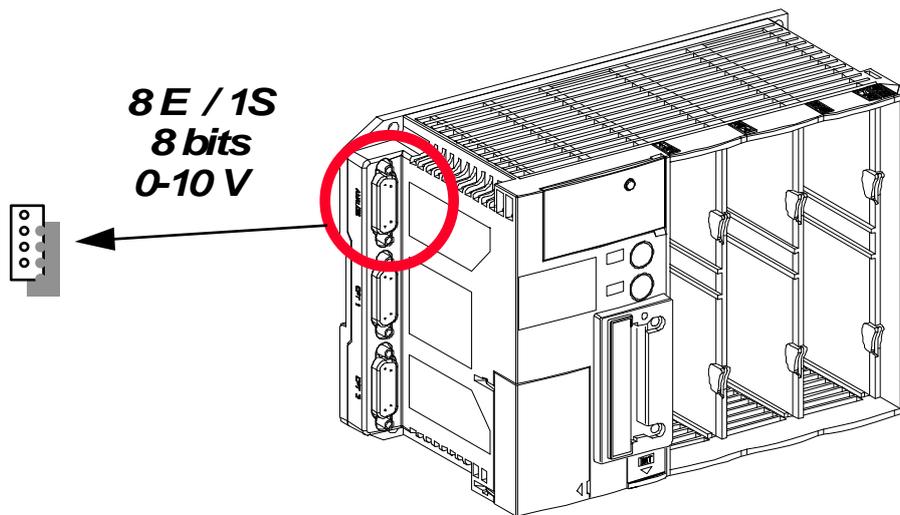
- Placa de simulación de sistema de segundo orden.

- Programa SCADA SPIA.

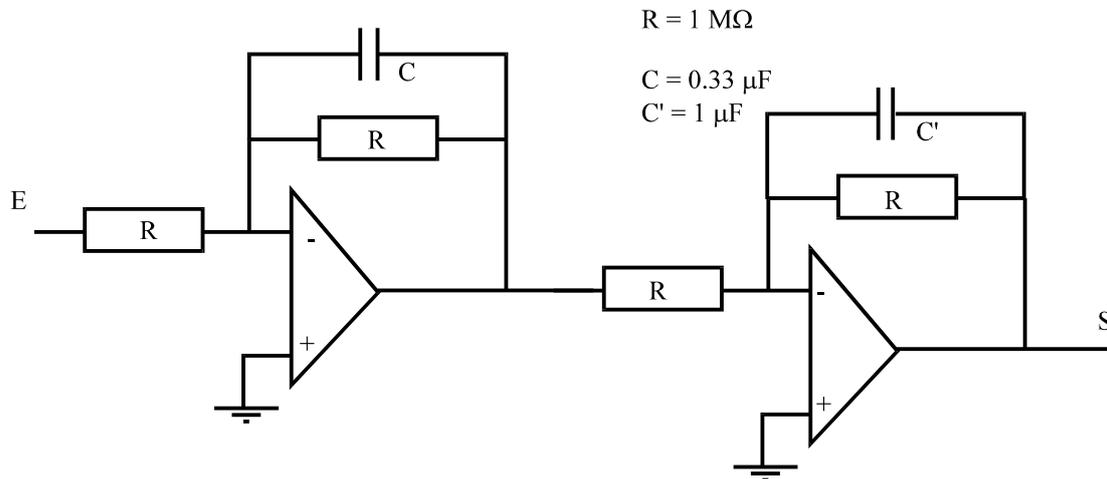
El Autómata Programable Tsx3722. Dispone de:

- 8 vías de entrada analógica. Rango 0..10voltios. 8 bits de resolución.

- 1 vía de salida analógica. Rango 0..10voltios. 8 bits de resolución.



El sistema continuo a controlar está realizado por medio de circuitos basados en el amplificador operacional. Su función de transferencia es de segundo orden:



Sistema de Segundo orden.

$$G(s) = \frac{1}{(1 + 0.33s)(1 + s)}$$

Función de Transferencia.

El canal de salida analógica se lleva a la entrada E del sistema físico. La salida del sistema S se lleva la entrada analógica 2. La masa del circuito se lleva a la masa del canal de entrada analógica y a la masa del canal de salida analógica.

## Proceso de creación de aplicaciones de Autómata.

Dentro del grupo de programas Modicon-Telemecanique ejecutar el programa PL7 Pro V3.1. A continuación crear una nueva Aplicación (menú Fichero, Nuevo)

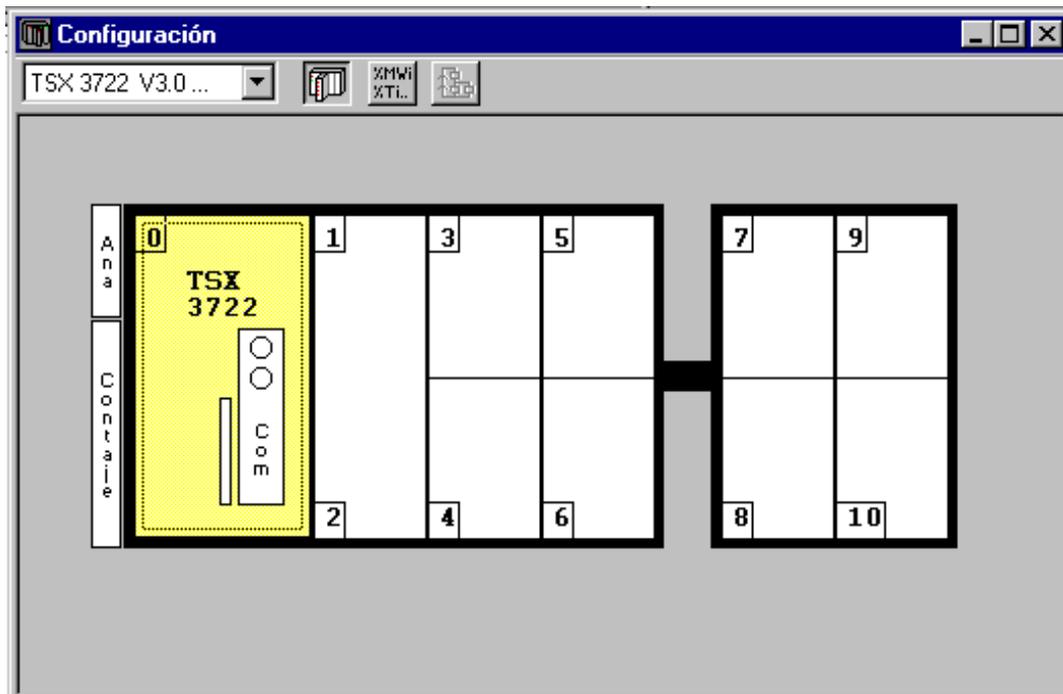


Elegir TSX Micro, la versión del procesador del autómata (TSX 3710 ó TSX 3722 ) versión del S.O. 2.0. La aplicación no contendrá graphicet.

Aparece la pantalla del Navegador de Aplicación:

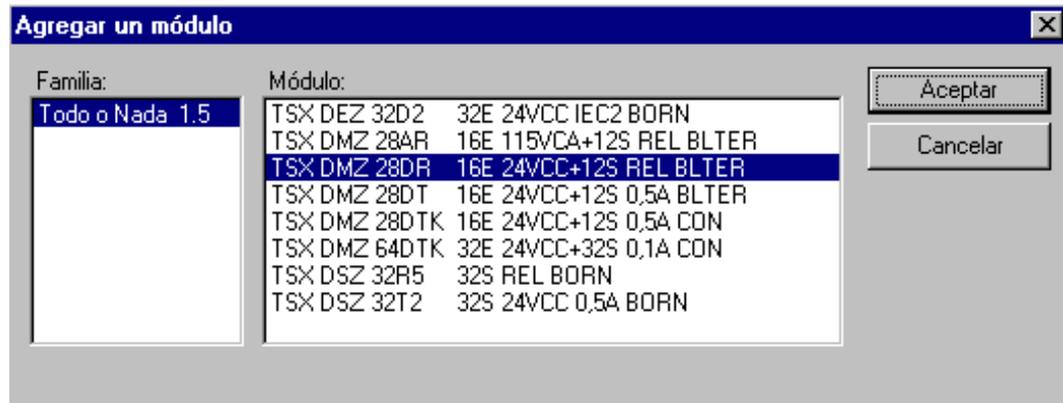


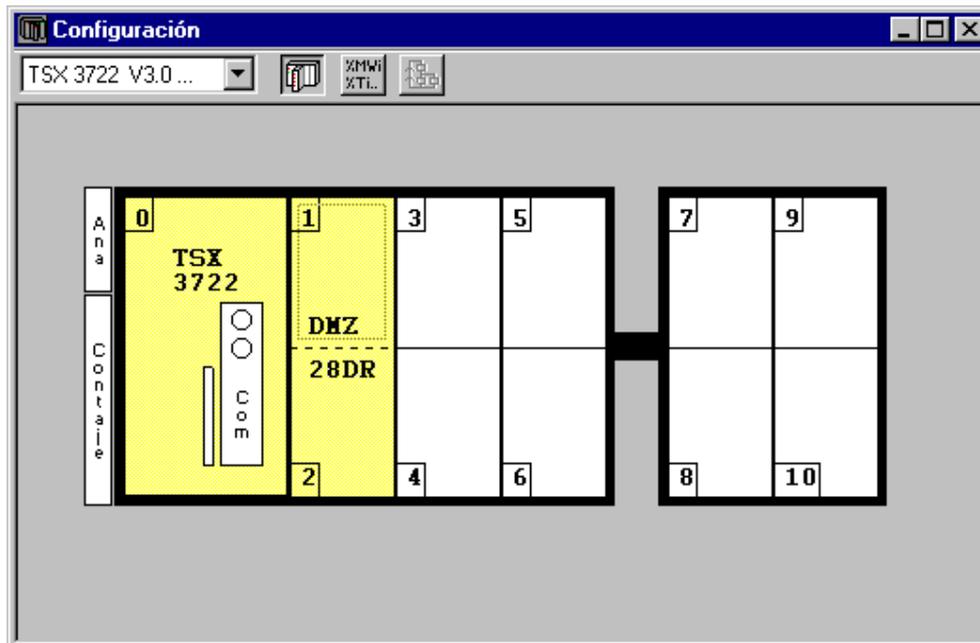
Hacer click en el icono de configuración y a continuación doble click en configuración hardware.



Configuración Hardware. El Procesador elegido es el TSX3722.

Añadir los módulos de entradas-salidas presentes en el automático que vais a programar (hacer doble click en la zonas del rack). Si la configuración hardware de la aplicación no coincide con la del automático es posible que no funcione correctamente la lectura de entradas y la escritura de salidas.





*Configuración hardware para procesador TSX3722.*

La configuración hardware depende del autómata que se vaya a programar. No tiene porque coincidir con las dos figuras anteriores. Una vez introducida la configuración

hardware, hay que validarla haciendo un click sobre el icono  o pulsando Mayusculas+Enter.

Volver a la pantalla del navegador de la aplicación. Hacer un click sobre Programa- Secciones-Tarea Mast. Ir al menú de Edición -Crear:



Seleccionar el lenguaje ST (Literal Estructurado).

## Ejercicio 1.

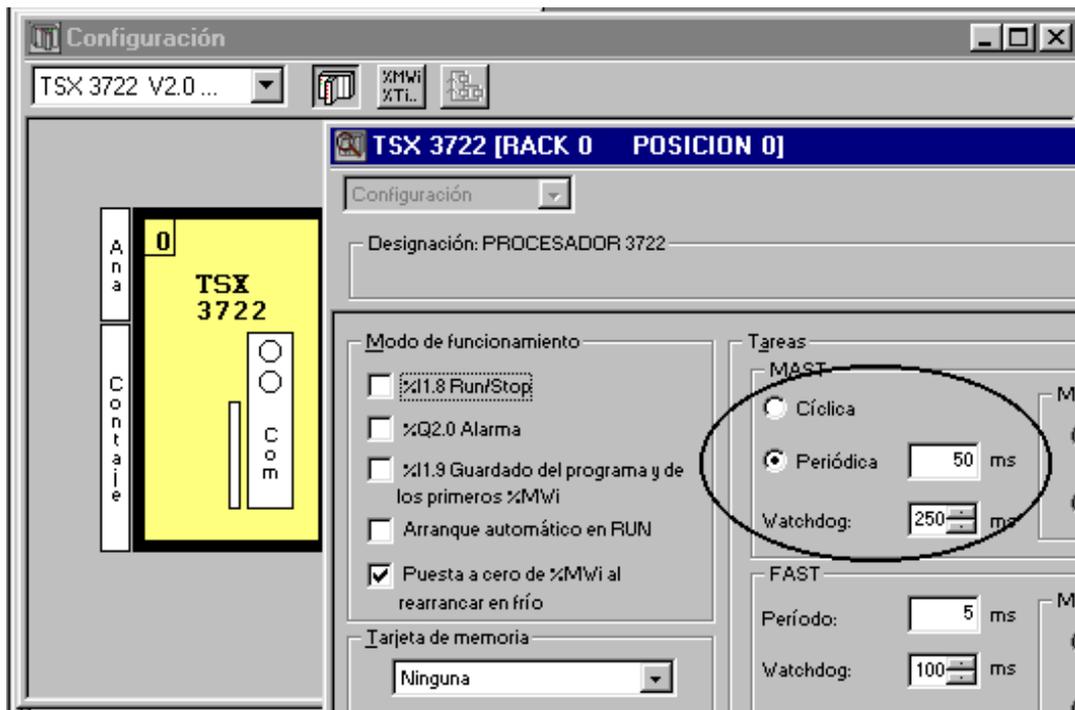
Introducir el siguiente programa en la Tarea Mast.

```
PID(' ',' ',%IW0.2,%QW0.10,%M10,%MW20:43);
(*control manual desde el panel de control *)
%M10:=%I1.0;
(*Incremento-decremento de la referencia *)
IF %I1.5 THEN
%MW20:=%MW20+20;
END_IF;
IF %I1.6 THEN
%MW20:=%MW20-20;
END_IF;
(*Incremento-decremento de la Acción Manual *)
IF %I1.7 THEN
%MW21:=%MW21+20;
END_IF;
IF %I1.8 THEN
%MW21:=%MW21-20;
END_IF;
(* definición del valor de la ganancia del regulador kp*100 *)
%MW22:=320;
(* tiempo de acción integral en décimas de segundo*)
%MW23:=10;
(* tiempo de acción derivada en decimas de segundo*)
%MW24:=0;
(*periodo de muestreo en centesimas de segundo*)
%MW25:=10;
(* acción Maxima*)
%MW26:=10000;
%MW28:X0:=TRUE;
%MW28:X8:=TRUE;
```

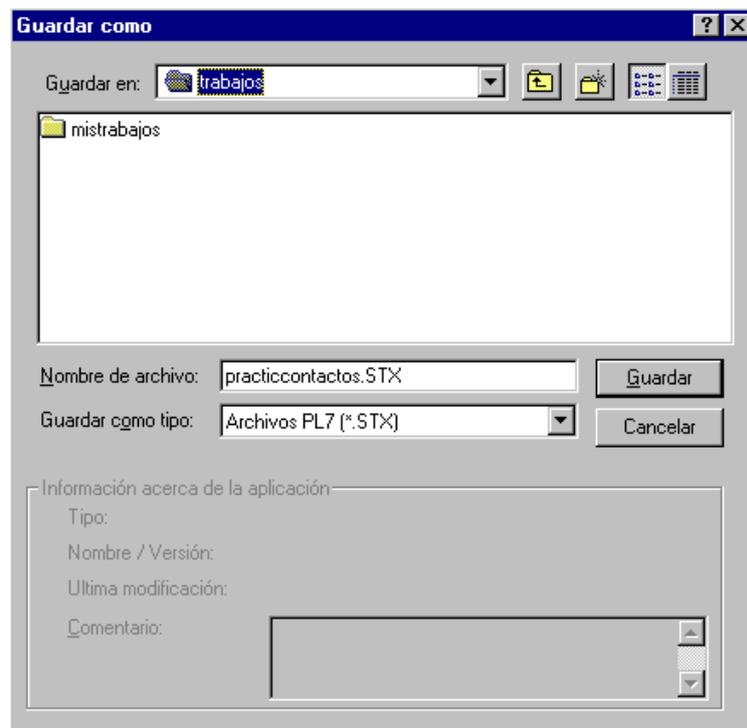
Una vez introducido el primer ejercicio, pulsar shift+ enter o hacer click en el icono



Ir a la Pantalla del Navegador de Aplicación hacer doble click sobre Configuración - Configuración Hardware

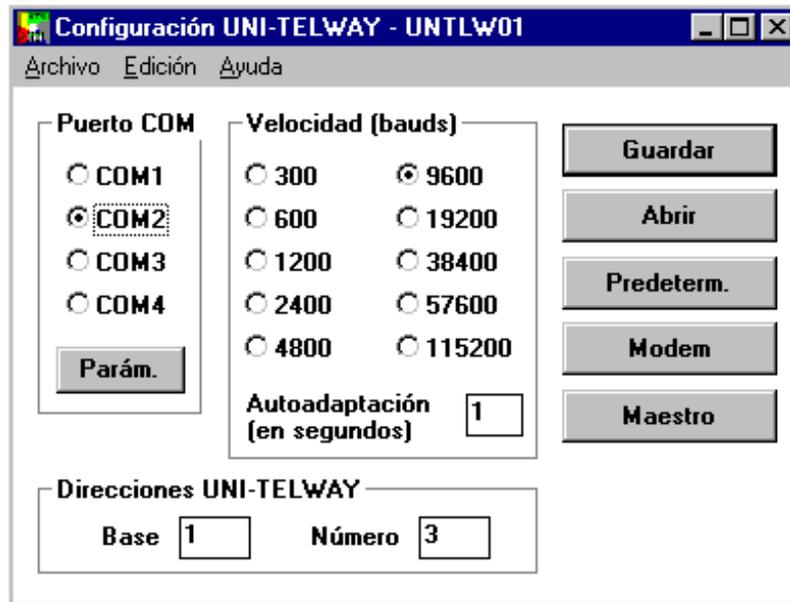


Configurar la tarea maestra como periódica con periodo 50 ms.  
Se deberá guardar la aplicación al disco duro. Dentro del directorio trabajos, crear un subdirectorio, y guardar la aplicación. Procurar copiar la aplicación a disquete, ¡¡no se garantiza su supervivencia en los discos duros!!.

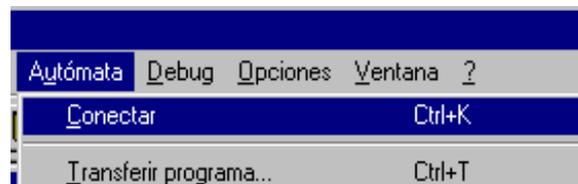


A continuación transferir la aplicación al automático. El automático deberá estar alimentado, y su cable de programación conectado al COM2 del ordenador (suele ser

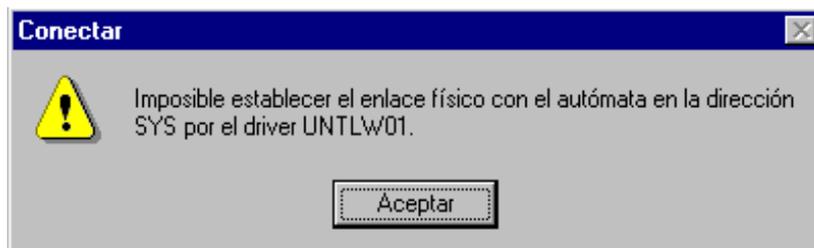
el puerto macho de 25 pines). Se deberá comprobar que el driver de comunicaciones esta correctamente configurado en el COM2 a una velocidad de 9600 baudios. Ejecutar el programa Unitelway del grupo Modicon Telemecanique:



A continuación ejecutar:



Si en este instante se cuelga el ratón, reiniciar el ordenador y configurar el driver Unitelway correctamente. Si aparece:



El Software PL7 Pro no encuentra el autómata. Las razones pueden ser varias:

No tenéis Autómata (os lo ha quitado el compañero).

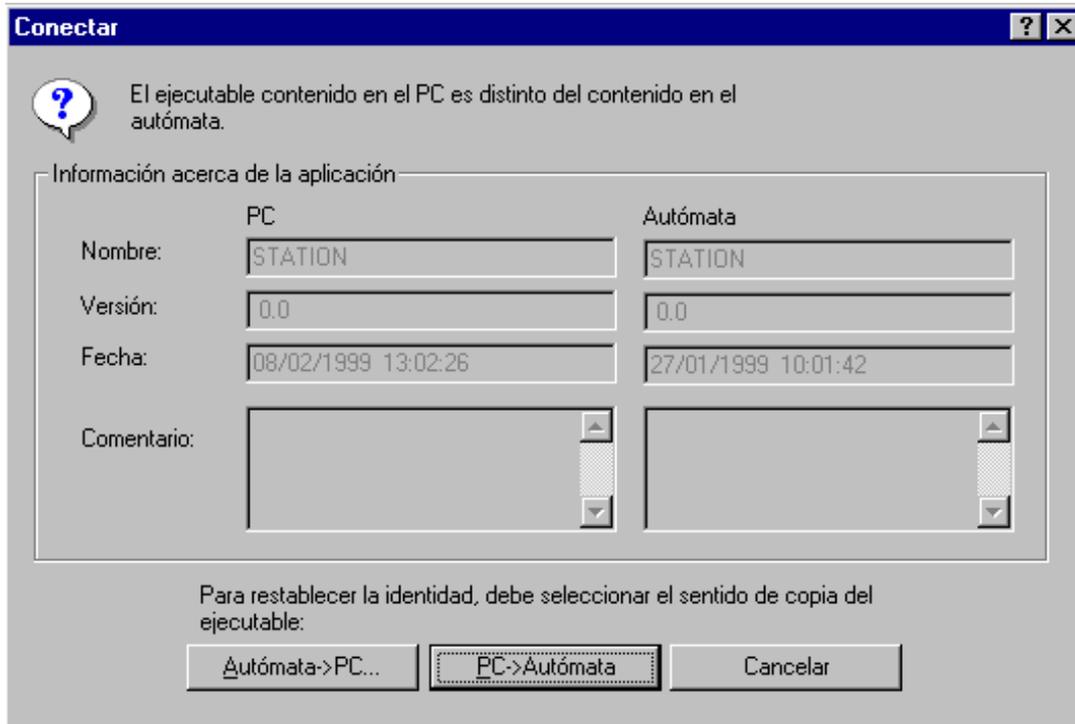
El cable de comunicación esta suelto ( en el ordenador o en el autómata)

Estáis intentando comunicar por el puerto de impresora (¡¡es el puerto hembra de 25 pines!!).

El Automata no tiene alimentación.

Verificar lo anterior y volver a intertarlo. Si todo esta correcto y sigue sin comunicar, puede ser que el puerto serie del ordenador este roto ( a veces pasa).

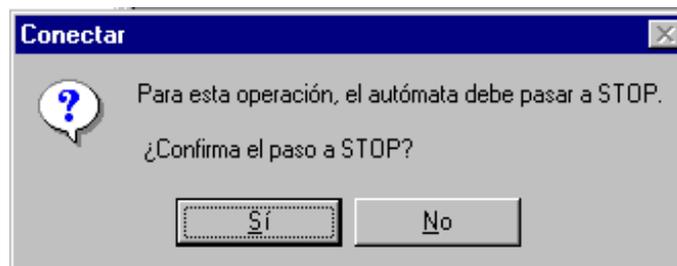
Si se logra comunicar saldrá la siguiente pantalla:



Elegir PC->Autómata.

(Si se desea cargar la aplicación que contiene el autómata en el ordenador se elegirá Autómata ->PC).

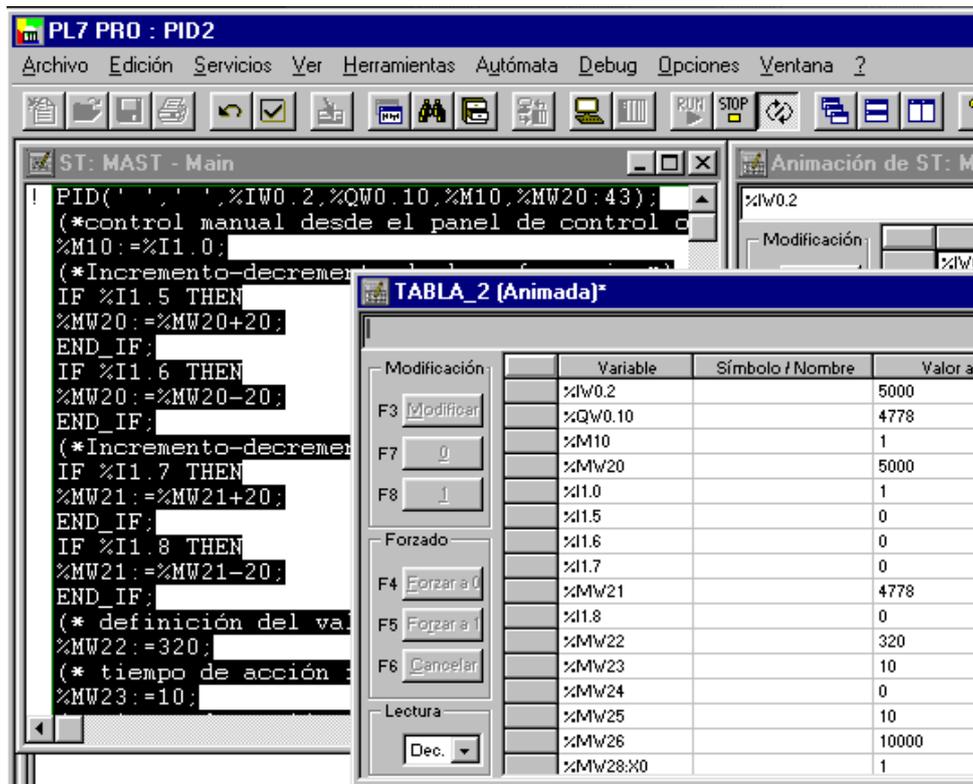
Si el autómata se encontraba ejecutando una aplicación es necesario pararlo:



Una vez transferida la aplicación poner el autómata en Run y se ejecutará el programa.

Para comprobar el funcionamiento de un programa en lenguaje literal es conveniente crear una tabla de animación. Seleccionar el código arrastrando el ratón y ir al menú Servicios- Inicialización de una tabla de animación.

Con estas ordenes se crea una tabla con los objetos utilizados en el programa.



Pulsar %I1.0 . El regulador funciona en automático. Incrementar con %I1.5 y decrementar con %I1.6 la referencia de mando y comprobar que la variable controlada %IW0.2 la sigue.

En la palabra %QW0.10 se escribe la acción del PID. Estimar la ganancia del sistema físico controlado.

Modificar el programa y hacer el tiempo de acción integral (%MW23) cero. Hacer  $K_p=100$ . Poner la referencia %MW20 a 7000 ¿Cuanto vale el error de Posición?.

Poner la referencia %MW20 a 3000 ¿Cuanto vale el error de Posición?.

Razona las respuestas.

El algoritmo sin acción integral ( $T_i = 0$ ) efectúa la operación siguiente:

$$\varepsilon_t = SP - PV, \text{ (error = referencia - medida)}$$

la acción del regulador es

$$OUT = KP [\varepsilon_t + D_t] / 100 + 5000$$

donde  $D_t$  = acción derivada,

El algoritmo con acción integral ( $T_i \neq 0$ ) efectúa la operación siguiente:

$$\text{para } \varepsilon_t = SP - PV,$$

la acción del regulador es:

$$\Delta OUT = KP [\Delta \varepsilon_t + (T_s / 10 \cdot T_i) \cdot \varepsilon_t + \Delta D_t] / 100$$

$$OUT = OUT + \Delta OUT$$

donde  $D_t$  = acción derivada.

$T_s$  es el periodo de muestreo.

## Ejercicio 2.

Introducir el siguiente programa en la Tarea Mast. (Fichero Pidspia2.stx en directorio c:\Pract7)

```
PID(",",%IW0.2,%QW0.10,%M10,%MW20:43);
(*control manual desde el spia*)
%M10:= %MW118:X0;
(*lectura de la referencia del Spia*)
%MW20:=%MW110;
(*si estamos en manual, la acción es igual a la acción manual del Spia*)
IF NOT %M10 THEN
%MW21:=%MW111;
END_IF;
(*se envia el valor de la variable controlada al Spia*)
%MW100:=%IW0.2;
(*se envia el valor de la Acción al Spia*)
%MW101:=%QW0.10;
(* definición del valor de la ganancia del regulador kp*)
%MW22:=%MW112;
(* tiempo de acción integral en décimas de segundo*)
%MW23:=%MW113;
(* tiempo de acción derivada en décimas de segundo*)
%MW24:=%MW114;
(*periodo de muestreo en centésimas de segundo*)
%MW25:=%MW115;
(* acción Máxima*)
%MW26:=10000;
%MW28:X0:=TRUE;
%MW28:X8:=TRUE;
```

A continuación transferir la aplicación al autómata.

Una vez transferida la aplicación poner el autómata en Run  y se ejecutará el programa.

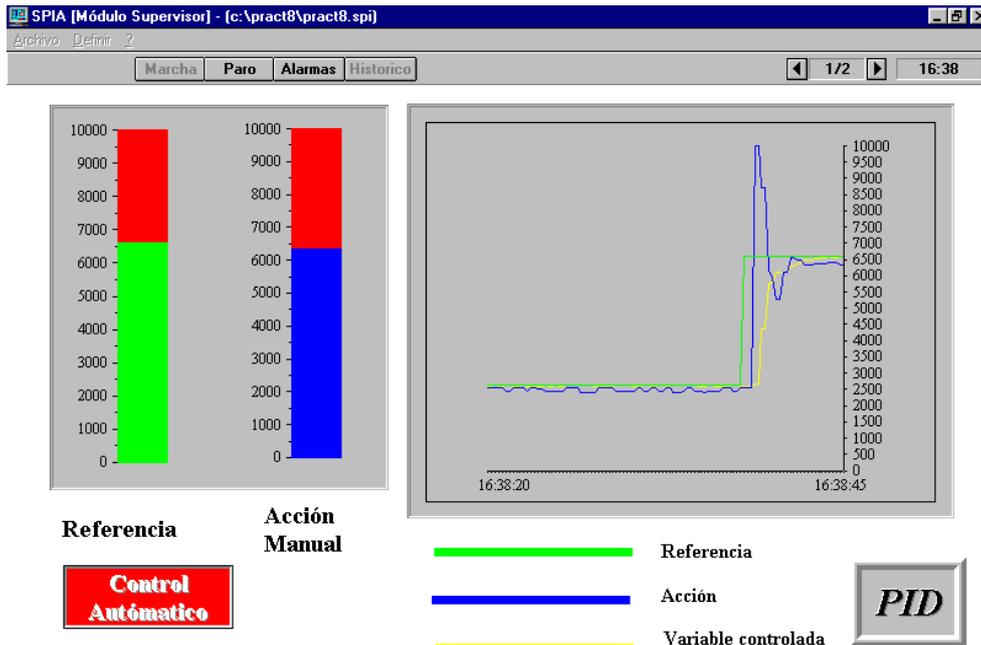


Desconectar el autómata

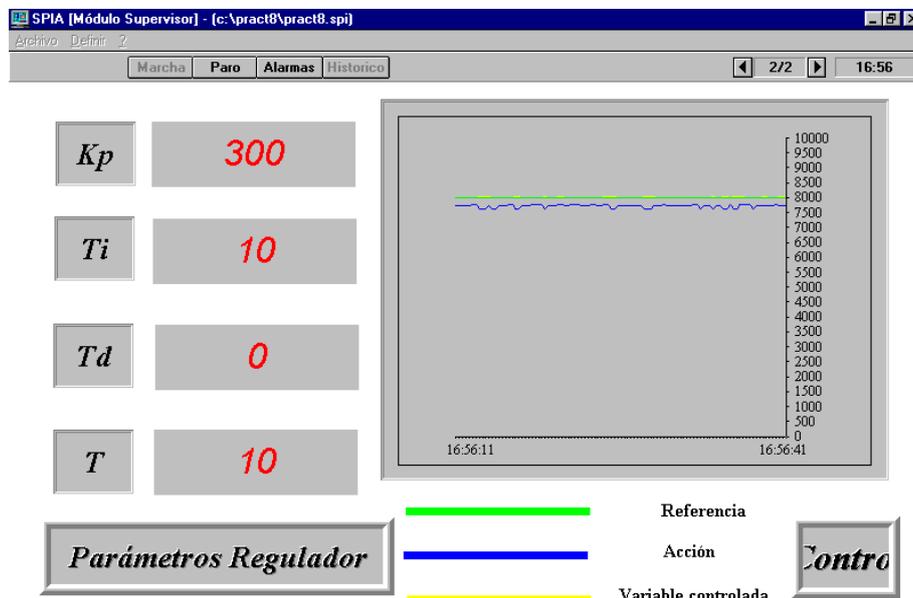
Ejecutar el programa Spia\_s.exe se encuentra en el directorio C:\Spia\_s.



Abrir el proyecto Prac7.spi se encuentra en el subdirectorio c:\pract7



La pantalla es un panel de control por Computador. El operador dispone de:  
 Barra gráfica de mando para introducir la referencia del control en estado automático (mediante regulador PID del automático).  
 Barra gráfica de mando para introducir la acción manual. El valor de la acción manual se vuelca directamente en el canal de salida analógica del automático en el caso de control en estado manual.  
 Interruptor para cambiar de estado de control manual a control automático.  
 Ventana para visualizar las curvas de referencia, acción y variable controlada.



Parámetros del regulador:

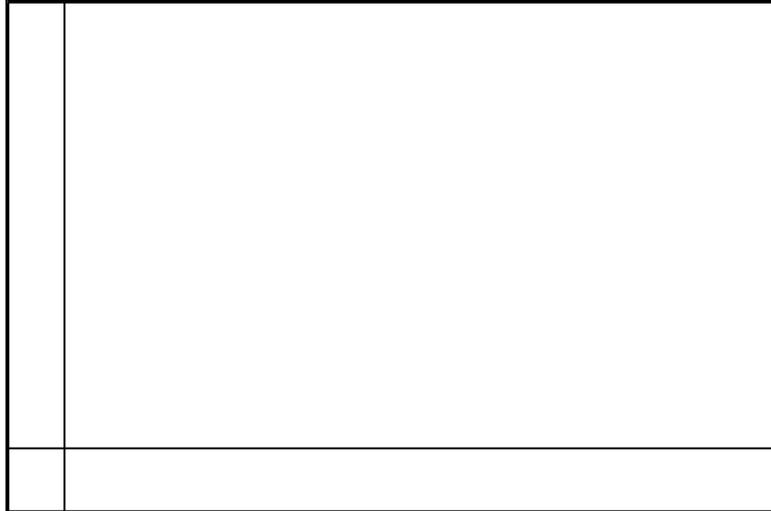
- $K_p$ : Es el valor de la ganancia del regulador.
- $T_i$ : Es el valor de la constante del tiempo integral.
- $T_d$ : Es el valor de la constante del tiempo derivativo.

T: Es el periodo de muestreo.

Dar a Marcha

Hacer  $K_p=300$  (3),  $T_i=10$ (1 segundo),  $T_d=0$ ,  $T=10$  (0.1segundos).

Activar el control Automático. Introducir un escalón de 0 a 4000.



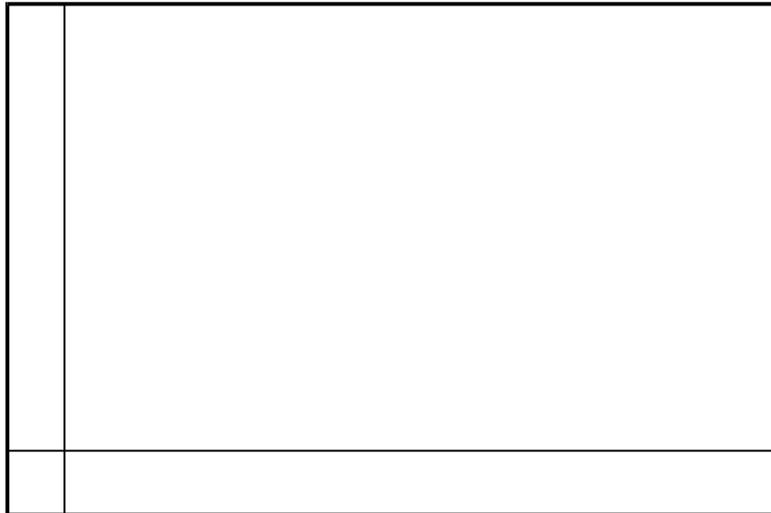
*Gráfica 1 Gráfica de la referencia, de la acción y de la variable controlada.*

Hacer  $T_i=0$ , el regulador ahora es proporcional puro. Situar la referencia en 3000.

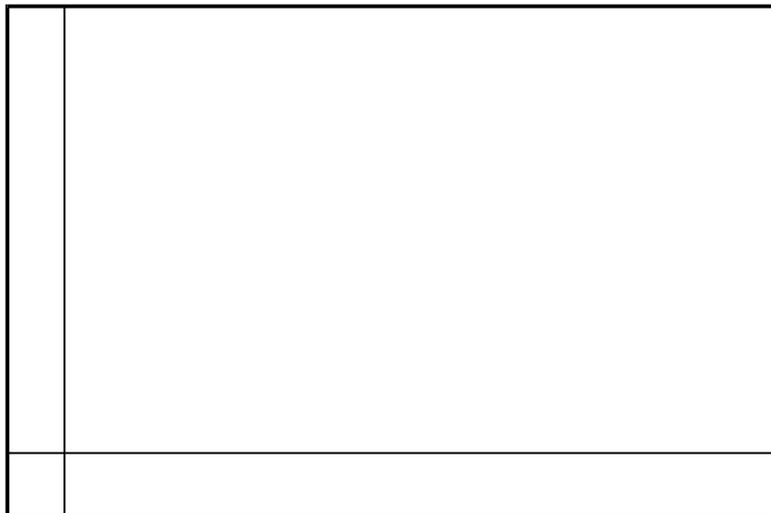


*Gráfica 2 Gráfica de la referencia, de la acción y de la variable controlada (referencia 3000).*

*Indica en la gráfica el valor del error de posición.*



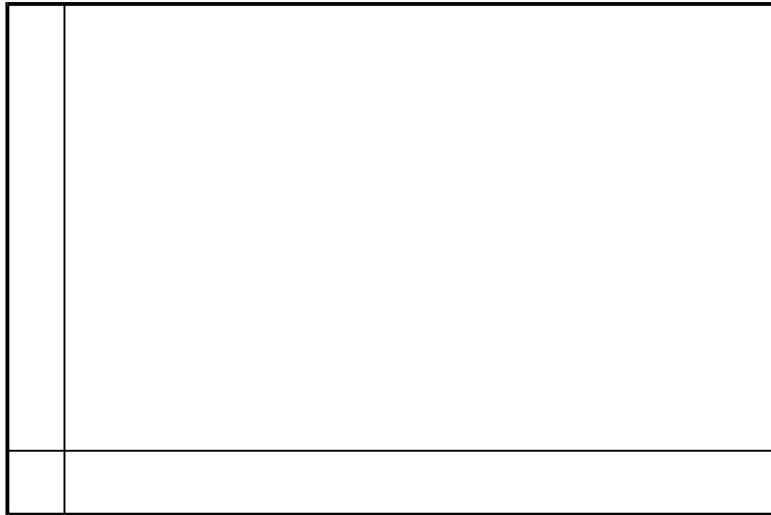
*Gráfica 3 Gráfica de la referencia, de la acción y de la variable controlada (referencia 7000).  
Indica en la gráfica el valor del error de posición.*



*Gráfica 4 Gráfica de la referencia, de la acción y de la variable controlada (referencia 5200).  
Indica en la gráfica el valor del error de posición.*

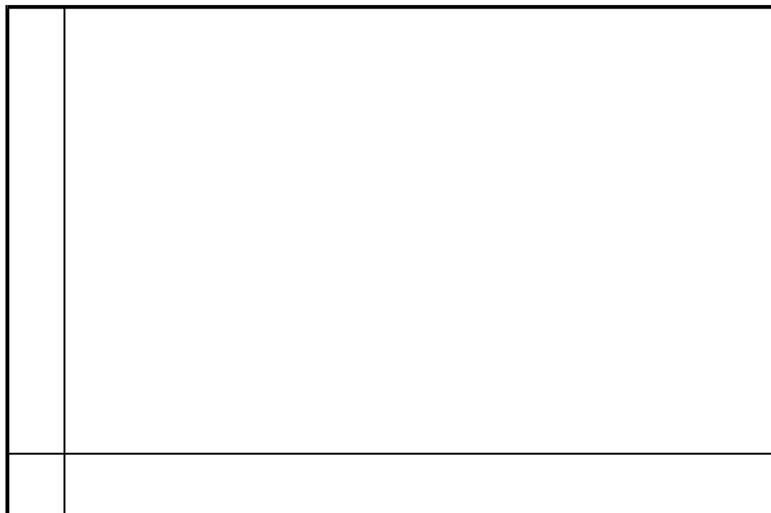
Razona los valores del error de posición en los tres casos anteriores

Hacer  $K_p=300$  (3),  $T_i=10$  (1 segundo),  $T_d=100$  (10 segundos),  $T=10$  (0.1 segundos).  
Activar el control Automático. Introducir un escalón de 0 a 6000.



*Gráfica 5 Gráfica de la referencia, de la acción y de la variable controlada*

Hacer  $K_p=300$  (3),  $T_i=10$  (1 segundo),  $T_d=20$  (2 segundos),  $T=10$  (0.1 segundos).  
Activar el control Automático. Introducir un escalón de 0 a 6000.  
¿Qué función de transferencia se tiene en bucle cerrado?.



*Gráfica 6 Gráfica de la referencia, de la acción y de la variable controlada*

### Ejercicio 3.

Introducir el siguiente programa en la Tarea Mast. Fichero PidSpia3.stx en directorio c:\Pract7

```
PID(",",%IW0.2,%QW0.10,%M10,%MW20:43);
PWM(%QW0.10,%Q2.1,%MW70:5);
%MW70:=%MW25*100;
(*control manual desde el Spia*)
%M10:=%MW118:X0;
(*lectura de la referencia del Spia*)
%MW20:=%MW110;
(*si estamos en manual, la acción es igual a la acción manual del Spia*)
IF NOT %M10 THEN
%MW21:=%MW111;
END_IF;
(*se envia el valor de la variable controlada al Spia*)
%MW100:=%IW0.2;
(*truco para ver la salida PWM en el spia*)
(*pulsar %i1.2*)
IF %I1.2 THEN
    IF %Q2.1 THEN %MW101:=5000;END_IF;
    IF NOT %Q2.1 THEN %MW101:=0;END_IF;
    END_IF;
(*se envia el valor de la Acción al Spia*)
IF NOT %I1.2 THEN
%MW101:=%QW0.10;
END_IF;
(* definición del valor de la ganancia del regulador kp*)
%MW22:=%MW112;
(* tiempo de acción integral en décimas de segundo*)
%MW23:=%MW113;
(* tiempo de acción derivada en décimas de segundo*)
%MW24:=%MW114;
(*periodo de muestreo en centésimas de segundo*)
%MW25:=%MW115;
(* acción Maxima*)
%MW26:=10000;
%MW28:X0:=TRUE;
%MW28:X8:=TRUE;
```

A continuación transferir la aplicación al autómata.

Una vez transferida la aplicación poner el autómata en Run  y se ejecutará el programa.



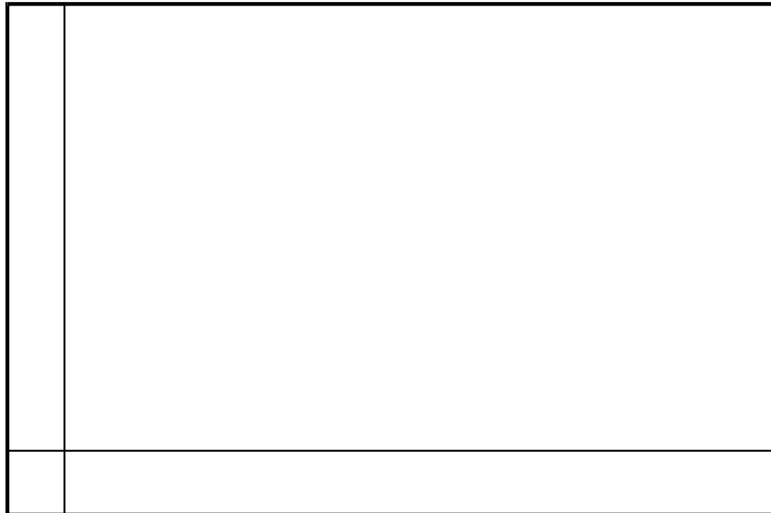
Desconectar el autómata

Volver al Spia y poner en marcha .

Hacer  $K_p=300$  (3),  $T_i=10$  (1 segundo),  $T_d=0$  ,  $T=10$  (0.1segundos).

Pulsar %1.2 para visualizar la salida PWM.

Activar el control Automático. Introducir una referencia de 5000.



*Gráfica 7 Gráfica de la referencia, de la acción PWM y de la variable controlada*

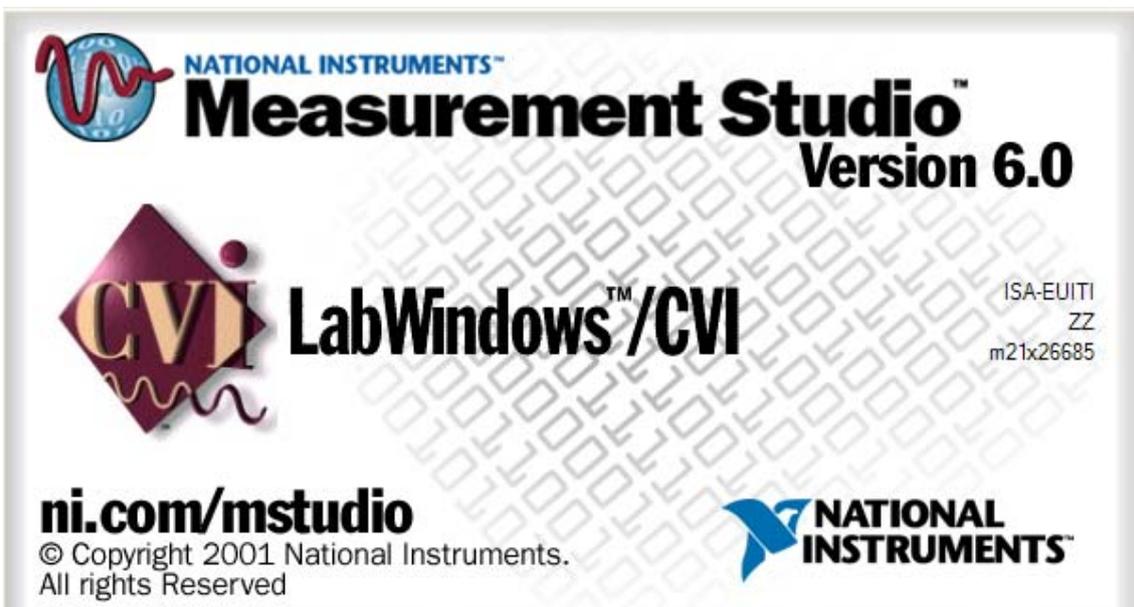
Comprobar que cuanto más alta es la referencia más tiempo permanece activa la salida PWM.

---

## *Práctica 5*

# *Implementación programada de reguladores. Introducción al entorno de desarrollo CVI*

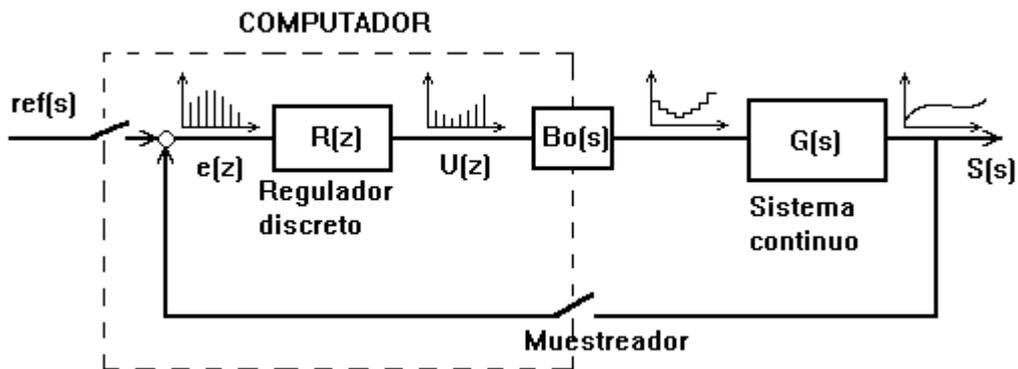
---



## OBJETIVOS

En esta práctica se estudia el comportamiento (análisis) de los sistemas continuos controlados mediante reguladores de tiempo discreto (Algoritmo de control ejecutándose en un computador). También se diseñarán reguladores de tiempo discreto que se implementarán sobre un computador. Las aplicaciones serán desarrolladas utilizando un entorno de desarrollo comercial para esta finalidad (CVI de *National Instruments*). Se adjunta manual de usuario del mismo.

El sistema de la figura representa esquemáticamente el control de procesos continuos por medio del computador:

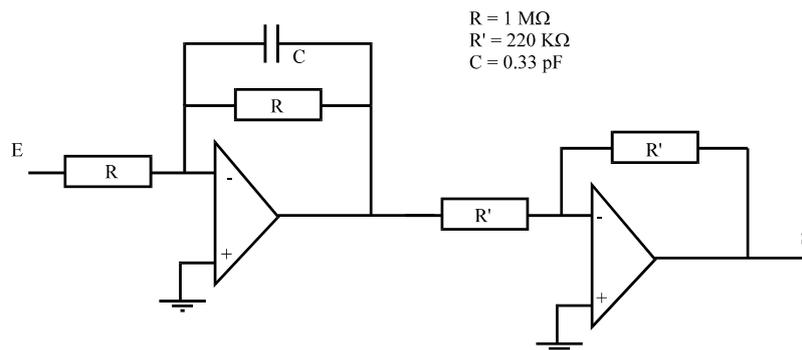


El sistema continuo a controlar está realizado por medio de circuitos basados en el amplificador operacional. Su función de transferencia puede ser de primer o segundo orden en función de que el interruptor correspondiente esté o no abierto:

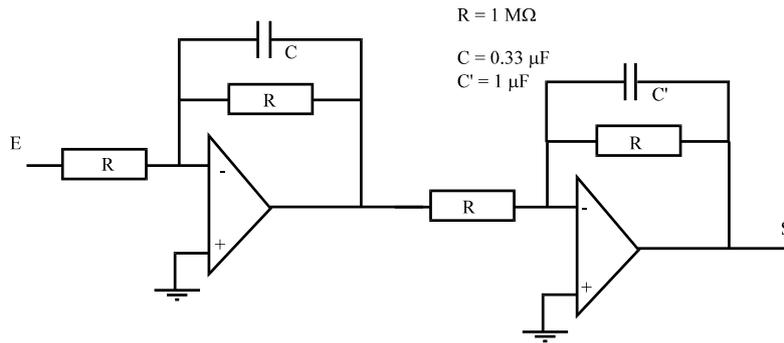
$$G(s) = \frac{1}{1 + 0.33s}$$

o bien

$$G(s) = \frac{1}{(1 + 0.33s)(1 + s)}$$



Esquema circuital del sistema de primero orden.



Esquema circuital del sistema de segundo orden.

## BREVE INTRODUCCION A LA HERRAMIENTA

CVI (*C for Virtual Instrumentation*) es un entorno de desarrollo de *National Instruments*, (<http://www.ni.com>) específicamente pensado para la generación de aplicaciones en las que, de una forma u otra intervienen instrumentos de medida y la correspondiente interfase con el computador. En nuestro ámbito, las aplicaciones de control por computador utilizan instrumentos de medida conectados a tarjetas de adquisición de datos, por lo que CVI es susceptible de ser utilizado con el fin de diseñar de manera sencilla una aplicación de control que contendrá, al menos:

1. Un interfaz de usuario con una fuerte componente visual. Dicha interfaz contemplará todos aquellos elementos de representación gráfica o numérica, además de aquellos elementos, tanto de parametrizado de la aplicación en cualquiera de sus facetas (algoritmo de control, la propia interfaz gráfica...), como de explotación (arranque, parada, modo de operación...).
2. El algoritmo de servocontrol, que constituirá el núcleo de la aplicación
3. El manejador del dispositivo específico de E/S, proporcionado en su mayor parte por el fabricante de la tarjeta de adquisición.

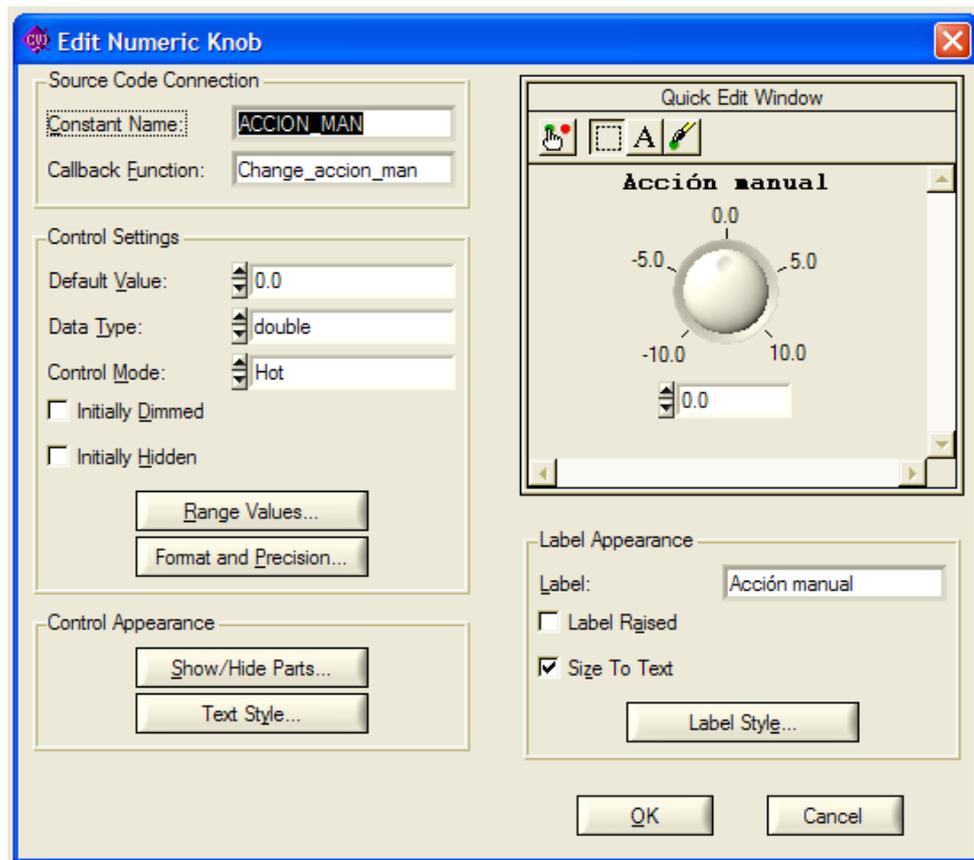
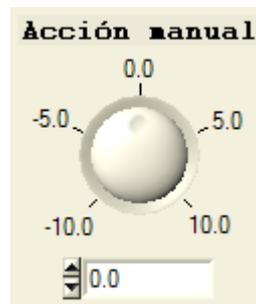
De los componentes anteriormente enumerados, la GUI conllevaría el mayor esfuerzo de programación si únicamente se contara con un lenguaje de programación de propósito general, aún cuando éste tuviese una orientación "visual". Este motivo supone una razón de suficiente peso como para justificar la existencia de herramientas como la que nos ocupa, facilitadoras de esta etapa del diseño. El proceso de diseño de este tipo de aplicaciones comprende al menos 2 pasos:

- La edición de la GUI
- La implementación del código asociado al programa principal, así como a los eventos generados desde la GUI

Además, en nuestro caso deberemos integrar en la aplicación aquellas funciones de E/S asociadas al hardware disponible. En las siguientes secciones nos ocuparemos de cada una de estas cuestiones.

### Edición de la GUI

En primer lugar la creación de la GUI (Graphic User Interface) por medio de un editor específicamente pensado para tal fin. Los objetos creados con esta herramienta son referenciados en un archivo de cabecera de lenguaje C generado automáticamente, en el que para cada objeto se contempla un identificador elegido por el usuario en la fase de edición de la GUI; además en este fichero están prototipadas aquellas funciones (*Callback functions*) asociadas a la actuación del usuario sobre el elemento (eventos de ratón u otros...). Las figuras siguientes muestran respectivamente, un elemento de la GUI, el cuadro de diálogo que permite su parametrizado durante la edición y la sección del fichero de cabecera generado automáticamente en la que se muestra toda referencia al mencionado elemento.



```
/* LabWindows/CVI User Interface Resource (UIR) Include File */
/* Copyright (c) National Instruments 2005. All Rights Reserved. */
/*
/* WARNING: Do not add to, delete from, or otherwise modify the contents
/* of this include file.
/*****
*****/

#include <userint.h>

#ifdef __cplusplus
extern "C" {
#endif

    /* Panels and Controls: */

    .....
#define PANEL_ACCION_MAN 3 /* callback function: Change_accion_man */
    .....

    /* Callback Prototypes: */

int CVICALLBACK Change_accion_man(int panel, int control, int event, void
*callbackData, int eventData1, int eventData2);
    .....

#ifdef __cplusplus
}
#endif
#endif
```

Para la realización de esta práctica se ha preparado una interfaz de usuario que no será necesario modificar (fichero p9.uir), así como el fichero de enlace (p9.h) que ha sido generado automáticamente al guardar el diseño.

### **Implementación del código. Generación automática con CodeBuilder**

Una vez generada la GUI, debe llevarse a cabo la implementación del código correspondiente al programa principal y a los diferentes eventos asociados a los elementos, tanto de la GUI, como eventos asociados a relojes (*Ticks*). Esta fase del diseño puede verse facilitada desde CVI por medio de *Code Builder*: un generador de código que proporciona una primera versión del código fuente en la que se incluye un esqueleto de aplicación que contendrá, tanto las *Callback functions* como el programa principal. Al programador le resta especificar con detalle cuáles son las acciones que se generarán ante la llegada de los sucesivos eventos. El siguiente cuadro muestra, a modo de ejemplo, el fragmento de código asociado a los eventos generados al manipular el elemento que nos ha ocupado hasta ahora:

```
int CVICALLBACK Change_accion_man (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (panelHandle, PANEL_ACCION_MAN, &u_man);
            break;
        case EVENT_RIGHT_CLICK:
            MessagePopup ("AYUDA", "Varía la acción manual utilizada");
            break;
    }
    return 0;
}
```

### **Software de E/S**

En las aplicaciones de instrumentación y control, el software de manejo de los controladores específicos de E/S forma parte de la aplicación. En nuestro caso va a ser utilizada una tarjeta de E/S multipropósito de ADLINK Technology Inc. (<http://www.adlinktech.com>). De entre sus características, cabe destacar:

- Bus PCI
- 16 entradas + 12 salidas analógicas
- Precisión: 12 bits
- 16 entradas + 16 salidas binarias
- 3 relojes independientes
- 3 diferentes modos de disparo
- Posibilidad de transferencia DMA
- Tasas de muestreo de hasta 110 KHz

El software de manejo reside en una DLL cuyas funciones están prototipadas en el correspondiente archivo de cabecera. Para facilitar el manejo de esta tarjeta se han creado 2 ficheros (tarjeta\_E\_S.h y tarjeta\_E\_S.c) exclusivamente con las funciones a utilizar (inicialización, lectura/escritura analógica y liberación de uso) Dichas funciones evitan al usuario la parametrización detallada de la tarjeta, así como ciertos detalles de la implementación de la E/S que no son merecedoras de atención en nuestro contexto. El cuadro siguiente muestra el contenido del fichero tarjeta\_E\_S.h

```
/*
*****
***** INTERFAZ BASICA TARJETA ACL-9112 *****
*****
***** Autor: ANTONIO ROMEO TELLO *****
***** Paquete: TARJETA_E_S *****
***** Lenguaje: ansi C *****
***** Mision: Proporciona herramientas para *****
***** realizar la entradas/salida *****
***** Analógica *****
***** Modo disparo: por software. *****
***** Sincronización: por encuesta. *****
***** Fecha: ABRIL- 2005 *****
*****
void init_9112 (void);
/* Inicializa la tarjeta para trabajar con las entradas analógicas */
/* en modo de disparo por software y sincronización por encuesta. */
/* Debe ser llamado una vez al comienzo del uso de la tarjeta */

void release_9112 (void);
/* Libera la tarjeta. Debe ser llamado una vez al fnalizar su uso */

double analog_input_9112 (int analog_channel);
/* canales de entrada analógicos: 0..15 */

void analog_output_9112 (double data);
*/
```

### **Programa principal**

El fichero generado de forma automática con *CodeBuilder* contiene, además de la atención a los eventos generados desde la GUI (*Callback functions*), el programa principal, cuya misión fundamental es la carga de la GUI y la ejecución del entorno que se encarga de generar y capturar los eventos asociados a los elementos de la misma. El cuadro siguiente muestra el programa principal generado en primera instancia con *CodeBuilder* y modificado posteriormente con la inclusión posterior de las funciones de inicialización y liberación de la tarjeta de E/S

```
/* Programa principal: carga la pantalla y ejecuta la interfaz de usuario */
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "p9.uir", PANEL)) < 0)
        return -1;
    init_9112 ();
    DisplayPanel (panelHandle);
    RunUserInterface ();
    DiscardPanel (panelHandle);
    analog_output_9112 (0);
    release_9112 ();
    return 0;
}
```

### **La rutina de servicio del evento asociado al reloj**

En toda aplicación periódica (en nuestro caso, las tareas de control lo son) la rutina que se ejecuta cada período de muestreo deberá formar parte de la atención al evento *Timer\_tick* de un reloj (*timer*) que el usuario deberá haber incluido en la GUI. Este reloj no es un elemento visible en la fase de ejecución de la aplicación, pero permite la generación de eventos temporizados que, a la postre, permitirán desencadenar las operaciones conducentes al cálculo y posterior aplicación de la acción de control.

```
/* Timer_tick: ejecuta el algoritmo de control cada vez que dispara el tick del timer */
/* Juega un papel equivalente al de la rutina de servicio de interrupción de reloj */
int CVICALLBACK Timer_tick (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK:
            {
                sal=analog_input_9112 (0);
                error=ref-sal;
                integral_error=integral_error+T*error;
            }
    }
    return 0;
}
```

Más info.: en la guía rápida de CVI (<http://www.ni.com/pdf/manuals/323552b.pdf>)

## **Estudio teórico**

**ET1-**Utilizando el método de emulación programada de reguladores continuos, diseñar el regulador proporcional para el sistema de primer orden (¡atención al interruptor!) que haga el error de posición menor que 0.2. Elegir un valor para el período de muestreo, discretizar el regulador y extraer la ecuación en diferencias de control.

**ET2-**Con el valor de K obtenido en el apartado anterior, calcula el valor del período de muestreo por encima del cual el sistema se inestabiliza.

## Estudio práctico

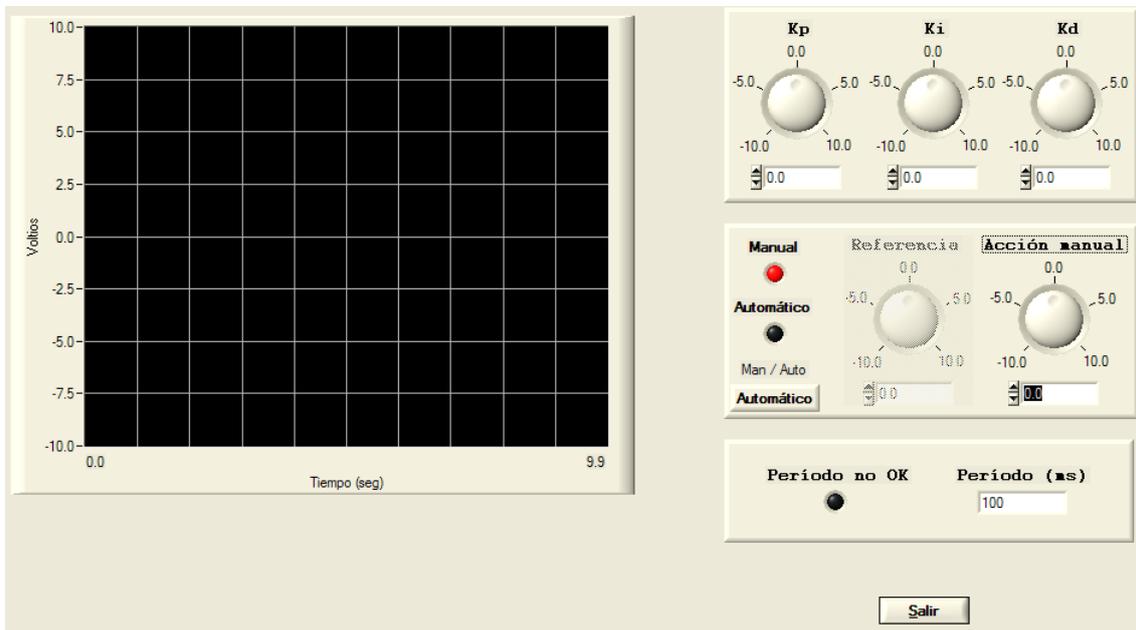
Abrid una carpeta nueva (al finalizar la práctica deberéis borrarla) en **d:\usuarios\**; sobre ella copiaréis los archivos contenidos en la carpeta correspondiente del servidor de prácticas departamental **\\lis1\practicasyreg\_aut\_electricos\prac5**.

Abrid el fichero de proyecto de CVI denominado **p5.prj**. Una vez abierto el proyecto deberemos añadirle (Edit ->add file to Project...) el fichero de librería **p5.dsk.lib** del la tarjeta AdLink, que se encuentra en **c:\adlink\lib** en el que se han implementado las cuestiones referentes a la representación gráfica e inicialización.

**EP1**-Completar la implementación del regulador proporcional obtenido. El período de muestreo se modificará por medio de la correspondiente entrada dispuesta a tales efectos. Nótese que una modificación del mismo provocará la inicialización de la representación gráfica. Comprobar la estabilidad del sistema controlado con distintos períodos de muestreo.

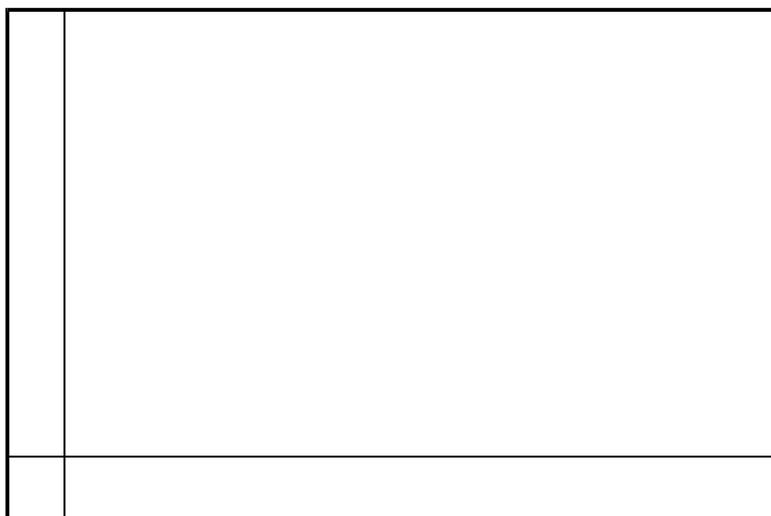
Tómese como punto de partida el fichero p9.c del proyecto que ha sido abierto con anterioridad, en el que se han implementado las cuestiones referentes a la representación gráfica e inicialización.

La GUI es la que se muestra en la figura siguiente





*Gráfica 1 Representar la referencia, la acción y la variable controlada ante una entrada escalón unitario.*



*Gráfica 2 Representar la referencia, la acción y la variable controlada en el instante en que se inestabiliza debido al aumento del periodo de muestreo.*

## Código fuente a completar:

```
int CVICALLBACK Timer_tick (int panel, int control, int event,
                             void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK:
            sal=analog_input_9112 (0);
            if (automatico==1)
            {
                u_auto=.....
                accion=u_auto;
            }
            else
            {
                accion=u_man;
            }
            analog_output_9112 (accion);
            vector_repres[0]=ref;
            vector_repres[1]=sal;
            vector_repres[2]=accion;
            PlotStripChart (panelHandle, PANEL_GRAFICA, vector_repres, 3, 0,
                            0, VAL_DOUBLE);
            break;
    }
    return 0;
}
```

## **Estudio teórico**

**ET3-** Calcular el mínimo tiempo de respuesta que se puede conseguir con un regulador integrador puro (K/s). Utiliza el mismo método de síntesis que en el caso anterior.

¿Qué sucede si se aumenta la ganancia del regulador?

## **Estudio práctico**

**EP3-**Implementar de forma programada el regulador Integral puro obtenido en Observar la influencia de la saturación de la salida.



*Gráfica 3 Representar la referencia y la variable controlada ante una entrada escalón.*



*Gráfica 4 Representar la acción ante una entrada escalón.*

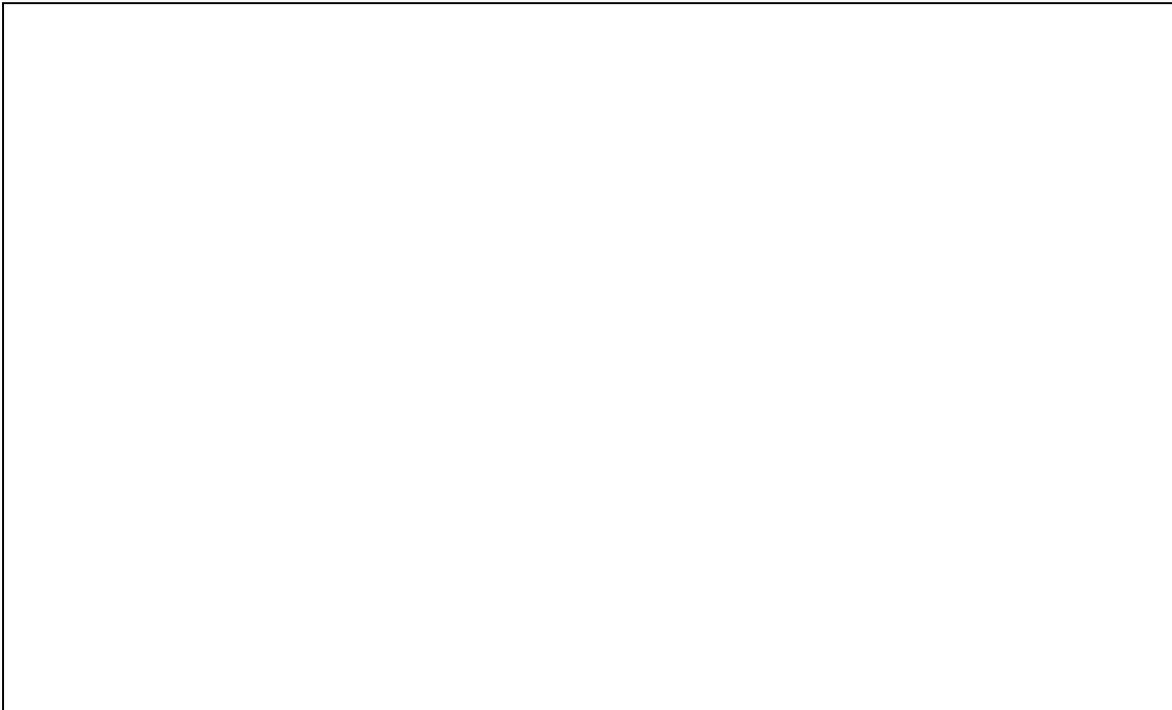
## Código fuente a completar

```
int CVICALLBACK Timer_tick (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK:
            sal=analog_input_9112 (0);
            if (automatico==1)
            {
                u_auto=.....
                accion=u_auto;
            }
            else
            {
                accion=u_man;
            }
            analog_output_9112 (accion);
            vector_repres[0]=ref;
            vector_repres[1]=sal;
            vector_repres[2]=accion;
            PlotStripChart (panelHandle, PANEL_GRAFICA, vector_repres, 3, 0,
                0, VAL_DOUBLE);
            break;
        }
    return 0;
}
```

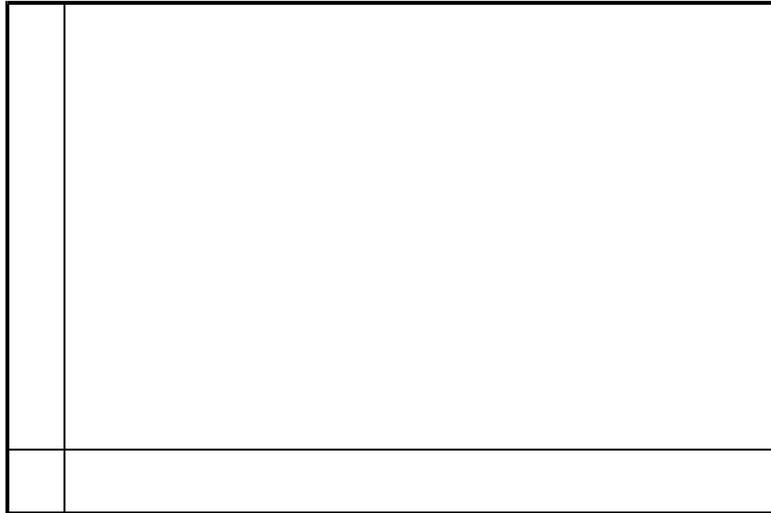
**ET4-** Utilizando el mismo método de síntesis que en los apartados anteriores, pero esta vez sobre el sistema de segundo orden (¡Atención al interruptor!), diseñar un regulador de manera que se consigan las siguientes especificaciones:

$$e_v < 0.1$$
$$t_r < 2.5 \text{ segundos}$$

Elegir un valor para el período de muestreo, discretizar el regulador y extraer la ecuación en diferencias de control.



**EP4-** Implementar de forma programada el regulador Proporcional Integral obtenido en ET4. Comprobar el efecto que el paso de manual a automático tiene sobre la acción y la variable controlada. Para ello llevar el sistema en modo manual al valor de referencia (ajustarla previamente), para después conmutar al modo automático.



*Gráfica 5 Representar la referencia y la variable controlada antes y después del paso a modo automático.*



*Gráfica 6 Representar la acción antes y después del paso a modo automático.*

## Código fuente a completar

```
int CVICALLBACK Timer_tick (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK:
            sal=analog_input_9112 (0);
            if (automatico==1)
            {
                u_auto=.....
                accion=u_auto;
            }
            else
            {
                accion=u_man;
            }
            analog_output_9112 (accion);
            vector_repres[0]=ref;
            vector_repres[1]=sal;
            vector_repres[2]=accion;
            PlotStripChart (panelHandle, PANEL_GRAFICA, vector_repres, 3, 0,
                0, VAL_DOUBLE);
            break;
        }
    return 0;
}
```

## **Estudio práctico**

**EP5**-modificar el regulador anterior de forma que se habilite el paso adecuado entre los modos manual y automático. Utilícese para ello el método de inicialización de la integral. Reproducir condiciones similares a las solicitadas en EP4 para evaluar la transferencia entre los modos manual y automático.

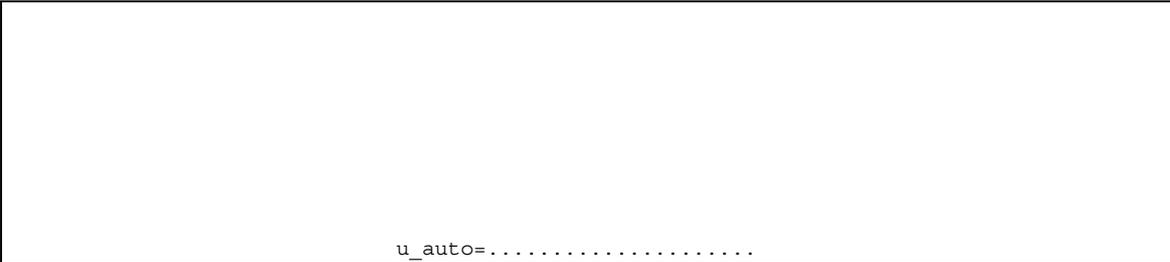


*Gráfica 7 Representar la referencia y la variable controlada antes y después del paso a modo automático..*



*Gráfica 8 Representar la acción antes y después del paso a modo automático.*

## Código fuente a completar

```
int CVICALLBACK Timer_tick (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_TIMER_TICK:
            sal=analog_input_9112 (0);
            if (automatico==1)
            {
                
                u_auto=.....
                accion=u_auto;
            }
            else
            {
                
                accion=u_man;
            }
            analog_output_9112 (accion);
            vector_repres[0]=ref;
            vector_repres[1]=sal;
            vector_repres[2]=accion;
            PlotStripChart (panelHandle, PANEL_GRAFICA, vector_repres, 3, 0,
                0, VAL_DOUBLE);
            break;
        }
    return 0;
}
```