

**GUIA INTRODUCTORIA AL USUARIO DEL SIMULADOR XCOS DE SCILAB
APLICADO A LA SIMULACION DE SISTEMAS DE CONTROL**

**EDINSON BAUTISTA QUIROGA
C.C 91.538.733 BMGA**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA
DEPARTAMENTO DE CONTROL
PROGRAMA INGENIERIA ELECTRONICA
BUCARAMANGA
2012**

**GUIA INTRODUCTORIA AL SIMULADOR XCOS DE SCILAB APLICADO A LA
SIMULACION DE SISTEMAS DE CONTROL**

**EDINSON BAUTISTA QUIROGA
C.C 91.538.733 BMGA**

Trabajo de grado para optar el título de Ingeniero Electrónico

**Director
EDGAR BARRIOS URUEÑA
Ing. MsC Ingeniera Eléctrica**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA
DEPARTAMENTO DE CONTROL
PROGRAMA INGENIERIA ELECTRONICA
BUCARAMANGA
2012**

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bucaramanga, Mayo 2012

Quiero agradece a Dios por su infinita sabiduría e iluminar cada uno de mis pasos y confió plenamente que las cosas en la vida nos pasan por una razón divina.

Eternamente agradecido con mi padre Alirio Bautista y mi madre Mariela Quiroga que con su apoyo incondicional y sus consejos hicieron posible este logro más en mi vida.

A mi esposa por creer en mí y por su infinito apoyo en esta lucha de mi vida, por estar conmigo en los momentos más difíciles y por estar siempre detrás de mí motivándome para hacer posible esta meta

Finalmente doy gracias por contar con el apoyo, conocimiento y orientación del Ingeniero Edgar Barrios durante el desarrollo de mi tesis.

EDINSON BAUTISTA QUIROGA

CONTENIDO

ABSTRACT OF THESIS PROJECT	11
INTRODUCCIÓN	12
1. OBJETIVOS	13
2. Marco teórico.....	14
2.1 Software de simulación.....	14
2.2 Software licenciado.....	15
2.3 Diagrama de bloques.....	15
3. Metodología.....	16
1) Recopilación y análisis de información.....	16
2) Desarrollo de ejercicios y prácticas del laboratorio de control.....	16
3) Realización de la guía introductoria de Xcos de Scilab	17
4. Iniciando con Scilab	18
4.1 Consola.....	18
4.2 Editor	20
4.3 Ejemplo.....	21
5. Scilab/Xcos.....	23
6. Bloques principales de Xcos	26
6.1 Derivada	26
6.2 Integrador	27
6.3 Continuous transfer function	28
6.4 PID regulator.....	28
6.5 Continuous state-space system CLSS.....	30
6.5.1 Ejemplo.....	30
6.6 Electrical capacitor.....	32
6.7 Electrical diode	32
6.8 Electrical resistor	33
6.9 Electrical inductor	33
6.10 Ground (zero potential reference).....	34
6.11 NPN transistor	34
6.12 Electrical DC voltage source.....	35
6.13 Activation clock	35
6.14 Constant	35

6.15 Ramp	35
6.16 Step Function.....	36
7. Ejercicios	37
7.1 Ejemplo configuración muestreo CSCOPE.....	37
7.2 Practica de laboratorio. Experiencia1. Familiarización con el simulador SIMULINK de MATLAB,.....	39
7.3 Control a lazo cerrado de un PID.....	42
8. Desarrollo guías laboratorio	44
8.1 EXPERIENCIA 1: Familiarización con el simulador SIMULINK de MATLAB	44
8.2 EXPERIENCIA 2: Comportamiento de Sistemas de Control	49
8.3 EXPERIENCIA 2: Estado Transitorio y Efecto del Período de Muestreo sobre la Estabilidad de Sistemas Discretos	58
8.4 EXPERIENCIA 3: Introducción a Controladores PID de Sistemas Discretos	62
CONCLUSIONES.....	72
Bibliografía	73

LISTA DE FIGURAS

Figura 1 Consola scilab	18
Figura 2 Navegador de ayuda	19
Figura 3 Editor Scilab	20
Figura 4 Scilab	21
Figura 5 Controlador PD	22
Figura 6 Ventana principal Scilab.....	23
Figura 7 Ventana principal Xcos.....	24
Figura 8 Parámetros de simulación	24
Figura 9 Librería Xcos	25
Figura 10 Bloques principales Xcos	26
Figura 11 Propiedades integrador	27
Figura 12 Transfer function	28
Figura 13 Control PID.....	29
Figura 14 Propiedades PID	29
Figura 15 Propiedades variable de estados	30
Figura 16 Ejemplo variable de estado	31
Figura 17 Propiedades capacitor.....	32
Figura 18 Cuadro de diodo.....	33
Figura 19 Propiedades de resistencia	33
Figura 20 Propiedades bobina	34
Figura 21 Propiedades transistor NPN	34
Figura 22 Activation clock	35
Figura 23 Ramp.....	36
Figura 24 Generador senoidal.....	37
Figura 25 Periodo muestreo CSCOPE	37
Figura26 Esquemas de simulación	39
Figura 27 Inicio unión de bloques.....	39
Figura 28 Esquemático de simulación.....	40
Figura 29Escalón	40
Figura 30 CLR.....	40
Figura 31 CSCOPE	41
Figura 32 Configurar parámetros	41

Figura 33 Simulación practica 1	42
Figura 34 Control PID.....	42
Figura 35 Respuesta control PID	43
Figura 36 Resultado PID 2	43
Figura 37Esquema simulación práctica 1	44
Figura 38 Esquemático practica 1.3	47
Figura 39 Esquemático simulación práctica 2	50
Figura 40 Esquemático simulación práctica 2.2	52
Figura 41 Esquema simulación practica 2.....	58
Figura 42 Esquema simulación practica 2.2.....	61
Figura 43 Esquemáticos simulación practica 3	62
Figura 44 Esquema de simulación práctica 3.....	64

INDICE DE TABLAS

Tabla 1 Comandos	20
Tabla 2 periodo de muestro.....	38
Tabla 3 Respuesta de los sistemas vs tiempo	46
Tabla 4 Respuesta de los sistemas vs tiempo	48
Tabla 5 Porcentaje de Overshoot (PO) del sistema de segundo orden vs el Coeficiente de Amortiguación (ζ).....	52
Tabla 6 Errores de Posición, Velocidad y Aceleración vs Ganancia K.....	57
Tabla 7 Porcentaje de Overshoot (PO) vs Período de Muestreo	60
Tabla 8 Muestreo de inestabilidad.....	61
Tabla 9 Porcentaje de overshoot y Errores de posición y velocidad del sistema sin controlador digital	63
Tabla 10 Porcentaje de overshoot y Errores de posición y velocidad del sistema con P	66
Tabla 11 Porcentaje de overshoot, Tiempo de asentamiento y Errores de posición y de velocidad del sistema con PI	69
Tabla 12 Porcentaje de overshoot, Tiempo de asentamiento y Errores de posición y de velocidad del sistema	71

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: **GUIA INTRODUCTORIA AL SIMULADOR XCOS DE SCILAB
APLICADO A LA SIMULACION DE SISTEMAS DE
CONTROL**

AUTORES: **EDINSON BAUTISTA QUIROGA**

FACULTAD: **INGENIERÍA ELECTRÓNICA**

DIRECTOR: **EDGAR BARRIOS URUEÑA**

RESUMEN

En este proyecto se desarrolla una guía introductoria para el manejo de la herramienta Xcos del software Scilab que le permite al lector encontrar información suficiente para aprender a manejar Xcos. El Xcos es una herramienta de simulación de sistemas de control análogo y discreto. Para facilitar la comprensión de Xcos se muestra el desarrollo de varias prácticas de control. El Xcos trabaja en un ambiente de diagramas de bloques de los cuales se da su definición y forma de utilización. Se presenta una serie de ejemplos de simulación de sistemas de control y así mismo el desarrollo de las guías de laboratorio de control que usando Simulink de Matlab y Xcos logrando comparar los resultados de estos dos software de simulación. Se abre la posibilidad de trabajar sistemas más complejos con el uso de tarjetas de comunicación entre el software Xcos de Scilab con plantas físicas.

PALABRAS CLAVE:..Scilab, Xcos, Simulink, Simuladores

V° B° DIRECTOR DE TRABAJO DE GRADO

ABSTRACT OF THESIS PROJECT

TITLE: **INTRODUCTORY GUIDE TO THE XCOS SIMULATOR OF
SCILAB APPLIED TO THE SIMULATION OF CONTROL
SYSTEMS**

AUTHORS: **EDINSON BAUTISTA QUIROGA**

DEPARTMENT: **ELECTRONICS ENGINEERING**

DIRECTOR: **EDGAR BARRIOS URUEÑA**

ABSTRACT

This project is an introductory guide for the management of the Scilab software Xcos tool, which manages the simulation of analog and discrete control allowing you to find enough information to learn how to handle Scilab Xcos with the help of different practices control developed. There is a block diagram environment more common with its definition and explain its configuration. He presented several examples of simulation and control systems as well as the development of guidelines for control laboratory using Simulink in achieving Xcos compare the results of these two simulation software. In this way more complex project works with the use of cards for communication between the software Scilab Xcos in real working areas.

KEYWORDS Scilab, Xcos, Simulink, Simulator

V° B° THESIS DIRECTOR

INTRODUCCIÓN

El software Scilab elaborado por INRIA (Institut National de Recherche en Informatique et Automatique) en el año 1990 fue desarrollado para la solución de procesamiento de señales, sistemas de control y operaciones matemáticas. Este software es distribuido de manera gratuita, por lo cual es posible realizar la descarga desde su página oficial www.scilab.org. Como complemento Scilab es posible descargar módulos adicionales desde su página oficial. La versión actual es la 5.3 con la cual se realizara este Proyecto bajo la plataforma Windows 7.

La instalación de Scilab no requiere de muchos conocimientos en programación, debido a que es posible montarlo en sistemas operativos tanto Linux(32bits,64bits), Mac y Windows(32bits,64bits); no requiere mucho espacio en el disco duro, aproximadamente unos 130Mb de espacio libre, requiere de algunos compiladores de C y Fortran para su ejecución.

La introducción de Scilab está motivada por las limitaciones de la comercialización de Matlab ya que es un software licenciado; para el estudiante obtener un software licenciado se convierte en un problema por su alto costo en el mercado. La eficiencia y el uso permanente de Matlab han llevado a que el estudiante desconozca la existencia de otros software similares de descarga gratuita en internet.

Una de las principales características de Scilab es que está habilitada para el manejo de matrices y múltiples tipos de operaciones con ellas; permite el manejo de polinomios, números complejos y muchas otras funciones.

También cuenta con un entorno de simulación en diagramas de bloques, similar a Simulink de Matlab, llamado Xcos. En Xcos es posible simular sistemas de control.

En el cuarto capítulo se describe los inicios de Scilab junto con su manejo; en el capítulo cinco se muestra el manejo y entorno de Xcos; los bloques de simulación se describen en el capítulo seis junto con la ubicación de librería; en el capítulo siete se desarrollaran ejemplos de simulación en Xcos y finalmente en el capítulo ocho se muestra el desarrollo de experiencias de laboratorio de control analógico y digital de la Universidad Pontificia Bolivariana-Seccional Bucaramanga, haciendo un paralelo entre la simulación en Simulink y Xcos.

1. OBJETIVOS

El objetivo principal de este Proyecto fue elaborar una guía de aprendizaje de Xcos de Scilab aplicado a la simulación de sistemas de control.

Los objetivos específicos del Proyecto fueron los siguientes:

- Recopilar información sobre el uso de Xcos, especialmente en sistemas de control
- Desarrollar las prácticas de laboratorio de control que usando el simulador de Simulink de Matlab y Xcos de Scilab.
- Elaborar una guía que facilite a profesores y estudiantes el aprendizaje de manejo de Xcos, que incluya entre otros, sus características principales y ejemplos de simulación.

2. Marco teórico

En el proceso de formación del ingeniero electrónico se tiene la oportunidad de conocer diferentes herramientas de trabajo, tales como libros, revistas, consultas por internet, salas de laboratorios, instrumentos de medición, software, etc., que ayudan al estudiante, adquirir conocimientos básicos para la solución en problemas de mayor magnitud.

El uso del software ha cambiado la forma de operar de los estudiantes y de los profesionales en el área de control con la ayuda de simuladores que permiten obtener mejoras en el análisis y diseño de sistemas sin la necesidad de materializar el sistema mismo y la detección de fallas comparando el sistema real con el sistema simulado.

Uno de los problemas que se presentan en las aulas académicas hoy día es la implementación de software licenciado por sus altos costos de comercialización. Sobra decir que la universidad presta estas herramientas (software licenciado) para el desarrollo de prácticas de laboratorio. La problemática se presenta cuando el estudiante o el profesional quieren realizar trabajos de simulación en control fuera de la universidad puesto que obtener la licencia de este software es costoso.

Con este Proyecto se logra desarrollar una guía para los estudiantes y los docentes, donde pueden encontrar información del simulador Xcos de Scilab. Este software es de descarga libre en internet facilitando su uso convirtiéndose en una herramienta nueva de trabajo.

2.1 Software de simulación

Un simulador o software de simulación es una herramienta computarizada que permite realizar diseños, pruebas, correcciones y finalmente obtener datos de lo que sería el comportamiento de un sistema real. Es posible realizar la simulación para un ambiente específico, no requiriendo esfuerzo en programación. El gran avance tecnológico ha desarrollado hoy en día simuladores para la gran mayoría de ambientes laborales e industriales permitiendo obtener mejores resultados en el desarrollo del modelo.

Estos simuladores en el área de control normalmente funcionan o se programan por medio de diagramas de bloques que simulan los elementos reales, el usuario puede configurar y modificar fácilmente el sistema agregando, eliminando o configurando los diferentes componentes que ofrece el simulador.

En el mercado existen diferentes tipos de software para el desarrollo de procesos antes de su construcción, basados en un modelo virtual evaluado por diferentes procesos de simulación. Algunos de estos son software licenciado y otros son de libre descarga.

2.2 Software licenciado

El Software Licenciado es un producto que se comercializa con ciertas condiciones o términos de uso, establecidas por el fabricante. Una de las condiciones es la limitación del número de equipos en los que es posible instalar este producto comúnmente conocida como "licencias". Aunque en los laboratorios de control se trabaja con software licenciado como Simulink de Matlab esto se convierte en un problema fuera de las aulas para el estudiante o profesionales que necesiten estas herramientas ya que se necesitan un permiso legal para su instalación.

2.3 Diagrama de bloques

Un diagrama a bloques es una representación matemática gráfica del modelo matemático de un sistema. En muchos casos, estos diagramas nos permiten entender el comportamiento y conexión del sistema y a su vez, esta descripción puede ser programada en simuladores que tienen un ambiente gráfico. Cabe destacar que cada simulador se programa diferente aunque tenga herramientas de trabajo similares este parámetro marca la diferencia del manejo de Simulink de Matlab y Xcos de Scilab.

Para lograr trazar un diagrama de bloques en cualquiera de estos simuladores deben cumplir con los siguientes parámetros:

- Entrada del sistema: para la simulación de cualquier sistema es importante la excitación del sistema.
- Sistema: Modelo matemático de la planta a simular representado por medio de ecuaciones.
- Salida: Respuesta del sistema ante la variable de entrada.

3. Metodología

Para ampliar las herramientas de trabajo en los laboratorios de control se realiza una guía introductoria al usuario donde se exponen las múltiples herramientas del simulador Xcos de Scilab; en ella se explica el funcionamiento del software y la configuración de sus herramientas más comunes. Además se desarrollan experiencias de laboratorio de control que usualmente utilizan el simulador de Simulink de Matlab en Xcos ofreciendo una alternativa de trabajo con un software de libre comercialización.

Para el desarrollo del Proyecto se siguieron las siguientes actividades:

1) Recopilación y análisis de información

Como primer paso se recopiló información de diferentes autores bibliográficos, basándose en libros, revistas, manuales, tesis y website relacionada con el software Scilab para lograr conocer su funcionamiento y características del mismo, observando su similitud en manejo y desarrollo con el software de simulación Simulink.

Una vez analizada y resumida esta información, se hace una pequeña introducción a Scilab previamente descargado de internet, conociendo el funcionamiento de la consola generando ejemplos que permitan el mejor entendimiento de sus funciones. Se mencionan las opciones de ayuda que brinda este software para su mejor funcionamiento y algunos de los comandos más comunes.

Una vez familiarizado con Scilab se accede al simulador Xcos. Por definición se puede decir que Xcos es una herramienta grafica que brinda múltiples aplicaciones y cuenta con gran cantidad de bloques de trabajo permitiendo simular diferentes sistemas de control. Con este concepto se definen los bloques más comunes y sus parámetros de configuración.

2) Desarrollo de ejercicios y prácticas del laboratorio de control

Para el desarrollo de las prácticas del laboratorio de control, se clasifican las guías que usan el simulador de Simulink de Matlab y se desarrollan en el simulador de Xcos de Scilab, comparando los diagramas de bloques de cada software y observando los resultados obtenidos en ambos simuladores. Se plantea ejercicios básicos de configuración simple para un mejor entendimiento de programación de Xcos pasando a ejercicios más complejos como controladores PID.

3) Realización de la guía introductoria de Xcos de Scilab

Una vez recopilada la información necesaria se procede al desarrollo de la guía introductoria de Xcos de Scilab como de muestra en los siguientes capítulos:

Capitulo 4: Iniciando con Scilab, se muestra el entorno grafico del software y conceptos básicos de programación y simulación.

Capitulo 5: Scilab/Xcos, se dan a conocer las diferentes opciones de inicio a Xcos desde Scilab, mostrando su entorno de trabajo y la barra de herramientas.

Capitulo 6: Bloques principales; se describen los bloques comúnmente usados, con su descripción y configuración.

Capitulo 7: Ejercicios; se realizan ejemplos para facilitar el entendimiento y utilización del simulador.

Capitulo 8: Desarrollo guía de laboratorio; se desarrollan las guía de laboratorio de control en Simulink y Xcos comparando los diagramas de bloques y los resultados obtenidos en ambos simuladores.

4. Iniciando con Scilab

En esta sección, se mostrara los primeros pasos con scilab y algunos ejemplos sencillos desde la consola. Hay varias maneras de usar este software, como se muestra a continuación:

- Uso de la consola de modo interactivo,
- Usando el editor,
- Usando la función **exec**,
- Con el procesamiento por lotes,

El procesamiento por lotes y la función `exec` son usados en procesos donde se van a ejecutar varios comandos

4.1 Consola

Este es el primer caso para manejar Scilab de una manera interactiva, por medio de escritura en el comando en la consola, analizando los resultados y continuando su proceso hasta obtener los resultados finales.

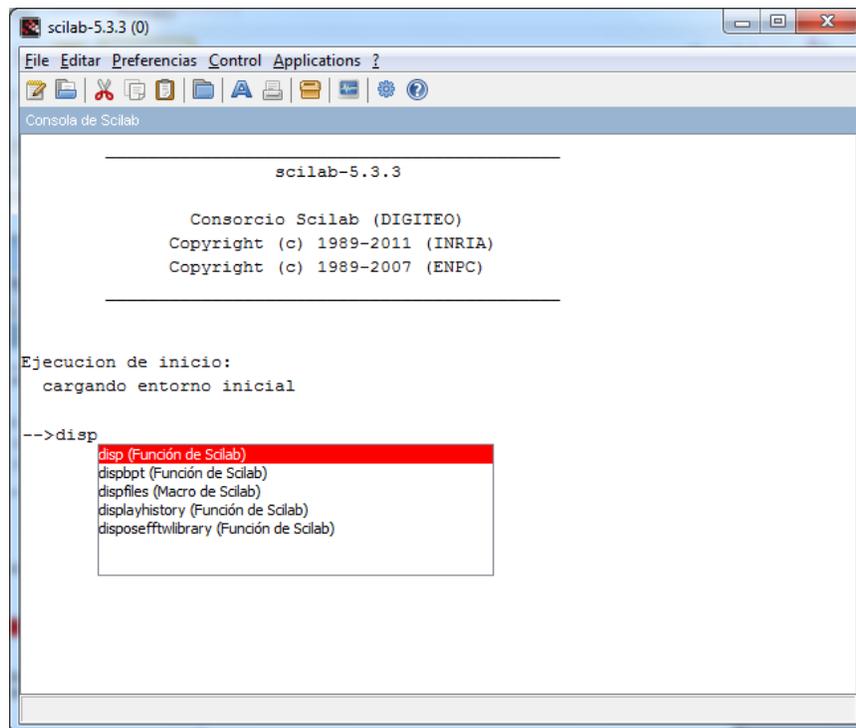


Figura 1 Consola scilab

Para mostrar un ejemplo de comandos se genera una forma de mostrar un resultado o contenido de una variable, como se muestra a continuación,

```
-->a="hola"
```

```
a =
```

```
hola
```

```
-->disp(a)
```

```
hola
```

Como se muestra en la figura 1, se observa una ayuda de comandos, esto se logra por medio de la tecla <Tab>, la cual permite terminar la función de una manera conveniente.

Para obtener ayuda con comandos se logra por medio de la consola como se muestra a continuación,

```
-->help disp
```

Nos aparecerá el navegador de ayuda, junto con ejemplos de cómo usar el comando.

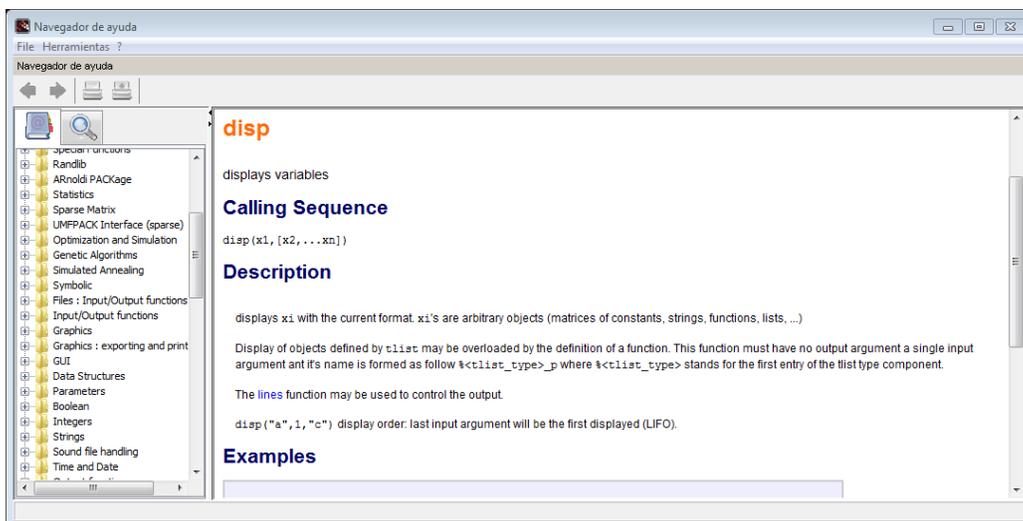


Figura 2 Navegador de ayuda

Los nombres de los comando habituales son los habituales, algunos de ellos son:

sqrt(x)	Raiz cuadrada	sin(x)	Seno(radianes)
abs(x)	Módulo	cos(x)	Coseno(radianes)
Conj(z)	Complejo conjugado	Tan(z)	Tangente(radianes)
Real(z)	Parte real	Cotg(x)	Cotangente(radianes)
Imag(z)	Parte imaginaria	Asin(x)	Arcoseno
Exp(x)	Exponencial	Acos(x)	arcocoseno
Log(x)	Logaritmo natural	Atan(x)	Arcotangente
Log10(x)	Logaritmo decimal	Cosh(x)	Coseno hiperbolico
Rat(x)	Aprox racional	Sinh(x)	Seno hiperbolico
Modulo(x,y)	Resto dividir x por y	Tanh(x)	Tangente hiperbolica
Floor(x)	N tal que $n \leq x < (n+1)$	Acosh(x)	Arcoseno hiperbolico

Tabla 1 Comandos

4.2 Editor

Scilab ofrece un editor que permite editar scripts fácilmente como se muestra en la figura 3. Hay dos maneras de obtener esta ventana, desde consola escribiendo **editor** o desde el menú aplicaciones/SciNotes.

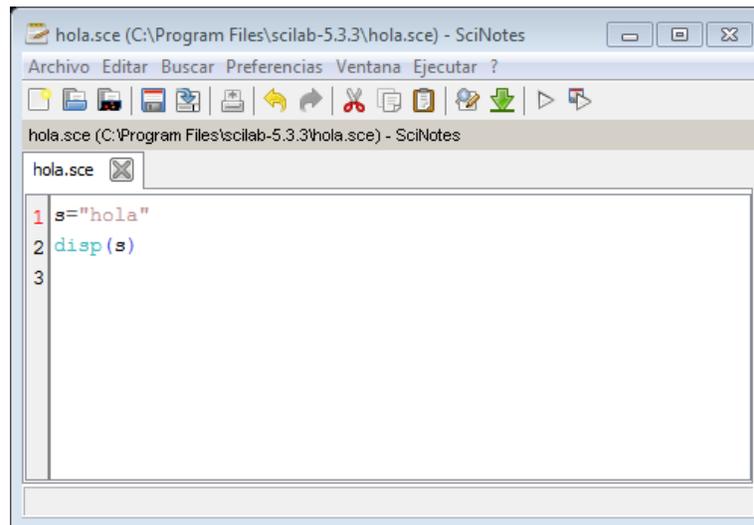


Figura 3 Editor Scilab

Al ejecutar los programas el resultado es mostrado en la consola de scilab.

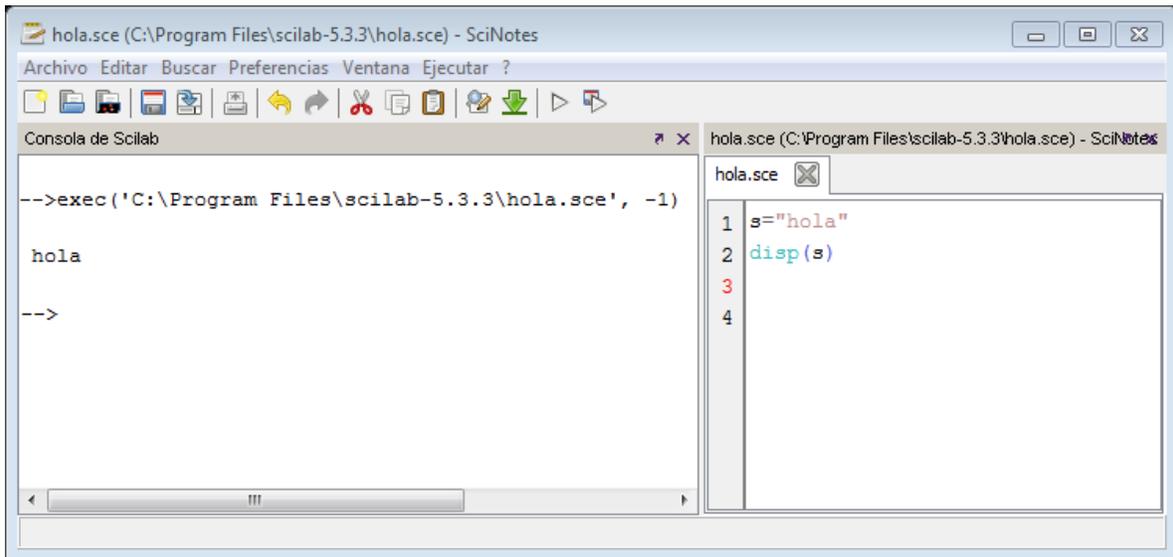


Figura 4 Scilab

Es posible integrar el editor a Scilab como se muestra en la figura 4.

Para colocar comentarios se escribe *//comentario* en la línea de comando donde se necesite.

4.3 Ejemplo

En Scilab es posible simular la respuesta de una planta controlada por PD con una señal escalón de entrada. Se utiliza las funciones `syslin()` y `csim()`.

`csim()` simula la respuesta en el tiempo de un sistema lineal, la estructura del comando es

```
r=csim(u,t,sl)
```

Dónde:

u : es la función de entrada en este caso step (escalón)

t : el vector tiempo de evaluación

sl : La respuesta de la función `syslin`

r : respuesta

Usando la consola de Scilab tenemos,

```
-->s=poly(0,'s');  
  
-->t=0:0.01:6;  
-->//planta  
-->pl=1/(s*2+1);  
-->//controlador  
-->K=1;  
-->D=1;  
-->sis=K*(1+D*s);  
-->res=sis*pl/(1+sis*pl);  
-->y=syslin('c',res);  
-->z=csim('step',t,y);  
-->plot2d(t,z,2);  
-->xtitle('Proporcional.Derivativo (1/(s*2+1)) K=1 D=1','tiempo','y(t)');
```

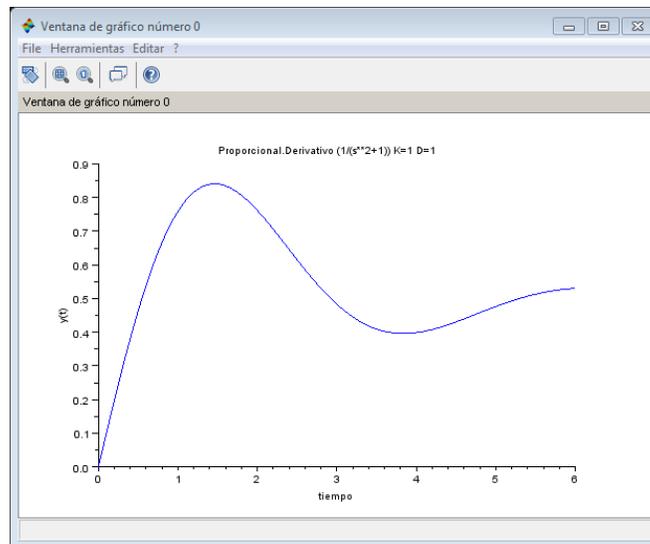


Figura 5 Controlador PD

5. Scilab/Xcos

Scilab cuenta con una herramienta llamada Xcos, siendo un simulador gráfico, con el cual es posible la simulación de sistemas dinámicos, solución de sistemas de ecuaciones diferenciales, procesamiento de señales, sistemas de control, estudio de sistemas biológicos, etc. Este software es desarrollado por INRIA de libre distribución. Este entorno posee varios paquetes para simulación como herramientas de circuitos eléctricos, sistemas de control, termo hidráulica, etc.

Para familiarizarse con este software se comienza por las opciones que hay para acceder a Xcos en Scilab. Se cuenta con un icono en la barra de herramientas de Scilab como se muestra en la figura 6; la otra forma es ingresando por el menú Applications donde se elige la sub-opción Xcos; la manera más rápida es escribiendo en el comando de Scilab la palabra Xcos; estas tres opciones nos abrirá las ventanas mostradas en las figuras 7 y 8

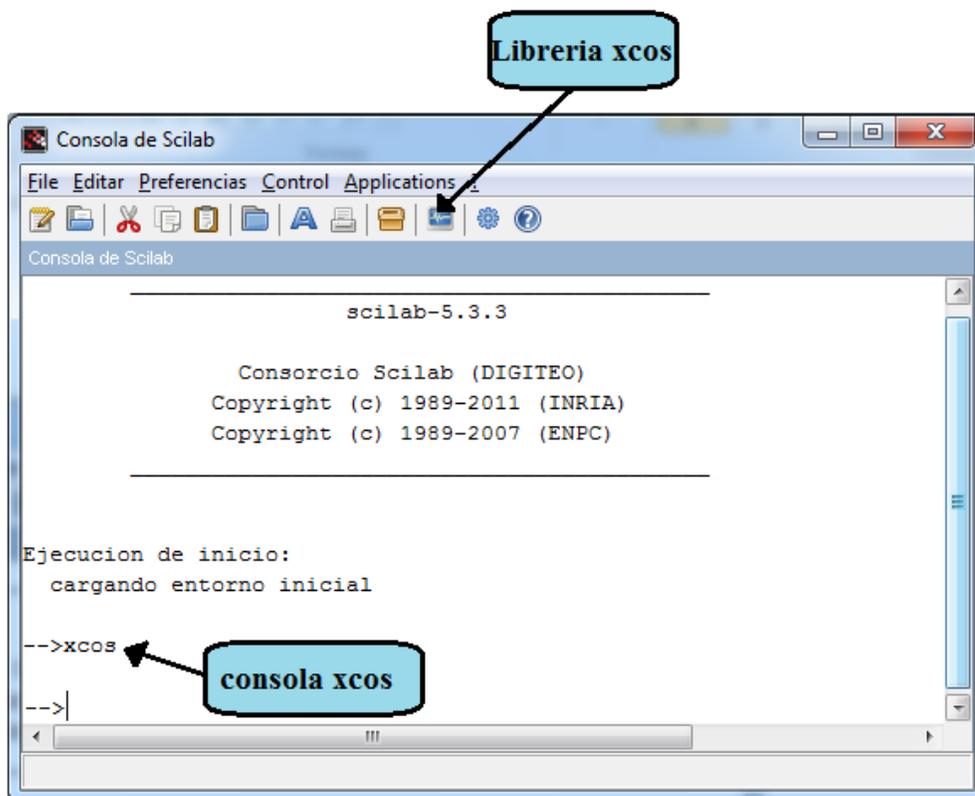


Figura 6 Ventana principal Scilab

La figura 7 muestra la ventana principal de Xcos, donde se crea cualquier modelo o controlador a simular. También es posible configurar los parámetros de simulación en el menú simulación seleccionando la sub-opción configuración, donde nos muestra la ventana de la figura 8, donde se puede establecer el tiempo de simulación.

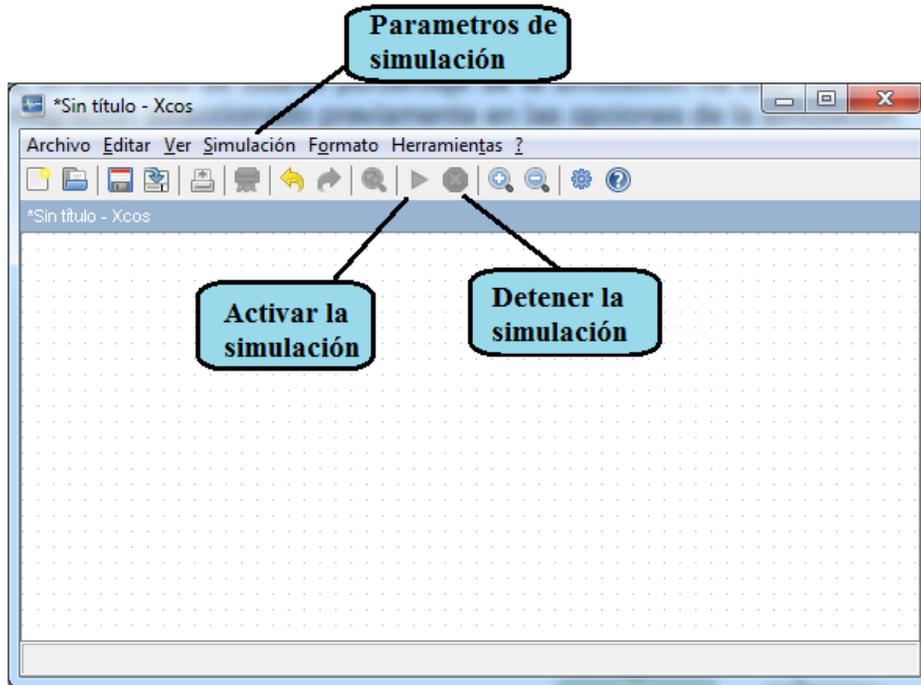


Figura 7 Ventana principal Xcos

Al activar la herramienta Xcos también abre la ventana explorador de paletas-Xcos como se ve en la figura 9. En el toolbox se encuentran todos los bloques para la construcción de sistemas en Xcos, organizado en diferentes librerías, dependiendo de su uso. Al seleccionar algún elemento del toolbox nos mostrara los diferentes bloques disponibles.

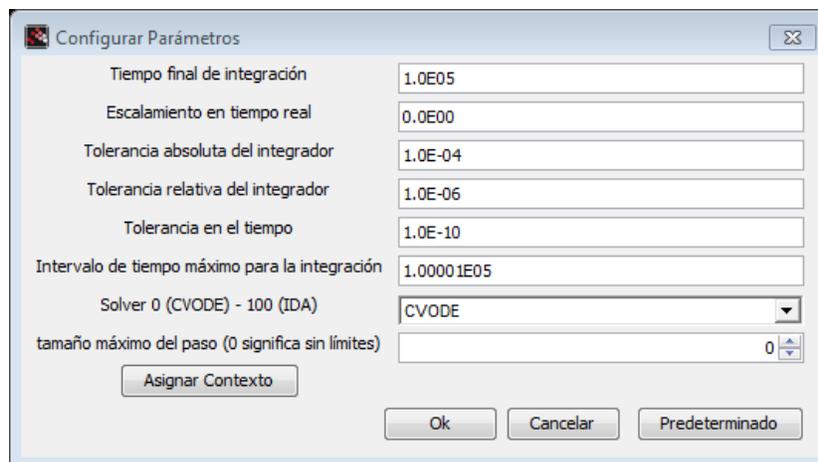


Figura 8 Parámetros de simulación

Estos bloques son posibles usarlos al desplazarlo a Xcos directamente o seleccionando el icono y escoger la opción añadir a "nombre del sistema"-Xcos. Al seleccionar también se logra acceder a la ayuda Help, donde ofrece una mayor descripción y funcionalidad para la programación de Xcos.

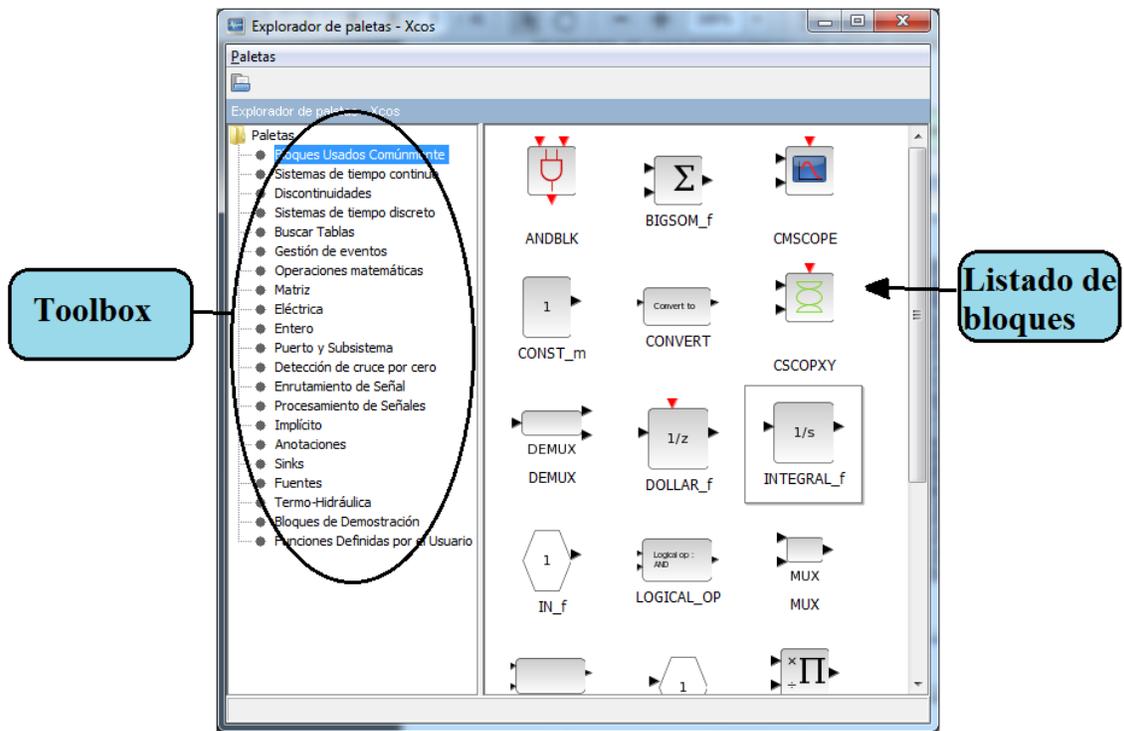


Figura 9 Librería Xcos

6. Bloques principales de Xcos

En este capítulo se presentan los bloques más usados en Xcos para la simulación de sistemas en tiempo continuo, discreto, circuitos eléctricos, fuentes entre otros. Se mostrara la configuración de los bloques donde el usuario será capaz de configurarlo a necesidad de la exigencia del problema a resolver. Para acceder a la ventana de configuración se logra por medio de doble click en el icono en la ventana de Xcos.

En la gráfica 10 se muestra los bloques más usados, los cuales se detallaran a continuación.

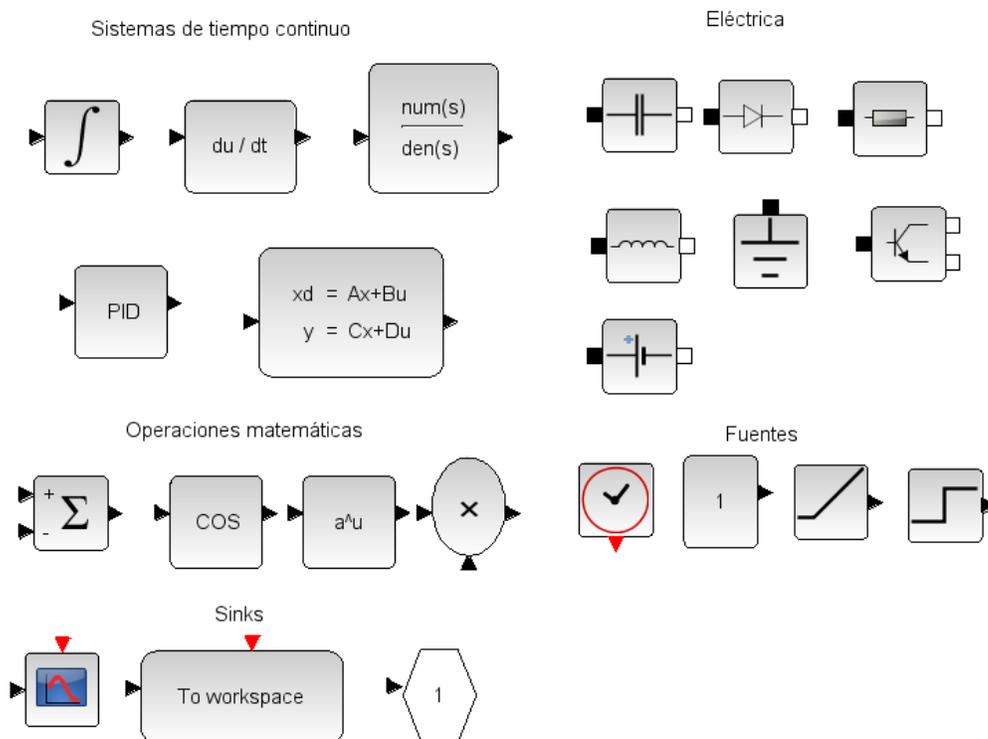


Figura 10 Bloques principales Xcos

6.1 Derivada



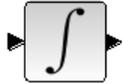
El bloque de Derivada aproxima la derivada instantánea de su entrada con respecto al tiempo t . Es una medida de cómo cambia una salida como su entrada cambia Δu .

El derivado está dada por:

$$\frac{\Delta u}{\Delta t}$$

El bloque derivado no tiene ningún valor y la salida inicial para el bloque es cero.

6.2 Integrador



La salida es la integral de la entrada U, en el paso un instante de tiempo t.

$$y(t) = \int_{t_0}^t u(t)dt + y_0$$

Donde y_0 es una condición inicial y t_0 es el tiempo correspondiente.

El cuadro de diálogo de parámetros de la figura 11, le permite definir:

- La condición inicial con o sin límite de saturación.
- Los límites superior e inferior de la integral.

Le permite también añadir un puerto de evento para el bloque con el fin de restablecer su Estado a la condición inicial especificada. En este caso, el valor de la señal de reposición depende de una señal externa.

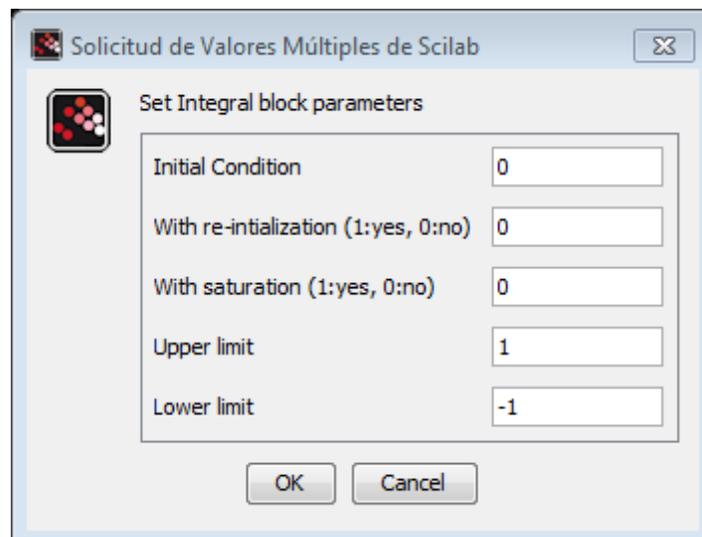
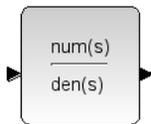


Figura 11 Propiedades integrador

A continuación se hará una explicación de cómo usar las propiedades del integrador

- *Initial condition*, una condición inicial un vector/escalar. Con este parámetro se puede definir el tipo de dato de la entrada/salida. Puede ser de tipo real o complejo.
- *With re-initialization*, para restablecer su estado a la condición inicial especificada.
- *With saturation*, si se selecciona, limita a los estados a un valor comprendido entre el límite de saturación inferior y superior.
- *Upper limit*, el límite superior para la integral.
- *Lower limit*, el límite inferior para la integral.

6.3 Continuous transfer function



Este bloque comprende un sistema lineal SISO, representado por su función de transferencia Numerador/denominador. La función debe ser adecuado (El denominador debe ser de menor grado o igual al grado numerador). En la figura 12, se logra construir la función de transferencia como se ve en la misma gráfica.

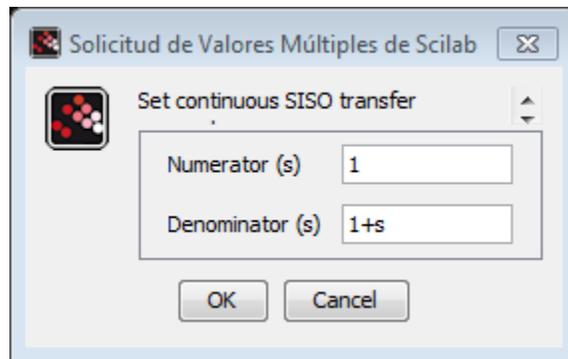
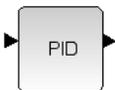


Figura 12 Transfer function

6.4 PID regulator



Este bloque implementa un controlador PID (Proporcional Integral Diferencial). Se calcula un "error" valor de U_e como la diferencia entre una medida variable de proceso del U_{pr} , y un punto de ajuste deseado U_r .

$$U_e = U_r - U_{pr}$$

El propósito es hacer que la variable del proceso siga el valor de U_r . El controlador PID es ampliamente utilizado en el control de realimentación de los procesos industriales.

El cálculo del controlador PID (algoritmo) consta de tres parámetros distintos; el K_p Proporcional Integral, el K_i y K_d valores derivados.

Estos términos describen tres funciones básicas de matemáticas aplicadas a la señal de error de U_e . K_p determina la reacción al error actual, K_i determina la reacción basada en la suma de errores recientes y K_d determina la reacción a la velocidad a la que el error ha sido cambiante.

La suma ponderada de estas tres acciones se utiliza para ajustar el proceso a través de un elemento de control tales como la posición de una válvula de control o la fuente de alimentación de un elemento de calentamiento. La estructura básica de los sistemas convencionales de control de realimentación se muestra a continuación:

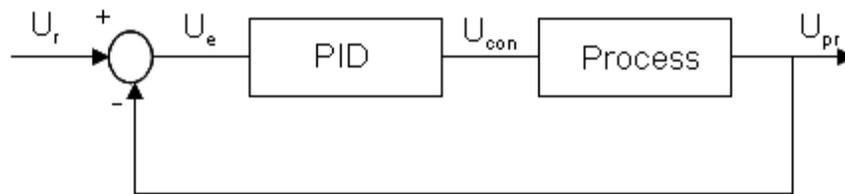


Figura 13 Control PID

Para configurar el controlador PID se logra por medio de sus propiedades como muestra la figura 14

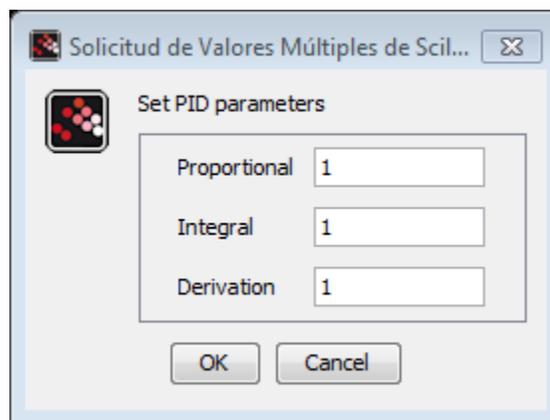
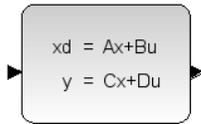


Figura 14 Propiedades PID

6.5 Continuous state-space system CLSS



Este bloque comprende un continuo en el tiempo lineal de espacio de estado del sistema.

$$\dot{\tilde{x}} = Ax + Bu$$

$$y = Cx + Du$$

Donde x es el vector de variables de estado, u es el vector defunciones de entrada y y es el vector de variables de salida.

El sistema está definido por las matrices(A,B, C, D) y el estado inicial X_0 . Las dimensiones deben ser compatibles.

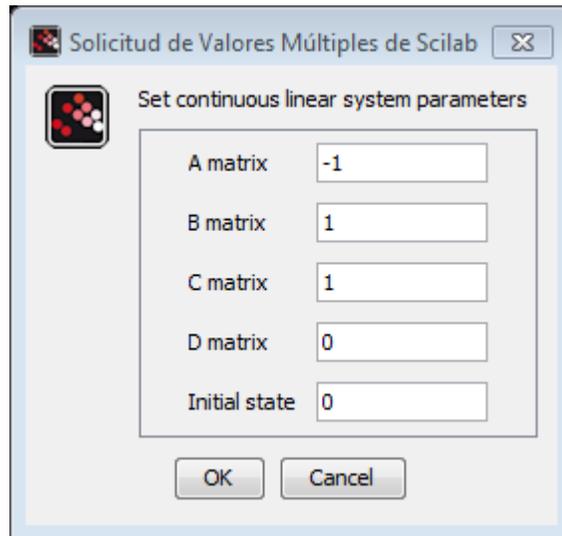
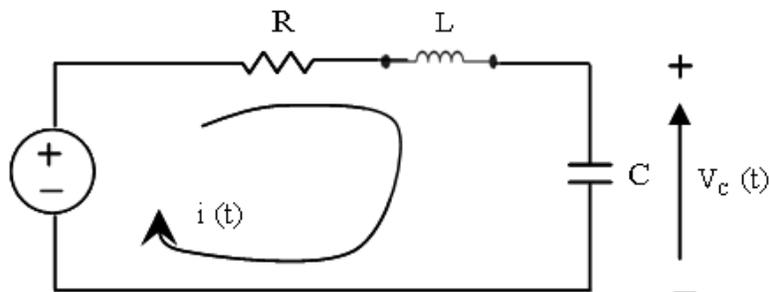


Figura 15 Propiedades variable de estados

6.5.1 Ejemplo

Este ejemplo ilustra cómo utilizar bloque CLSS para simular y visualizar la forma de onda de salida y $(t) = V_c(t)$ del circuito RLC que se muestra a continuación.



Del anterior circuito obtenemos que,

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_0(t)$$

$$y(t) = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

Donde $R=10\Omega$, $L=5 \text{ mH}$ y $C=0.1\mu\text{F}$; los valores iniciales son $x_1=0$ y $x_2=0.5$

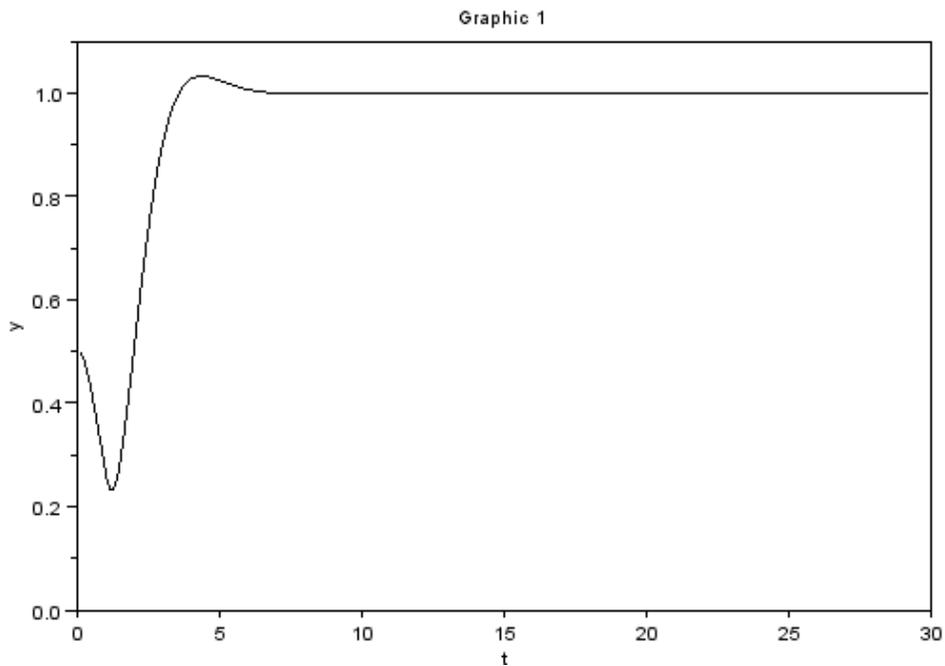
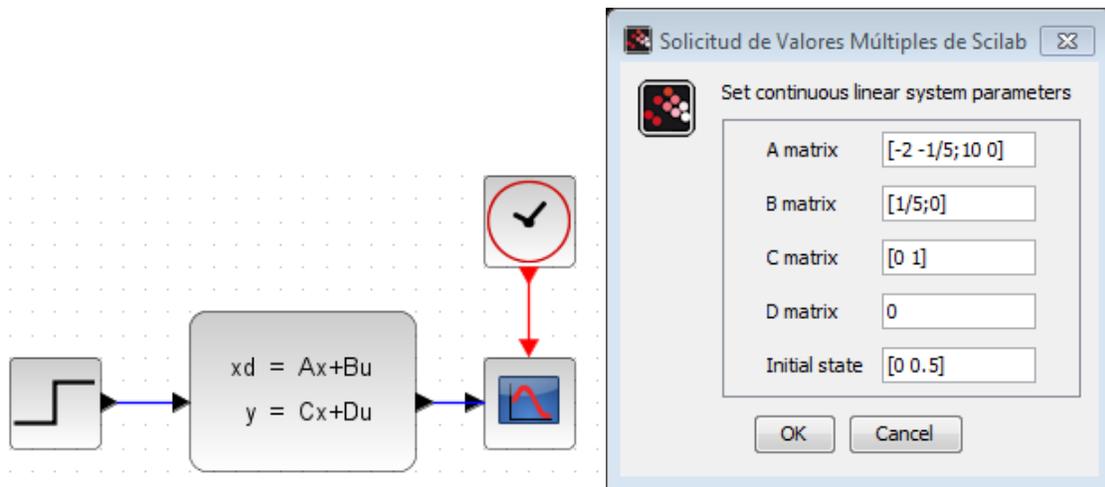


Figura 16 Ejemplo variable de estado

6.6 Electrical capacitor



Un condensador es un componente eléctrico que puede almacenar la energía en los circuitos eléctricos. La relación entre el voltaje a través de un condensador con la capacitancia y la corriente que pasa a través de ella está dada por la:

$$I = C \frac{dV}{dt}$$

Condensadores también puede utilizarse para diferenciar entre alta frecuencia y baja frecuencia y las señales de este los hace útiles en filtros electrónicos. Un condensador tiene una alta impedancia cuando una señal es de baja frecuencia.

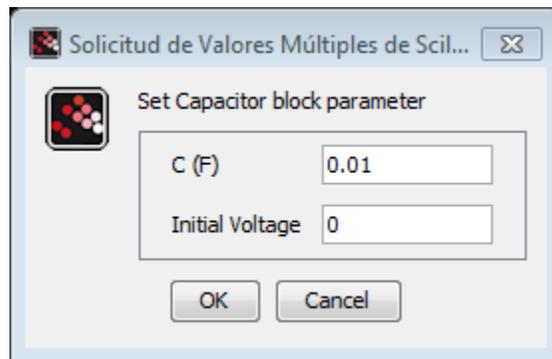


Figura 17 Propiedades capacitor

6.7 Electrical diode



Este componente consiste en un simple diodo en paralelo con una resistencia óhmica(R). La corriente (I) que pasa a través de este componente se define como una función del voltaje a través de los puertos(V):

$$i = I_{ds} \left(\exp^{v/V_t} - 1 \right) + \frac{v}{R}$$

Donde I_{ds} y V_t son la corriente de saturación y el equivalente de voltaje de la temperatura, respectivamente. Si el exponente alcanza un cierto límite (Max exponente para la continuación lineal), la característica del diodo se vuelve lineal para evitar el desbordamiento.

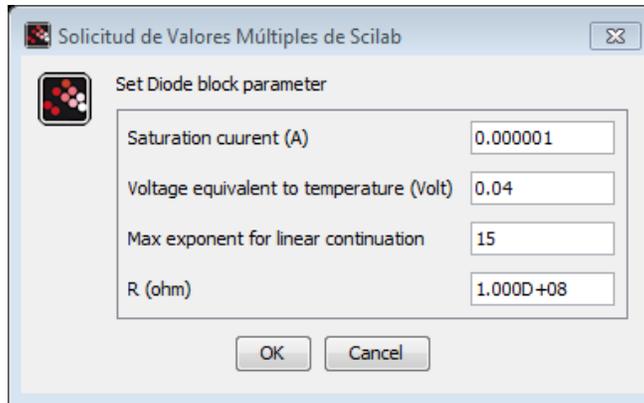
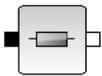


Figura 18 Cuadro de diodo

6.8 Electrical resistor



Una resistencia es un componente eléctrico de dos puertos que resiste una corriente eléctrica (I) por producir una caída de voltaje (V) entre sus terminales de acuerdo con la ley de Ohm.

$$R = \frac{V}{I}$$

La resistencia eléctrica (R) es igual a la caída de voltaje a través de (V) la resistencia dividida por la corriente (I) a través de la resistencia.

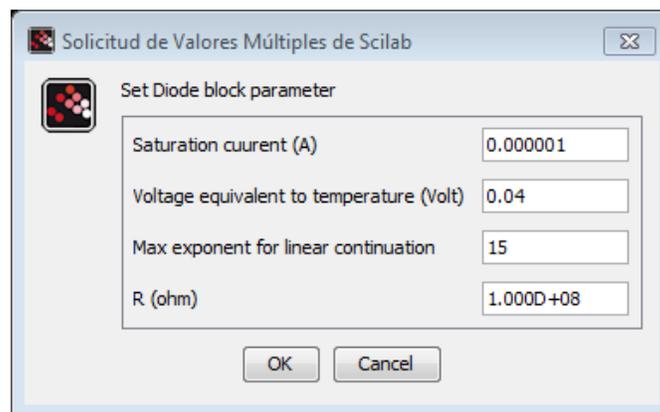


Figura 19 Propiedades de resistencia

6.9 Electrical inductor



Inductor es un componente eléctrico que puede almacenar la energía en los circuitos eléctricos. La relación entre el voltaje a través de los puertos de un inductor de la inductancia y la corriente que pasa a través de ella está dada por.

$$v = L \frac{dI}{dt}$$

Los inductores también puede utilizarse para diferenciar entre alta frecuencia y baja frecuencia y las señales de este los hace útiles en filtros electrónicos. Un inductor muestra una alta impedancia para señales de alta frecuencia.

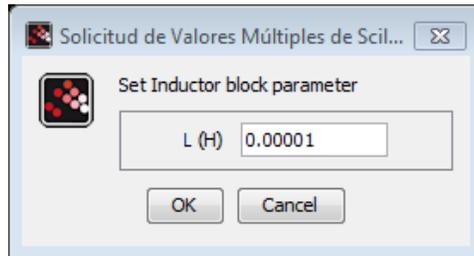


Figura 20 Propiedades bobina

6.10 Ground (zero potential reference)



El elemento de tierra es un único componente el cual proporciona un voltaje de referencia en los circuitos eléctricos. El potencial en el nodo de tierra es igual a cero. Cada circuito eléctrico tiene que contener al menos un elemento de tierra.

6.11 NPN transistor



Este modelo es un modelo simple de un transistor de unión bipolar NPN. En las propiedades de este bloque como muestra la figura 21 es posible configurar diferentes parámetros como son descritos en la siguiente tabla 2.

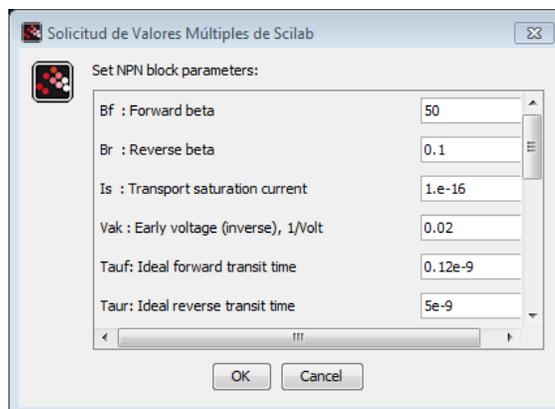
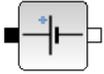


Figura 21 Propiedades transistor NPN

6.12 Electrical DC voltage source



Este componente es un modelo para cualquier dispositivo o sistema que produce una fuerza electromotriz constante entre su puerto. La tensión de salida de esta fuente de tensión de CC está definido por el usuario. El puerto negro indica el voltaje positivo. La resistencia óhmica de esta fuente de tensión continua es cero.

6.13 Activation clock



La única salida de este bloque es generar un muestreo que están programadas por el período de los parámetros en cuestión de segundos. La fecha de inicio de la generación de eventos se puede configurar en cuestión de segundos con el parámetro de tiempo de inicialización.

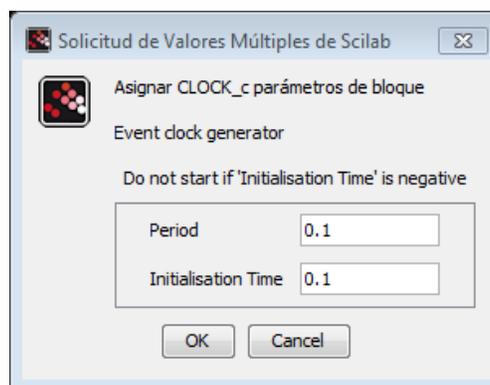


Figura 22 Activation clock

6.14 Constant



Este bloque genera un valor constante definido por el usuario.

6.15 Ramp



El bloque de rampa genera una señal que se inicia en un tiempo especificado y el valor inicial y se incrementa en una tasa especificada (parámetro Slope).

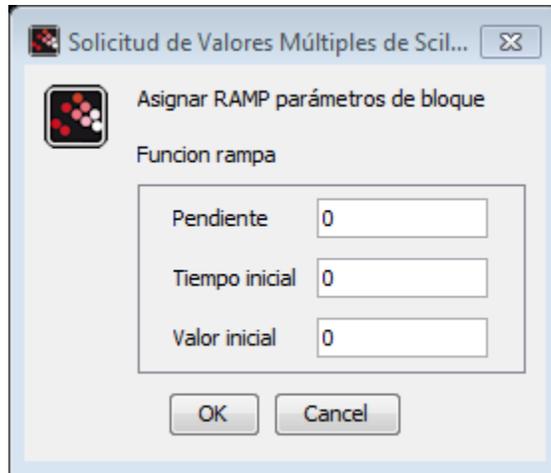


Figura 23 Ramp

6.16 Step Function



En su uso más simple este bloque de salida de una señal de paso entre dos niveles definidos por el valor inicial y el valor final a partir de un paso de tiempo especificado.

7. Ejercicios

En este capítulo se muestra como se construye el diagrama de bloque en Xcos con algunos ejemplos configurando los parámetros para su respectiva simulación.

7.1 Ejemplo configuración muestreo CSCOPE

Se muestra la configuración del clock  usado en el CSCOPE, en el cual se escoge el número de muestra de nuestra señal. Se toma como referencia un generador senoidal y se grafica su salida, como se ve en la figura 24, para variar el periodo de muestreo se hace doble click en el y aparece la ventana de la gráfica 25.

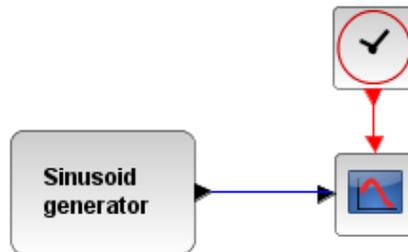


Figura 24 Generador senoidal

Se configura el periodo de muestreo en 0.1 para ver la señal del generador y se variara aumentándolo a 0.5 y 1 y se observa el comportamiento

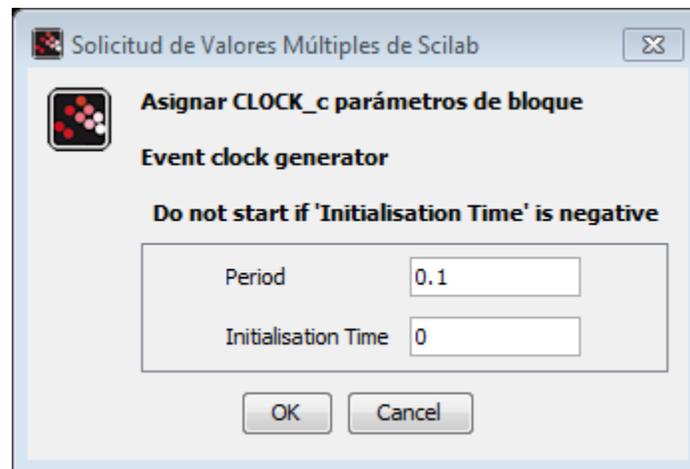


Figura 25 Periodo muestreo CSCOPE

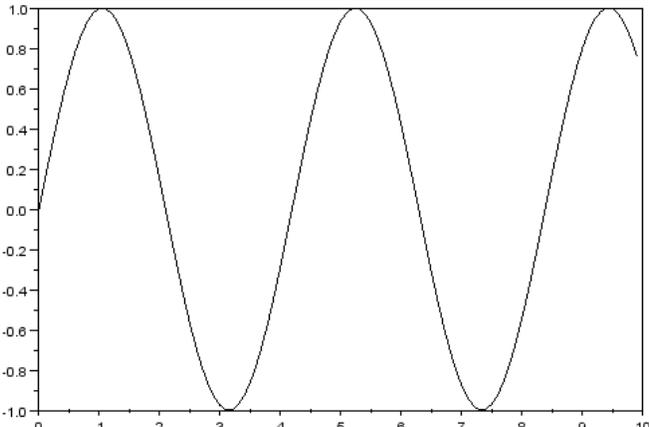
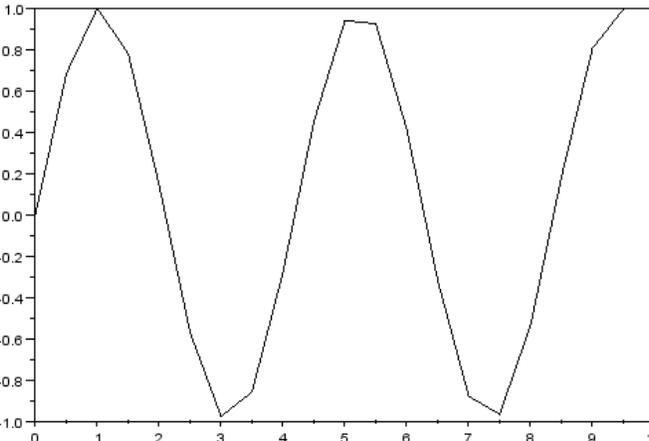
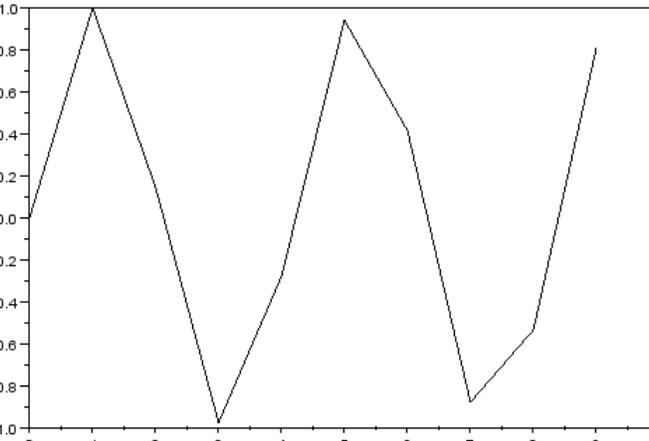
Periodo muestreo	Gráfica
0.1	
0.5	
1	

Tabla 2 periodo de muestro

Como se observa entre más grande sea el tiempo de muestreo más perdida de datos va a existir, por lo cual se recomienda tomar muestreos bajos para obtener una optima señal.

7.2 Practica de laboratorio. Experiencia1. Familiarización con el simulador SIMULINK de MATLAB,

La simulación de esta práctica se muestra en el siguiente capítulo, en esta sección se hace referencia al desarrollo paso a paso del modelo de simulación de Xcos y configuración para la simulación de cada uno de los bloques usados en el esquemático.

Inicialmente se ubica los iconos que se necesita para armar el esquema de simulación, estos se encuentran ubicados en diferentes carpetas de la toolbox paletas del explorador de paletas. Se ubica un escalón que se encuentra en fuentes, la sumatoria en los bloques más usados, el bloque de sistemas de tiempo continuo contiene el bloque CLR el cual es usado como planta y/o función de transferencia, se ubica un CSCCOPE ubicado en sinks junto con un reloj ubicado en fuentes.

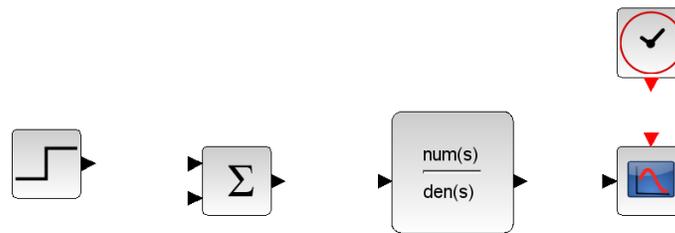


Figura26 Esquemas de simulación

Para unir los bloques, se ubica el cursor sobre los triángulos negros de salida de los bloques y aparecerá un cuadro verde indicando el punto para dar click como se muestra en la figura 27a, con click sostenido se lleva hasta la entrada del bloque a unir, donde aparece el bloque verde indicando que la conexión es posible realizarla figura 27b, a continuación se realiza todas las conexiones necesarias para la simulación como se observa en la figura 28.

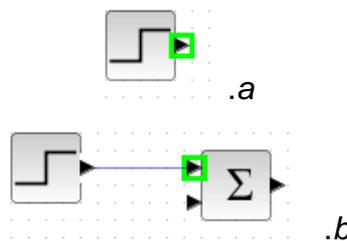


Figura 27 Inicio unión de bloques

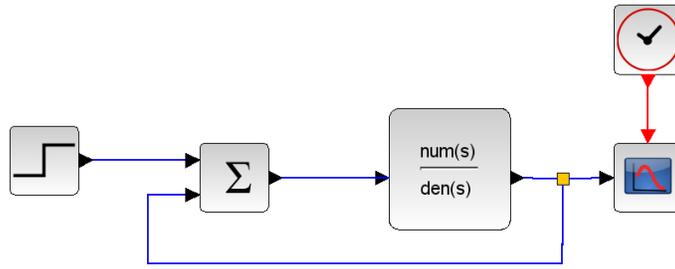


Figura 28 Esquemático de simulación

A continuación se configura los parámetros de simulación, se comienza por el generador de señal escalón, muestra un cuadro como la figura 29, se le define salida máxima de 1 y comenzar desde 0. Luego se configura el CLR, donde se ingresa la función de transferencia a simular como se muestra en la figura 30, en la cual nos pide un numerador y un denominador.

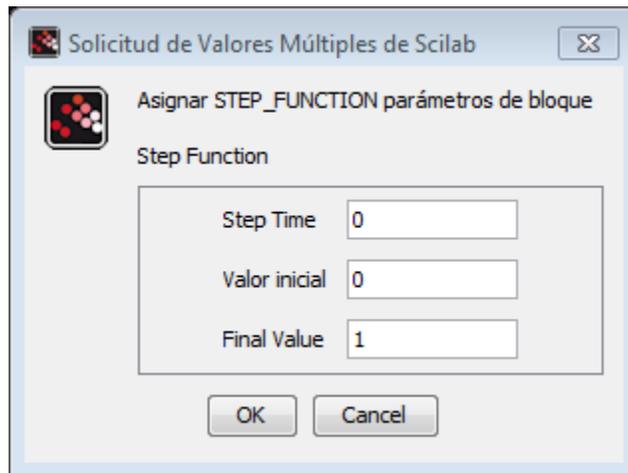


Figura 29 Escalón

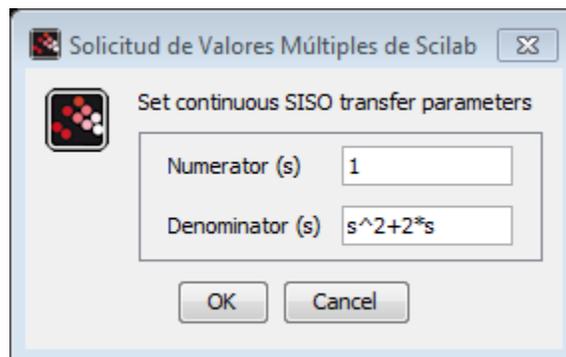


Figura 30 CLR

En el cscope se ingresa los límites de su eje a graficar como se muestra en la figura 31, en este caso es de 0 a 1.4 en el eje Y y un tiempo máximo de 10 en el eje X. Finalmente se configura el tiempo de simulación en el menú

simulación/configuración como se ve en la figura 32, el CSCOPE tiene la opción de configurar el periodo de muestro por medio del reloj

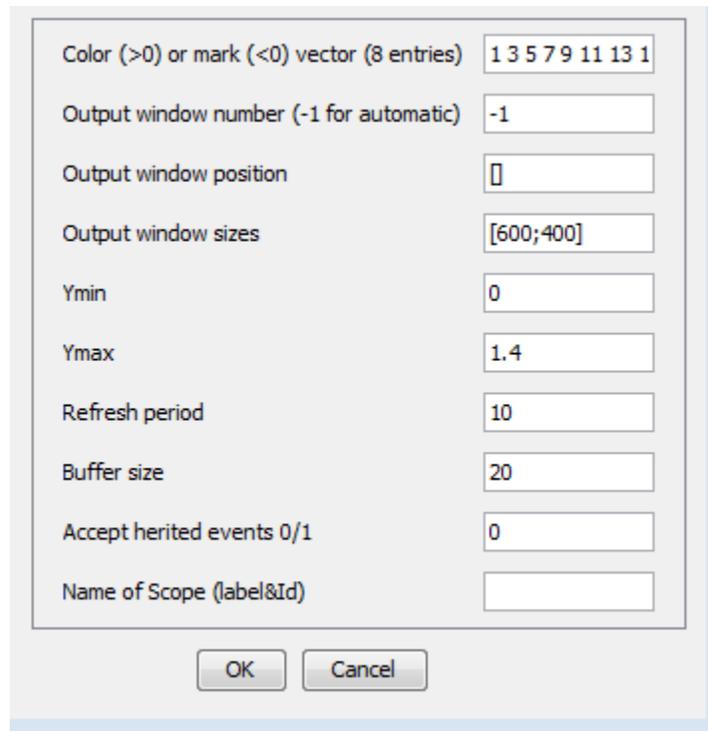


Figura 31 CSCOPE

En tiempo final de integración definimos el tiempo de simulación y a continuación se procede a simular el sistema con el icono , el CSCOPE nos muestra el resultado de la gráfica 33.

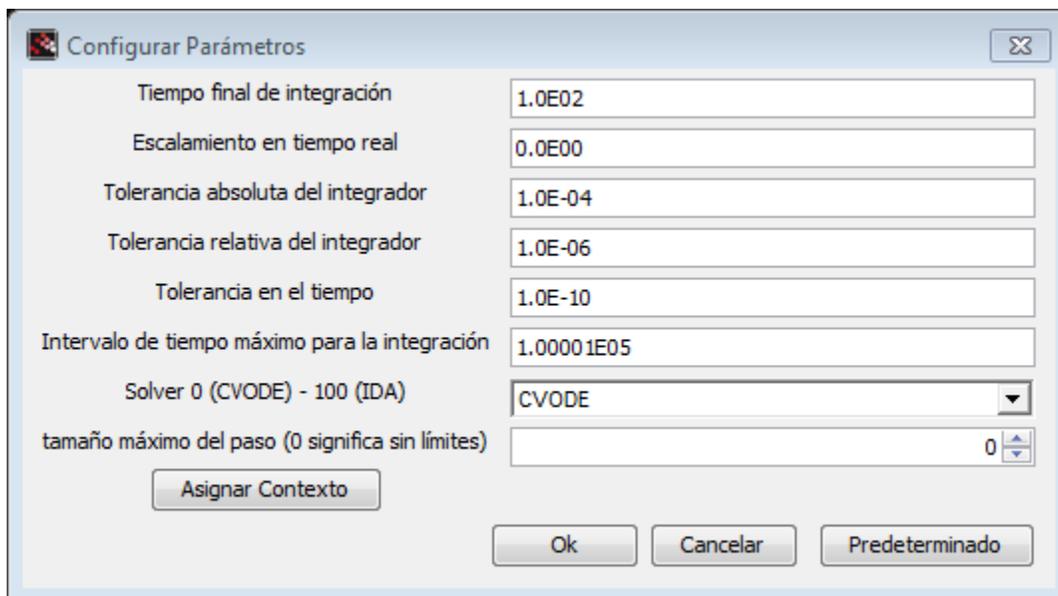


Figura 32 Configurar parámetros

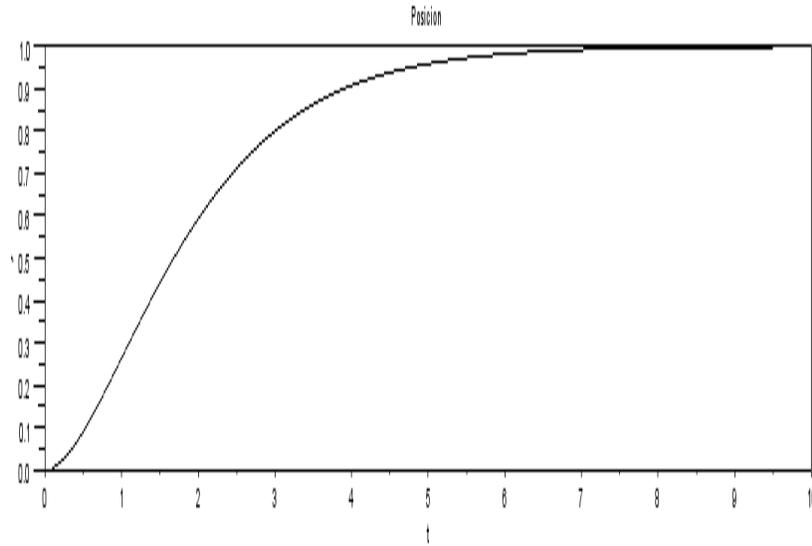


Figura 33 Simulación practica 1

7.3 Control a lazo cerrado de un PID

La construcción del esquema del sistema de la figura 34, se construye del mismo modo que el anterior ejemplo. Se elabora un sistema PID en el cual se pueda manipular las constantes K_p , K_i y K_d para obtener diferentes respuestas del sistema, estos valores son escogidos al azar.

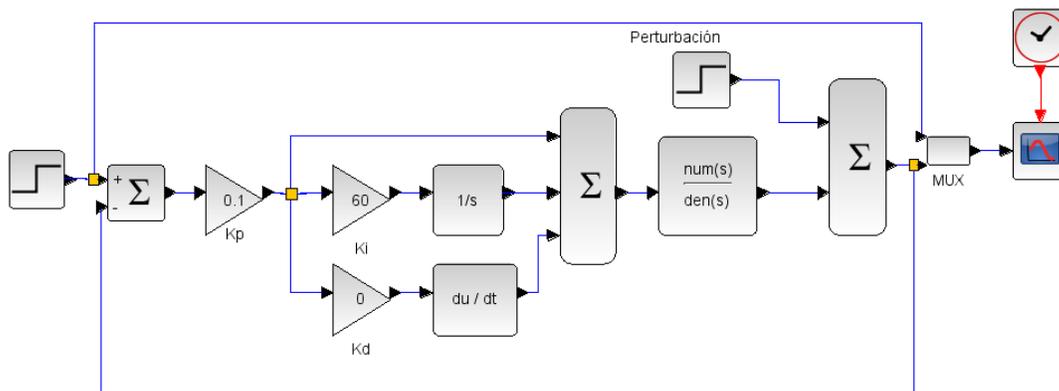


Figura 34 Control PID

Para los valores como se muestra en la figura 34 se obtiene el resultado de la gráfica 35. Para valores de $K_p=0.6$, $K_i=2$ y $K_d=0.125$ se obtiene el resultado de la gráfica 36.

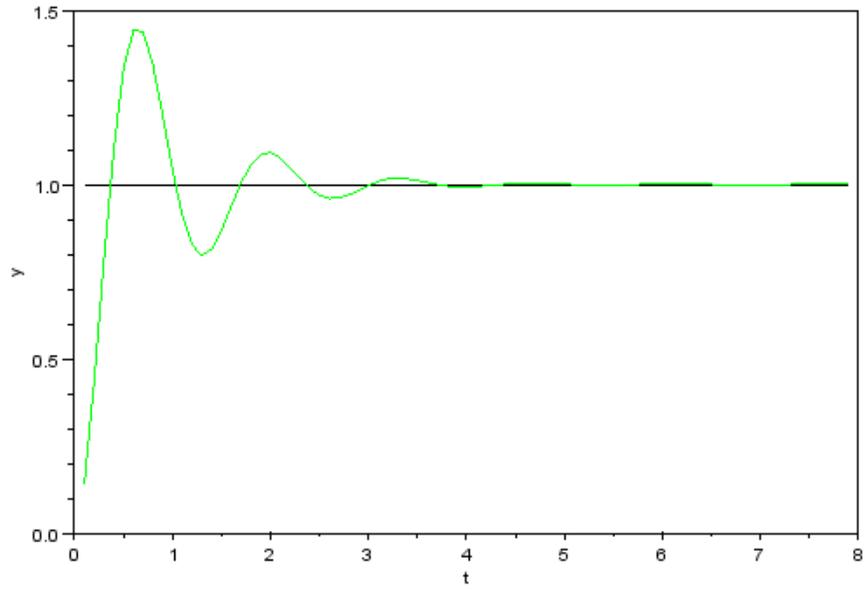


Figura 35 Respuesta control PID

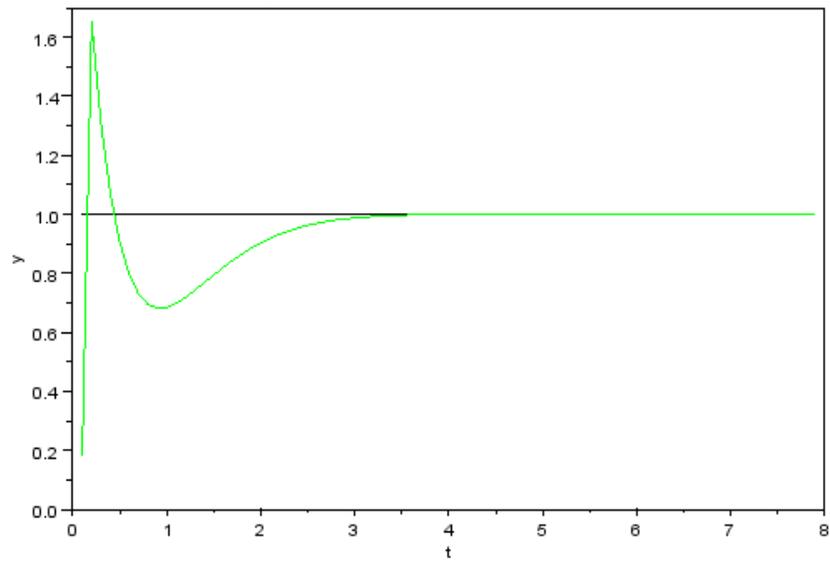


Figura 36 Resultado PID 2

8. Desarrollo guías laboratorio

8.1 EXPERIENCIA 1: Familiarización con el simulador SIMULINK de MATLAB

Práctica 1: Acceder a Simulink

Entrar a Matlab
Digitar Simulink
Abrir un proyecto nuevo
Observar las opciones disponibles

Práctica 2: Simulación de un sistema analógico

Construir un diagrama de simulación que incluya entrada, sistema analógico de segundo orden realimentado y graficador.

Estudiar la opción “Parámetros de simulación” y ejecutar la simulación con entrada de paso para varios valores de tiempo se simulación.

Ejecutar la simulación del sistema para entradas de posición, velocidad y aceleración.

Esquema de Simulación:

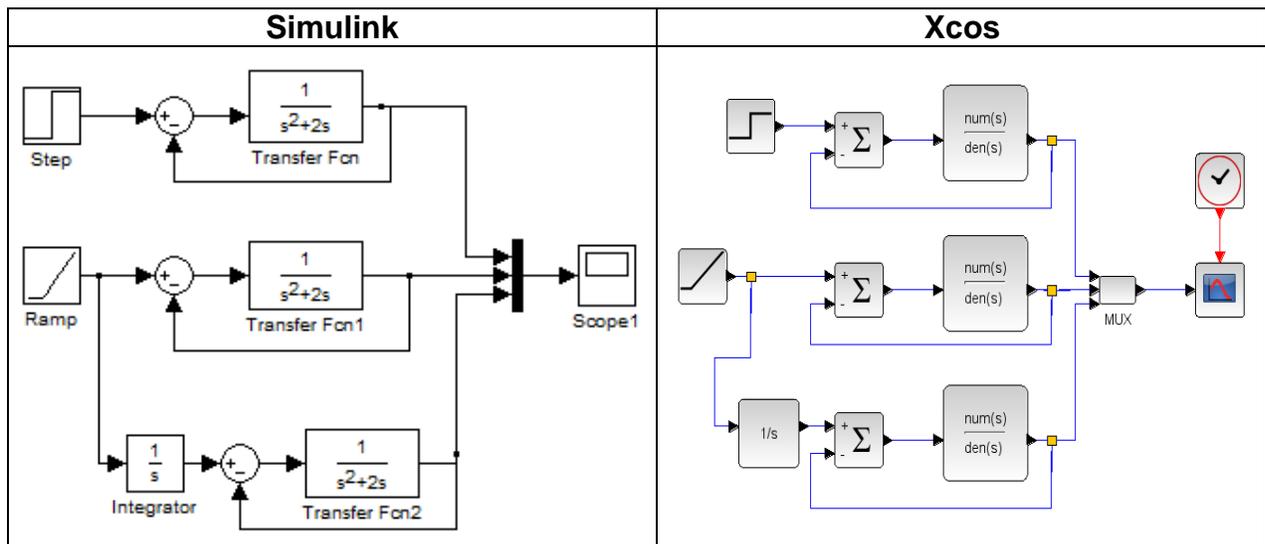
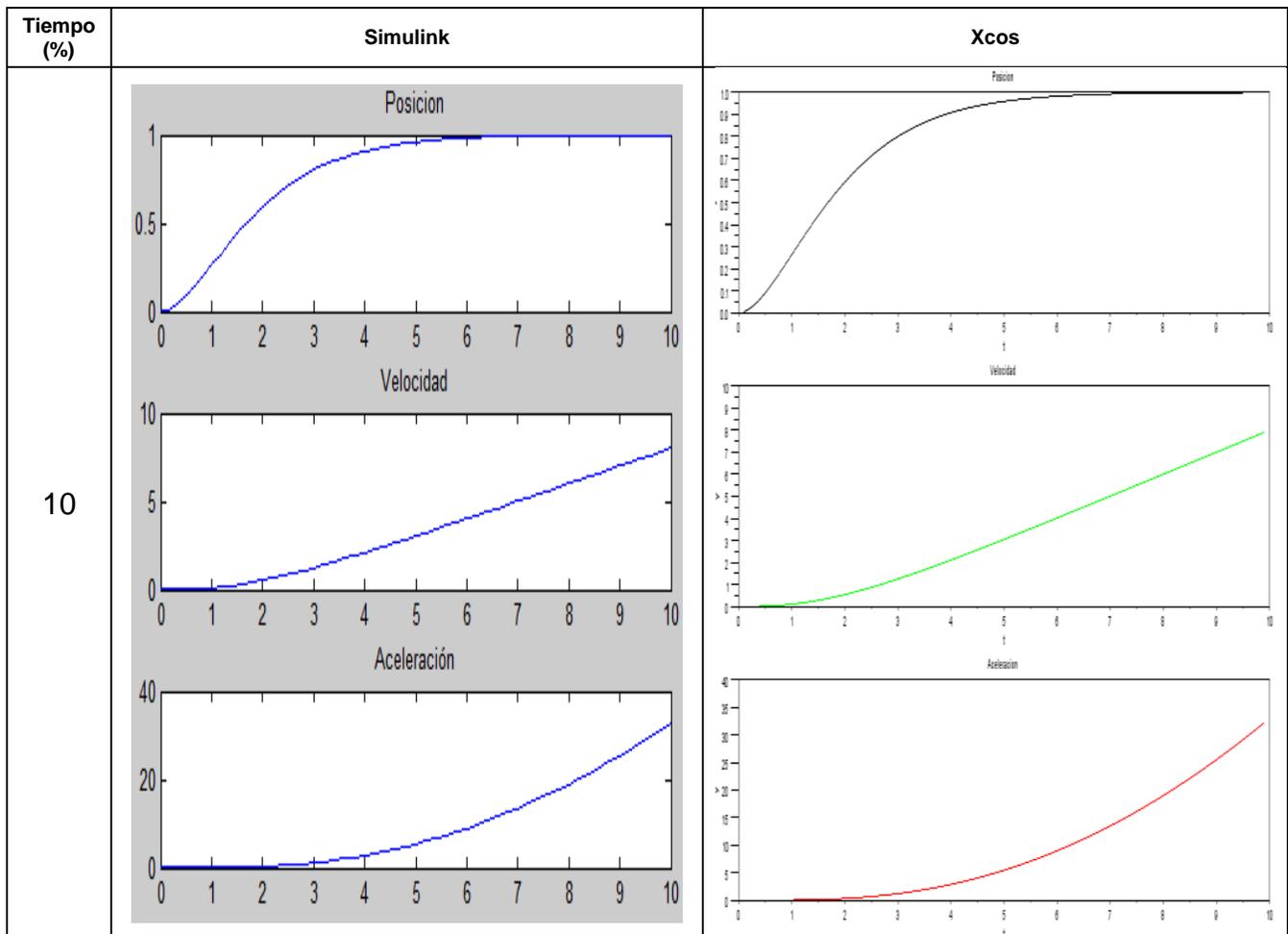


Figura 37 Esquema simulación práctica 1

Resultados, Gráficas y Análisis:



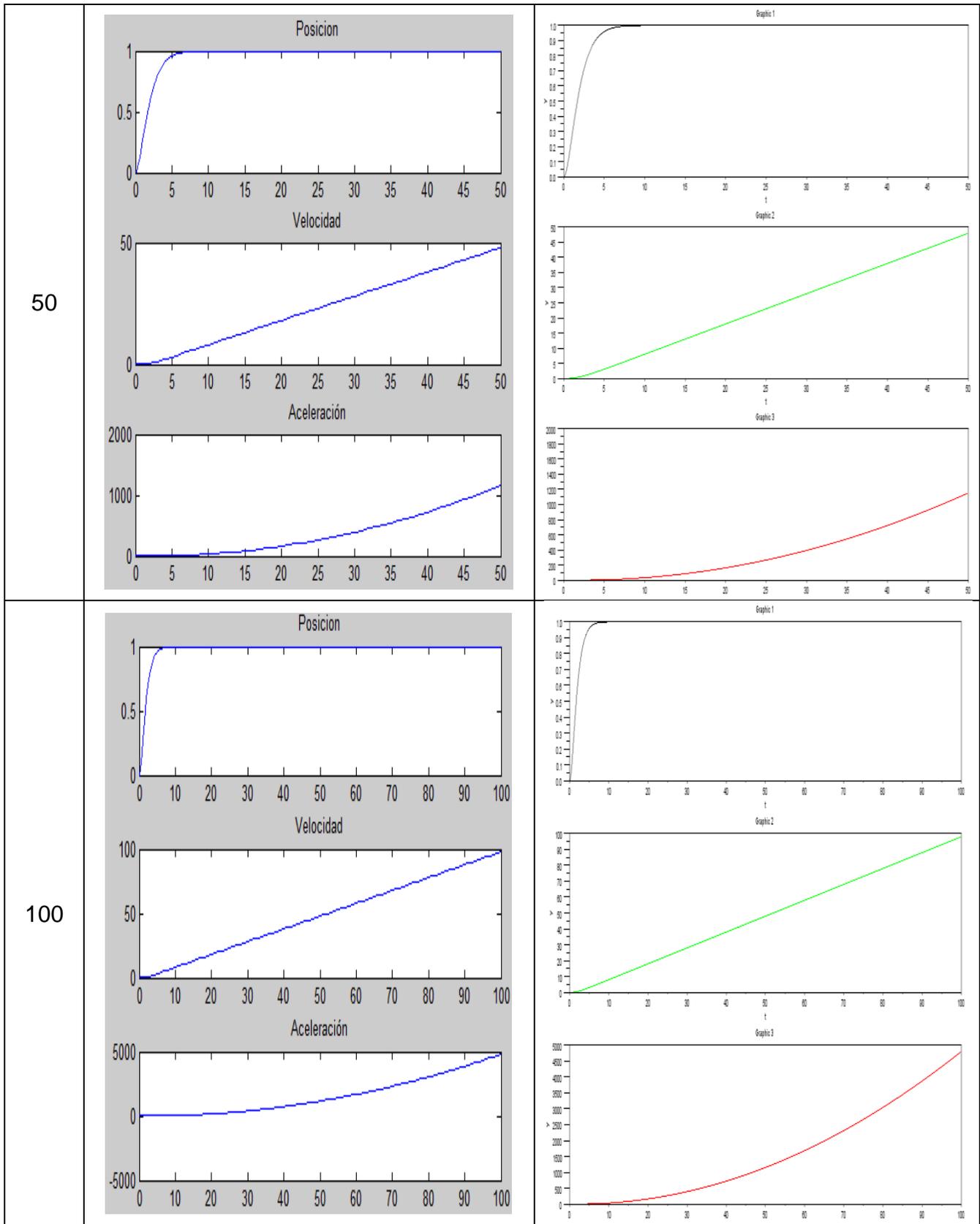


Tabla 3 Respuesta de los sistemas vs tiempo

Práctica 3: Construir un diagrama de simulación que incluya entrada de paso, sistema digital realimentado y graficado.

Ejecutar la simulación para varios valores del período de muestreo y entrada de posición.

Esquema de Simulación:

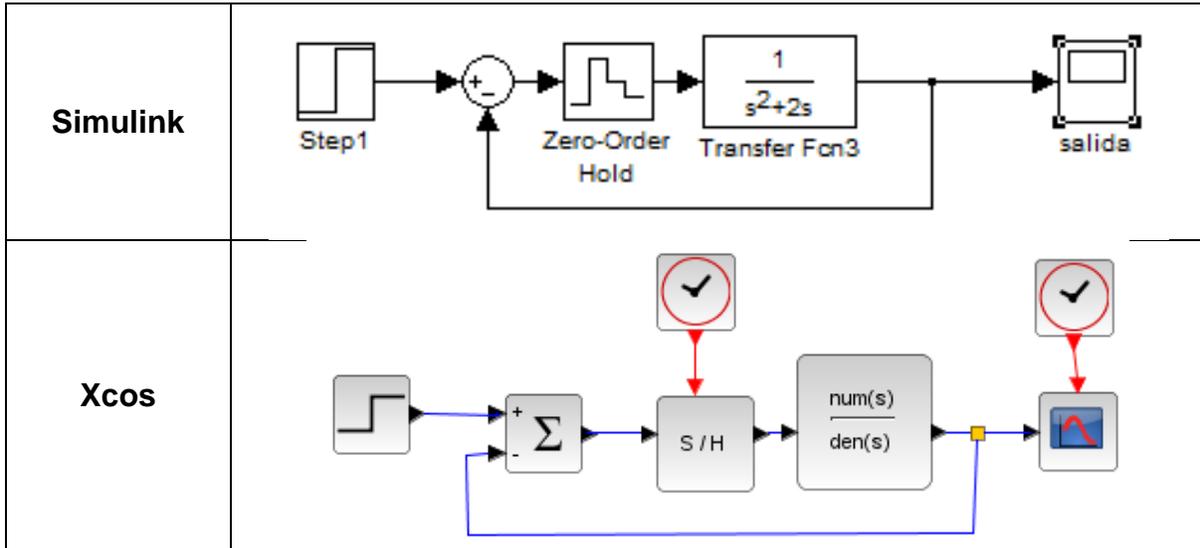
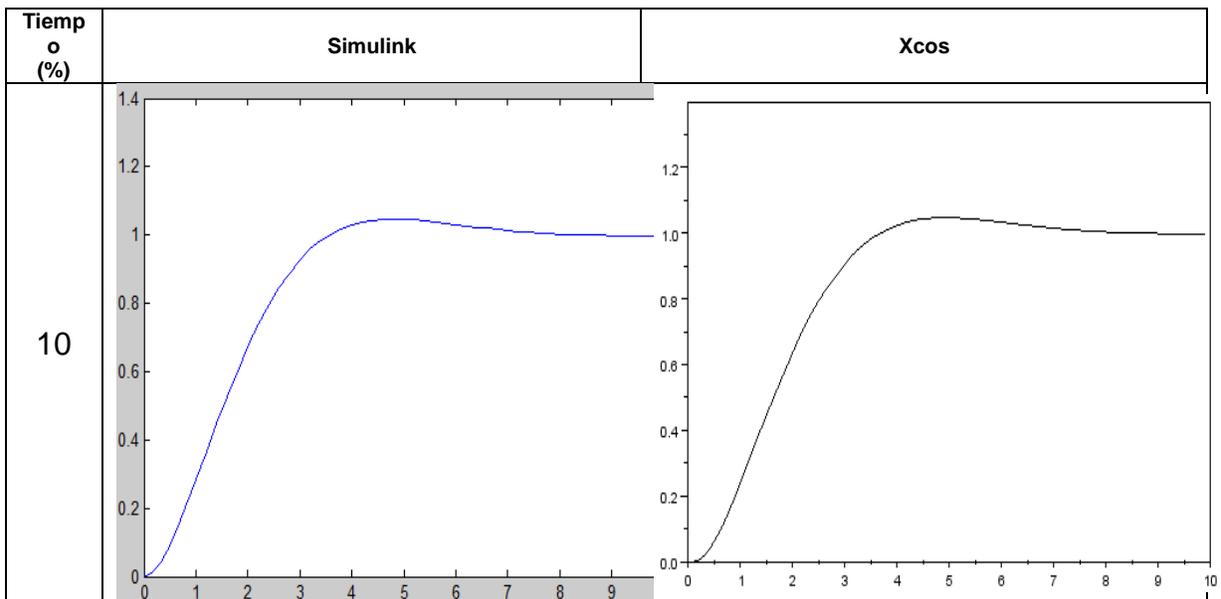


Figura 38 Esquemático practica 1.3

Resultados, Gráficas y Análisis:



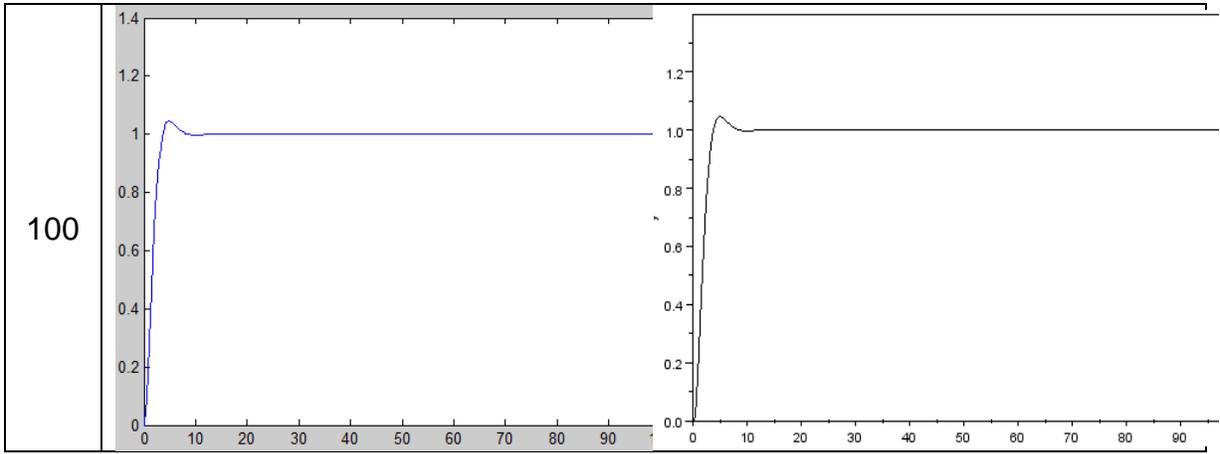
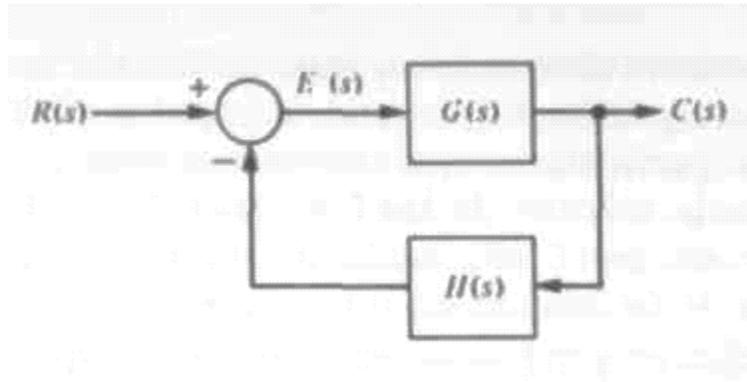


Tabla 4 Respuesta de los sistemas vs tiempo

8.2 EXPERIENCIA 2: Comportamiento de Sistemas de Control

Práctica Nº 1: Respuesta transitoria - Porcentaje de overshoot y Tiempo de asentamiento

Se tiene el sistema de la figura siguiente:



K

Donde $G(s) = \frac{K}{s(s+a)}$ y $H(s) = 1$

Para $R(s) = 1/s$

Simular el sistema de control de la figura para $K = 1$ y $\zeta = 2, 1, 0,7, 0,4, 0,2, 0,1$ Y $0,0$.

i) Hallar el Porcentaje de Overshoot (PO) para los valores del coeficiente de amortiguación (ζ) simulados. Dibujar la gráfica $PO = f(\zeta)$.

ii) Hallar el Tiempo de Asentamiento (t_s) para los valores del coeficiente de amortiguación (ζ) simulados. Dibujar la gráfica $t_s = f(\zeta)$.

Esquema de Simulación:

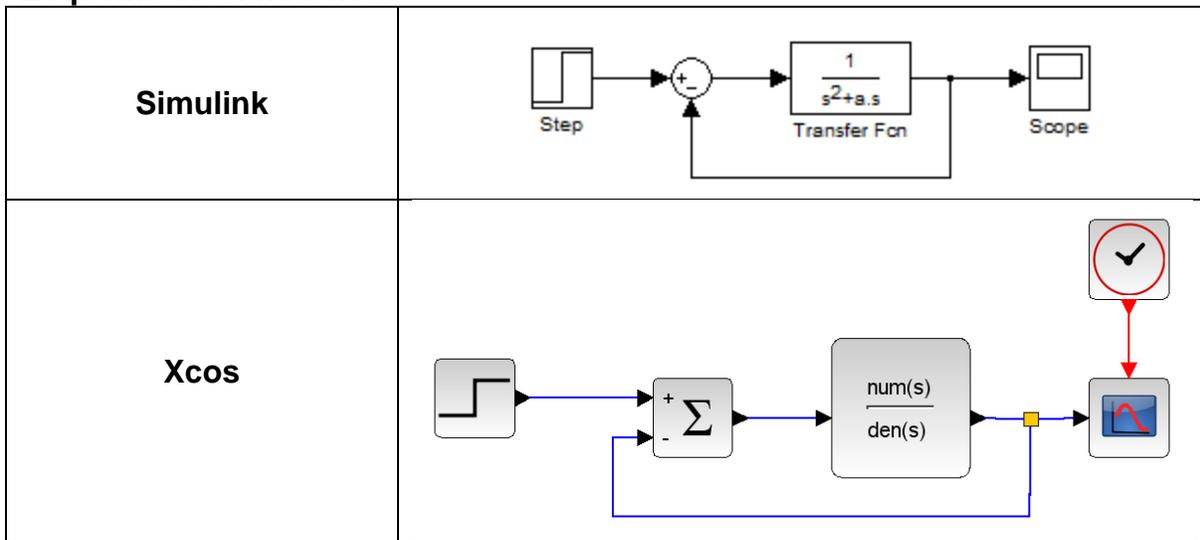
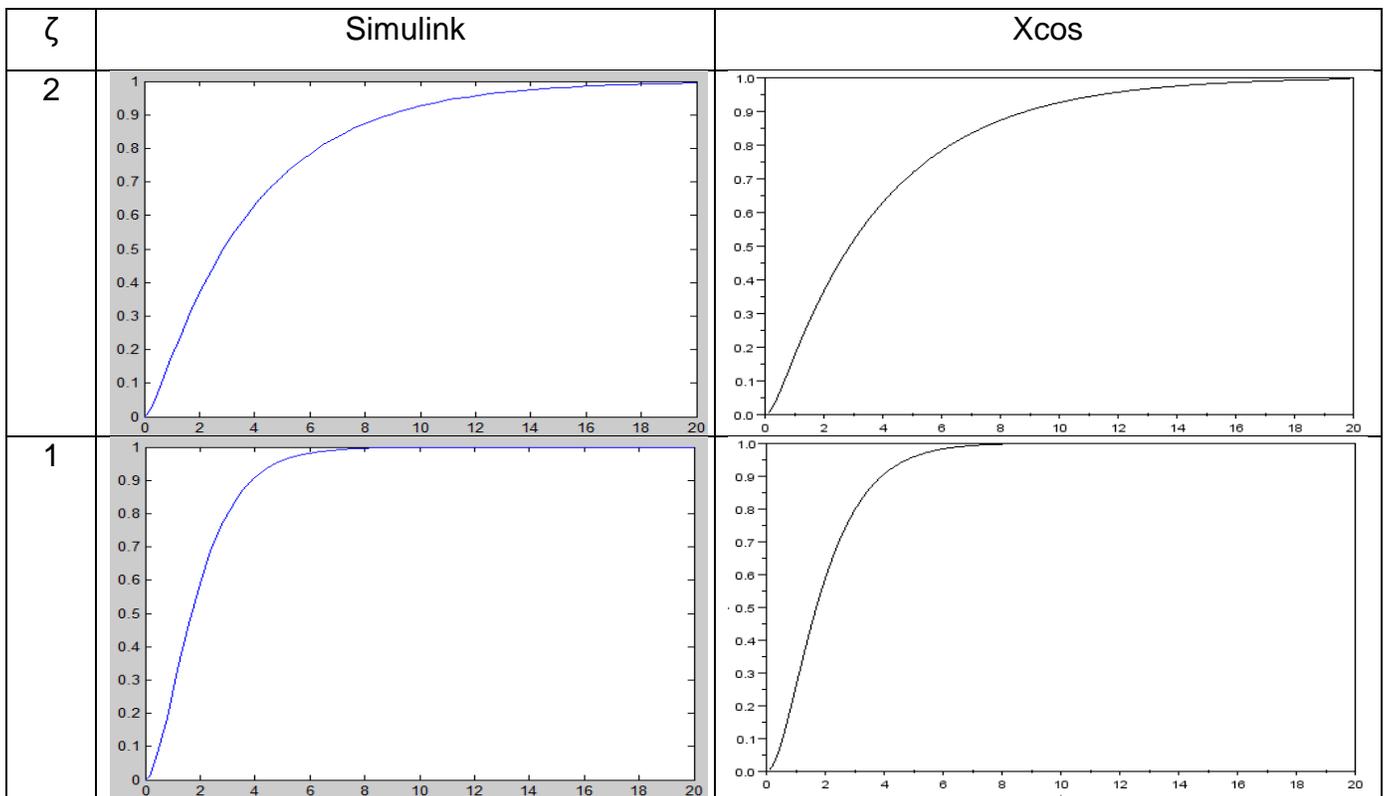
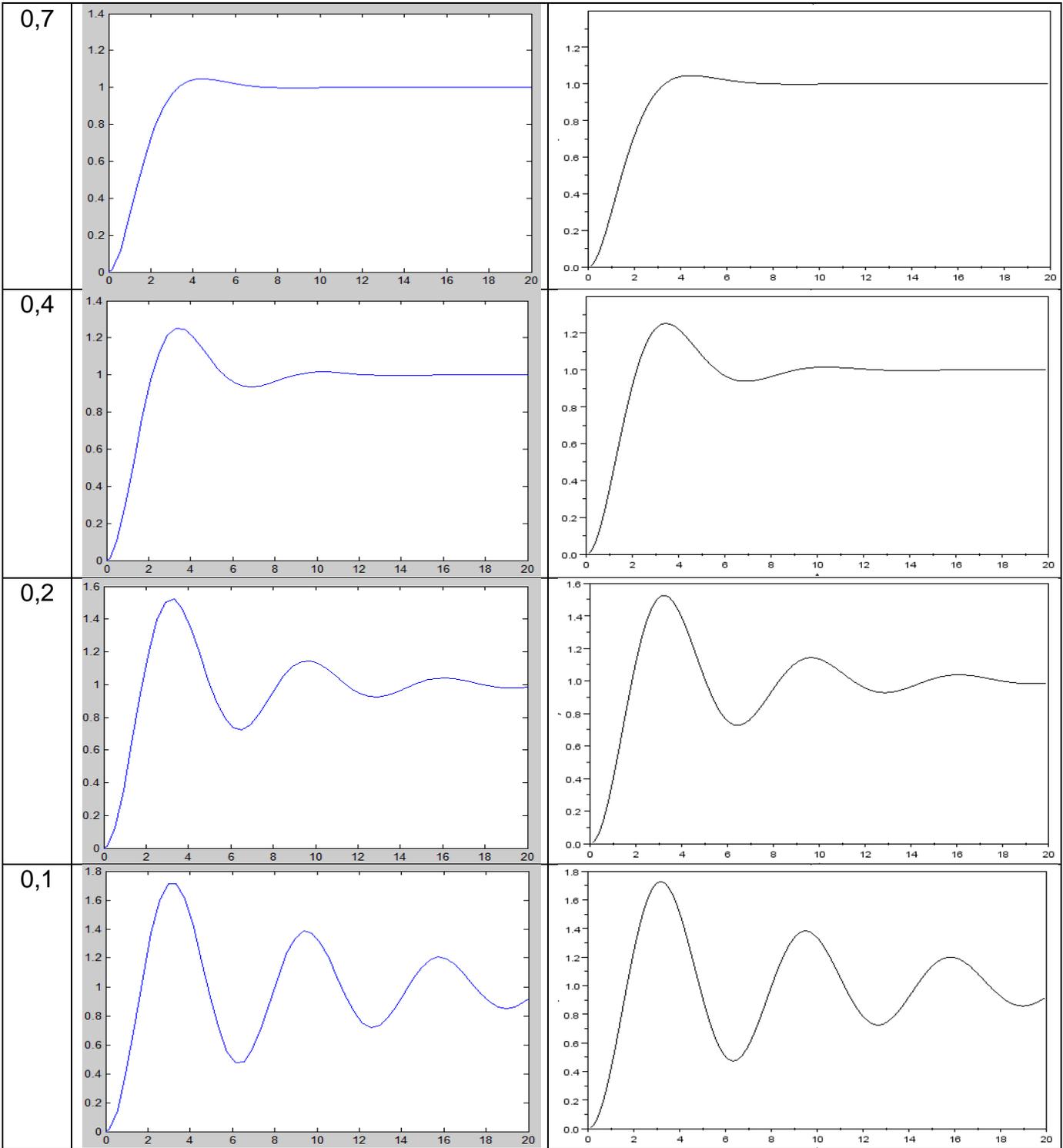


Figura 39 Esquemático simulación práctica 2

Resultados:





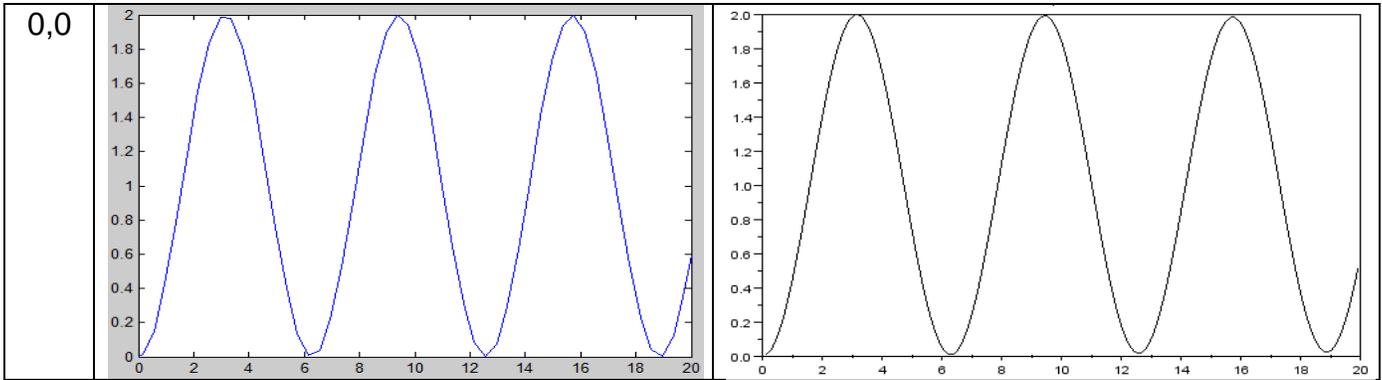


Tabla 5 Porcentaje de Overshoot (PO) del sistema de segundo orden vs el Coeficiente de Amortiguación (ζ)

Práctica 2: Respuesta en Estado estacionario - Error de posición, de velocidad y de aceleración de un sistema analógico de segundo orden

- i) Construir el diagrama de simulación del sistema de control realimentado de la figura anterior Con $G(s)$ (Función de la planta del lazo abierto) es igual a $G(s) = K/[s(s + a)]$ y $H(s) = 1$ (Realimentación unitaria).
- ii) Hallar, usando simulación en Simulink, los errores de posición, velocidad y aceleración del sistema para los valores de $K = 0.5, 1.0, 2.0, 5.0$ y 10.0

Esquema de Simulación:

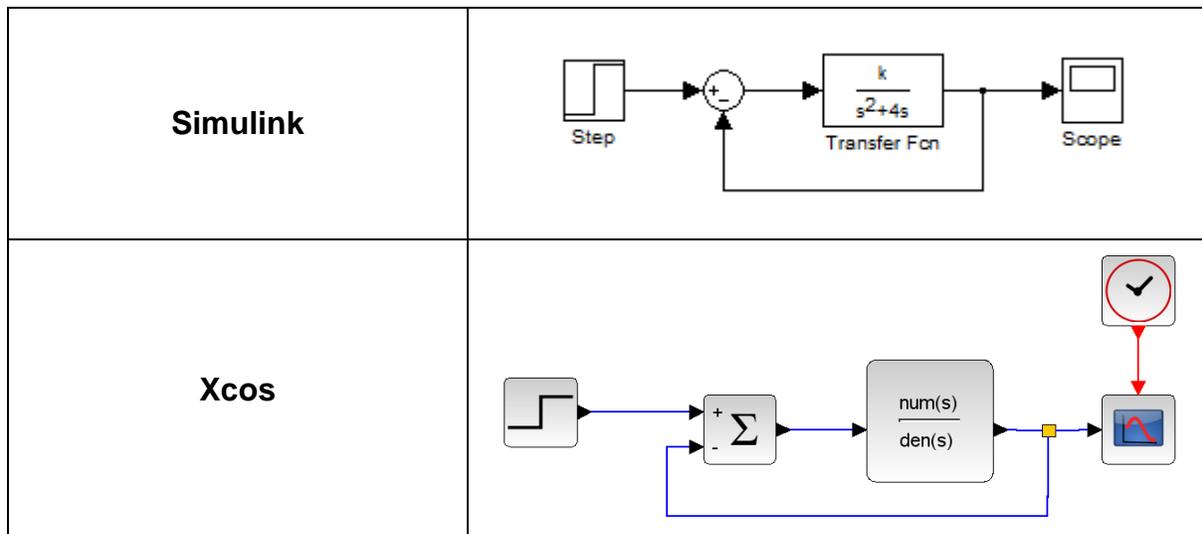
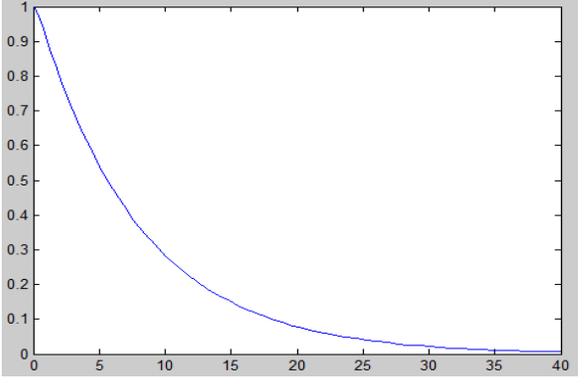
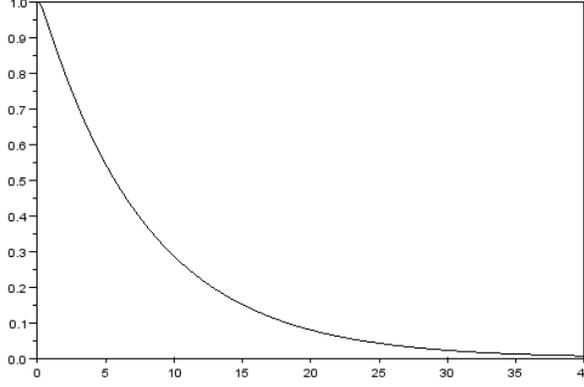
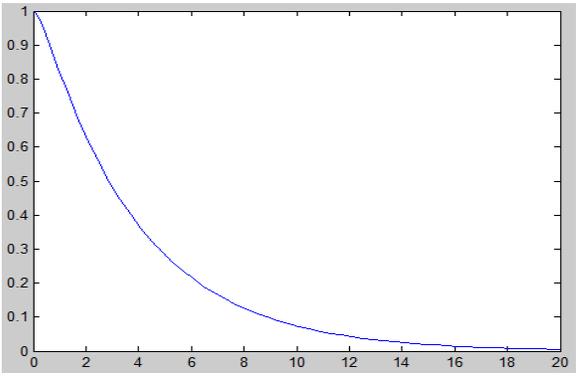
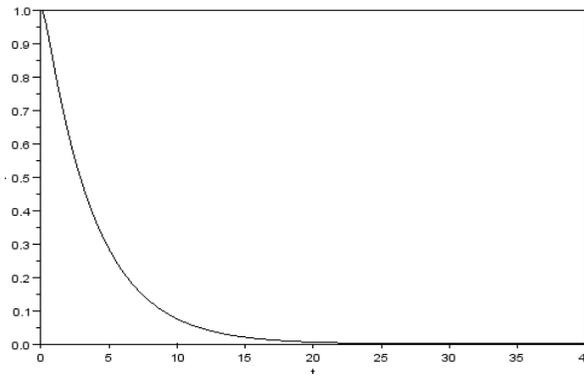
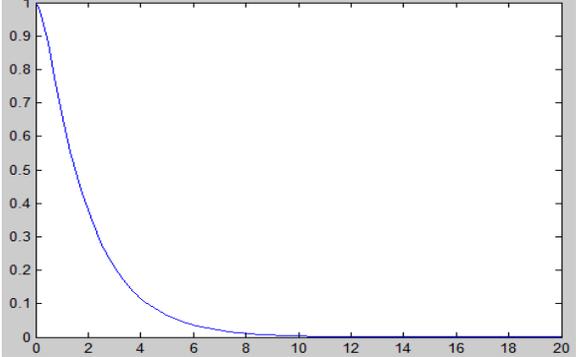
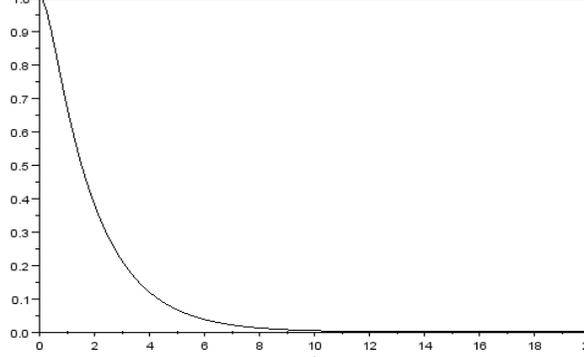
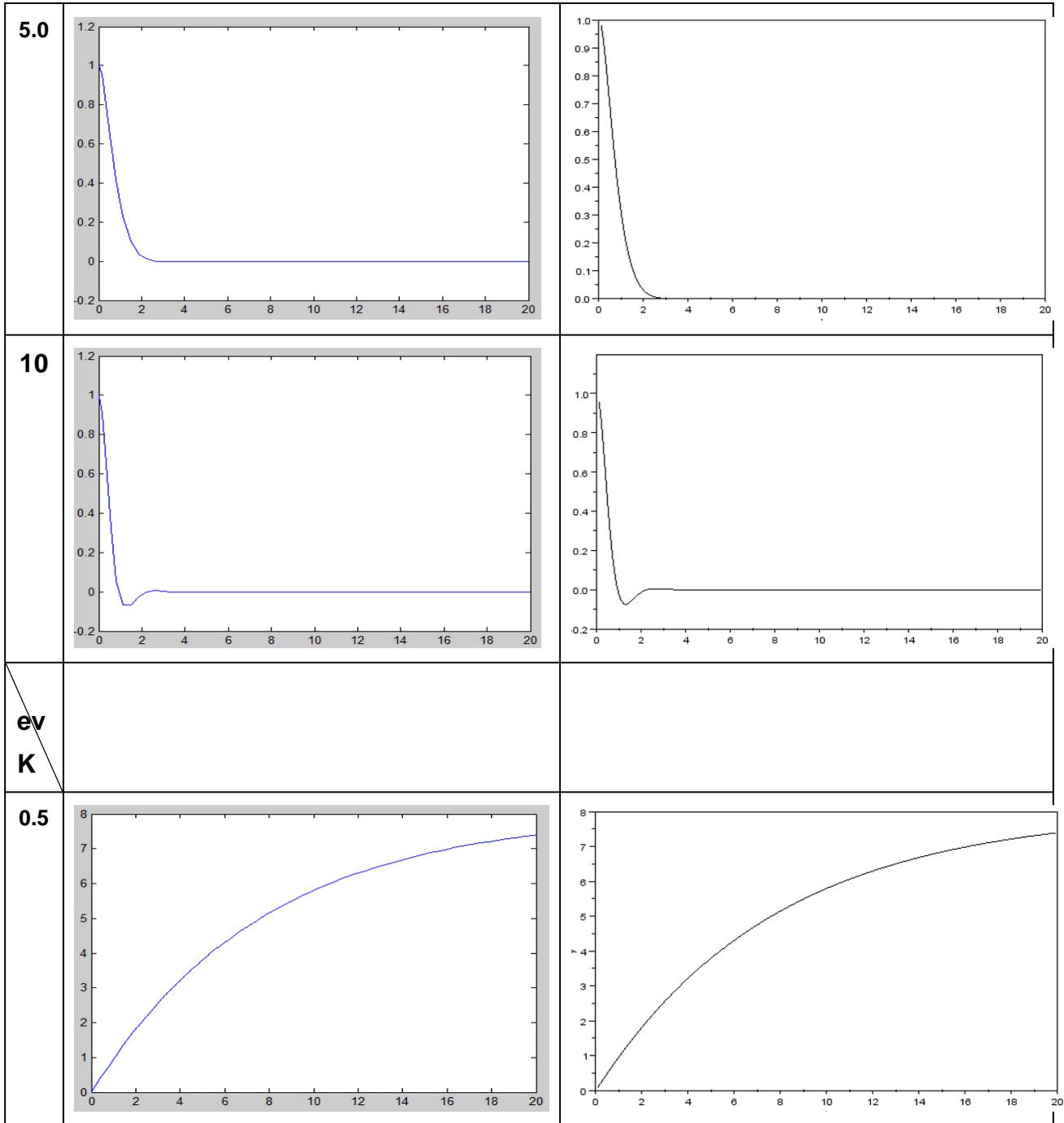
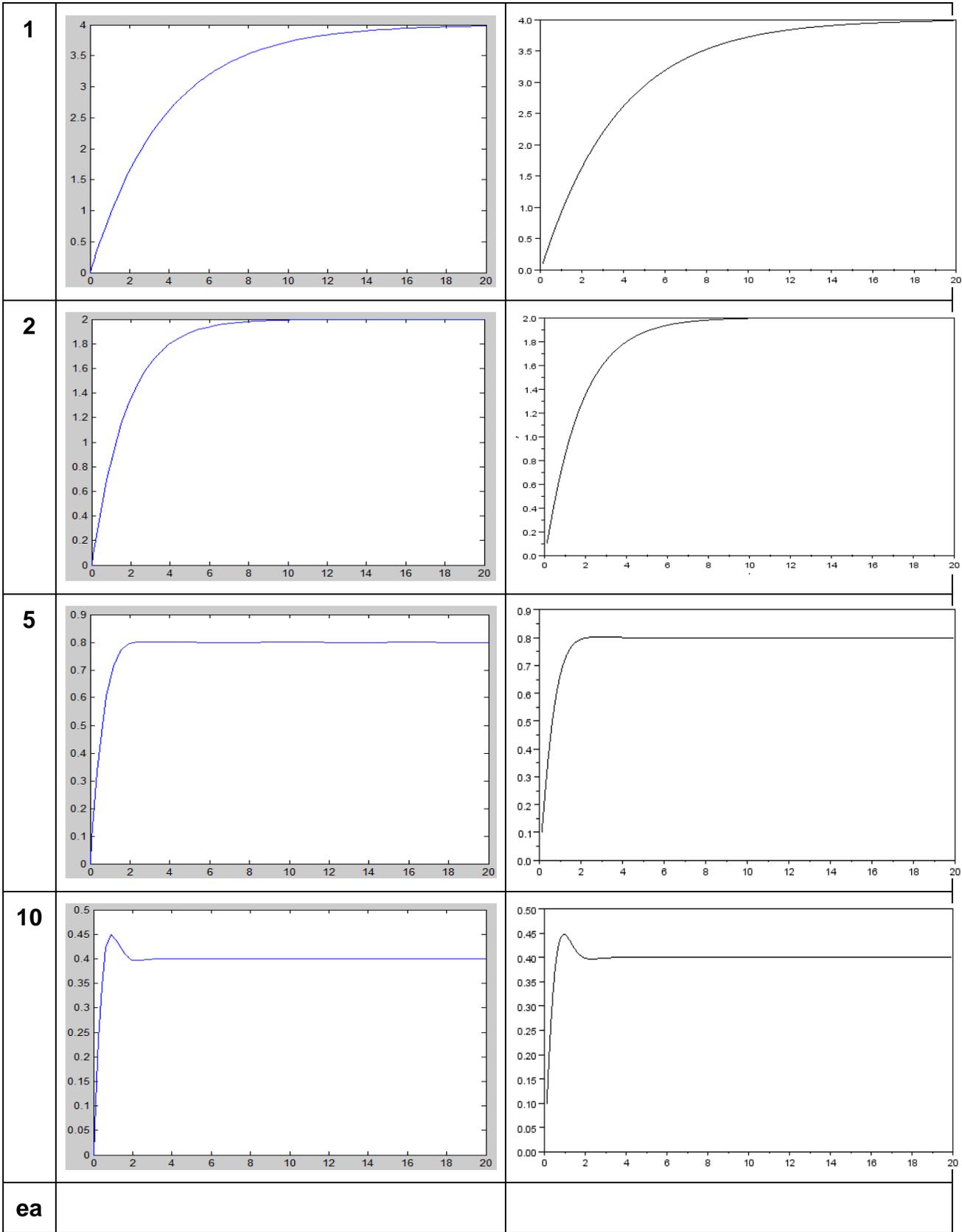


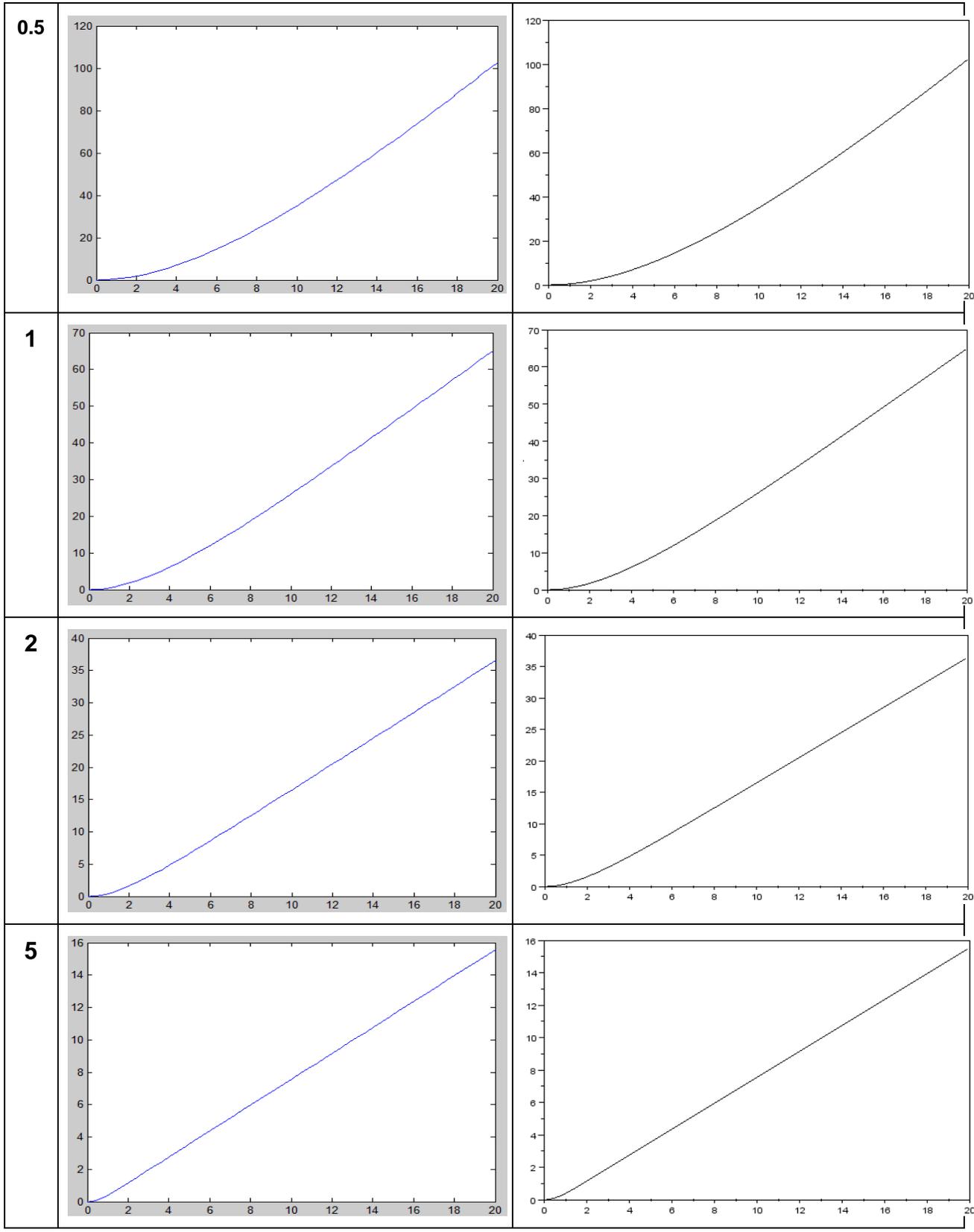
Figura 40 Esquemático simulación práctica 2.2

Resultados:

e_p k	Simulink	Xcos
0.5		
1.0		
2.0		







10

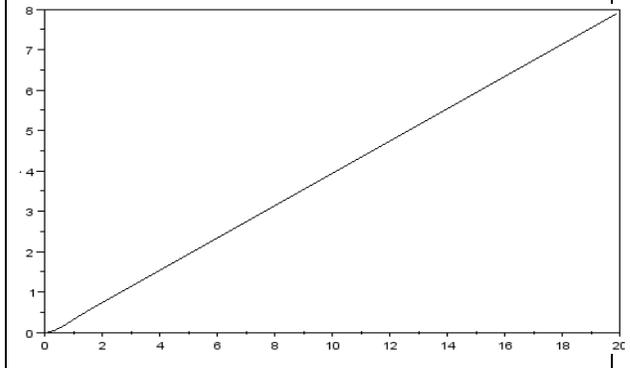
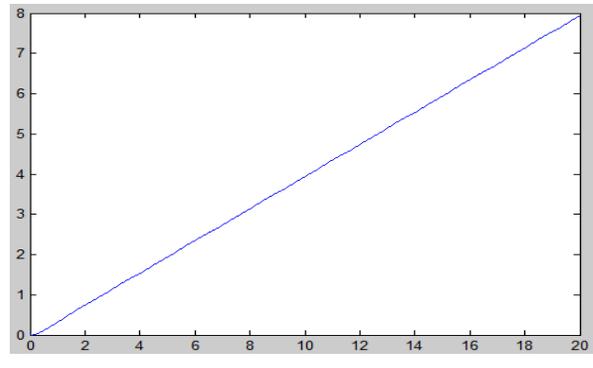


Tabla 6 Errores de Posición, Velocidad y Aceleración vs Ganancia K

8.3 EXPERIENCIA 2: Estado Transitorio y Efecto del Período de Muestreo sobre la Estabilidad de Sistemas Discretos

Práctica 1:

a) Simular un sistema de control con realimentación unitaria, de segundo orden, de primer tipo, con planta de ganancia tres (3) y polo en (-1), con y sin muestreador retenedor de orden cero antes de la planta, con período de muestreo de 0,1, 0,3, 0,5, 0,7 y 0,9 seg. Evaluar el porcentaje de overshoot y el tiempo de asentamiento del sistema analógico y del sistema discreto para los períodos de muestreo dados.

Esquemas de Simulación:

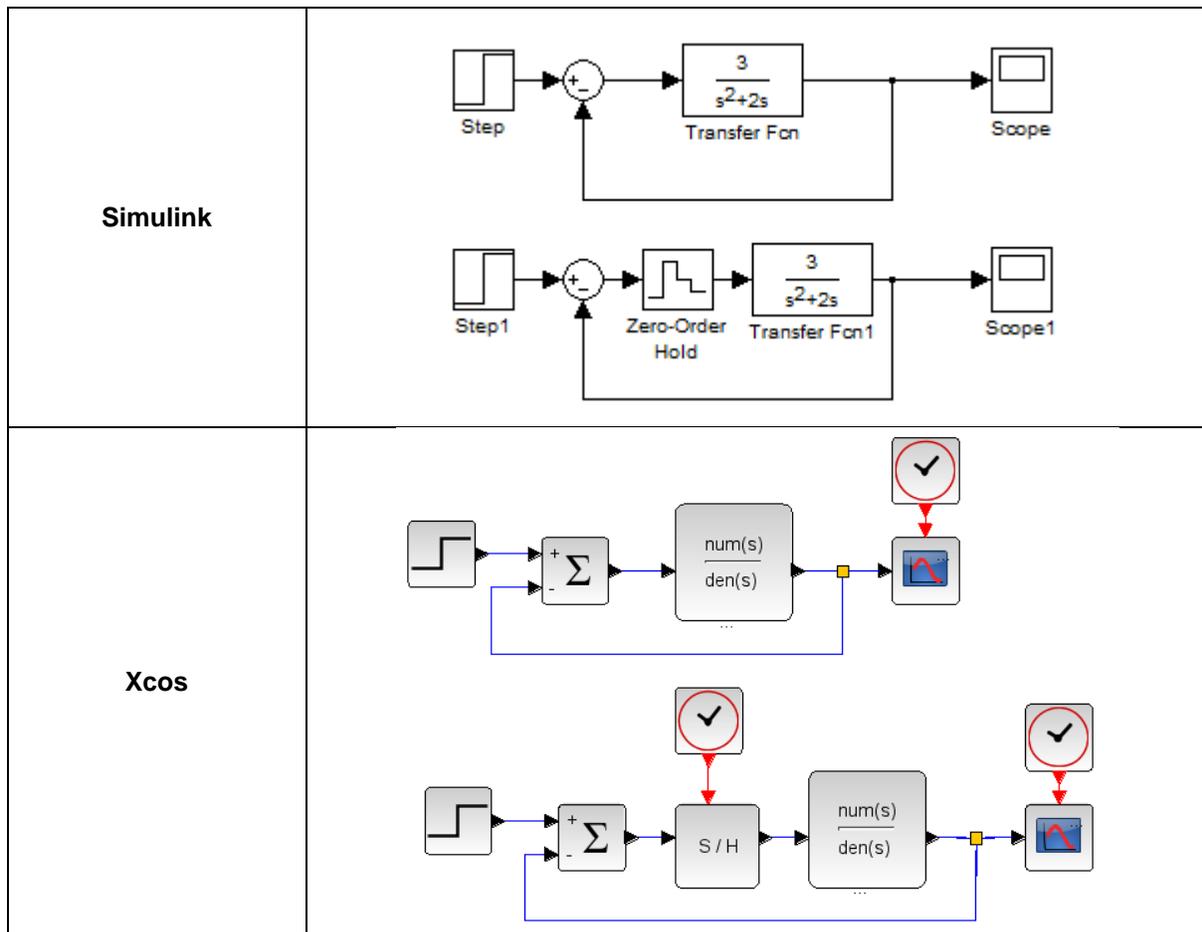
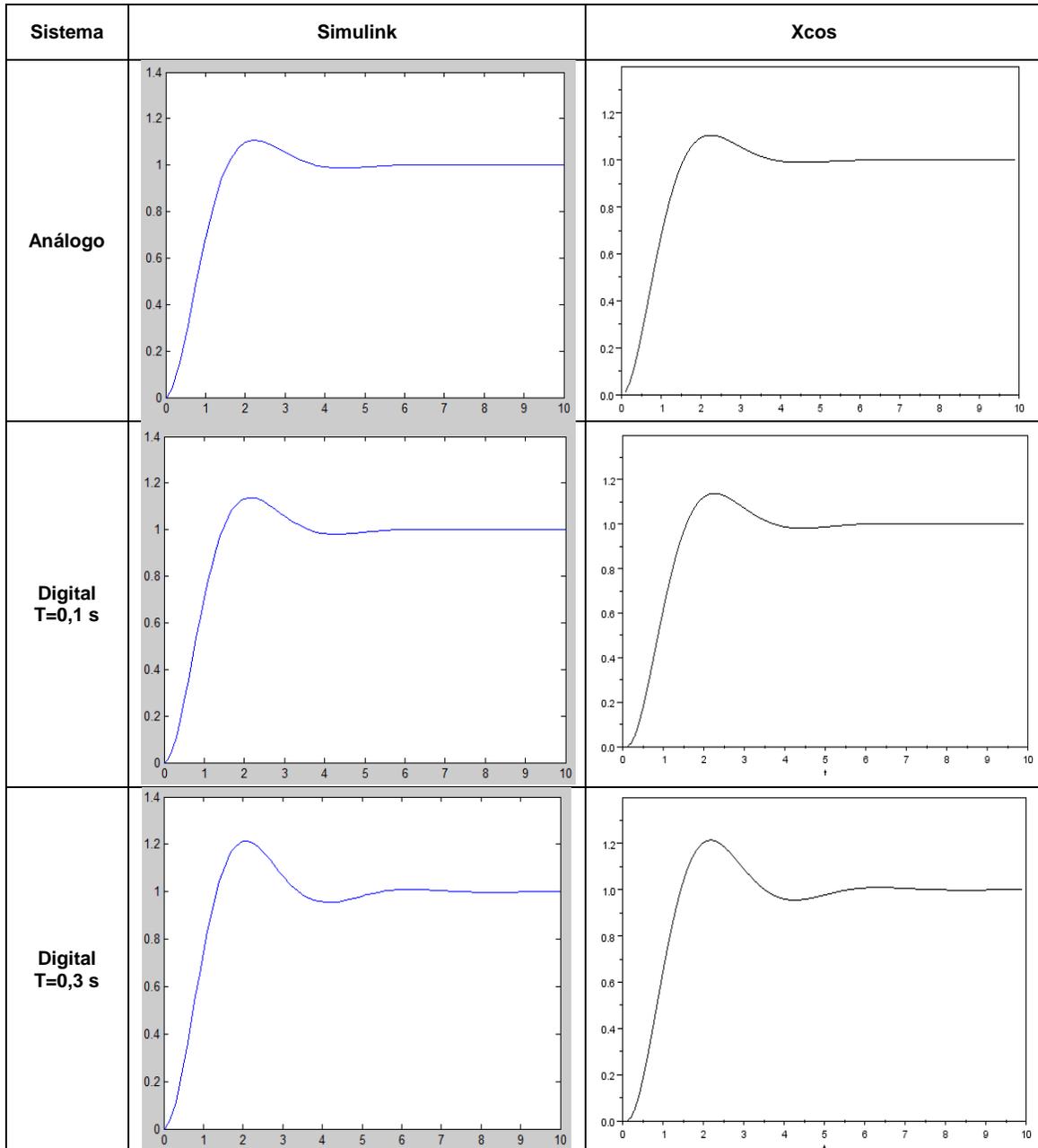


Figura 41 Esquema simulación practica 2

Resultados y Análisis:



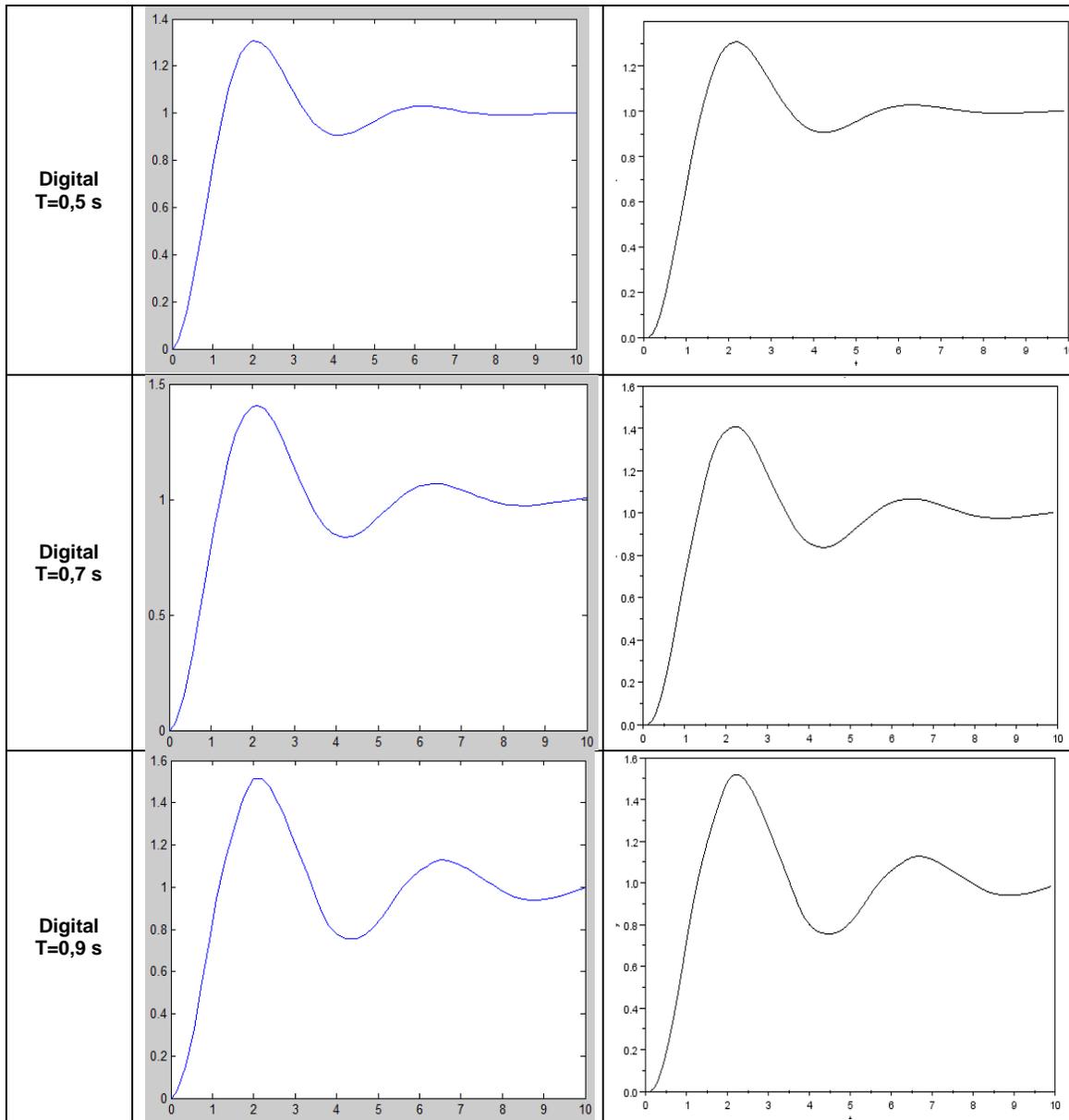


Tabla 7 Porcentaje de Overshoot (PO) vs Período de Muestreo

Práctica 2: Hallar el período de muestreo que hace inestable un sistema de control digital

a) Se tiene un sistema de control con realimentación unitaria, de segundo orden, de primer tipo, con planta de ganancia tres (3) y polo en (-1) y entrada de paso unitario.

Simular el sistema anterior adicionándole muestreador de orden cero antes de la planta con período de muestreo inicial de 0,1. Hallar el período de muestreo que vuelve el sistema inestable

Esquemas de Simulación:

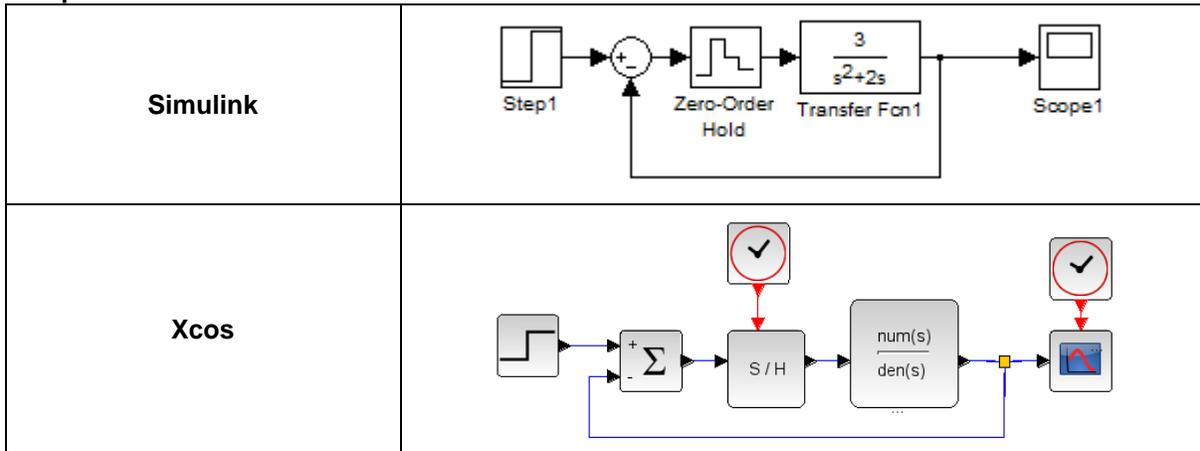


Figura 42 Esquema simulación practica 2.2

Resultados, Gráficas y Análisis:

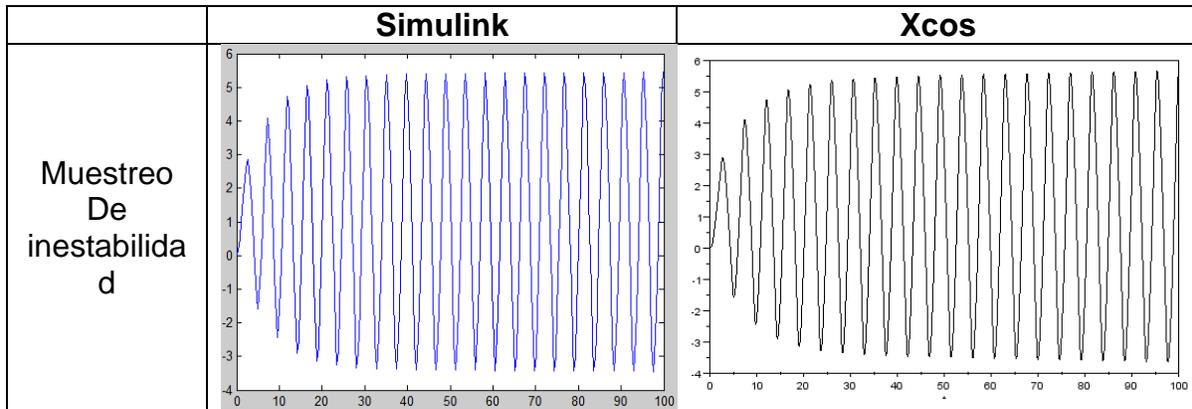


Tabla 8 Muestreo de inestabilidad

Período de muestreo que hace inestable el sistema de control discreto: 2.314

8.4 EXPERIENCIA 3: Introducción a Controladores PID de Sistemas Discretos

Práctica 1:

a) Simular un sistema de control con realimentación unitaria, de segundo orden, de primer tipo, con planta de ganancia tres (3) y polo en (-1), muestreador de orden cero antes de la planta con período de muestreo de 0,1. Evaluar el porcentaje de overshoot, el tiempo de asentamiento (t_s) y los errores de posición y velocidad.

Esquemáticos simulación

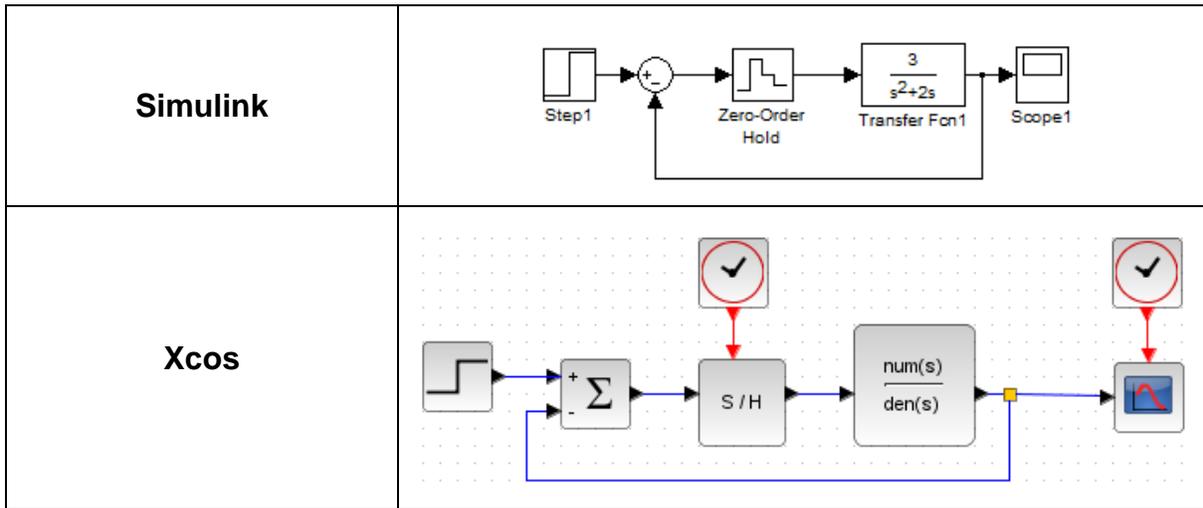
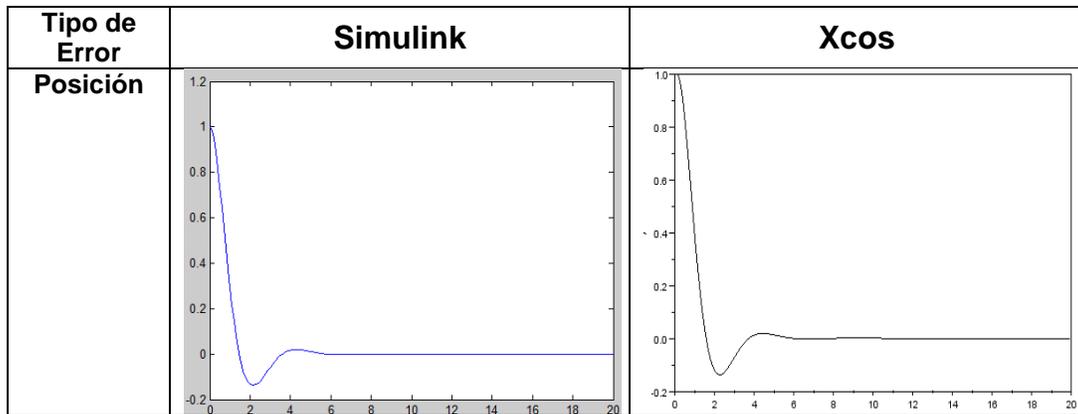


Figura 43 Esquemáticos simulación practica 3



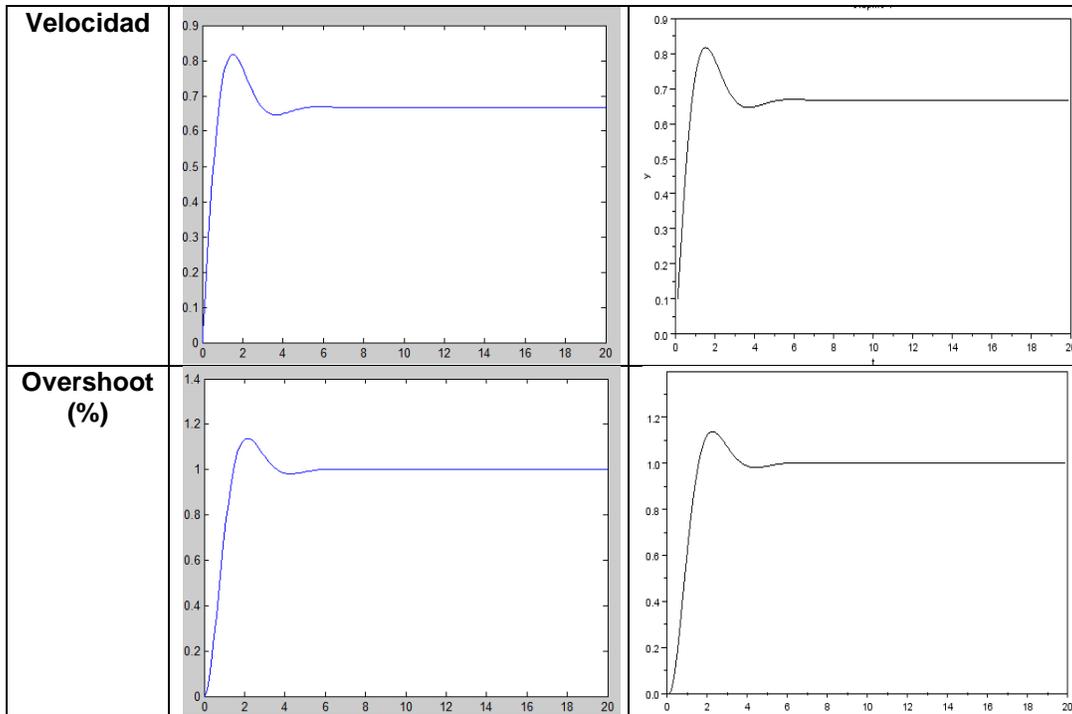


Tabla 9 Porcentaje de overshoot y Errores de posición y velocidad del sistema sin controlador digital

A continuación se muestra un algoritmo del PID que se utiliza como controlador digital.

ALGORITMO DEL P I D:

$$D(z) = \frac{M(z)}{E(z)} = K_p + K_i \frac{T}{z-1} + K_D \frac{z-1}{Tz}$$

$$D(z) = \frac{K_p(z^2 - z) + K_i(T/2)(z^2 + z) + (K_D/T)(z^2 - 2z + 1)}{z^2 - z}$$

$$M(z)(z^2 - z) = [K_p(z^2 - z) + K_i(T/2)(z^2 + z) + (K_D/T)(z^2 - 2z + 1)]E(z)$$

$$M(z)(1 - z^{-1}) = [K_p(1 - z^{-1}) + K_i(T/2)(1 + z^{-1}) + (K_D/T)(1 - 2z^{-1} + z^{-2})]E(z)$$

$$m(k) = m(k-1) + [K_p + K_i T/2 + K_D/T]e(k) + [K_i T/2 - K_p - 2K_D/T]e(k-1) + K_D/T e(k-2)$$

b) Adicionar al sistema del punto (a) la función de transferencia $D(z)$. Estudiar el efecto de K_p con K_i y K_D iguales a cero. Evaluar el porcentaje de overshoot, el tiempo de asentamiento (t_s) y los errores de posición y velocidad.

c) Estudiar el efecto de K_i con K_i igual a cero y K_p constante. Evaluar el porcentaje de overshoot y los errores de posición y velocidad.

K_i y K_p constantes. Evaluar el porcentaje de overshoot, el tiempo de asentamiento (t_s) y los errores de posición y velocidad.

c) Estudiar el efecto de K_D con K_i y K_p constantes. Evaluar el porcentaje de overshoot, el tiempo de asentamiento (t_s) y los errores de posición y velocidad.

Esquemas de Simulación:

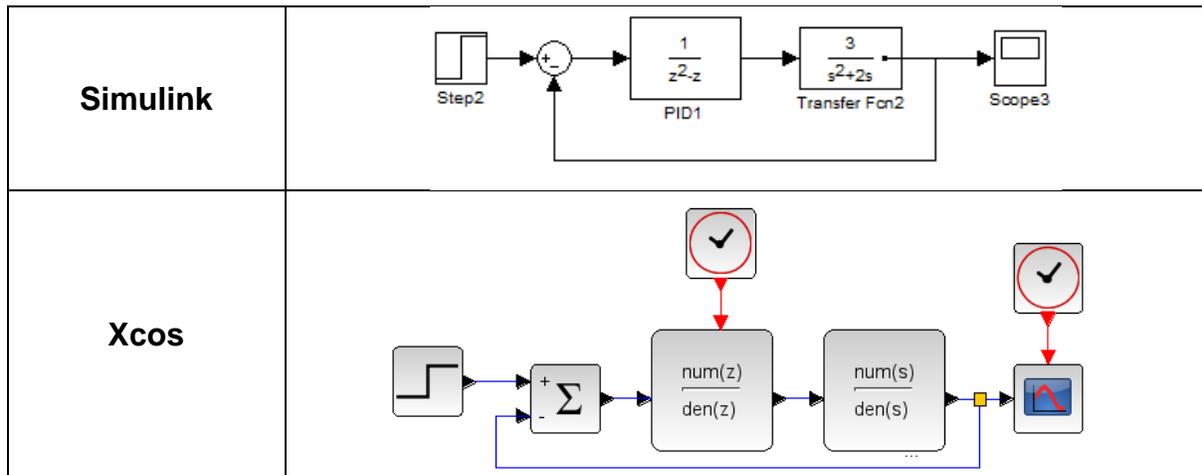
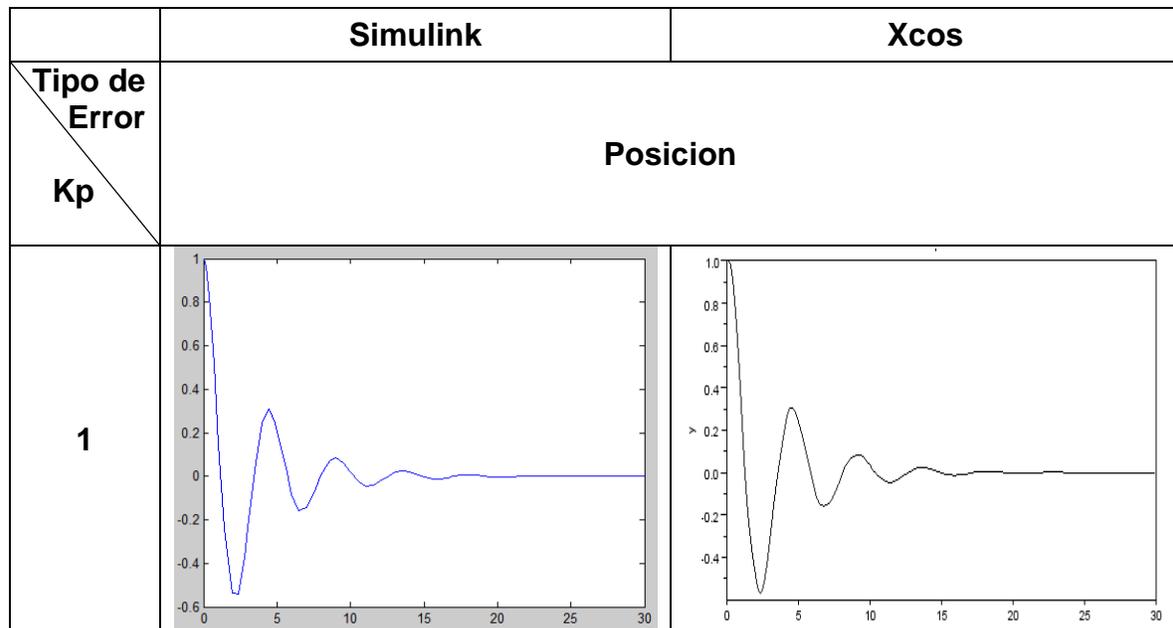
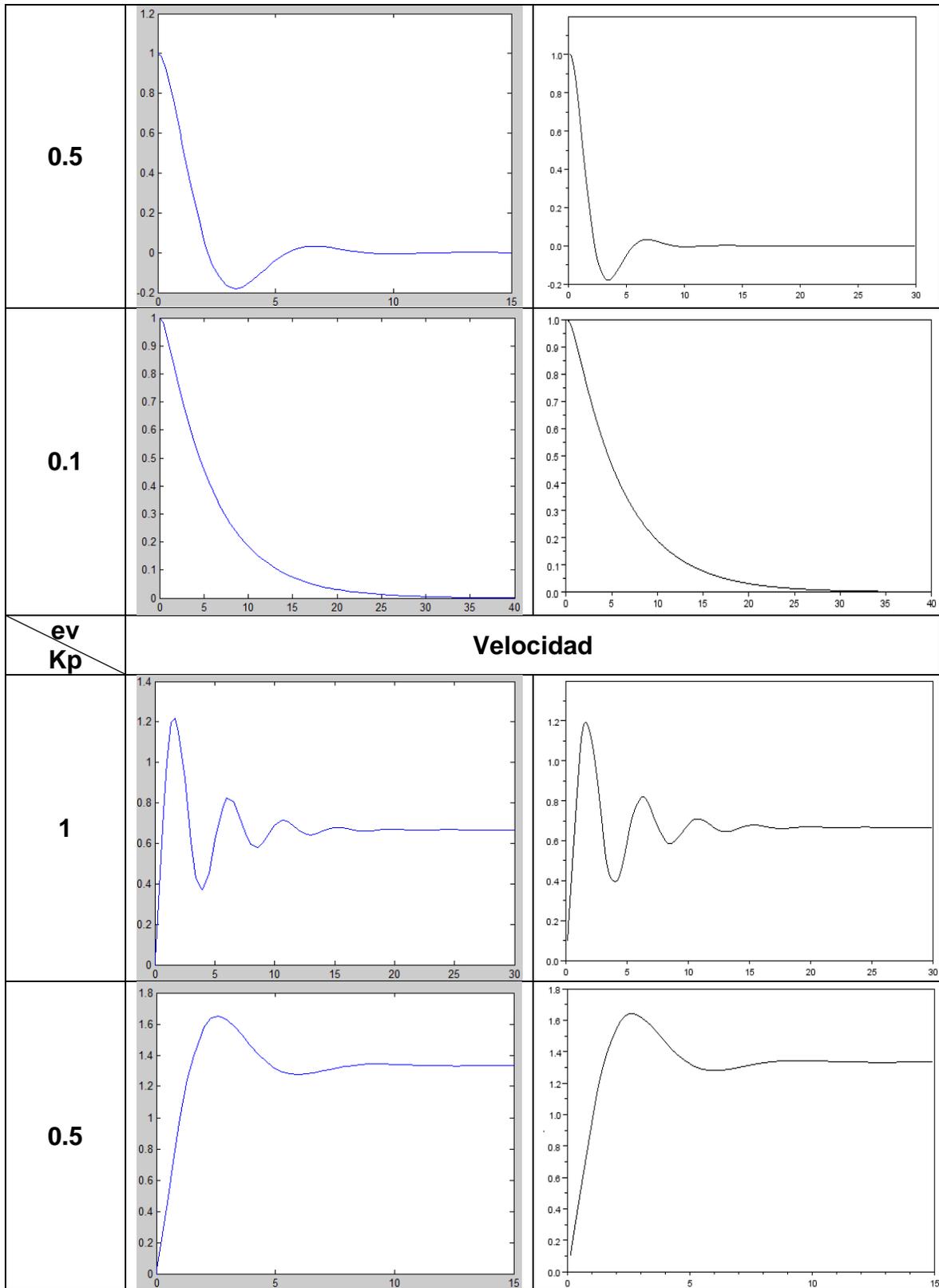


Figura 44 Esquema de simulación práctica 3

Resultados y Análisis:





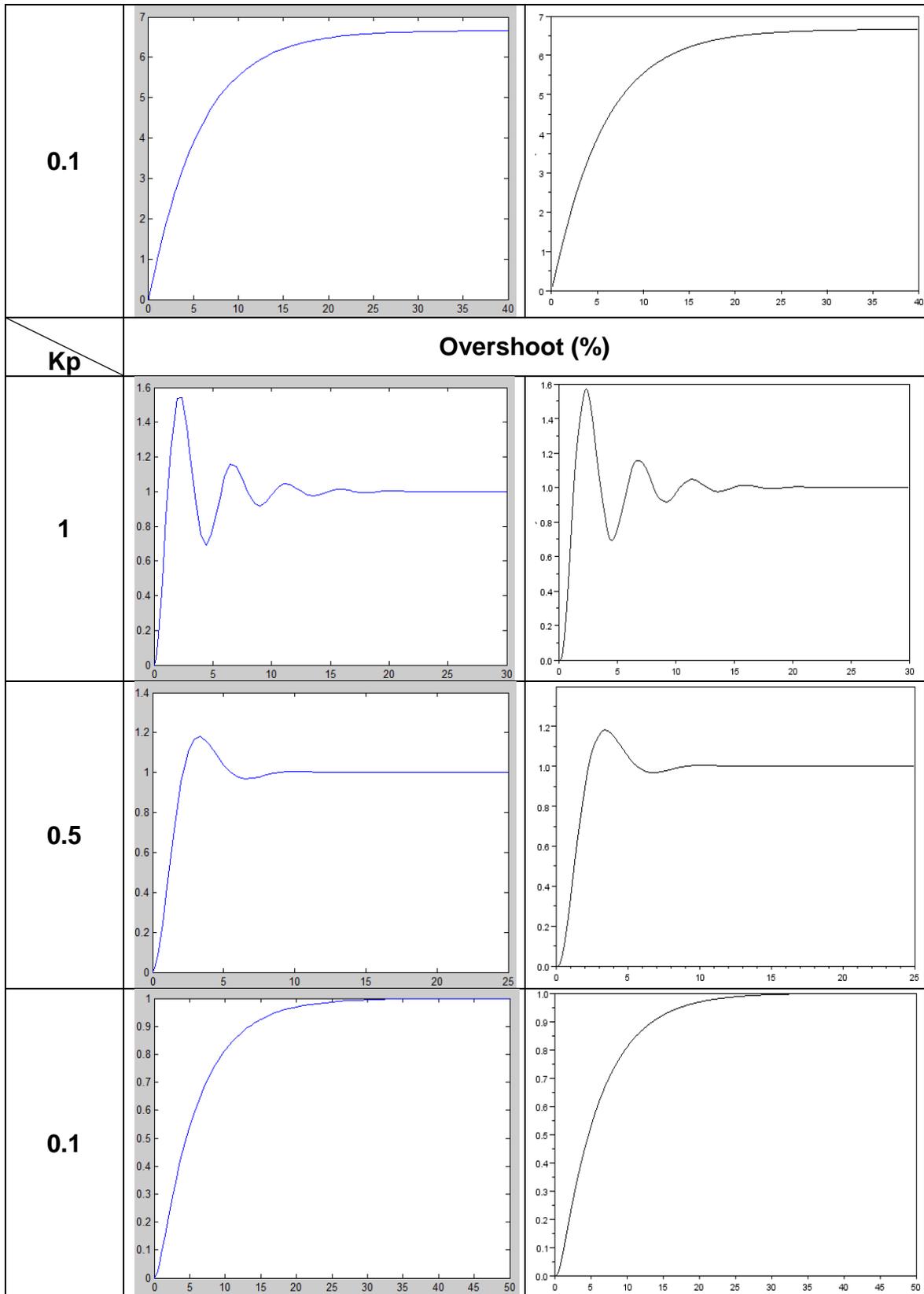
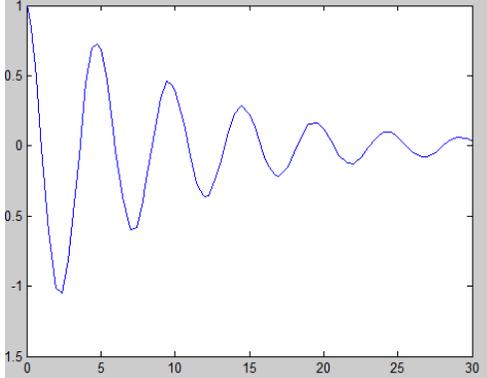
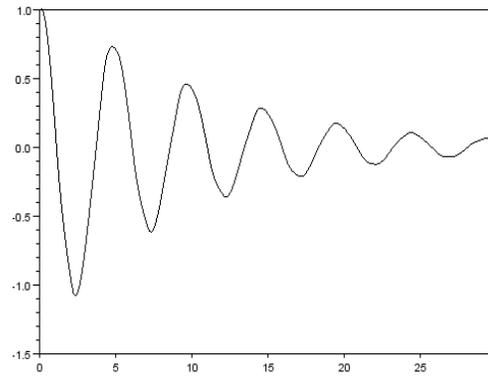
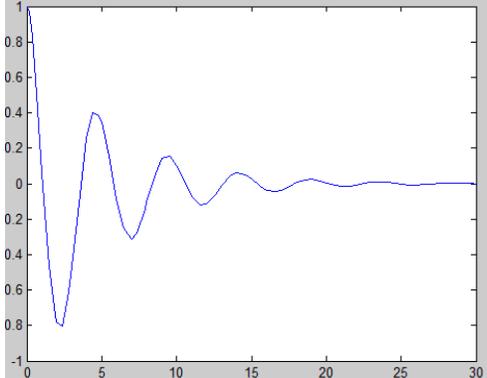
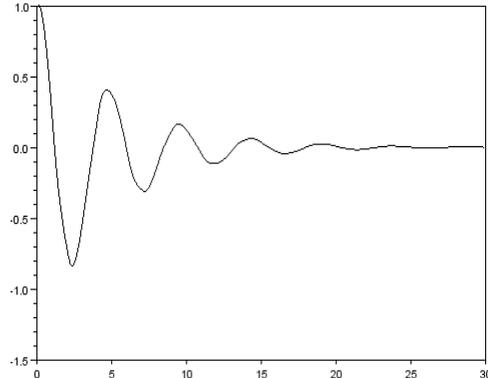
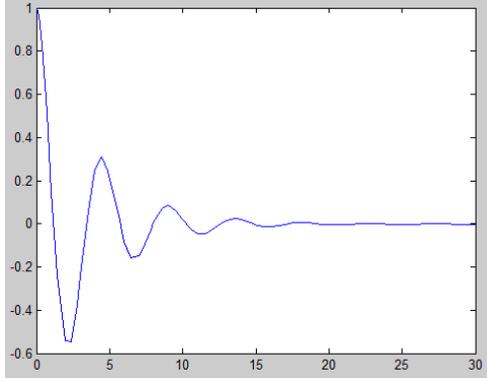
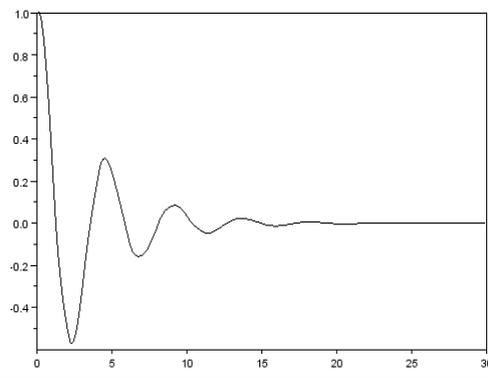
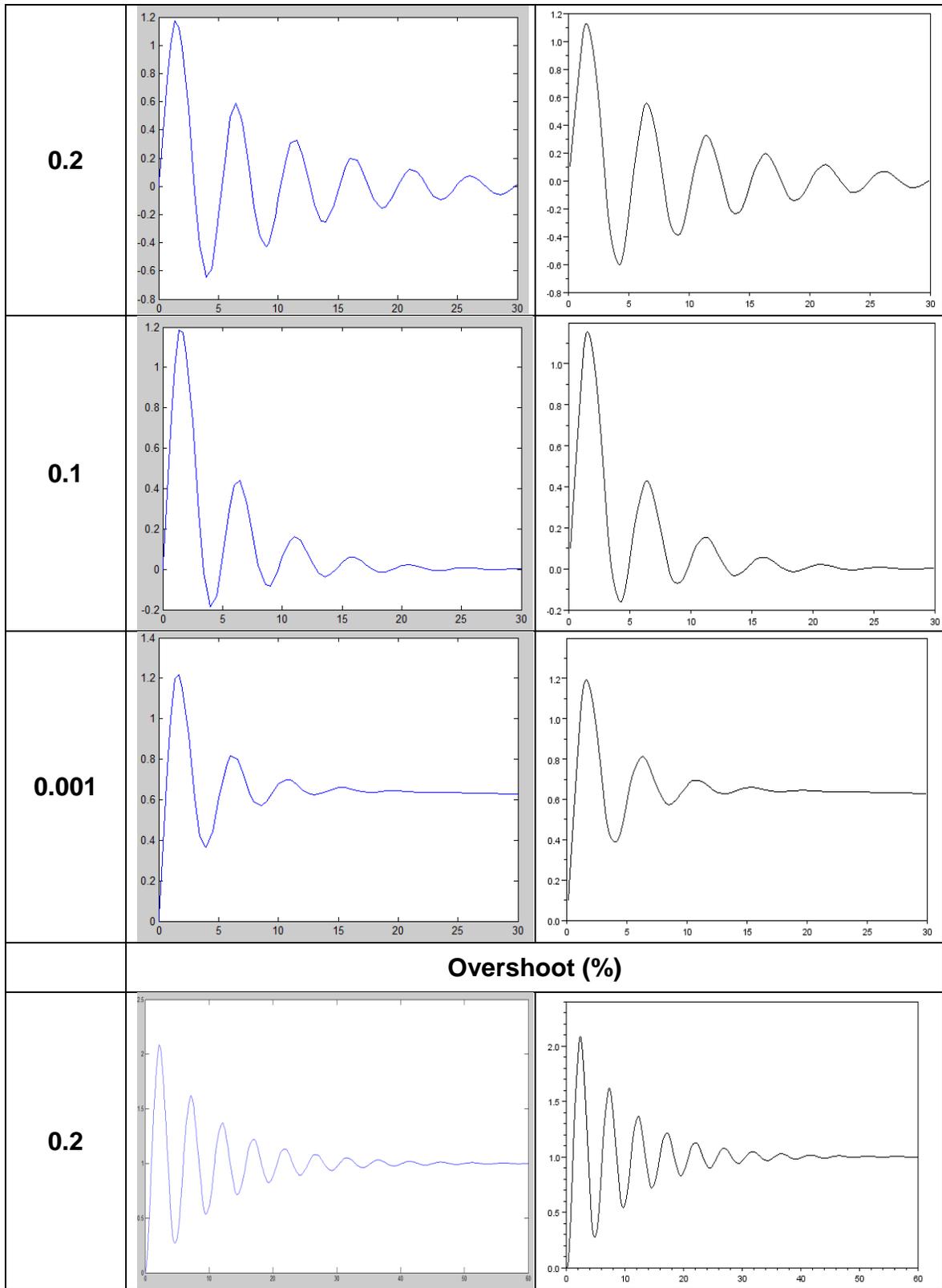


Tabla 10 Porcentaje de overshoot y Errores de posición y velocidad del sistema con P

$K_p=1$

	Simulink	Xcos
Tipo de Error	Posicion	
Ki		
0.2		
0.1		
0.001		
	Velocidad	



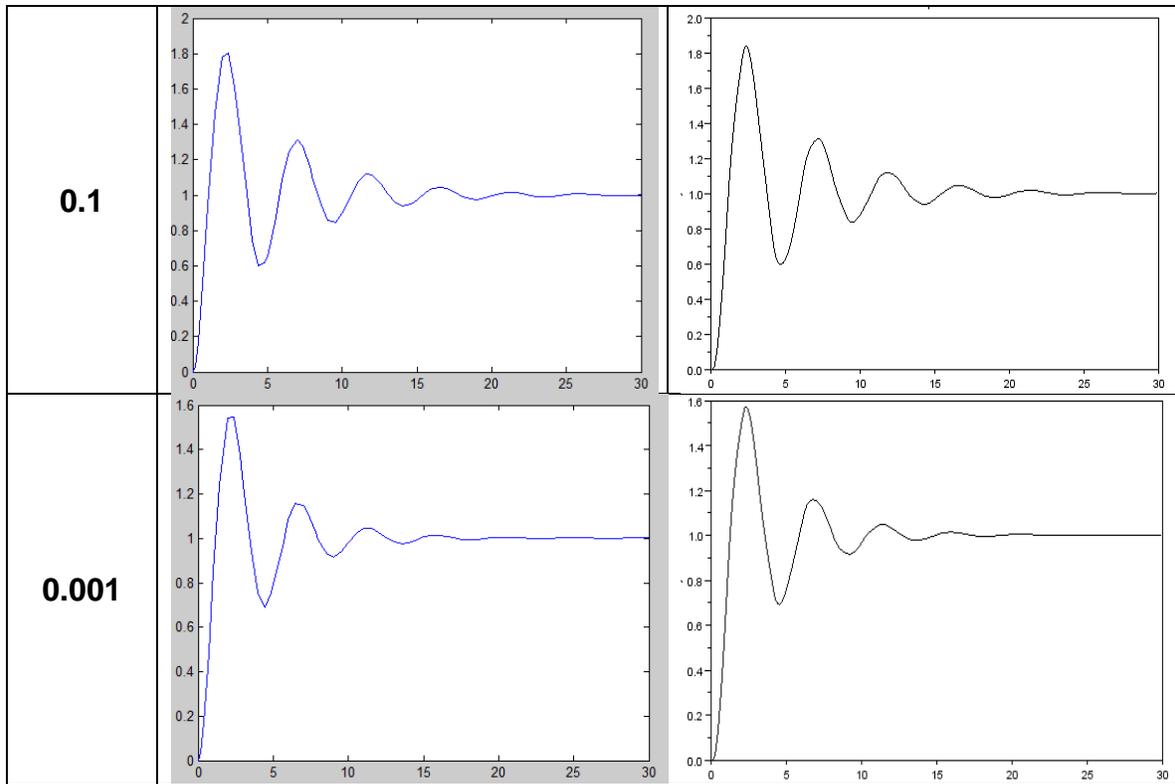
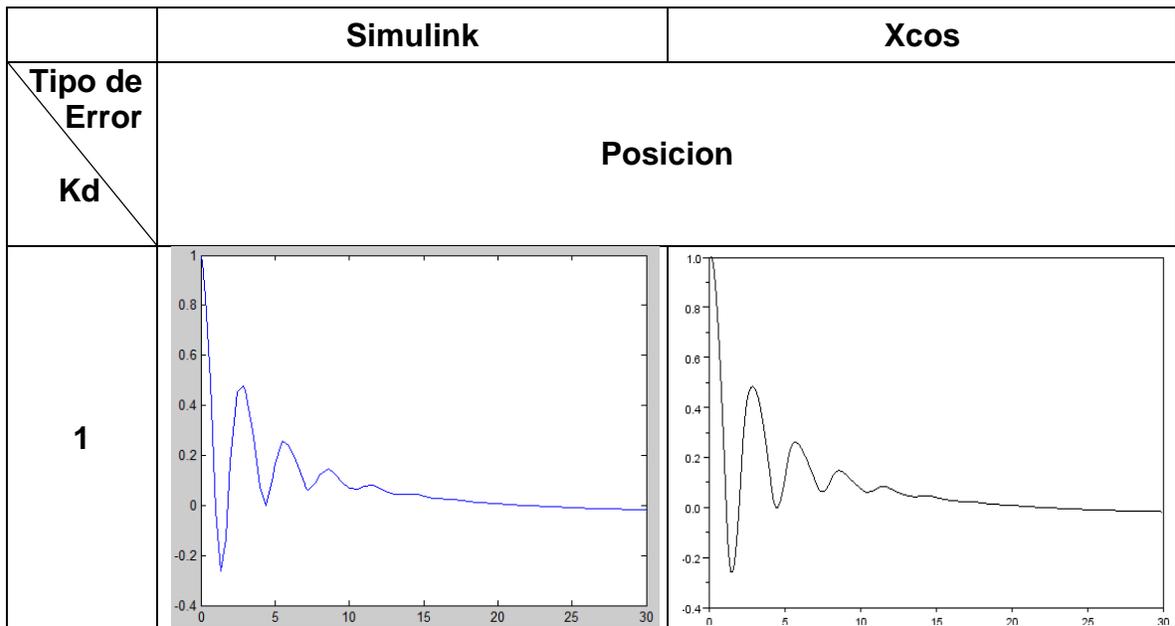
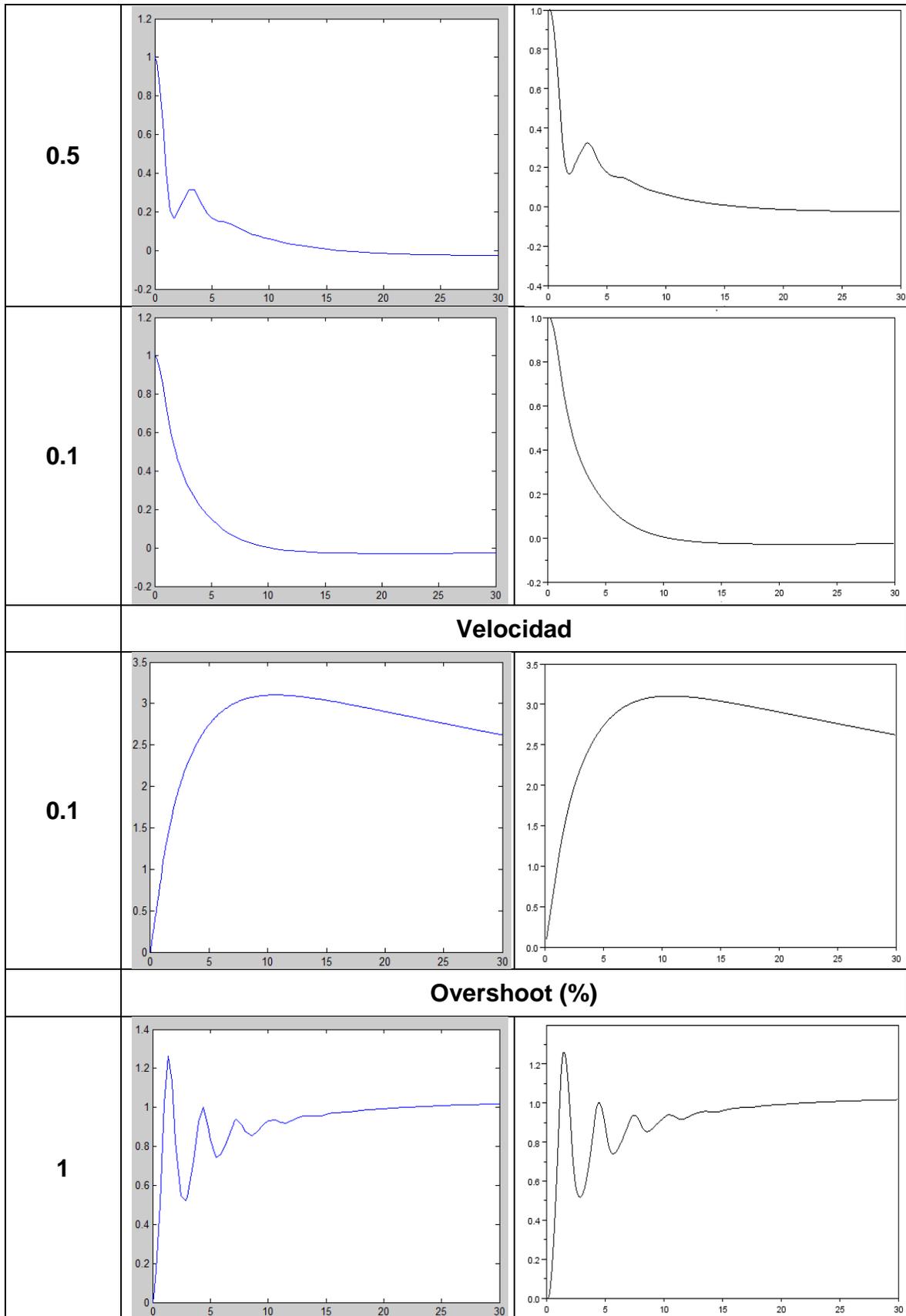


Tabla 11 Porcentaje de overshoot, Tiempo de asentamiento y Errores de posición y de velocidad del sistema con PI

$K_p=0.2$, $K_i=0.001$





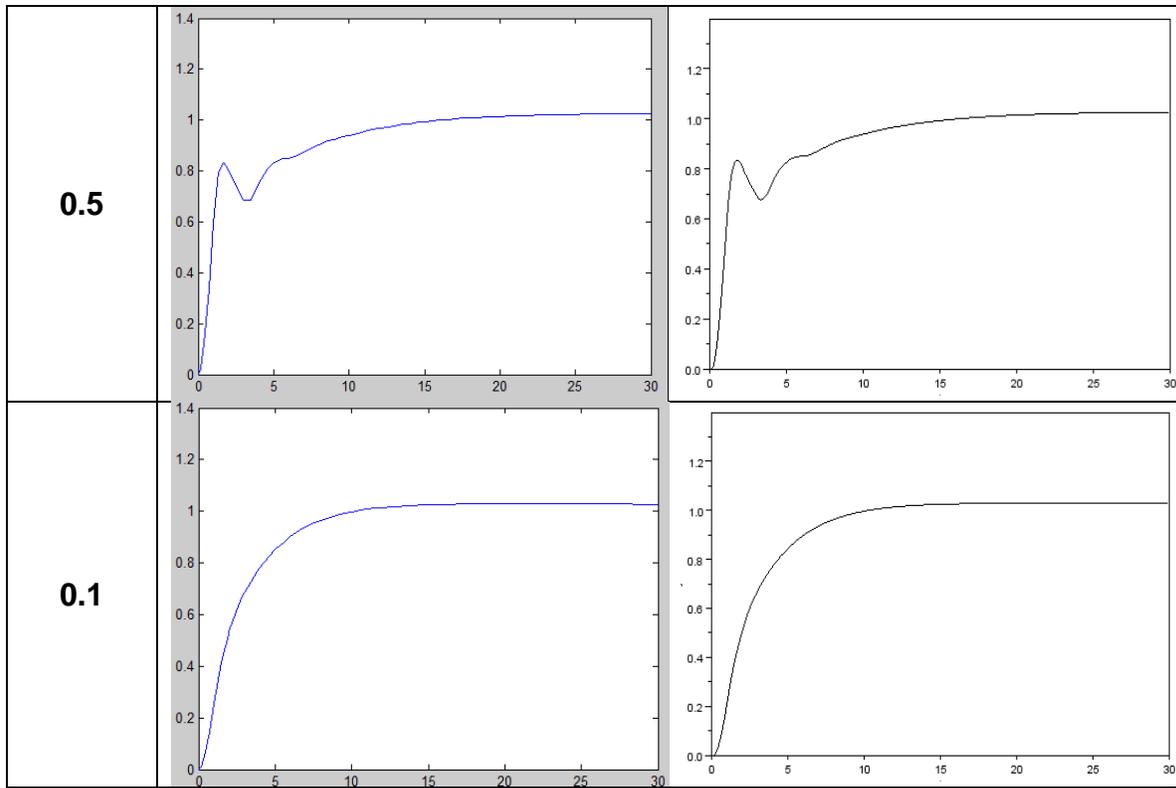


Tabla 12 Porcentaje de overshoot, Tiempo de asentamiento y Errores de posición y de velocidad del sistema

CONCLUSIONES

Se desarrolló las guías del laboratorio de control mostrando el manejo y similitud entre los software de simulación Xcos y Simulink, comprobando así la utilidad y confiabilidad para realizar trabajos de controles digitales o análogos y diferentes tipos de simulaciones en Xcos.

Se comprobó la equivalencia de las simulaciones generadas en las prácticas de laboratorio de control, comparando los resultados gráficos obtenidos entre los software Simulink y Xcos.

Se elaboró una guía de introducción a Scilab y Xcos mencionados en los capítulos 4 y 5, mostrando su entorno y su espacio de trabajo para la simulación de diferentes tipos de procesos.

Se analizó el software Xcos, generando algunas ventajas como la simulación de elementos eléctricos, sistemas termo hidráulicos, tiempos de muestreos independientes para los bloques que los necesite como los de sistemas digitales, graficas entre otros, configuración de los bloques de sistema en forma algebraica a diferencia de Simulink en donde se configura por medio de solo los valores reales de la ecuación, también se encuentran algunas desventajas vistas en este documento, como son la configuración manual de los parámetros de simulación, como ejemplo para las gráficas en el cscope, en donde los limites verticales como tiempo de simulación son ingresados de forma manual.

En las gráficas de simulación de Simulink se generó un código para graficar de nuevo los resultados en donde apreciara mejor las curvas obtenidas, a diferencia de las gráficas adquiridas por medio de Xcos, las cuales son generadas como se muestran en este documento.

Para trabajos futuros se propone,

Se proyecta el estudio de compatibilidad de comunicación del software Xcos por medio de tarjetas como los de la national instruments o tarjetas diseñadas especialmente para este software para el control de planta físicas, en estudio como elaboración de proyectos de grados o trabajos de laboratorio.

Bibliografía

1. Bautista Quiroga, Edison, guía introductoria al usuario del simulador Xcos de Scilab aplicado a la simulación de sistemas de control, Universidad Pontificia Bolivariana, 2012
2. Calvo Rolle, José Luis, Scilab. Programación y simulación, editorial RA-MA EDITORIAL,2009.
3. Scilab (a Free Software to Matlab), S Chand & Co Ltd, 2011.
4. Hybrid dynamic systems modeler and simulator
<http://www.scilab.org/products/xcos>