Parte IV
Observaciones y conclusiones

Observaciones y conclusiones

Los resultados obtenidos en los experimentos desarrollados con el RCP concuerdan con los obtenidos en las simulaciones, lo cual demuestra la validez del modelo computacional desarrollado.

La utilización de una herramienta como Matlab (con su librería RTW) posibilita la realización de simulaciones en tiempo real. Esto permite ir más allá de la simulación tradicional y producir un acercamiento al sistema real. Además la utilización de Simulink se logra un camino rápido y directo desde el diseño del sistema hasta su realización, reduciendo de este modo el tiempo y los costos en el diseño de sistemas de control.

El hecho de poder desarrollar controladores sin necesidad de implementarlos físicamente y de poder probar su funcionamiento de manera inmediata sobre el sistema real hace que la utilización del RCP aporte enormes ventajas.

La posibilidad de incluir en nuestro sistema, no solamente los controladores sino otros componentes de gran importancia como filtros, amplificadores, multiplexores y unidades que pueden realizar operaciones matemáticas complejas sin necesidad de implementarlos físicamente hace de un RCP una herramienta muy poderosa.

La facilidad de visualizar el comportamiento del sistema mediante una interfaz muy fácil de diseñar utilizando Control Desk nos brinda una información completa del mismo, lo cual facilita el análisis y las conclusiones.

La enorme ventaja de poder hacer cambios en tiempo real mediante la utilización de instrumentos en la interfaz del usuario de Control Desk, facilita la verificación del correcto funcionamiento del sistema.

En caso de que existan problemas de funcionamiento, los instrumentos de Control Desk sirven para la identificación de posibles fallas, lo cual reduce sustancialmente el tiempo de desarrollo.

Se ha demostrado la potencialidad que brinda la tecnología DSP como en el caso del sistema de desarrollo dSPACE y el alto rendimiento del lenguaje Matlab.

Al terminarse con éxito la elaboración del RCP, se plantea la necesidad de cambiar el enfoque de los cursos de control, máquinas eléctricas, electrónica de potencia, instrumentación.

6 Bibliografía

[1] Schütte H., *"TDE, An Integrated Toolset for Real-Time Control Applications"*, DSPACE GmbH, Paderborn, Germany, 1998.

[2] W. Robbins, N. Mohan, C. Henze and T. Undeland, *"A building block based power electronics instructional laboratory"* in Proc.IEEE Power Electron. Spec. Conf. ,vol2, Cairns, Australia, 2002, pp.467-472.

[3] Mathworks Co., *"Real Time Workshop for use with Simulink, User's guide"*, 2000

[4] Otterbach R. y otros, *"Real-Time Simulation: Requirements and the State of Technology"*, DSPACE GmbH, 1999.

[5] Bruno M. (2003) *"Control contínuo de velocidad por flujo orientadote un motor de inducción utilizando metodología R.C.P."*
Facultad de ingeniería , universidad de Buenos Aires, Argentina.

[6] Rossi P. D., *"Control discreto de velocidad por flujo orientado, de un motor de inducción, utilizando metodología RCP (desarrollo Rápido de un Control Prototipo)"*, Facultad de ingeniería, Universidad de Buenos Aires, Argentina, 2003

[7] The Math Works inc. *"The Student Edition of Matlab"* Prentice Hall

[8] Hanselman Duane, Littlefield Bruce *"Mastering Matlab"* Prentice Hall

[9] The Math Works *"Simulink, Dynamic System Simulation Software"* The Math Works inc.

[10] Proakis John, Manolakis Dimitris, "Tratamiento Digital de Señales" Prentice Hall

[11] Romero M., *"Introducción al Procesamiento Digital de Señales"*

[12] Department of Electrical and Computer Engineering, University of Minnesota *"DSP Based Electric Drives Laboratory"*

[13] Mohan Ned, *"Power Electronics"* John Wiley & Sons Inc.

[14] Mohan Ned, *"Electric Drives, an Integrative Aproach"* Mnpere

[15] N.Mohan, G Shanmugavel, P. Holenarsipur, and G. Nirgude, *"First course on power electronics simplified by unifying analysis of PWM converters as cycle-by-cycle averaged two-ports, and by PSpice-based simulations"* in proc.8th Eur. Conf. Power Electron. Applicat. ,Lausanne, Switzerland, 1999, p.8.

[16] Electric Drives an integrative approach. – Published by MNPERE Minneapolis.
Ned Mohan.

[17] DSP Based Electric Drives Laboratory - User Manual – Department of Electrical and Computer Engineering – University of Minnesota

Anexo 1

GUIAS DE LABORATORIO PARA CONTROLAR MOTORES DE INDUCCION Y MOTORES DE DC UTILIZANDO UN RCP (RAPID CONTROL PROTOTYPING)

Familiarización con el inversor trifásico[17]

1. Introducción

La tarjeta con los inversores trifásicos para ser usada en el laboratorio de máquinas eléctricas ha sido diseñada para permitirnos la realización de una variedad de experimentos en máquinas AC/DC. Las principales características de la tarjeta son:

- Dos inversores trifásicos PWM para un control completo y simultáneo de dos máquinas.
- Alimentación DC de bajo voltaje para reducir el peligro de choques eléctricos.
- Canales digitales de entrada PWM para control digital en tiempo real.
- Interfaz completa digital/análo ga con la tarjeta dSPACE.

2. Familiarización con la tarjeta.

El diagrama de bloques básico se muestra en la figura 1 y una fotografía de la tarjeta se muestra en la figura 2. Se anexa la identificación de varios componentes de la tarjeta en la tabla 1.

2.1 Inversores

Cada inversor trifásico utiliza MOSFETS como elementos de conmutación. Las salidas trifásicas del primer inversor están marcadas A1, B1, C1 y los del segundo inversor están marcados A2, B2, C2.

2.2 Voltajes de Alimentación

La alimentación de ± 12V se requiere para el manejo de las señales análogas. Esto se obtiene de una fuente aislada, la cual se acopla al conector J90. El interruptor S90 controla la potencia de la tarjeta. El led verde D70 indica si la tarjeta se encuentra debidamente aliment ada. Los fusibles F90 y F95 sirven de protección para las fuentes de +12V y -12V respectivamente. Nótese que apagando el interruptor S90 las señales PWM que van a las compuertas de los inversores seguirán llegando.La fuente de alimentación para los puentes trifásicos de los inversores se toma del bus de DC a través de un conversor Flyback.

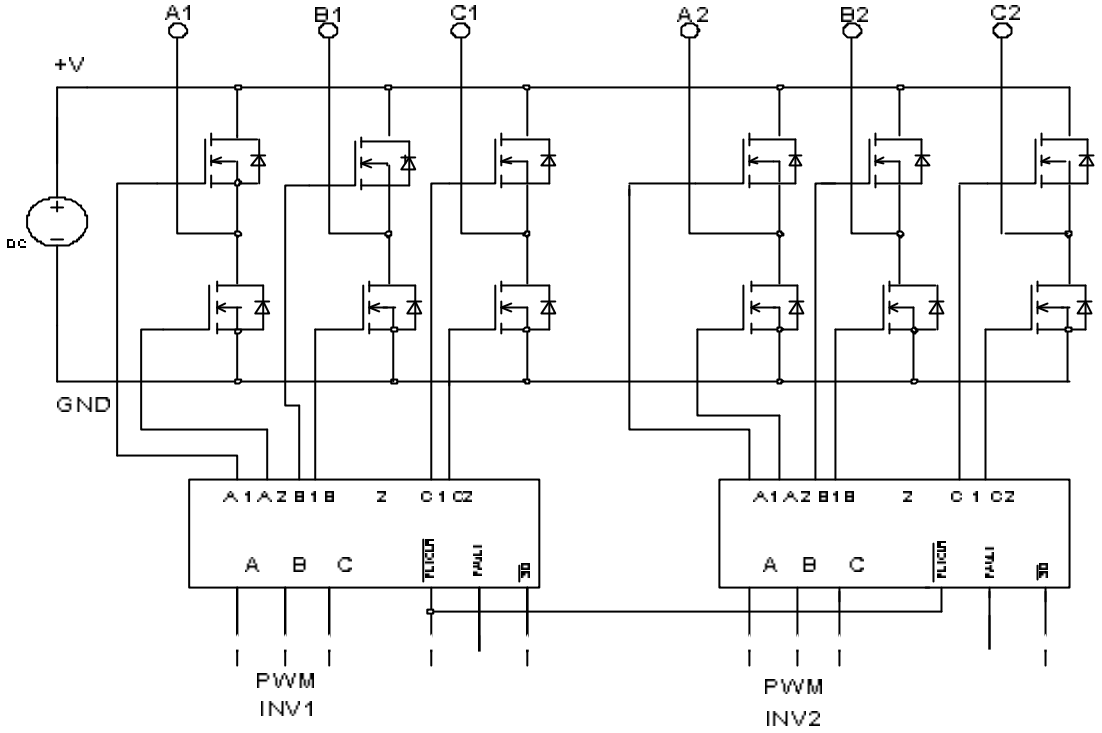
2.3 Medición de voltaje

Se han previsto puntos de prueba para observar los voltajes de salida de los inversores. El conector BNC VOLT DC se ha previsto para sensar el voltaje de DC del Bus de voltaje. Para medir el voltaje de DC del Bus de voltaje,

- Conecte un cable BNC al conector VOLT DC BNC.
- El factor de escalamiento del voltaje de entrada es 1/10

N°.	Componente	Código	Localización en la figura 2
1	Terminal + 42V	J1	A – 4
2	Terminal GND	J2	A – 3
3	Terminal fase A1	J3	D – 6
4	Terminal fase B1	J4	E – 6
5	Terminal fase C1	J5	G – 6
6	Terminal fase A2	J6	J – 6
7	Terminal fase B2	J7	K – 6
8	Terminal fase C2	J8	L – 6
9	Conector DIN para fuente ± 12V	J90	B – 2
10	Interruptor de alimentación	S90	C – 2
11	Fusible + 12V	F90	C – 2
12	Fusible -12V	F95	B – 2
13	Led indicador de alimentación	D70	C – 2
14	Led falla motor 1	D66	D – 2
15	Led falla motor 2	D67	L – 2
16	Led potencia digital	D68	I – 2
17	Interruptor principal	D69	B – 3
18	Inversor 1		D – 3 a G - 4
19	Inversor 2		I – 3 a L – 4
20	Condensador de acople DC inversor 1	C1	B – 5
21	Condensador de acople DC inversor 2	C2	G – 5
22	Driver IR2133 para inversor 1	U1	E – 2
23	Driver IR2133 para inversor 2	U3	J – 2
24	Fusible fuente digital	F2	G – 1
25	Conector de entrada dSPACE	P1	H – 1 e I – 1
26	Interruptor de reset	S1	L – 1
27	Sensor de corriente fase A1 (LEM)	CS2	C – 5
28	Sensor de corriente fase B1 (LEM)	CS3	D – 5
29	Sensor de corriente fase A2 (LEM)	CS5	H – 5
30	Sensor de corriente fase B2 (LEM)	CS6	J – 5
31	Sensor de corriente de acople de DC (LEM)	CS1	L – 5
32	Voltaje de DC	BNC5	B – 4
33	Corriente A1	BNC1	B – 3
34	Corriente B1	BNC2	C – 3
35	Corriente A2	BNC3	H – 3
36	Corriente B2	BNC4	I - 3

Tabla 1: Componentes de la tarjeta de los inversores trifásicos


A1
B1
C1
A2
B2
C2
+V
DC
GND
A1 A2 B1 B2 Z C1 C2
A B C
PWM
INV1
PWM
INV2

2.5 Circuito de los excitadores de los inversores

Los inversores están manejados por los excitadores de puentes trifásicos IR2133. Las entradas PWM se aíslan antes de ser alimentadas a los excitadores.

2.5 PWM / Señales digitales

El PWM y otras señales digitales para la tarjeta se suministran al conector de 37 pines DSUB. Para información sobre el conector ver la tabla 2.

2.6 Protección contra fallas

La tarjeta tiene protecciones de sobre-corriente para cada inversor. Una falla de sobre-corriente que ocurra en el inversor número 1 se indica por el led rojo “MOTOR FAULT 1”, mientras que una falla de sobre-corriente en el inversor número 2 se indica por el led rojo “MOTOR FAULT 2”. Cada vez que ocurra una falla se debe activar el interruptor de reset. Todas las fallas son reseteadas mediante este interruptor.

Nº.	Identificación	Descripción
1	GND Digital	Tierra digital
2	FAULT 1	Falla en la salida del inversor 1
3	NC	Sin conexión
4	GND-Digital	Tierra digital
5	NC	Sin conexión
6	GND Digital	Tierra digital
7	PWM A1	Señal de la fase A del inversor 1
8	PWM B1	Señal de la fase B del inversor 1
9	PWM C1	Señal de la fase C del inversor 1
10	PWM A2	Señal de la fase A del inversor 2
11	PWM C2	Señal de la fase C del inversor 2
12	GND Digital	Tierra digital
13	GND Digital	Tierra digital
14	GND Digital	Tierra digital
15	GND Digital	Tierra digital
16	$\overline{SD\ \ 1}$	Señal de apagado para el inversor 1
17 18	$\overline{FLTCLR\ \ -IN}$ VCC	Señal de borrado de falla
19	VCC	
20	GND Digital	Tierra digital
21	FAULT 2	Falla en la salida del inversor 2
22	NC	Sin conexión
23	NC	Sin conexión
24	NC	Sin conexión
25	GND Digital	Tierra digital
26	NC	Sin conexión
27	NC	Sin conexión
28	NC	Sin conexión
29	PWM B2	Señal de la fase B del inversor 2
30	GND Digital	Tierra digital
31	GND Digital	Tierra digital
32	GND Digital	Tierra digital
33	GND Digital	Tierra digital
34	NC	Sin conexión
35	$\overline{SD\ \ 2}$	Señal de apagado para el inversor 2
36	GND Digital	Tierra digital
37	GND Digital	Tierra digital

Tabla 2: Conector de 37 pines DSUB

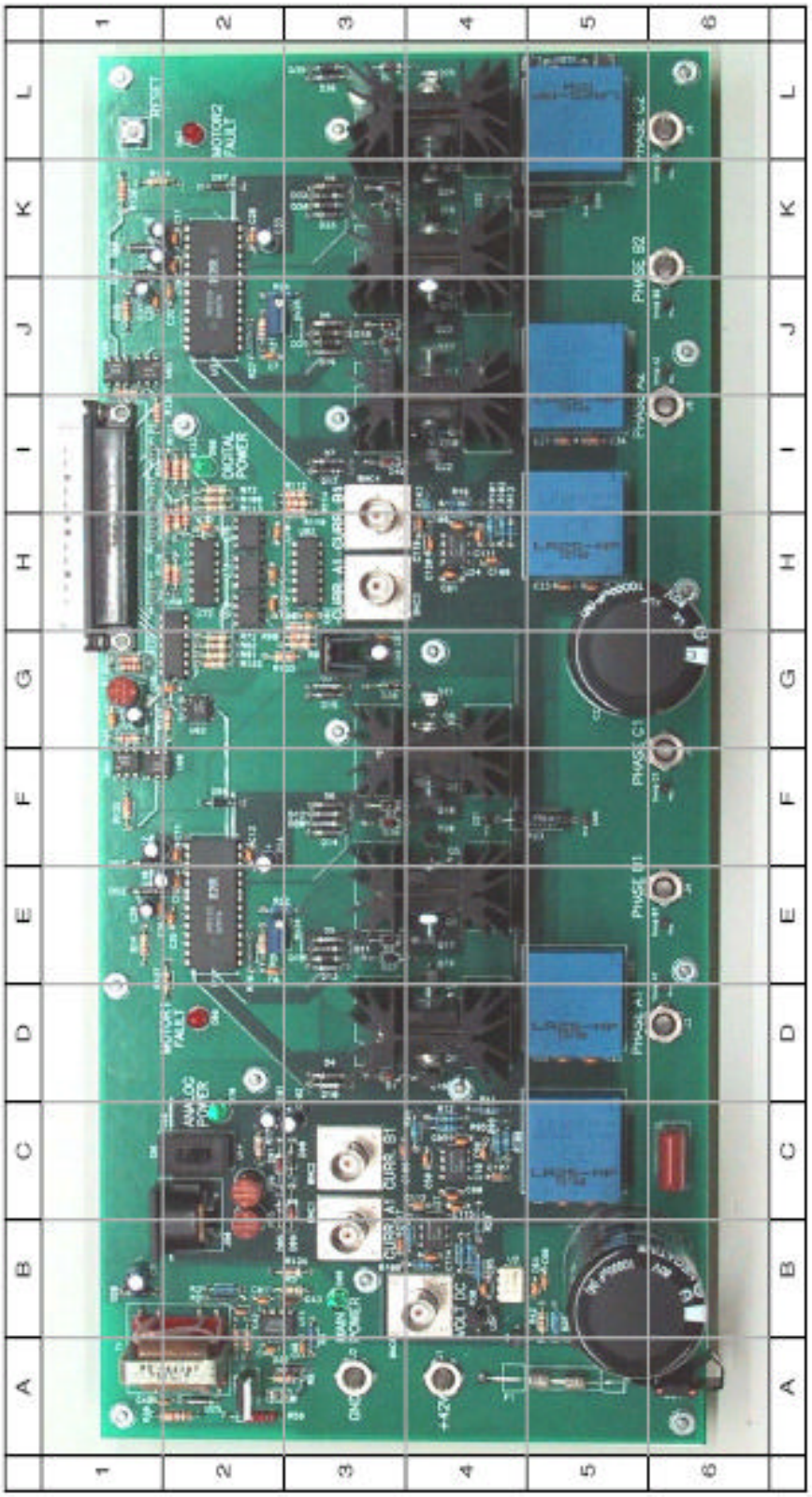


Figura 2: Tarjeta de los inversores trifásicos

Experimento1

Creación de un modelo simple en SIMULINK[17]

1.1 Introducción

El modelado matemático de las máquinas eléctricas y sus excitadores (drivers) involucra la solución de ecuaciones diferenciales. Herramientas matemáticas tales como Matlab ayudan en la solución de esas ecuaciones diferenciales de un modo fácil y rápido. Matlab también contiene una herramienta de modelado llamada SIMULINK, la cual ayuda a representar el problema de una manera gráfica utilizando bloques interconectados y con capacidad para visualizar la solución de las ecuaciones diferenciales usando gráficas y representaciones. De hecho, muchos sistemas en tiempo real tales como sistemas de desarrollo DSP vienen con una interfaz con SIMULINK por medio de la cual ellas pueden convertir el conjunto de bloques de SIMULINK en un código de máquina, el cual puede correr en un sistema de desarrollo basado en DSP. (Para la realización de nuestras prácticas, utilizaremos como sistema de desarrollo una tarjeta DS1104 de dSPACE).
En este experimento, haremos una primera aproximación a SIMULINK, de tal modo que seamos capaces de:

- Construir un modelo simple usando bloques de SIMULINK
- Simular el modelo para ver el comportamiento del sistema en diferentes condiciones.

1.2 Creando un modelo en SIMULINK

Desarrollaremos un sistema simple continuo en el tiempo utilizando SIMULINK para entender la interacción mecánica de los sistemas rotatorios consistente en motor y carga. Realice los pasos siguientes:

- Cree una carpeta exp1
- Arranque Matlab 7.1 o superior
- En la parte superior de la pantalla usted verá una caja de dialogo en la cual aparece el directorio actual cámbielo por la carpeta que usted acabó de crear (exp1)
- Abra el SIMULINK se abrirá una nueva ventana que contiene varias librerías de SIMULINK Y dSPACE.

Estamos tratando de desarrollar un modelo para un sistema rotacional simple, para el cual rigen las siguientes ecuaciones.

$$T_J = T_M - T_L \tag{1.1}$$

$$J_{eq} = J_M + J_L \tag{1.2}$$

$$a = \frac{T_J}{J_e q} \tag{1.3}$$

$$w_m = \int a.dt \tag{1.4}$$

$$q = \int w_m.dt \tag{1.5}$$

Para nuestro ejemplo utilizaremos $J_{eq} = 0.058\, Kg.m$

- Para crear este modelo en SIMULINK primero cree un nuevo modelo del menú *File*.
- Para agregar bloques a su modelo en SIMULINK simplemente haga clic en el bloque que usted desea en la ventana de la librería y arrástrelo hasta su modelo. Para nuestro modelo necesitaremos los siguientes bloques:
  - ✓ Un bloque de suma de la librería *Math*
  - ✓ Un bloque de ganancia de la librería *Math*
  - ✓ Dos ganancias deslizables de la librería *Math*
  - ✓ Dos integradores de la librería *Continuous*
  - ✓ Dos constantes de la librería *Source*
  - ✓ Un *Mux* de la librería *Signals and Systems*
  - ✓ Un osciloscopio de la librería *Sinks*
- En muchos casos los bloques de SIMULINK tienen propiedades que pueden ser modificadas haciendo doble clic sobre ellos. Haga las siguientes modificaciones:
  - ✓ Cambie el bloque de suma de tal manera que el tenga ?+- en lugar de ?++
  - ✓ Cambie los límites de la ganancia deslizable a -10 y 10. Ajuste los valores para un bloque a 5 y para el otro a 0.
  - ✓ Cambie el *mux* a 3 entradas
  - ✓ Cambie el valor de la ganancia a $\frac{1}{J_{eq}} = \frac{1}{0.058}$
- Para conectar los bloques haga clic en una flecha de entrada o salida y arrástrela hasta la otra entrada o salida a la cual usted quiere ir. Haga las conexiones tal como están establecidas en las ecuaciones para el sistema mecánico rotatorio.
- Para renombrar bloques, haga clic sobre los nombres actuales y cámbielos. Para darle nombre a las conexiones haga clic en ellas y aparecerá un cuadro de texto.
- Después de que se termine las conexiones el modelo debe aparecer tal como en la figura 1.1

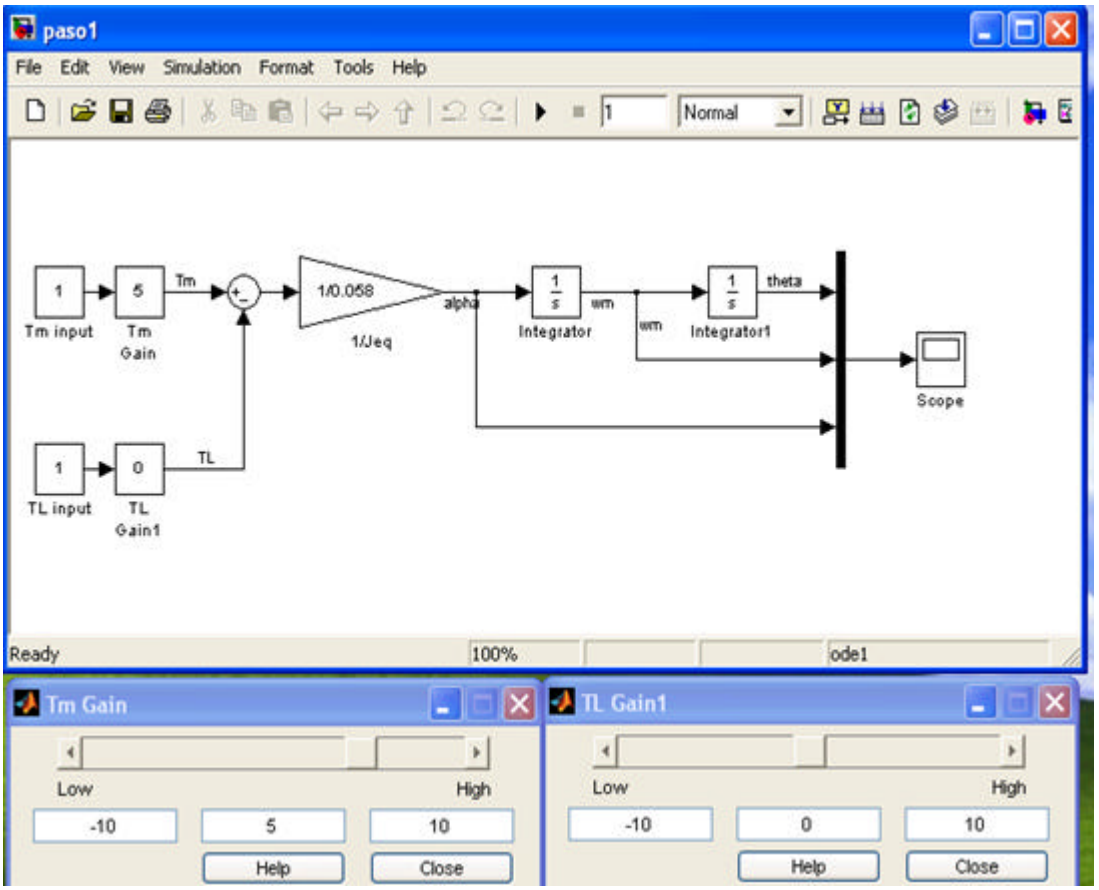


Figura 1.1: Modelo de un Sistema Mecánico Simple

- El siguiente paso es correr la simulación. Antes de correr la simulación ajuste los parámetros. Para hacer esto vaya al menú de simulación y seleccione *simulation parameters.* Ajuste los parámetros como se muestra en la figura 1.2

- Una vez que se termina el ajuste de parámetros, el modelo está listo para la simulación.

- Haga clic en el botón triangular para la simulación.

- Una vez que la simulación está terminada, haga doble clic en la caja del osciloscopio. El resultado debe ser como el de la figura 1.3.

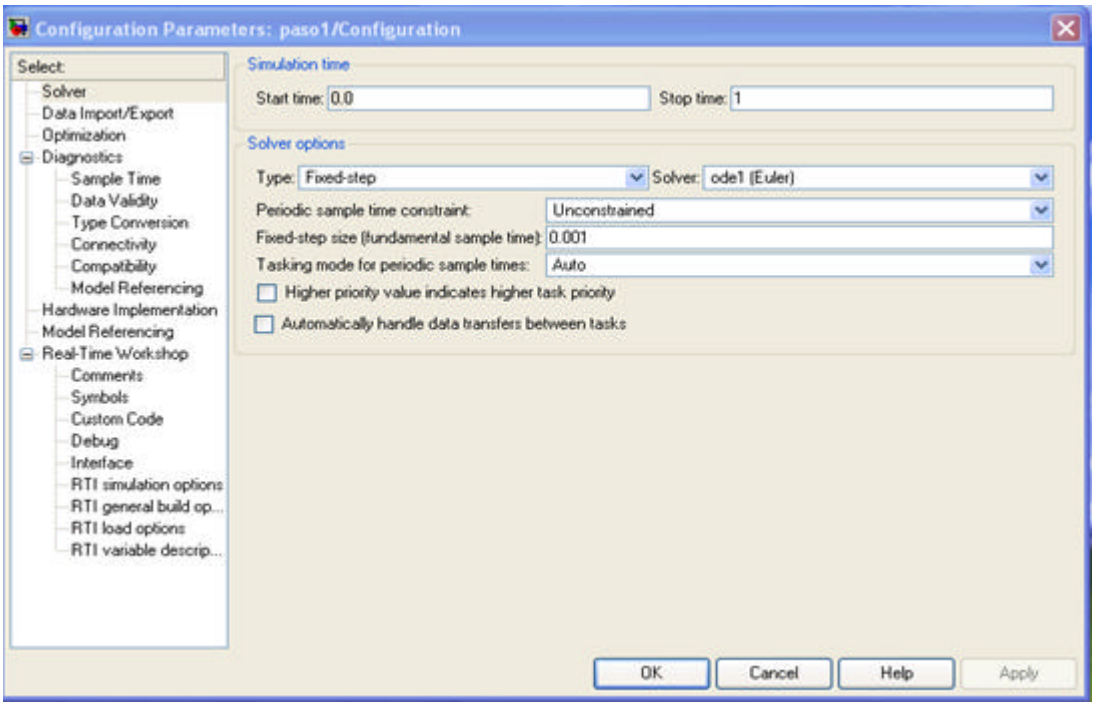


Figura 1.2 Ajuste de los parámetros de simulación

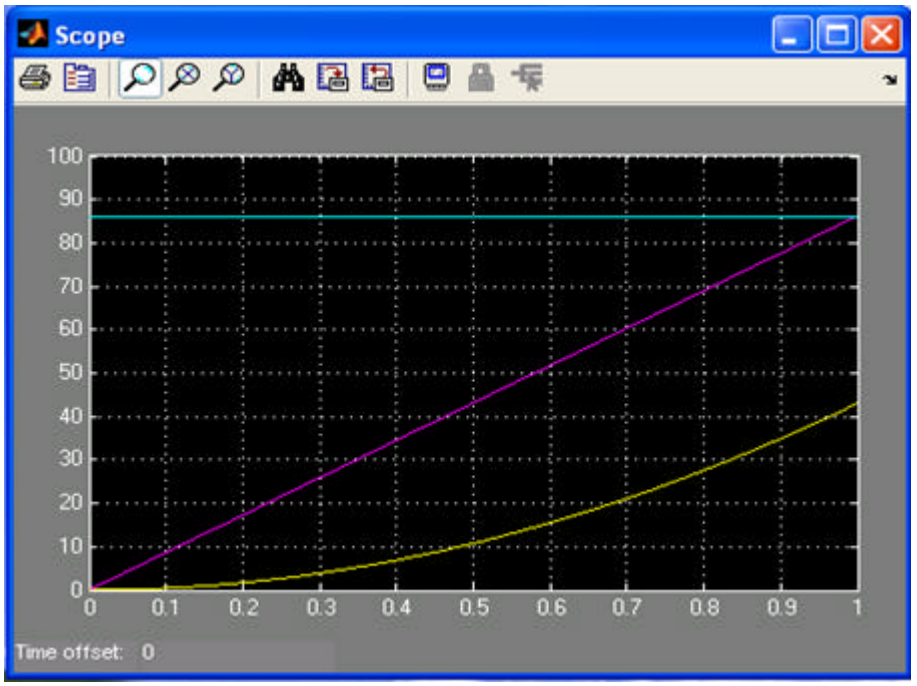


Figura 1.3: Resultados de la simulación paso1

- Ahora cambiando cualquiera de las ganancias deslizables y corriendo la simulación usted obtendrá diferentes resultados en su osciloscopio.

1.3 Cambiando el modelo de simulación

Ahora que hemos creado un modelo de un sistema mecánico rotatorio simple, agreguémosle fricción

$$T_{fric} = B.w \qquad (1.6)$$

Para nuestro ejemplo utilizaremos B = 0,025

- Primero salve el ejercicio anterior con un nombre cualquiera, por ejemplo paso1. Ahora haga otra copia de él y nómbrelo paso2.
- Aparte de hacer el cambio correspondiente a la ecuación (1.6) sería muy interesante si ? fuera una secuencia repetitiva. Para hacer esto añada un bloque de seno o coseno de la librería *Math.*
- También retire el bloque *mux* y añada otro puerto de entrada al osciloscopio haciendo doble clic sobre él. Encuentre el icono de los parámetros en la parte superior derecha y cambie el número de ejes a dos y el rango de tiempo a un segundo como en la figura 1.4

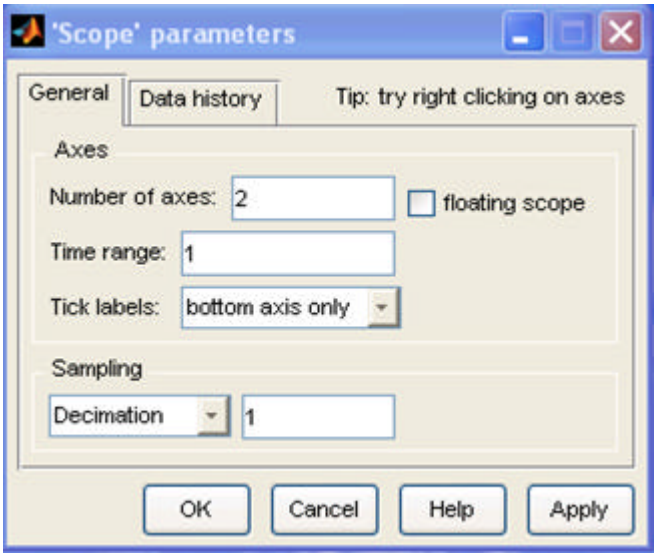


Figura 1.4: Ajuste de los parámetros del osciloscopio.

- Sería también muy importante si el sistema corriera continuamente. Para hacer esto vaya a la simulación de parámetros como se hizo en el paso1 y esté seguro de que todos los parámetros permanezcan iguales menos STOP TIME, que debe pasarse a *"inf"*.
- Cuando usted termine de cambiar el modelo deberá aparecer como el de la figura 1.5.

- Corra la simulación nuevamente y cambie T_M y T_L mientras la simulación está corriendo para ver como responde el sistema

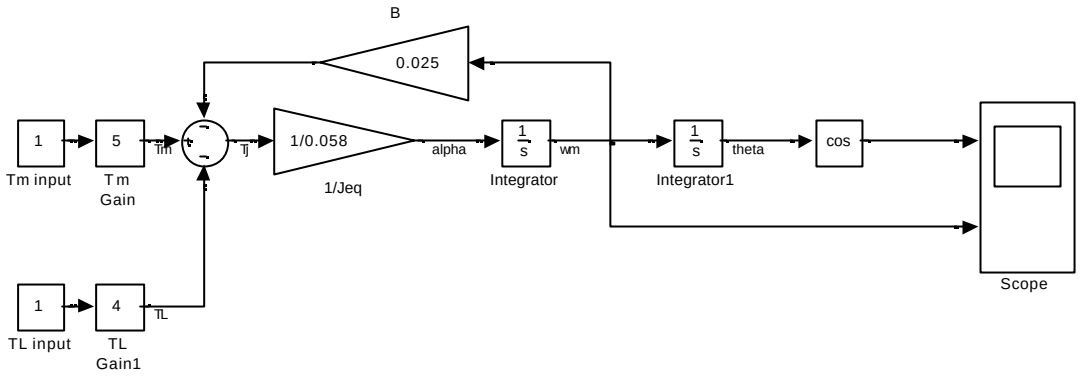


Figura 1.5: Modelo de simulación incluyendo fricción

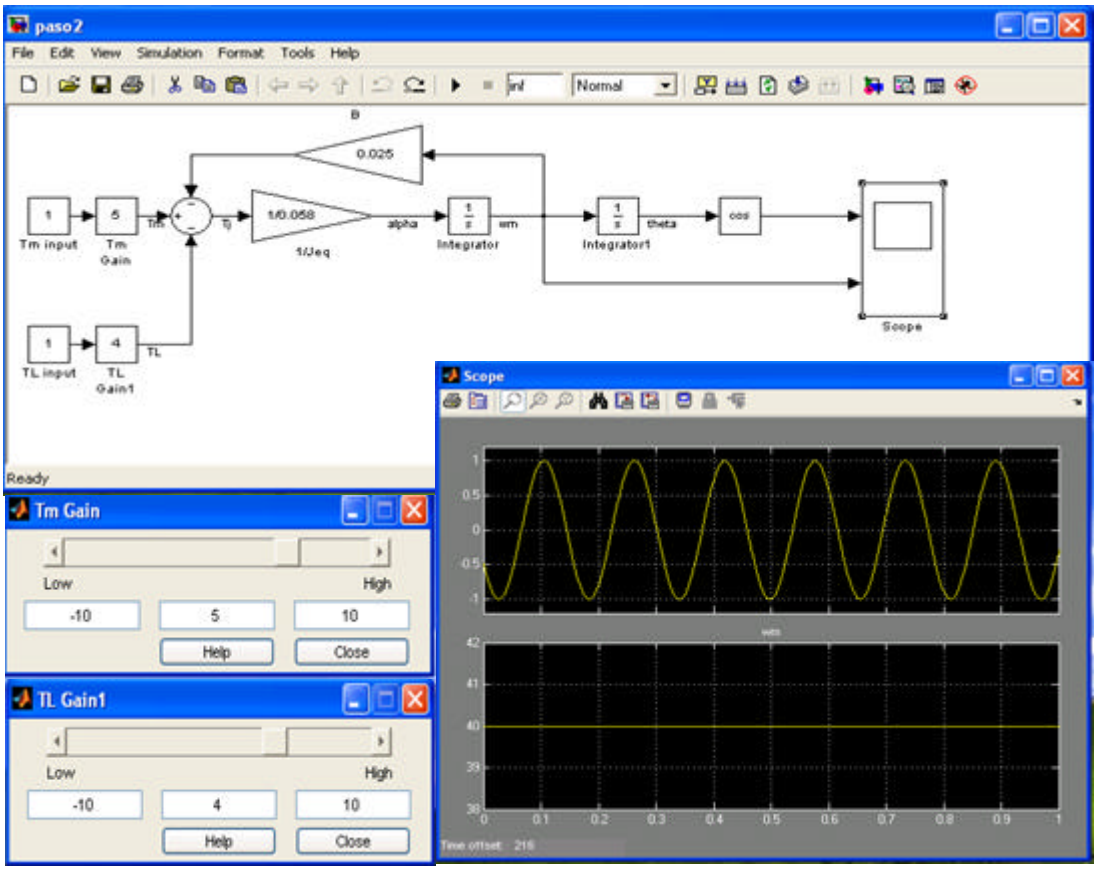


Figura 1.6: Resultados de la simulación del modelo incluyendo fricción

1.4 Reporte del laboratorio

El reporte debe incluir entre otros, los siguientes detalles:

- Resultados de la simulación del sistema mecánico obtenidos en la sección 1.2, con los comentarios de los resultados obtenidos.
- Resultados de la simulación del sistema mecánico con la fricción incluida en el modelo de acuerdo a la sección 1.3, con los comentarios de los resultados obtenidos. ¿Cuál es el efecto de la fricción en el comportamiento del sistema?
- Cambie los valores de T_M y T_L. ¿cuál es el efecto de cada uno? Explique.
- Cambie el torque de la carga T_L. ¿Cuál es el efecto de un cambio repentino? Observe los cambios generados con el cambio de carga y haga comentarios acerca de su influencia en velocidad, aceleración y q.

Experimento 2
Construcción de un modelo simple en tiempo real utilizando SIMULINK[17]

2.1 Introducción

Una de las mejores características de un RCP utilizando dSPACE es la facilidad con que se construyen aplicaciones en tiempo real. El tiempo transcurrido entre la conversión del diseño en instrucciones digitales para el DSP y la ejecución efectiva de la aplicación en tiempo real depende solamente de la rapidez del computador para compilar el código inicial.

Básicamente una aplicación en tiempo real puede llevarse a cabo mediante dos métodos:

- Utilizando Simulink de Matlab para construir el modelo y automáticamente generar el código en C y enviarlo al sistema de desarrollo DSP.
- Hacer "a mano" el código en C y compilar el modelo en código DSP.

El método más rápido de desarrollar código para una aplicación en tiempo real es, por supuesto, desarrollar el modelo en Simulink y preparar un modelo en tiempo real a partir de este. Básicamente una vez que se ha completado el modelo en Simulink, si usted quiere correrlo en tiempo real, el único comando requerido es **RTW Build** del menú **Tools** en Simulink. Una vez que se ejecuta el comando, el software de dSPACE crea un archivo objeto (*.obj), lo envía a la tarjeta DS1104 y automáticamente arranca la ejecución del hardware.

Sin embargo, hay algunos ajustes importantes que hacer antes de la ejecución en tiempo real. Empecemos con un ejemplo muy sencillo:

2.2 Creación de un modelo en Simulink

Nuestro primer ejemplo introduce los canales análogos de comunicación entrada – salida con el mundo exterior. Para este ejemplo utilizamos un generador de señales para generar diferentes formas de onda como entrada para nuestro algoritmo de procesamiento de señales. El resultado de este proceso se obtiene en un canal análogo de salida para ser monitoreado en un osciloscopio.

Supongamos que necesitamos analizar la respuesta de un sistema de segundo orden a diferentes tipos de señales con amplitudes variables. El sistema de segundo orden se define con los siguientes parámetros:

- El amortiguamiento (x) = 0.7, Frecuencia f = 10 Hz
- Queremos ver la respuesta del sistema a diferentes tipos de señales de entrada tales como una onda senoidal, una onda cuadrada, una diente de sierra.
- La entrada tiene una ganancia variable, en el rango de [0…5].

Primero desarrollaremos el modelo de simulación usando un modelo para el generador de señales y un modelo para el osciloscopio.

- Cree una carpeta exp03.
- Arranque Matlab y ajuste la ruta (*path*) a su carpeta de trabajo (exp03).
- Abra Simulink y cree un nuevo modelo en el menú *File*.
- En Simulink, seleccione el bloque **Transfer Fcn** de **Continuos library** y arrástrelo al nuevo modelo de simulación.

El sistema de segundo orden con los parámetros especificados puede ser descrito como una función de transferencia con la siguiente relación:

$$G(s)= \frac{w_n^2}{s^2 + 2x w_n s + w_n^2} \qquad (2.1)$$

Donde $w_n = 2pf = 2p.10 = 62.83$ y $x = 0.7$ El modelo numérico quedará:

$$G(s)= \frac{3947.84}{s^2 + 87.962\, s + 3947.84} \qquad (2.2)$$

- Ajuste los parámetros del bloque **Transfer Fcn** como se muestra en la figura 2.1
- Desde la librería **Sources**, arrastre un bloque de generador de señal, un **Slider Gain** de la librería **Math** y un **Scope** de la librería **Sinks**. Conecte todos los bloques como en la figura 2.2.
- Ajuste los parámetros de simulación presionando Ctrl.-E como en la figura 2.1:

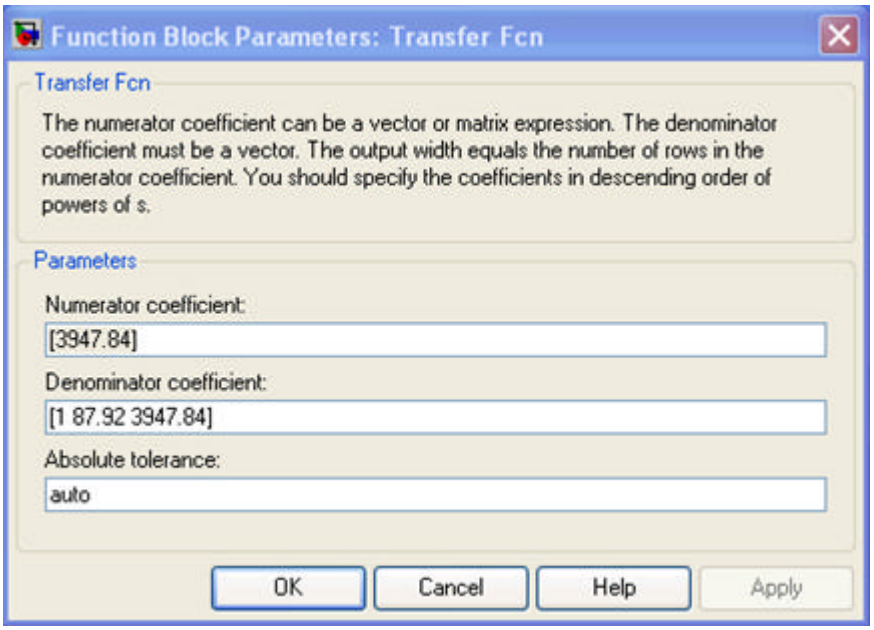


Figura 2.1: Parámetros para el sistema de segundo orden

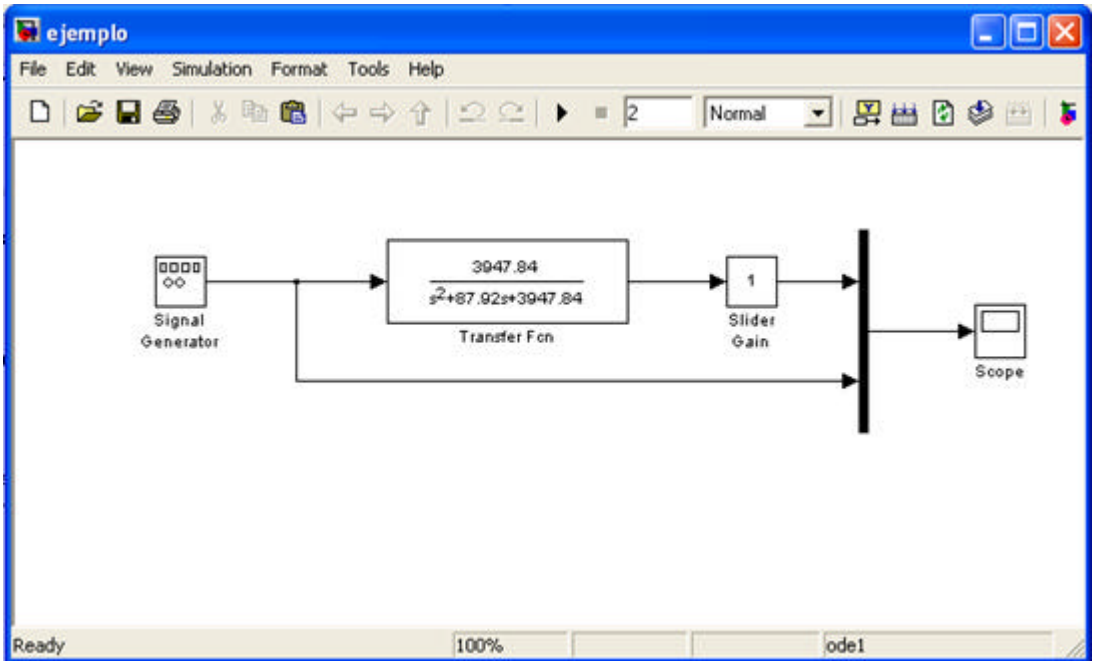


Figura 2.2: Modelo en Simulink para un sistema de segundo orden

-Ajuste las **solver options** en ***fixed step*** y ***ode1***
-Se necesita especificar un tamaño de escalón fijo, déjelo en "0.001", 1ms.
-Ajuste el "*Stop-time*" en 2 segundos.
-Ajuste la salida del generador de señales a *Square-wave* y frecuencia 2 Hz.

-Mantenga los otros parámetros

- Corra la simulación haciendo click en el ícono triangular y ajuste la pantalla del osciloscopio con la opción *autoscale*. Usted obtendrá una gráfica similar a la mostrada en la figura 2.3.

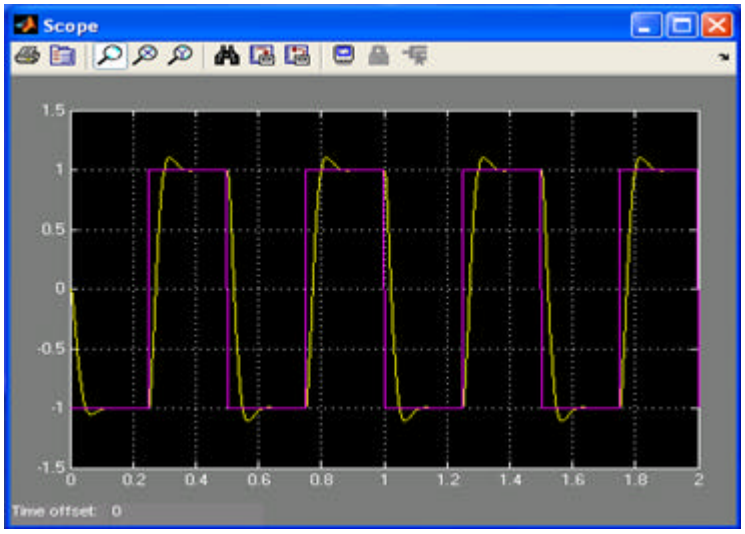


Figura 2.3: Resultados de la simulación de un sistema de 2º orden para entrada de una onda cuadrada.

La forma de onda del generador puede ser cambiada haciendo doble click en el bloque del generador de señales. Ahora, una vez que tenemos idea de cómo trabaja el sistema, implementémoslo en tiempo real y observemos los resultados.

2.3 Construcción del modelo de simulación en tiempo real

El modelo desarrollado para simulación está listo para ser conectado a equipos externos (Generador de señales y osciloscopio). Dado que estos equipos están físicamente generando o recibiendo señales que entran o salen de la tarjeta DSP, necesitamos acceder a esas señales vía los canales análogos de entrada/salida.
Primero que todo, asegúrese de que tanto el generador de señal como el osciloscopio estén conectados con cables apantallados y conectores BNC a ADC#5 y ADC#1 respectivamente. La salida del generador debe ser ajustada con una amplitud aproximada de un voltio.
La comunicación con los canales de entrada/salida se lleva a cabo mediante dos bloques dSPACE que se encuentran en la librería dSPACE RTI 1104, en la sublibrería DS1104 MASTER PPC, llamada **DS1104ADC_C5** y **DS1104DAC_C1.**
Ellas reemplazarán nuestros bloques **Generador de señales** y **Osciloscopio** respectivamente. Al canal análogo de entrada se le hace una escalización por medio del hardware de 1:10. Esto significa que 10V a la entrada se leerán como 1 V en nuestro modelo. El canal análogo de salida también sufre una escalización en el hardware con la misma relación (1:10), lo cual significa que una señal de 1V generada en el modelo tendrá una amplitud de 10V en el conector. De este modo se necesitarán dos bloques de ganancia

de la librería **Math** para corregir la lectura y escritura de los valores desde y hacia los canales análogos.

- Arrastre los bloques **DS1104ADC_C5** y **DS1104DAC_C1** al modelo de SIMULINK y reemplace los bloques de generador de señal y osciloscopio. Ubique las ganancias de 10 y 0.1 en la entrada y la salida respectivamente. El modelo debe ser similar al de la figura 2.4. Si usted quiere puede salvar el modelo con otro nombre para preservar el modelo de simulación.

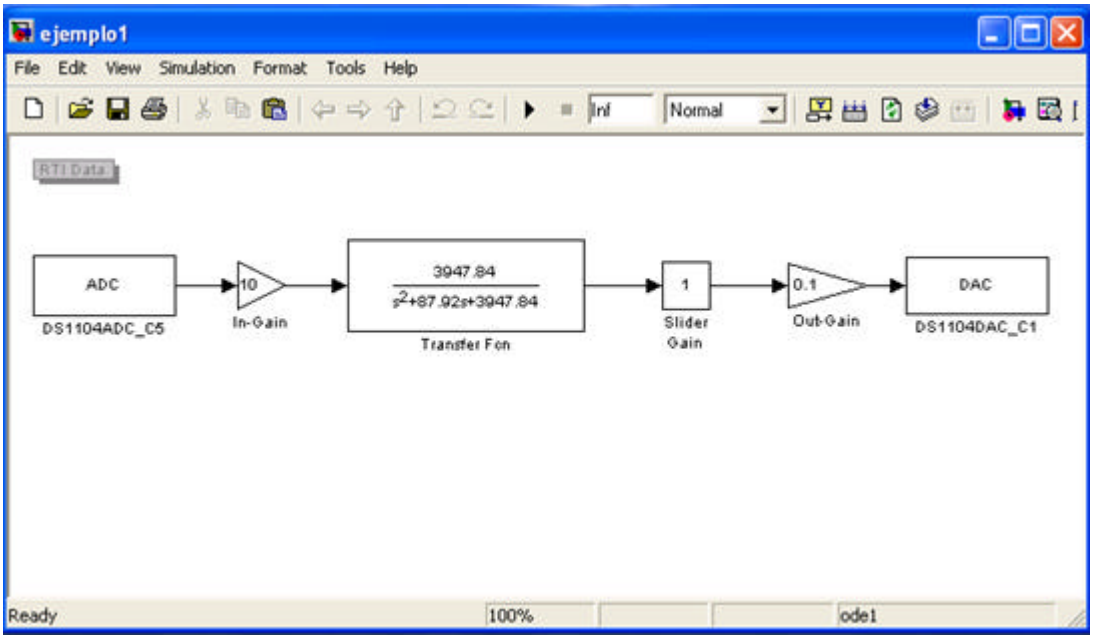


Figura 2.4 Modelo en tiempo real para dSPACE en Simulink

Recuerde que ejecutamos la simulación por solo 2 segundos, sin embargo, en tiempo real, el sistema necesita correr continuamente. Por lo tanto, presione **CTRL_E** para ajustar los parámetros de simulación y:

- Ajuste el **Stop-Time** en "**inf**". Los parámetros de simulación serán como los de la figura 2.5
- En parámetros de simulación señale **Advanced** y ponga **Block Reduction** en "**OFF**".
- Seleccione **Real-Time Workshop**, **Build Model** del menu de herramientas. (Tools)

Una vez que se han dado los anteriores comandos, usted observará una lista de mensajes en la ventana de comandos. Estos mensajes corresponden a los diferentes pasos que el software RTI leva a cabo para transformar el código de Simulink del archivo *ejemplo1.mdl* en código DSP.

Primero hay una etapa de compilación en la cual el archivo de Simulink se transforma en un archivo en C, después viene la etapa de enlace, en la cual todas las variables y subrutinas se correlacionan con el ambiente DSP y finalmente el código es transformado

en un archivo objeto y bajado a la memoria de la herramienta de desarrollo DSP. La ventana de de Matlab debe ser como la figura 2.6. Observe que el archivo se construyó exitosamente. El resultado es “ejemplo1.obj”, el cual ya fue cargado en la memoria DSP y arrancó su ejecución

Note que el directorio en el cual se realizó el modelo es el mismo que usted utilizó para crear el modelo de Simulink. Si usted observa este directorio, encontrará varios archivos generados mediante el comando BUILD.

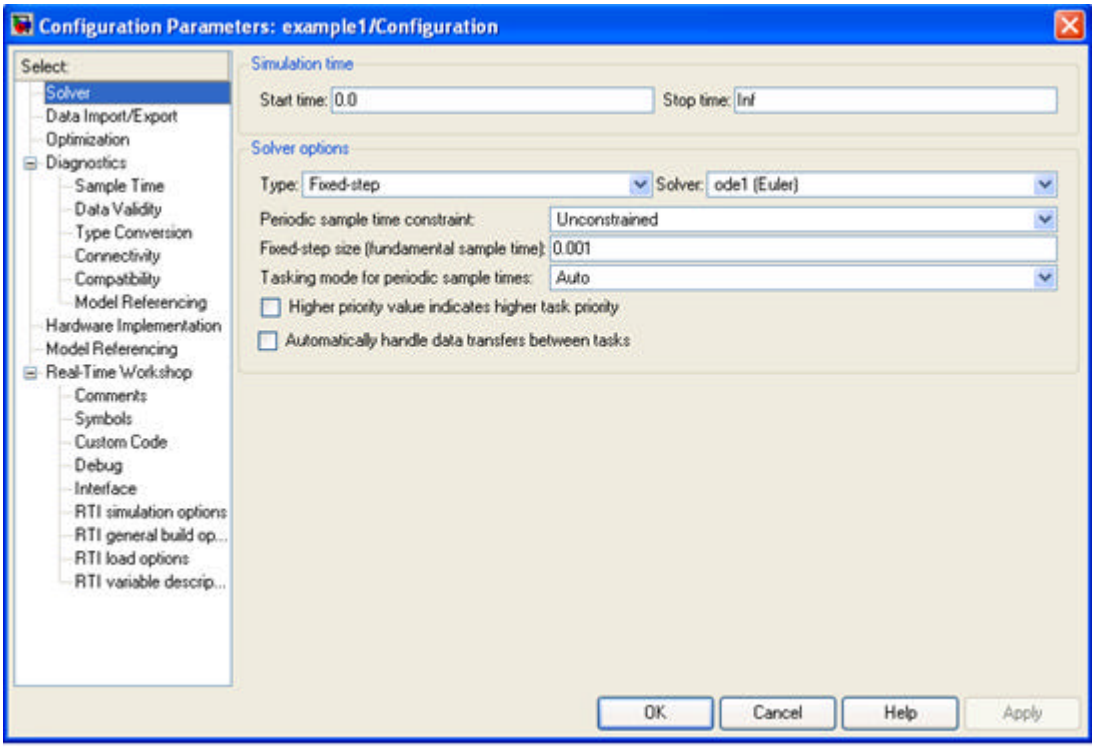


Figura 2.5 Ajuste de los parámetros de simulación para el modelo en Tiempo Real

Debido a la gran cantidad de archivos que se generan, se aconseja que cada proyecto se almacene en subdirectorios separados.

Ahora, la simulación está corriendo en la tarjeta DSP en un formato digital en tiempo real. De hecho, podemos anticipar que es mucho más rápida que aquella que vimos en la pantalla del osciloscopio en Simulink. Ahora sigue la parte más importante: la interacción con el sistema. Necesitamos visualizar, modificar y analizar las variables. Para esto, dSPACE viene con su propia interfaz gráfica del usuario llamada **CONTROL DESK**.

Ahora aprendamos como usar Control Desk para interactuar con la simulación en tiempo real.

2.4 Creando un nuevo archivo de experimento con Control Desk

Control Desk es un software que permite al usuario observar las variables, visualizar su comportamiento y modificar los parámetros de simulación interactuando directamente con la tarjeta DSP.

- Arranque Control Desk y seleccione solamente las opciones marcadas como se ve en la figura 2.7.
  La ventana **Tool** se visualiza en el fondo de la pantalla como se ve en la figura 2.7. Las pestañas inferiores del menú (Tabs) muestran las herramientas disponibles, en la figura solamente están disponibles 3 de las herramientas de trabajo: *Log Viewer*, *Interpreter* y *File Selector*.
  Como describiremos más tarde existe una herramienta muy importante a la cual prestaremos especial atención. Esta herramienta se llama **Variable Browser** y el **Parameter Editor** y suministra acceso a las variables de una aplicación. Estas variables son almacenadas en un archivo llamado **ejemplo1.sdf**.

  Así, para manipular las variables de una simulación debemos cargar el archivo *.sdf antes de arrancar el diseño gráfico.
- Primero iniciamos un nuevo experimento. Haga clic en **File/New Experiment**. En la ventana del menú que aparece, escriba el nombre del experimento y no olvide dar la ruta donde están almacenados los archivos de simulación. (Ver figura 2.8). Nota: cuando cree un nuevo experimento, no olvide dar el directorio de trabajo correcto (*Working directory*).
- Cargue el archivo que contiene las variables de simulación. Haga click en **File/Open** Variable File y seleccione ejemplo1.sdf.

La pestaña **Variable Manager** aparece al fondo de la pantalla (ver figura 2.9). La ventana contiene la estructura del modelo de simulación.
En el nivel más alto vemos las variables de control de la simulación. Su función se describe en la tabla 2.1.

```
MATLAB
File  Edit  Debug  Desktop  Window  Help
Current Directory: C:\matlab ejemplos
Shortcuts  How to Add  What's New

-------------------------------------------------------------------------
Starting build procedure with RTI 5.3.5 (RTI1104, 09-Nov-2005)
Model: "ejemplo1" (C:\matlab ejemplos\ejemplo1.mdl)
-------------------------------------------------------------------------
*** Using configuration set : "Configuration"
*** Working directory       : "C:\matlab ejemplos"
*** Optional User System Description File ejemplo1_usr.sdf not available
*** Initializing code generation
### Starting Real-Time Workshop build procedure for model: ejemplo1
### Generating code into build directory: C:\matlab ejemplos\ejemplo1_rti1104
### Invoking Target Language Compiler on ejemplo1.rtw
    tlc
    -r
    C:\matlab ejemplos\ejemplo1_rti1104\ejemplo1.rtw
    c:\dspace\matlab\rti1104\tlc\rti1104.tlc
    -OC:\matlab ejemplos\ejemplo1_rti1104
    -Ic:\dspace\matlab\rti1104\tlc
    -Ic:\dspace\matlab\rti1104\tlc
    -IC:\matlab ejemplos\ejemplo1_rti1104\tlc
    -IC:\MATLAB71\rtw\c\tlc\mw
    -IC:\MATLAB71\rtw\c\tlc\lib
    -IC:\MATLAB71\rtw\c\tlc\blocks
    -IC:\MATLAB71\rtw\c\tlc\fixpt
    -IC:\MATLAB71\stateflow\c\tlc
    -aEnforceIntegerDowncast=1
    -aFoldNonRolledExpr=1
    -aInlineInvariantSignals=0
    -aInlineParameters=0
    -aLocalBlockOutputs=0
    -aRollThreshold=5
    -aGenerateReport=0
    -aGenCodeOnly=0
    -aRTWVerbose=1
    -aIncludeHyperlinkInReport=0
```

```
### Building ejemplo1: dsmake -f ejemplo1.mk WORKINGBOARD=ds1104  EXTMODE_STATIC_ALLOC_SIZE=1000000

BUILDING APPLICATION (Single Timer Task Mode)

WORK  DIRECTORY "C:\matlab ejemplos"
BUILD DIRECTORY "C:\matlab ejemplos\ejemplo1_rti1104"
TARGET COMPILER "C:\PPCTools20w"

COMPILING  ejemplo1.c
COMPILING  ejemplo1_data.c
COMPILING  rt_nonfinite.c
COMPILING  C:\dSPACE\MATLAB\RTI1104\C\rti_sim_engine.c
COMPILING  C:\dSPACE\MATLAB\RTI1104\C\rti_init_c.c
COMPILING  C:\MATLAB71\rtw\c\src\ode1.c
COMPILING  C:\MATLAB71\rtw\c\src\rt_sim.c
COMPILING  C:\dSPACE\MATLAB\RTI1104\C\rti_assert.c

USING LIBRARY "C:\dSPACE\MATLAB\RTI1104\C\Lib\rtwlib_r14sp3_ds1104.lib"

LINKING APPLICATION ...
LINKING FINISHED

LOADING APPLICATION "ejemplo1.sdf" ...
[#1] ds1104 - RTI:  Initializing ... (720)
[#2] ds1104 - RTI:  Initialization completed (721)
[#3] ds1104 - RTI:  Simulation state: RUN (700)
LOADING FINISHED

MAKE PROCESS SUCCEEDED


### Successful completion of Real-Time Workshop build procedure for model: ejemplo1
*** Finished RTI build procedure for model ejemplo1
>>
Start
Inicio  MATLAB  Simulink Library Brow...  ejemplo1 *  Dibujo1 - Microsoft Visio  Laboratorio - Microso...  ES  14:53
```

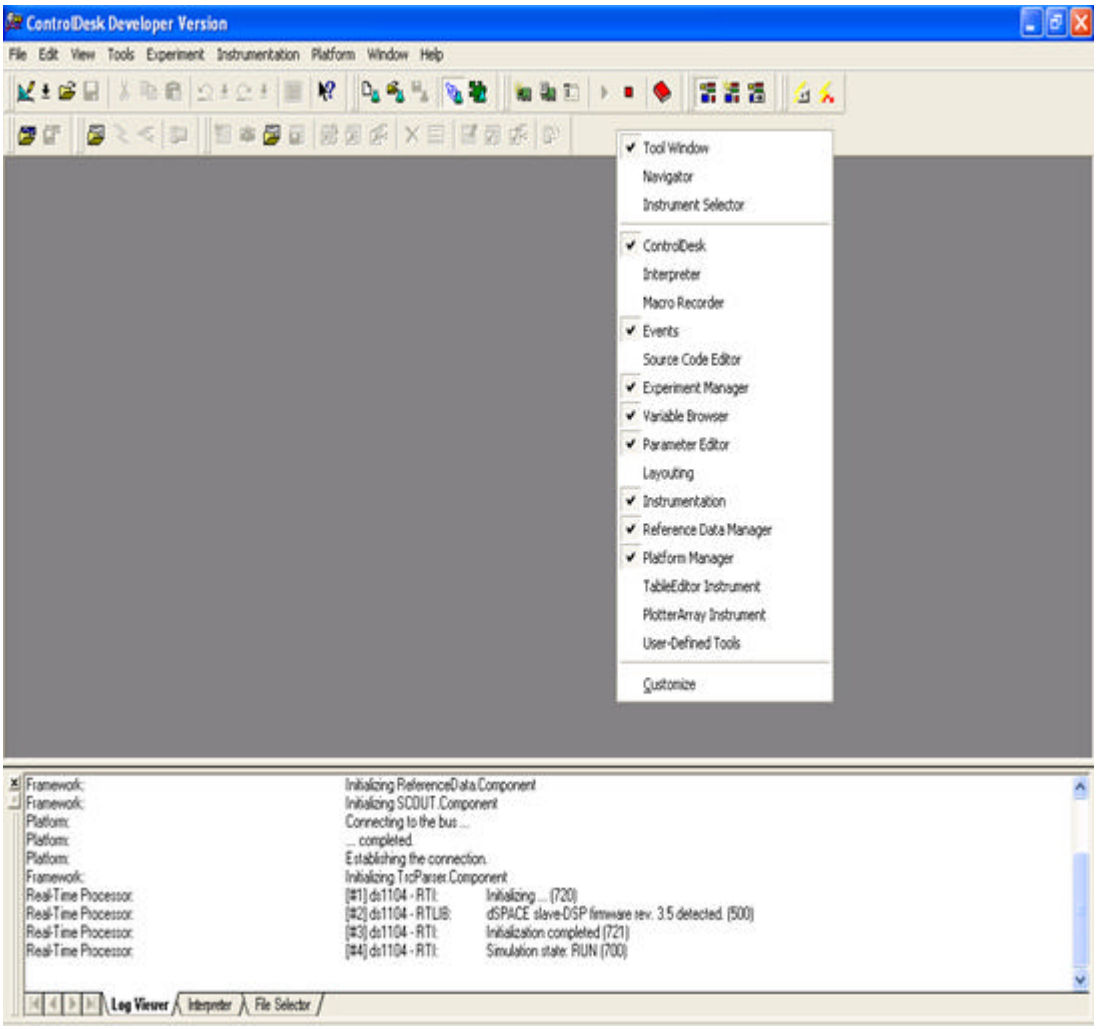


Figura 2.7 Pantalla de Control Desk

Para el propósito de nuestra simulación, las variables de interés están contenidas en el grupo **Model Root**. Este grupo contiene las variables que pertenecen al modelo de Simulink. Las variables de los subsistemas van en otros grupos organizados en niveles jerárquicos hacia abajo. Las variables disponibles tienen un prefijo para distinguir los diferentes tipos de ellas y se generan en el orden detallado en la tabla 2.1.

Prefijo: Ejemplo	**Tipo de Variable**
L: Salida	Señales etiquetadas
B: Integrador	Bloques de salida
S: Osciloscopio	Entradas de señales generadas
P: Amplitud generador de señal	Bloques de parámetros

Tabla 2.1 Tipos de Variable y señal de correlación

El nombre que sigue al prefijo es el nombre del bloque, con excepción de las señales etiquetadas donde la etiqueta misma es usada. Todas las variables correspondientes al **Model Root** para nuestro ejemplo se detallan en la tabla 2.2.

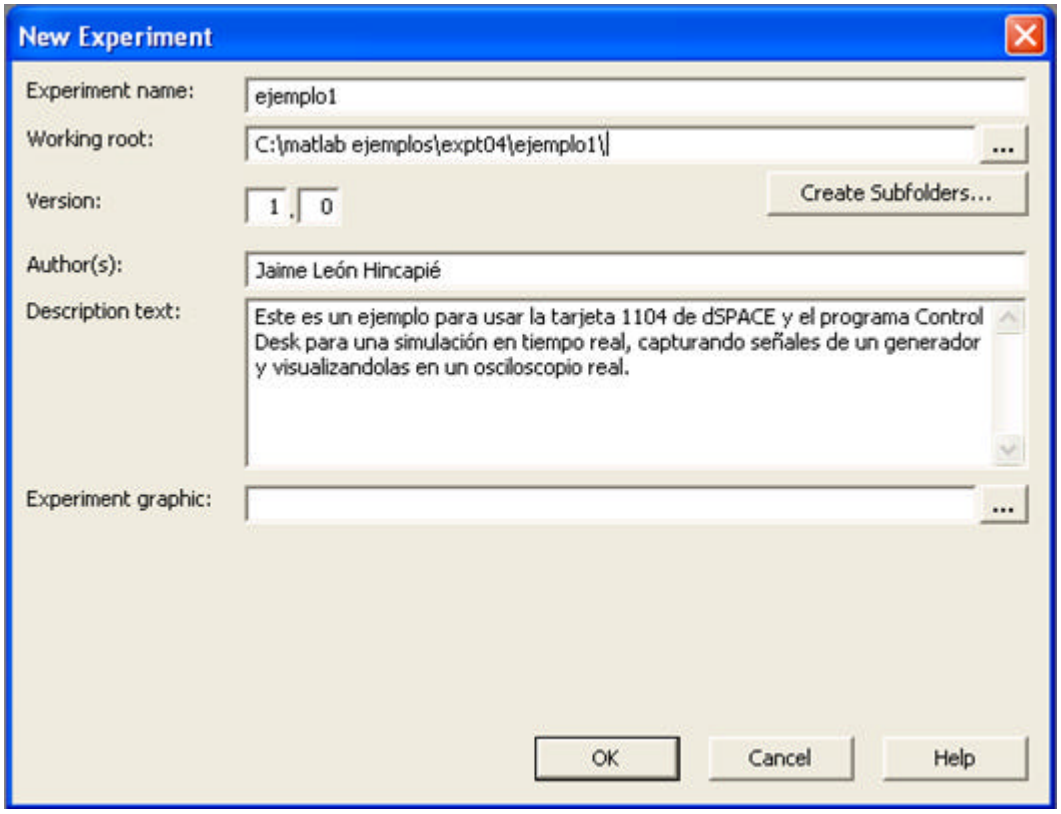


Figura 2.8: Parámetros del Experimento

Figura 2.9: Variable Tab Manager

2.5 Controles de Visualización y Osciloscopios con Instrumentation Management Tools

Con el fin de ver el comportamiento de cada variable y modificar los parámetros en tiempo real mientras el sistema está corriendo, necesitamos una serie de botones, comandos, ganancias deslizables (*slider gains*), graficadotes etc. Que puedan manipular estas variables.

Por lo tanto, necesitamos un nuevo "**LAYOUT**" en el cual se puedan añadir estos instrumentos.

- Haga click en: *File/New/Layout* del menu.

Dos nuevas ventanas aparecen en Control Desk como se muestra en la figura 2.10. Una de ellas llamada Layout1, contiene los instrumentos usados para manejar el experimento. La segunda ventana es en realidad una barra de herramientas con la cual podemos arrastrar y crear los controles necesarios para el experimento.

'Model Root/B:DS1104ADC_C5->ADC#5'	Visualiza la variable recibida en el canal A/D #5
'Model Root/B:Slider Gain›Out1'	Visualiza la salida de la ganancia deslizante
'Model Root/B:In-Gain'	Visualiza la salida del bloque de ganancia de entrada
'Model Root/B:Out-Gain'	Visualiza la salida del bloque de ganancia de salida
'Model Root/B:Transfer Fcn'	Visualiza la salida de la función de transferenc segundo orden
'Model Root/B:Slider Gain/P:Slider Gain.Gain'	Contiene el valor de la ganancia deslizante
'Model Root/P:In-Gain.Gain'	Contiene el valor de la ganancia de entrada
'Model Root/P:Out-Gain.Gain'	Contiene el valor de la ganancia de salida
'Model Root/P:Transfer Fcn.A(1)'	Maneja los parámetros de la función de transferencia (numerador y denominador)
'Model Root/P:Transfer Fcn.A(2)'	
'Model Root/P:Transfer Fcn.C(1)'	
'Model Root/P:Transfer Fcn.C(2)'	

Tabla 2.2: Variables manejadas por dSPACE en el experimento ejemplo1.

Los controles visualizados en *Virtual Instruments Toolbar* nos dejan manipular solamente las variables que pueden ser modificadas "en línea" (*on-line*), por ejemplo las variables tipo **P**. Nuestro ejemplo tiene 5 variables como esas. Cuatro de ellas están en la tabla 2.2 y hay otra más en los niveles jerárquicos de *Slider Gain*, la cual controla la amplitud del bloque de ganancia.

- Seleccione el botón *Slider* de la barra de herramientas de la derecha.
- El cursor se cambia a un cuadrado. Haga *click* y sostenga mientras arrastra una forma rectangular en la ventana *Layout1* (ver figura 2.11).

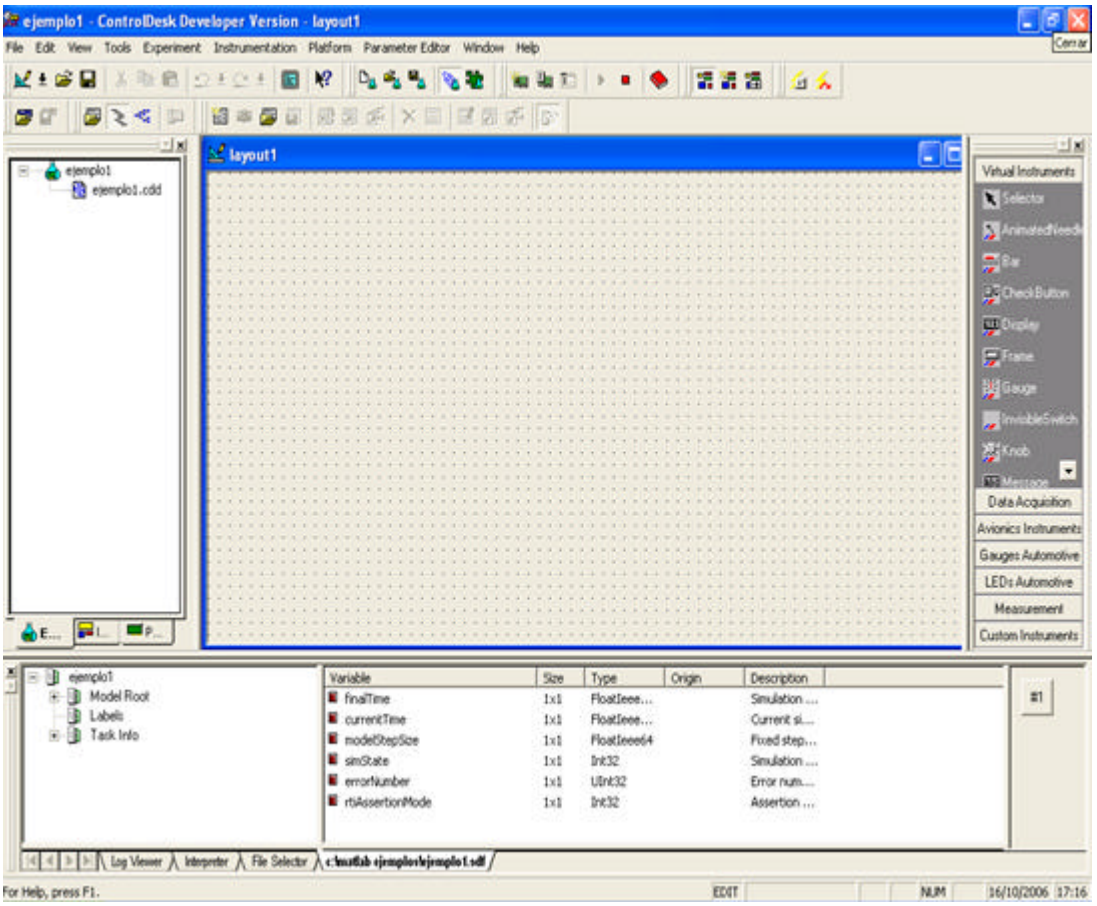


Figura 2.10: Nueva ventana Layout para instrumentación y control

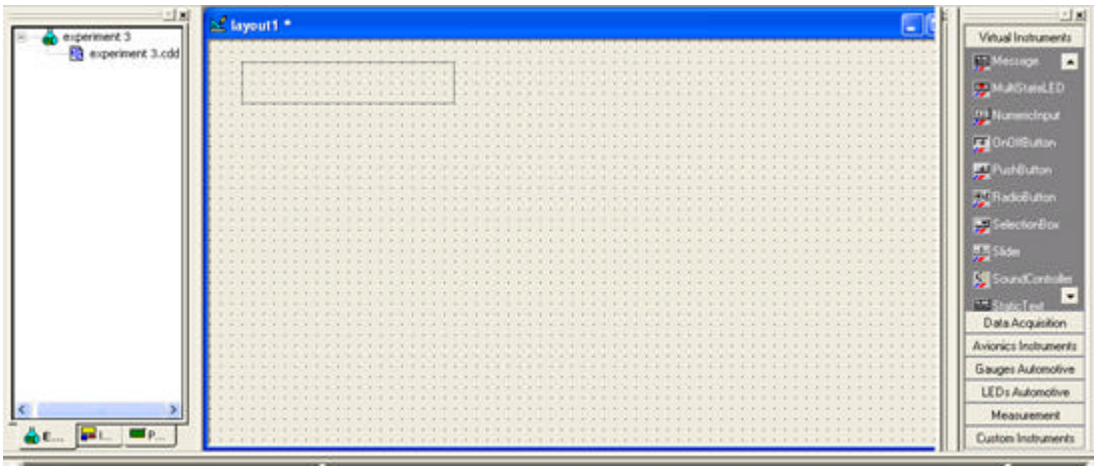


Figura 2.11: Selección de una barra deslizante de Visual Instruments.

- Cuando se suelta el ratón, usted tendrá un control deslizante, el cual le permitirá continuamente cambiar cualquier variable P dentro de los límites seleccionados.
- Si decidimos utilizar amplificaciones entre 0 y 5, haga doble *click* en el control deslizante seleccione la pestaña "Deslizador" (*Slider Tab*) y ajuste el rango mínimo y máximo como se muestra en la figura 2.12.
- Ahora los límites de ajuste del cursor son 0 y 5. Pero el deslizador se halla todavía bordeado por una línea roja. Esto significa que todavía no se ha asignado una variable a controlar.
- En la ventana *Variable Manager*, en la parte baja de la pantalla, seleccione **Slider Gain**
- Haga click en la variable **P:Slider Gain.Gain** y arrástrela al rectángulo dibujado en la ventana *Layout*.
- El nuevo control **Slider** mostrará la variable manipulada y desaparecerá el borde rojo como se muestra en la figura 2.13.

Ahora podemos añadir algunos instrumentos de visualización.

En nuestro ejemplo solamente dos variables merecen ser monitoreadas: la entrada del sistema, la cual viene del generador de señales real conectado a **ADC#5** y la salida del sistema de segundo orden. Las dos pueden ser monitoreadas en el mismo osciloscopio. La señal de salida también se puede observar en el osciloscopio conectado al canal **DAC#1** en la interfaz de dSPACE. Recuerde que los valores reales de las señales de entrada se obtienen después del bloque **In-Gain** mientras que los valores reales de la señal de salida se obtienen antes del bloque **Out-Gain**.

- Haga click en la pestaña **Data Adquisition**, en la barra de instrumentos de la derecha.
- Seleccione el icono *Plotter* y dibuje un rectángulo grande en la ventana *Layout*.
- Arrastre las señales que necesite al monitor en el nuevo graficador (*plotter*): **´Model Root/In-Gain->Out1´ y ´Model Root/Slider Gain->Out1´**

- Cuando arrastre la segunda señal, esté seguro que libera el botón del ratón sobre la primera, en el eje vertical. De otro modo, se dibujará un nuevo eje vertical y tendrá menos espacio disponible para visualizar las formas de onda. Ahora las dos señales están asignadas al *plotter* y se visualizarán en diferentes colores. La etiqueta en el eje vertical mostrará únicamente la última señal arrastrada al osciloscopio. Su *Layout* debe ser como el que aparece en la figura 2.14.

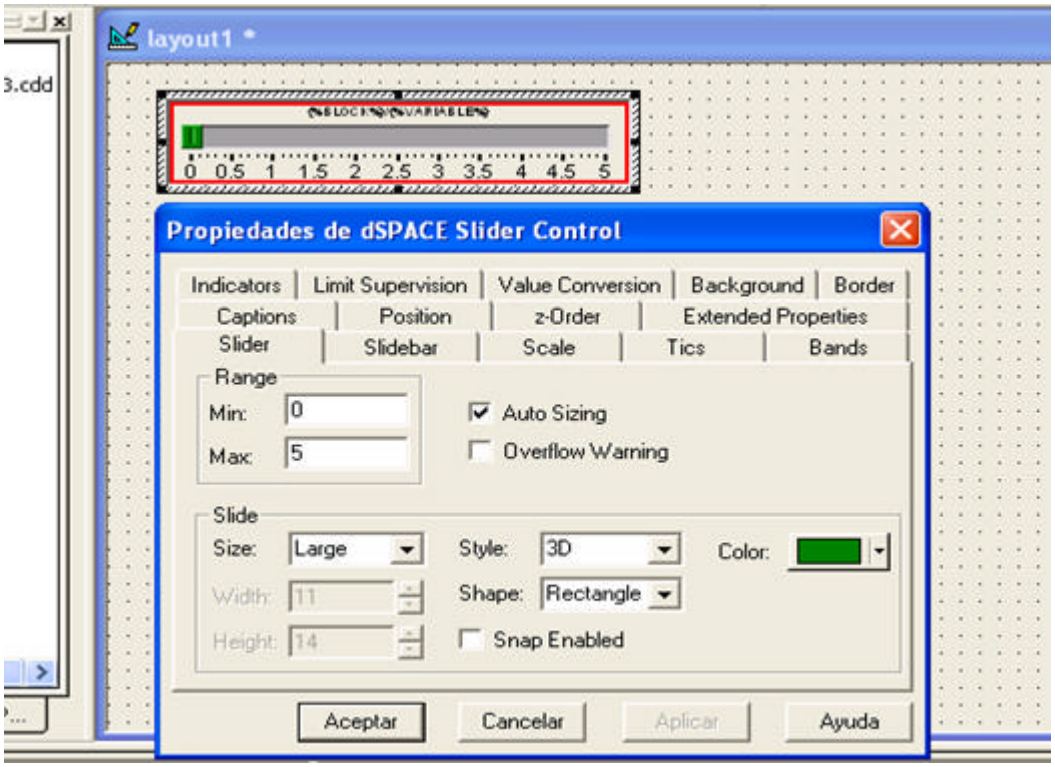


Figura 2.12: Ventana de ajuste de parámetros de control del "Slider"

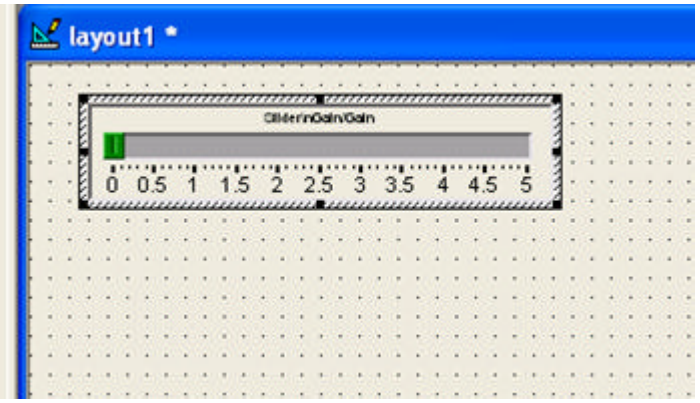


Figura 2.13: Control de "Slider" con variable manipulada y sus límites de control

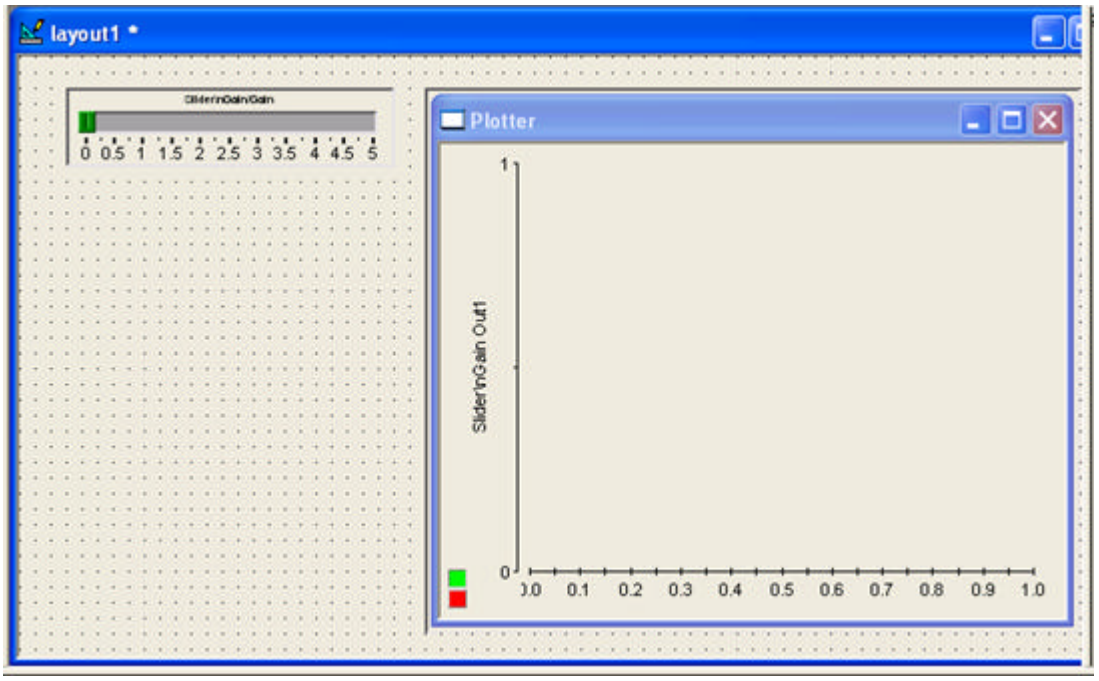


Figura 2.14: Haciendo un graficador para monitoreo de señales

- Cada vez que quiera hacer una corrección o mirar cuál señal fue agregada a la gráfica, haga click derecho en el área del osciloscopio y seleccione el comando **Edit Data Connections**. Usted puede borrar cualquier señal seleccionándola y oprimiendo la tecla Delete. Usted puede también borrar o modificar sus señales o cambiar su color haciendo doble click en el eje y seleccionando *"Signals"*.

- Par dar formato al osciloscopio, de clic derecho en el área del graficador y seleccione *plotter properties*, luego puede seleccione la pestaña Y-axis o X-axis. Cuando las señales se visualizan en el mismo eje Y, los ajustes serán los mismos para las dos señales. De otra manera, tendrá que seleccionar diferentes ejes Y con límites para cada una de las señales independientemente.

- Ajuste los límites de Y, Ancho, Modo de escalamiento como se muestra en la figura 2.15.

- En la etiqueta X usted puede escribir "Tiempo" y en la etiqueta Y escriba "señales de entrada y salida". Un paso más antes de empezar la simulación: ajuste de los parámetros de visualización. Los parámetros del proceso pueden ser ajustados mediante la ventana **Capture Settings Window** en el menú **View/Controlbars**.

- Abra la ventana **Capture Settings** y ajuste el tamaño (Length) de la simulación en 2, dejando sin cambiar el **Downsampling number**. Sus ajustes de captura deben

ser como los mostrados en la figura 2.16. Para sistemas más complejos este número debe ser incrementado cuando el parámetro **Length** es mayor que 20 veces el tiempo de muestreo.

Ahora estamos listos para empezar la simulación.

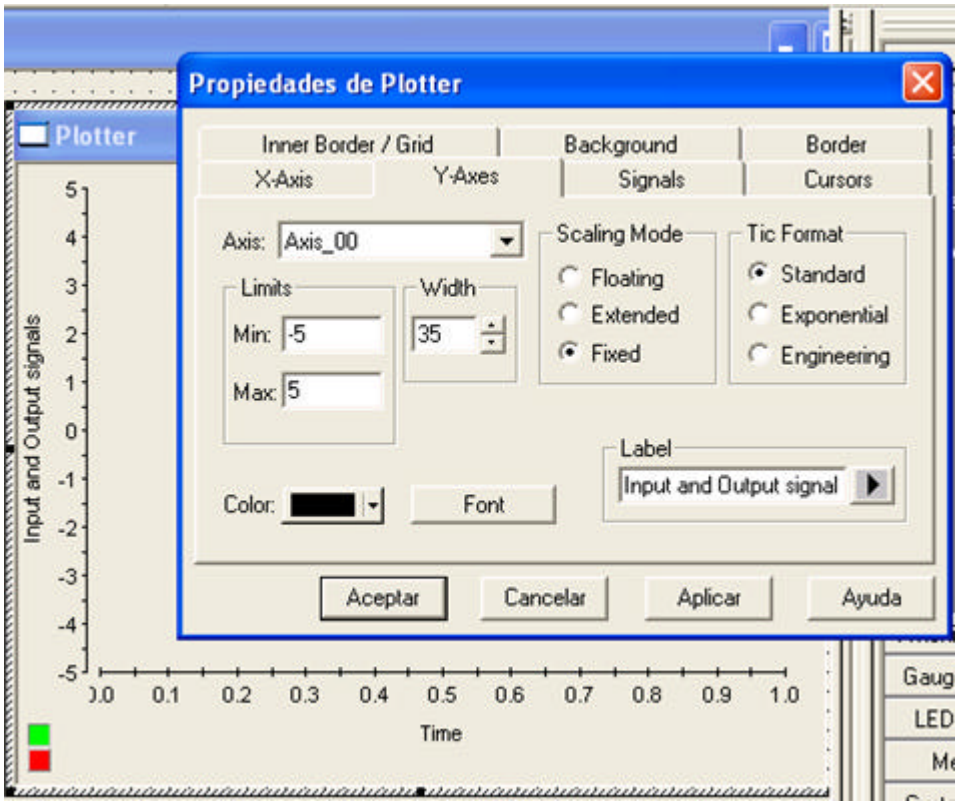


Figura 2.15: Ventana Propiedades de Plotter

2.6 Corriendo el experimento

Hay dos operaciones que deben hacerse para correr/arrancar (*run/stop*) el experimento. Primero se debe arrancar la ejecución DSP. Después se debe iniciar la animación (*animation*) y *Data acquisition/printing*. Cuando se para un experimento las operaciones deben hacerse en orden inverso.

- Para arrancar y parar el DSP se pueden usar los iconos de la barra de herramientas *Platform Management Toolbar* (ver figura 2.17).
- La barra de herramientas *Edit*, contiene iconos de edición, chequeo y animación. Refiérase a la figura 2.17 para ver los íconos de edición, chequeo y animación.

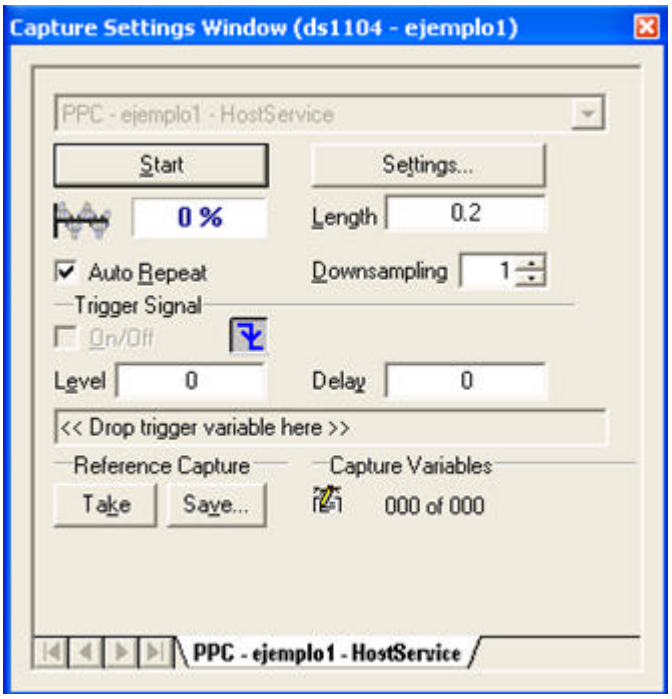


Figura 2.16: Ventana ajuste de captura

- El modo de edición es usado para editar o modificar el *Layout*, arrancar y parar la simulación en tiempo real. Cuando se empieza la simulación en tiempo real, haga click en el icono de animación esto le permitirá modificar los parámetros en tiempo real y observar la forma de onda. Usted no podrá seleccionar este modo mientras no empiece la simulación.

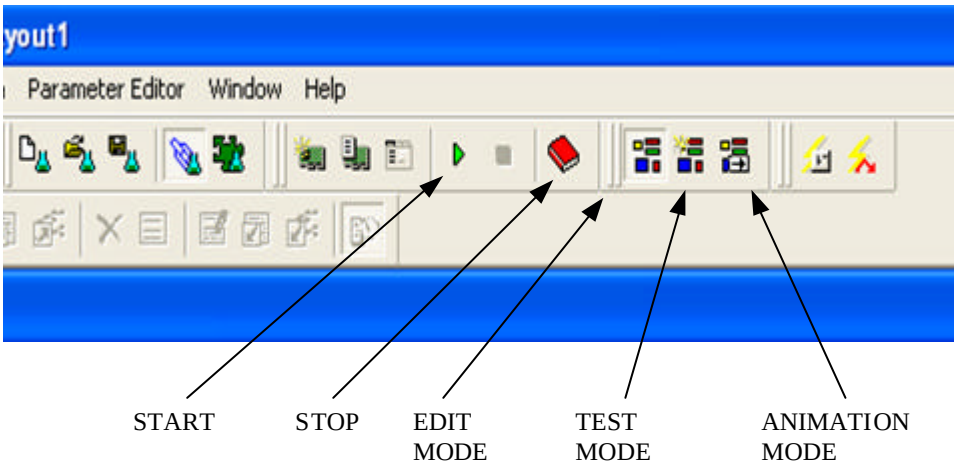


Figura 2.17: Barra de herramientas de ejecución y animación

- Inicie el DSP presionando el triangulo verde. Si ejemplo1.obj se carga en la memoria del DSP entonces empezará a correr y el rectángulo rojo de parada se activará.
- Arranque la animación presionando el icono de animación de la barra de herramientas *Edit*.
- Usted verá las señales del generador de señal y de la salida del sistema en el graficador. Cada dos segundos la pantalla se borrará y se visualizarán nuevos datos.
- Cambie algunos de los parámetros y observe las modificaciones de las señales que se visualizan. También observe las formas de onda en las ventanas del graficador, ellas podrán parecerse a las de la figura 2.18, donde también se muestra la ganancia del *slider gain*. Ahora el experimento está corriendo y hemos terminado nuestro diseño.

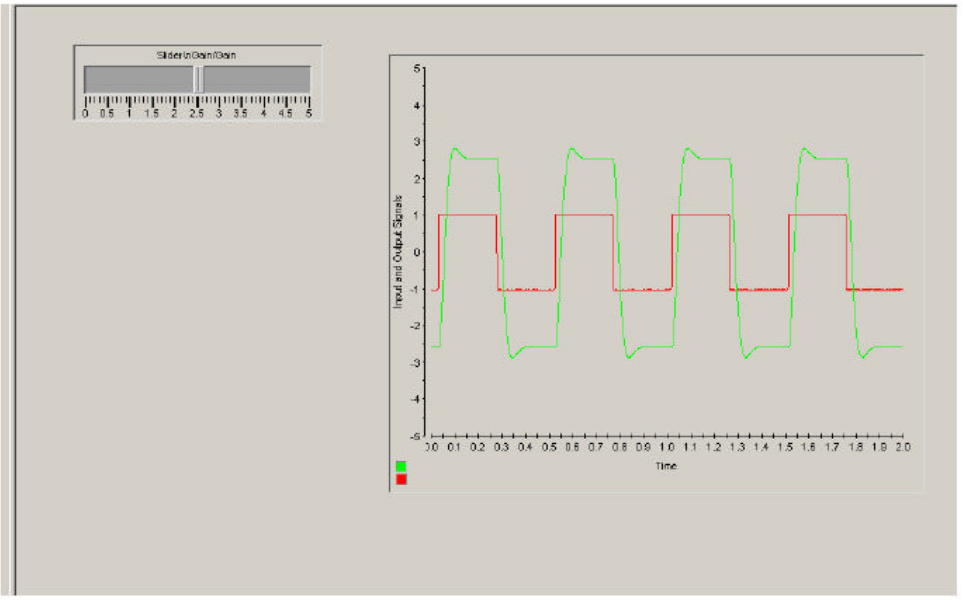


Figura 2.18 Formas de onda de la simulación en tiempo real

- Para salvar un experimento siga los siguientes pasos de lo contrario usted podría perder valiosa información:

1. Haga clic en *File/Add All Opened Files*. Esto implica que el experimento recordará el archivo *.sdf que contiene las variables manipuladas, reconocerá la ruta para todos los otros archivos y abrirá las conexiones entre el *layout* y las variables.
2. Salve el *layout* en un archivo con la extensión *.lay.
3. Salve el experimento en un archivo tipo *.cdx.

- Para cargar un experimento, simplemente haga clic en File/opened experiment y si fue salvado de acuerdo a las instrucciones anteriores aparecerá el layout y las variables en la pantalla.

Nota importante: Recuerde que el archivo objeto compilado debe estar en la memoria del DSP. No se carga implícitamente con el experimento.

2.7 Reporte de laboratorio

El reporte de laboratorio debe contener todos los detalles realizados en el mismo y los puntos siguientes:

- Resultados de la simulación del sistema de segundo orden obtenido en la sección 2.4.
- Resultados obtenidos en el osciloscopio y en Control Desk en la sección 2.6.
- Compare los resultados de la simulación en tiempo real y haga algún comentario sobre los mismos, cambie el valor de la ganancia deslizante en la simulación en tiempo real y observe su efecto.

Experimento 3

Conversor DC de conmutación y chequeo del motor sin carga [17]

3.1 Introducción

En este experimento se estudiará el funcionamiento de un conversor de potencia DC de conmutación. Primero se hará el diagrama de bloques para ser utilizado en la simulación de la operación del conversor. Después se implementará un algoritmo PWM para un conversor de potencia DC. El propósito de la implementación en tiempo real es obtener voltaje variable en la salida del conversor e potencia mientras se controla su amplitud con una interfaz de usuario basada en dSPACE.

3.2 Simulación del conversor DC de conmutación.

3.2.1 Forma de onda triangular

Para modular el ancho de los pulsos de voltaje en un conversor de potencia, un control de voltaje debe compararse con una señal triangular. Antes de construir el modelo para el conversor e debe construir un generador de onda triangular en simulink usando el bloque de secuencias *sources/repeating.*

- Cree un nuevo directorio para el experimento expt03
- Arranque matlab y ajuste la ruta a este directorio.
- Abra un nuevo modelo de simulink.
- Arrastre y suelte el bloque Repeating Sequence de la librería Sources.
- Haga doble clic en el bloque y entre los siguientes datos:
  - ✓ *Time values:* [0 0,5/fsw 1/fsw]
  - ✓ *Output values* [-1 1 -1]

Donde la variable fsw será nuestra frecuencia de conmutación seleccionada como una variable global en *Matlab.*

- Añada un osciloscopio al bloque de salida *Repeating Secuence*.
- Introduzca el valor de fsw en *Matlab* (por ejemplo fsw=10000, que significa 10Khz.).
- Ajuste los parámetros de simulación en los siguientes valores:
  - ✓ *Stop time*: 0.002
  - ✓ *Fixed Step Size*: 0.000001
  - ✓ *Solver options: fixed step, ode1 (Euler*)
- Corra el experimento

El resultado debe ser similar al de la figura 3.1

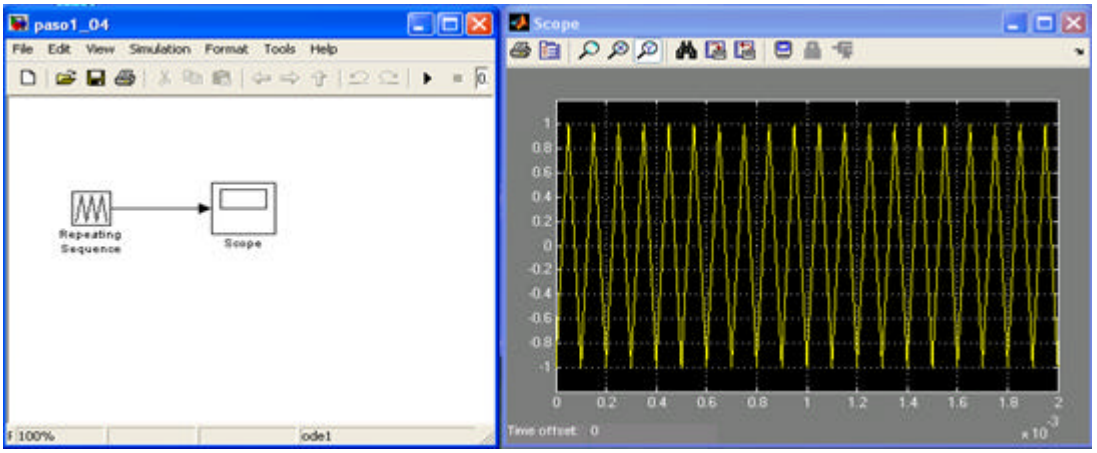


Figura 3.1: Generador de onda triangular con una frecuencia de 10 Khz.

3.2.2 Relación de trabajo (Duty Ratio) y función de conmutación

Para obtener la relación de trabajo de la generación de un PWM, podemos usar las ecuaciones [1]:

$$d_A = \frac{1}{2} + \frac{1}{2}\frac{v_{control,A}}{\hat{V}_{tri}} \tag{3.1}$$

$$d_A = \frac{\overline{v}_{AN}}{V_d} \tag{3.2}$$

Solucionando las ecuaciones 3.1 y 3.2 y considerando la amplitud del voltaje triangular igual a la unidad; $\hat{V}_{tri} = 1$, obtenemos la relación para el control de voltaje de un conmutador:

$$v_{control,A} = \frac{2\overline{v}_{AN}}{V_d} - 1 \tag{3.3}$$

La relación 3.3 se implementa en Simulink. El voltaje de control se compara con la señal triangular. El bloque relé (*Relay Block*) de salida se ajusta en 1 cuando la diferencia es positiva y 0 cuando es negativa.

El voltaje deseado $\overline{v}_{AN}$ con respecto a tierra DC, se ajusta por medio de un bloque Constante (*Constant Block*) a un valor de 1, y puede ser variado con una ganancia deslizable (*Slider Gain*) desde 0 hasta el máximo valor del bus DC V_d. El modelo en Simulink se muestra en la figura 3.2.

3.2.3 Modelo del conversor de dos polos

Para obtener el modelo de un conversor de DC necesitamos añadir a nuestro modelo dos bloques para un conmutador. Un bloque es el ***Switch*** de la librería ***Non Linear***, el cual Permite que la señal superior pase cuando la entrada intermedia es más grande que el umbral especificado y la señal inferior es el caso opuesto. El voltaje de salida del conversor será la diferencia entre los dos voltajes de salida del conmutador medidos con respecto a la tierra DC.

Ahora, el valor de referencia para nuestro modelo será v_{AB}. Sabemos que:

$$\bar{v}_{AB} = \bar{v}_{AN-}\bar{v}_{BN} \tag{3.4}$$

En algún instante de tiempo, los voltajes de control de los dos conmutadores son complementarios así:

$$v_{control,A} = -v_{control,B} \tag{3.5}$$

Así, resolviendo la ecuación 3.1 mediante 3.5, podemos escribir:

$$v_{control,A} = -v_{control,B} = \frac{\bar{v}_{AB}}{V_d} \tag{3.6}$$

Y

$$d_A = \frac{1}{2}(v_{control,A} + 1)$$

$$d_B = \frac{1}{2}(v_{control,A} + 1) \tag{3.7}$$

Estas relaciones pueden ser implementadas en un modelo Simulink, junto con los conmutadores de dos polos como en la figura 3.3. La entrada es el valor deseado del voltaje promedio de salida v_{AB}. El valor instantáneo del voltaje de salida será una señal cuadrada y el valor promedio será igual al valor ajustado con la ganancia deslizable (*Slider Gain*).

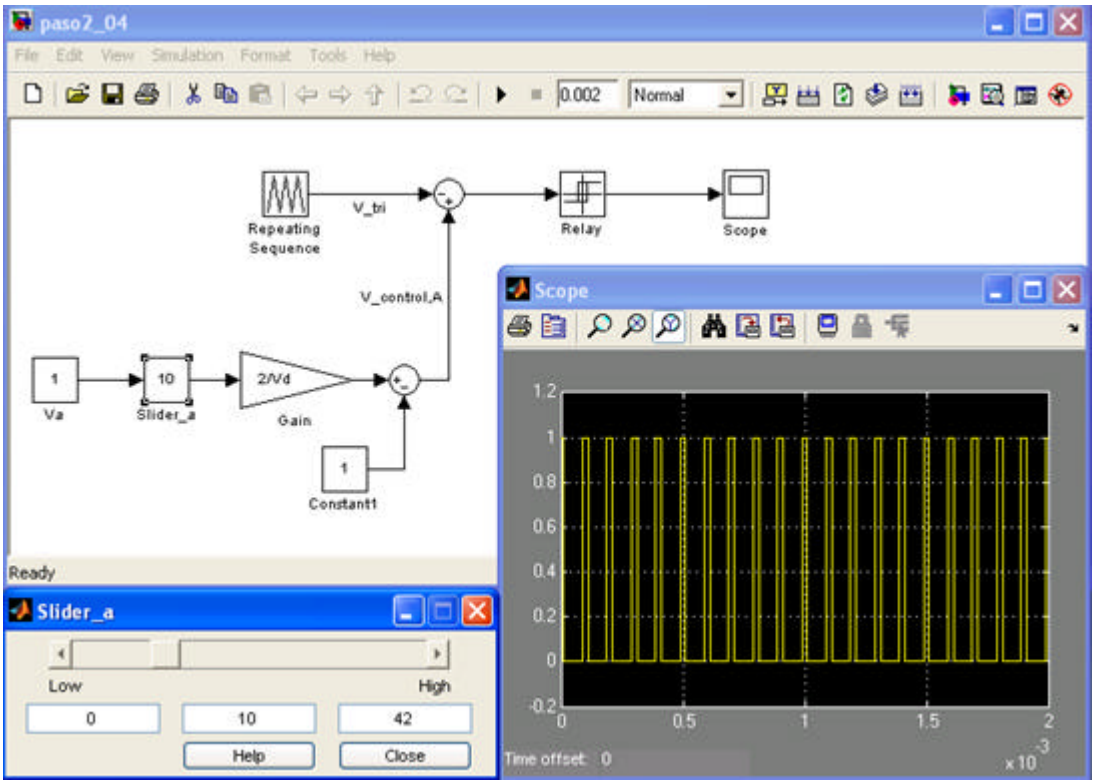


Figura 3.2: Generación de la señal de conmutación para un inversor de un polo simple

Variando el valor de la ganancia deslizable, (por ejemplo, variando el valor deseado del voltaje promedio de salida), cambia el voltaje de salida. Ahora, ajuste el tiempo de simulación a ***inf*** y empiece a simular. Varíe el valor de la ganancia y observe la forma de onda del voltaje de salida. Ajuste los ejes del osciloscopio como se muestra en la figura 3.3.