

## ANEXO I

## DESCRIPCIÓN DE BLOQUES RTI DE LA TARJETA DE PROCESAMIENTO DE SEÑAL DSPACE 1102

Una vez obtenido el modelo (función de transferencia) de la planta, y después de haber diseñado los controladores desde el entorno de Simulink en Matlab en el capítulo 2 de la presente tesis; se procede a realizar la parte experimental y simulación del sistema bola-varilla desde SIMULINK Real-Time Workshop, haciendo uso del hardware de la tarjeta de procesamiento de señal DSPACE1102:

### ACCEDER A LA LIBRERÍA RTI DE LA TARJETA DSPACE1102:

Existen dos maneras para poder acceder a la librería RTI:

1. **Opción A:** Se puede cargar directamente desde la pantalla de simulink, para esto tenemos que ingresar a través de la ventana de comandos de MATLAB. Colocando **simulink** en la línea de comandos o dando doble clic al icono en la parte de la barra de menús como se muestra en la Fig. 1.1:

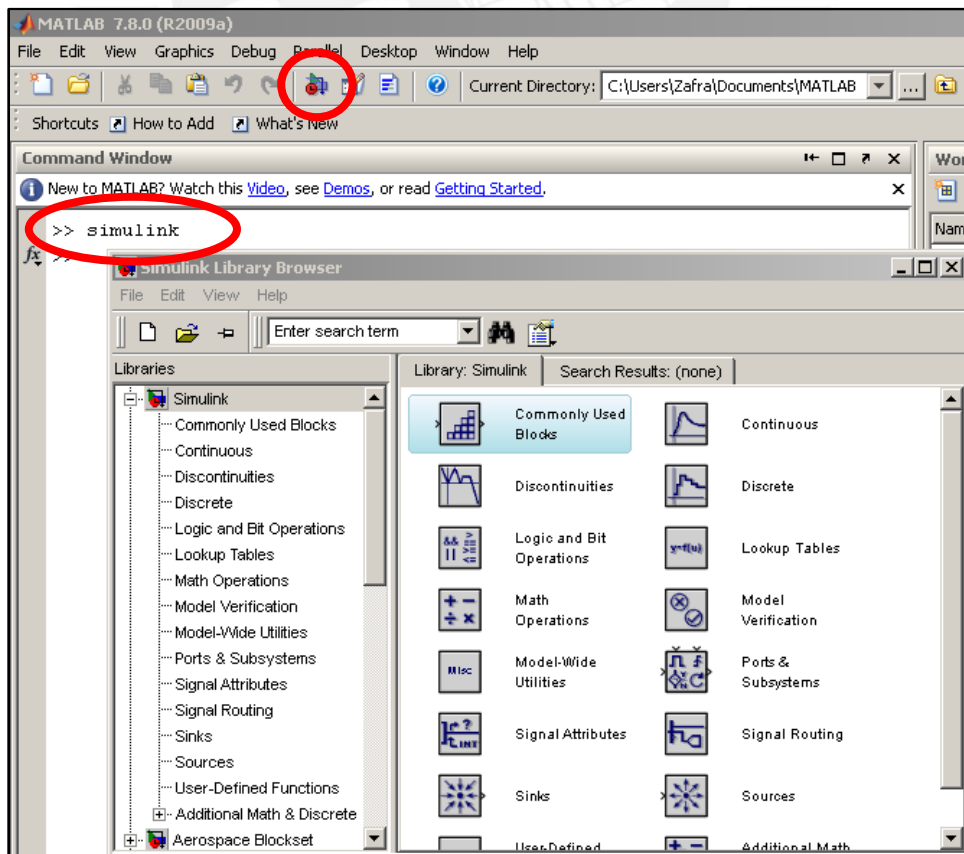


Fig. 1.1: Acceder a la Librería RTI por la opción A

Luego de abrir la ventana de simulink, ingresamos en la librería de RTI de la tarjeta DSpace 1102, como se muestra a través de la Fig. 1.2:

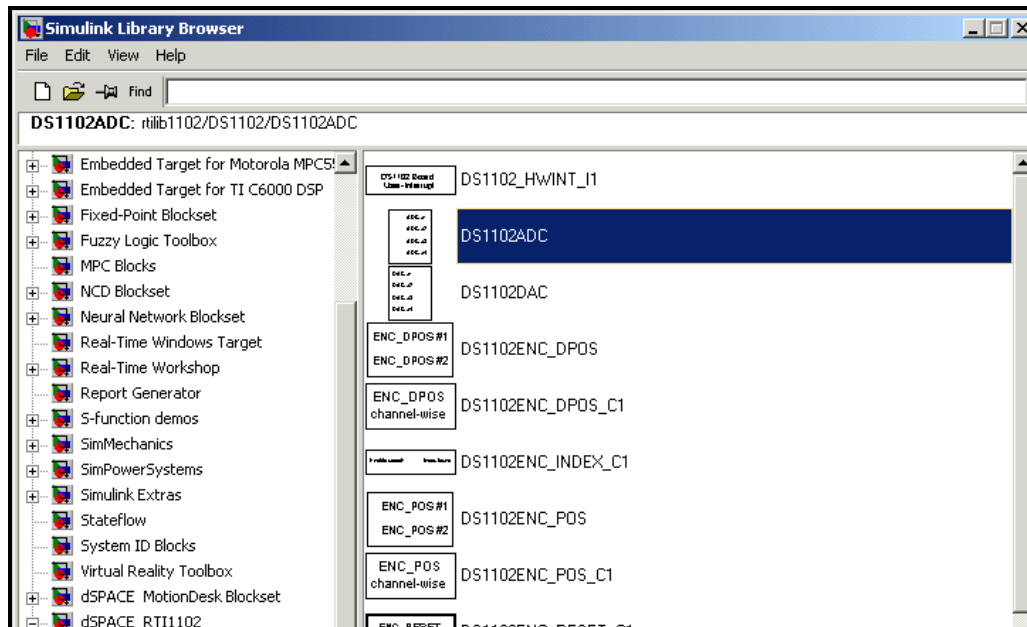


Fig. 1.2: Librería RTI

2. **Opción B:** Otra manera de acceder a los bloques es escribiendo “RTI” desde la ventana de comandos de MATLAB. Se muestra la Fig. 1.3 que muestra la ventana donde se encuentran los boques RTI a utilizar:

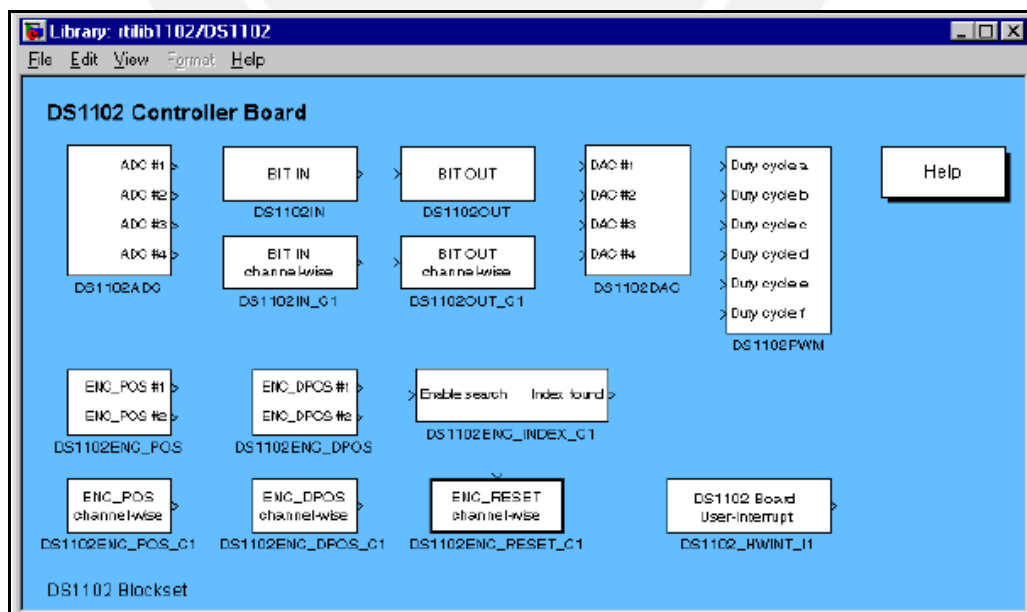


Fig. 1.3: Ventana con librería RTI para la opción B

Como se puede apreciar en la Fig. 1.3, además de los bloques de funciones, existen otros como el de ayuda, que sirve de orientación sobre la funcionalidad y características de cada uno de los bloques

**BLOQUES DE LIBRERÍA RTI:**

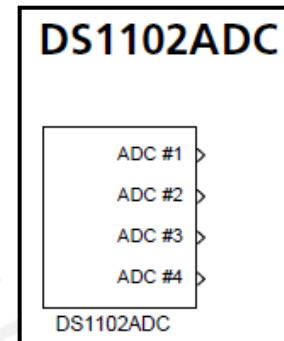
A continuación describiremos lo bloques utilizados en esta tesis:

**Bloque DS1102ADC:**

**Propósito:** Provee el acceso de lectura de los cuatro convertidores A/D paralelos.

**Características:**

- La escala entre de la entrada analógica de voltaje y la salida del bloque es:



Rango de voltaje de entrada	Salida de simulink
±10 Voltios	±1

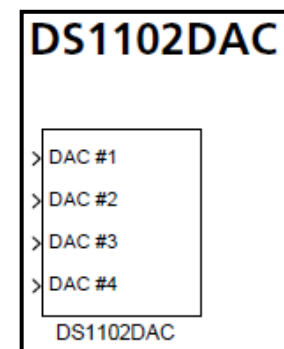
- La resolución de los canales 1 y 2 es 16 bits, mientras que los canales 3 y 4 proveen 12 bits.
- El tiempo de conversión de los canales 1 y 2 es casi 4µs, la conversión de los canales 3 y4 toma 1.25 µs cada uno.

**Bloque DS1102DAC:**

**Propósito:** Provee el acceso de escritura para los cuatro convertidores D/A paralelos.

**Características:**

- La escala entre de la salida analógica de voltaje y la entrada del bloque es:



Rango de voltaje de salida	Entrada de simulink
±10 Voltios	±1

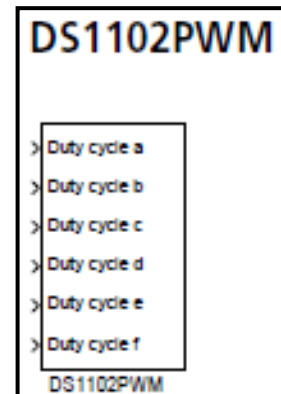
- La tarjeta provee sus salidas en modo *unlatched*, que significa que cada canal es convertido y enviado inmediatamente.
- La resolución de todos los canales es 12 bits.

**Bloque DS1102PWM:**

**Propósito:** Para llevar a cabo la generación de señales PWM en un máximo de seis canales.

**Características de I/O:**

- La escala entre el ciclo de trabajo y la entrada del bloque:

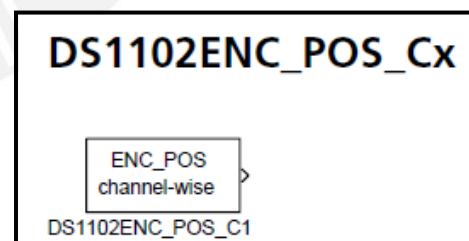


Ciclo de trabajo (Duty Cycle)	Entrada de simulink
0... 100%	0 ... 1

- Si los canales 5 y 6, etiquetados con *Duty cycle e* y *Duty cycle f*, están desconectados, el bloque DS1102PWM representa a un generador de PWM de 4 canales. Si uno de los canales 5 y 6 es conectado, el bloque representa un generador de PWM de 6 canales.
- La resolución PWM es de 40 ns para frecuencias de PWM por encima de aproximadamente 385 Hz, y 160 ns para otras.

**Bloque DS11002 End Pos Cx:**

**Propósito:** Para leer la posición de uno de los dos canales de la interfaz del encoder.



**Descripción:**

Cada canal de la interfaz de encoder contiene un contador de 24 bits para dar la posición absoluta del transductor conectado, correspondiendo a un rango entero de  $-2^{23} \dots +2^{23} - 1$ . Este rango es automáticamente escalado a  $-1 \dots +1$ . Para recibir el número de cuentas como un valor entero, el valor de la salida será multiplicado por  $2^{23}$ .

Cada canal de la interfaz del codificador utiliza una línea de subdivisión de 4

veces para mejorar la resolución. Esto significa que cada línea del encoder produce 4 cuentas en el contador de posición.

**Características de I/O:**

- La escala entre las líneas físicas del encoder y la salida del bloque:

Líneas del encoder	Salida de simulink
$-2^{23} \dots +2^{23} - 1$	$\pm 1$

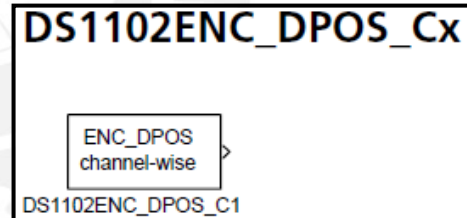
- La salida del bloque está relacionado con uno de los dos contadores de posición del encoder individual.

**Inicialización:**

Cuando una aplicación es cargada a la DS1102, el contador de posición del canal seleccionado es automáticamente borrado, lo que significa que se establece en cero antes que la aplicación en tiempo real se inicie.

**Bloque DS1102 ENC DPOS Cx:**

**Propósito:** Provee el acceso de lectura de la posición delta de uno de los dos canales de entrada de la interfaz de encoder.



**Descripción:**

La posición delta representa la diferencia escalada de los valores de dos posiciones de un canal. Cada DS1102 canal de la interfaz de encoder contiene un contador de 24 bits para dar la posición absoluta del transductor conectado, correspondiendo a un entero del rango  $-2^{23} \dots +2^{23} - 1$ . Este rango es automáticamente escalado a  $-1 \dots +1$ . Para recibir el número de cuentas como un valor entero, el valor de la salida será multiplicado por  $2^{23}$ .

Cada canal de la interfaz del codificador utiliza una línea de subdivisión de 4, para mejorar la resolución. Esto significa que cada línea del encoder produce 4 cuentas en el contador de posición.

**Características de I/O:**

- La escala entre las líneas físicas del encoder y la salida del bloque:

Líneas del encoder	Salida de simulink
$-2^{23} \dots +2^{23} - 1$	$\pm 1$

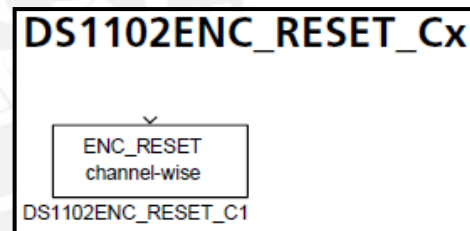
- La salida del bloque está relacionado con uno de los dos contadores de posición del encoder individual.

**Inicialización:**

Cuando una aplicación es cargada a la DS1102, el contador de posición del canal seleccionado es automáticamente borrado, lo que significa que se establece en cero antes que la aplicación en tiempo real se inicie.

**Bloque DS1102 ENC RESET Cx:**

**Propósito:** Provee el acceso de escritura para los contadores de posición de los dos canales de la interfaz de encoder. Si el bloque se dispara el valor del contador especificado se pone a cero.



**Características de I/O:**

- La escala entre las líneas físicas del encoder y la salida del bloque:

Líneas del encoder	Salida de simulink
$-2^{21} \dots +2^{21} - 0.25$	$\pm 1$

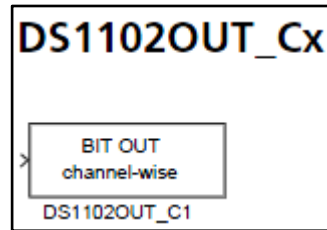
- La salida del bloque está relacionado con uno de los dos contadores de posición del encoder individual.

**Inicialización:**

Cuando una aplicación es cargada a la DS1102, el contador de posición del canal seleccionado es automáticamente borrado, lo que significa que se establece en cero antes que la aplicación en tiempo real comience.



**Bloque DS1102OUT\_Cx:**



**Propósito:** Escribir un único bit de una salida digital de 16 bits.

**Características de I/O:**

- RTI admite la escritura de datos para este bloque. Como resultado de ello, el bloque puede ser utilizado con todos los operadores lógicos sin necesidad de utilizar el bloque de conversión de datos.
- Relación entre la salida digital y la entrada del bloque:

Salida digital (TTL)	Entrada Simulink	
	Sin datos	Con datos
High	> 0 (double)	1 (boolean)
Low	≤ 0 (double)	0 (boolean)

**CALCULANDO EL ÁNGULO EN RADIANES DEL ENCODER**

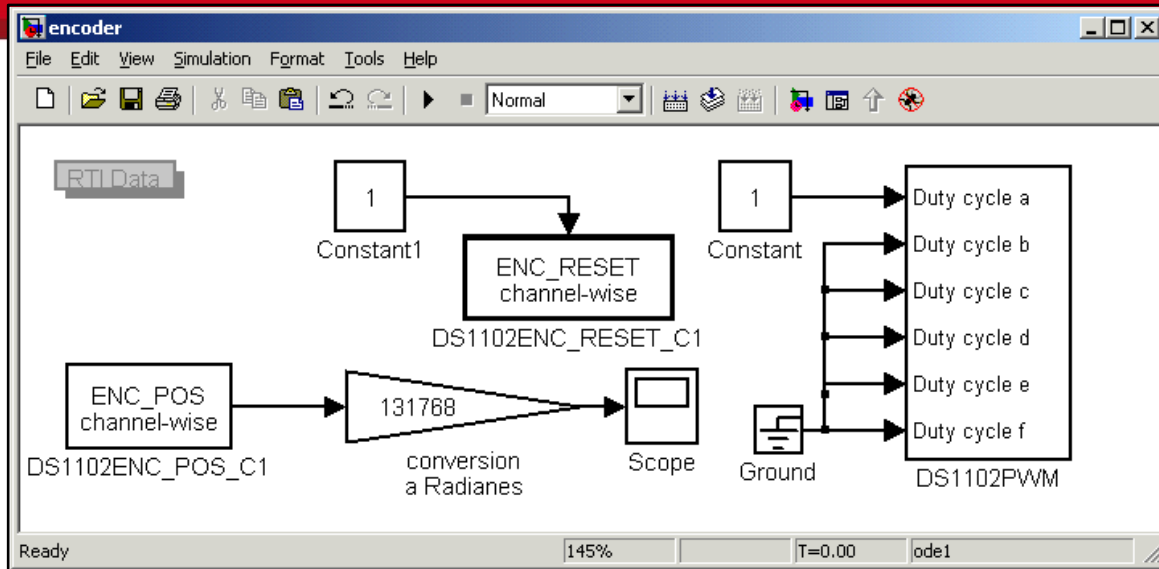
Para recibir el ángulo  $\phi$  en radianes desde el valor de salida escalado de los bloques DS1102ENC\_POS, DS1102ENC\_POS\_Cx, DS1102ENC\_DPOS y DS1102ENC\_DPOS\_Cx, se debe aplicar el siguiente cálculo:

$$\phi = 2^{23} \times \frac{2 \times \pi}{4 \times (\text{Cuentas\_Encoder})} \times (\text{Salida\_escalada})$$

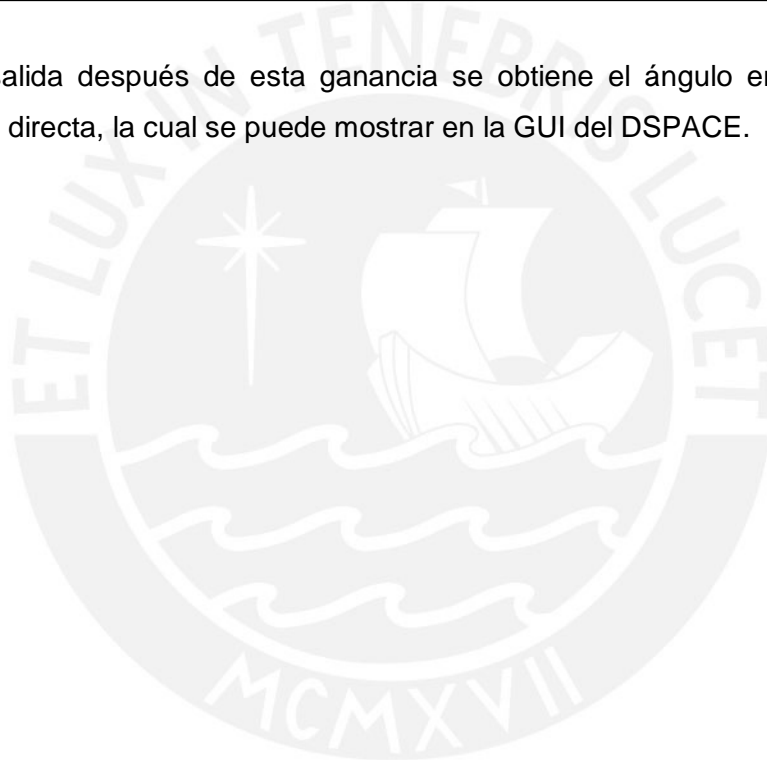
Para nuestro caso utilizamos 100 cuentas (líneas) por revolución, por lo tanto tenemos que escalar el valor de la salida por el factor:

$$2^{23} \times \frac{2 \times \pi}{4 \times 100} = 131768$$

Este valor lo introducimos en el diagrama simulink como una ganancia, con el bloque **Gain**. Por ejemplo se muestra en la siguiente figura a la salida del bloque **ENC\_POS channel-wise**:



En la salida después de esta ganancia se obtiene el ángulo en radianes de manera directa, la cual se puede mostrar en la GUI del DSPACE.





## ANEXO II

## CONTROLDESK E INTERFAZ GRÁFICA DE USUARIO

El propósito de este anexo es presentar la información necesaria para familiarizarse con el entorno ControlDesk, interactuando con la tarjeta de control dSPACE para crear interfaces gráficas para la adquisición y control de sistemas creados desde la plataforma Simulink.

Antes de crear una interfaz gráfica de usuario, empezaremos por el procedimiento a seguir desde Simulink, Real Time Workshop (RTW) y Real Time Interface (RTI):

- Se debe asegurar que se haya instalado el software y el hardware Dspace correctamente y que las licencias requeridas estén disponibles y activas.
- Abrir *ControlDesk* (**Programs – dSPACE Tools – ControlDesk**) desde el menú Inicio como se muestra en la Fig. 2.1

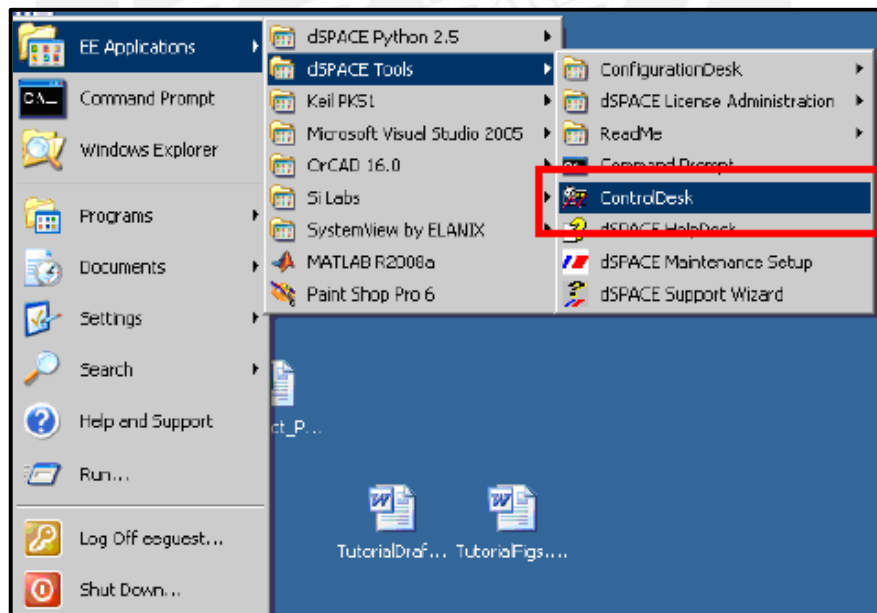


Fig. 2.1. Ubicación del icono de ControlDesk en el menú Inicio

- Ir a la pestaña Platform en el menú de ControlDesk, como lo muestra en al Fig. 2.2:

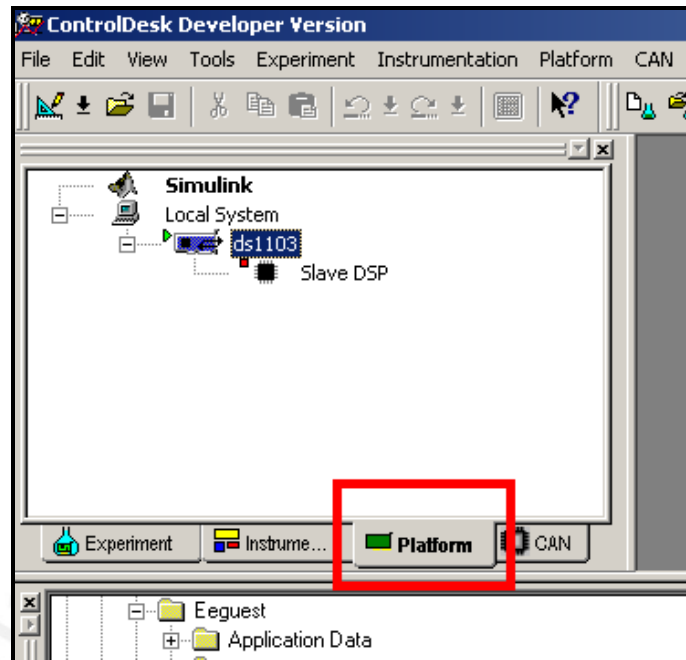


Fig. 2.2: Ubicación de la pestaña Platform en la ventana de ControlDesk.

- Abrir MATLAB, como se observa en la Fig. 2.3:

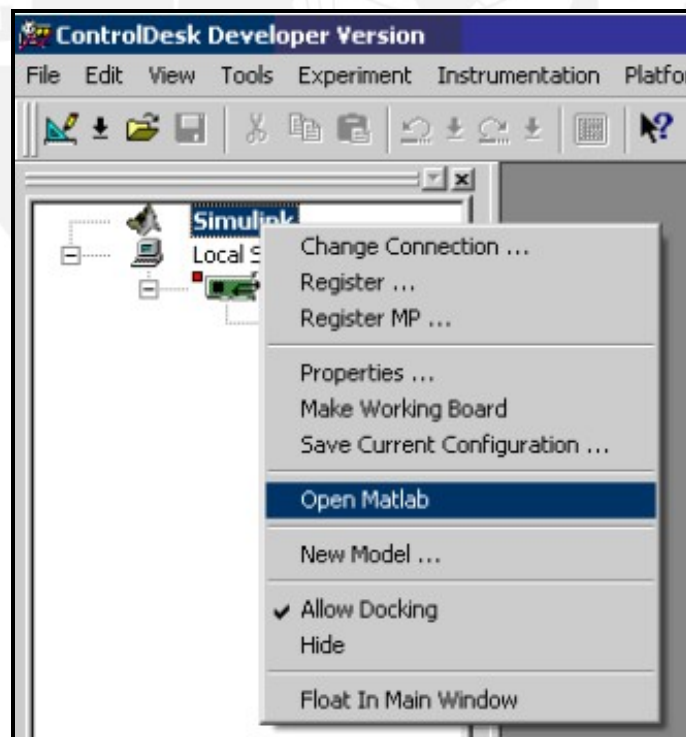


Fig. 2.3: Enlace para abrir MATLAB desde ControlDesk.

- Abrir SIMULINK, escribiendo "simulink" en la ventana de comandos de MATLAB, como lo muestra la Fig. 2.4:

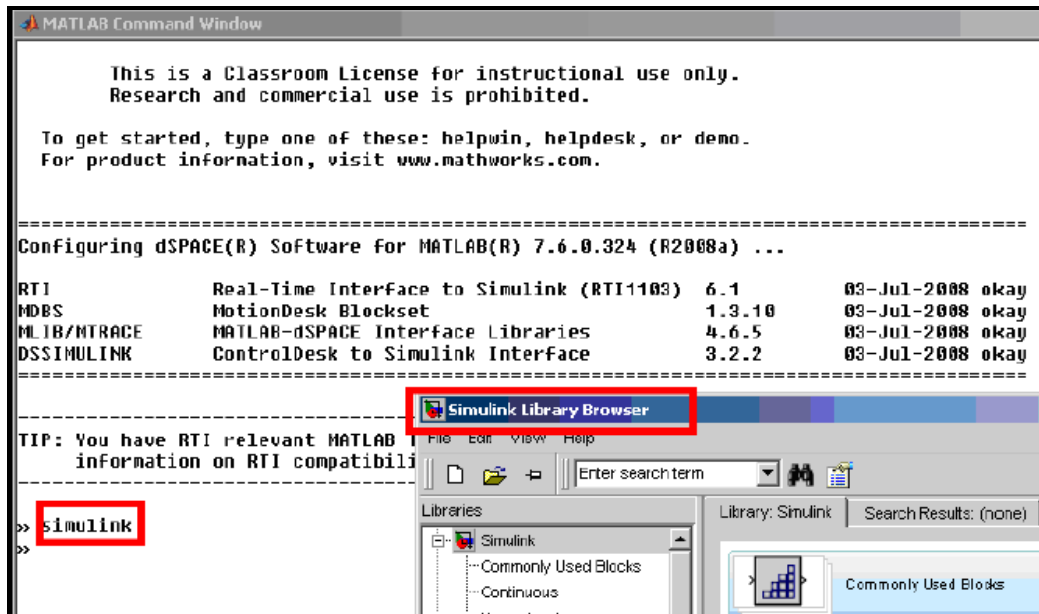


Fig. 2.4: Ingreso a simulink desde la ventana de comando de MATLAB.

- Fig. 2.5: En la ventana Simulink Library Browser, hacer clic en los signos +/- para examinar la librería RTI1102

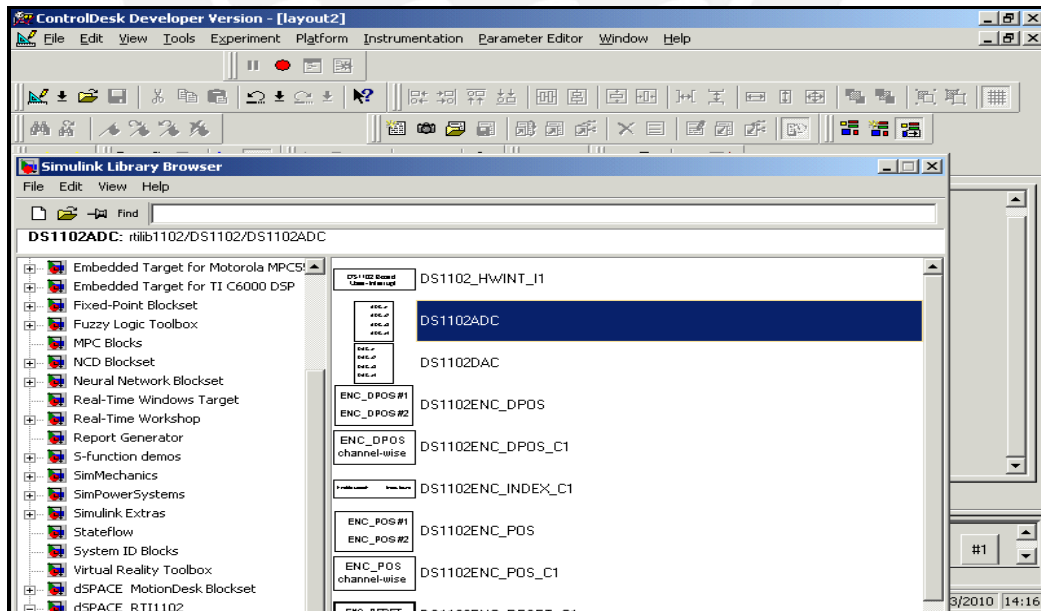


Fig. 2.5: Bloques de la librería de la DSpace1102

- Construir el diagrama de bloques del sistema de control para que pueda interactuar con el proceso a controlar, es decir agregando los bloques de la librería RTI1102.
- Una vez obtenido el modelo, se debe cambiar algunos parámetros en la simulación. Para hacer esto, hacer clic de la barra de menú *Simulation* y abrir la ventana de *Simulation Parameters* como se observa en la Fig. 2.6:

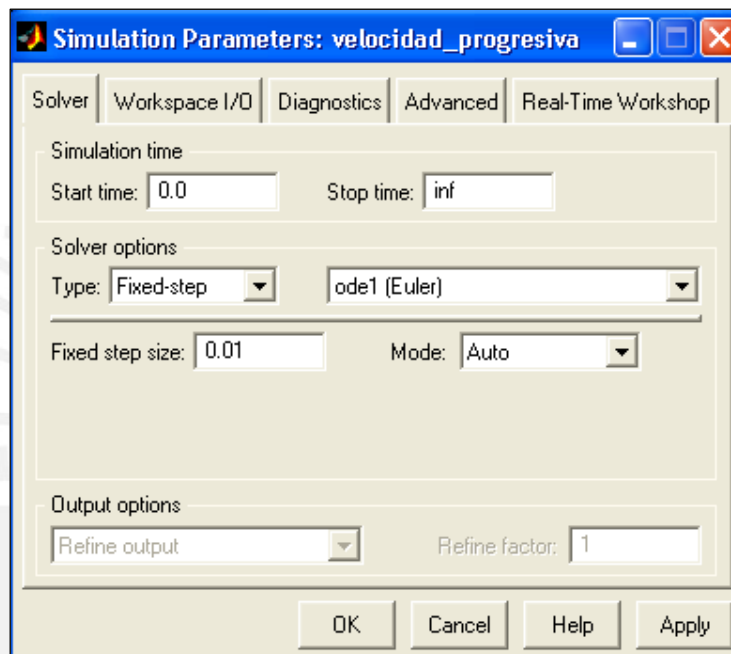


Fig. 2.6: Panel de configuración de parámetros

Existen algunas consideraciones importantes a tomar en cuenta y que se describen a continuación. En la pestaña Solver, el “Start time” ponerlo en 0 (necesario para aplicaciones en tiempo real). El “Stop Time” depende como se requiera que se ejecute la aplicación. Si se desea que se ejecute en un lazo sin fin se debe poner “inf” o si se desea que solo sea por 20 segundos deberá poner ese tiempo en este campo.

Luego, en “Type” seleccionar la opción *Fixed-step*, y en el campo de la derecha se recomienda un *solver* como “Euler” o quizás “ode5” (mientras más complejo sea el solver que se seleccione, más tiempo demorará el programa en ejecutarse).

Finalmente definir un tiempo de muestreo para la aplicación. Si el algoritmo demanda un mayor tiempo, que el del periodo de muestreo se presentará lo que usualmente se denomina un *overrun* y se generará un error que no permitirá descargar la aplicación a la tarjeta de control y la única solución es incrementar este periodo de muestreo. Después de estos cambios ubicarse en la pestaña “Advanced” y se recomienda deshabilitar la opción de reducción de bloques, tal como aparece en la Fig. 2.7:

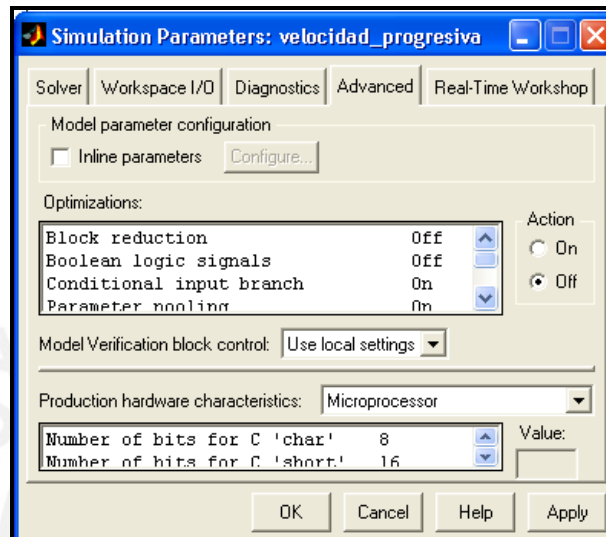


Fig. 2.7: Panel de configuración de parámetros, pestaña ADVANCE

Además en la pestaña de Real Time Workshop verificar que se tenga la siguiente configuración:

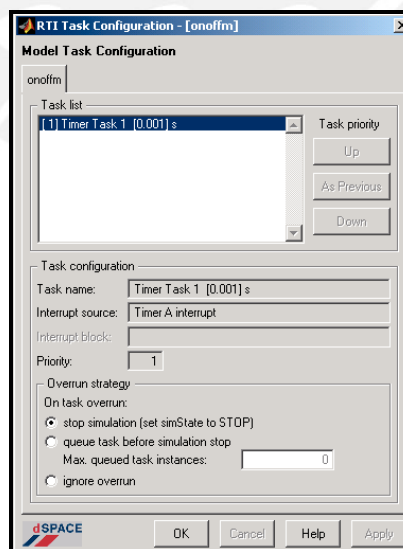


Fig 2.8: Ventana Configuración del modelo de tarea

Como se mencionó, una situación de **overrun** ocurre si una aplicación es enviada a iniciarse, pero la ejecución previa aun no está terminada. Por defecto la opción esta puesta en “*stop simulation*” y debe mantenerse así, de caso contrario si se ignora no se sabrá si la aplicación está corriendo en tiempo real o no.

Después de configurar el cuadro de dialogo anterior hacer clic en Ok para iniciar el proceso “Build”. Se genera un código C que es compilado por el Texas Compiler para producir un archivo ejecutable con extensión \*.obj que se descarga a la tarjeta DS1102 y el algoritmo empieza a correr. Para detener o reanudar la aplicación se usa los botones del menú de la interfaz de ControlDesk.

Para crear simulaciones en ControlDesk y observar el comportamiento de la aplicación en tiempo real y cambiar parámetros específicos, se requiere de tres archivos:

- El archivo con extensión TRC provee información de las variables disponibles y como están agrupadas.
- EL archivo MAP, que es siempre creado al final de la compilación, asigna nombres simbólicos hacia direcciones físicas.
- El archivo descriptivo del sistema SDF especifica que ejecutable será descargado hacia el procesador. Cuando se carga el archivo SDF, el ejecutable y todos los archivos relacionados al experimento son cargados automáticamente.

Real Time Interface (RTI) crea los archivos SDF, TRC y MAP cuando se genera y compila el código.

## ELEMENTOS BASICOS DE CONTROLDESK

Ahora haremos referencia a ControlDesk, una herramienta para poder configurar algún hardware dSPACE, específicamente la tarjeta de control DS1102, y además una tarea más elaborada la de configurar la plataforma para ejecutar aplicaciones en tiempo real desde Simulink.

Para poder implementar experimentos donde se pueda visualizar, adquirir, y controlar parámetros es necesario conocer un poco más el entorno de ControlDesk. Entonces ejecutamos el acceso directo desde el escritorio:



Otra manera es siguiendo la ruta: *Inicio/ Programas/dSPACE Tools*, a partir de la cual aparecerá la Fig. 2.9:

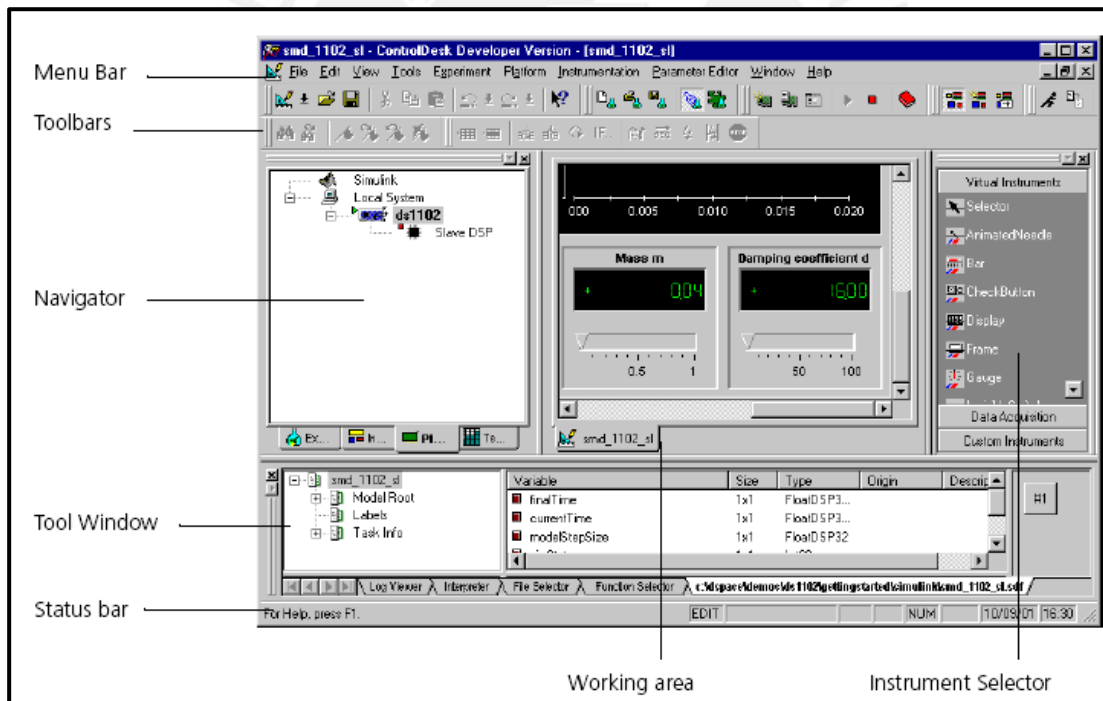


Fig. 2.9: Ventana completa del DSpace1102, se aprecian lo paneles internos.

La ventana Navigator tiene diferentes vistas para estructurar funciones como: administrar la plataforma, manejar archivos y construcción de paneles virtuales\_



- **Experiment Navigator:** Muestra todos los archivos (por ejemplo, paneles de instrumentación, configuración de parámetros, y los datos adquiridos) pertenecientes al experimento actual y provee funciones para manejar el experimento y sus componentes.
- **Instrumentation Navigator:** Permite construir paneles de instrumentación para controlar y monitorear las variables del modelo.
- **Platform Navigator:** Muestra todas las plataformas instaladas en el sistema (por ejemplo Matlab y DS1102) y provee funciones para registrar otras. Además se puede cargar, ejecutar y detener las aplicaciones.

En la región inferior se encuentra la ventana denominada “*Tool Window*”, la cual contiene también diferentes vistas. Por ejemplo la vista “*Log Viewer*” muestra mensajes de advertencias y errores. La vista “*File Selector*” es similar al explorador de Windows ya que permite buscar los archivos del sistema de la computadora, seleccionar y descargar las aplicaciones arrastrando dicho archivo con el mouse hacia el icono de la plataforma DS1102 que se encuentra en el “*Platform Navigator*”, como lo muestra la Fig. 2.10:

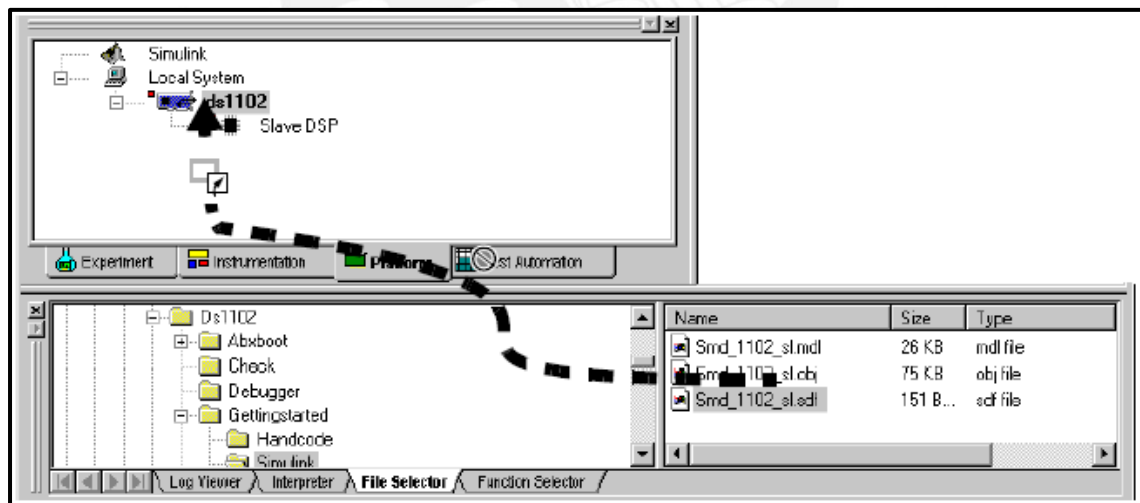


Fig. 2.10: Nos muestra como arrastrar los archivos a la ventana de Platform.

Otra área importante es el área general de trabajo (working área en la Fig.2.11) donde se crea y muestran los la interfaz gráfica de control, que permite monitorear, controlar y grabar valores utilizando los comandos necesarios.

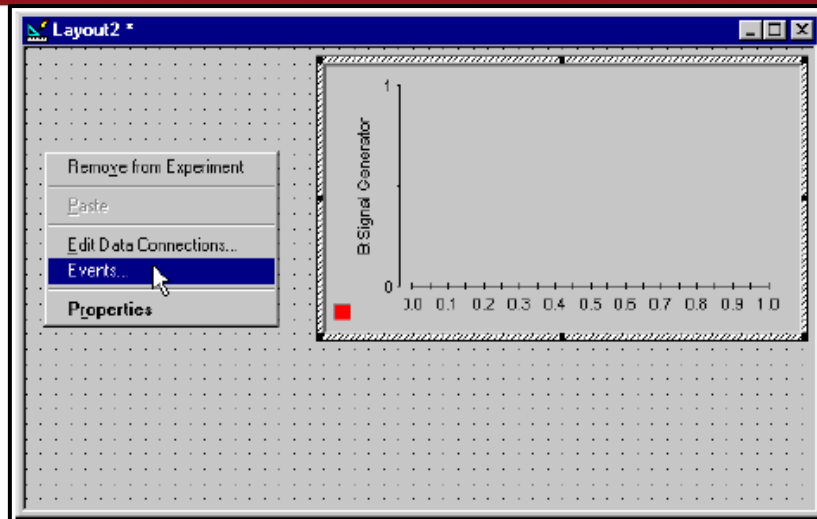


Fig. 2.11: Ventana de WORKING AREA

## CREACIÓN DE UNA INTERFAZ GRAFICA DE USUARIO – GUI

La interfaz gráfica permite ver en pantalla varios aspectos de operación cuando el sistema de control se está ejecutando, hacer cambios en parámetros mientras el sistema esté operando (por ejemplo las ganancias del controlador o los puntos de referencia), observar y guardar la información en ploters y reportes respectivamente.

1. Después de haber creado un diagrama en simulink con los bloques del RTI se crea un esquema o "layout" siguiendo la ruta *File > New > Layout* y también en *View > Controlbars > Instrument Selector*. Esta última acción mostrará las librerías que podrán ser usadas para crear la interfaz gráfica de usuario, que están agrupadas en dos tipos de instrumentos: Los instrumentos virtuales y los instrumentos de adquisición de datos, tal como se muestra en la Fig. 2.12:

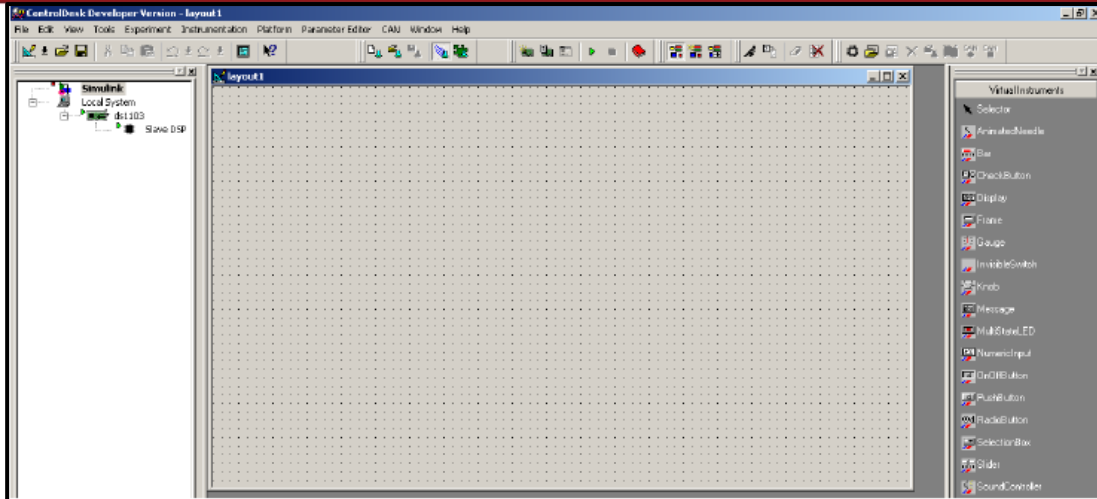


Fig. 2.12: Ventana completa del Layout y los instrumentos virtuales.

2. Seleccionar el modo editar en la barra de menú de la ventana del Layout.

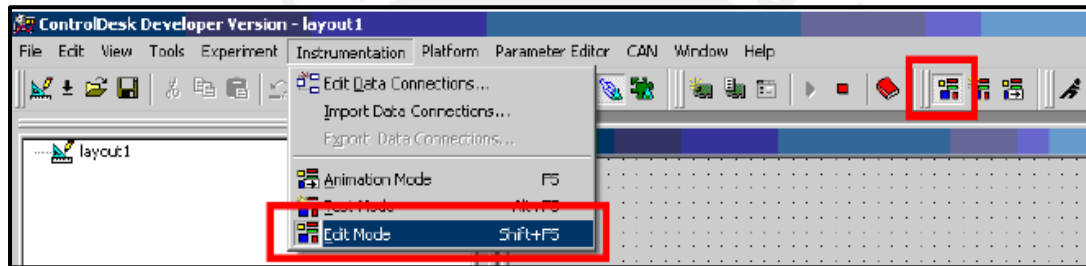


Fig. 2.13: Ubicar modo EDIT MODE en la ventana del LAYOUT

3. Ahora con las ventanas de “Virtual Instruments” y “Data Acquisition” se van agregando cada uno de los bloques deseados al área de trabajo en el layout. Se da clic en el bloque deseado y luego se grafica el bloque en el campo del layout como se muestra en la Fig. 2.14:

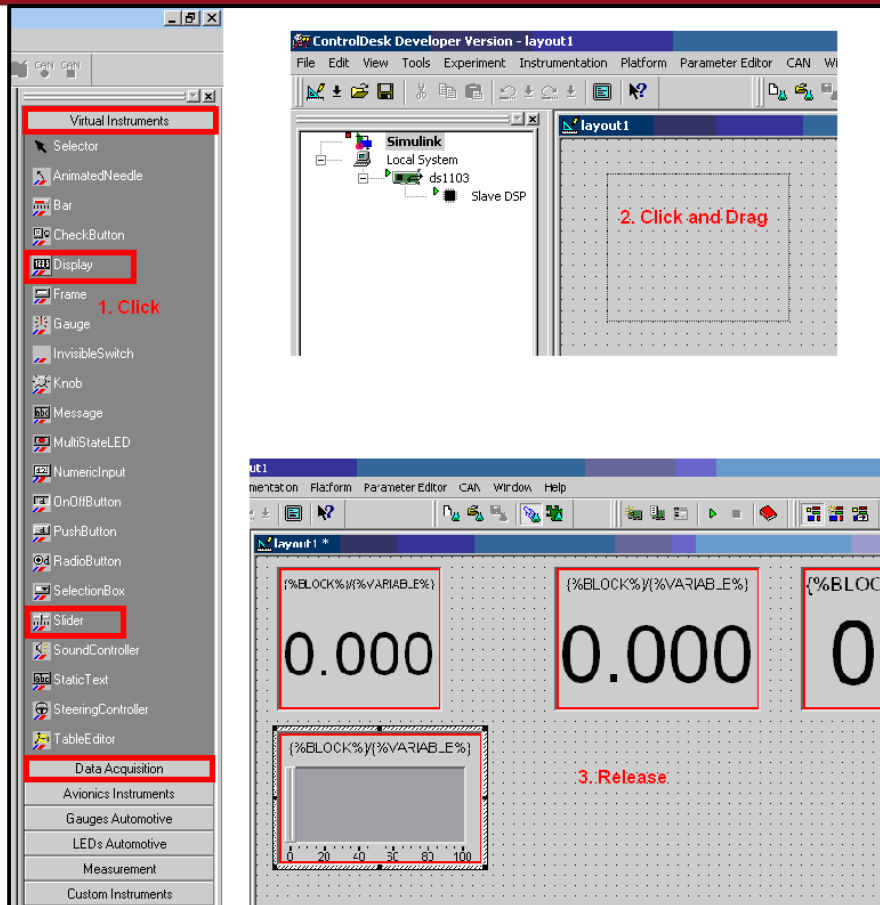


Fig.2.14: Agregar un bloque del menu de instrumento virtual.

4. Verificar que se encuentre en la parte del inferior, en la ventana "Tool Window" el archivo de extensión *.sdf* creado una vez realizada la construcción del modelo. Pues es de suma importancia para poder realizar el monitoreo, control y grabado de las variables deseadas, ya que si no está creado correctamente no se podrán mostrar esas variables como se muestra en la Fig. 2.15:

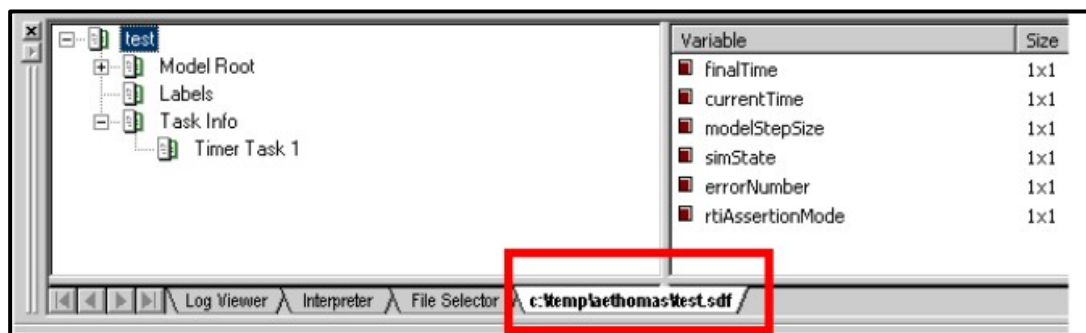


Fig. 2.15: Ubicacion del archivo con extensión *.sdf* en la venta "Tool Window".

- Una vez encontrada la pestaña con las variables en el archivo .sdf se procede a asignar a cada bloque la variable deseada a monitorear, controlar o grabar. Para esto, se tiene que elegir la variable que se asignara al bloque y se arrastra hacia él, como lo muestra la Fig. 2.16:

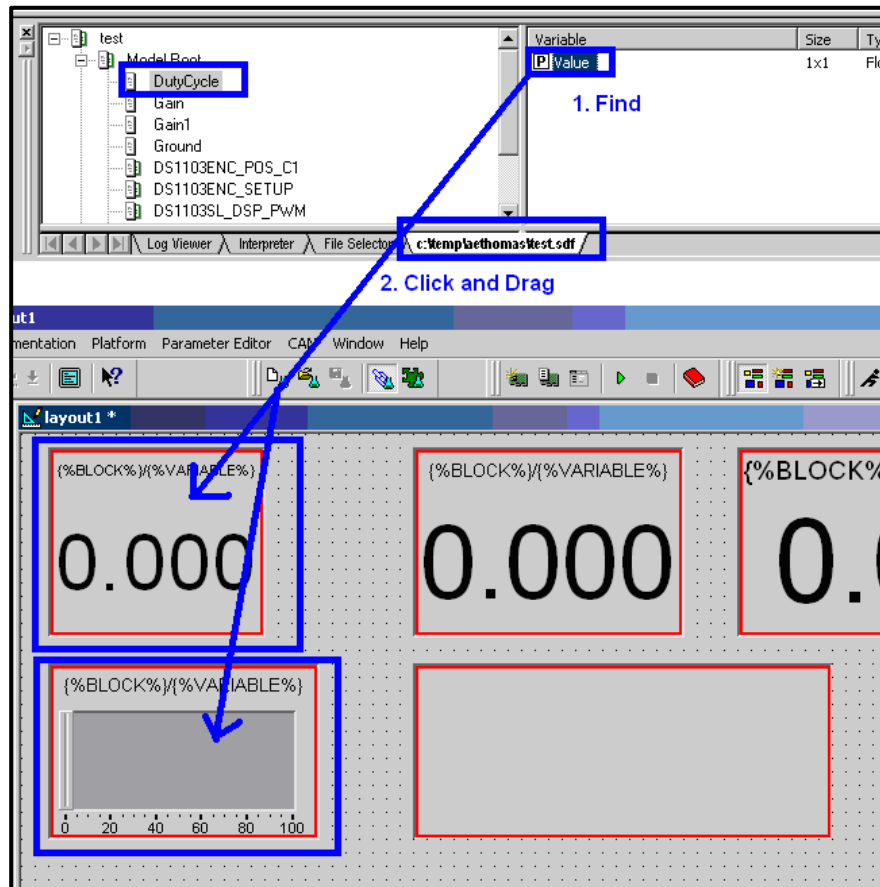


Fig. 2.16: Asignar las variables a los bloques.

- Una vez asignadas todas las variables deseadas se puede además configurar cada bloque por tamaño, número de fuente, color del fondo del bloque, color de fondo de la interfaz, cantidad de decimales deseados, se selecciona los valores máximos y mínimos, etc. Una vez terminado todo esto se procede a realizar la simulación en tiempo real de la interfaz dando clic en el botón "Animation Mode" de la barra de menú como se señala en la Fig. 2.17:

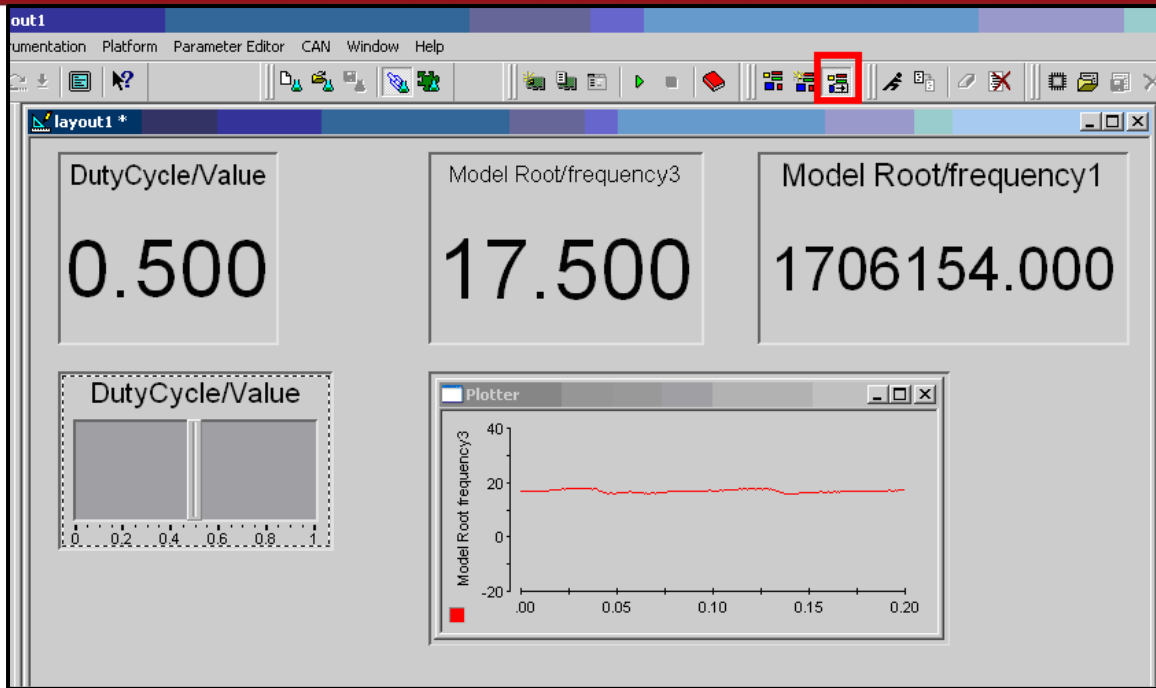
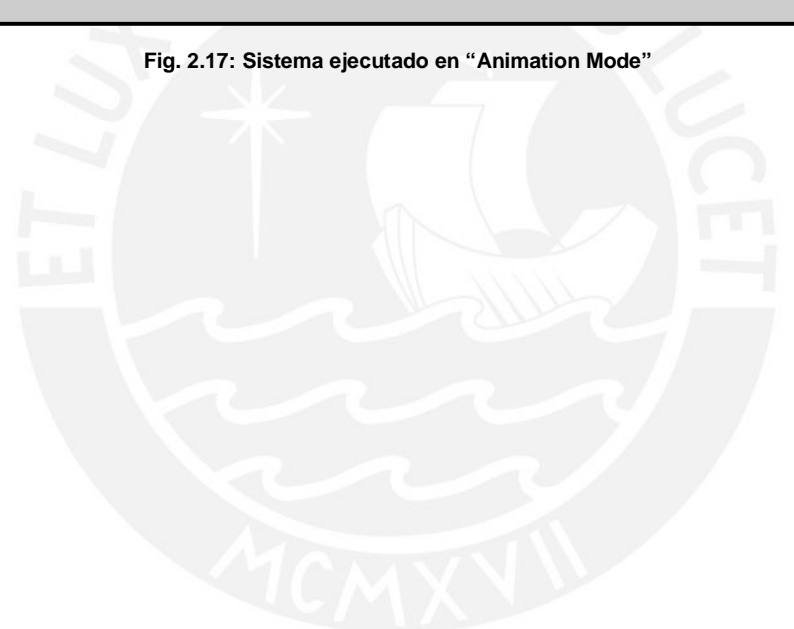


Fig. 2.17: Sistema ejecutado en "Animation Mode"



## CODIGO DEL SISTEMA EN MATLAB

```
J=10.656e-2;
B=4.21e-5;
K=48.2478e-3;
R=4.9125;
num=K;
den=[(R*J) ((K*K)+(R*B)) 0];
```

*% se halla la función de transferencia del lazo interno*

```
TF1=tf(num/0.5235,den/0.5235)
```

**Transfer function:**

$$\frac{0.09216}{s^2 + 0.004842 s}$$

*% se halla el diagrama del Bode del sistema para verificar el estado de nuestro lazo interno mediante las siguientes líneas*

```
figure
bode(num,den)
margin(num,den)
t=0:0.01:3;
```

*% Luego se agrega una ganancia para incrementar el ancho de banda del sistema y obtener un tiempo de respuesta menor.*

```
num=2000*K;
bode(num,den)
margin(num,den)
```

*% Para terminar de mejorar la respuesta transitoria del sistema agregamos un compensador en adelanto centrado en 20 rad/seg*

```
p=224;
z=1.79;
numa=[1 z];
dena=[1 p];
numb=conv(num,numa);
denb=conv(den,dena);
```



*% Se vuelven a hallar el diagrama de bode del lazo interno más el compensador para verificar el margen de fase y margen de ganancia.*

figure

margin(numb,denb)

*% Se multiplica por una ganancia de 25 para lograr obtener el mayor margen de fase en la frecuencia central de 20 rad/seg*

numb=conv(25\*num, numa);

margin(numb,denb)

*% Una vez verificado el margen de fase y de ganancia del sistema hallamos la respuesta una entrada escalón*

figure

[numc,denc]=cloop(5\*numb, denb, -1);

step(numc,denc,t)

*% Ahora agregamos la función de transferencia del lazo externo y verificamos su margen de fase y de ganancia.*

numle=[5.72];

denle=[1 0 0];

numleconv=conv(numle\*numc);

denleconv=conv(denle,denc);

margin(numleconv,denleconv)

*% Se observa un margen de fase negativo y margen de ganancia también negativo, lo que evidencia inestabilidad por lo tanto se debe agregar un compensador en adelanto para aumentar ese margen de fase y mejorar el margen de ganancia. El compensador esta centrado ahora en 10rad/seg y se muestra a continuación*

numale=[1 0.89]; *% Numerador del compensador en adelanto*

denale=[1 111]; *% Denominador del compensador en adelanto*

numble=conv(numleconv,numale);

denble=conv(denleconv,denale);

margin(numble,denble)

*% Agregar una ganancia la sistema para poder lograr el margen de fase mayor a 60° en la frecuencia de 10 rad/seg, multiplicamos por 200.*

margin(numble\*200,denble)

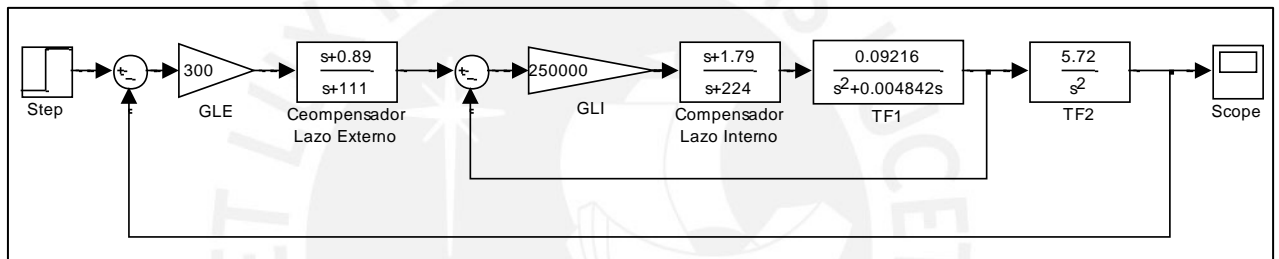
*% Se deberá volver a hallar el diagrama de bode del sistema y verificamos el margen de fase y ganancia, se logra lo adecuado ahora se procede a hallar la %respuesta del sistema una entrada escalón.*

```
[numcle,dencle]=cloop(numble*200, denble, -1);
step (numcle,dencle,t)
```

*% La respuesta es bastante aceptable pero para cumplir con los requerimientos se agrega una ganancia mayor para cumplirlos a cabalidad.*

```
[numcle,dencle]=cloop(numble*300, denble, -1);
step (numcle,dencle,t)
```

*% Finalmente se logra lo deseado, quedando todo el sistema de la siguiente manera:*



Lazo total con los bloques en simulink