

# SISTEMAS ELECTRÓNICOS DE CONTROL

## PRÁCTICA 6: Control en Velocidad de un Servomotor de CC

### 1. Objetivos.

- Controlar en velocidad un servomotor de CC mediante un regulador PID discreto utilizando el computador como elemento de control.
- Aprender el manejo de una tarjeta de adquisición de datos para interactuar con un proceso físico.
- Conocer el funcionamiento del software de comunicación con la tarjeta de adquisición desde Simulink: *Extended Real Time Toolbox*.
- Identificar un sistema de primer orden mediante un computador.

### 2. Introducción.

En esta práctica se pretende realizar el control en velocidad de un servomotor de CC utilizando un regulador PID discreto. Dicho control será implementado en un computador, por lo que será necesario utilizar una tarjeta de adquisición de datos para interactuar con el servomotor. Concretamente se utilizará la tarjeta de adquisición ACL-8112PG, de NuDAQ, diseñada para trabajar en computadores IBM PC o compatibles.

Generalmente la forma de llevar a cabo un control por computador es desarrollar una aplicación en la que se implementa el algoritmo de control. La tarjeta ACL-8112PG posee drivers, librerías en C y DLLs para poder ser programada en un gran número de plataformas. Sin embargo el desarrollo de una aplicación de control no es demasiado útil en la fase de diseño del controlador, en la que es necesaria realizar numerosas pruebas. Por esta razón se realizará el algoritmo de control directamente en Simulink. Para interactuar desde Simulink con la tarjeta de adquisición de datos se utilizará el software *Extended Real Time Toolbox*, de Humusoft.

Para diseñar el algoritmo de control del servomotor es necesario conocer el modelo matemático del sistema. Para obtener el modelo matemático correspondiente al servomotor se realizará una identificación del sistema. Utilizando el modelo matemático se determinará el controlador que satisfaga las especificaciones de diseño. Este controlador será probado en primer lugar mediante simulación, de forma que cuando se verifique el funcionamiento simulado, se implementará el control en el sistema real.

### 3. Control por Computador.

Actualmente el control digital ofrece unas mayores prestaciones que el control analógico. Las causas de este hecho son las ventajas que éste ofrece. Entre otras podemos citar las siguientes:

- **Potencia:** el control digital permite un procesamiento más sofisticado que el control analógico. La implementación de funciones no lineales complejas ya no suponen ningún problema pues se reducen a una línea de programa dentro del algoritmo de control.
- **Precisión:** aunque en el control digital aparecen retardos de cálculo, efectos de cuantización (errores en las conversiones A/D y D/A), su precisión es superior a los sistemas de control analógico
- **Versatilidad:** el cambio de un esquema de control a otro consiste únicamente en el cambio de una línea de programa.
- **Eficacia:** un único computador puede implementar múltiples tareas, desde varios lazos de control hasta sistemas monitor, comprobación de errores, presentación de resultados, etc.
- **Fiabilidad:** toda la teoría desarrollada para la fiabilidad y tolerancia a fallos en sistemas de microprocesadores son también aplicables en el control, haciendo de éste un método fiable y seguro.
- **Interfaz con el operador más amigable y potente** por medio de periféricos fáciles de utilizar (teclados, ratones...) y de presentaciones gráficas, ventanas, menús, ...

Como gran inconveniente se puede citar el retardo de cálculo inherente al procesamiento de la información (aunque con la potencia de los microprocesadores actuales se considera casi despreciable), los efectos de cuantización y un diseño más laborioso que en el control analógico.

El control digital se basa en la utilización de un elemento de naturaleza discreta para la realización del control. Dicho elemento se introduce en el lazo de control para hacer las veces de regulador. Este hecho supone introducir un elemento de naturaleza discreta en un entorno donde los sistemas a controlar son continuos y por tanto las señales que éstos generan también. Es por tanto necesario el uso de elementos adicionales que acomoden las señales de distinta naturaleza (continuas y discretas) para que estas puedan ser utilizadas por ambos elementos. Dichos elementos son los muestreadores, que permiten a un elemento discreto manipular señales continuas, y los retenedores, que realizan la operación inversa, es decir, a un elemento continuo manipular señales de naturaleza discreta.

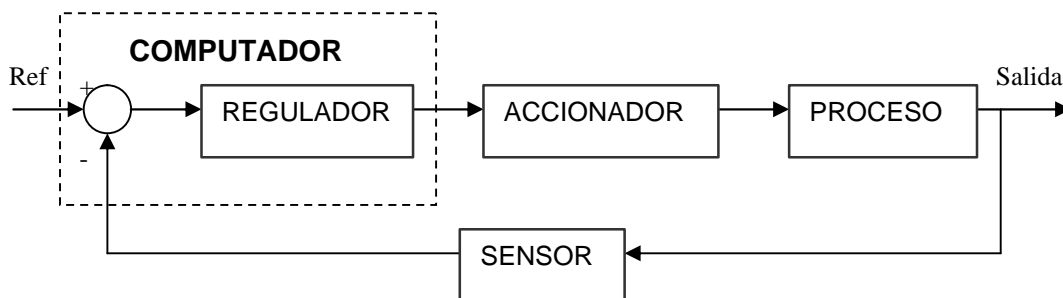
En principio, el computador, por si solo, es un elemento pasivo en el lazo de control. Para que éste pueda funcionar como regulador necesita un mecanismo que permita por una parte establecer una comunicación con el entorno, es decir, capturar y enviar señales, y por otro adecuar estas señales a la naturaleza del entorno con el que se está trabajando (continuo o discreto). Dicho mecanismo viene implementado por una tarjeta de adquisición de datos. Dicha tarjeta permite la comunicación del computador con el entorno (adquiriendo y enviando señales) a la vez que realiza las conversiones

necesarias mediante muestreadores o retenedores. El hecho de adquirir una señal continua por el computador (elemento de naturaleza discreta) se denomina conversión analógico – digital (abreviadamente conversión A/D), mientras que la operación inversa, el envío de una señal discreta a un sistema o medio continuo se denomina conversión digital – analógica (conversión D/A).

Además de la tarjeta de adquisición de datos, el computador necesita un entorno de programación que permita realizar el algoritmo de control. No olvidemos que un regulador discreto se puede representar mediante una ecuación en diferencias de forma que se puede convertir ésta en una línea de programa. Obviamente, dicho entorno de programación debe ofrecer los mecanismos adecuados (librerías) para la correcta comunicación entre el computador y la tarjeta de adquisición de datos.

#### 4. Descripción del Sistema Experimental.

Esquemáticamente, el diagrama de bloques general de un sistema de control por computador es el siguiente:



El funcionamiento del lazo de control anterior es el siguiente: el operador genera una señal de consigna o referencia que será utilizada por el regulador para que la salida del sistema coincida con dicha señal. El regulador a partir de dicha consigna, de la salida del sistema y con el algoritmo de control adecuado, genera una acción de control que pasa al accionador, que se encargará de adecuar dicha señal a las características de la entrada del proceso. Un sensor lee la salida del sistema y la traslada al controlador. Nótese que en los pasos anteriores se han obviado las operaciones de conversión A/D y D/A realizados por el sistema de adquisición de datos conectado al computador.

Tomando como referencia el esquema de bloques anterior, identifiquemos los componentes físicos correspondientes que se van a utilizar en esta práctica:

- a) Proceso: el proceso será el servomotor de corriente continua (la descripción del servomotor se proporciona en el siguiente apartado).
- b) Accionador: está incluido en el servomotor y es la electrónica necesaria para generar la señal que actúa sobre el motor. Nótese que este elemento, al estar incluido en el propio proceso, no nos debe preocupar en principio a la hora del diseño del regulador ya que la identificación del proceso engloba el comportamiento del accionador.

- c) Sensor: el sensor que se utilizará para leer la velocidad del servomotor será un tacogenerador. Este sensor proporciona un valor de tensión que es proporcional a la velocidad angular de giro del eje del motor. Este sensor también se encuentra englobado en el mismo proceso.
- d) Regulador: el regulador en este caso está formado por el computador (con su software asociado) y el sistema de adquisición de datos, que en este caso será una tarjeta de adquisición de datos de la casa NuDAQ, en concreto la ACL-8112PG cuyas prestaciones y características se comentarán más adelante.

Hasta aquí se ha descrito brevemente el sistema con el que se va a trabajar, sin embargo, es necesario realizar algunas matizaciones. Es necesario aclarar que se trata de un equipo de aplicación docente y como tal se han previsto algunos de los efectos no deseables que se presentan en entornos industriales como pueden ser el ruido electromagnético, las condiciones ambientales y climáticas, etc. Es por esto por lo que en aplicaciones industriales se suelen utilizar los denominados PC's industriales que están preparados para soportar estos efectos y por tanto están provistos de una eficaz protección contra ruidos e interferencias (mediante el correspondiente apantallamiento), polvo y otros elementos perjudiciales (mediante complejos sistemas de ventilación y filtrado) y temperaturas excesivamente bajas o elevadas (mediante la calidad de sus componentes). Además, en aplicaciones donde tanto la precisión como las prestaciones exigidas son elevadas se suele trabajar con sistemas distribuidos donde por ejemplo el sistema de adquisición de datos es independiente del computador (mucho más preciso y más caro) y la comunicación entre ambos y/u otros subsistemas se realiza mediante buses industriales especiales.

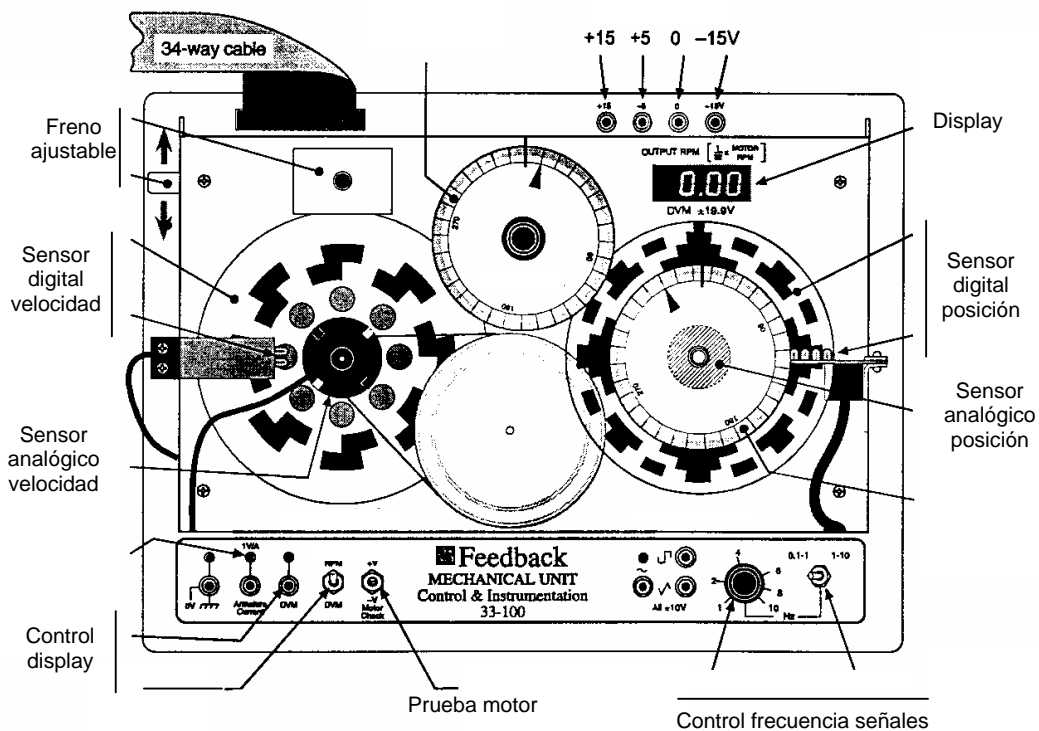
## 5. Servomotor de CC.

Se pretende realizar el control en velocidad por computador de un servomotor. Un servomotor es un motor eléctrico con capacidad de ser **controlado**, en **velocidad** y/o **posición**.

- Control en velocidad: posibilidad de hacer girar al motor a una velocidad determinada independientemente de la carga o fuerza que deba vencer.
- Control en posición: posibilidad de realizar desplazamientos entre dos posiciones determinadas con precisión.

### 5.1. Descripción del servomotor.

La figura siguiente muestra el aspecto del servomotor que emplearemos, así como sus principales componentes:



Los elementos principales del servomotor son:

- Motor de CC
- Reductor de velocidad
- Freno magnético ajustable

Sensores disponibles:

- Sensor analógico de posición (potenciómetro)
- Sensor digital de posición (encoder)
- Sensor analógico de velocidad (tacogenerador)
- Sensor digital de velocidad (tren de pulsos)

Señales de prueba utilizables:

- Senoidal
- Cuadrada
- Triangular

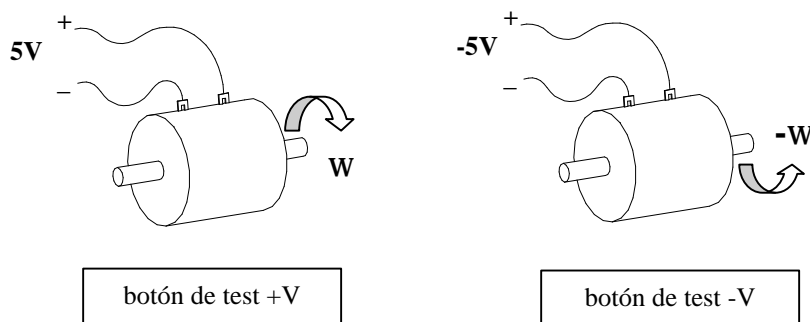
## 5.2. Fuente de alimentación.

Debe estar conectada al equipo continuamente, a través de las entradas +15, +5, 0 y –15V de la parte superior.

¡¡ **ATENCIÓN !!** El botón de encendido de la fuente de alimentación será el que utilizemos para apagar y encender el equipo completo.

Prueba: accionamiento del motor con el botón de test

Para comprobar que el motor está perfectamente conectado utilizaremos el botón de test para moverlo. Este botón (indicado como ‘**prueba motor**’ en la figura) introduce en el motor una tensión de aproximadamente  $\pm 5V$  según se pulse en una u otra dirección, de modo que hace girar el motor continuamente en un sentido u otro.



La velocidad que alcanza el motor en cada uno de los sentidos se puede leer en el display de la parte superior de la maqueta.

### 5.3. Tarjeta de expansión.

Mediante esta tarjeta de expansión están accesibles todas las entradas y salidas del servomotor.

### 5.4. Sensores a utilizar.

En la descripción del equipo se han mencionado los sensores disponibles:

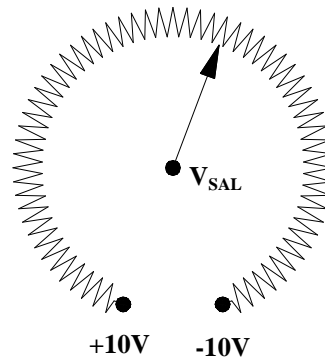
- **Sensor analógico de posición (potenciómetro)**
- **Sensor digital de posición (encoder)**
- **Sensor analógico de velocidad (tacogenerador)**
- **Sensor digital de velocidad (tren de pulsos)**

La curva de comportamiento de cada uno de estos sensores es la curva que relaciona los valores eléctricos (**tensiones** en algunos casos, **frecuencias** en otros) con los valores que se desean medir: **ángulos** o **velocidades de giro**.

A continuación se detallan en mayor profundidad los sensores analógicos del servomotor.

### Sensor analógico de posición (potenciómetro)

El sensor analógico de posición está basado en un **potenciómetro** (resistencia variable). El circuito representativo es el siguiente:



La figura representa una resistencia conectada a **+10V** por uno de sus extremos y a **-10V** por el otro. El cursor o aguja gira solidario con el eje del cual queremos conocer la posición. La tensión de salida  $V_{SAL}$  tomará valores entre **-10V** y **+10V** en función del ángulo girado por el eje.

La maqueta dispone de dos potenciómetros, uno para cada uno de los círculos graduados. El círculo graduado de la izquierda se puede girar manualmente. El círculo de la derecha, en cambio, gira solidario al motor.

La tensión de salida  $V_{SAL}$  correspondiente al potenciómetro de salida (potenciómetro derecho) se encuentra accesible a través del **pin 12** de la tarjeta de expansión del servomotor.

### Sensor analógico de velocidad (tacogenerador)

El sensor analógico de velocidad está basado en un **tacogenerador**. Al igual que en el caso del potenciómetro, el sensor nos ofrece una tensión proporcional a la magnitud que queremos medir.

El voltaje proporcionado por el tacogenerador es accesible desde la tarjeta de expansión del servomotor a través del **pin 35**.

## **6. Tarjeta de adquisición ACL – 8112PG.**

El control digital se basa en la utilización de un elemento de naturaleza discreta para la realización del control. Dicho elemento se introduce en el lazo de control para hacer las veces de regulador. Este hecho supone introducir un elemento de naturaleza discreta en un entorno donde los sistemas a controlar son continuos y por tanto las señales que éstos generan también. Es por tanto necesario el uso de dispositivos adicionales que

acomoden las señales de distinta naturaleza (continuas y discretas) para que estas puedan ser utilizadas por ambos elementos. Dichos dispositivos son los muestreadores, que permiten a un elemento discreto manipular señales continuas, y los retenedores (bloqueadores), que realizan la operación inversa, es decir, a un elemento continuo manipular señales de naturaleza discreta.

En principio, el computador, por si solo, es un elemento pasivo en el lazo de control. Para que éste pueda funcionar como regulador necesita un mecanismo que permita por una parte establecer una comunicación con el entorno, es decir, capturar y enviar señales, y por otro adecuar estas señales a la naturaleza del entorno con el que se está trabajando (continuo o discreto). Dicho mecanismo viene implementado por una tarjeta de adquisición de datos. Dicha tarjeta permite la comunicación del computador con el entorno (adquiriendo y enviando señales) a la vez que realiza las conversiones necesarias mediante muestreadores o retenedores. El hecho de adquirir una señal continua por el computador (elemento de naturaleza discreta) se denomina conversión analógico – digital (abreviadamente conversión A/D), mientras que la operación inversa, el envío de una señal discreta a un sistema o medio continuo se denomina conversión digital – analógica (conversión D/A). La tarjeta de adquisición de datos que emplearemos para la práctica es la ACL – 8112PG, de NuDAQ.

La tarjeta de adquisición de datos ACL – 8112PG es una tarjeta de adquisición de datos diseñada para trabajar con computadores IBM PC o compatibles. Esta tarjeta combina todas las funciones de la adquisición de datos, tales como las conversiones A/D, D/A, la entrada – salida de señales digitales (DIO), y cuenta, además, con temporizadores y contadores.

Utilizaremos esta tarjeta para leer la señal analógica correspondiente a la velocidad angular del servomotor: tensión generada por el tacogenerador ( $\pm 10V$ ).

Para leer señales analógicas la tarjeta dispone de 16 canales de entrada analógica. El rango de valores de tensión de entrada es de  $-10V$  a  $+10V$ .

También usaremos la tarjeta de adquisición para generar la tensión de entrada al servomotor. Esta tarjeta posee 2 canales de salida analógica. El rango de valores de tensión de salida que puede tomar cada canal es de  $0V$  a  $+10V$ .

### **6.1. Conexión de la tarjeta ACL – 8112PG con el servomotor.**

Para realizar la conexión con el servomotor se utiliza una tarjeta adicional conectada mediante un cable y un conector de 37 pines a la ACL - 8112PG. Dicha tarjeta adicional consiste en una regleta donde se puede acceder fácilmente a cada uno de los canales, a la vez que aporta la electrónica necesaria para acondicionar las señales. La tarjeta en cuestión es la ACL – 9138 y la disposición de los distintos canales y señales de la ACL - 8112PG se muestra en la siguiente tabla:



PIN	DESCRIPCIÓN
1 a 8	Canal 0 a 7 de entrada analógica
9 – 10	Toma de tierra de los canales de entrada analógica
11	Referencia de tensión externa 2
13	Salida de +12 V
14	Toma de tierra de los canales de entrada analógica
15	Toma de tierra de los canales de salida digital
16	Señal de salida del contador 0
17	Señal de disparo externo
18	No conectada
19	+5V
20 a 27	Canal 8 a 15 de entrada analógica
28 – 29	Toma de tierra de los canales de entrada analógica
30	Canal 0 de salida analógica
31	Referencia de tensión externa 1
32	Canal 1 de salida analógica
33	Entrada de reloj para 8254
34	Entrada para 8254
35 – 36	No conectada
37	Entrada de señal de reloj externo
38	Toma de tierra de los canales de salida analógica

Para esta práctica los pines que nos van a interesar son los correspondientes a las entradas analógicas, 1 a 8 (y los correspondientes a tierra, 9 a 10) para leer las señales provenientes del servomotor, y los de salida analógica 30 y 32 (y su señal de tierra correspondiente, 38) para generar la acción de control. Más adelante veremos cómo conectar cada una de las señales del sistema a la tarjeta ACL – 9138.

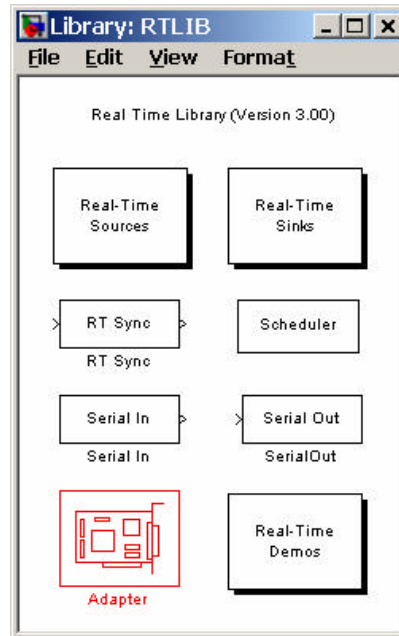
## 7. Software de comunicación con la tarjeta ACL – 8112PG: *Extended Real Time Toolbox*.

### 7.1. Descripción del software.

Para comunicarse con la tarjeta de adquisición ACL – 8112PG se va a utilizar el software *Extended Real Time Toolbox*, de Humusoft. Este software permite trabajar con la tarjeta de adquisición de datos directamente desde Simulink, ofreciendo así una gran potencia y facilidad en la etapa de desarrollo de un sistema de control.

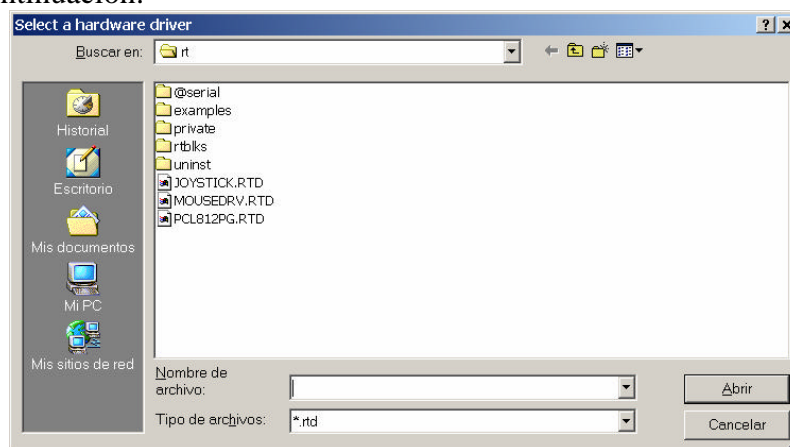
Este toolbox posee una librería de bloques para Simulink, denominada *Real Time Toolbox Simulink Block Library* (RTLIB), que permite la interacción con la tarjeta de adquisición desde Simulink. La utilización de la RTLIB en un esquema Simulink pasa por dos fases: primero la inclusión y configuración del driver del hardware de nuestra tarjeta, y segundo, la correcta utilización de los diferentes bloques de entrada/salida en tiempo real.

Para acceder a la librería RTLIB ejecutaremos en el entorno de Matlab el comando `rtlib`. Al ejecutar dicho comando aparecerá la siguiente ventana de Simulink:



A partir de aquí, tras ejecutar Simulink, abriremos un nuevo esquema y podremos comenzar a trabajar con los bloques de la RTLIB.

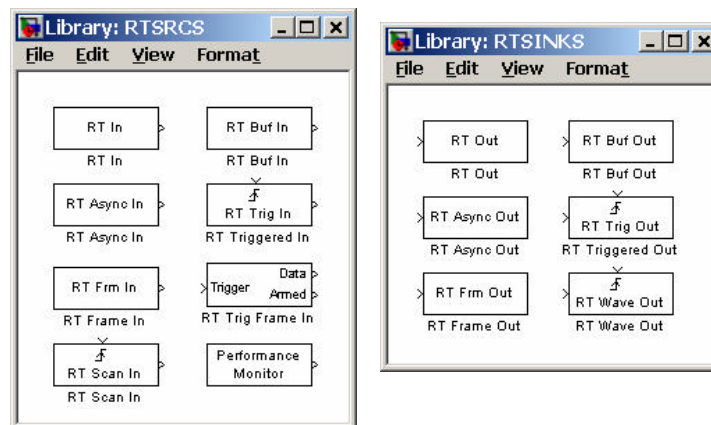
El primer paso es incluir en nuestro esquema un bloque adaptador. El bloque *Adapter* es un bloque que representa la tarjeta de adquisición de datos y se utiliza para disponer dentro del esquema Simulink de los canales de la tarjeta de adquisición de datos. Una vez incluido dicho bloque en el esquema Simulink, lo primero que hay que hacer es configurarlo para que utilice el driver de la tarjeta que poseemos. Para ello haremos doble clic con el ratón sobre el bloque *Adapter* y se abrirá una ventana como la que se muestra a continuación:



En nuestro caso, el driver que debemos cargar es el correspondiente al fichero `PCL812PG.RTD`, para ello lo seleccionamos y pulsamos 'Aceptar'. A partir de este momento ya tenemos cargado en nuestro esquema Simulink los drivers de la tarjeta

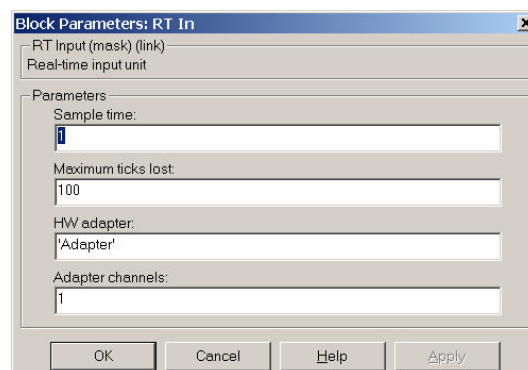
ACL – 8112PG con las opciones por defecto. En el caso de querer cambiar dichas opciones haremos de nuevo doble clic sobre el bloque *Adapter* y se abrirá una ventana de selección de parámetros. En principio y a menos que se indique lo contrario, se trabajará con las opciones del driver por defecto, por lo que no será necesario realizar este último paso.

Ahora ya podemos empezar a utilizar los bloques de entrada/salida de la librería. Se dispone de dos tipos de bloques: *sources* y *sinks*. Para acceder a los mismos haremos doble clic sobre los bloques *Real – Time Sources* o *Real – Time Sinks* respectivamente. De esta forma aparecerán las siguientes ventanas:



Como se puede apreciar aparecen multitud de bloques tanto en una como en otra ventana. En este documento sólo se van a analizar aquellos que se van a utilizar en la práctica: los bloques '*RT In*' y '*RT Out*'.

El bloque '*RT In*' está diseñado para la adquisición de datos en tiempo real, el procesamiento de señales, y las aplicaciones de control donde los datos han de ser procesados tan rápido como sea posible. Para configurar este bloque haremos doble clic sobre el mismo, apareciendo así una ventana como la que se muestra a continuación:



Veamos los diferentes parámetros de este bloque:

- *Sample Time*: Periodo de muestreo al cual se debe realizar la adquisición de los datos. Se permite especificar un vector en el caso de que se quiera acceder a más de un canal con el mismo bloque.

- *Maximum ticks lost*: Máximo número de ‘tics’ (muestreos) que se pueden perder antes de mostrar un mensaje de error. Esto es un mecanismo de “seguridad” para asegurar que la adquisición se ha realizado de forma correcta (no ha habido excesivas pérdidas de datos). Hay que tener en cuenta que este sistema en tiempo real se ejecuta sobre Matlab y Simulink, y éstos a su vez sobre Windows, por lo que cabe la posibilidad de que en un determinado instante no se pueda realizar la adquisición con la frecuencia correcta. Esto, claro está, depende tanto de la potencia del microprocesador como de la cantidad de recursos de la máquina que están ocupados.
- *HW adapter*: Nombre del bloque *Adapter* incluido en el esquema Simulink. Esto es así pues cabe la posibilidad de tener en un mismo esquema más de un adaptador para gestionar dos sistemas de adquisición distintos (por ejemplo uno de entradas analógicas y otro de salidas analógicas independientes).
- *Adapter channels*: Canal analógico de entrada al que se quiere acceder. En este caso, puesto que son canales de entrada analógica, tendremos 8 (ya que la ACL – 8112PG tiene 8) numerados del 1 al 8, correspondiendo a los canales de entrada analógica 0 a 7 de la tarjeta de adquisición. Es posible especificar un vector de canales por si se quiere acceder a más de uno de ellos desde el mismo bloque.

Los parámetros a fijar en el bloque ‘*RT Out*’ son idénticos a los anteriores, con la salvedad de que en este caso se trata de un bloque de salida y se emplea para generar las señales con el mínimo retardo posible.

- *Sample Time* especifica el periodo al cual se actualiza el canal de salida correspondiente en la ACL – 8112PG.
- *Maximum ticks lost* indica el máximo número de veces que pueden no actualizarse los canales de salida, debido a la carga del sistema, sin que se muestre un mensaje de error.
- *HW adapter* es idéntico al caso anterior.
- *Adapter channels* especifica el canal de salida analógica al que se accede. En este caso se dispone de 2 canales, numerados del 1 al 2, correspondiendo a los canales de salida analógica 0 a 1 de la tarjeta ACL – 8112PG. Al igual que en el caso anterior es posible especificar un vector de canales.

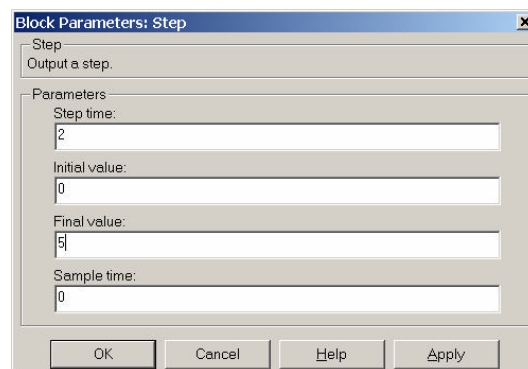
## 7.2. Metodología de trabajo con *Extended Real – Time Toolbox*.

El rango de tensiones de salida y de entrada soportables por la tarjeta (en nuestro caso 0V a 10V y –10V a 10V, respectivamente) se transforma en un rango de –1 a 1 en los bloques del RTT. Esto implica realizar una transformación para obtener la tensión deseada. Dicha transformación se realizará en Simulink mediante el bloque de función *Fcn*. Las funciones que se deben implementar son:

- Antes del bloque de salida:  $(u(1)/5)-1$
- Después del bloque de entrada:  $(u(1)*10)$

Además, por problemas, errores o imperfecciones del driver de la tarjeta ACL – 8112PG, el modo en que el sistema inicializa la tarjeta cuando se inicia una simulación

(en este caso la ejecución en tiempo real del esquema Simulink) es poniendo las salidas analógicas de la misma a una tensión de 5 voltios en vez de a los 0 voltios esperados. Por esto, cuando ejecutemos el esquema Simulink (más adelante veremos como hacerlo), es necesario que la señal de referencia empiece a actuar unos dos segundos después de iniciar el programa, permitiendo de este modo la inicialización de la señal de referencia. Para ello, los bloques que generan esta señal de referencia (por ejemplo un escalón que es lo más habitual) deben comenzar con un valor inicial de 0 y a los dos segundos generar el valor de referencia deseado. Por ejemplo, para una referencia en escalón de 5 voltios, la configuración de dicho bloque será:



De esta forma estamos indicando que el valor inicial será 0 y que a los dos segundos el valor final será el de la referencia deseada, es decir, 5. Notemos también que el último parámetro *Sample Time* puede permanecer a 0, ya que el periodo de muestreo al que se va a realizar la conversión se especifica en el bloque *RT Out*.

### Esquema básico de generación y lectura de señales

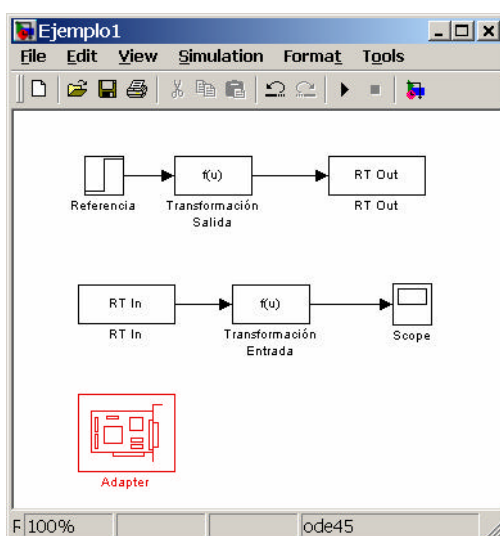
Se va a construir un primer esquema para afianzar lo expuesto hasta ahora. Este esquema introducirá un escalón de tensión en el servomotor durante 8 segundos y leerá la velocidad a la que gira utilizando el sensor analógico de velocidad. Para ello es necesario realizar las siguientes conexiones de la tarjeta con el servomotor:

Pin ACL – 9138	Pin tarjeta de expansión del servomotor
1 (canal 0 de entrada analógica)	35 (señal del tacogenerador)
10 (GND de entrada analógica)	<i>Señal de tierra del propio servomotor</i>
30 (canal 0 de salida analógica)	33 (tensión de entrada positiva)
GND (GND de salida analógica)	34 (tensión 0)

- La salida del tacogenerador (pin 35 de la tarjeta de expansión del servomotor) proporciona una tensión entre 0 y 10 voltios proporcional a la velocidad del motor. La masa de dicha señal se tomará del propio servomotor.
- El pin 33 de la tarjeta de expansión del servomotor corresponde a la entrada al servomotor y es donde se aplicará por tanto la acción de control. La masa de dicha señal se tomará del pin 34 de la tarjeta de expansión.

**NOTA:** Todas las conexiones se deben realizar con la fuente de alimentación desconectada y se deben repasar las mismas antes de conectar. Además, siempre que no se esté ejecutando un programa sobre el motor, la fuente de alimentación debe permanecer también desconectada.

Con las conexiones realizadas se adquirirá la velocidad del servomotor por el canal 0 de entrada analógica y se generará una tensión de 3 voltios por el canal 0 de salida analógica (recordemos que ambos canales se denominan 1 en los bloques de RTT). El periodo de muestreo deberá ser de 0.01 seg. tanto para la entrada como para la salida de datos. Además, se visualizará la señal adquirida utilizando un bloque *Scope*. El esquema tendrá la forma siguiente:

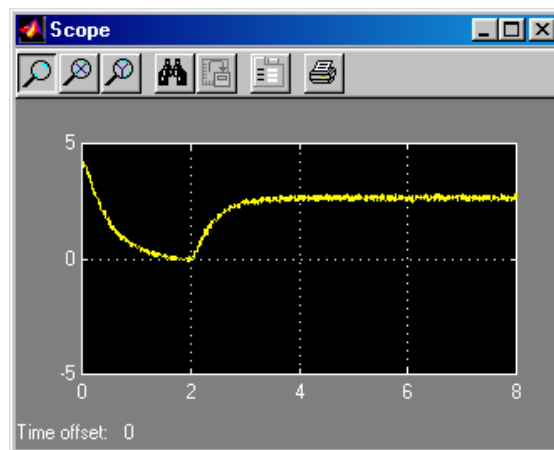


En este esquema, la señal de referencia se genera tal y como se ha explicado previamente, los bloques de ganancia de conversión implementan las funciones de transformación ya vistas. Ahora lo único que resta es ejecutar dicha simulación. Los parámetros de simulación (menú *Simulation* → *Parameters*) idóneos son los siguientes:

- Solver options – Type: Fixed – Step (discrete (no continuous states))
- Fixed step size: 0.01 (periodo de muestreo al que realizamos la adquisición)

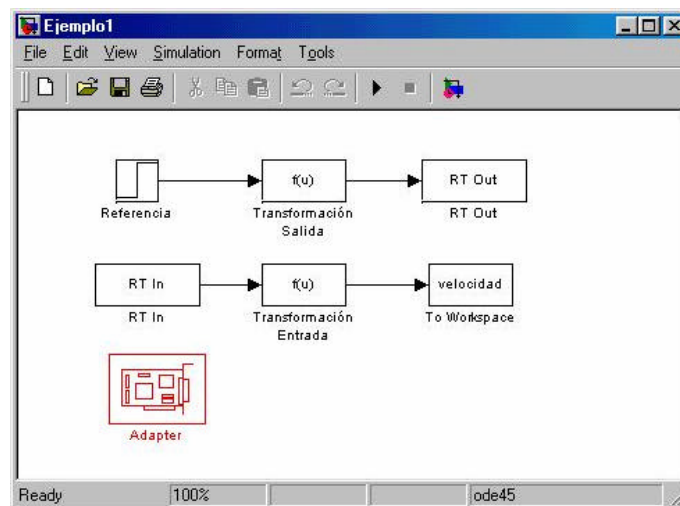
Por supuesto, en los bloques *RT Out* y *RT In* debemos especificar el canal de salida y entrada (en ambos caso el canal 0 que corresponde al *Adapter Channel 1*) y el periodo de muestreo (0.01 seg.).

Al ejecutar la simulación teniendo el motor conectado a la tarjeta de adquisición de datos, lo que se está haciendo es inyectar al motor una entrada de 3 voltios y recogiendo la salida en velocidad. Así, la señal que debe aparecer en el bloque *Scope* será parecida a la que se muestra en la figura siguiente:



Como vemos, debido al error de los drivers, inicialmente el valor de la salida es la correspondiente a la de un impulso de entrada de 5 voltios. Esta respuesta impulsional decrece exponencialmente hasta llegar a cero voltios. El motivo de esperar dos segundos a generar el escalón de 3 voltios es esperar a que el motor pare. Transcurridos dos segundos aparece el escalón de 3 voltios a la entrada del motor y se genera la salida que se observa en la gráfica anterior.

Si se hubiera recogido la señal de salida en un bloque *To Workspace* (que es como se debe hacer ya que el bloque *Scope* consume muchos más recursos y hace la ejecución más lenta) podríamos representar la señal mediante el comando *plot*. Para ello sustituiremos en el esquema anterior de Simulink el bloque *Scope* por un bloque *To Workspace*:



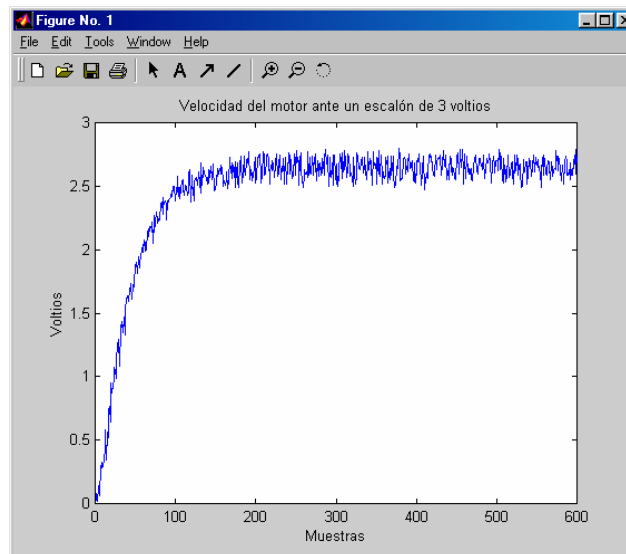
Puesto que sabemos el número de muestras que se recogen en los 2 segundos iniciales, ya que conocemos el periodo de muestreo, ( $n^\circ \text{ muestras} = 2 / T$ ), podemos eliminar del vector en el que se almacena la señal, los datos correspondientes a los 2 primeros segundos. En este caso, el número de muestras recogidas en los dos primeros segundos es  $2/0.01 = 200$ . Puesto que la ejecución ha durado 8 segundos, se han recogido en total 801 muestras. Para eliminar las 200 primeras muestras de la variable en la que están almacenadas, desde la línea de comandos de Matlab haremos lo siguiente:

```
» velocidad = velocidad(201:801)
```

De esta forma ya se pueden analizar y representar los datos como si se hubiera realizado el escalón de referencia en el instante cero de la ejecución. Para ello simplemente ejecutaremos desde Matlab:

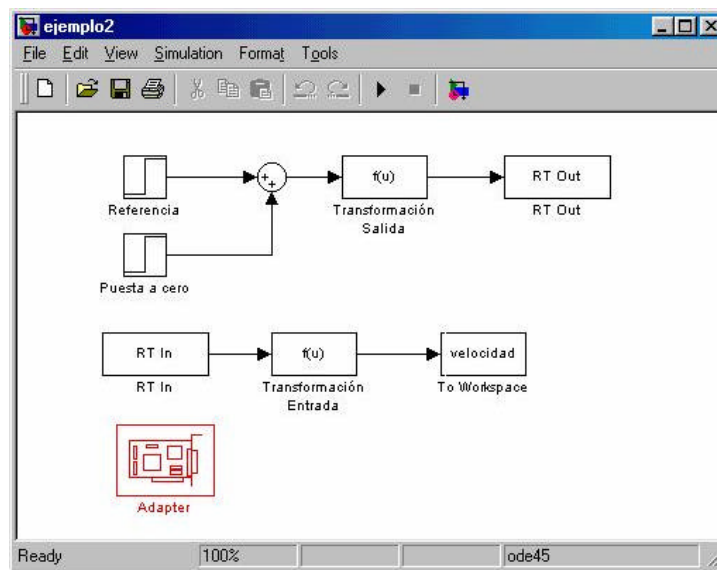
```
» plot(velocidad)
```

La gráfica que se obtendrá será la que se muestra a continuación:



Otro aspecto a destacar es que la tarjeta de adquisición de datos, una vez ejecutado el programa, mantiene en sus canales de salida el último valor y es recomendable dejar el valor de la salida de estos canales a 0 voltios. Para ello se hará lo siguiente: suponiendo que la ejecución (tiempo de simulación) dura 10 segundos, la referencia a generar será la suma de las señales de dos bloques escalón. El primero será idéntico al anterior, y el otro será un escalón con valor inicial 0, y a los 8 segundos generará un valor final opuesto en signo al valor final del primero (en nuestro ejemplo  $-3$  voltios). De esta forma, al ser la referencia la suma de ambos hará que a los ocho segundos la referencia sea 0 y por tanto se permita parar el motor. El esquema Simulink resultante sería:





Al igual que en el caso anterior, puesto que conocemos el número de muestras que se toman, podemos eliminar aquellas que no nos interesan.

## 8. Control en velocidad.

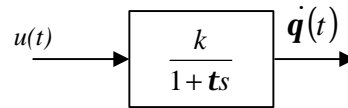
Se pretende realizar el control en velocidad del servomotor. El objetivo es que se introduzca al servomotor una velocidad de referencia y que éste alcance dicha velocidad satisfaciendo las condiciones impuestas de diseño. Para controlar en velocidad el servomotor deberán realizarse los siguientes pasos:

- 1) El primer paso es obtener el modelo matemático del servomotor. Este modelo se obtendrá realizando una identificación del sistema.
- 2) Una vez obtenido el modelo, se pasará al dominio discreto.
- 3) Utilizando el modelo discreto del servomotor y considerando las especificaciones de diseño, se diseñará el regulador.
- 4) Se comprobará mediante simulación que el comportamiento del sistema con el regulador seleccionado es el deseado.
- 5) Por último, tras comprobar que se cumplen las especificaciones de diseño en el sistema simulado, se implementará el regulador en el sistema real.

### 8.1. Identificación del servomotor.

El proceso de identificación pretende caracterizar matemáticamente la función de transferencia del proceso mediante la respuesta del sistema, obtenida en la entrada del convertidor A/D, como resultado de aplicar una tensión de excitación o control generada por el computador mediante el convertidor D/A.

Para realizar la identificación de este sistema se debe estudiar su respuesta en bucle abierto frente a una entrada en escalón. El esquema de control en velocidad de un motor de CC en bucle abierto se puede modelar como:



donde  $u(t)$  es la tensión de referencia que se introduce en el servomotor, y  $\dot{q}(t)$  es la tensión proporcional a la velocidad angular.

Por tanto, para identificar el modelo del servomotor, debe determinarse el sistema de primer orden cuya respuesta es la velocidad del servomotor. Es decir, hay que identificar la función de transferencia:

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

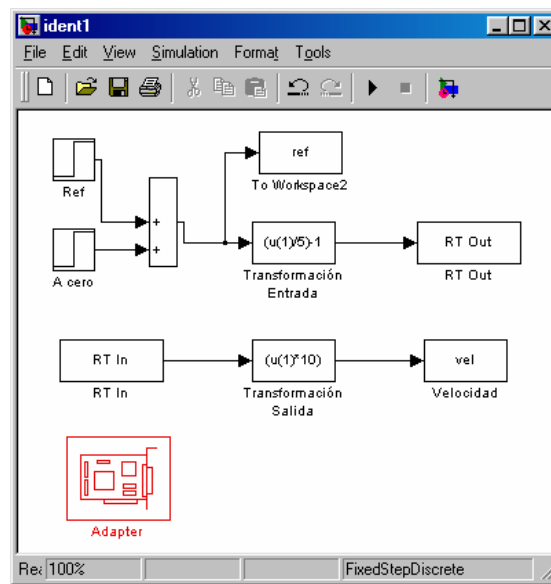
donde  $k$  es la ganancia del sistema y se obtiene como el cociente del valor final alcanzado por la señal de salida y la magnitud del escalón aplicado al sistema; y la constante de tiempo  $\tau$  corresponde al tiempo transcurrido desde el inicio del escalón hasta que la salida alcanza el 63%, aproximadamente, del valor final.

---

**Ejercicio 1.** Identificar el sistema de primer orden que a partir de una tensión de entrada proporciona como salida la velocidad del servomotor (sin freno).

Puesto que el rango de tensiones que se puede suministrar va de 0 a 10V, se deben tomar medidas de la respuesta del sistema para una entrada de 5V. No obstante no es suficiente hallar la función de transferencia evaluando los resultados obtenidos ante un único escalón. Se debe comprobar también la linealidad del sistema, es decir, comprobar si con escalones de entrada de distinta magnitud, se obtiene la misma función de transferencia. Para ello, obtendremos la función de transferencia para escalones de 1, 3, 5, 7 y 9 voltios (por ejemplo). A partir de las respuestas obtenidas (en el caso de que el proceso no sea lineal, y siempre es no lineal en mayor o menor medida) debemos elegir una función de transferencia que sea un compromiso entre todas ellas.

Para tomar los datos de cada uno de los experimentos (para escalones de distinta magnitud) utilizaremos el esquema siguiente:



Previamente será necesario realizar las siguientes conexiones de la tarjeta con el servomotor:

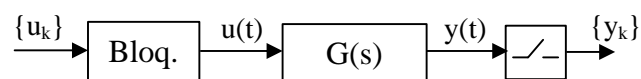
Pin ACL – 9138	Pin tarjeta de expansión del servomotor
1 (canal 0 de entrada analógica)	35 (señal del tacogenerador)
10 (GND de entrada analógica)	Señal de tierra del propio servomotor
30 (canal 0 de salida analógica)	33 (tensión de entrada positiva)
GND (GND de salida analógica)	34 (tensión 0)

La salida del sistema será la tensión de referencia que se introduce al servomotor. Dicha tensión será introducida por el canal 0 de salida analógica. La entrada al sistema corresponderá a la velocidad analógica del servomotor, que será leída por el canal 0 de entrada analógica.

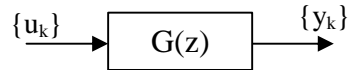
En los bloques *RT Out* y *RT In* se especificará el canal de salida y entrada (en ambos casos el canal 0, que corresponde al *Adapter Channel 1*) y el periodo de muestreo (0.01 seg).

## 8.2. Discretización del sistema.

Como es conocido, cuando se realiza un control digital de un proceso continuo, es necesario que la señal de entrada al proceso se convierte de discreta a continua, y que la señal de salida se convierta de continua a discreta. Esto se consigue utilizando un bloqueador (retenedor) a la entrada del proceso y un muestreador a la salida:



Para realizar el análisis y diseño del sistema de control es necesario calcular el sistema discreto equivalente al sistema continuo. Es decir, hay que obtener el sistema discreto que sustituye al conjunto formado por el bloqueador, sistema continuo y muestreador:



En Matlab la función **c2dm** obtiene el sistema discreto equivalente a un sistema continuo. La sintaxis de esta función es la siguiente:

» [numd,dend] = c2dm(num,den,Ts,'método')

Esta función obtiene el sistema discreto  $G(z) = \frac{numd(z)}{dend(z)}$  equivalente al sistema continuo  $G(s) = \frac{num(s)}{den(s)}$  considerando un periodo de muestreo de  $T_s$  segundos. Si se desea considerar un retenedor de orden cero en la entrada del sistema, el parámetro 'método' debe ser 'zoh'.

### Elección del periodo de muestreo

Para una correcta discretización del sistema continuo, el periodo de muestreo debe ser menor que la constante de tiempo del sistema. Habitualmente se selecciona el periodo de muestreo  $T$  de forma que se cumpla:

$$T < \frac{t}{10}$$

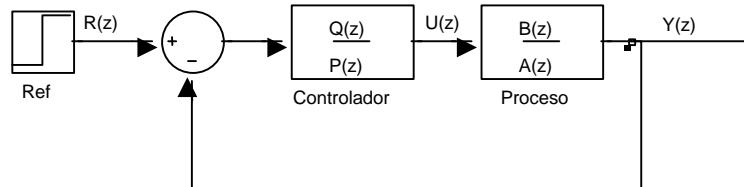
donde  $t$  es la constante de tiempo del sistema.

Lógicamente el periodo de muestreo seleccionado debe ser admisible por el sistema de adquisición. Para la realización de la práctica un periodo de muestreo de 0.01 segundos será válido.

**Ejercicio 2.** Obtener el sistema discreto equivalente al sistema continuo que modela el servomotor de CC. Considerar un periodo de muestreo de 0.01 segundos, comprobando que dicho periodo es válido para discretizar el sistema.

### 8.3. Control del sistema.

Para controlar en velocidad el servomotor se debe realizar el siguiente esquema de control:



donde el bloque *Proceso* corresponde al modelo discreto del servomotor y el bloque *Controlador* representa el regulador discreto que se debe diseñar para controlar de forma adecuada el servomotor.

**Ejercicio 3.** Diseñar un regulador PID discreto para controlar en velocidad el servomotor de forma que:

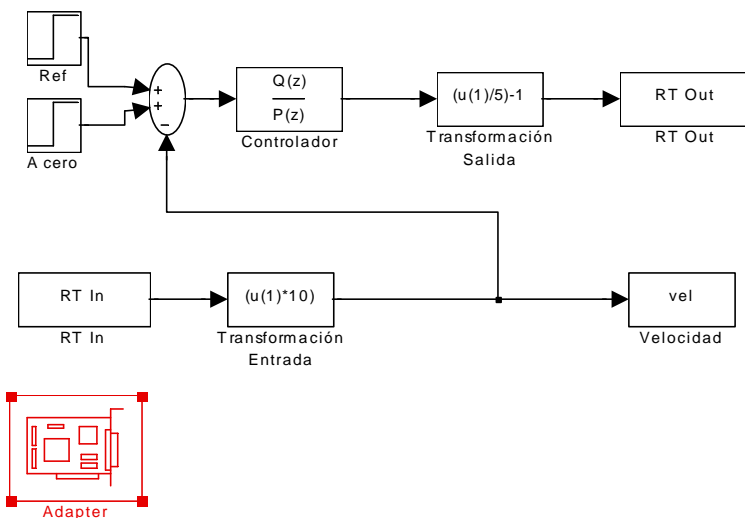
- (i) Asigne el polo en  $s = -10.5$  en bucle cerrado y elimine el error en régimen permanente ante una entrada en escalón.
- (ii) Asigne el polo en  $s = -35.6$  en bucle cerrado y elimine el error en régimen permanente ante una entrada en escalón

En cada apartado:

- Comprobar que el sistema cumple **en simulación** las especificaciones de diseño.
- Analizar la señal de control requerida.
- Discutir los resultados obtenidos.

Se recomienda que al simular el sistema se tenga en cuenta que la señal de entrada al motor no excede los valores de tensión admisibles por el servomotor (de 0V a +10V). Para ello puede utilizarse el bloque *Saturation* de la librería *Nonlinear*,

Una vez que se ha comprobado en simulación que el regulador diseñado satisface las especificaciones establecidas, el siguiente paso es llevar a cabo la implementación real del control en el servomotor. Para ello se utilizará el siguiente esquema de Simulink utilizando el software de comunicación con la tarjeta de adquisición *Extended Real Time Toolbox*:



Como puede observarse, para implementar el control real el bloque del proceso que modela el servomotor se ha sustituido por un bloque de entrada y de salida de la tarjeta de adquisición de datos (que representan la salida y la entrada al proceso respectivamente).

Los periodos de muestreo de los distintos bloques así como los parámetros de simulación serán los mismos utilizados en el apartado de identificación.

**Ejercicio 4.** Implementar de forma real el control diseñado para cada uno de los apartados del ejercicio 3.

- Introducir diferentes escalones y comprobar la repuesta real del servomotor.
- Analizar la señal de control.
- Comentar los resultados obtenidos.

### Importante

- Debe entregarse un informe detallando cada uno de los ejercicios realizados en la práctica.