

Control por computador

Manual de la Práctica 1: Implementación digital de reguladores analógicos



Jorge Pomares Baeza

Francisco Andrés Candelas Herías

Grupo de **I**nnovación **E**ducativa en **A**utomática



Universitat d'Alacant
Universidad de Alicante

© 2009 GITE – IEA

Introducción

En este experimento se pretende realizar la programación de un controlador digital para un servomecanismo

Resultaría difícil diseñar el sistema de control sin pensar en un proceso concreto para controlar ya que muchas características del sistema de control dependen directamente de qué se va a controlar. Por esa razón seguidamente se expone el diseño de un compensador para un servomecanismo de posición sencillo que hará el papel de proceso a controlar. No se va a profundizar mucho en ese diseño puesto que no es este el objetivo de este experimento.

Objetivos

- Repasar los principales conceptos de control analógico y su relación con el control por computador.
- Describir conceptos fundamentales en control digital tales como bloqueador de orden cero.
- Dar a conocer los principales pasos requeridos para realizar el control por computador de un sistema físico.

1. Funciones de transferencia

En un sistema amplificador es común hablar de la ganancia del amplificador. Este término referencia cuanto mayor es la señal de salida del sistema amplificador comparada con la señal de entrada. Por ejemplo, un amplificador con una ganancia de voltaje de 10 proporcionará para un voltaje de entrada de 2 mV, una salida de 20 mV, o si la entrada es de 1 V una salida de 10 V. La ganancia representa la relación matemática entre la salida y la entrada para el denominado bloque de ganancia:

$$\text{Ganancia} = \frac{\text{salida}}{\text{entrada}}$$

Sin embargo, en muchos sistemas la relación entre la salida y la entrada viene dada por una ecuación diferencial y, por lo tanto, no puede expresarse con una simple ganancia. En estos casos se puede transformar la ecuación diferencial en una ecuación algebraica utilizando la denominada Transformada de Laplace. De esta manera, las ecuaciones diferenciales describen cómo se comportan los sistemas con el tiempo y se transforman haciendo uso de la transformada de Laplace en ecuaciones algebraicas, que no implican el tiempo. Este proceso suele denominarse como el paso del dominio del tiempo al dominio s o de Laplace. En este último caso se puede definir la relación entre la salida y la entrada en términos de la denominada función de transferencia. Una función de transferencia representa la relación entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada:

$$G(s) = \frac{Y(s)}{X(s)}$$

Donde $Y(s)$ representa la transformada de Laplace de la salida, $X(s)$ la transformada de Laplace de la entrada y $G(s)$ la función de transferencia (véase Figura 1).

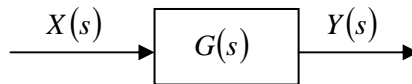


Figura 1. Función de transferencia.

Cabe mencionar que cuando una señal se encuentra expresada en el dominio del tiempo vendrá indicada como $f(t)$, mientras que si se encuentra en el dominio de Laplace se expresará como $F(s)$.

Los siguientes comandos de MATLAB calculan la transformada de Laplace de t^5 :

```
>> syms s t // variables simbólicas  
>> y=laplace(t^5)
```

El resultado obtenido es:

```
y = 120/s^6
```

Se puede obtener en formato con mayor legibilidad haciendo uso del comando:

```
>> pretty(y)
```

Se puede ejecutar el paso inverso, es decir, aplicar la transformada inversa de Laplace con la siguiente instrucción:

```
>> y=ilaplace(y)
```

El resultado es:

```
y = t^5
```

que se trata de la función original.

2. Conceptos básicos de sistemas de control discretos

La figura 2 muestra un sistema continuo genérico con realimentación:

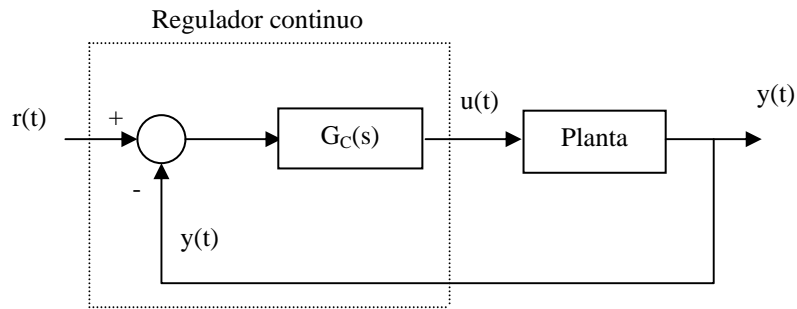


Figura 2. Sistema realimentado con control analógico.

El regulador continuo, encerrado en un cuadro en la figura anterior, puede ser reemplazado por un regulador discreto, como el mostrado en la figura 3, que realiza la misma tarea de control. La diferencia básica entre estos controladores es que el sistema digital utiliza para su funcionamiento señales discretas (o de muestras). Se observa que el equivalente discreto del controlador $G_c(s)$ es $G_c(z)$. Este último controlador es el obtenido haciendo uso de la transformada z que tiene la misma función en sistemas discretos que la transformada de Laplace en el análisis de sistemas continuos.

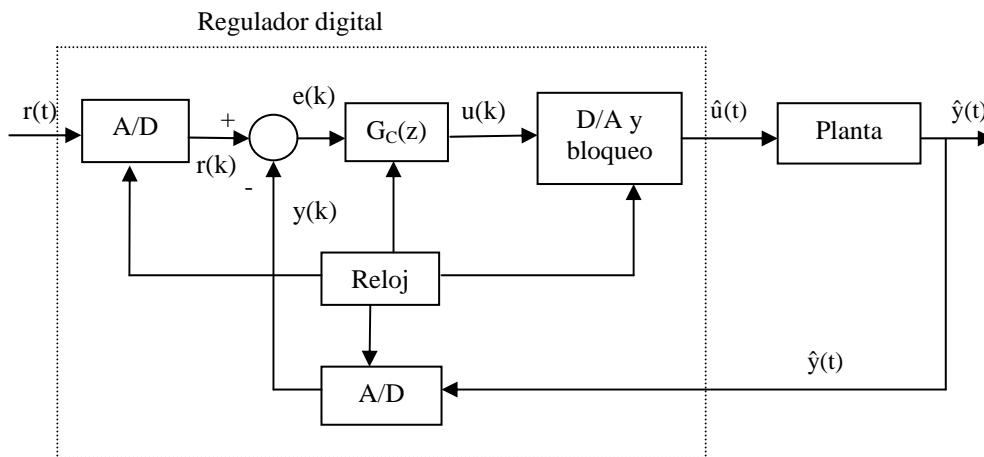


Figura 3. Sistema con control digital.

Los diferentes tipos de señales empleadas en la figura anterior se muestran en la figura 4:

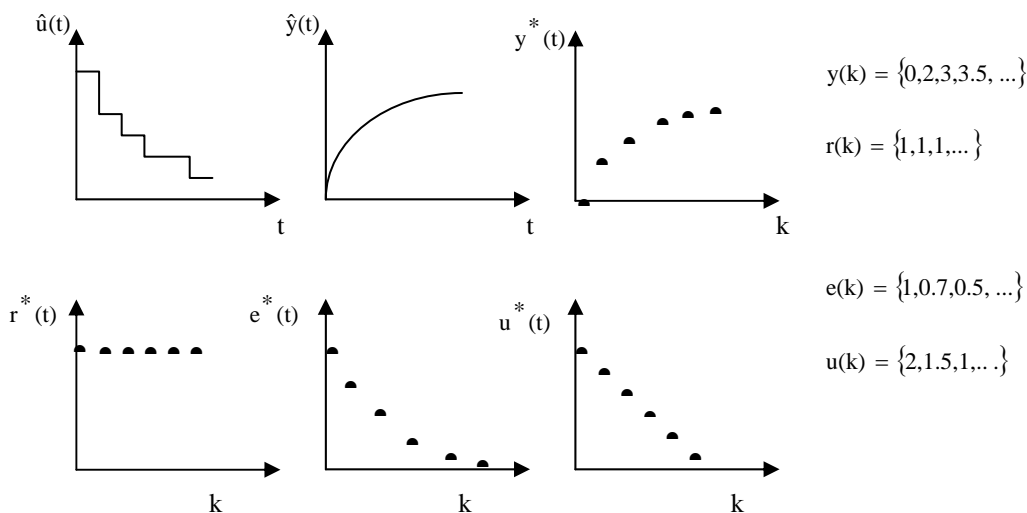


Figura 4. Distintos tipos de señales empleadas.

El propósito de este primer apartado de la práctica es mostrar como puede utilizarse MATLAB para trabajar con funciones discretas y diseñar sistemas de control digital.

En la anterior representación esquemática de un sistema de control digital, se observa que el sistema de control digital contiene partes discretas y continuas. Cuando se diseña un sistema de control digital, se necesita encontrar la equivalencia discreta de una porción continua de manera que solo se trabaje con funciones discretas. Un aspecto a considerar es que la referencia $r(t)$ puede ser también digital, $r(k)$, con lo que se podría prescindir del convertor A/D que realiza esta función.

Con este propósito, se va a considerar la porción del sistema de control digital que se indica en la figura 5 y se desglosa en la 6:

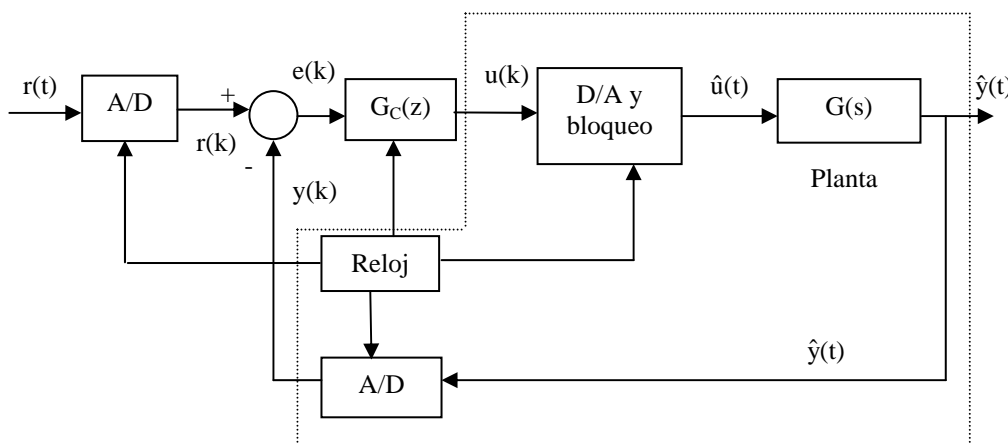


Figura 5. Conversión A/D y D/A cuando la planta es analógica.

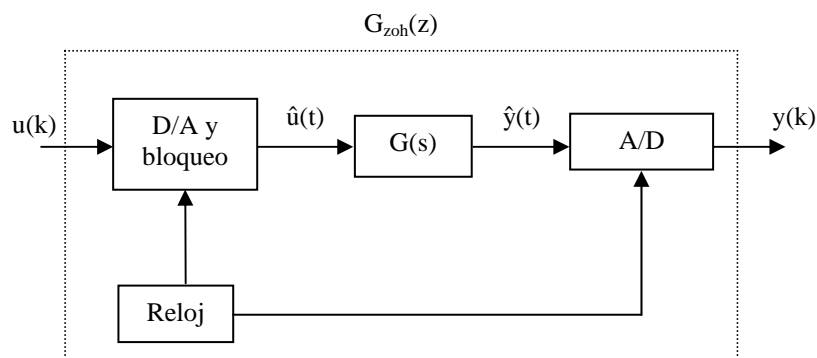


Figura 6. Circuito bloqueador de orden cero.

La señal de reloj, conectada a los convertidores D/A y A/D proporciona un pulso cada T segundos y cada D/A y A/D manda una señal únicamente cuando llega el pulso. Por otra parte, el convertor A/D de la salida se encarga de obtener las muestras $y(k)$ en los instantes indicados.

La filosofía del diseño es la siguiente; se quiere encontrar una función discreta $G_{zoh}(z)$ de manera que para una entrada constante a un sistema continuo $G(s)$, la salida muestreada del sistema continuo sea igual a la salida discreta. Supóngase que la señal $u(k)$ representa una muestra de la señal de entrada. Hay técnicas para

tomar esta muestra $u(k)$ y mantenerla para producir una señal continua $u_{\text{hat}}(t)$ (ver figura 7). El circuito denominado “bloqueador de orden cero” es el encargado de mantener $u_{\text{hat}}(t)$ constante para cada valor de $u(k)$ en el intervalo kT a $(k+1)T$.

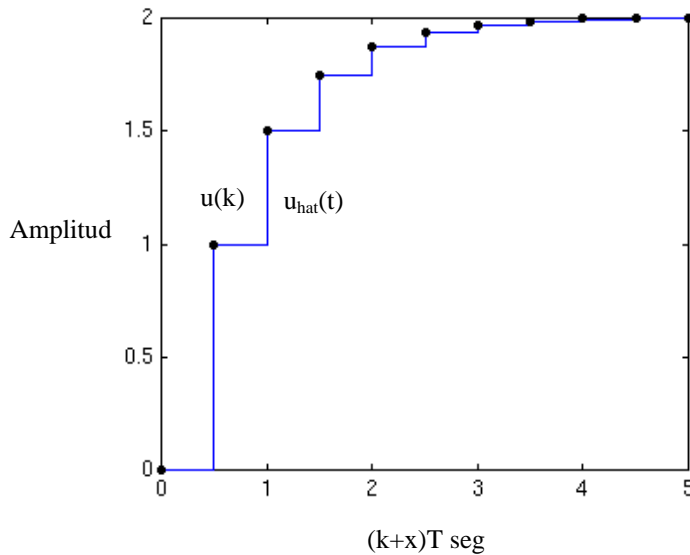


Figura 7. Señal muestreada.

La señal $u_{\text{hat}}(t)$ se aplica a $G(s)$ y el conversor A/D produce la salida $y(k)$, que será la misma señal que la que se obtendría si la señal discreta $u(k)$ circulara a través de $G_{\text{zoh}}(z)$, según la figura 8.

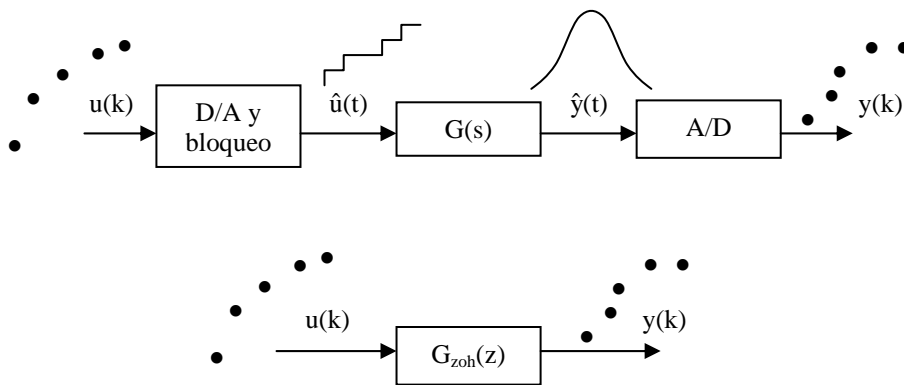


Figura 8. Conversiones D/A, A/D con planta analógica.

A continuación se muestra el modelo del bucle de control introduciendo $G_{\text{zoh}}(z)$ en lugar de la sección continua:

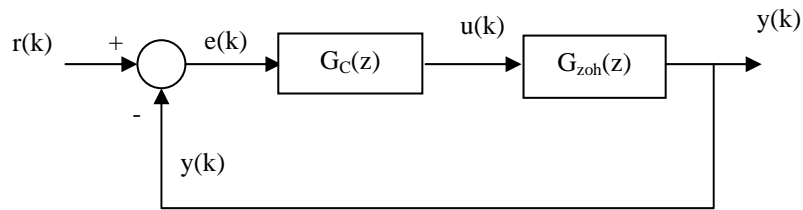


Figura 9. Bucle resultante con control digital y planta analógica.

Introduciendo $G_{zoh}(z)$, se puede modelar y diseñar sistemas de control digitales tratando únicamente funciones discretas.

3. Conversión de un sistema continuo a discreto

Existe una función en MATLAB denominada **c2d** que convierte un sistema continuo dado a un sistema discreto según lo comentado en el apartado anterior. El comando básico para **c2d** es el siguiente:

```
sys_d = c2d(sys, Ts, 'zoh')
```

donde T_s se trata del periodo de muestreo (en seg/muestra) y **zoh** indica que se utilizará un bloqueador de orden cero.

Supóngase que se dispone de la siguiente función de transferencia continua:

$$\frac{Y(s)}{U(s)} = \frac{1}{Ms^2 + bs + k}$$

- $M = 1$ kg
- $b = 7$ N.s/m
- $k = 15$ N/m

A continuación, ejecutando los siguientes comandos en MATLAB se obtendrá el sistema continuo equivalente:

```
M = 1;
b = 7;
k = 15;

num = 1;
den = [M b k];
sys = tf(num, den);

Ts = 1/100; // se selecciona el periodo de muestreo Ts igual a 1/100 seg
sys_d = c2d(sys, Ts, 'zoh')
```

Ejecutando estos comandos se obtendrá el siguiente resultado:

Transfer function:

$$\frac{4.885e-005 z + 4.772e-005}{z^2 - 1.931 z + 0.9324}$$

 Sampling time: 0.01

Con lo que ya se dispone de la función de transferencia en forma discreta.

4. Caso Práctico: El motor de corriente continua y la carga

Antes de estudiar el sistema servomecanismo al completo se va a tratar el motor de corriente continua empleado y la carga a controlar.

El esquema del motor considerado junto con la carga se puede ver en la figura siguiente. Se ha escogido un motor controlado por inducido (campo magnético excitador fijo y corriente por la armadura variable) ya que este tipo de motores permiten un control de posición más suave. Para simplificar los cálculos se va a suponer que la inductancia L_a y el coeficiente de amortiguamiento viscoso B son despreciables. Esta es una buena consideración debido a que realmente esos valores son muy bajos.

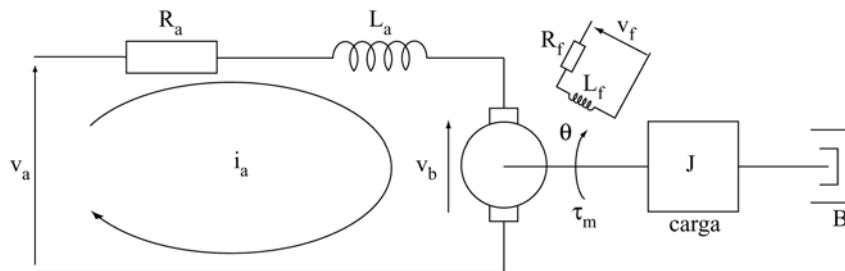


Figura 10. Esquema de un motor controlado por inducido.

Los valores para el resto de parámetros son los siguientes:

- $R_a = 1.62 \Omega$
- $J = 4,7281 \times 10^{-5} \text{Kg} \cdot \text{m}^2$
- $K_T = 7,55086 \times 10^{-2} \text{Nxm/A}$
- $K_E = 0,0754 \text{V/rad/s}$

Donde R_a se refiere a la resistencia del bobinado, J al momento de inercia, K_T es la constante de proporcionalidad entre i_a y τ_m , y K_E es la constante de fuerza contraelectromotriz.

A continuación se describen las ecuaciones del motor:

$$\tau_m = K_T \cdot i_a$$

$$v_a - v_b = i_a \cdot R_a + L_a \cdot \frac{d\theta}{dt} = i_a \cdot R_a$$

$$v_b = K_E \cdot \frac{d\theta}{dt} = K_E \cdot \omega_s$$

$$\tau_m = J \cdot \frac{d^2\theta}{dt^2} + B \cdot \frac{d\theta}{dt}$$

Que pasadas al dominio de Laplace son:

$$\tau_m(s) = K_T \cdot I_a(s)$$

$$V_a(s) - V_b(s) = R_a \cdot I_a(s)$$

$$V_b(s) = K_E \cdot s \cdot \theta(s) = K_E \cdot \omega_s(s)$$

$$\tau_m(s) = J \cdot s^2 \cdot \theta_s(s) = J \cdot s \cdot \omega_s(s)$$

Para obtener con mayor facilidad la función de transferencia del conjunto motor-carga, se puede dibujar antes su diagrama de bloques. Esto se hace considerando primero las ecuaciones eléctricas y dinámicas del motor en el dominio del tiempo y transformándolas luego al dominio de Laplace. Expresando estas últimas en un diagrama de bloques se obtiene el diagrama de la siguiente figura.

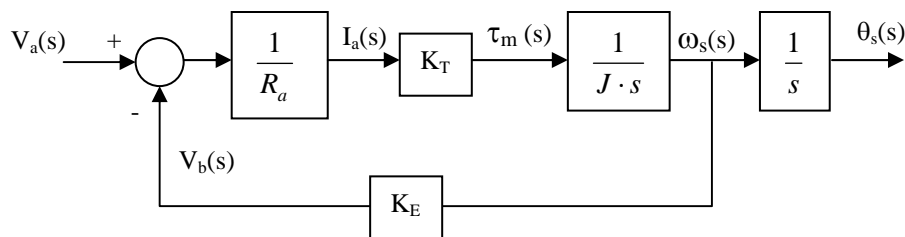


Figura 11. Diagrama de bloques del motor de corriente continua.

A partir de las expresiones anteriores resulta fácil obtener ya la función de transferencia a partir del diagrama de bloques. Esta función de transferencia será utilizada para realizar la simulación del servomecanismo (tras su conversión a digital) empleando un controlador digital.

$$\omega_s(s) = \frac{\frac{K_T}{R_a} \cdot \frac{1}{J \cdot s}}{1 + \frac{K_T}{R_a} \cdot \frac{1}{J \cdot s} \cdot K_E} E_a(s) = \frac{K_T}{R_a \cdot J \cdot s + K_T \cdot K_E} E_a(s) =$$

$$= \frac{7,55086 \cdot 10^{-2}}{0,010854 \cdot s + 5,6933 \cdot 10^{-3}} E_a(s)$$

$$\theta_s(s) = \frac{7,55086 \cdot 10^{-2}}{s \cdot (0,010854 \cdot s + 5,6933 \cdot 10^{-3})} E_a(s) \Rightarrow G_m(s) = \frac{\theta_s(s)}{E_a(s)} = \frac{7,55086 \cdot 10^{-2}}{s \cdot (0,010854 \cdot s + 5,6933 \cdot 10^{-3})}$$

4.1 Servomecanismo de control digital

En la figura 12 se observan los bloques básicos de un servomecanismo analógico así como el lazo de realimentación de la salida. La planta suele estar formada por un amplificador que adapta la señal de control

a una señal con suficiente potencia para gobernar el motor y la carga que este controla. El regulador es el elemento encargado de generar la señal de control necesaria para gobernar la planta de forma que la salida de ésta (salida del sistema) tenga el valor indicado por la señal de consigna (entrada al sistema). Para ello evalúa la señal de error, que es la diferencia entre la salida deseada que marca la señal de consigna y la salida real del sistema. Aunque la realimentación es necesaria para poder gobernar correctamente el sistema, ésta produce oscilaciones en la salida, siendo el objetivo del regulador reducir esas oscilaciones en lo posible, de forma que se cumplan unas especificaciones exigidas. En este caso, el regulador es un circuito electrónico formado por componentes analógicos y situado en la cadena directa de realimentación.

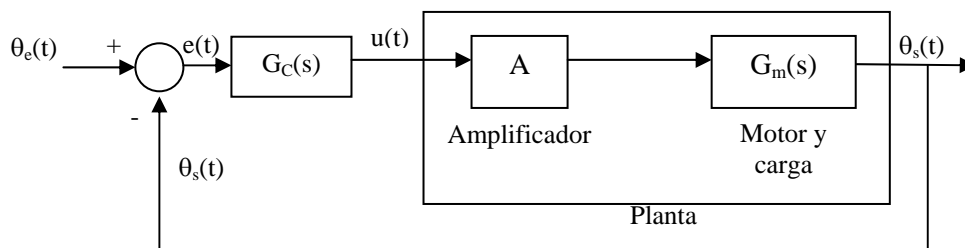


Figura 12. Diagrama de bloques de un bucle de control continuo.

donde:

- $\theta_e(t)$: Posición de entrada o consigna
- $e(t)$: Señal de error
- $u(t)$: Señal de control
- $\theta_s(t)$: Posición de salida

Se puede realizar un regulador con un sistema microprocesador o un computador, de forma que el regulador esté básicamente constituido por un programa software que analice una entrada y calcule una salida. Esto da lugar a un regulador digital y a un sistema digitalizado. Este es el propósito del presente experimento.

Sin embargo, antes de describir la implementación del controlador digital se habrá de considerar conceptos ya descritos anteriormente. Se requiere un convertidor A/D a la entrada del regulador y un convertidor D/A a la salida, ya que la planta del sistema está constituida por componentes analógicos y gobernada por señales continuas, y el sistema computador está formado por elementos digitales que procesan señales discretas, que son en realidad tablas con valores digitalizados de la señal tomados en ciertos instantes de tiempo. Así, el convertidor A/D presente a la entrada del regulador debe ser un elemento capaz de tomar los valores de la señal de error cada cierto intervalo de tiempo (periodo de muestreo) y obtener el código digital correspondiente (muestreo de la señal). Por otra parte, el convertidor D/A debe obtener un valor analógico correspondiente a un código digital dado en un instante y mantenerlo hasta que le llegue un nuevo código digital (efecto bloqueador o mantenedor). Esto último es necesario porque la planta del sistema es analógica, y debe ser controlada por una señal analógica continua en el tiempo.

En la siguiente figura se observa un esquema completo de un servomecanismo que emplea un regulador digital.

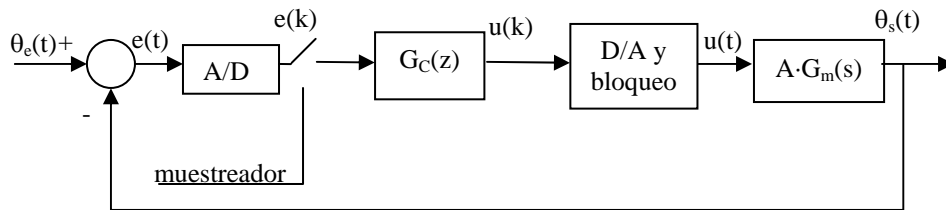


Figura 13. Diagrama de bloques de un bucle de control discretizado.

donde:

- $\theta_e(t)$: Posición de entrada o consigna
- $e(t)$: Señal de error
- $u(t)$: Señal de control
- $\theta_s(t)$: Posición de salida
- $e(k)$: Señal de error digital
- $u(k)$: Señal de control digital

El servomecanismo de posición que se va a utilizar en este experimento se basa en el esquema de la figura 13, pero tiene ciertas variaciones. La consigna o entrada al sistema no será una señal continua o analógica, sino un valor que se proporcionará al programa que actuará como regulador; será un dato numérico ya dentro del sistema que no hay que convertir a digital. Para medir el valor de salida del servomecanismo en muchas implementaciones se emplea un encóder óptico que es capaz de proporcionar el valor de posición ya en forma digital (este tipo de sensores será objeto de estudio en la práctica 2 de la asignatura). Sin embargo, para el desarrollo de este experimento se obtendrá la salida del sistema tras realizar la discretización de la planta, lo que nos permitirá obtener la salida del sistema en función de la entrada (procedente del regulador) y las entradas y salidas en instantes anteriores. Otra circunstancia que simplifica el diseño en la práctica es que no se requiere realmente un elemento bloqueador del valor analógico dado por el convertidor D/A: se puede mantener el valor digital del dato que el programa regulador da al convertidor D/A en arrays de memoria. El esquema de este servomecanismo se puede observar en la figura 14. El regulador digital debe tomar el valor de la entrada al sistema y de la salida de este en un mismo instante de tiempo (muestreo) para calcular el siguiente valor de la señal de control que será convertido a un valor analógico y actuará como entrada a la planta hasta el siguiente instante de muestreo.

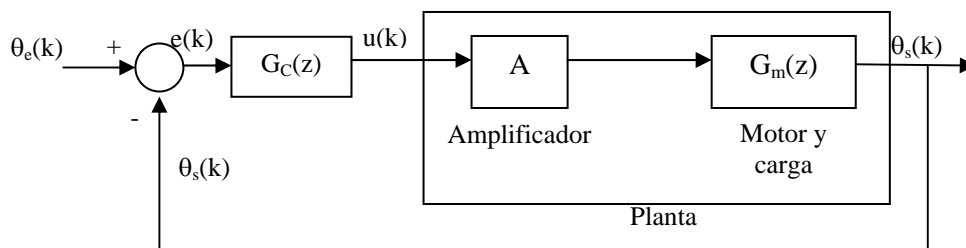


Figura 14. Diagrama de bloques discreto modificado.

donde:

- $\theta_e(t)$: Posición de entrada o consigna
- $e(t)$: Señal de error

- $u(t)$: Señal de control
- $\theta_s(t)$: Posición de salida
- $e(k)$: Señal de error digital
- $u(k)$: Señal de control digital

A continuación se trata de obtener un modelo del sistema para poder operar a la hora de diseñar el regulador. Para ello se va a considerar que tanto el amplificador como el convertidor D/A están ajustados para introducir una ganancia unitaria. Además, para representar el funcionamiento del convertidor D/A y su efecto mantenedor se utiliza el modelo del bloqueador de orden cero. Estas consideraciones quedan reflejadas en el esquema de la figura 15 donde $G_m(s)$ representa la función de transferencia del motor de corriente continua junto con la carga.

Lo dicho anteriormente puede resumirse en el siguiente diagrama de bloques:

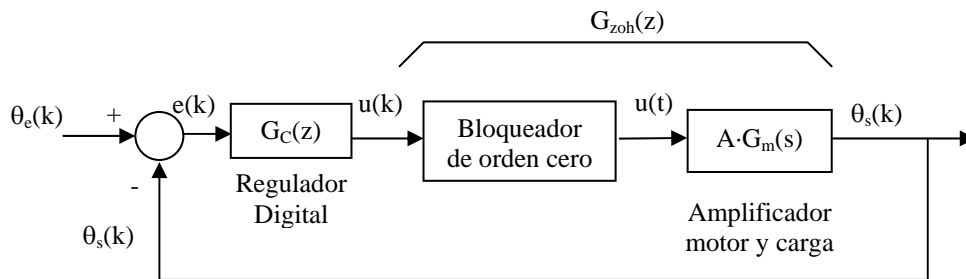


Figura 15. Diagrama de bloques discreto con bloqueador de orden cero.

4.2 Regulador PID

Aunque en este experimento se va a emplear un regulador digital tipo PID, conviene repasar el funcionamiento del regulador analógico PID, pues de este se va a derivar el regulador digital.

Las siglas PID indican las tres componentes que forma este tipo de regulador: componente proporcional, componente integral y componente diferencial. En la figura 16 se puede ver desglosado un regulador PID, expresado con la transformada de Laplace. Cada componente es por si sola un regulador que trata la señal de error obtenida al operar la señal de entrada al servomecanismo con la señal de salida realimentada y que produce una señal de control adecuada para lograr el mínimo valor de la señal de error.

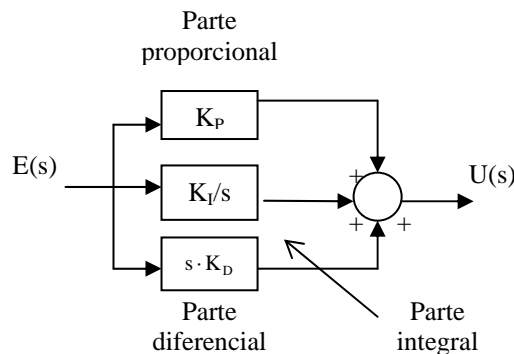


Figura 16. Componentes de un PID.

$$G_C(s) = \frac{U(s)}{E(s)} = K_P + s \cdot K_D + K_I / s$$

A continuación se recuerdan las características básicas de cada componente.

4.2.1 Regulador proporcional

La señal de control que genera es de valor proporcional a la señal de error según una constante K_P . Este tipo de regulador hace que el sistema alcance la posición de equilibrio (salida y entrada iguales) más rápidamente. Pero esa ventaja viene acompañada de un inconveniente: la respuesta de la salida se vuelve más oscilante, con menor amortiguamiento. Además, en algunos sistemas, este regulador puede producir inestabilidad. La mayoría de sistemas suelen tener un límite superior para el valor de la constante de ganancia K_P . A veces es indispensable para eliminar el error, aunque no se suele usar por sí solo.

4.2.2 Regulador integral

La principal característica de este tipo de regulador es que puede proporcionar un valor de señal de control sin que exista señal de error. Esto se debe a que la señal de control que produce depende de valores anteriores de la señal de error además del valor actual. La forma de la regulación integral es la siguiente:

$$u(t) = K_I \int_0^t e(t) dt \Rightarrow G_C(s) = \frac{U(s)}{E(s)} = \frac{K_I}{s}$$

Por sus características, este regulador es adecuado para eliminar o reducir los errores en el estado estacionario. Los inconvenientes de este tipo de regulador es que suele hacer más inestable el sistema controlado, y que puede causar saturación en la señal de control cuando los valores de la señal de error son altos y se mantiene bastante tiempo.

Debido a que los elementos físicos que puede gobernar la señal de control tienen un rango de trabajo limitado, puede ser que un valor excesivo de la señal de control produzca un exceso de esfuerzo o consumo en el elemento que controla. Un regulador integral puede causar que la señal de control tome valores excesivos si el error permanece en valores altos. Para solucionar este problema se puede parar la acción integral cuando el elemento gobernado alcance la saturación. Esto resulta más sencillo si el regulador está implementado digitalmente.

4.2.3 Regulador diferencial

En general mejora la respuesta transitoria del sistema al aumentar la amortiguación y la estabilidad. También hace que el sistema actúe más rápidamente. Se suele emplear junto con un regulador proporcional o integral ya que la regulación diferencial por sí sola no lleva el error a cero, a no ser que el sistema tenga algún término proporcional. La expresión para este regulador es la siguiente:

$$u(t) = K_D \frac{de(t)}{dt} \Rightarrow G_C(s) = \frac{U(s)}{E(s)} = s \cdot K_D$$

4.3 Regulador PID digital

Un regulador PID resulta de la suma de los tres anteriores tipos de reguladores, según se puede ver en la figura 16. El objetivo de tal combinación es lograr una reducción aceptable del error junto con valores aceptables de estabilidad y amortiguación. Muchos sistemas de control utilizan esta forma de regulador y su diseñador solo tiene que ajustar de forma conveniente los valores de las constantes, aunque hay varias técnicas para el diseño del regulador. Para el servomecanismo utilizado en este experimento se tratará de ajustar los valores de un regulador PID pero en versión discreta, como se explica a continuación.

La elección de que el regulador digital empleado sea del tipo PID está avalada por el hecho de que este tipo de regulador se puede implementar sin excesiva dificultad con un programa y suele dar buenos resultados. De hecho es el regulador usado normalmente para servomecanismos como el que se está estudiando, aunque hay otros como el regulador de “pulso muerto” que no provoca oscilaciones en la salida.

Se puede obtener la expresión de un regulador PID digital a partir del regulador PID analógico con ciertas transformaciones. Además, para obtener el efecto integral en su forma digital también se puede aplicar el concepto de integral. A continuación se muestra como se pueden obtener por separado las expresiones de las componentes del regulador PID digital, tanto en función de la variable tiempo en forma discreta (instantes de tiempo nT), como en función de la variable z . Las expresiones en el dominio de la transformada z serán útiles para obtener una expresión del regulador digital PID a partir de la cual se obtendrán expresiones con tiempo discreto en forma de ecuaciones en diferencias que serán las utilizadas en el algoritmo del regulador.

Para obtener la expresión de la componente diferencial se puede seguir estos pasos:

- Forma analógica en el plano s :

$$G_{CD}(s) = \frac{U_D(s)}{E(s)} = K_D \cdot s \Rightarrow U_D(s) = K_D \cdot s \cdot E(s)$$

- Pasando la expresión al dominio del tiempo:

$$u_D(t) = K_D \frac{de(t)}{dt}$$

- Transformándola a discreta, según la definición de la derivada, al considerar el tiempo como valores discretos y tomando la diferencia hacia atrás:

$$u_D(k) = K_D \frac{e(k) - e(k-1)}{T}$$

- Aplicando a la expresión anterior la transformada z :

$$\begin{aligned} U_D(z) &= \frac{K_D}{T} [E(z) - z^{-1}E(z)] = \frac{K_D}{T} (1 - z^{-1})E(z) \Rightarrow \\ \Rightarrow G_{CD}(z) &= \frac{U_D(z)}{E(z)} = \frac{K_D}{T} (1 - z^{-1}) = \frac{K_D}{T} \cdot \frac{z-1}{z} \end{aligned}$$

La expresión integral analógica se puede pasar directamente a discreta con una transformación bilineal.

$$s = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$$

$$R_I(s) = \frac{U_I(s)}{E(s)} = \frac{K_I}{s} \Rightarrow R_I(z) = \frac{U_I(z)}{E(z)} = K_I \cdot \frac{T}{2} \cdot \frac{1+z^{-1}}{1-z^{-1}} \Rightarrow U_I(z) = K_I \cdot \frac{T}{2} \cdot \frac{1+z^{-1}}{1-z^{-1}} E(z)$$

La expresión temporal en forma discreta será la siguiente:

$$U_I(z) = z^{-1}U_I(z) + K_I \frac{T}{2} [E(z) + z^{-1}E(z)] \Rightarrow u_I(k) = u_I(k-1) + TK_I \frac{e(k) + e(k-1)}{2}$$

Teniendo en cuenta que el término proporcional no sufre ninguna variación, se puede plantear ya el esquema del regulador PID digital. En la figura 18 se puede ver el esquema de regulador en función de instantes de tiempo y en la figura 19 aparece el esquema del mismo regulador pero, esta vez, en función de la variable z.

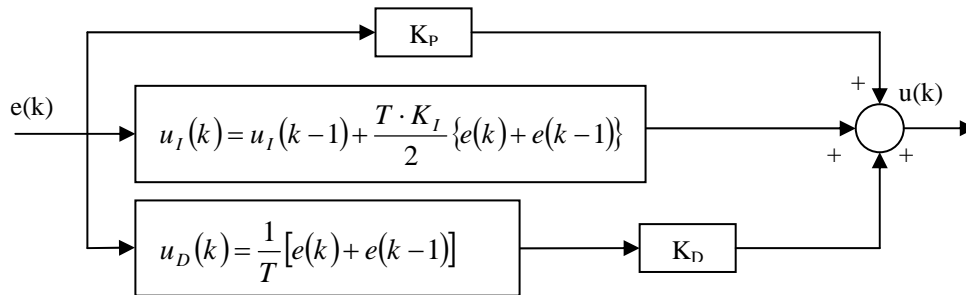


Figura 18. Regulador PID en función de instantes de tiempo.

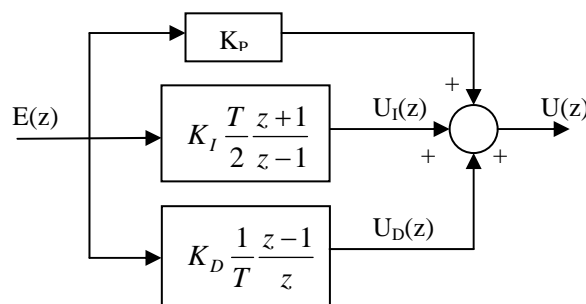


Figura 19. Regulador PID en función de la variable z.

A continuación se detalla el algoritmo del regulador PID que se habrá de implementar en este experimento.

4.4 Algoritmo del regulador PID digital

Para implementar el regulador digital con un programa se utiliza la expresión en tiempo discreto anteriormente obtenida. La misión del programa será tomar los valores de entrada y salida del sistema en el instante de muestreo nT ($\theta_e(k)$ y $\theta_s(k)$), calcular la señal de error $e(k)$ y con ella realizar los cálculos de la expresión del regulador para obtener el valor de la señal de control $u(k)$. Estos pasos se deben realizar repetidamente para cada instante de muestreo. Además el programa debe ir almacenando los valores de $e(k)$ y $u_i(k)$ en cada ejecución para en la siguiente ejecución disponer de ellos como $e(k-1)$ y $u_i(k-1)$. En este apartado se muestra el algoritmo general para este programa.

$n=0$

Para cada instante de muestreo hacer

Leer salida $\theta_s(k)$

Leer consigna $\theta_e(k)$

$e(k) = \theta_e(k) - \theta_s(k)$

$u_I(k) = u_I(k-1) + K_I \frac{e(k) + e(k-1)}{2} T$

$u_D(k) = \frac{e(k) - e(k-1)}{T} K_D$

$u(k) = K_P \cdot e(k) + u_I(k) + u_D(k)$

Enviar a la planta $u(k)$ y mantener

$n=n+1$

Orientado el algoritmo más hacia la programación se puede escribir de esta otra forma:

ERROR_ANTERIOR=0

INTEGRAL=0

Para cada instante de muestreo hacer:

Leer SALIDA

Leer CONSIGNA

ERROR = CONSIGNA-SALIDA

$INTEGRAL = INTEGRAL + K_I \frac{ERROR + ERROR_ANTERIOR}{2} T$

$DERIVADA = \frac{ERROR - ERROR_ANTERIOR}{T} K_D$

CONTROL= $K_P \cdot ERROR + INTEGRAL + DERIVADA$

ENVIAR CONTROL A LA PLANTA Y MANTENER

4.5 Ajuste de las constantes del regulador digital PID

Hasta este apartado se ha ido viendo el diseño del sistema servomecanismo y del regulador digital que se van a utilizar. Solo queda obtener unos valores adecuados para las constantes del regulador PID de forma que la respuesta del sistema sea aceptable. No se van a exigir especificaciones concretas del servomecanismo visto; solo se tratará de obtener una respuesta buena sin demasiada oscilación, ya que eso será suficiente para realizar el programa regulador y ver si este funciona correctamente.

Para el ajuste se tuvo en cuenta la manera en que influyen los valores de las constantes del regulador en el sistema: Valores de K_P y K_I mayores reducen el error pero disminuyen la estabilidad, y valores de K_D mayores mejoran la estabilidad a la vez que hacen más rápido al sistema.

Realizando distintas pruebas para un periodo de muestreo de 0,0625 y con un escalón de entrada de amplitud 10, se ha llegado a una respuesta bastante satisfactoria al tomar los valores:

$$K_P = 4$$

$$K_D = 1$$

$$K_I = 8$$

En las siguientes gráficas se observa la señal de salida, la señal de control y la señal de error.

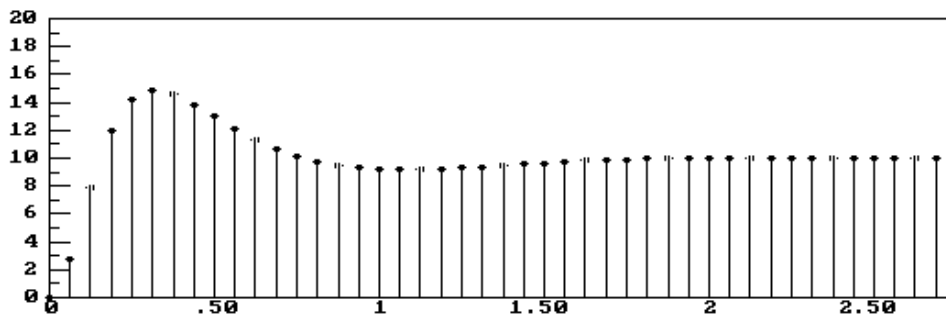


Figura 20. Señal digital de salida para $T=0,0625$ seg, $K_P=4$, $K_D=1$, $K_I=8$

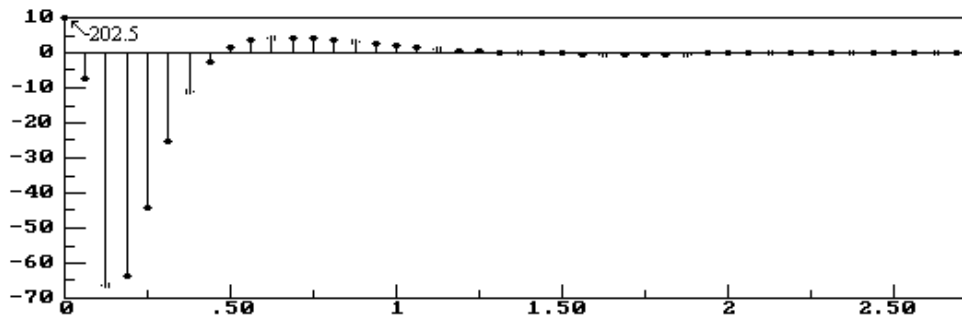


Figura 21. Señal de control para $T=0,0625$ seg,

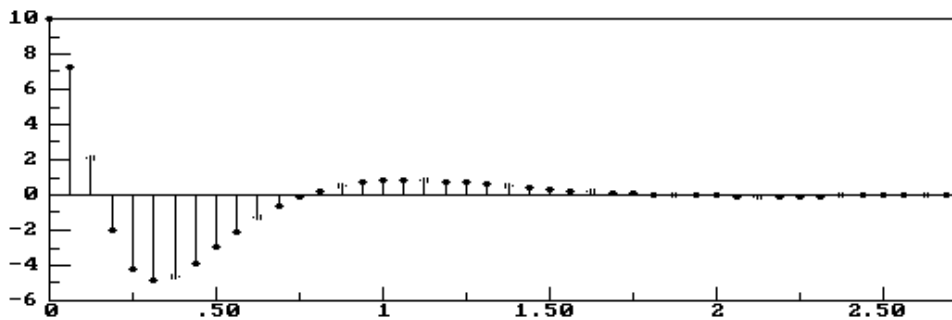


Figura 22. Señal digital de error para $T=0,0625$ seg,

4.6 Programa a implementar

Una vez expuestos los principales conocimientos teóricos requeridos, se desea obtener una implementación, en el lenguaje que se considere oportuno, del algoritmo expuesto en el apartado “Algoritmo del regulador PID digital”. Para el desarrollo de este algoritmo se harán uso de los parámetros establecidos en el apartado “Ajuste de las constantes del regulador digital PID”.

Una vez implementado el regulador, éste establecerá las acciones que representarán las entradas a la planta. Por otro lado es necesario establecer las salidas que constituirán la realimentación del bucle de control. Para obtener estas salidas se pasará a digital la planta cuya función de transferencia se desarrolló en el apartado “El motor de corriente continua y la carga”. Una vez pasado a digital se realizará un programa que tome las entradas procedentes del regulador y las salidas en instantes anteriores para generar la salida en el instante actual (**considerar un periodo de muestreo de 0.01**).

Ambos programas (regulador y planta) podrán combinarse en uno único, a partir del cual se pide obtener los siguientes datos:

- a) Determinar la evolución de la señal de control ante entrada escalón de 10, 5 y 1 unidad.
- b) Determinar la evolución de la salida del sistema ante las entradas anteriores.
- c) Determinar la evolución de la señal de error ante las entradas consideradas en el apartado a).
- d) Tratar de ajustar mejor los valores K_p , K_D y K_I para tener menos oscilación y respuesta más rápida.
- e) Realizar un estudio de cómo afecta cada una de las constantes anteriores al comportamiento del sistema.
- f) Obtener una combinación para los valores de las constantes K_p , K_D y K_I con la cual se obtiene un resultado satisfactorio.

Anexo: Introducción a MATLAB

Manejo de vectores y matrices

Se puede crear una matriz de ceros, M , de tamaño 3×6 con el comando:

```
M = zeros(3,6);
```

Cuando se desea dar el valor a toda una matriz se puede realizar directamente de la siguiente forma:

```
M=[1 2 3 4; 5 6 7 8;9 0 1 2];
```

obteniendo la siguiente matriz:

```
1     2     3     4
5     6     7     8
9     0     1     2
```

Las filas se separan por punto y coma y las columnas por espacios o comas. Por lo tanto, para definir un vector fila bastará con ejecutar el siguiente comando:

```
v=[1 2 3];
```

Y par un vector columna:

```
v=[1; 2; 3];
```

Puede crearse un vector con elementos entre 0 y 20 espaciados en incrementos de 2 de la siguiente manera:

```
t = 0:2:10
t =
    0    2    4    6    8   10
```

La manipulación de vectores es aún más fácil. Supóngase que se desea añadir 2 a cada uno de los elementos del vector 't'. Esto puede realizarse sencillamente de la siguiente manera:

```
b = t + 2
t =
    2    4    6    8   10   12
```

Supóngase ahora que se desea sumar dos vectores. Si ambos son de la misma longitud, la suma puede obtenerse con el siguiente comando:

```
c = t + b
c =
    2    6   10   14   18   22
```

Otra forma de asignar valores a una matriz es haciendo uso de los subíndices. El menor subíndice utilizado por MATLAB es 1. Los subíndices se escriben entre paréntesis. Por ejemplo:

`M(3, 4)=8;` Asigna al elemento en la fila 3, columna 4 el valor de 8.

Si se desea cambiar todo el valor de una fila o una columna, se puede hacer con el operador ":", de la siguiente manera:

`M(1, :)= [6 7 8];`

Asigna a la fila 1 el vector [6, 7, 8]. De esta manera, si M era una matriz de 3x3 de ceros, ahora queda:

```
6     7     8
0     0     0
0     0     0
```

Este mismo operador también puede utilizarse para extraer filas o columnas de matrices. Por ejemplo, si se desea almacenar en v el valor de la primera columna de la matriz M:

`v=M(:, 1);`

Si se dispone de dos matrices, y se quieren: sumar, multiplicar ó restar bastará con tratar esta operación como si fueran números normales. Por ejemplo:

Si se quieren multiplicar dos matrices A y B y almacenar el resultado en C:

`C=A*B;`

También existen comandos para matrices como `size`, que permite conocer las dimensiones de la matriz, `inv`, que permite invertir la matriz, `pinv`, que calcula la pseudoinversa de una matriz, o `det` que calcula su determinante. Para vectores también existen operaciones definidas como `length` que determina su longitud.

Otra operación que se puede realizar con matrices es su modificación de tamaño utilizando el comando `reshape`.

`B = reshape(A, m, n)`

devuelve una matriz de m por n cuyos elementos son tomados columna a columna de A. Devuelve un error si A no dispone de m*n elementos. En el siguiente ejemplo se muestra como modificar una matriz de tamaño 3x4 en otra de tamaño 2 x 6.

```
A =
  1     4     7    10
  2     5     8    11
  3     6     9    12
```

`B = reshape(A, 2, 6)`

```
B =
  1     3     5     7     9    11
  2     4     6     8    10    12
```

Comando FOR

La sintaxis de este comando es la siguiente:

for (contador), (ordenes) end;

El contador tiene la forma siguiente:

variable = a [,b] : c

donde:

variable es el contador en sí.

a es el valor inicial del contador.

b es el segundo valor del contador (opcional, si se omite, $b=a+1$), su función es determinar el incremento del contador.

c es el valor final del contador (variable).

En el siguiente ejemplo se muestra el uso del bucle for:

```
for c=0,0.5:2.5,  
    c %Se muestra el valor de la variable c  
end;
```

La salida del programa anterior es la siguiente:

```
c=  
    0  
c=  
    0.5  
c=  
    1  
c=  
    1.5  
c=  
    2  
c=  
    2.5
```

Funciones

MATLAB incluye muchas funciones standar como sin, cos, log, exp, sqrt, así como muchas otras. Matlab incorpora constantes comunes como 'pi', 'i' o 'j' para la raíz cuadrada de -1. Ejemplo:

```
sin(pi/4)  
  
ans =  
  
    0.7071
```

Para determinar como utilizar una función puede ejecutarse desde matlab help [function name].

Representación gráfica

Supóngase que se desea representar una onda senoidal como una función del tiempo. Para ello en primer lugar se debe crear un vector de tiempos:

```
t=0:0.25:7;
```

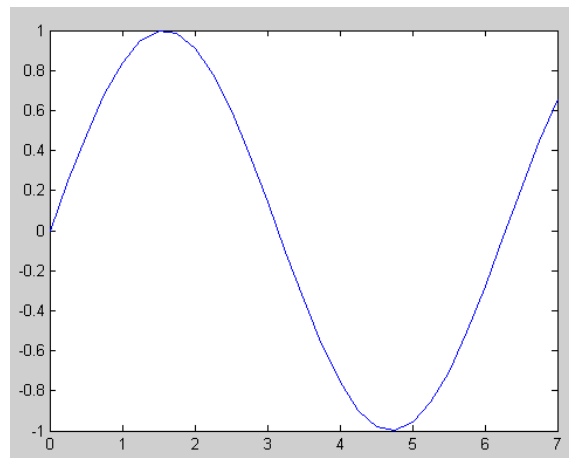
y posteriormente calcular el valor del seno en cada instante de tiempo:

```
y = sin(t);
```

Por último, ejecutando el siguiente comando:

```
plot(t,y)
```

se obtiene la gráfica buscada:



El comando plot dispone de distintos parámetros para el ajuste de distintas propiedades de la representación tales como la escala de los ejes.