

Señales Análogas y Digitales

Ricardo Rodríguez Bustinza
robust@uni.edu.pe

Índice

1. Definición de Señales	2
1.1. Clasificación de las Señales	2
2. Señales en Tiempo Continuo	2
2.1. Onda Seno	3
2.2. Onda Cuadrada	3
3. Señales en Tiempo Discreto	4
3.1. Impulso Unitario	4
4. Conversor Digital Análogo	5
4.1. DAC MCP4725	5
4.2. Características	6
5. Generador de Señales	7

1. Definición de Señales

Los conceptos de señales y sistemas aparecen en una amplia variedad de campos, de manera que las ideas y técnicas asociadas con estos conceptos juegan un papel importante en áreas tan diversas como: *comunicaciones, aeronáutica y astronáutica, diseño de circuitos, acústica, óptica, sismología, ingeniería biomédica, sistemas de generación y distribución de energía, control de procesos, reconocimiento de patrones, entre otros*. En general las señales contienen información acerca de la naturaleza o comportamiento de algún fenómeno, los sistemas son elementos que transforman señales, es decir, un sistema recibe señales de entrada, ejecuta algún procedimiento sobre éstas y produce señales de salida. Esta relación entre señales y sistemas se representan de manera general en un bloque como en la Figura 1.

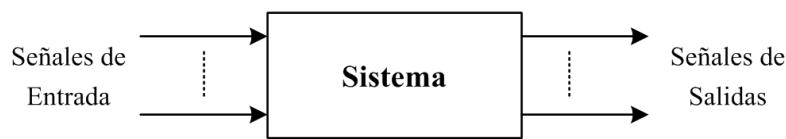


Figura 1: Diagrama de bloques de un sistema en general.

1.1. Clasificación de las Señales

En general una señal es cualquier cantidad cuya magnitud se representa matemáticamente como función de una o más variables independientes. En este curso se tratará exclusivamente el caso de funciones de una variable independiente y esta variable normalmente será el tiempo, aunque en algunas aplicaciones como en la geofísica interesa el comportamiento de la densidad, porosidad, resistividad eléctrica, por ejemplo, con respecto a la profundidad, o en la meteorología, interesa la variación de la presión, velocidad del viento, etc. Además del número de variables independientes del que dependen, las señales se clasifican de acuerdo a sus propiedades básicas como sigue:

- Señales Continuas y Señales Discretas

Existen dos tipos básicos de señales dependiendo de la naturaleza de la variable independiente (tiempo) que consideran: Señales de tiempo continuo y señales de tiempo discreto.

- Señales Determinísticas y Señales Estocásticas

Cualquier señal que se modela mediante una expresión matemática precisa, un conjunto específico de datos o una regla bien definida, de manera que los valores que toma la señal a lo largo del tiempo son conocidos sin ninguna incertidumbre, se denomina señal determinística sus valores futuros son perfectamente predecibles. Una señal estocástica es aquella relacionada con las señales aleatorias y cuyo valor futuro no se predice con exactitud.

2. Señales en Tiempo Continuo

En una señal de tiempo continuo, la variable tiempo es una variable continua y por ello estas señales están definidas para cualquier par de instantes de tiempo y para cualquier instante comprendido entre este par. Para este tipo de señales usaremos t para denotar a la variable independiente de tiempo continuo. Las señales más comunes usadas en la teoría de control se exponen a continuación:

2.1. Onda Seno

Una señal en tiempo continuo es aquella que toma cualquier valor en cualquier instante de tiempo, donde la variable independiente tiempo en cualquier instante desde $-\infty$ a $+\infty$. Como ejemplo de este tipo de señales está cualquier función matemática que dependa del tiempo, como $v(t) = v_p \cos(2\pi f t)$, donde t es la variable independiente y $v(t)$, la variable dependiente.

En este caso particular tenemos una función que se repite periódicamente en el tiempo y f indica la velocidad de repetición, normalmente f tiene unidades de Hertz o ciclos/seg. El inverso de f es el periodo T o el tiempo de duración de un ciclo. En la Figura 1 tenemos un ciclo de la señal $v(t)$, para $v_p = 10$ voltios y $f = 100$ hz.

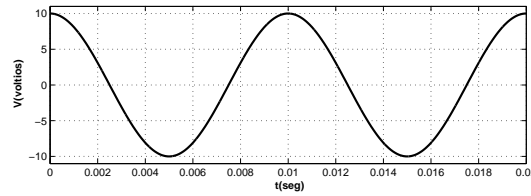


Figura 2: Señal senoidal periódica.

Si ahora a la onda senoidal de la Figura 2, le queremos agregar un ruido, esto lo logramos adicionando el comando `randn` de MATLAB. Lo cual nos produce la señal contaminada de la Figura 3.

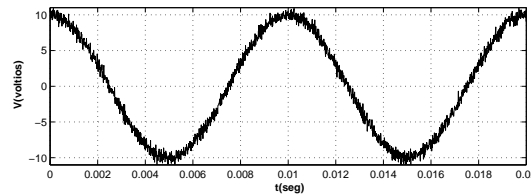


Figura 3: Señal senoidal periódica contaminada.

2.2. Onda Cuadrada

Mediante la función `square` de MATLAB, generamos onda cuadrada, con un periodo 2π y ciclo útil, especificado a través de un parámetro. Por ejemplo generamos una onda cuadrada de frecuencia 50Hz con un paso 1×10^{-4} y graficar hasta 0.08 segundos de la señal, duty de 50%.

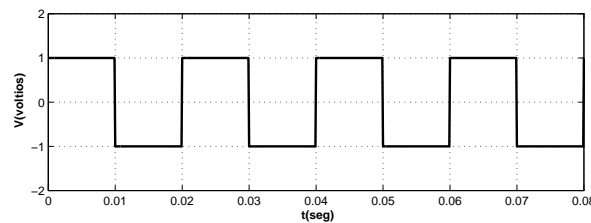


Figura 4: Señal onda cuadrada.

3. Señales en Tiempo Discreto

Una señal de tiempo discreto $x(k)$ solamente está definida en ciertos instantes discretos de tiempo. Una señal de tiempo discreto por lo tanto también se representa como una lista o secuencia de valores $\{x(1), x(2), x(3), \dots\}$. En este tipo de señales usaremos n para denotar la variable independiente. Una señal en tiempo discreto, solo está definida para ciertos valores del tiempo. Supóngase que tomamos muestras de la señal analógica ($f = 100\text{hz}$) de la Figura 2 (onda seno) y que estas muestras las tomamos cada 0.001 segundos (período de muestreo). Según esto, definimos la frecuencia de muestreo, $f_s = 1/T = 1/0.001 = 1000\text{S/seg}$. Si tomamos el eje de tiempo de la señal de la Figura 2, estarán contenidas 21 muestras (ver Figura 5).

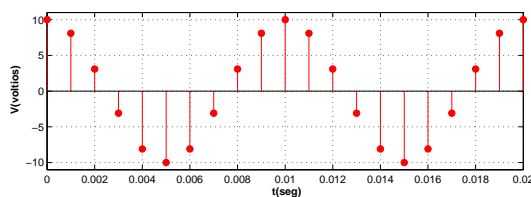


Figura 5: Señal senoidal periódica en tiempo discreto.

En la Figura 5 se ha considerado el eje de tiempo $t = kT$, es decir, si deseamos plotear la señal en el eje de muestras, el cambio sería $k = t/T$ el que indica eje de muestras (ver Figura 6).

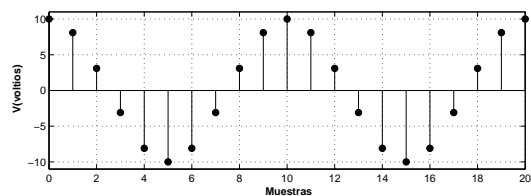


Figura 6: Señal senoidal periódica discreta con eje de muestras.

3.1. Impulso Unitario

Es una señal muy importante en el análisis de sistemas lineales. La forma de graficar una señal en tiempo discreto es usando la función `stem` en lugar del comando `plot` (ver Figura 7).

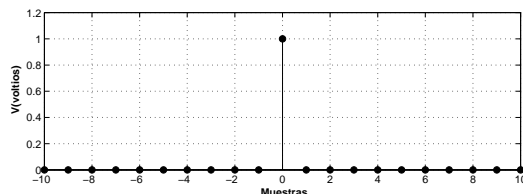


Figura 7: Señal impulsiva en tiempo discreto.

Si la señal es continua se debe convertir a una señal digital mediante un proceso según el proceso de *muestreo*, conversión a señal de tiempo discreto y *cuantización*, conversión de su conjunto de valores a un conjunto finito.

4. Conversor Digital Análogo

En un microcontrolador es un circuito integrado que convierte valores digitales (números binarios) en voltajes o corrientes analógicas continuas. El microcontrolador envía el valor digital (por ejemplo, un número entre 0 y un valor máximo, como 4095 para 12 bits) a través de un pin específico o una interfaz de comunicación. La resolución en bits del DAC (Digital Analog Converter) determina la precisión del voltaje de salida, con una mayor resolución (más bits) resultando en un voltaje más preciso y con más pasos intermedios.

- **Funcionamiento de un DAC en un Microcontrolador:**

Entrada Digital: El microcontrolador envía un dato digital, que representa el valor deseado en forma de un número binario.

Convertidor DAC: Este dato ingresa al DAC, el cual lo interpreta y lo transforma.

Salida Analógica: El DAC produce un voltaje o corriente analógica que es proporcional al valor digital enviado, con un rango que va de 0 a un valor máximo (por ejemplo, 3.3V).

- **Aplicación:**

Generador de Formas de Onda: comúnmente los generadores son equipos físicos (ver Figura 8). En nuestro caso, se requiere crear señales analógicas como senoidales o escalonadas desde el DAC, sera objeto de estudio del curso a través de los generados virtuales.



Figura 8: Equipo generador de ondas.

4.1. DAC MCP4725

Si deseamos generar voltajes analógicos desde un microcontrolador, el MCP4725 es el DAC mostrado en la Figura 9 es el idóneo para estas aplicaciones, ya que el controlado por el bus I2C que incluye las líneas de comunicación, SCL, SDA, además del suministro de voltaje, VCC y conexión a tierra (GND) para la asignación básica de pines I2C.

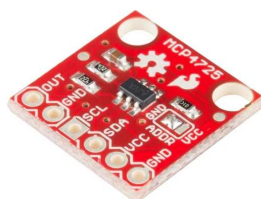
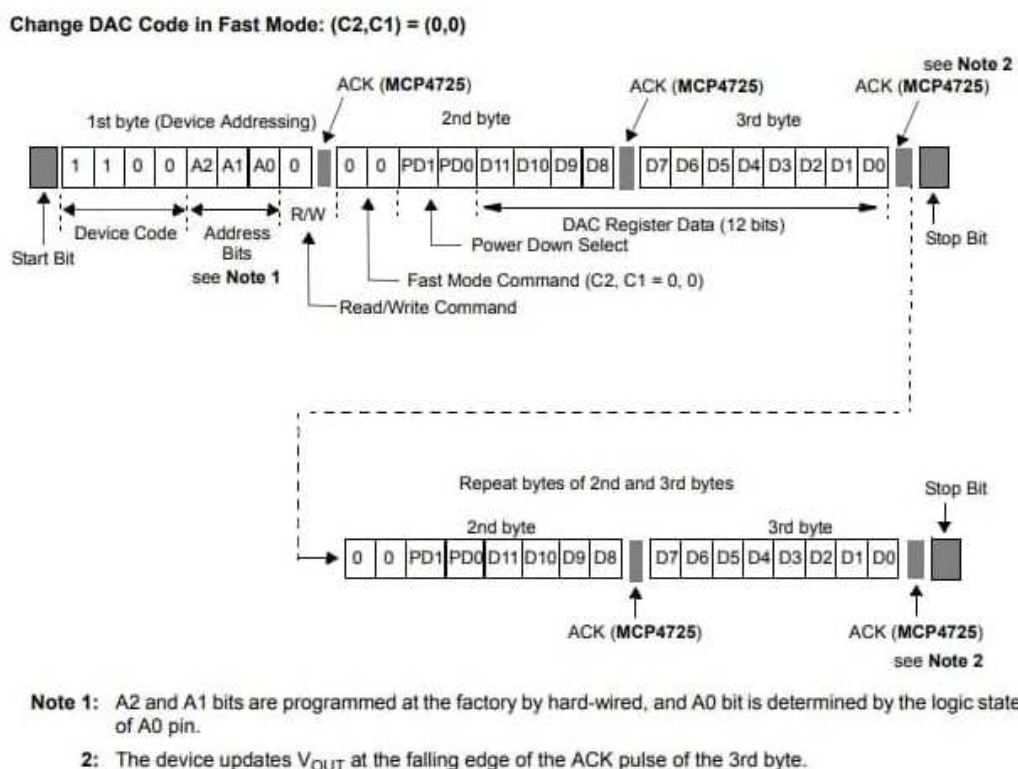


Figura 9: Circuito DAC MCP2725.

4.2. Características

- Resolución de 12 bits
- Interfaz I2C (compatible con estándar, rápida y alta velocidad)
- Compacto
- Alimentación de 2.7V a 5.5V
- EEPROM interna para almacenar la configuración



Para hallar la dirección del DAC MCP4725 desde MATLAB escribimos las instrucciones:

```

1 clear; close all; clc
2 % funciona para DAC Sparkfun como Adafruit
3 a = arduino('COM5');
4 dir_i2c=scanI2CBus(a);
5 disp(['DIR I2C = ', dir_i2c])
6 % {'DIR I2C = '      {'0x60'} Sparkfun
7 % {'DIR I2C = '      {'0x62'} Adafruit

```

El pin de dirección I2C del modulo DAC es accesible para cambiar el bit ADDR/A0 de la dirección del módulo en el bus I2C y así poder usar hasta 2 DACs en un mismo bus I2C. El chip es capaz de operar a 3.3V o 5V, el rango del voltaje de salida depende del voltaje de alimentación: Si es alimentado con 5V el voltaje de salida será de 0 a 5V y de la misma forma para 3.3V.

5. Generador de Señales

Para dar inicio al estudio del generador de señales es importante realizar la conexión del un MCP4725 a Arduino, según el esquema de la Figura 10. Se conecta la alimentación (VCC y GND), el pin SCL del módulo al SCL de Arduino (normalmente el A5), el pin SDA del módulo al SDA de Arduino (normalmente el A4) y el pin de salida (OUT) del módulo a un pin analógico de Arduino, como el A0, si quieres medir el voltaje de salida. Luego, instala la librería Adafruit MCP4725 desde el Administrador de librerías del IDE de Arduino y usa su código de ejemplo para enviar comandos al DAC.

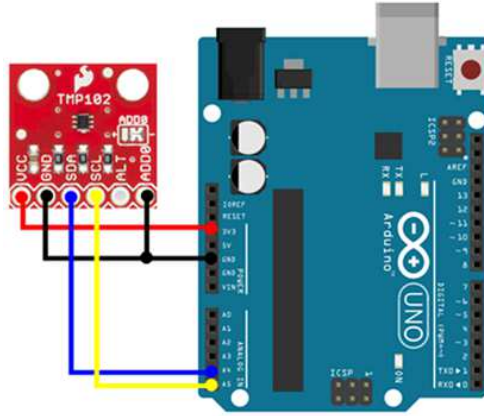


Figura 10: Conectando Arduino con el DAC MCP4725.

El estudio de un generador de señales concierne a un hardware y software necesario para producir por ejemplo una señal seno con cierta amplitud y baja frecuencia.

Ejemplo: Consideremos el esquema de la Figura 10, se desea generar una señal seno que posea los argumento de acurdo con la siguiente ecuación:

$$v(t) = A_1 \sin(2\pi f_c t) + A_2$$

Siendo A_1 la amplitud de la señal, A_2 el offset, f_c la frecuencia análoga, y t el tiempo análogo. Además se conoce las relaciones:

$$t = nT, \quad T = \frac{1}{f_s}$$

Donde f_s es la frecuencia de muestreo en Hz y n el numero de ciclos. Reemplazando valores en $v(t)$ obtenemos:

$$v(nT) = A_1 \sin\left(2\pi f_c \frac{n}{f_s}\right) + A_2 = 2 \sin\left(2\pi 100 \frac{n}{10000}\right) + 2$$

De donde la relación de frecuencias nos proporciona el periodo mínimo de esta señal seno.

$$NS = \frac{f_s}{f} = \frac{10000}{100} = 100$$

Se genera una tabla de 100 valores los cuales son obtenidos en MATLAB:

```

1 k=10000/100;
2 n=0:k;
3 v=2*sin(2*pi*n/100)+2;
4 stem(n,v,'m')

```

En la Figura 11 se muestra la forma de onda senoidal para conformar la tabla de valores.

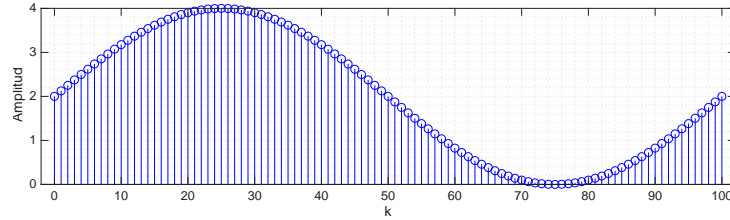


Figura 11: Señales senoidal con STEM.

Caso de Estudio: La resolución de un DAC se calcula generalmente como el voltaje de referencia (V_{ref}) dividido entre el número total de niveles de salida, que es 2^n , donde n es el número de bits. La fórmula para la resolución en voltios es:

$$q = \frac{V_{ref}}{2^n}$$

Por ejemplo, un DAC DAC0808LCN de 8 bits con V_{ref} de 5V tiene una resolución de:

$$q = \frac{5}{2^8} = 19.5 \text{ mV} \approx 0.02 \text{ V}$$

Los valores de la Tabla 1 corresponde a los datos almacenados en orden por filas provienen del código MATLAB a la salida de v .

Tabla 1: Data de valores generados.

2.0000	2.1256	2.2507	2.3748	2.4974	2.6180	2.7362	2.8516	2.9635	3.0717
3.1756	3.2748	3.3691	3.4579	3.5410	3.6180	3.6887	3.7526	3.8097	3.8596
3.9021	3.9372	3.9646	3.9842	3.9961	4.0000	3.9961	3.9842	3.9646	3.9372
3.9021	3.8596	3.8097	3.7526	3.6887	3.6180	3.5410	3.4579	3.3691	3.2748
3.1756	3.0717	2.9635	2.8516	2.7362	2.6180	2.4974	2.3748	2.2507	2.1256
2.0000	1.8744	1.7493	1.6252	1.5026	1.3820	1.2638	1.1484	1.0365	0.9283
0.8244	0.7252	0.6309	0.5421	0.4590	0.3820	0.3113	0.2474	0.1903	0.1404
0.0979	0.0628	0.0354	0.0158	0.0039	0	0.0039	0.0158	0.0354	0.0628
0.0979	0.1404	0.1903	0.2474	0.3113	0.3820	0.4590	0.5421	0.6309	0.7252
0.8244	0.9283	1.0365	1.1484	1.2638	1.3820	1.5026	1.6252	1.7493	1.8744

Valor Digital (VD): Este valor se calculan tomando el entero más cercano de la división entre el valor correspondiente de la tabla y de la resolución.

$$VD = \frac{N}{q}$$

Por ejemplo de la Tabla 1 extraemos el primer elemento, $N = 2$ entonces el valor digital es: $VD = 2/0.02 = 100$. Esto se realiza para todos los valores de la Tabla 1, luego estos valores son los que deben entregarse al DAC según los valores de la Tabla 2.

Tabla 2: Data de valores para el DAC.

100	106	113	119	125	131	137	143	148	154
159	164	168	173	177	181	184	188	190	193
195	197	198	199	200	200	200	199	198	197
195	193	190	188	184	181	177	173	168	164
159	154	148	143	137	131	125	119	113	106
100	94	87	81	75	69	63	57	52	46
41	36	32	27	23	19	16	12	10	7
5	3	2	1	0	0	0	1	2	3
5	7	10	12	16	19	23	27	32	36
41	46	52	57	63	69	75	81	87	94

Estos valores se escriben en el microcontrolador, de modo que se generen los datos de la Tabla 2.

Referencias

- [1] C. Phillips, H. Nagle. *“Control System Analysis and Design”*. Prentice Hall, 1984.
- [2] Sam Fadali. *“Digital Control Engineering Analysis and Design”*. Academic Press is an imprint of Elsevier, 2009.
- [3] Katsuiko Ogata. *“Sistemas de Control en Tiempo Discreto”*. Prentice Hall, 1996.
- [4] Karl. J. Astrom & Bjorn Wittenmark. *“Computer Controller System”*. Prentice Hall, 1997.