

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

CARRERA DE: INGENIERÍA MECATRÓNICA

Materia: CONTROL DIGITAL Y APLICACIONES

Período: Primavera 2013



PRACTICA CON ARDUINO:



Muestreo de una señal de segundo orden.
Arduino-Matlab

Alumno(s):

López García Delfino 200904211
Ramos Solano Lucero 200911508
Saucedo Flores Nathalie 200919674

Profesor: Dr. Jaime Julián Cid Monjaraz.

INDICE

I. Resumen -----	3
II. Introducción -----	3
III. Objetivo -----	3
IV. Marco Teórico -----	3
IV.1.Cálculos -----	5
V. Desarrollo Práctico -----	6
V.1.Observaciones para realización -----	6
V.2.Software -----	6
V.3.Simulación(es) -----	8
VI. Implementación y Pruebas -----	9
VI.1.Definición del experimento -----	10
VII. Conclusiones -----	12
Apéndice Bibliografía y Referencias de la WEB -----	13

I. Resumen

La siguiente práctica muestra una planta con un sistema de segundo orden con un sobretiro del 40% en 4 segundos, la cual será muestreada en Matlab, mediante la adquisición de datos mediante la tarjeta Arduino.

II. Introducción

Es de vital importancia el poder realizar un sistema de segundo orden, en nuestro caso con amplificadores, pero otro punto también importante es el poder adquirir las señales de salida en el tiempo y sobretiro necesarios, es por eso que se ha planteado la elaboración de ésta práctica. Para darle un mayor empleo a la tarjeta Arduino esta vez se usara como adquisición de datos de un sistema, por lo cual se necesita la ayuda de MatLab para mostrar la señal de salida. Cabe mencionar que para la elaboración de todo lo anterior es necesario tener los datos matemáticos para la elaboración e implementación de la planta, todo con la finalidad de obtener un sistema más preciso.

III. Objetivo

Obtener la señal de salida de la planta en la computadora, todo mediante una tarjeta Arduino y MatLab.

IV. Marco Teórico.

Amplificador LM-741

La serie LM741 son amplificadores operacionales de propósito general que ofrecen un mejor rendimiento frente a los estándares industriales. El LM741 es el remplazo directo de los CIs: 709C, LM201, MC1439 y 748 en la mayoría de las aplicaciones. Los amplificadores ofrecen

muchas características que hacen que su utilización sea casi infalible: Protección de sobrecarga en la entrada y la salida, su salida no queda con tensión cuando se excede el rango en modo común, ausencia de oscilaciones. Son idénticos a los LM741/LM741A salvo que el LM741C/LM741E tienen su funcionamiento garantizado en un rango de temperaturas de entre 0 °C a +70 °C, en lugar de -55 °C a +125 °C.

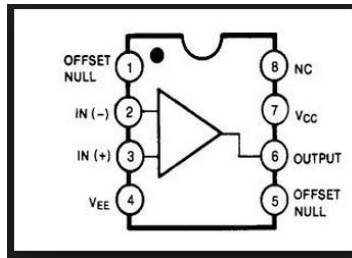


Fig.1 Diagrama LM-741

Sistema de segundo orden.

En ingeniería de control un *sistema de segundo orden* se caracteriza porque tiene 2 polos, la función de transferencia genérica de un sistema de segundo orden en bucle cerrado tiene la siguiente forma:

$K \equiv$ Ganancia

$\delta \equiv$ Factor de amortiguamiento o frecuencia propia no amortiguada

$\omega_n \equiv$ Frecuencia natural

Si sacamos las raíces del denominador observaremos que los sistemas de segundo orden pueden clasificarse en tres tipos diferente de sistemas, las raíces son:

$$s = -\delta\omega_n \pm \omega_n(\delta^2 - 1)^{1/2}.$$

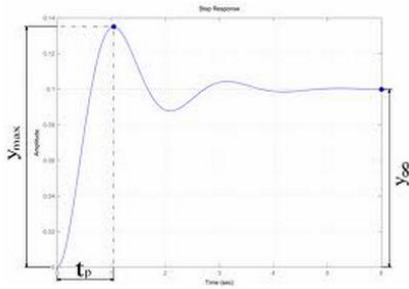


Fig.B Señal estable de un sistema de segundo orden

IV.1.Cálculos

Una vez que se eligen los valores del tiempo y voltaje pico de sobreelongación de nuestro sistema de segundo orden subamortiguado

$$M_p = 40\% \rightarrow 0.4$$

$$T_p = 4 \text{ segundos}$$

Se procede a realizar las siguientes operaciones para encontrar nuestra función de transferencia

$$\zeta = \sqrt{\frac{\ln(M_p)}{\pi^2 + \ln(M_p)}} = \sqrt{\frac{\ln(.4)}{\pi^2 + \ln(.4)}} = 0.31990$$

De donde

$$T_p = \frac{\pi}{\omega_d} \Rightarrow \omega_d = \frac{\pi}{T_p} = \frac{\pi}{4} = 0.7853$$

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}} = \frac{0.7853}{\sqrt{1 - 0.31990^2}} = 0.82895$$

Entonces la función de transferencia es de la siguiente manera

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$H(s) = \frac{0.8289^2}{s^2 + 2(0.3199)(0.8289)s + 0.8289^2} = \frac{0.68717}{s^2 + 0.5303s + 0.68717}$$

$$H(s) = \frac{0.68717}{s^2 + 0.5303s + 0.68717}$$

Ahora bien, se realizan los cálculos para los valores electrónicos:

$$H(s) = \frac{k\omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$$

$$\omega_0 = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}}$$

Si $\omega_0 = \omega_n = 0.8289 \Rightarrow \omega_n^2 = 0.6871$

Y además damos los siguientes valores $R_1 = 50K\Omega$; $R_2 = 500K\Omega$; $C_1 = 10\mu F$, $C_2 = ??? F$

$$C_2 = \frac{\omega_n^2}{R_1 R_2 C_1} = \frac{0.6871}{50K\Omega * 500K\Omega * 10\mu F} = 2.748\mu F$$

Por cálculos, vemos los valores, pero el resto de la práctica es por ensayo-error.

V. Desarrollo Práctico

V.1. Observaciones para realización

- Para poder realizar la práctica es necesario como ya se menciono, elaborar los cálculos matemáticos (en nuestro caso se hizo en MatLab).
- Es importante alimentar la planta de manera correcta, de lo contrario la señal de salida se puede ver alterada.
- Elaborar un código en MatLab que coincida con la cantidad de muestreos mandados desde Arduino, de no ser así la gráfica de salida no arrojará ningún dato.

V.2. Software (con diagramas de flujo o pseudocódigos)

```
clc
disp('Para 4 segundos 40%')
num1=[0 0 0.68717];
den1=[1 0.530368 0.68717];
g1=tf(num1,den1)
%Transfer function:
%      0.6872
%-----
% s^2 + 0.5304 s + 0.6872
plot(num1,den1)
step(g1)
title('Para 4 segundos 40%')
grid on
```

Fig.2 Programa en MatLab para obtener la función de transferencia.

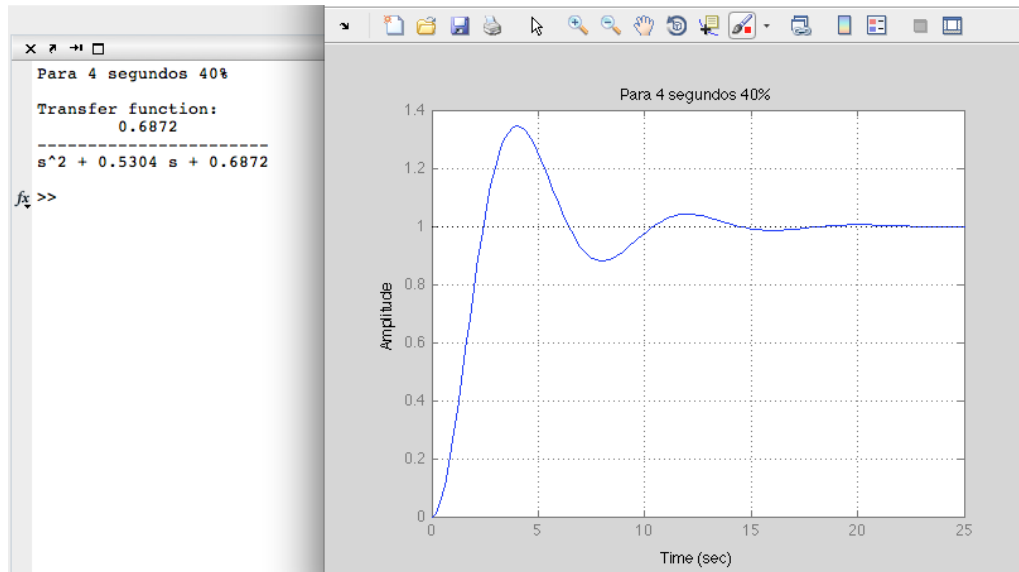


Fig.3 Función de transferencia y señal de salida para el sistema a 40% y 4 seg.

```

int amp=0; //Analog 0

void setup(){
  Serial.begin(9600);
}

void loop(){
  amp=analogRead(amp);
  Serial.println(amp);
  delay(200);
}

```

Fig.4 Programa en Arduino para la obtención de la señal de salida de la planta.

```

numero_muestras=50;

close all;
clc;
y=zeros(1,1000);

delete(instrfind({'Port'},{'/dev/tty.usbmodemfa131'}));
puerto_serial=serial('/dev/tty.usbmodemfa131');
puerto_serial.BaudRate=9600;
warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

fopen(puerto_serial);

contador_muestras=10;

title('Sistema de segundo orden');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;

%Bucle while para que tome y dibuje las muestras que queremos
while contador_muestras<=numero_muestras
    ylim([-2 2]);
    xlim([contador_muestras-20 contador_muestras+5]);
    amp=fscanf(puerto_serial,'%d');
    y(contador_muestras)=(amp(1))/1020;
    plot(contador_muestras,y(contador_muestras),'vr');
    drawnow
    contador_muestras=contador_muestras+1;
end

%Cierro la conexión con el puerto serial y elimino las variables
fclose(puerto_serial);
delete(puerto_serial);
clear all;

```

Fig.5 Código del programa en Matlab.

V.3. Simulación(es)

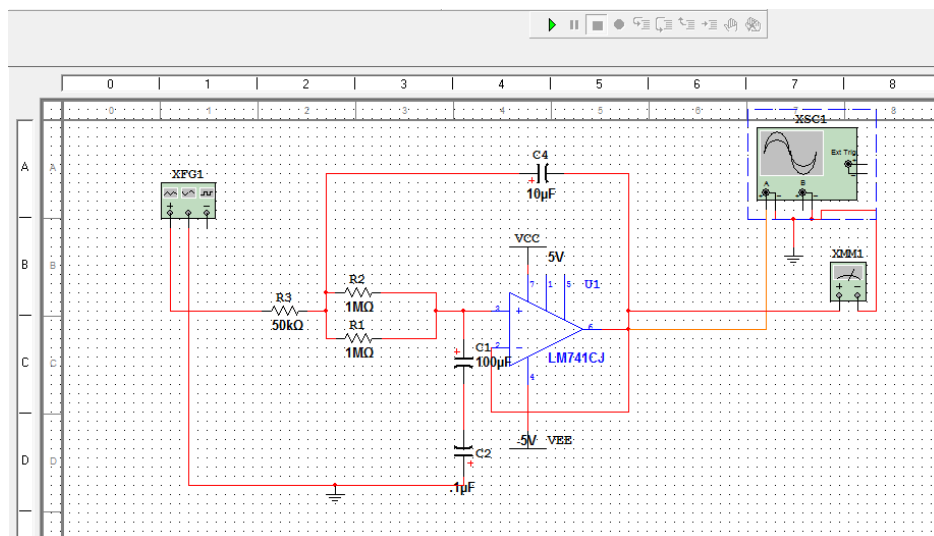


Fig.6 Simulación con Multisim del sistema de segundo orden.

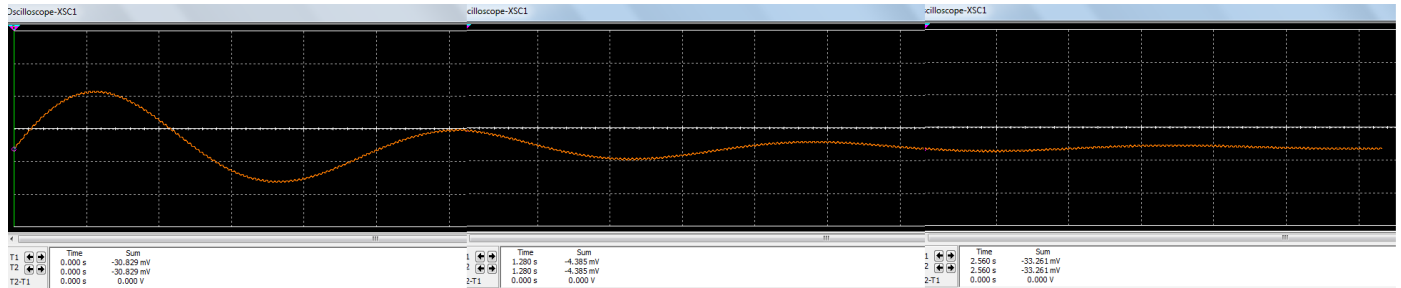


Fig.7 Señal estabilizada del sistema.

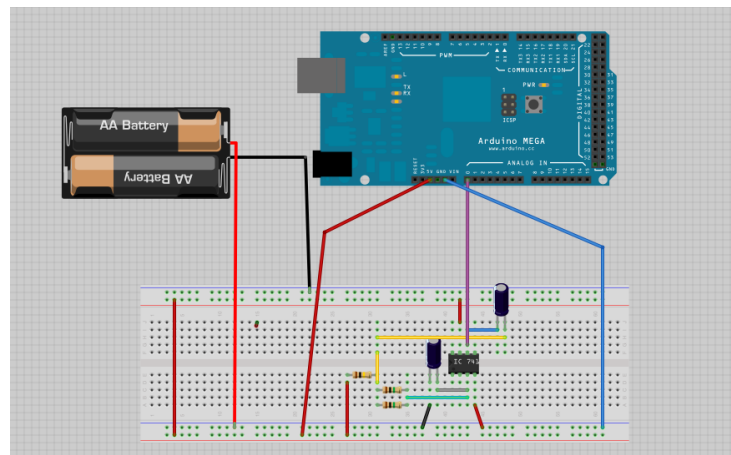


Fig.8 Simulación en fritzing de la conexión con Arduino Mega.

VI. Implementación y Pruebas

VI.1. Definición del experimento

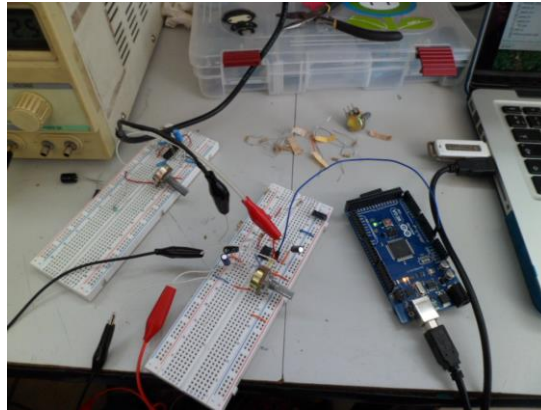


Fig.9 Implementación de la planta.

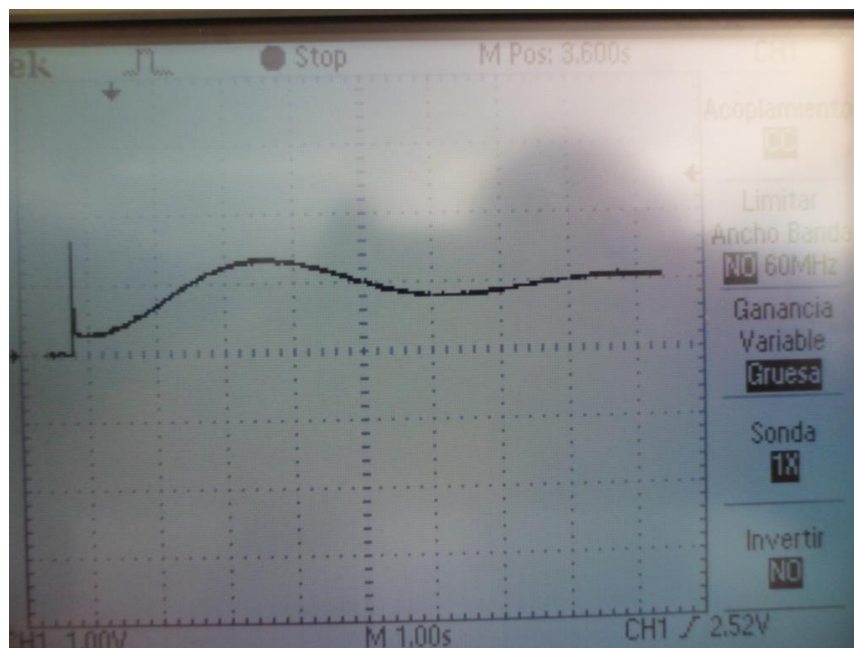


Fig.10 Señal de salida de la planta obtenida con el osciloscopio.

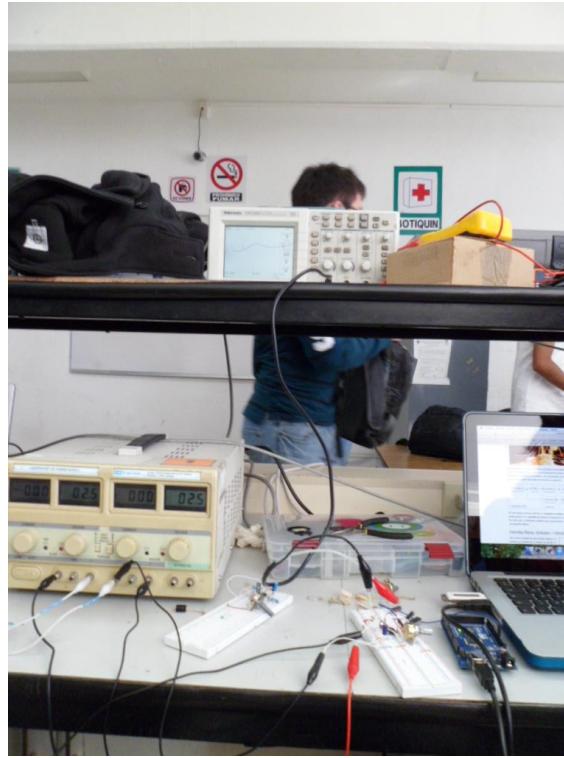


Fig.11 Muestreo de la señal en osciloscopio.

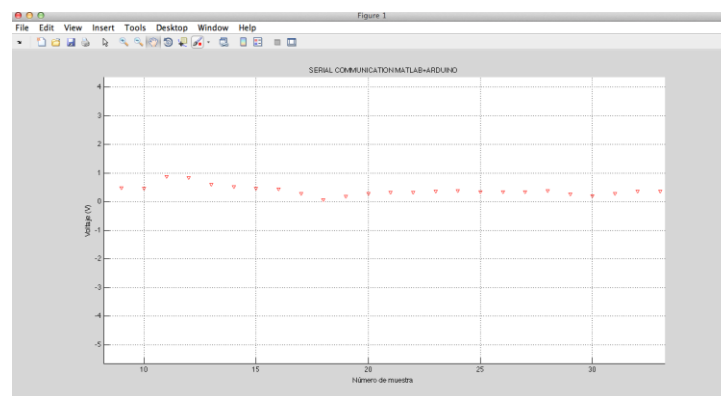


Fig.12 Señal de salida de la planta obtenida con Arduino y MatLab.

VII. Conclusiones

La práctica, en lo que respecta a mi opinión, fue sencilla, ya que una vez obteniendo la función de transferencia requerida, es decir, con voltaje pico de 40% y en 4 segundos, los valores electrónicos, tanto de las resistencias, como de los capacitores, se les tiene que estar buscando, como se dijo, por prueba y error, debido a que en la simulación nos puede, salir, pero nunca es exactamente igual en la vida real. También quiero añadir que una vez que se obtuvo la señal esperada en el osciloscopio, esperábamos que saliera lo mismo en la gráfica de MatLab, pero se tiene que buscar la conversión correcta y el tiempo adecuado en que va adquiriendo los datos.

López García Delfino

Un sistema de segundo orden es fácil de calcular, una vez teniendo o proponiendo el voltaje y tiempo pico, ya que realizamos las operaciones necesarias para obtener el factor de amortiguamiento, de ahí se obtienen otros valores necesarios para obtener la función de transferencia. Una vez que se obtuvieron estos datos, se implementa en el circuito con el OpAmp, y de ahí se procede a mandar esta señal por un programa en MatLab para graficarla, aunque cabe destacar que para mí fue más fácil visualizarlo en el Osciloscopio, ya que en la gráfica teníamos que hacer una pequeña conversión

Ramos Solano Lucero

Al ocupar Amplificadores Operacionales para observar la respuesta en un sistema de segundo orden, con una gráfica subamortiguada es realmente interesante, ya que primero valoramos la función de transferencia, es importante decir que con los valores de la ecuación se ven que resistencias y capacitores debemos implementar, ya que en nuestro caso el valor final del capacitor no fue el mismo al que calculamos, si no que tuvimos que buscarle otro valor, esto lo pudimos observar mediante el uso del osciloscopio. La segunda parte de la practica tuvimos que verificar que la misma señal que salía del osciloscopio, era la misma que veíamos en la gráfica

de MatLab, obviamente hay que definir los tiempos en que se adquieren los datos, para que no halla perdida de los mismos y nos muestre una gráfica suave.

Saucedo Flores Nathalie

Apéndice A. Bibliografía y Referencias de la WEB

- <http://electronica.webcindario.com/componentes/lm741.htm>
- Ingeniería De Control Moderna / 5 ED. O Gata, Katsuhiko