BUAP

# INGENIERÍA EN MECATRÓNICA

## Práctica 4: Control PID digital para un sistema de segundo orden.


Javier Sánchez Juárez 200828837
Jesús Alberto Ordaz Rivera 200813687


Control de Procesos por Computadora
Dr. Jaime Cid Monjaraz

12 de Octubre de 2012



OTOÑO 2012

# 1. RESUMEN

En el presente reporte se aborda el tema para la implementación de un controlador PID digital para un sistema de segundo orden con circuitos operacionales; para después analizar su respuesta con respecto al tiempo, esto mediante el uso de la tarjeta de desarrollo Arduino, la cual se implementa como interfaz con la computadora para obtener de forma gráfica la respuesta del sistema en estudio.

**Palabras clave:** Implementación, controlador, PID digital, sistema de segundo orden, Arduino, respuesta del sistema.

# 2. OBJETIVOS

## 2.1 Objetivo General:

- Implementar un controlador de tipo PID digital para un sistema de segundo orden para después analizar su respuesta con respecto al tiempo por medio del uso de la tarjeta de desarrollo Arduino.

## 2.2 Objetivos Particulares:

- Implementar un sistema de segundo orden mediante circuitos electrónicos, para este caso en particular utilizando circuitos operacionales (pasabajos).
- Encontrar los parámetros de control para la planta.
- Realizar el acoplamiento de señales necesario para obtener los resultados esperados (niveles de voltaje).
- Implementar el controlador de tipo PID digital.
- Realizar la interfaz con la computadora mediante la tarjeta Arduino y MatLab.
- Analizar la respuesta encontrada y representada de forma grafica en MatLab.

# 3. MARCO TEÓRICO

Para el desarrollo de esta práctica necesitamos de algunos conocimientos previos, así como de algunas definiciones para tener buenos resultados al término de esta.

## 3.1 Amplificador Operacional

Un amplificador operacional es un circuito electrónico que tiene dos entradas y una salida. La salida es la diferencia de las dos entradas multiplicada por un factor (G) (ganancia): $Vout = G \cdot (V_+ - V_-)$.

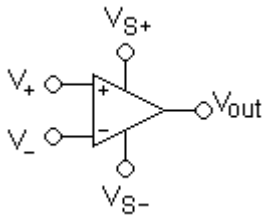



Figura 3.1 Símbolo de un amplificador.

Existen diversos arreglos para un amplificador operacional en los cuales se encuentran:

a) Inversor
b) No inversor
c) Derivador
d) Integrado
e) Sumador
f) Comparador
g) Seguidor

Cada arreglo tienen determinadas características y comportamiento eléctrico en especifico, dependiendo de su configuración y valores RC. Tales valores determinan la ganancia del sistema.

### 3.2 Sistemas de segundo Orden.

Los sistemas de segundo orden son muy comunes en los procesos dinámicos y están asociados principalmente a los sistemas de balance de fuerzas y a los sistemas con interacción y retroalimentación, por lo que un sistema bajo control resulta habitualmente ser de 2º orden.

Un sistema de 2º orden este descrito pos la siguiente ecuación diferencial:

$$\tau^2 \frac{dY^2(t)}{dt^2} + 2\xi\tau \frac{dY}{dt} + Y(t) = K\,u(t)$$

$$\tau = Periodo\ de\ oscilacion\ \ natural$$

$$\xi = Factor\ \ de\ \ Amortiguación$$

$$K = Ganancia\ \ Estatica$$

Si aplicamos Transformadas de Laplace, la correspondiente función de transferencia es

$$\frac{Y(s)}{u(s)} = \frac{K}{(\tau^2 S^2 + 2\xi\tau S + 1)}$$

La respuesta de un sistema de 2º orden depende del valor del factor de amortiguación. Según el valor de $\xi$ se tendrán distintos patrones de comportamiento que se analizaran a continuación.

### 3.2.1 Sistemas Sobre Amortiguados , $\xi \geq 1$

Cuando el factor de amortiguación es mayor o igual a 1 la solución a la ecuación es una combinación de soluciones exponenciales, equivalentes a 2 sistemas de 1º orden en serie.

La respuesta es sin oscilaciones se muestra en la Figura 3.2.1.

A medida que el factor de amortiguación crece, la respuesta se hace más lenta. Cuando el factor de amortiguación vale 1 el sistema se llama "Críticamente Amortiguado". Valores menores hacen al sistema oscilatorio.

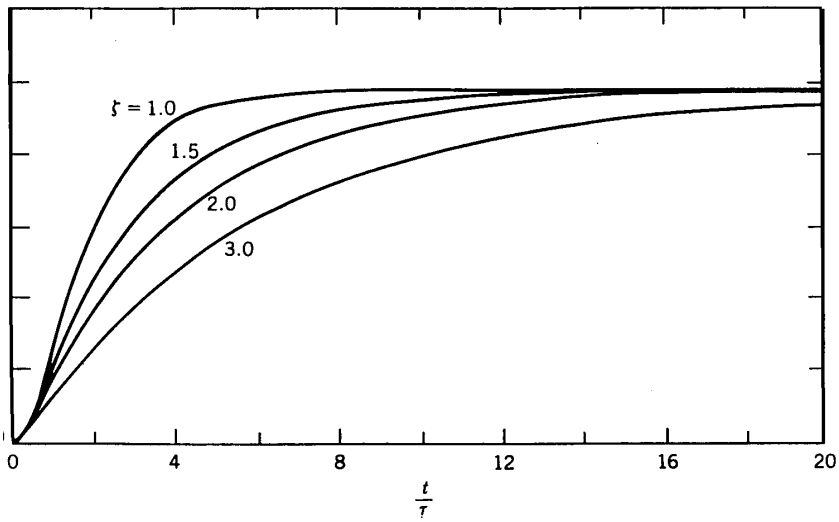



Figura 3.2.1 Respuesta de un sistema sobre-amortiguado

### 3.2.2 Sistemas Sub-Amortiguados, $0 \leq \xi \leq 1$

Cuando el factor de amortiguación está entre 0 y 1 las soluciones presentan componentes imaginarias por lo que la salida es una superposición entre funciones exponenciales y sinusoidales. De esta manera el comportamiento es una señal oscilante con amplitud decreciente que tiende a un valor estable. A este comportamiento se le llama sub-amortiguado.

A modo de ejemplo, la respuesta en el tiempo a un escalón de magnitud A en la entrada esta dada por:

$$Y(t) = K[1 - \frac{1}{\alpha} e^{-\xi/\tau t} \sin(\omega t + \varphi)]$$

$$\alpha = \sqrt{1 - \xi^2} \;\; ; \;\; \varpi = \alpha / \tau$$

$$\varphi = Tan^{-1}(\alpha / \xi)$$

La respuesta temporal se aprecia en la siguiente figura:

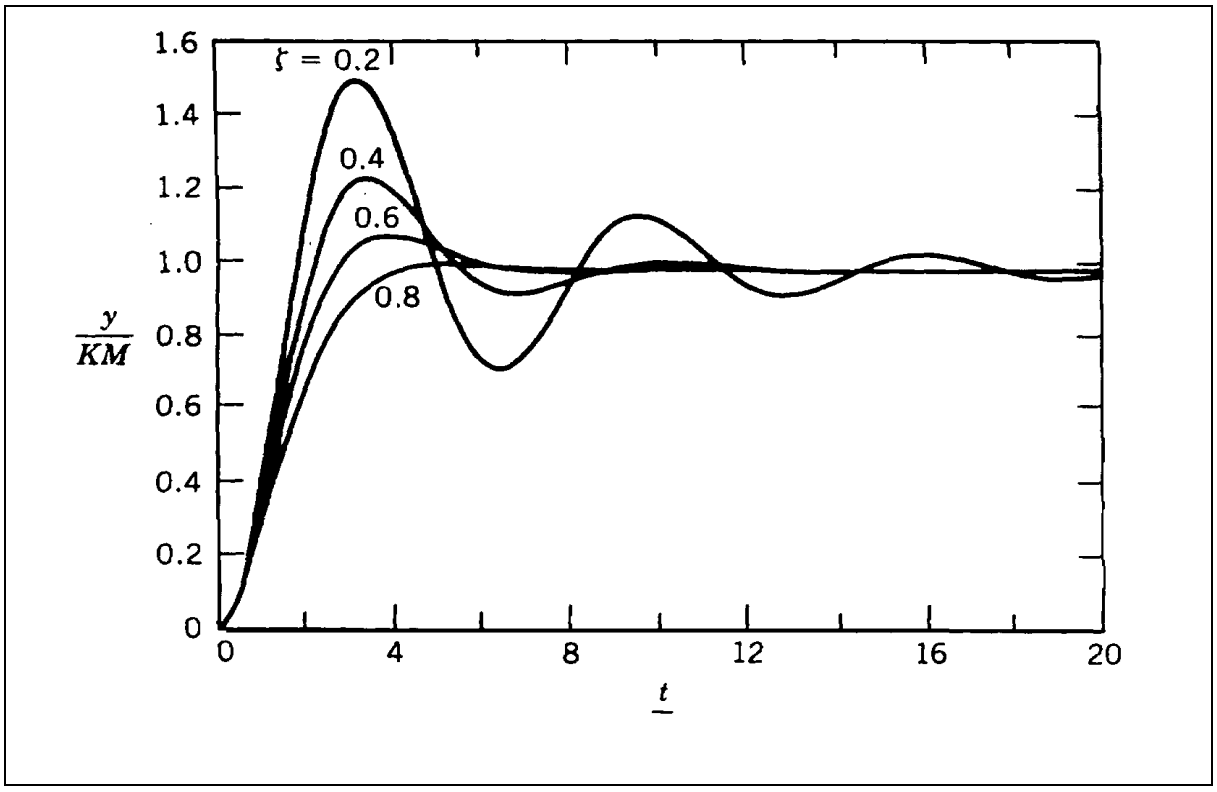



Figura 3.2.2 Respuesta de un sistema sub-amortiguado

Se puede apreciar que mientras más cerca de cero es el factor de amortiguación, la oscilación aumenta y el sistema demora más en estabilizarse. Por otro lado se advierte que la salida presenta valores temporales superiores al valor de estabilización de estado estacionario. Este sobresalto (overshoot) es relevante ya que en algunos procesos no es permitido superar ciertos valores limites.

El comportamiento sub-amortiguado se observa principalmente en los sistemas con control retroalimentado donde las características de oscilación, sobresalto, y tiempo de respuesta son importantes en el desempeño del lazo de control.

Dada la relevancia de este comportamiento es importante caracterizar la respuesta sub-amortiguada de acuerdo a siguiente figura.

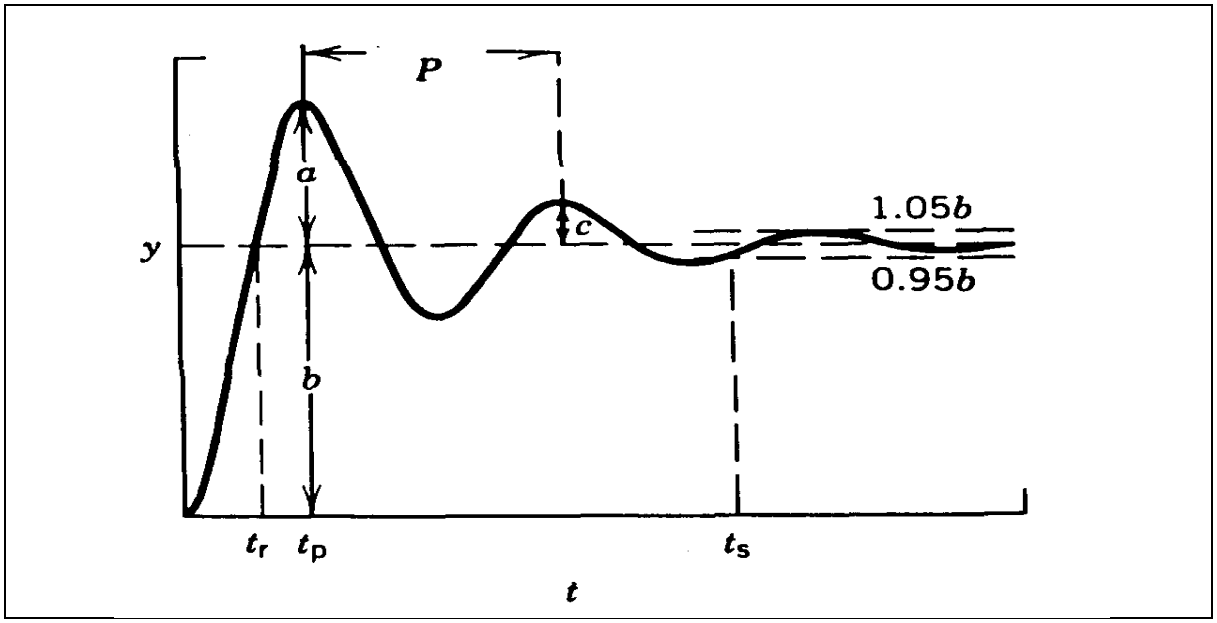



Figura 3.2.2a Caracterización de una respuesta sub-amortiguada

En base a la Figura 3.2.2a se definen los siguientes parámetros de comportamiento:

- Sobresalto (Overshoot) : Medida del sobresalto (peek) en la primera oscilación

$$\frac{a}{b} = \exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right)$$

- Razón de decaimiento : Medida de la amortiguación (ideal 4:1)

$$\frac{c}{a} = \exp\left(\frac{-2\pi\zeta}{\sqrt{1-\zeta^2}}\right) = \frac{a^2}{b^2}$$

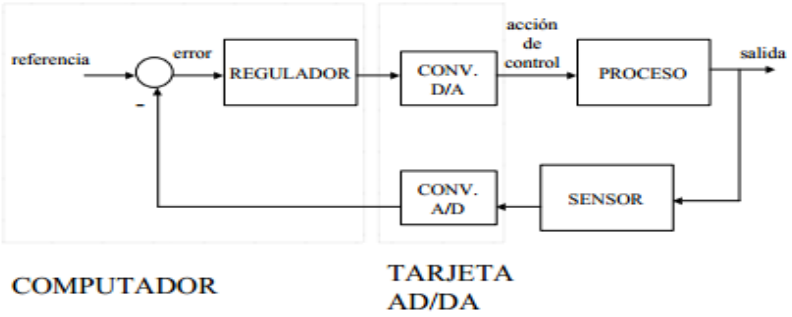
- Tiempo de despegue: Tiempo en cruzar por primera vez el valor de estado estacionario

$$t_p = \frac{\pi\tau}{\sqrt{1-\zeta^2}}$$

- Período de oscilación: Tiempo de una oscilación.

$$p = \frac{2\pi\tau}{\sqrt{1-\zeta^2}}$$

### 3.3 Controlador PID digital.




Podemos obtener un PID discreto a partir de la versión continua, que es:

$$\frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_i s} + sT_d\right)$$

Realizando la sustitución de las s por $\frac{1-z^{-1}}{T}$ y operando, finalmente obtenemos la función de transferencia del PID discreto obtenido por la aproximación de la derivada:

$$\frac{U(z)}{E(z)} = \frac{K\left[1 + \frac{T}{T_i} + \frac{T_d}{T} - \left(1 + 2\frac{T_d}{T}\right)z^{-1} + \frac{T_d}{T}z^{-2}\right]}{1 - z^{-1}}$$

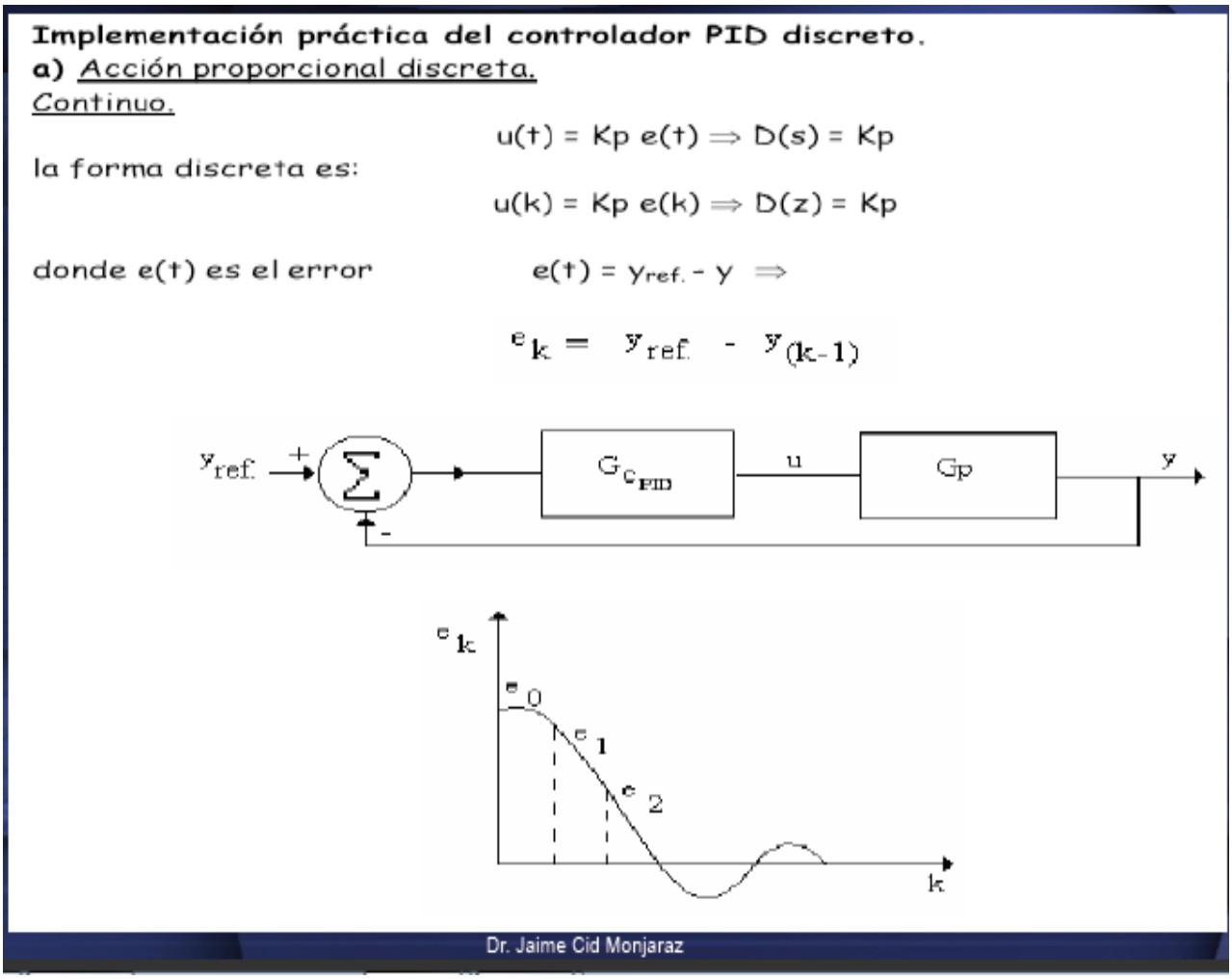
Que pasada a ecuación en diferencias es:

$$u_k = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + u_{k-1}$$

Donde los coeficientes de b valen:

$$
\begin{aligned}
b_0 &= K\left(1+\frac{T}{T_i}+\frac{T_d}{T}\right)\\
b_1 &= -K\left(1+2\frac{T_d}{T}\right)\\
b_2 &= K\frac{T_d}{T}
\end{aligned}
$$

### 3.3.1 Implementación de un controlador PID discreto.



**b)** La acción derivativa discreta.

Continua.

$$u(t) = KpTd\frac{de(t)}{dt} \quad \Rightarrow \quad D(s) = KpTds$$

Discreto.

$$u(k) = Kp\ Td\ (e(k) - e(k-1)) \Rightarrow \quad D(z) = KpTd\ \frac{1-z^{-1}}{h} = KpTd\frac{(z-1)}{hz}$$

otra forma:

$$\frac{u}{E}(z) = Kp\ Td\frac{(z-1)}{hz} \quad \Rightarrow \quad u(K) = \frac{Kp\ Td\ e(k) - e(k-1)}{h}$$

$$u_d(k) = \frac{KpTd}{h}(e(k) - e(k-1))$$

Dr. Jaime Cid Monjaraz

c).- La acción integral discreta. Continuo

$$U(t) = \frac{Kp}{T_i}\int_{\infty}^{t} e(t)dt \quad \Rightarrow \quad D(s) = \frac{Kp}{T_i s}$$

Discreto

$$U(k) = U(k-1) + \frac{Kp\,h}{T_i}e(k) \quad \Rightarrow \quad D(z) = \frac{Kp\,h}{T_i(1-z^{-1})} = \frac{Kp\,h\,z}{T_i(z-1)}$$

otra forma

$$\frac{U}{E}(z) = \frac{Kph(z+1)}{2T_i(z-1)} = \frac{Kph(1+z^{-1})}{2T_i(1-z^{-1})}$$

multiplicando en ambos lados por $(1 - z^{-1})$ y dejando $z^{-1}$

$$U(k) - U(k-1) = \frac{Kph}{2T_i} = (e(k) + e(k-1))$$

$\Rightarrow$

$$U_i(k) = U(k-1) + \frac{Kph}{2T_i}(e(k) + e(k-1)$$

$$U_i(k) = U(k-1) + \frac{Kph}{T_i}e(k)$$

Dr. Jaime Cid Monjaraz

$$PID(k) = u_P(k) + u_I(k) + u_D(k) = P(k) + I(k) + D(k)$$

$$PID(k) = Kp\, e(k) + \frac{Kp\, Td}{h}(e(k) - e(k-1)) + U_I(k-1) + \frac{Kp\, h}{T_I} e(k)$$

$$PID(k) = Kp\, e(k) + \frac{Kp}{h}(e(k) - e(k-1)) + U_I(k-1) + Kp\, h\, e(k)$$

$$U(k) = PID(k)$$

CONTROLADORES DIGITALES PID

Dr. Jaime Cid Monjaraz

Para la implementación del controlador del tipo PID en un Microcontrolador o en este caso en la tarjeta de desarrollo Arduino es necesario tener presente el algoritmo para control:

## Pseudo código de un control digital.

**Inicio**

**inicializar variables y constantes.**

*a1, a2, b1*

*e(n)=0*

*e(n-1)=0*

*u(n)=0*

*u(n-1)=0*

**Capturar el valor de la variable**

*c(nT)=ADC*

**calcular el error:**

*e(n)=r(nT)-c(nT)*

**evaluar la acción de control:**

$m(nT)=a_1*e(n)+a_2*e(n-1)+b_1*u(n-1)$

**Emitir la acción de control**:

*DAC( m(n) + m(n-1) )*

**Actualizar variables:**

*e(n-1)=e(n)*

*m(n-1)=m(n)*

**GOTO** captura valor de la variable

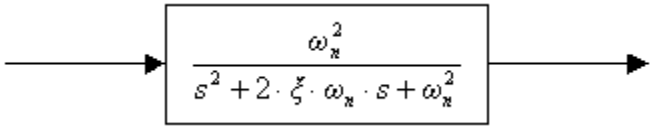
M. C. Jaime Cid Monjaraz

# 4. MATERIAL

- 1 Protoboard.
- Conectores para Arduino.
- 1 Arduino uno.
- 1 Cable USB.
- 3 Amplificador Operacional LM741.
- 1 Resistencia de 10 kΩ.
- 1 Resistencia de 100 kΩ.
- 1 Resistencia de 1 MΩ.
- 4 Resistencias de 330 Ω.
- 1 Capacitor de 90.61 nF.
- 1 Capacitor de 10 µF.
- 1 Fuente de 12 v y -12 v.

## 5. DESARROLLO PRÁCTICO

Para el desarrollo de la práctica es necesaria la implementación de un circuito de segundo orden, para así enseguida analizar su comportamiento y respuesta. Analicemos un sistema de segundo orden.



$$D(s) = s^2 + 2 \cdot \xi \cdot \omega_n \cdot s + \omega_n^2 = 0$$

s raíces de D(s) (polos del sistema) son:

$$s_{1,2} = -\xi \cdot \omega_n \pm \omega_n \cdot \sqrt{\xi^2 - 1}$$

Si $\xi < 1$ las raices son complejas conjugadas :

$$\sigma = \xi \cdot \omega_n \quad \omega_d = \omega_n \cdot \sqrt{1 - \xi^2}$$
$$s_{1,2} = -\sigma \pm j \cdot \omega_d$$

$T = 2 \cdot \xi / \omega_n \rightarrow$ Constante de tiempo
$\omega_n \rightarrow$ Pulsación natural o propia
$\xi \rightarrow$ Constante de amortiguamiento
$\sigma \rightarrow$ Constante de amortiguamiento o factor de decrecimiento
Si $\xi < 1, \omega_d \rightarrow$ frecuencia amortiguada

- Estudio de la respuesta para una señal de excitación tipo escalón unitario.

Señal de excitación:

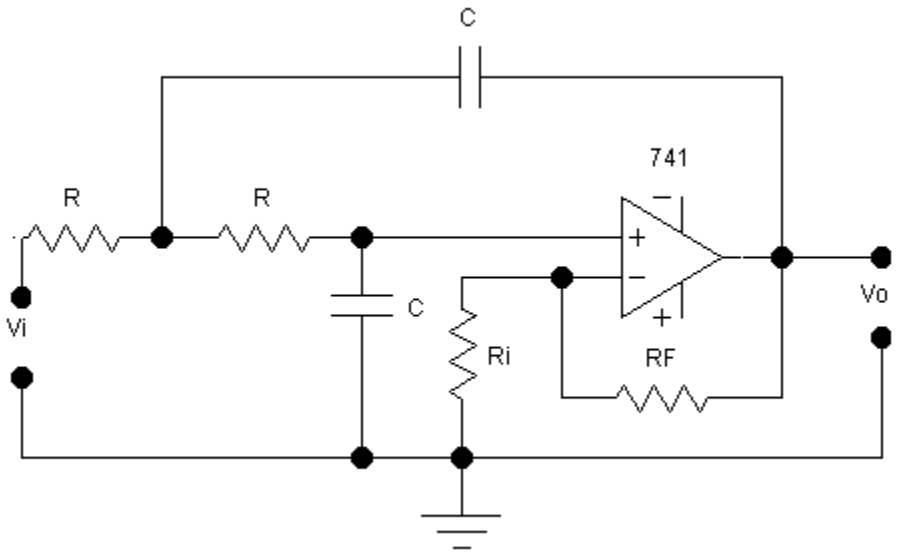
$$X(s) = \frac{1}{s} \text{ (escalón unitario)}$$

Respuesta del sistema:

$$Y(s) = \frac{1}{s} \cdot \frac{\omega_n^2}{s^2 + 2 \cdot \xi \cdot \omega_n \cdot s + \omega_n^2} \Rightarrow y(t) = L^{-1}[Y(s)]$$

### 5.1 Circuito de segundo orden.

Consideremos el siguiente circuito, el cual es la implementación de un pasabajos de segundo grado:




Cuyos valores son los siguientes:

- 1 Resistencia de 10 kΩ.
- 1 Resistencia de 100 kΩ.
- 1 Resistencia de 1 MΩ.
- 1 Capacitor de 90.61 nF.
- 1 Capacitor de 10 µF.

## 5.2 Simulación

Implementamos el circuito de segundo orden como se muestra en la siguiente imagen:

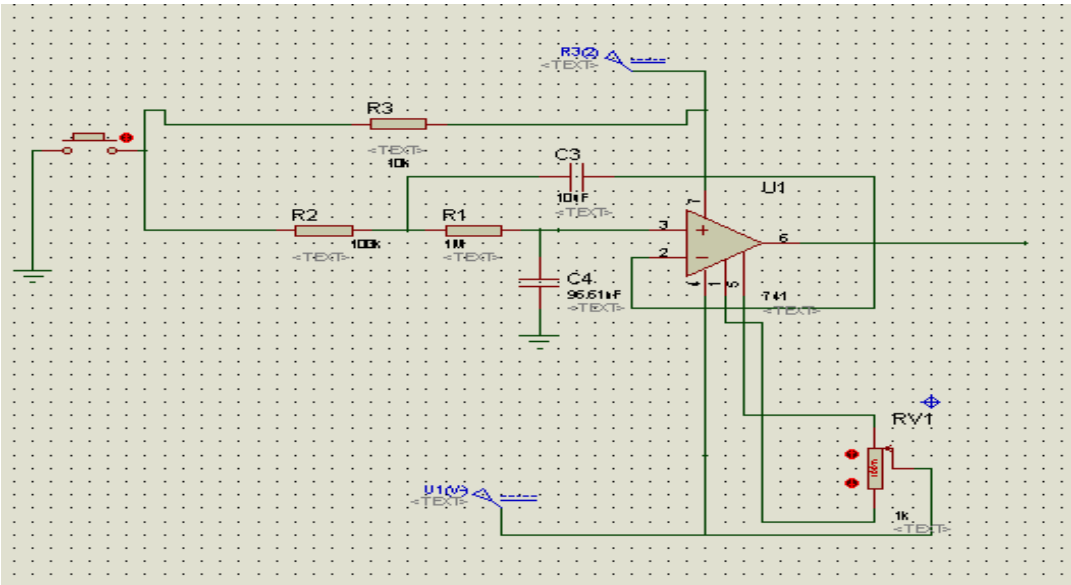



Figura 5.2.1 Implementación del circuito de segundo orden.

Para en seguida proseguir a conectar un osciloscopio, de esta manera podremos visualizar la respuesta de nuestro sistema.

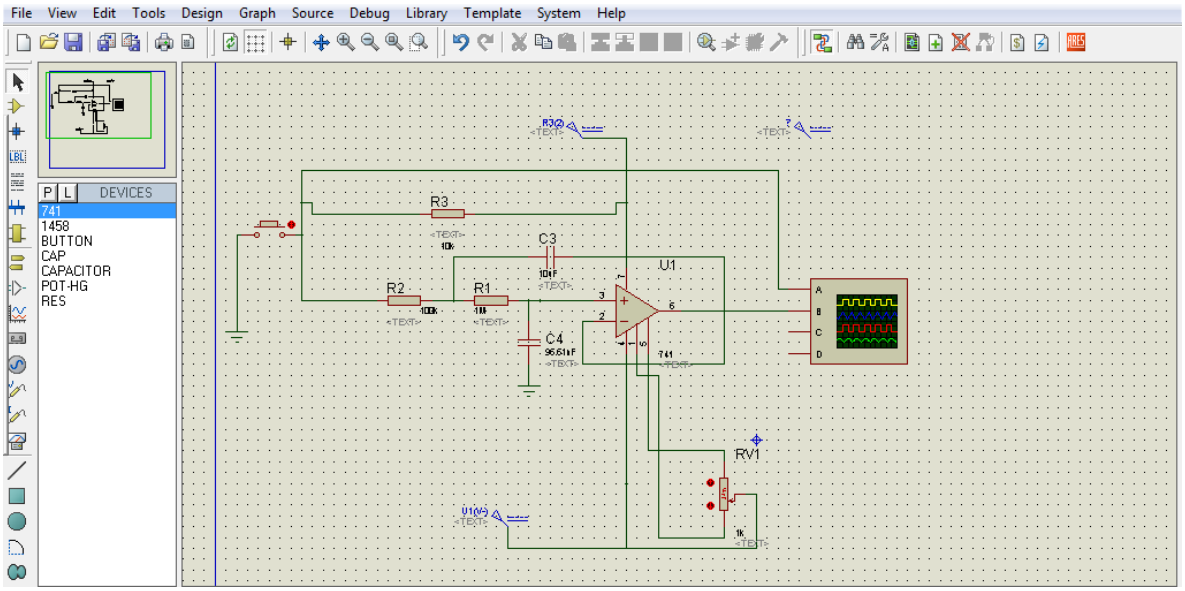



Figura 5.2.2 Implementación para obtener la respuesta de forma gráfica.

Comportamiento del sistema mediante el osciloscopio en la simulación:

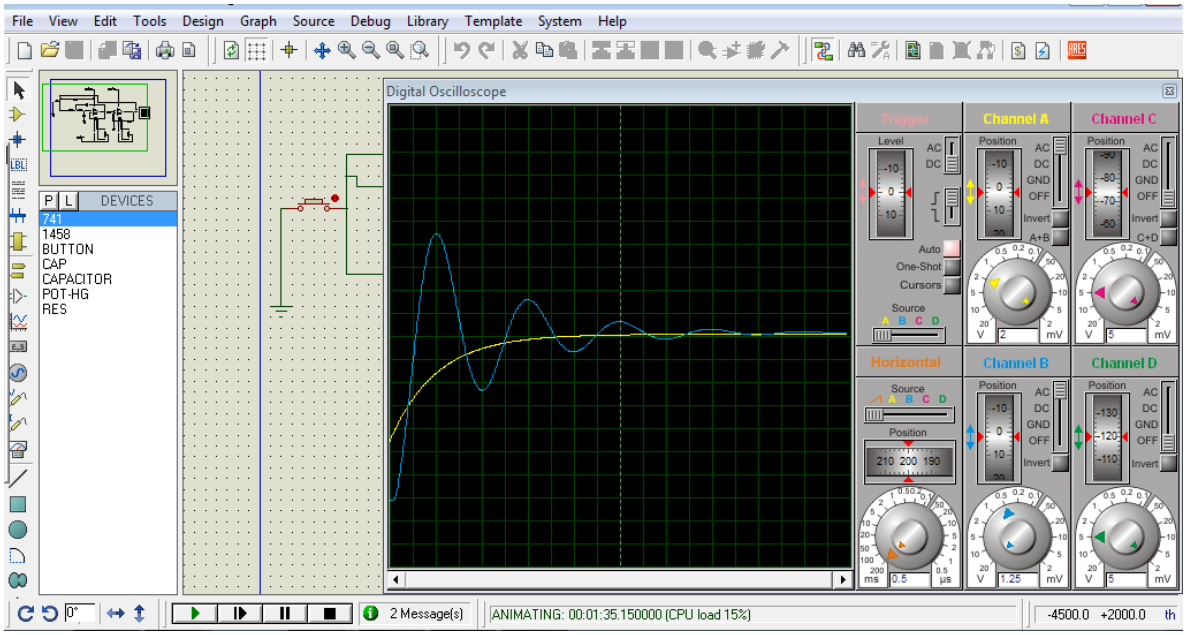



Figura 5.2.3 Simulación de la respuesta del sistema.

A continuación, hemos de programar el Arduino, para que recoja la información de A0 y A1 y la envíe vía serie; el formato del envío consiste en: valor de A0, coma, valor de A1 y retorno de carro + línea nueva (debido a que se usa la función Serial.println() al final).

### 5.3 Código Generado para Arduino

```
//PRACTICA 4 CONTROL PID DISCRETO CON LA TARJETA ARDUINO
// definir variables
int ref = 0;
int sal = 0;
int kp = 2;
int ki = 0;
int kd = 0;
int Tf = 1;
float h = 0.02;
int Tt = 5;
int err = 0;
int eant = 0;
int pidant = 0;

void setup() {
// inicializar puerto serie
Serial.begin(9600);
// leer pines
ref = analogRead(A0);
sal = analogRead(A1);

eant = ref - sal;

int I = 0;
}

void loop() {

delay(20);
// leer pines
ref = analogRead(A0);
sal = analogRead(A1);

err = ref - sal;

int P = kp * err;
int I = I + (err * h);
int D = (err - eant)/ h;

int pid = P + (ki * I) + (kd * D);

pid /= 4 ;

analogWrite(3,pid);

// enviar
```

```
Serial.print(ref);
Serial.print(",");
Serial.println(sal);

eant = err;
}
```

### 5.5 Códigos para MatLab.

Implementación física del controlador PID digital para la planta

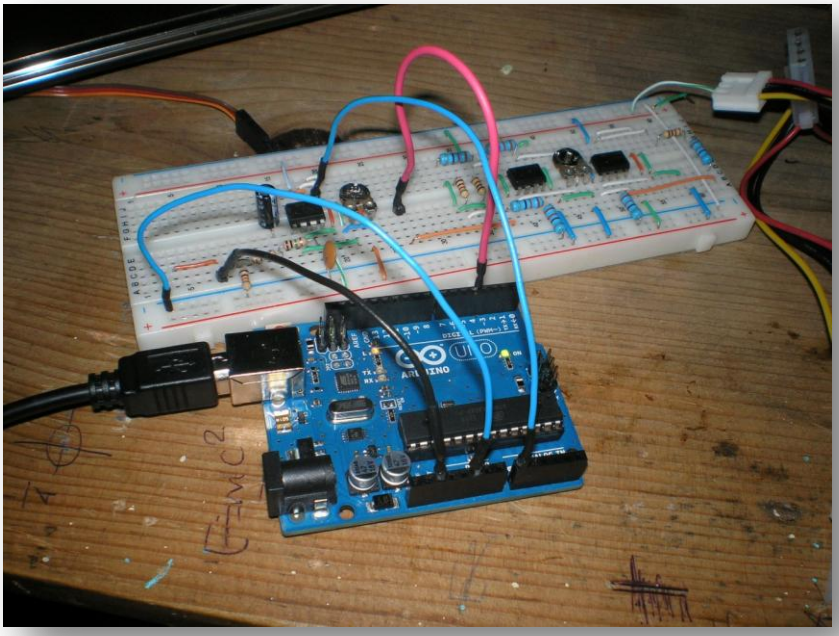


Figura 5.6 Circuito físico del control PID

## 5.6 Implementación de la tarjeta Arduino

A continuación se muestran las conexiones necesarias para la adquisicion de las señales mediante la tarjeta Arduino y Matlab.

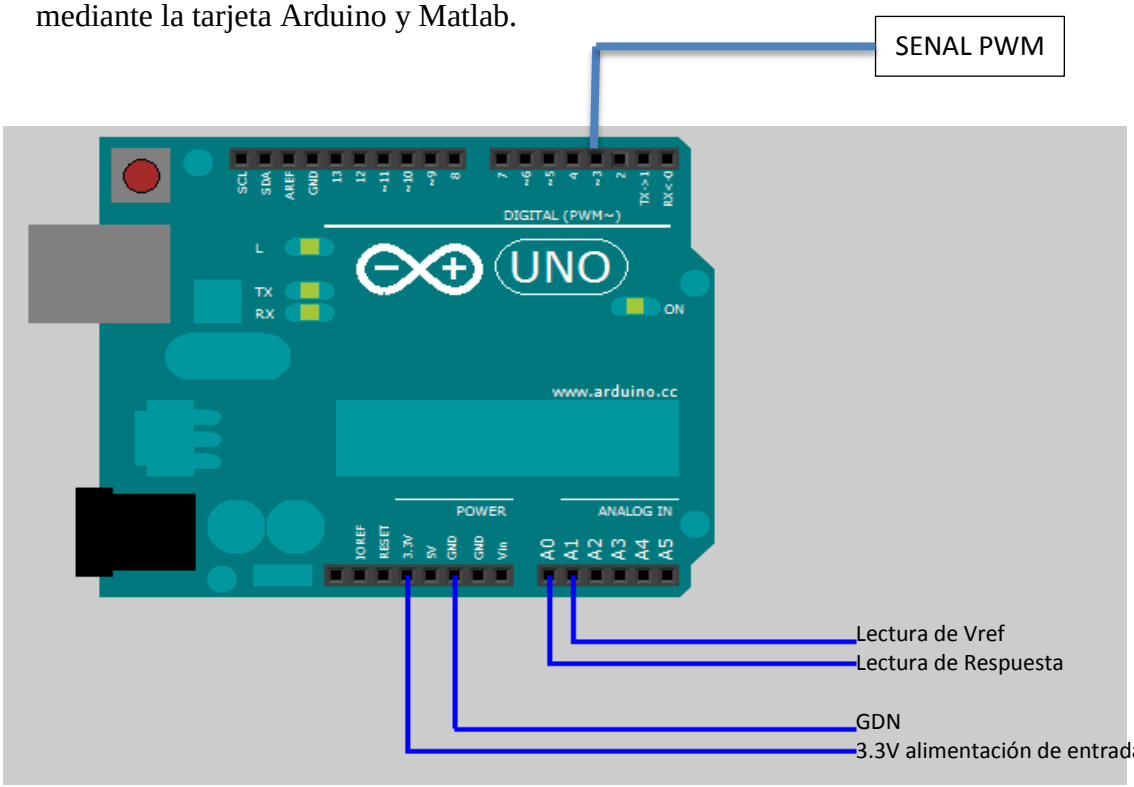



Figura 5.6 Conexiones de la tarjeta de arduino, las conexiones van al circuito implementado de la planta.

## 5.7 Códigos para MatLab.

Una vez configurado el Arduino, los datos se enviarán por la propia conexión USB, pero simulando un puerto serie que deberemos conocer (desde la propia interfaz de Arduino se puede averiguar). En este caso, en el entorno Windows, el puerto será el COM18. El primer paso será crear un objeto serie en Matlab y abrirlo para empezar a leer:

**Programa 0 para MatLab.**

```
%borrar previos
delete(instrfind({'Port'},{'COM18'}));
%crear objeto serie
s = serial('COM18','BaudRate',9600,'Terminator','CR/LF');
warning('off','MATLAB:serial:fscanf:unsuccessfulRead');
%abrir puerto
fopen(s);
```

El siguiente paso es preparar la medida, ajustando dos parámetros: el tiempo total de medida, y la velocidad de capturas por segundo. Éste último parámetro hay que estimarlo, pero haremos que el programa nos devuelva el valor real, con lo que si éste se aleja de lo estimado será inmediato corregirlo.

**Programa 1 para MatLab.**

```
% parámetros de medidas
tmax = 60; % tiempo de captura en s
rate = 33; % resultado experimental (comprobar)
```

A continuación preparamos la figura en la que leeremos la señal de ambos potenciómetros. Abrimos una nueva ventana y unos nuevos ejes, y creamos dos objetos gráficos de tipo línea, que iremos actualizando a medida que tengamos los datos. De esta manera Matlab no se saturará, que es lo que pasaría si intentásemos utilizar la función plot() dentro del bucle.

```
% preparar la figura
f = figure('Name','Captura');
a = axes('XLim',[0 tmax],'YLim',[0 6.1]);
l1 = line(nan,nan,'Color','r','LineWidth',2);
l2 = line(nan,nan,'Color','b','LineWidth',2);

xlabel('Tiempo (s)')
ylabel('Voltaje (V)')
title('Captura de voltaje en tiempo real con Arduino')
grid on
hold on
```

El núcleo del programa es el bucle de medida, en el cual iremos leyendo del puerto serie los datos en el formato que hemos especificado, midiendo el tiempo de ejecución y actualizando los dos objetos línea creados anteriormente: los datos Y serán los voltajes medidos hasta el momento y los datos X el tiempo de ejecución. Al salirse del bucle, imprimiremos el dato de capturas por segundo que hemos estimado.

**Programa 2 para MatLab.**

```
% inicializar
v1 = zeros(1,tmax*rate);
v2 = zeros(1,tmax*rate);
i = 1;
t = 0;

% ejecutar bucle cronometrado
tic
while t<tmax
    t = toc;
    % leer del puerto serie
    a = fscanf(s,'%d,%d')';
    v1(i)=a(1)*5/1024;
    v2(i)=a(2)*5/1024;


    % dibujar en la figura
    x = linspace(0,i/rate,i);
    set(l1,'YData',v1(1:i),'XData',x);
    set(l2,'YData',v2(1:i),'XData',x);
    drawnow
    % seguir
    i = i+1;
end
% resultado del cronometro
clc;
fprintf('%g s de captura a %g cap/s \n',t,i/t);
```

Por último, cerramos el puerto serie (para que otras aplicaciones lo puedan utilizar) y eliminamos el objeto serie que hemos creado en el primer paso.

**Programa 3 para MatLab.**

```
%% Limpiar la escena del crimen
fclose(s);
delete(s);
clear s;
```

## 5.8 Respuesta en MatLab.

En las siguientes figuras se aprecia el programa implementado en MatLab para graficar la respuesta del sistema con la entrada mediante la implementación de la tarjeta Arduino.

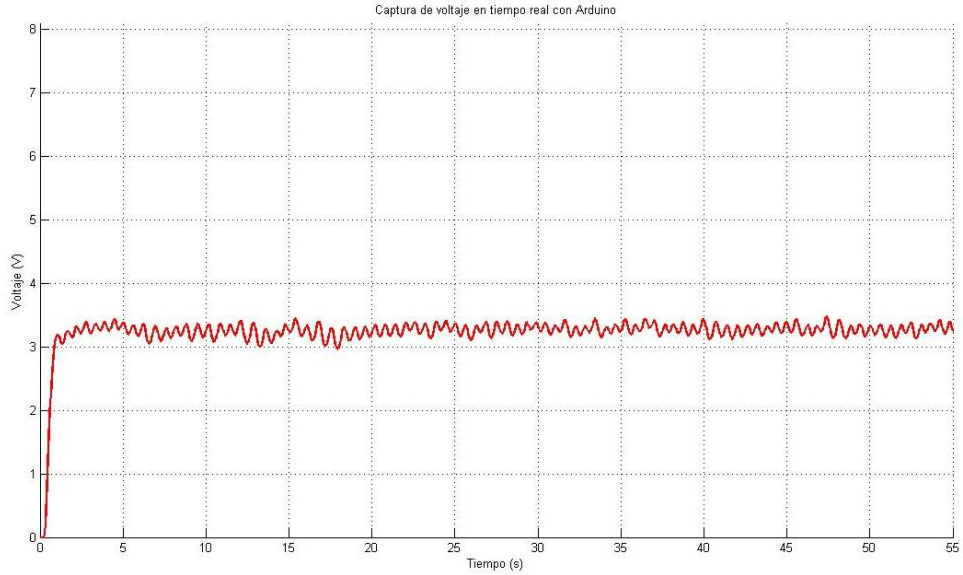



Figura 5.8.1 Respuesta del sistema MatLab usando solo PWM como salida analógica para controlar y realimentar a la planta.

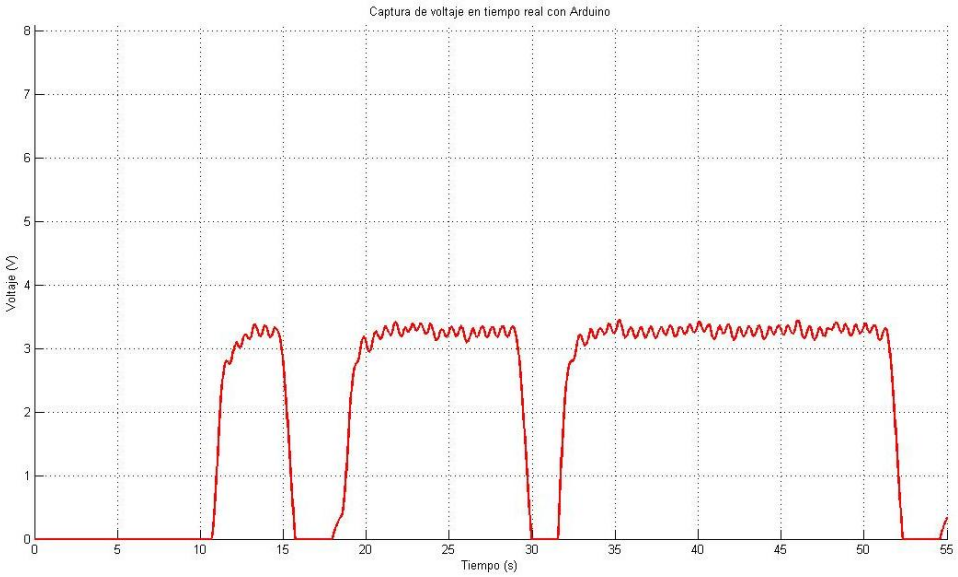



Figura 5.8.2 Respuesta del sistema para la entrada escalón en diferentes tiempos.

## 6. RESULTADOS

Los resultados obtenidos durante la realización de la práctica concuerdan con lo que se esperaba teóricamente de la misma, la implementación de controladores PID digitales reduce en gran medida los componentes necesarios para su implantación, teniendo así una mayor flexibilidad del sistema.

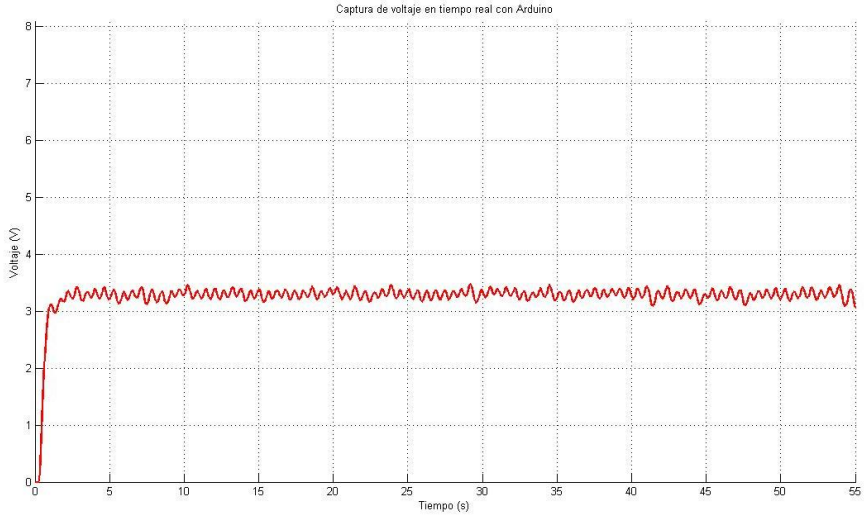



Figura 6.1

Respuesta del sistema a la entrada escalón

En la grafica anterior podemos apreciar que ante una entrada escalón de 3.3 V el controlador PID actúa de manera que el sistema llega a la referencia de manera suave y sin sobrepasos además el tiempo en que llega a la respuesta es menor que en lazo abierto.

Usando una etapa de amplificación de la señal de control se obtiene la siguiente respuesta

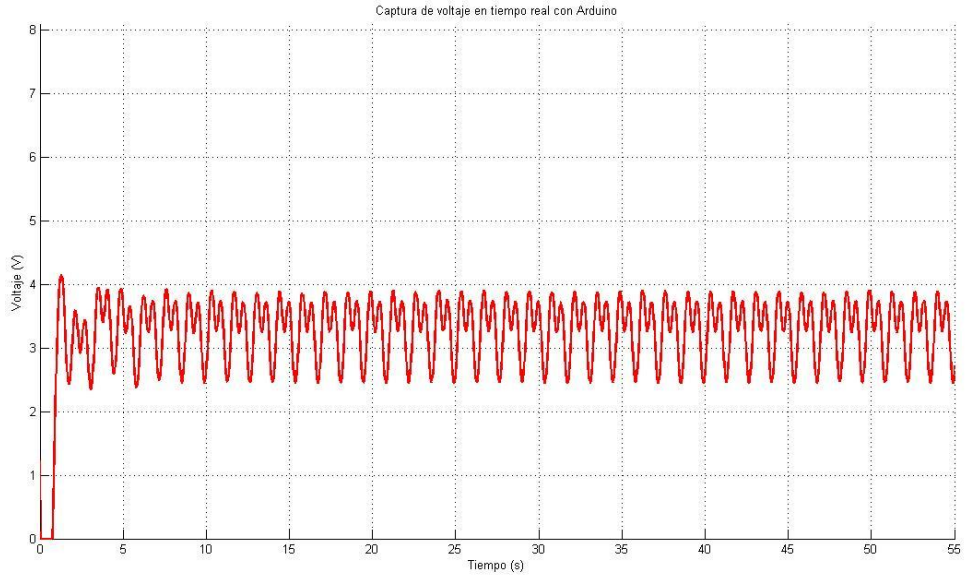



Figura 6.2 Respuesta del sistema a la entrada escalón usando una etapa de amplificación

## 7. CONCLUSIONES

- Los controladores PID permiten mejorar la respuesta de un sistema, aunque esta respuesta no siempre sea optima. Las reglas de ajuste propuestas presentan una forma de obtener los parámetros del controlador PID, siempre y cuando se tenga un modelo matemático del sistema. Un controlador PID permite que la respuesta de un sistema pueda llegar a tener un error nulo.

- Los valores obtenidos a partir de las reglas de ajuste no siempre nos permiten obtener una deseada por lo que los valores deben ser modificados conforme a lo que se desea. Las reglas de ajuste de los controladores PID son convenientes cuando se conoce el modelo del sistema.

- Obteniendo los parámetros de un controlador PID y observando la respuesta del controlador y el sistema, se puede trabajar en un sistema que permita obtener esos parámetros de manera autónoma y así permitir que el controlador PID pueda ser auto-ajustado.

- El uso e implementación de la tarjeta Arduino para el desarrollo de proyectos cada día se convierte en una alternativa más llamativa, ya que por su lenguaje de alto nivel para su programación, se puede realizar aplicaciones muy completas de forma fácil y completa.

# 8. BIBLIOGRAFÍA

Referencias:

Katsuhito Ogata – Ingeniería de Control Moderno- Prentice Hall – 1993

http://wechoosethemoon.es/2011/07/15/arduino-matlab-adquisicion-de-datos/

http://www.monografias.com/trabajos-pdf/sistema-control-digital/sistema-control-digital.pdf