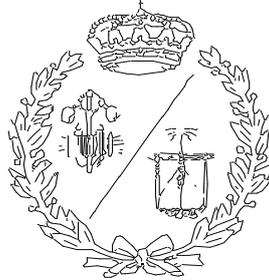


**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Grado

**DISEÑO E IPLEMENTACIÓN DE UN
SISTEMA DE ENTRENAMIENTO PARA EL
CONTROL AUTOMÁTICO DE PROCESOS**

**(DESIGN AND IMPLEMENTATION OF A
TRAINING SYSTEM FOR AUTOMATIC
PROCCES CONTROL)**

Para acceder al Título de

**GRADUADO EN INGENIERÍA ELECTRÓNICA
INDUSTRIAL Y AUTOMÁTICA**

Autor: Rubén Zabala De Cos

Julio - 2020

AGRADECIMIENTOS

En primer lugar quisiera agradecer a mi tutor,
Luciano Alonso Rentería, pues sin él no habría sido
posible llevar a cabo este proyecto, gracias a su ayuda
y su plena disponibilidad para resolver dudas y atenderme.

Por otro lado, y en igual medida, quisiera agradecer
a mi familia, a mi pareja y a mis amigos,
que han sido un pilar fundamental
para poder llevar a cabo el proyecto,
pues me han motivado y ayudado
en todo lo que han podido.

ÍNDICE

1.	INTRODUCCIÓN	11
2.	POTENCIÓMETRO	14
3.	CONTROLADOR	17
3.1.	SEÑAL DE REFERENCIA Y SEÑAL DE ERROR	18
3.2.	TIPOS DE CONTROLADORES	19
3.2.1.	<i>Control Proporcional</i>	20
3.2.2.	<i>Control Proporcional-Derivativo</i>	22
3.2.3.	<i>Control Proporcional-Integrativo</i>	24
3.2.4.	<i>Control Proporcional-Integrativo-Derivativo (PID)</i>	27
3.3.	OBTENCIÓN DE LAS GANANCIAS DEL REGULADOR	28
3.3.1.	<i>Sintonización manual de los parámetros del PID</i>	29
3.3.2.	<i>Método Ziegler-Nichols</i>	31
4.	CIRCUITOS ELECTRÓNICOS	48
4.1.	AMPLIFICADOR OPERACIONAL.....	49
4.2.	CIRCUITO CONVERSOR 0-5 V A ± 5 V	49
4.3.	CIRCUITO CONVERSOR ± 5 V A 0-5 V	54
4.4.	ALIMENTACIONES DEL CIRCUITO	57
5.	MICROCONTROLADOR.....	59
5.1.	ENTORNO Y LENGUAJE DE ARDUINO	62
5.1.1.	<i>Interfaz de la aplicación</i>	62
5.1.2.	<i>Estructura de la programación</i>	64
5.1.3.	<i>Monitor serie</i>	65
5.1.4.	<i>Comandos básicos</i>	65
5.2.	ENTRADAS ANALÓGICAS.....	66
5.3.	PUERTOS DIGITALES.....	68
5.3.1.	<i>Conmutación entre el modo PC o el modo planta</i>	69
5.3.2.	<i>Sentido de giro del motor</i>	70
5.3.3.	<i>Señal PWM</i>	71
6.	PUENTE EN H.....	75
6.1.	ESTRUCTURA	75

6.2.	PUENTE EN H CON TRANSISTORES	76
6.3.	PUENTE EN H CON CIRCUITOS INTEGRADOS.....	77
6.3.1.	<i>Conexiones del driver controlador L298N.....</i>	<i>78</i>
6.3.2.	<i>Alimentación del driver.....</i>	<i>79</i>
7.	MOTOR CORRIENTE CONTINUA.....	81
7.1.	PARTES DEL MOTOR CC.....	82
7.2.	FUNCIONAMIENTO DEL MOTOR CC	83
7.3.	TIPOS DE MOTORES CC.....	84
7.3.1.	<i>Motor serie.....</i>	<i>84</i>
7.3.2.	<i>Motor Shunt o de derivación en paralelo.....</i>	<i>84</i>
7.3.3.	<i>Motor Compound o compuesto.....</i>	<i>85</i>
8.	TRANSMISIÓN	86
8.1.	¿QUÉ SON LOS ENGRANAJES?	86
8.2.	PARTES DE LOS ENGRANAJES.....	87
8.3.	TIPOS DE ENGRANAJES.....	88
8.4.	RELACIÓN DE TRANSMISIÓN	91
8.5.	VENTAJAS E INCONVENIENTES	92
9.	FUENTE DE ALIMENTACIÓN.....	93
10.	IMPLEMENTACIÓN DEL PROYECTO.....	94
10.1.	FUNCIONAMIENTO DEL SISTEMA.....	94
10.2.	FLUJOGRAMA.....	95
10.3.	ALIMENTACIÓN ARDUINO.....	95
10.4.	CONMUTACIÓN ENTRE LOS MODOS PC/PLANTA.....	96
10.5.	LECTURAS ANALÓGICAS	98
10.6.	CONEXIONADO DEL DRIVER MOTOR.....	99
10.7.	CÓDIGO.....	100
10.7.1.	<i>Modo Planta</i>	<i>101</i>
10.7.2.	<i>Modo PC.....</i>	<i>105</i>
10.8.	REPRESENTACIÓN DEL CONEXIONADO DEL SISTEMA.....	108
10.9.	ESTRUCTURA DE LA PLANTA.....	109
11.	ANEXOS.....	111

11.1.	CÓDIGO ARDUINO	111
11.2.	CÓDIGO ZIEGLER-NICHOLS Y APLICACIÓN DEL REGULADOR	113
11.3.	HOJAS DE CARACTERÍSTICAS DE LOS COMPONENTES	120
12.	PRESUPUESTO	121
13.	BIBLIOGRAFÍA.....	124

ÍNDICE DE IMÁGENES

Figura 2-1 Simbología Del Potenciómetro	14
Figura 2-2 Representación Resistencia-Posición	14
Figura 2-3 Terminales Potenciómetro	15
Figura 3-1 Partes Del Controlador.....	18
Figura 3-2 Sistema Ejemplo	19
Figura 3-3 Entrada (amarillo) y salida (azul) del sistema con una ganancia $K_p=1$	20
Figura 3-4 Entrada (amarillo) y salida (azul) del sistema con una $K_p=10$	21
Figura 3-5 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$	21
Figura 3-6 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$ y $K_d=1$	23
Figura 3-7 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$ y $K_d=10$	23
Figura 3-8 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$ y $K_d=100$	24
Figura 3-9 Entrada (amarillo) y salida (Azul) del sistema con una $K_p=10$ y una $K_i=1$	25
Figura 3-10 Entrada (amarillo) y salida (Azul) del sistema con una $K_p=10$ y una $K_i=10$	26
Figura 3-11 Entrada (amarillo) y salida (Azul) del sistema con una $K_p=10$ y una $K_i=40$	26
Figura 3-12 Entrada (amarillo) y salida (Azul) del sistema con los siguientes valores $K_p=K_d=K_i=10$	27
Figura 3-13 Sistema Ejemplo	28
Figura 3-14 Respuesta del sistema con una $K_p=1$	29
Figura 3-15 Respuesta del sistema con una $K_p=1$ y $K_i=1$	30
Figura 3-16 Respuesta del sistema con una $K_p=1$, $K_i=1$ y $K_d=0.25$	31
Figura 3-17 Sistema	32
Figura 3-18 Posibles respuestas del sistema	32
Figura 3-19 Respuesta del sistema ante la entrada escalón	33
Figura 3-20 Representación del parámetro "L" o retardo del sistema.....	34
Figura 3-21 Representación del parámetro "L" y "T"	34
Figura 3-22 Representación polos y ceros	36
Figura 3-23 Sistema	37
Figura 3-24 Respuesta del sistema ante entrada escalón unitario	38
Figura 3-25 Punto de máxima inflexión.....	39
Figura 3-26 Representación de la recta, respuesta y derivada	39
Figura 3-27 Respuesta del sistema ante el regulador proporcional	41
Figura 3-28 Respuesta del sistema ante el regulador Proporcional-Integrativo	42

Figura 3-29 Respuesta del sistema ante el regulador Proporcional-Integrativo-Derivativo ...	43
Figura 3-30 Sistema en lazo cerrado	44
Figura 3-31 Respuesta del sistema críticamente estable.....	44
Figura 3-32 Respuesta del sistema críticamente estable y del periodo crítico.....	45
Figura 3-33 Representación polos y cero.....	47
Figura 4-1 Amplificador Operacional UA741	49
Figura 4-2 Circuito convertidor 0-5 Voltios a ± 5 Voltios	50
Figura 4-3 Primera parte del circuito convertidor 0-5 Voltios a ± 5 Voltios	50
Figura 4-4 Salida (rojo) y entrada (verde) de la primera parte del circuito	51
Figura 4-5 Segunda parte del circuito convertidor 0-5 Voltios a ± 5 Voltios	52
Figura 4-6 Salida (rojo) y entrada (verde) del circuito	53
Figura 4-7 Circuito convertidor ± 5 Voltios a 0-5 Voltios	54
Figura 4-8 Salida (rojo) y entrada (verde) del circuito	57
Figura 4-9 Divisores -5V	58
Figura 4-10 Divisor -2.5V	58
Figura 5-1 Logo Arduino	59
Figura 5-2 Arduino Uno REV3	61
Figura 5-3 Barra de opciones	62
Figura 5-4 Botonera de accesos rápidos.....	63
Figura 5-5 Editor de código.....	63
Figura 5-6 Consola y área de mensajes	64
Figura 5-7 Estructura del programa	64
Figura 5-8 Monitor Serie	65
Figura 5-9 Salidas analógicas Arduino	66
Figura 5-10 Señal digital.....	68
Figura 5-11 Salidas Digitales Arduino	69
Figura 5-12 Conmutador	69
Figura 5-13 Señal PWM.....	73
<i>Figura 5-14 Salidas PWM Arduino</i>	<i>74</i>
Figura 6-1 Puente en H.....	75
Figura 6-2 Estados de funcionamiento del puente en H.....	76
Figura 6-3 Puente en H con transistores.....	76
Figura 6-4 Controlador de motores L298N	77
Figura 6-5 Pines controlador de motores L298N.....	79

Figura 6-6 Alimentación tipo 1 controlador de motores L298N	80
Figura 6-7 Alimentación tipo 2 controlador de motores L298N	80
Figura 7-1 Motor CC.....	81
Figura 7-2 Partes motor CC.....	83
Figura 7-3 Funcionamiento motor CC.....	83
Figura 7-4 Tipos de motores corriente continua	85
Figura 8-1 Partes del engranaje	88
Figura 8-2 Engranajes rectos	88
Figura 8-3 Engranajes helicoidales.....	89
Figura 8-4 Engranajes cónicos.....	89
Figura 8-5 Engranajes exteriores	90
Figura 8-6 Engranajes interiores	90
Figura 8-7 Engranajes de igual número de dientes	91
Figura 9-1 Pin Out fuente ATX	93
Figura 10-1 Alimentación Arduino	96
Figura 10-2 Conexión conmutador.....	97
Figura 10-3 Conexión Señales analógicas	99
Figura 10-4 Conexión control del motor.....	100
Figura 10-5 Conexión alimentación driver.....	100
Figura 10-6 Representación de la recta	106
Figura 10-7 Conexión completo	108
Figura 10-8 Estructura metálica.....	109
Figura 10-9 Estructura okumen	109
Figura 10-10 Estructura física final.....	110

INDICE DE TABLAS

Tabla 1 Valor de las ganancias para los diferentes reguladores a través del primer método Ziegler-Nichols.....	35
Tabla 2 Valor de las ganancias para los diferentes reguladores a través del segundo método Ziegler-Nichols.....	46
Tabla 3 Valor digital de los puertos para determinar el sentido de giro del motor CC	71
Tabla 4 Estado del sistema en función del conmutador	97
Tabla 5 Sentido de giro en función del valor del PID.....	104

1. INTRODUCCIÓN

El proyecto a realizar consistirá en el diseño e implementación de un sistema de entrenamiento para el control automático de procesos, en el cual se podrán tomar medidas a tiempo real, tanto de la entrada, como de la salida del sistema, a través de una serie de conexiones.

El sistema constará de una entrada, la cual estará formada por una circunferencia que indicará los grados de la posición y será accionada manualmente por el usuario, los grados irán desde cero hasta los trescientos sesenta, dicha entrada variará directamente un potenciómetro giratorio, el cual se encontrará alimentado por cinco voltios, con la variación de la tensión al provocar el giro de la entrada, se podrá calcular con una gran exactitud los grados en los que se encuentra. Por otro lado, el sistema constará también de una salida, que tendrá la misma apariencia que la entrada, una circunferencia móvil indicando los mismos grados que la entrada, esta estará accionada por un motor eléctrico de corriente continua, el cual será controlado a su vez por el controlador, el giro de dicho motor será transmitido a otro potenciómetro a través de un conjunto de engranajes, determinando la posición actual de la salida.

En la planta de entrenamiento también se podrá observar tres potenciómetros giratorios, los cuales se corresponderán a cada una de los parámetros del controlador del sistema, es decir, la ganancia proporcional (K_p), la ganancia derivativa (K_d) y finalmente la ganancia integrativa (K_i), cada uno de los potenciómetros estará alimentado por una fuente de cinco voltios, estos se podrán variar y observar la respuesta del sistema.

La planta constará de dos modos, un modo planta, el usuario podrá interactuar directamente sobre el sistema de entrenamiento, variando la entrada del sistema, así como los valores de cada una de las ganancias del regulador, pudiendo observar la respuesta del sistema. Por otro lado, constará de un modo PC, en el cual será necesario la utilización de una tarjeta de adquisición de datos, en este modo se controlará el sistema directamente desde el ordenador, el cual será el encargado de controlar el motor de corriente continua. En ambos se podrá realizar una lectura de los datos de entrada y de salida, con la principal diferencia del lugar de control del motor, vía ordenador o vía planta.

La toma de datos a tiempo real, se podrá realizar a través de unos bornes situados en nuestra planta, en ellos se podrá obtener señales analógicas las cuales estarán comprendidas entre menos cinco y cinco voltios, para poder obtener estas señales se deberá de implementar dos

circuitos electrónicos capaces de convertir señales analógicas de cero a cinco voltios en señales analógicas de menos cinco voltios a cinco voltios, dicho rango de señal es lo ideal para nuestra tarjeta de adquisición de datos, para poder trabajar con el mayor número de bits posible, y obtener lecturas de mayor calidad. Por otro lado, tendremos un circuito electrónico capaz de convertir la señal analógica de menos cinco a cinco voltios que nos va a generar la tarjeta de adquisición de datos, la convertiremos en un rango legible para nuestro microcontrolador, es decir, de cero a cinco voltios, esta señal de entrada será la que aportará el ordenador para controlar nuestro sistema.

1. INTRODUCTION

The project to be carried out will consist of the design and implementation of a training system for automatic process control, in which real-time measurements can be taken, both from the input and the output of the system, through a series of connections.

The system will consist of an entry, which will consist of a circumference that will indicate the degrees of the position and will be manually operated by the user, grades will range from zero to three hundred and sixty, this input will directly vary a rotary potentiometer, which will be powered by five volts, with the variation in tension by causing the inlet to rotate, the degrees in which it is located can be calculated with great accuracy. On the other hand, the system will also consist of an output, which will look the same as the input, a moving circumference indicating the same degrees as the input, it will be driven by an electric motor with direct current, which will be controlled in turn by the controller, the rotation of that motor will be transmitted to another potentiometer through a set of gears, determining the current position of the output.

Three rotary potentiometers can also be observed at the training plant, which will correspond to each of the system driver parameters, i.e, proportional gain (K_p), derivative gain (K_d) and finally integrative gain (K_i), each of the potentiometers will be powered by a five-volt source, these can be varied and the system response observed.

The plant will consist of two modes, a plant mode, the user will be able to interact directly on the training system, varying the system input, as well as the values of each of the gains of the regulator, being able to observe the response of the system. On the other hand, it will consist of a PC mode, in which it will be necessary to use a data acquisition card, in this ode you will be controlled directly from your computer, which will be in charge of controlling the DC motor. In

both cases will be able to read the input and output data, with the main difference of the engine control place.

Real-time data storage can be done through terminals located on our plant, analog signals can be obtained in which they will be between less than five and five volts, in order to obtain these signals, two electronic circuits must be implemented, capable of converting analog signals from zero to five volts into analog signals of less than five volts to five volts, that signal range is ideal for our data acquisition card, to be able to work with as many bits as possible, and get higher quality readings. On the other hand, we will have an electronic circuit capable of converting the analog signal from less than five to five volts that will generate the data acquisition card, we will turn it into a readable range for our microcontroller, that is, from zero to five volts, his input signal will be the one that the computer will provide to control our system.

2. POTENCIÓMETRO

Los potenciómetros serán una parte fundamental de nuestro sistema, se encontrarán, en la entrada del sistema, donde el usuario variará directamente su posición, en la salida, pilotado por un conjunto de engranajes que transmiten el movimiento del eje del motor, y finalmente se utilizarán potenciómetros para determinar los valores de las ganancias del regulador.

Un potenciómetro [1] tiene un funcionamiento sencillo, consta de un cursor el cual tiene un contacto deslizante, dicho contacto deslizante se desplaza sobre una tira resistiva, determinando la capacidad resistiva de nuestro potenciómetro. Por otro lado, tiene dos terminales por los cuales se alimentará el sistema, en nuestro caso, alimentaremos los terminales con 5 voltios y 0 voltios, en la *Figura 2-1*, podemos observar los terminales del potenciómetro, así como su simbología.

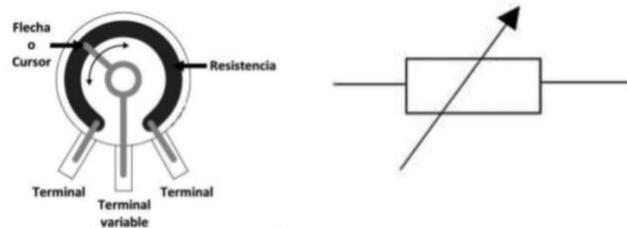


Figura 2-1 Simbología Del Potenciómetro

La resistencia de los potenciómetros será proporcional al ángulo de giro que le aportemos, la respuesta del sistema corresponde con la *Figura 2-2*.

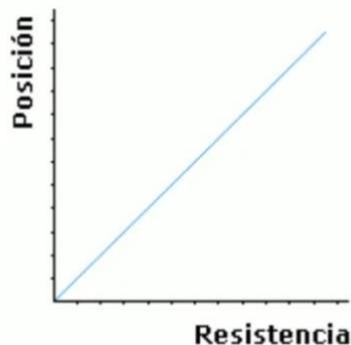


Figura 2-2 Representación Resistencia-Posición

El potenciómetro está compuesto por tres terminales, como se puede observar en la *Figura 2-3*, el terminal 1 y el terminal 3 corresponden a los bornes de alimentación, por otro lado, el terminal 2, corresponde a la tensión de salida.

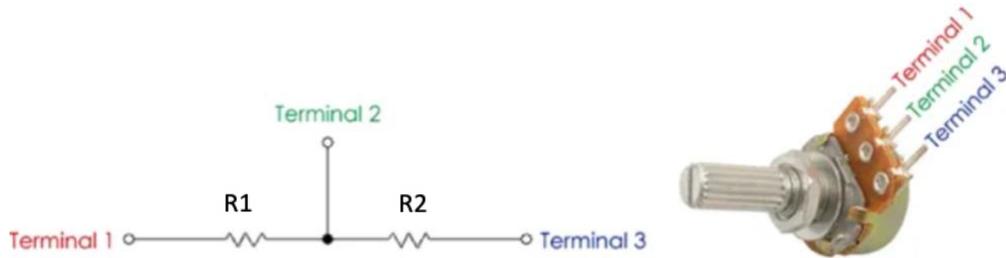


Figura 2-3 Terminales Potenciómetro

Se procede a calcular la tensión de salida en función del valor de la resistencia del potenciómetro, para ello aplicamos nudos en el sistema de la *Figura 2-3*, obteniendo la ecuación 2.1, una relación de la señal de entrada (V_{in}), con el valor de las resistencias (R_1 y R_2) y la tensión de salida (V_{out}).

Donde:

- Terminal 1 = 5 Voltios = V_{in}
- Terminal 2 = V_{out}
- Terminal 3 = 0 Voltios

$$\frac{V_{in} - V_{out}}{R_1} = \frac{V_{out}}{R_2}$$

(2.1)

$$\frac{V_{in}}{R_1} = V_{out} \left(\frac{R_1 + R_2}{R_1 * R_2} \right)$$

Finalmente obtenemos el valor de la tensión de salida (V_{out}), en función de la posición del cursor, como se puede observar en la ecuación 2.2.

$$V_{out} = V_{in} * \left(\frac{R_2}{R_1 + R_2} \right)$$

(2.2)

POTENCIÓMETRO

En nuestro proyecto se van a utilizar cinco potenciómetros, tres de los cuales van a estar dedicados a cada una de las ganancias del regulador (K_p, K_i y K_d), estos potenciómetros tendrán una resistencia máxima de $4.7K\Omega$, se tratan de potenciómetros giratorios de hasta 300° de giro. Por otro lado, los otros dos potenciómetros restantes se utilizarán para la entrada y la salida del sistema, estos se tratan de potenciómetros giratorios, continuo-vueltas, es decir, tienen la capacidad de girar sobre sí mismo, como si de un tornillo sin fin se tratase, estos potenciómetros tendrán una resistencia de $2 K\Omega$.

3. CONTROLADOR

Un controlador [2] es un dispositivo que permite controlar un sistema en lazo cerrado, en el tendremos un conjunto de ganancias, la ganancia proporcional (K_p), la cual será multiplicada directamente por el error, por otro lado, la ganancia integrativa, la cual será multiplicada por la integral del error y finalmente la ganancia derivativa (K_d), la cual será multiplicada directamente por la derivada del error.

Al variar los diferentes valores de las ganancias podremos obtener diferentes tipos de controladores: controladores proporcionales, aquellos controladores en los que únicamente existe la ganancia proporcional, el resto de ganancias es cero, por otro lado, tenemos los controladores proporcional-derivativo, donde existe y una ganancia proporcional y una ganancia derivativa, mientras que la ganancia integrativa es igual a cero, también existe el controlador proporcional-integral, en el cual será la ganancia proporcional y la integrativa las que se utilizarán para controlar el sistema y finalmente el controlador proporcional-integral-derivativo, comúnmente conocido como PID, en el cual las tres ganancias serán utilizadas, en busca del mejor controlador posible.

A continuación, podemos observar el diagrama de bloques de un controlador en la *Figura 3-1*, en el se pueden observar las diferentes partes del controlador, la parte proporcional, representada con el color amarillo, la parte integrativa, representada con un color verde claro, y la parte derivativa la cual viene representada por un color naranja. La señal del controlador es la suma de cada una de las partes, en función del valor de cada una de las partes, formaremos los distintos tipos de controladores. El controlador obtenido se aplicará al proceso y obtendremos una señal de salida.

En la *Figura 3-1* podemos observar el controlador con las tres partes que lo forman, la parte proporcional, la parte integrativa y finalmente la parte derivativa, este ejerce un control sobre el proceso obteniendo una señal de salida (Output), esta señal de salida, a través de una realimentación se le resta a la señal de entrada del sistema (Setpoint), obteniendo el error, pudiendo aplicar una vez más el controlador.

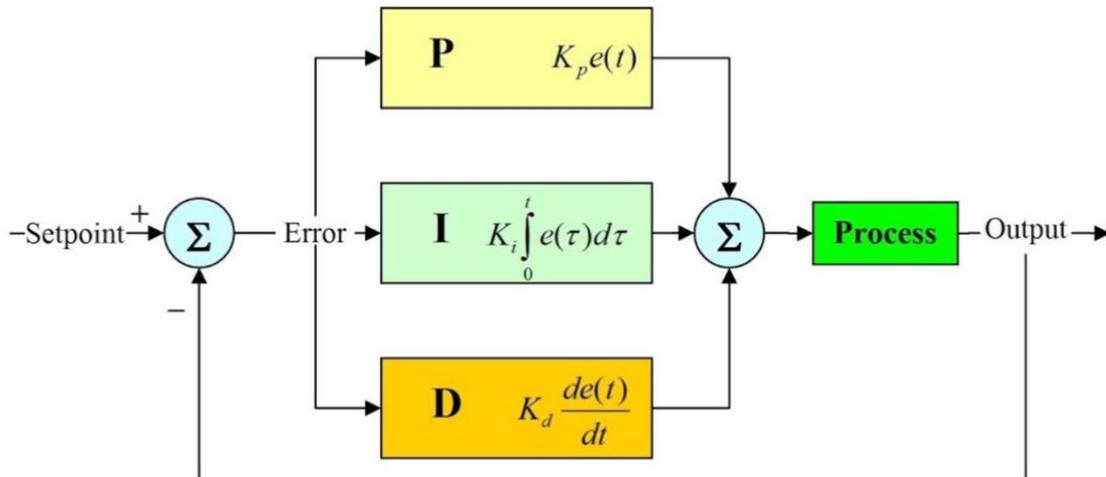


Figura 3-1 Partes Del Controlador

El controlador viene definido por la ecuación 3.1.

$$Controlador(t) = Kp * Error(t) + Ki * \int Error(t) * dt + Kd * \frac{dError(t)}{dt} \tag{3.1}$$

3.1. SEÑAL DE REFERENCIA Y SEÑAL DE ERROR

La señal de referencia, denominada en la *Figura 3-1* como "Setpoint", indica la señal de entrada del sistema, a la cual el sistema deberá de responder con la mayor similitud posible, para reducir el error considerablemente o anularlo directamente, por otro lado podemos obtener reacciones completamente contrarias, generando un sistema inestable o un aumento del error. En nuestro sistema la señal de entrada vendrá directamente definida por el potenciómetro de entrada el cual giraremos a través de una ruleta, seleccionando la posición de nuestro sistema.

Una de las partes más importantes en la creación de cualquier tipo de controlador, ya sea proporcional, proporcional-derivativo, el propio PID, entre otros, es la señal del error, pues aparece en las diferentes partes que forman nuestro regulador, unas veces directamente como error, otras veces como su derivada y otras como su integral, para así conformar las diferentes partes, la parte proporcional, la integrativa y la derivativa.

El error [3] consiste en la diferencia que existe entre la entrada del sistema y la salida del sistema, se verá variado en función de los parámetros que utilizemos a la hora de generar el controlador, un controlador ideal es aquel en el cual la entrada y la salida del sistema es la misma, es decir, el error es igual a cero.

La ecuación 3.2, representa el error, la diferencia entre la señal de entrada (Setpoint) y la señal de salida (Output).

$$Error(t) = Setpoint(t) - Output(t) \tag{3.2}$$

Si la señal del error es grande, significa que la salida se encuentra muy distante de lo que se ha aplicado a la entrada, por lo que, el sistema es muy poco preciso, por el contrario si el error es pequeño o cero, significa que el sistema ha obtenido un estado deseado y se trata de un sistema mucho más preciso.

3.2. TIPOS DE CONTROLADORES

En este apartado se procederá a realizar el análisis sobre el sistema de la *Figura 3-2*, que utilizaremos como ejemplo, este sistema viene definido por la función de transferencia 3.3.

$$Process(s) = \frac{1}{s^2 + 3 * s + 2} \tag{3.3}$$

A continuación, aplicamos una entrada escalón unitario y obtenemos el siguiente sistema.

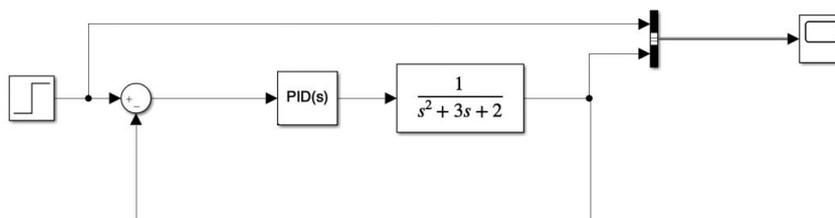


Figura 3-2 Sistema Ejemplo

3.2.1. Control Proporcional

Esta parte del controlador viene definida por el valor del error (Error) y la ganancia proporcional del sistema (K_p), se obtiene a través de la multiplicación de ambos, obteniendo la acción del controlador 3.4.

$$\text{Controlador}(t) = K_p * \text{Error}(t) \quad (3.4)$$

Este tipo de control intenta minimizar el error del sistema. Cuando el error es grande se ejercerá un control elevado, tendiendo a minimizar lo máximo posible el error.

Al aumentar el valor de la ganancia proporcional (K_p) varía la salida del sistema, generando los siguientes efectos:

- Aumenta la velocidad de respuesta del sistema.
- Disminuye el error del sistema en régimen permanente.
- Aumenta la inestabilidad del sistema, llegando a generar un sistema inestable.

En el primer ejemplo aplicaremos únicamente una ganancia proporcional igual a uno ($K_p=1$), como se puede observar en la *Figura 3-3*. La señal amarilla se corresponde con la entrada escalón unitaria aplicada, la señal de azul es la respuesta del sistema.

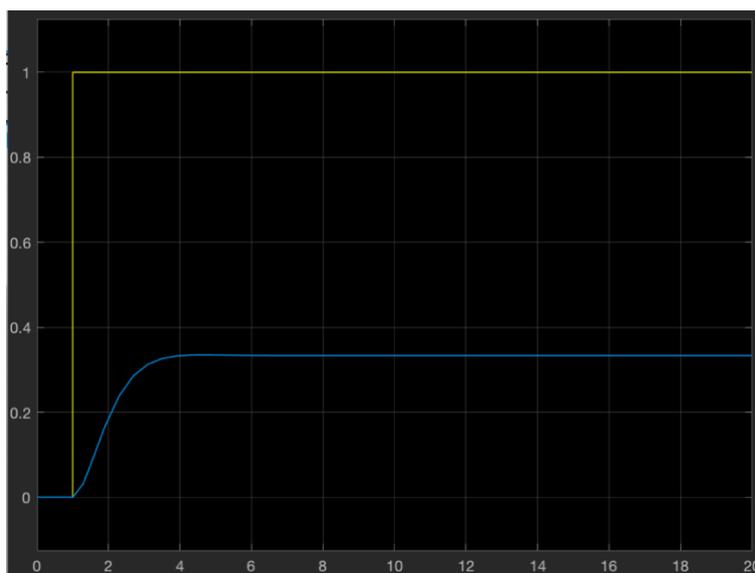


Figura 3-3 Entrada (amarillo) y salida (azul) del sistema con una ganancia $K_p=1$

A continuación, variamos la ganancia proporcional del sistema, utilizando una ganancia proporcional de valor 10 ($K_p=10$), y observamos la respuesta del sistema en la *Figura 3-4*.

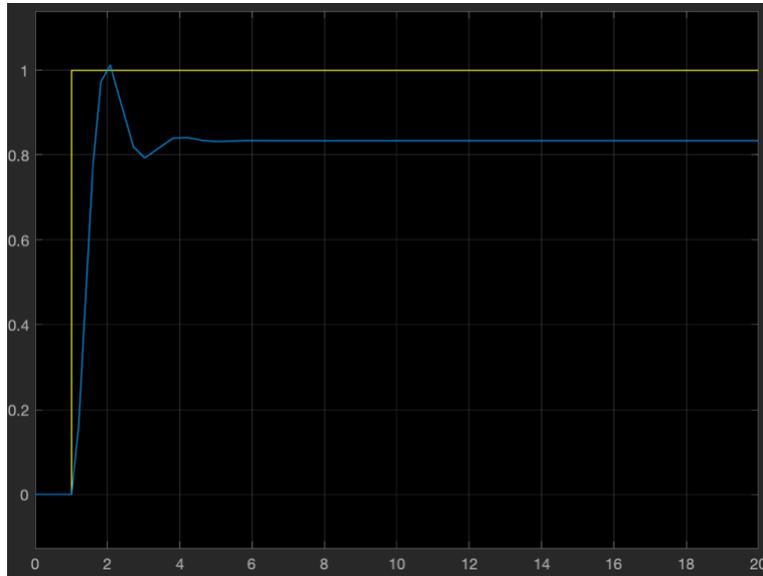


Figura 3-4 Entrada (amarillo) y salida (azul) del sistema con una $K_p=10$

Finalmente aportamos una ganancia de 100 ($K_p=100$) y observamos el comportamiento del sistema en la *Figura 3-5*.

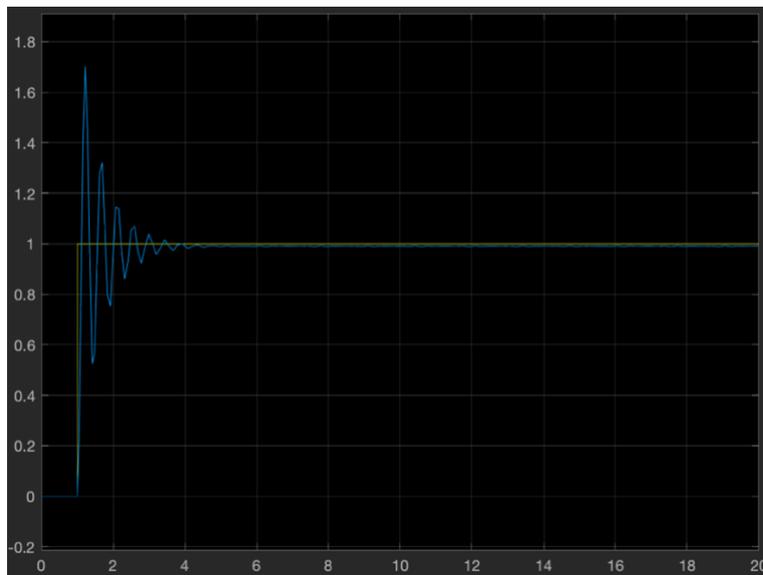


Figura 3-5 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$

Conclusión

Como conclusión, podemos observar que a medida que aumentamos el valor de la ganancia regulador proporcional, el sistema disminuye considerablemente el error de la salida respecto de la entrada, pero por otro lado pierde estabilidad, además de generar oscilaciones y sobreimpulsos en la salida.

3.2.2. Control Proporcional-Derivativo

Este tipo de control utilizará la parte proporcional y la parte derivativa del sistema, utilizando la ganancia proporcional (K_p) y la ganancia derivativa (K_d), definiendo el controlador como la ecuación 3.5.

$$Controlador(t) = K_p * Error(t) + K_d * \frac{dError(t)}{dt} \quad (3.5)$$

Este tipo de controlador utiliza el error en la parte proporcional, y la derivada del error en la parte derivativa, la cual corresponde con la velocidad del error. La ganancia derivativa se utiliza cuando el sistema se mueve a una velocidad alta hacia el punto de referencia, el sistema se pasa de largo debido a su inercia, generando sobreimpulsos u oscilaciones en torno al punto de referencia, como podemos observar en la *Figura 3-6*, para evitar este problema, el controlador debe de conocer la velocidad a la que el sistema se acerca a la salida para así poder frenarle con antelación a medida que se acerque a la referencia deseada y así evitar que la sobrepase y genere las oscilaciones.

Al aumentar la ganancia derivativa se producen los siguientes efectos sobre el sistema:

- Aumenta la estabilidad del sistema controlado
- Disminuye la velocidad del sistema
- No altera el error en régimen permanente

A continuación, vamos a ver algunos ejemplos utilizando diferentes valores de sus ganancias, basándonos en el regulador proporcional con una ganancia proporcional igual a cien ($K_p=100$).

Podemos comprobar el comportamiento del sistema al añadir la parte derivativa, comparándolo con la *Figura 3-5* del apartado 3.2.1. *Control Proporcional*.

Cuando la ganancia derivativa es igual a 1 ($K_d=1$), observamos la respuesta del sistema en la *Figura 3-6*.

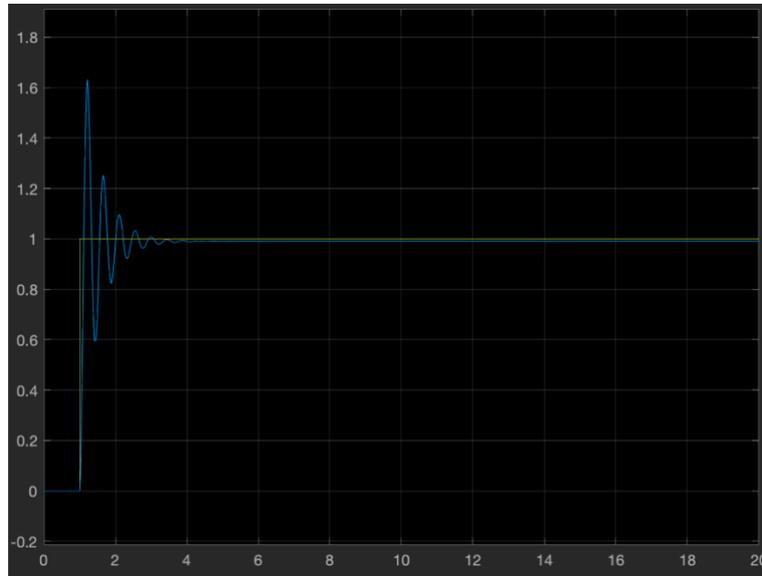


Figura 3-6 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$ y $K_d=1$

A continuación, aumentamos el valor de la ganancia derivativa hasta alcanzar un valor de 10 y obtenemos la respuesta del sistema de la *Figura 3-7*.

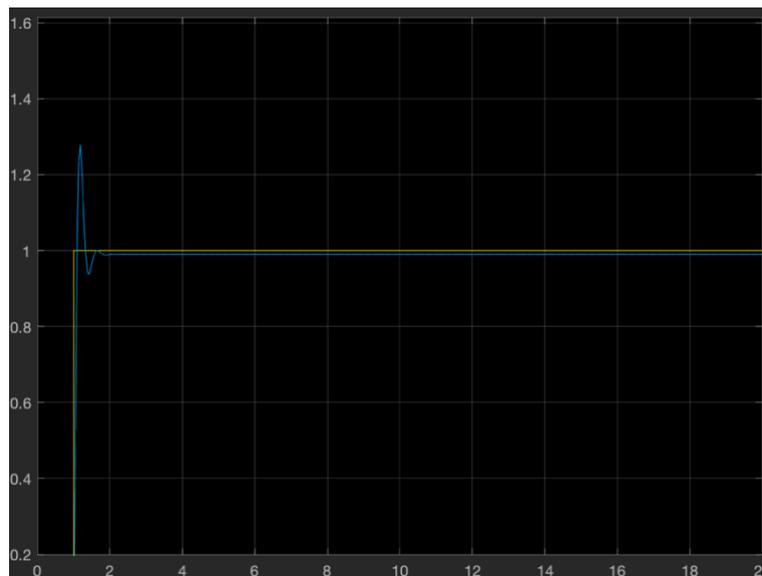


Figura 3-7 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$ y $K_d=10$

Finalmente aumentamos la ganancia derivativa hasta un valor de 100 y observamos el comportamiento del sistema en la *Figura 3-8*.

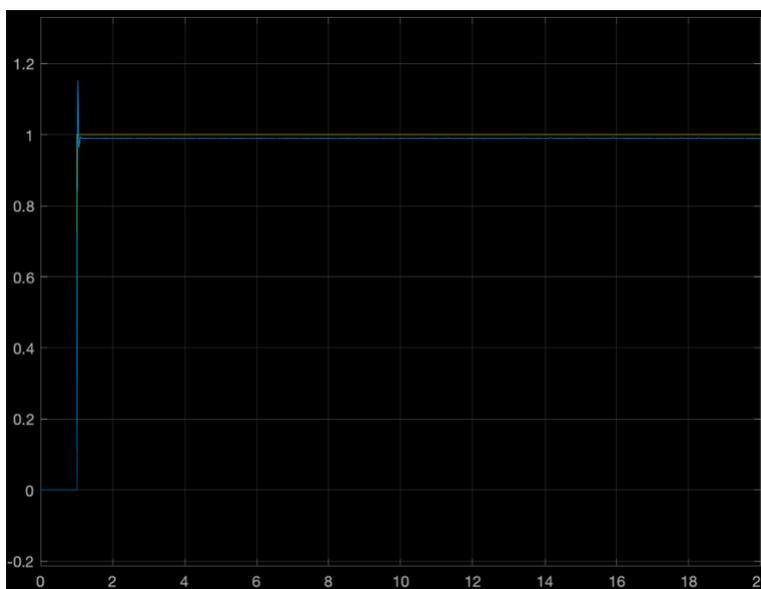


Figura 3-8 Entrada (amarillo) y salida (azul) del sistema con una $K_p=100$ y $K_d=100$

Conclusión

Como hemos podido observar en las imágenes superiores, cuanto más aumentamos la ganancia derivativa del sistema, menor es la oscilación que se produce, el error como podemos observar no varía nunca.

3.2.3. Control Proporcional-Integrativo

Esta acción de control, combinará la parte proporcional y la parte integrativa del regulador, es decir, utilizaremos una ganancia proporcional (K_p) y una ganancia integrativa (K_i), siendo la ecuación de nuestro controlador 3.6.

$$Controlador(t) = K_p * Error(t) + K_i * \int Error(t) * dt \quad (3.6)$$

La parte integrativa es la suma de la señal del error, es un acumulador de errores, es decir, a medida que va pasando el tiempo se va sumando el error del sistema, haciendo que la acción integral sea cada vez mayor, con ello se consigue reducir el error del sistema en régimen permanente.

Los efectos al aumentar la ganancia integrativa (k_i):

- Disminuye el error hasta hacerlo cero en régimen permanente.
- Aumenta la inestabilidad del sistema.
- Aumenta ligeramente la velocidad del sistema

A continuación, vamos a añadir la parte integrativa al control proporcional con una ganancia proporcional igual a diez ($k_p=10$), *Figura 3-4* del apartado 3.2.1. *Control Proporcional*, podemos observar que su respuesta tiene un error en su respuesta de 0.18 (Error = 0.18).

Probamos el sistema con una ganancia integrativa de valor 1 ($K_i = 1$), obteniendo la respuesta de la *Figura 3-9*.

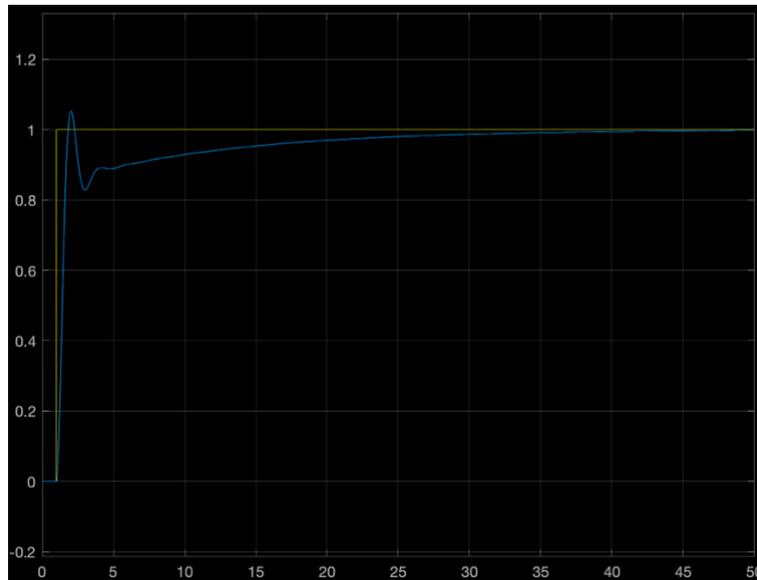


Figura 3-9 Entrada (amarillo) y salida (Azul) del sistema con una $K_p=10$ y una $K_i=1$

A continuación, variamos el valor de la ganancia integrativa, subimos su valor a diez ($K_i=10$) y obtenemos la respuesta del sistema *Figura 3-10*.

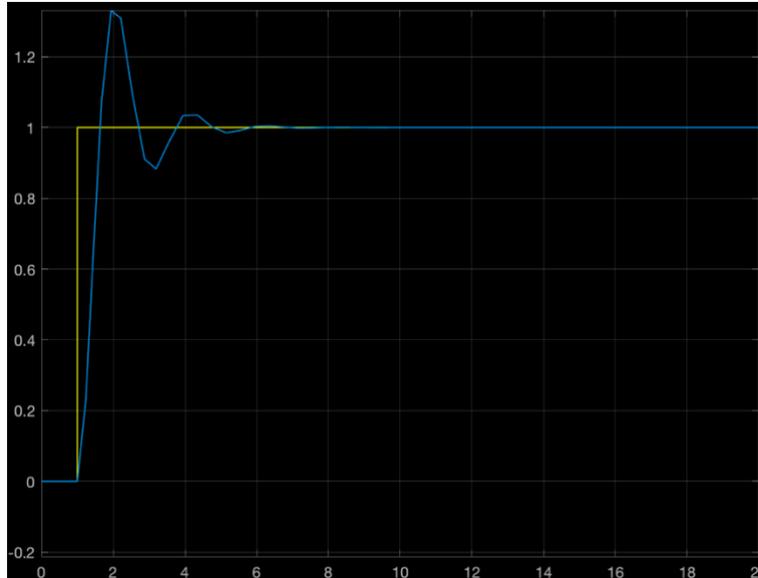


Figura 3-10 Entrada (amarillo) y salida (Azul) del sistema con una $K_p=10$ y una $K_i=10$

Finalmente en la *Figura 3-11*, probamos con una ganancia integrativa de cien ($K_i=40$).

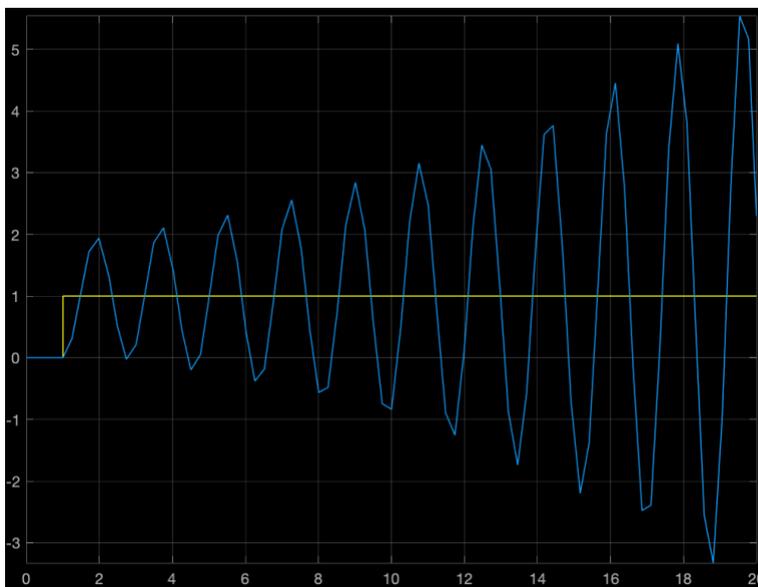


Figura 3-11 Entrada (amarillo) y salida (Azul) del sistema con una $K_p=10$ y una $K_i=40$

Conclusión

Como podemos observar en las imágenes a medida que subimos el valor de la ganancia integrativa, se alcanza un error cero mucho antes, por otro lado también observamos que aumenta la velocidad del sistema y por ello aparecen sobreimpulsos y oscilaciones, cuando aumentamos en exceso el valor de la ganancia integrativa el sistema se hace por completo inestable.

3.2.4. Control Proporcional-Integrativo-Derivativo (PID)

Este sistema constará de las tres partes posibles, como observamos en la ecuación 3.7, la parte proporcional, la parte integrativa y la parte derivativa.

$$Controlador(t) = Kp * Error(t) + Ki * \int Error(t) * dt + Kd * \frac{dError(t)}{dt} \tag{3.7}$$

Este tipo de control es uno de los controles más simples y eficaces que existen, este sistema será capaz de obtener un error nulo, además de poder variar la velocidad de la respuesta del sistema, así como sus sobreimpulsos y oscilaciones.

En el siguiente ejemplo (*Figura 3-12*) vamos a analizar el sistema con los siguientes valores de las diferentes ganancias del sistema, $Kp = 10$, $Ki = 10$ y $Kd = 10$.

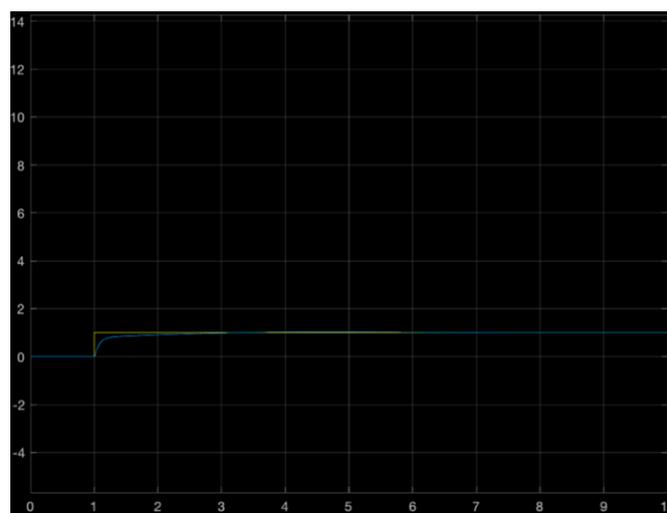


Figura 3-12 Entrada (amarillo) y salida (Azul) del sistema con los siguientes valores $Kp=Kd=Ki=10$

Como podemos observar el sistema no tiene ningún tipo de sobreimpulso o oscilación, la respuesta es estable y tiende a un error de cero con una cierta velocidad.

Conclusión

El comportamiento de los diferentes tipos de reguladores es muy variado, con variaciones en sus errores, en las velocidades de la respuesta, así como con el número de oscilaciones que tarda el sistema en estabilizarse, o incluso con la propia estabilidad del sistema.

El código correspondiente a esta parte se realizará con Arduino, y podemos encontrar el código correspondiente en el *ANEXO-PID*.

3.3. OBTENCIÓN DE LAS GANANCIAS DEL REGULADOR

Para conseguir el mejor regulador PID posible, debemos de tener unos valores de los diferentes parámetros concretos (K_p, K_i y K_d), para conseguir dichos parámetros ideales hay dos formas de conseguirlo, una forma manual, la cual es muy menos precisa y sencilla de realizar y por otro lado se puede aplicar el método Ziegler-Nichols, una forma mucho más precisa.

Procedemos a ajustar el sistema de la *Figura 3-13* con el que hemos estado realizando los apartados 3.2. *TIPOS DE CONTROLADORES*.

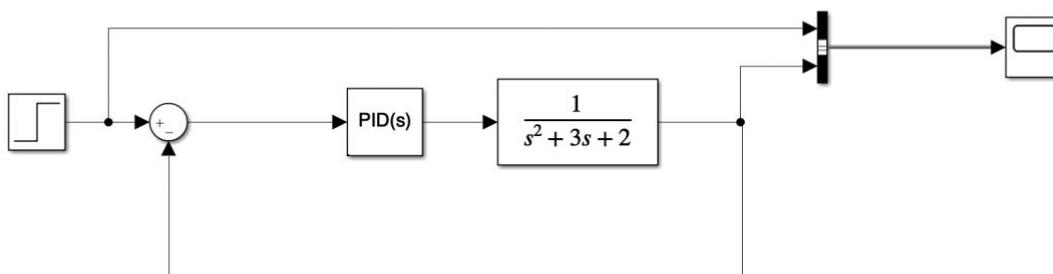


Figura 3-13 Sistema Ejemplo

3.3.1. Sintonización manual de los parámetros del PID

Esta forma de obtener los parámetros ideales del sistema se consigue manipulando directamente la planta [4], es decir, variando tanto la señal de la entrada como los diferentes valores de las ganancias, para ajustar cada uno de los valores debemos de seguir los siguientes pasos:

- Ajuste del valor de la ganancia proporcional (K_p)

En este primer paso vamos a ajustar la ganancia K_p , para ello incrementamos de forma ligera el valor de la ganancia proporcional, como efecto de este aumento disminuimos el error y aumentamos la velocidad de respuesta del sistema. Debemos parar el aumento de la ganancia proporcional cuando el sistema sobrepase o empiece a oscilar.

La ganancia proporcional ideal, tiene un valor 1 ($K_p=1$), como podemos observar en *Figura 3-14*, al aumentarla la respuesta del sistema comienza a oscilar.

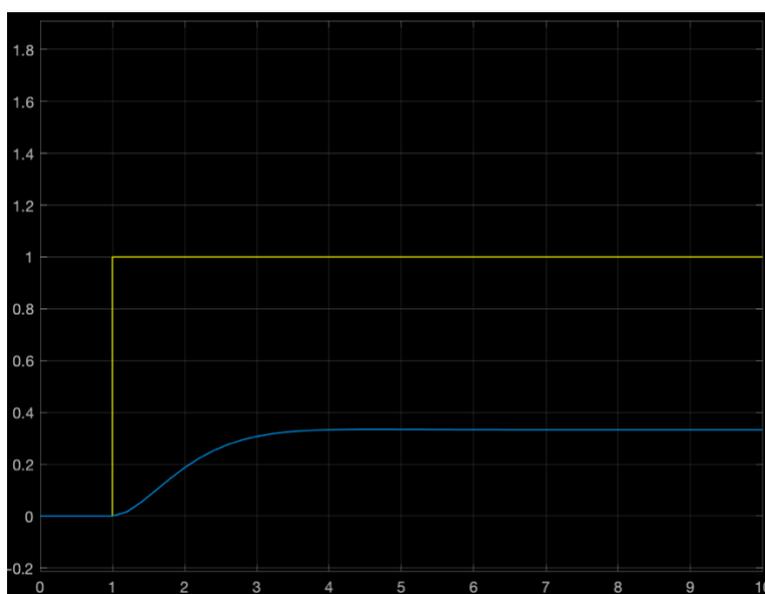


Figura 3-14 Respuesta del sistema con una $K_p=1$

- Ajuste del valor de la ganancia integrativa (K_i)

A continuación, se procede a aumentar la ganancia integrativa, la cual nos va a disminuir el error hasta igualarlo a cero, a medida que aumentemos la ganancia integrativa el error disminuirá

pero aumentará la velocidad del sistema y la inestabilidad, deberemos de aumenta el valor de la ganancia integrativa hasta eliminar el error.

Obtenemos que la ganancia integrativa es de valor 1 ($K_i=1$), como observamos en la *Figura 3-15*, el error se elimina por completo a partir de los 8 segundos.

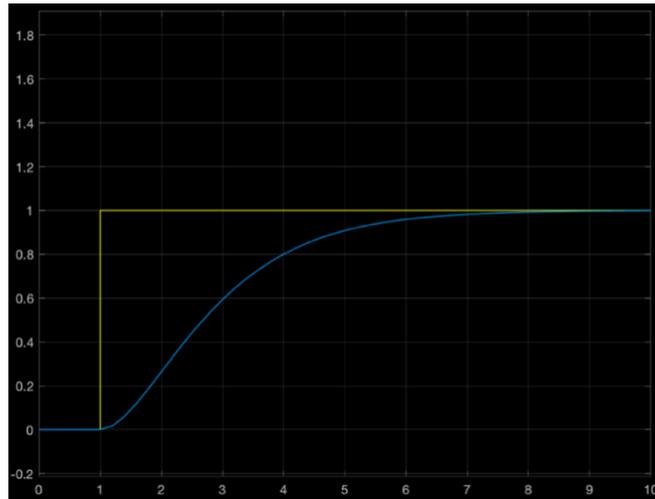


Figura 3-15 Respuesta del sistema con una $K_p=1$ y $K_i=1$

- Ajuste del valor de la ganancia derivativa (K_d)

Finalmente el ultimo paso consiste en ajustar el valor de la ganancia derivativa, a medida que la aumentemos el valor del parámetro, disminuye la velocidad de respuesta y la estabilidad del sistema. Aumentamos el valor de la K_d , hasta que el sistema comience a comportarse de una forma extraña.

Se aumenta el valor de la ganancia derivativa hasta alcanzar un valor de 0.25 (*Figura 3-16*).

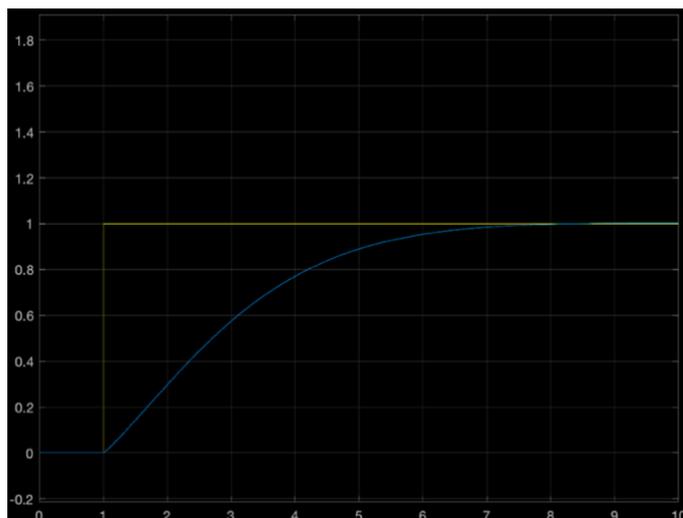


Figura 3-16 Respuesta del sistema con una $K_p=1$, $K_i=1$ y $K_d=0.25$

3.3.2. Método Ziegler-Nichols

El método Ziegler-Nichols [5] consiste en un método por el cual conseguiremos ajustar los diferentes parámetros que componen nuestro controlador PID de forma empírica, en este proceso no es necesario conocer la ecuación de transferencia que define nuestra planta.

Este sistema de ajuste propuesto por Ziegler y Nichols fue publicado en 1942 y desde su creación es uno de los métodos de sintonización más ampliamente difundido y utilizado, debido a su sencilla aplicación y su alta precisión a la hora de obtener las ganancias de un sistema.

Los valores de las ganancias obtenidos por este método intentan conseguir en el sistema realimentado una respuesta al escalón con un sobreimpulso máximo del 25%, en el cual tendremos buenas características de velocidad y estabilidad.

Existen dos tipos de sintonizaciones por este método, la primera se trata de aplicar el método de lazo abierto la otra técnica consiste en realizar la sintonía en lazo cerrado.

Para realizar este proceso vamos a utilizar la función de transferencia que en apartados anteriores 3.8, aunque hay que destacar que para realizar este método no es necesario conocer la función de transferencia del sistema, lo cual facilita considerablemente el proceso.

$$Process(s) = \frac{1}{s^2 + 3 * s + 2} \tag{3.8}$$

3.3.2.1. Método Ziegler-Nichols Lazo Abierto

El primer método consiste en realizar el ajuste de los parámetros del sistema en lazo abierto [6] *Figura 3-17*, donde simplemente se va a aplicar una entrada escalón unitario a el sistema y observaremos la respuesta de nuestro sistema.

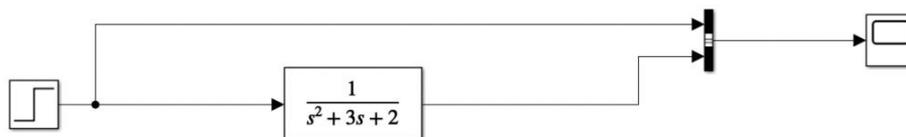


Figura 3-17 Sistema

Este método se podrá aplicar siempre y cuando la respuesta del sistema sea una respuesta sigmoideal (*Figura 3-18*), en el caso de que la respuesta del sistema se tratase de un respuesta integrador o una respuesta oscilatoria no se podría aplicar este método de lazo abierto.

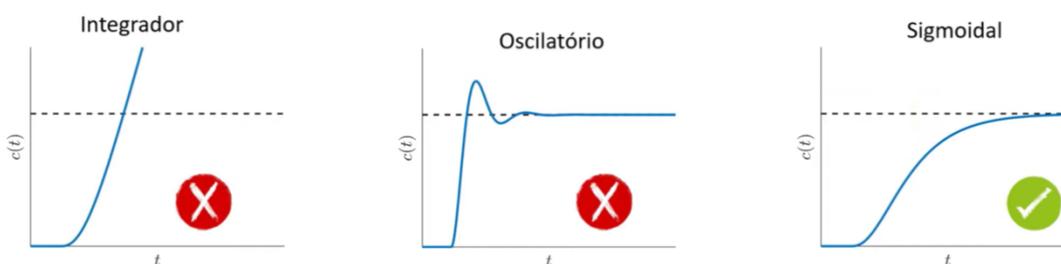


Figura 3-18 Posibles respuestas del sistema

Respuesta de nuestro sistema al aplicar la entrada escalón unitario, *Figura 3-19*.

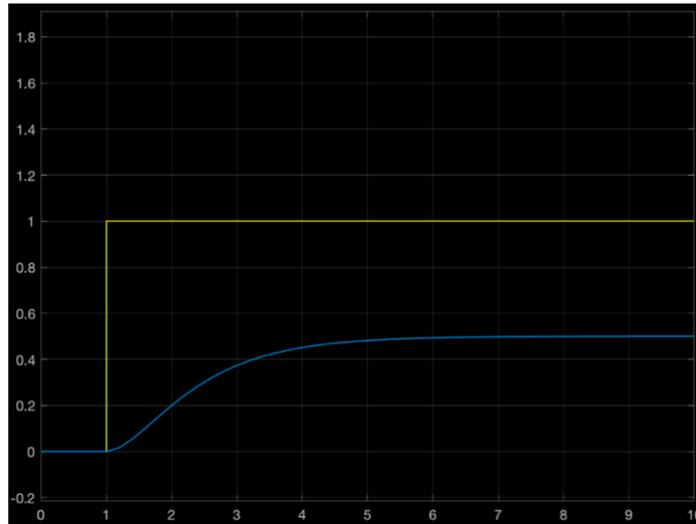


Figura 3-19 Respuesta del sistema ante la entrada escalón

Utilizamos este tipo de respuestas para aproximarlos a una ecuación de primer orden 3.9, debido que no conocemos la dinámica del proceso.

$$G(s) = \frac{K * e^{-L*s}}{T * s + 1}$$

(3.9)

El siguiente paso en este método consiste en trazar una línea recta tangente en el punto de inflexión de la respuesta de nuestro sistema, *Figura 3-20*. A continuación, para obtener el parámetro "L" o retardo del sistema, vamos a medir desde el instante inicial que se aplicó el escalón unitario hasta donde corta con el eje del tiempo. Por otro lado, obtenemos la constante "K" que es donde se estabiliza el sistema.

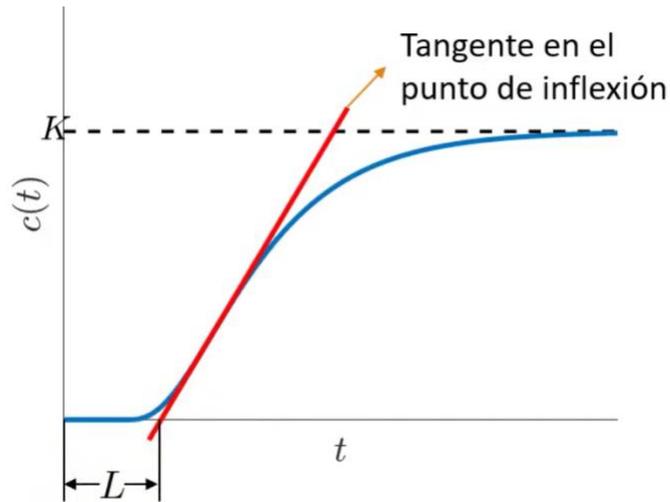


Figura 3-20 Representación del parámetro "L" o retardo del sistema

Finalmente procedemos a obtener "T" o constante de tiempo, Figura 3-21, es la distancia que existe entre el final de la constante del retardo del sistema (L) y donde dicha línea tangente se intercepta con la línea prolongada de la ganancia del sistema.

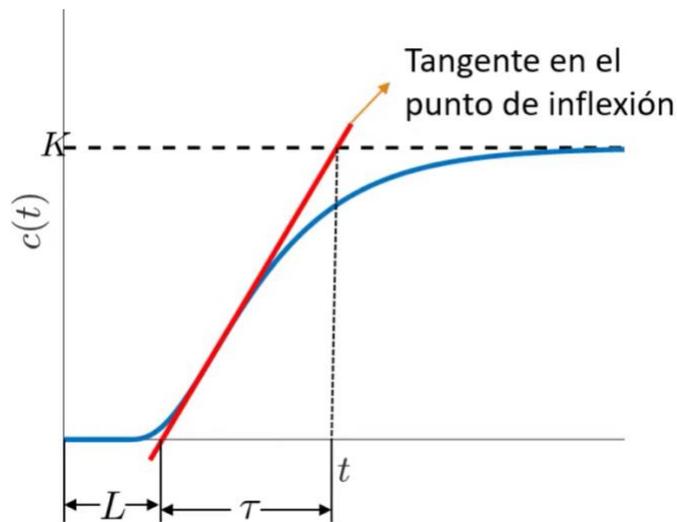


Figura 3-21 Representación del parámetro "L" y "T"

Una vez hemos obtenido los diferentes parámetros podemos proceder a obtener los diferentes valores de los parámetros para poder obtener los reguladores deseados, se puede obtener reguladores proporcionales, reguladores Proporcionales-Integrativos y reguladores Proporcionales-Integrativos-Derivativos.

Tabla 1 Valor de las ganancias para los diferentes reguladores a través del primer método Ziegler-Nichols

Regulador	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$\frac{0.9 * T}{L}$	$\frac{L}{0.3}$	0
PID	$\frac{1.2 * T}{L}$	$2 * L$	$0.5 * L$

Cuando estamos en este tipo de sintonías también es importante evaluar el valor que obtenemos de la división del retardo del sistema o "L" entre la constante de tiempo o "T", este factor es conocido como factor de incontrolabilidad, el valor debería de estar comprendido entre 0.1 y 0.3 (3.10).

$$0.1 \leq L/T \leq 0.3 \tag{3.10}$$

Según algunos autores el factor de controlabilidad podría ascender de límite superior hasta los 1.4, en vez de 0.3, sin embargo con un factor de controlabilidad tan grande no podríamos garantizar la robustez del sistema ante eventuales errores de modelo que se puedan llegar a tener.

Para obtener nuestro regulador deseado sustituimos los valores de la ganancias en la ecuación 3.11.

$$Regulador(s) = Kp * (1 + \frac{1}{Ti * s} + Td * s) \tag{3.11}$$

Sustituyendo L y T para obtener nuestro regulador PID (3.12)

$$ReguladorPID(s) = 1.2 * \frac{T}{L} * \left(1 + \frac{1}{2 * L * s} + 0.5 * L * s\right) \tag{3.12}$$

$$ReguladorPID(s) = 0.6 * \frac{T}{s} * \left(s + \frac{1}{L}\right)^2$$

Como podemos observar en la *Figura 3-22*, este regulador nos está aportando un polo en el cero y dos ceros a una distancia de 1/L en el plano negativo.

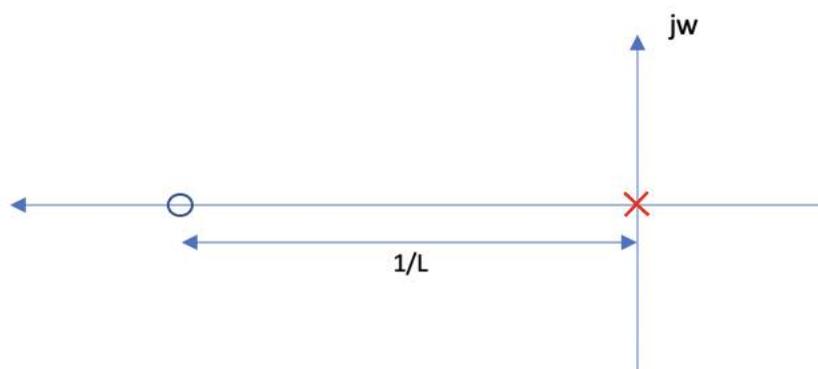


Figura 3-22 Representación polos y ceros

Muchas veces para mejorar la respuesta del sistema, deberemos disminuir la ganancia del controlador. Una práctica común para resolver este problema es dividir la ganancia del controlador a la mitad (3.13) para obtener una respuesta más suave.

$$Kp = Kp/2 \tag{3.13}$$

Cabe destacar que esta ganancia la podremos disminuir mucho más que la mitad, dependiendo de la respuesta del proceso, debido a que este método no garantiza la estabilidad del sistema.

Procedemos a analizar cada uno de los parámetros de nuestro sistema, primero analizamos la ganancia proporcional del controlador, la cual es inversamente proporcional a la constante de nuestro proceso (3.14).

$$K = \frac{1}{Kp} \tag{3.14}$$

Si el proceso tiene un ganancia grande (K), entonces nuestra Kp tendrá un valor pequeño.

También si miramos el factor de incontrolabilidad, también será inversamente proporcional a la ganancia del controlador (3.15), es decir, si el factor de controlabilidad es muy grande va a ser difícil dominar el proceso, porque va relacionado con el retardo de tiempo.

$$Kp = \frac{L}{T} \tag{3.15}$$

A continuación, analizamos la parte integrativa del regulador (Ti), el cual esta directamente relacionado con el retardo del tiempo (L), en el caso que nuestro sistema es muy lento, es decir, el valor de "L" es muy grande, entonces nuestro tiempo integral será también grande.

De la misma forma si procedemos a analizar la parte derivativa del regulador, está directamente relacionado con el retardo del tiempo, a mayor tiempo de retardo mayor tiempo derivativo.

Existe una relación entre el tiempo integral y el tiempo derivativo, Ziegler y Nichols estipularon la ecuación 3.16.

$$Ti = 4 * Td \tag{3.16}$$

Finalmente aplicamos el método a nuestro sistema ejemplo (3-23).

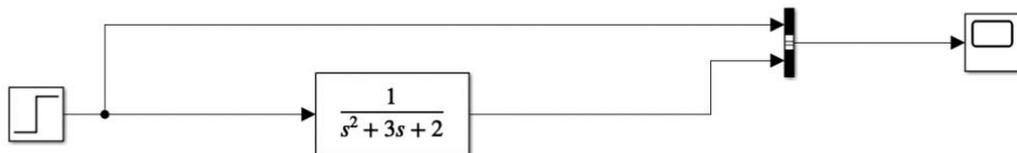


Figura 3-23 Sistema

El primer paso consiste en obtener la respuesta del sistema ante una entrada escalón (Figura 3-24).

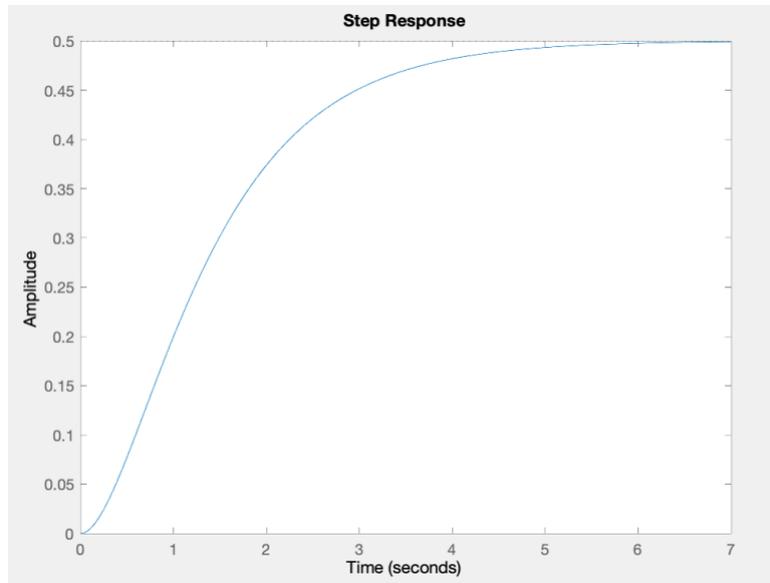


Figura 3-24 Respuesta del sistema ante entrada escalón unitario

El siguiente paso consiste en calcular el punto de máxima inflexión, es decir, calcular la derivada de la curva la cual será la pendiente máxima de nuestra respuesta, la calculamos y la representamos, *Figura 3-25* . La máxima derivada como podemos observar en la gráfica encuentra en la muestra número 70.

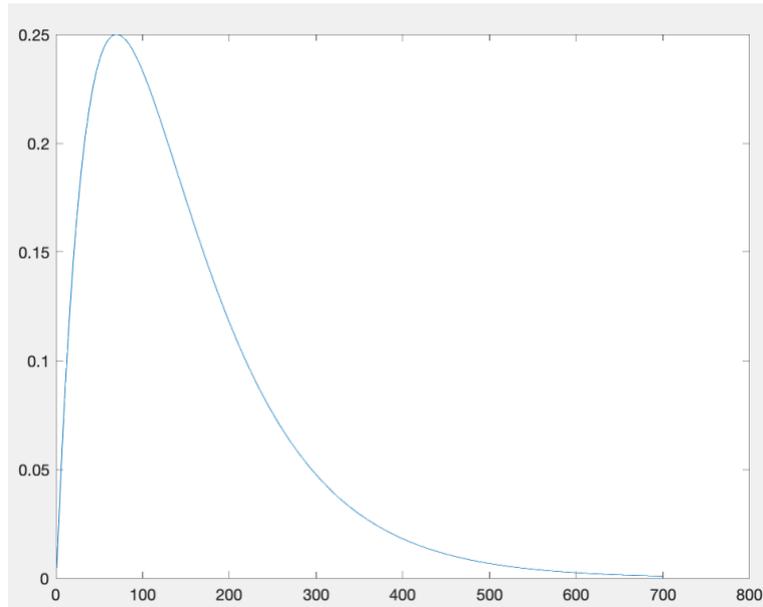


Figura 3-25 Punto de máxima inflexión

A continuación, procedemos a calcular y trazar la ecuación de la recta que es tangente al punto máximo de la derivada. En la *Figura 3-26*, obtenemos la recta, la respuesta del sistema ante la entrada escalón y la derivada de la respuesta.

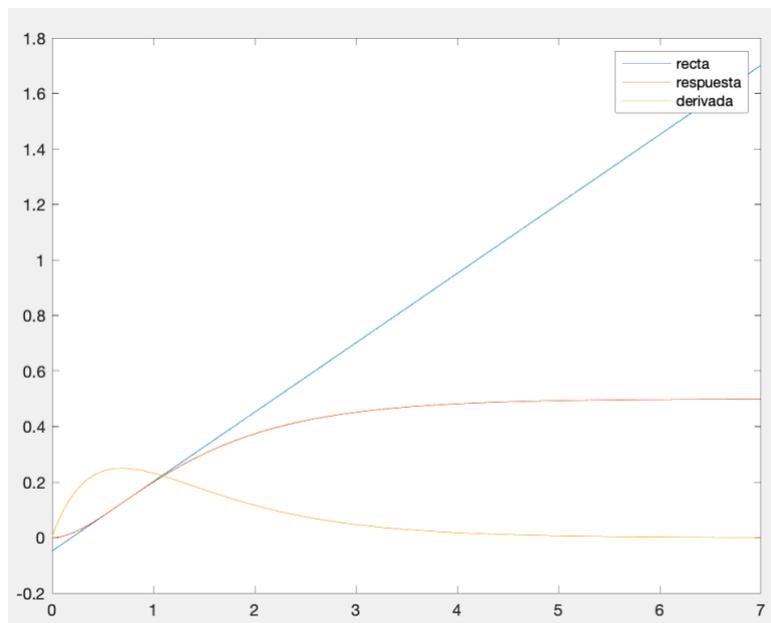


Figura 3-26 Representación de la recta, respuesta y derivada

El siguiente paso consiste en calcular el valor de retardo del sistema (L) y del valor del parámetro de la constante del tiempo (T). En este caso los valores de los parámetros obtenidos son los siguientes:

$$L = 0.2$$

$$T = 1.99$$

Como podemos observar en la ecuación 3.17, cumplen con el factor de incontrolabilidad.

$$Factor = \frac{0.2}{1.99} = 0.1$$

$$0.1 \leq Factor \leq 0.2$$

(3.17)

Finalmente procedemos a obtener los diferentes tipos de reguladores:

- Regulador Proporcional

Obtenemos las ganancias del regulador proporcional (3.18)

$$Kp = \frac{T}{L} = \frac{1.99}{0.2} = 9.95$$

$$Ki = \frac{Kp}{Ti} = \frac{9.95}{\infty} = 0$$

$$Kd = Kp * Td = 9.95 * 0 = 0$$

(3.18)

Seguidamente, procedemos a observar la respuesta del sistema ante el regulador proporcional
Figura 3-27.

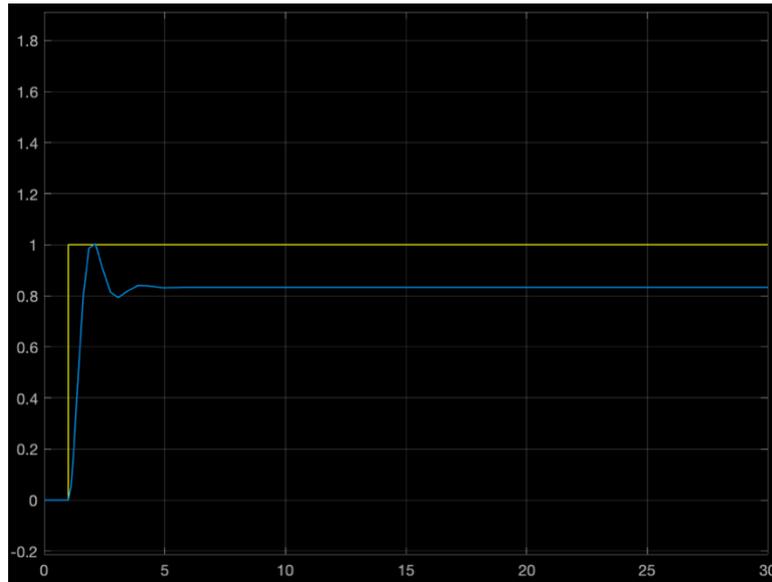


Figura 3-27 Respuesta del sistema ante el regulador proporcional

- Regulador Proporcional-Integrativo

Procedemos a obtener los parámetros que lo forman (3.19).

$$Kp = 0.9 * \frac{T}{L} = 0.9 * \frac{1.99}{0.2} = 8.955$$

$$Ki = \frac{Kp}{Ti} = \frac{Kp}{3.33 * L} = \frac{8.955}{3.33 * 0.2} = 13.45$$

(3.19)

$$Kd = Kp * Td = 9.95 * 0 = 0$$

Se introducen los valores del regulador y se observa la respuesta del sistema, *Figura 3-28*.

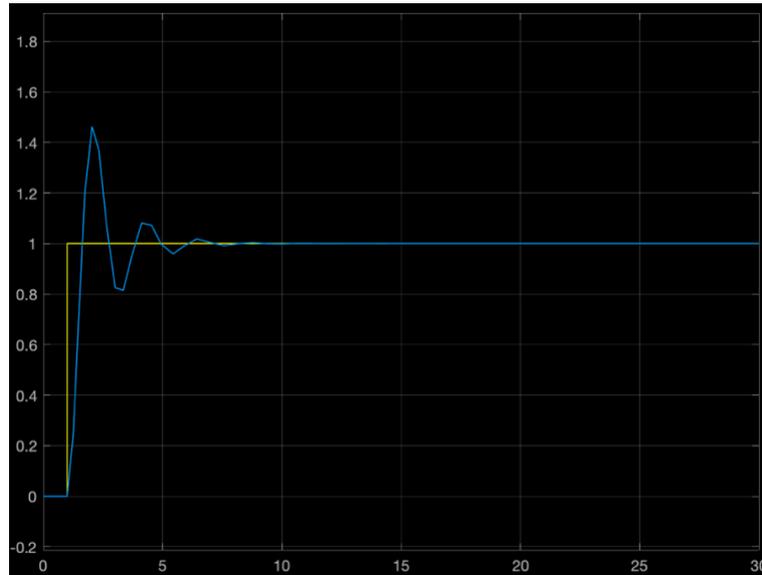


Figura 3-28 Respuesta del sistema ante el regulador Proporcional-Integrativo

- Regulador Proporcional-Integrativo-Derivativo

Se obtienen los parámetros que conforman el regulador (3.20)

$$Kp = 1.2 * \frac{T}{L} = 1.2 * \frac{1.99}{0.2} = 11.94$$

$$Ki = \frac{Kp}{Ti} = \frac{Kp}{2 * L} = \frac{11.94}{2 * 0.2} = 29.85$$

(3.20)

$$Kd = Kp * Td = Kp * 0.5 * L = 11.94 * 0.5 * 0.2 = 1.194$$

Procedemos a obtener la respuesta del sistema con la utilización del regulador PID, *Figura 3-29*.

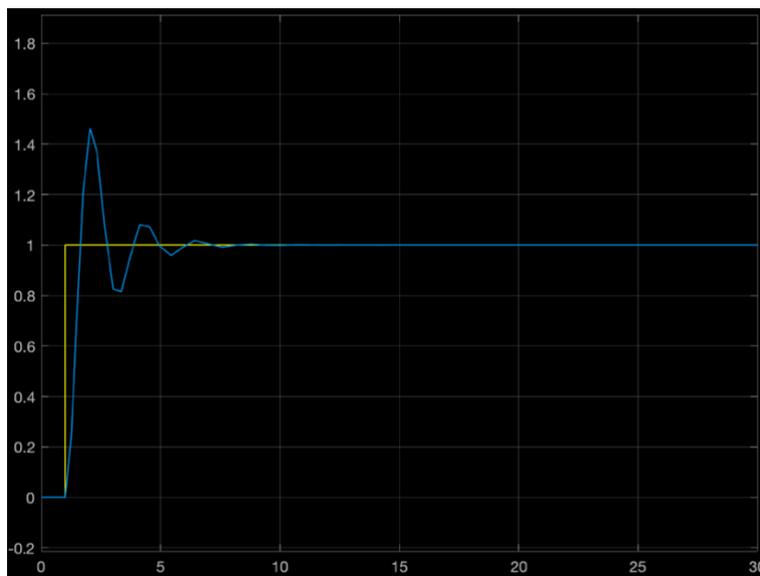


Figura 3-29 Respuesta del sistema ante el regulador Proporcional-Integrativo-Derivativo

Conclusión

Este método es el que se va a utilizar para poder obtener los valores de las ganancias que van a componer los diferentes tipos reguladores de nuestro sistema, se trata de un sistema muy sencillo, en el cual no es necesario conocer la función de transferencia del sistema, lo cual agiliza considerablemente el obtener los parámetros de los reguladores.

Como podemos observar en el ejemplo, la respuesta del PID, es una respuesta muy buena, con un cierto sobreimpulso y oscilación al principio, que al cabo de unos pocos segundos tiende a estabilizarse con un error cero y con una velocidad de respuesta muy buena. Este ejemplo, no tiene porque parecerse a nuestro sistema, pero se trabajaría con el método de la misma manera.

El código que se va a utilizar para este método se va a realizar en MATLAB, y le podemos encontrar en el *ANEXO- METODO ZIEGLE-NICHOLS*.

3.3.2.2. Método Ziegler-Nichols Lazo Cerrado

Este segundo método de Ziegler-Nichols [6] a diferencia del primer método, este método se ha de realizar en lazo cerrado, como se puede observar en la *Figura 3-30*.

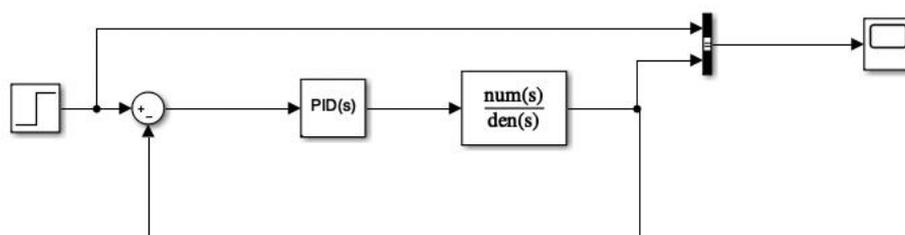


Figura 3-30 Sistema en lazo cerrado

El primer paso consiste en igualar a cero la acción derivativa y la acción integrativa y variar la acción proporcional, de forma paulatina, vamos a observar a la vez la salida del sistema, a medida que aumentamos el valor de la ganancia proporcional (K_p), el sistema comenzará a ser más oscilatorio, la aumentaremos hasta que las oscilaciones se mantengan con una amplitud constante, es decir, se trate de un sistema críticamente estable, obteniendo la salida de la *Figura 3-31*.

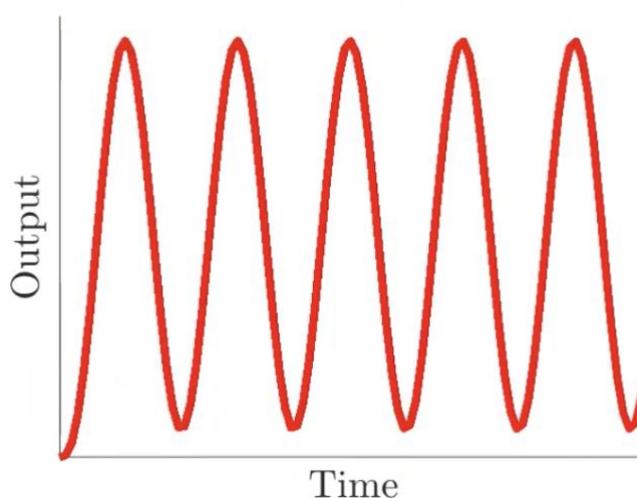


Figura 3-31 Respuesta del sistema críticamente estable

Tabla 2 Valor de las ganancias para los diferentes reguladores a través del segundo método Ziegler-Nichols

Regulador	Kp	Ti	Td
P	$0.5 * Ku$	∞	0
PI	$0.45 * Ku$	$\frac{Pu}{1.2}$	0
PID	$0.6 * Ku$	$0.5 * Pu$	$0.125 * Pu$

Finalmente una vez hemos obtenido los diferentes valores de los parámetros que van a conformar nuestro regulador, sustituimos dichos valores en la ecuación 3.22, obteniendo el regulador deseado.

$$Regulador(s) = Kp * \left(1 + \frac{1}{Ti * s} + Td * s\right) \tag{3.22}$$

Para el caso de que queramos utilizar el PID, sustituimos el valor de sus parámetros en la ecuación 3.22, obteniendo la ecuación 3.23.

$$Regulador(s) = 0.6 * Ku * \left(1 + \frac{1}{0.5 * Pu * s} + 0.125 * Pu * s\right) \tag{3.23}$$

$$Regulador(s) = 0.075 * Ku * \frac{Pu}{s} * \left(s + \frac{4}{Pu}\right)^2$$

Como podemos observar en la ecuación del regulador PID, nos va a aportar un polo en cero al sistema y dos ceros a una distancia de $4/Pu$, en el plano negativo. En *Figura 3-33*, podemos ver los dos polos dobles y el cero representados.

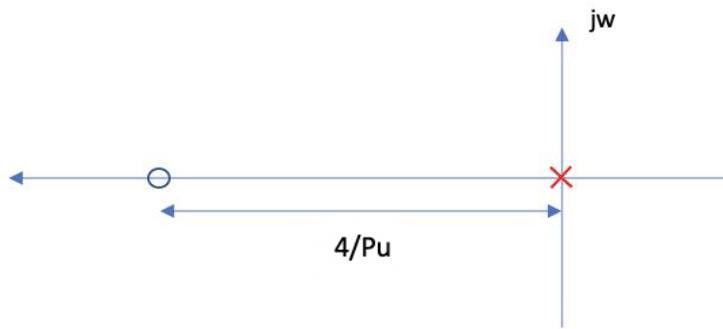


Figura 3-33 Representación polos y cero

4. CIRCUITOS ELECTRÓNICOS

Esta parte del proyecto es una de las más importantes pues será la parte encargada de facilitar la comunicación entre la tarjeta de adquisición de datos y la propia planta.

Nuestra planta solo trabajará de forma interna con señales analógicas que oscilen entre los 0 Voltios y los 5 Voltios, estas señales se corresponderán a la señal de entrada del sistema, la cual es aportada a través de un potenciómetro conectando con sus terminales por 5 Voltios y a tierra, y la señal de salida que corresponderá a otro potenciómetro que funcionará exactamente de la misma manera, solo que en vez de ser pilotado directamente de forma manual, como es el caso de la entrada, este será pilotado directamente por la transmisión del movimiento del eje del motor de corriente continua.

Sin embargo, nuestra tarjeta de adquisición de datos deberá de trabajar en un rango comprendido entre los -5 Voltios y los 5 Voltios, debido que se conseguirá abarcar la amplitud de las señales para conseguir el mayor número de bits posibles, y tener la mejor señal posible.

Existirán dos tipos de circuitos, por un lado deberemos de ser capaces de realizar una lectura de las señales de entrada y de salida del sistema, las cuales están entre 0-5 Voltios, estas señales deberán de ser transformadas a través de un circuito electrónico a señales con una amplitud de onda comprendida entre los -5 y los 5 Voltios. Por otro lado, el otro circuito, este deberá de ser capaz de convertir la señal que nos aporta la tarjeta de adquisición de datos, comprendida entre -5 y 5 Voltios, en una señal legible para nuestro sistema, es decir, entre 0-5 Voltios, deberá de trabajarse con este rango de tensiones debido a que nuestro microcontrolador (Arduino Uno Rev3), trabaja con dicho rango de tensiones, cualquier tensión superior provocará daños en la placa.

Por otro lado, nuestros circuitos van a estar alimentado por una fuente de alimentación de ordenador (460W ATX), *9. FUENTE DE ALIMENTACIÓN*, la cual tendrá alimentaciones de ± 12 Voltios, 5 voltios y una tierra común, por lo que deberemos de acondicionar nuestros circuitos a las alimentaciones disponibles.

4.1. AMPLIFICADOR OPERACIONAL

El amplificador operacional que se va a utilizar para realizar los dos circuitos electrónicos, es el amplificador operacional UA741.

En la *Figura 4-1*, podemos observar los pines de conexión de nuestro amplificador operacional, los pines de alimentación, así como los pines de la entrada inversa y la entrada no inversa y el pin de salida del amplificador.

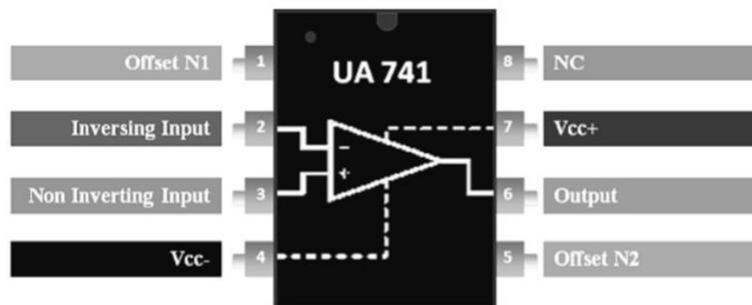


Figura 4-1 Amplificador Operacional UA741

Características de el operacional:

- Tipo de fuente de alimentación: doble
- Tipo de montaje: Montaje en orificio pasante
- Tipo de encapsulado: PDIP
- Tensión de alimentación típica dual: $\pm 15V$, $\pm 12V$, $\pm 9V$, $\pm 5V$, en nuestro caso se utilizará una alimentación dual de $\pm 12V$, debido que es la que nos facilita la fuente de alimentación.

4.2. CIRCUITO CONVERSION 0-5 V a $\pm 5 V$

En este apartado vamos a generar un circuito (*Figura 4-2*) capaz de transformar una tensión analógica de entrada con un amplitud de onda comprendida entre los 0 Voltios y los 5 Voltios, la convertiremos en una señal analógica de salida de ± 5 Voltios

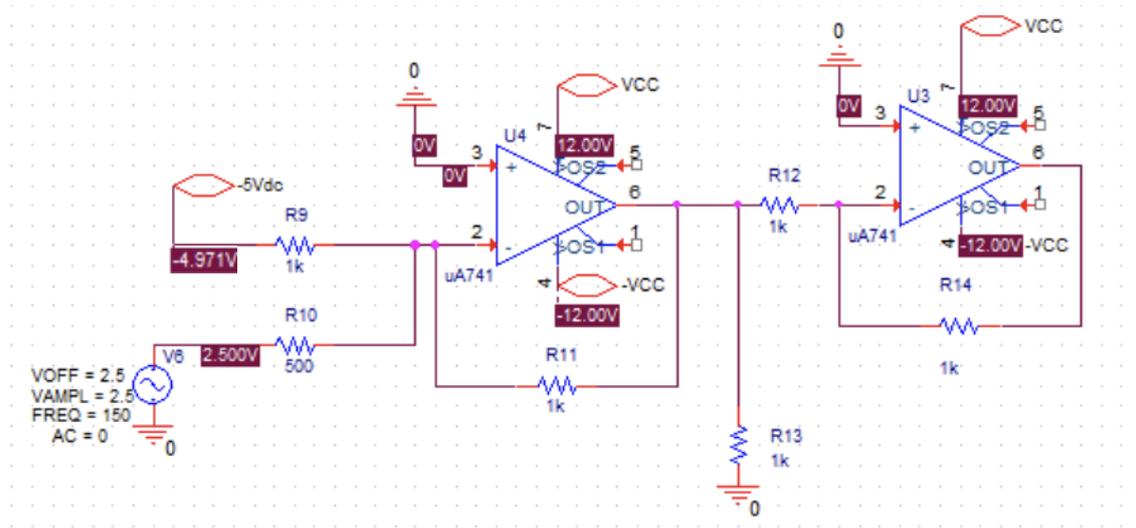


Figura 4-2 Circuito convertidor 0-5 Voltios a ±5 Voltios

Se procede a analizar la primera parte del circuito (Figura 4-3).

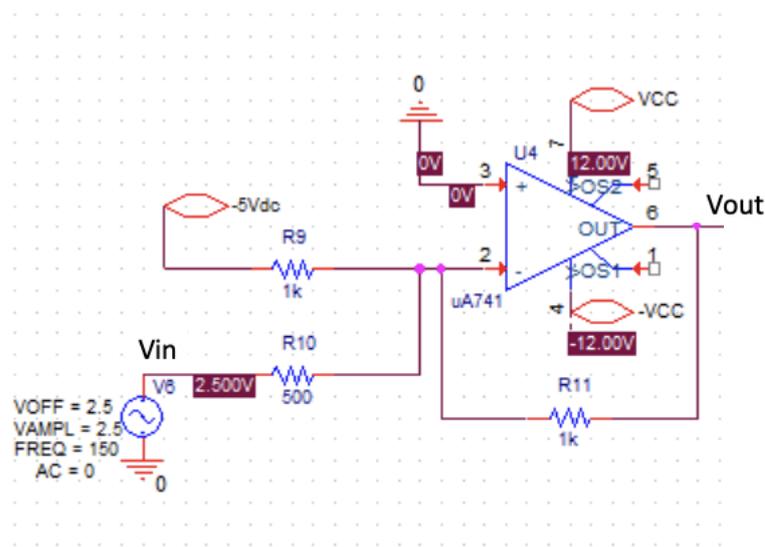


Figura 4-3 Primera parte del circuito convertidor 0-5 Voltios a ±5 Voltios

Aplicando nudos obtenemos la relación entre la señal de entrada y de salida (4.1).

$$\frac{(-5 - 0)V}{1K\Omega} + \frac{(Vin - 0)V}{500\Omega} = \frac{(0 - Vout)V}{1K\Omega}$$

$$V_{out} = 1K\Omega * \left(\frac{5V}{1K\Omega} - \frac{V_{in}}{500\Omega} \right)$$

$$V_{out} = 5V - 2 * V_{in}$$

(4.1)

En el caso de que la tensión de entrada (V_{in}) valiese 5 Voltios, sustituimos en la ecuación 4.1, obtenemos una tensión de salida (V_{out}), igual a -5V.

$$V_{out} = 5V - 2 * 5V = -5V$$

Por otro lado, en el caso de que la tensión de entrada (V_{in}), valiera 0 V, sustituyendo en la ecuación 4.1, obtendríamos una tensión de salida (V_{out}) igual a 5V.

$$V_{out} = 5V - 2 * 0V = 5V$$

En la *Figura 4-4*, podemos observar la representación de la señal de entrada y la señal de salida de esta primera parte del circuito.



Figura 4-4 Salida (rojo) y entrada (verde) de la primera parte del circuito

Como podemos observar la salida del circuito está desfasada de la entrada 120° , por lo que debemos de añadir una segunda parte para que no tengamos la onda desfasada

Analizamos la segunda parte del circuito *Figura 4-5*, donde la entrada del circuito será la salida de la primera parte, pero la vamos a trata como (V_{in}), la salida del circuito será la salida final de el convertor de tensión.

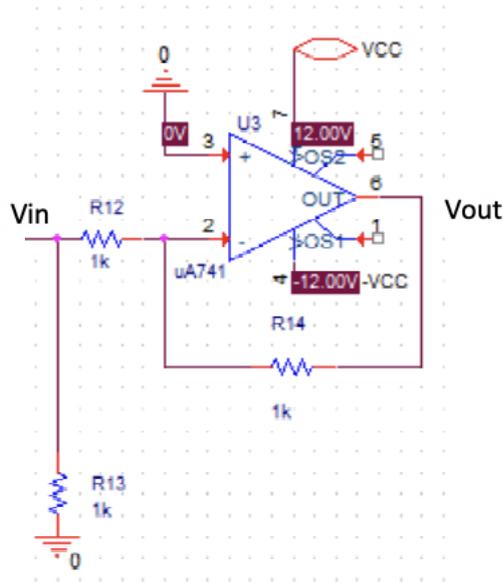


Figura 4-5 Segunda parte del circuito convertidor 0-5 Voltios a ± 5 Voltios

Aplicamos nudos y obtenemos la relación entre la entrada y la salida, ecuación 4.2 .

$$\frac{(V_{in} - 0)V}{1K\Omega} = \frac{(0 - V_{out})V}{1K\Omega}$$

$$V_{out} = -V_{in}$$

(4.2)

Por lo que, en el caso el caso de que la salida del primer circuito fuera -5, es decir, nuestra V_{in} de la segunda parte, sustituyendo en la ecuación 4.2.

$$V_{out} = -V_{in} = -(-5)V = 5V$$

Y en el caso de que nuestra V_{in} fuera de 5 Voltios, entonces sustituyendo en la ecuación 4.2.

$$V_{out} = -V_{in} = -5V = -5V$$

Con este último circuito como podemos comprobar numéricamente, invertimos el signo de las tensiones de salida, obteniendo tensiones que no están desfasadas. Analizando el circuito completo obtenemos la ecuación 4.3, con la cual obtendremos el valor de la tensión de salida (V_{out}), función de la tensión de entrada (V_{in}).

$$V_{out} = 2 * V_{in} - 5 \quad (4.3)$$

En el caso de que la entrada del circuito fuera igual a 5V, sustituimos en la ecuación 4.3, y obtenemos una tensión de salida de 5 V

$$V_{out} = 2 * V_{in} - 5 = 2 * 5 - 5 = 5V$$

Por otro lado, si el valor de la entrada fuera igual a 0V, sustituimos en la ecuación 4.3, y obtenemos una tensión de salida de -5V.

$$V_{out} = 2 * V_{in} - 5 = 2 * 0 - 5 = -5V$$

Como podemos observar, en la *Figura 4-6*, ya hemos convertido las tensiones en los rangos que se quería,

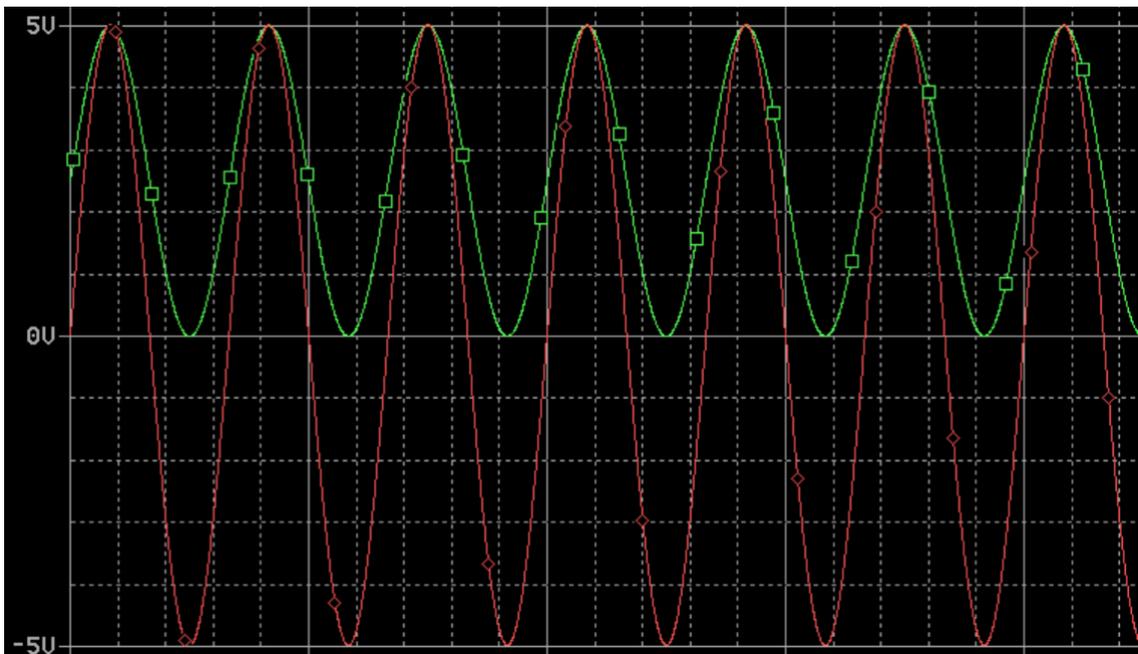


Figura 4-6 Salida (rojo) y entrada (verde) del circuito

Como vemos en la *Figura 4-6*, la entrada se corresponde a una onda cuya amplitud va desde los 0V hasta los 5V, dicha señal se ve amplificada a la salida a una señal comprendida entre los $\pm 5V$.

4.3. CIRCUITO CONVERSOR $\pm 5 V$ a 0-5 V

Este circuito va a ser el encargado de convertir una señal analógica de ± 5 , la cual va a provenir directamente de la tarjeta de adquisición de datos, a un señal de 0-5 Voltios, el rango de tensión en el cual trabaja nuestro microcontrolador, obteniendo el circuito de la *Figura 4-7*

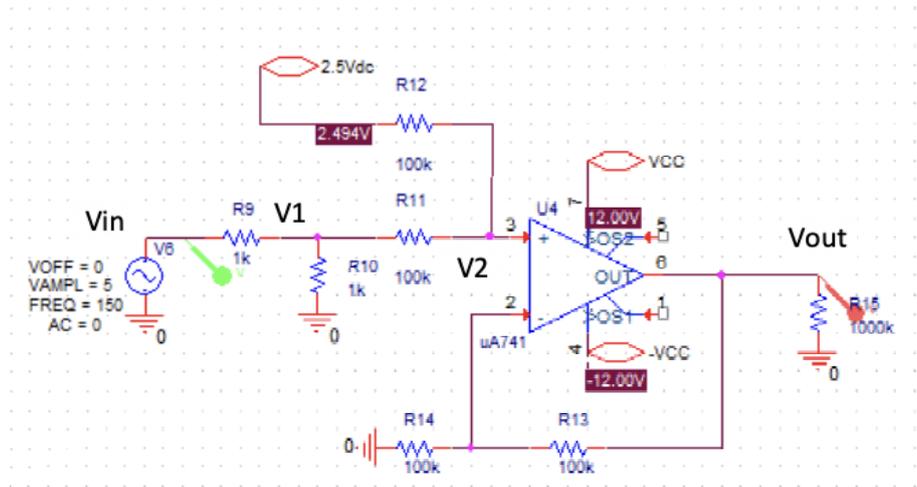


Figura 4-7 Circuito convertidor ± 5 Voltios a 0-5 Voltios

Aplicamos nudos y obtenemos tres ecuaciones, que podremos relacionar para obtener una ecuación que defina la tensión de salida (V_{out}) únicamente en función de la tensión de entrada (V_{in}).

$$\frac{V_{in} - V1}{1K\Omega} = \frac{V1}{1K\Omega} + \frac{V1 - V2}{100K\Omega} \quad (4.4)$$

$$\frac{V1 - V2}{100K\Omega} = \frac{V2 - 2.5V}{100K\Omega} \quad (4.5)$$

$$\frac{V_{out} - V2}{100K\Omega} = \frac{V2}{100K\Omega} \quad (4.6)$$

Se procede a obtener el valor de V_1 en función de V_2 , a partir de la ecuación 4.5.

$$\frac{V_1 - V_2}{100K\Omega} = \frac{V_2 - 2.5V}{100K\Omega}$$

$$V_1 - V_2 = V_2 - 2.5V$$

$$V_1 = 2 * V_2 - 2.5V$$

(4.7)

Una vez hemos obtenido el valor del punto V_1 , en función del punto V_2 (4.7), procedemos a sustituirlo en la ecuación 4.4. obteniendo la tensión de entrada en función del punto V_2 (4.8).

$$\frac{V_{in} - V_1}{1K\Omega} = \frac{V_1}{1K\Omega} + \frac{V_1 - V_2}{100K\Omega}$$

$$\frac{V_{in}}{1K\Omega} = 2 * \frac{V_1}{1K\Omega} + \frac{V_1}{100K\Omega} - \frac{V_2}{100K\Omega}$$

$$V_{in} = 2 * V_1 + \frac{V_1}{100} - \frac{V_2}{100}$$

$$V_{in} = \frac{201 * V_1 - V_2}{100}$$

$$V_{in} = \frac{401 * V_2 - 502.5V}{100}$$

$$V_{in} = 4.01 * V_2 - 5.025V$$

(4.8)

A continuación, se procede a obtener el valor de V_2 , en función de la tensión de salida (4.9), despejando de la ecuación 4.6.

$$\frac{V_{out} - V_2}{100K\Omega} = \frac{V_2}{100K\Omega}$$

$$V_{out} - V_2 = V_2$$

$$V_2 = 0.5 * V_{out}$$

(4.9)

Finalmente obtenemos la ecuación que relaciona la tensión de entrada con la tensión de salida (4.10), sustituyendo la ecuación 4.9, en la ecuación 4.8.

$$V_{in} = 4.01 * 0.5 * V_{out} - 5.025V$$

$$V_{out} = \frac{V_{in} + 5.025V}{2.005} \approx \frac{V_{in} + 5V}{2}$$

(4.10)

En el caso de que nuestra entrada V_{in} fuera de -5 Voltios, sustituyendo en la ecuación 4.10, obtendríamos una tensión de salida de 0 V.

$$V_{out} = \frac{V_{in} + 5V}{2} = \frac{-5V + 5V}{2} = 0V$$

Por otro lado, en el caso de que la entrada fueran 5V, sustituyendo en la ecuación 4.10, obtendríamos una tensión de salida de 5V.

$$V_{out} = \frac{V_{in} + 5V}{2} = \frac{5V + 5V}{2} = 5V$$

El siguiente paso consiste en realizar la simulación, en la *Figura 4-8* podemos observar la señal analógica de entrada, la cual se trata de una onda de $\pm 5V$, y la señal analógica de salida, la cual se trata de una señal de 0-5 V.

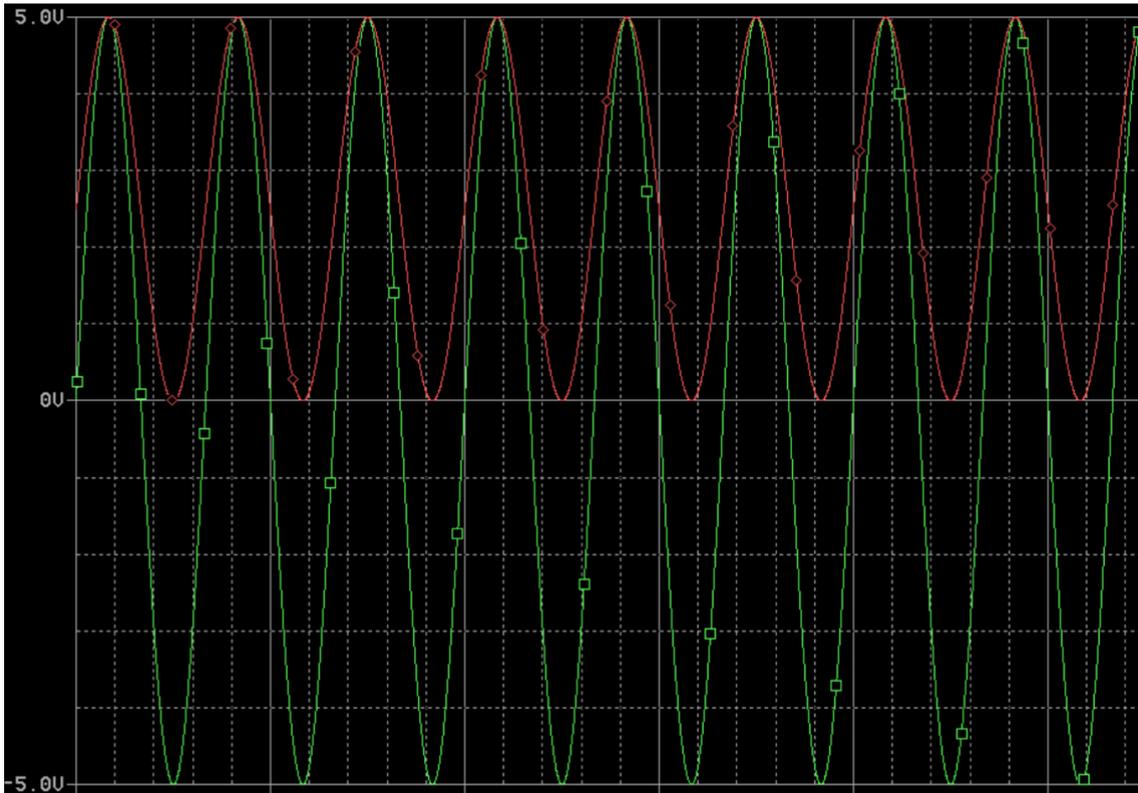


Figura 4-8 Salida (rojo) y entrada (verde) del circuito

4.4. ALIMENTACIONES DEL CIRCUITO

Nuestra fuente de alimentación tiene salidas de tensión de $\pm 12V$ y $5V$, las cuales van a ser utilizadas por un lado para alimentar nuestro operador y por otro lado, para alimentar los puntos de tensión constante.

En el caso de la alimentación de los operadores, la fuente de alimentación consta con tensiones de $\pm 12V$, tensión a la cual nuestro chip puede ser alimentado directamente.

En el caso del primer circuito, el circuito encargado de transformar una señal de tensión de $0-5$ a una tensión de $\pm 5V$, necesita una alimentación constante de $-5V$, como se puede observar, el problema radica en que nuestra fuente de alimentación solo nos aporta una única tensión de $-12V$, de la cual, tenemos que obtener dos tensiones de $-5V$, debido a que tenemos dos circuitos de este tipo. por lo que será necesario realizar el divisor que podemos observar en la *Figura 4-9*.

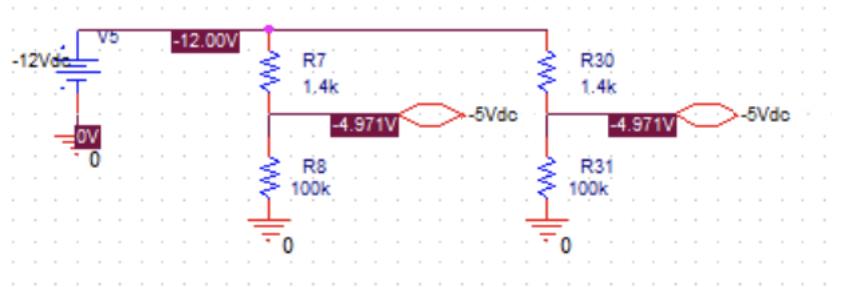


Figura 4-9 Divisores -5V

Como se puede observar obtenemos dos tensiones muy parecidas a los -5V.

En el segundo tipo de circuito, el circuito encargado de convertir una tensión de $\pm 5V$ a 0-5V, será necesario una tensión constante de 2.5V, esto se consigue con el divisor de tensión de la *Figura 4-10* con resistencias de $1K\Omega$.

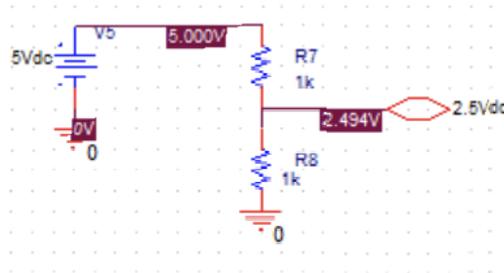


Figura 4-10 Divisor -2.5V

5. MICROCONTROLADOR

Todo nuestro sistema PID, va a estar implementado en un microcontrolador, en este proyecto se utilizará el microcontrolador Arduino Uno Rev3.

Arduino [7] es una compañía de desarrollo de software y hardware libres, que diseña y manufactura placas de desarrollo de hardware para la construcción de dispositivos digitales y otras aplicaciones que permitan la detección y control de objetos. El principal objetivo de Arduino es acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.

El proyecto Arduino (*Figura 5-1*) se origina en el año 2003, sus orígenes provienen del proyecto Wiring, el cuál se utilizaba como una herramienta de uso didáctico en el Interaction Design Institute Ivrea en Ivrea, Italia, el objetivo del proyecto era muy parecido al que tiene Arduino actualmente, era básicamente proporcionar una manera sencilla y económica, para tanto como profesionales, como principiantes, pudieran crear dispositivos que pudieran interactuar con su entorno a través de sensores y actuadores. Será en el año 2005 cuando aparezca el nombre Arduino, cuando se empiezan a industrializar y comercializar placas Arduino, ofreciéndolas a un coste bajo y con una gran facilidad de uso. En el año 2012, se incorporaron nuevos modelos de las placas que empleaban microcontroladores Cortex M3, ARM de 32 bits.



Figura 5-1 Logo Arduino

En la actualidad existe una gran variedad de placas Arduino, con diferentes microcontroladores AVR Atmel-8 bits, también la mayoría de ellas tienen un regulador lineal de 5V además de un oscilador de 16MHz, como es el caso de nuestra placa Arduino. Por otro lado, en la actualidad cabe destacar la gran cantidad de módulos externos que existen, existen módulos para cualquier tipo de montaje, con Arduino hoy en día se puede llevar a cabo cualquier tipo de proyecto, a un

bajo coste y con grandes fuentes de conocimiento en la red, en las cuales uno se puede apoyar para el desarrollo de su proyecto.

Generalmente el hardware consiste en un microcontrolador Atmel AVR, en nuestro caso se utiliza el ATmega328P, conectado sobre una placa de circuito impreso a la que se le pueden conectar placas de expansión a través de los puertos de entrada y de salida disponibles de la placa. Las placas de expansión complementan la funcionabilidad de la placa utilizada, agregando circuitería, sensores y otro tipo de módulos de comunicación externa.

La mayoría de las placas de Arduino puede ser alimentado por un puerto USB o un puerto Barrel Jack de 2.5 mm, es decir, lo podemos alimentar directamente conectándolo desde nuestra computadora, por otro lado, también se puede alimentar la placa a través del puerto Vin, aplicándole una tensión de entrada de entre los 6-20V, pero se le recomienda tener una alimentación que tenga un rango comprendido entre los 7-12V. En nuestro proyecto se alimentará la placa desde nuestra fuente de alimentación, utilizando los 12V de DC, que nos aporta.

Este microcontrolador tendrá la función de ejercer como regulador, tendrá la capacidad de actuar como regulador proporcional, regulador proporcional integrativo y como regulador proporcional integrativo derivativo. Por otro lado, también actuará como transformador de la señal analógica que nos aporte la tarjeta de adquisición de datos, es decir, cuando esté en modo PC, la cual ha sido transformada en una señal del rango de lectura del microcontrolador (0-5V), dicha señal se transformará en una señal PWM, para controlar el motor. La conmutación entre el modo planta y el modo PC, se realizará a través de una conmutación, utilizando una resistencia PULL UP, que más adelante se explicará.

El microcontrolador será capaz de realizar las lecturas sobre los potenciómetros de entrada y salida, como de los potenciómetros que establecerán el valor de las tres ganancias (K_p , K_d y K_i).



Figura 5-2 Arduino Uno REV3

Principales características de la placa Arduino Uno Rev3 [8]:

- Tensión de operación: 5V
- Tensión de alimentación 7-12 V (recomendable), pero también puede trabajar en un rango comprendido entre los 6-20V, en nuestro caso trabajará con una tensión de alimentación de 12V, proporcionados por la fuente de alimentación.
- Entradas/Salidas digitales: 14, 6 de los pines digitales se pueden utilizar como salidas PWM, vienen marcadas por el símbolo “ ~ ”
- Entradas analógicas: 6
- Corriente continua por cada pin de entrada/salida: 20mA
- Memoria flash: 32kB
- Memoria SRAM: 2KB
- EEPROM: 1KB
- Velocidad del reloj: 16MHz
- Largo: 68.6mm
- Ancho: 53.4 mm
- Peso: 25 gr

5.1. ENTORNO Y LENGUAJE DE ARDUINO

Lo primero que hay que saber de este entorno [9], es que se puede descargar de forma gratuita en la página oficial de Arduino, entre otras muchas páginas, aunque se recomienda que se realice la descarga siempre de páginas oficiales, por otro lado, este software esta disponible para diferentes tipos de sistemas operativos: Windows, MacOS y Linux.

El entorno de programación Arduino es conocido como IDE (Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación, por otro lado, puede dedicarse exclusivamente a un lenguaje de programación o puede utilizarse para varios.

El IDE es un entorno de programación que ha sido empaquetado como un programa, es decir, contiene un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI), además se puede cargar directamente el programa ya compilado a la memoria flash del hardware, siempre y cuando la placa Arduino este conectada con el ordenador vía USB.

5.1.1. Interfaz de la aplicación

Una vez abierta la aplicación podemos observar las distintas partes que componen nuestro programa.

- Menú de opciones: consiste en una barra situada en la parte superior del programa (*Figura 5-3*), en ella podemos encontrar algunas herramientas, como la apertura de nuevos archivos, así como su guardado, opciones de edición, de formato e insertar librerías, entre muchas otras opciones.



Arduino Archivo Editar Programa Herramientas Ayuda

Figura 5-3 Barra de opciones

- Botonera de accesos rápidos: consiste en una botonera, *Figura 5-4*, la cual nos permitirá, verificar el programa en busca de cualquier tipo de fallo de lenguaje, subir nuestro programa a nuestra placa Arduino, crear un nuevo programa, abrir un programa, salvar el programa actual y finalmente la apertura del monitor serie.

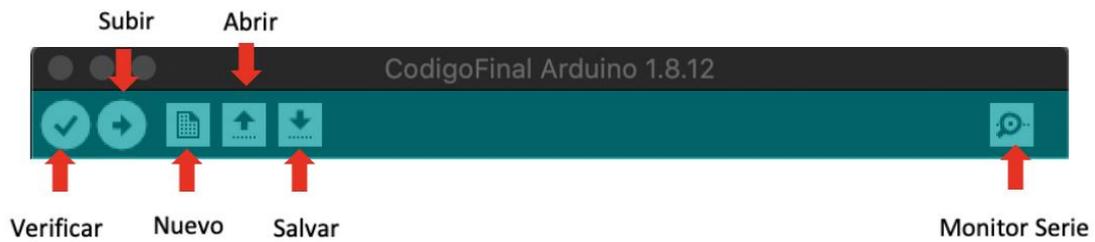


Figura 5-4 Botonera de accesos rápidos

- Editor de código: en esta parte del programa será donde se escriba el código, *Figura 5-5*, conocido como 'sketch'.

```

sketch_may07b
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
    
```

Figura 5-5 Editor de código

- Área de mensajes: es una zona intermedia donde nos muestra el estado actual del código, si se encuentra compilando si se encuentra subiéndose a la placa etc.
- Consola: en ella observaremos los posibles fallos cometidos a la hora de compilar el programa, por otro lado, en caso de que se haya compilado de forma correcta nos mostrara el tamaño que ocupa nuestro código de la memoria flash disponible de nuestra placa Arduino (*Figura 5-6*).

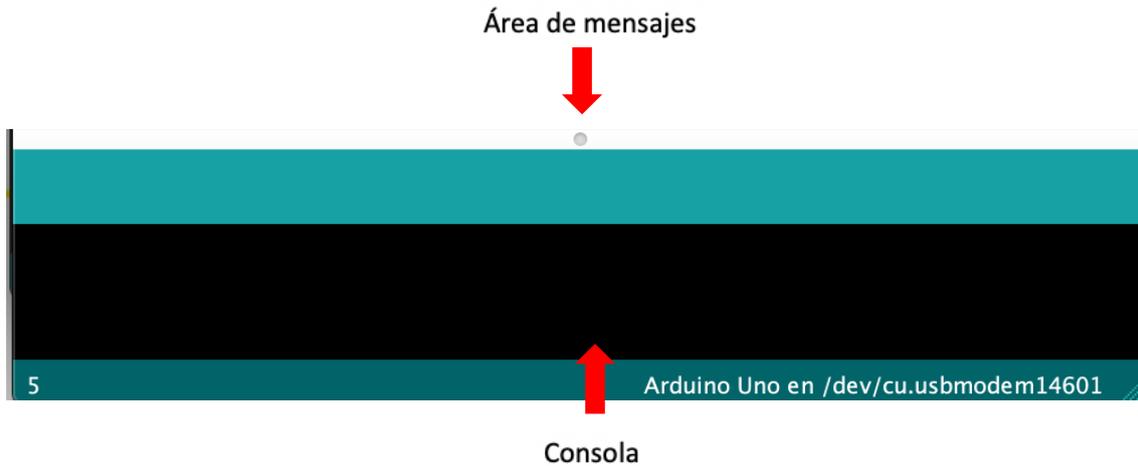


Figura 5-6 Consola y área de mensajes

5.1.2. Estructura de la programación

Nuestro programa deberá de constar de dos partes [10], en las cuales se deberán de incluir las diferentes variables, asignaciones de puertos y procesos que se deberán de llevar a cabo para ejercer la función requerida. En la *Figura 5-7*, podemos observar perfectamente diferenciadas las dos partes que componen nuestro código.

```
sketch_may07b
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Figura 5-7 Estructura del programa

Por un lado, encontramos el “void setup()” esta parte del programa solo se va a ejecutar una vez, al principio, esta parte se encargará de la configuración de los pines y de la comunicación.

Por otro lado, podemos observar la parte del programa “void loop()”, esta parte se va a ejecutar de forma repetitiva, en ella deberemos de programar nuestro regulador PID.

Debemos de tener en cuenta que nuestro programa siempre debe de tener ambas partes, y que la declaración de las variables que se van a utilizar se debe de realizar antes del “void setup()”.

5.1.3. Monitor serie

El monitor serie consiste en una ventana (*Figura 5-8*), la cuál se podrá utilizar para enviar datos desde nuestro ordenador hasta la placa así como para observar los valores de las diferentes variables de nuestro programa en cualquier momento.

Para obtener los datos de alguno de los puertos digitales o analógicos de Arduino y observemos en el monitor serie se deberá de utilizar el comando "Serial.println()" en el "void loop", con el puerto o variable del cual se quiere que se muestre el valor, además previamente deberemos de haber introducido en el "void setup()" el comando "Serial.Begin()", entra paréntesis se deberán de incluir el número de bits, llamados baudios.

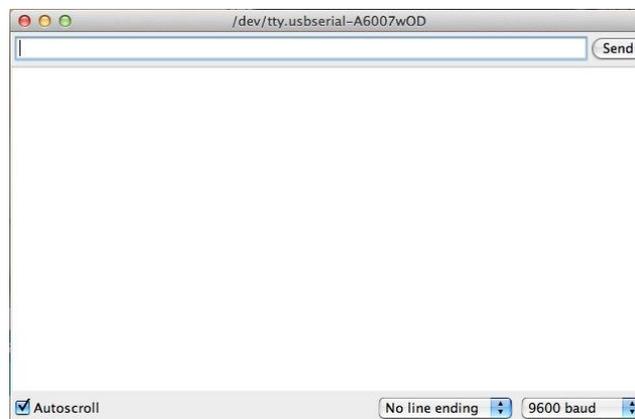


Figura 5-8 Monitor Serie

5.1.4. Comandos básicos

Los principales comandos que se van a utilizar en este programa van a consistir en la lectura analógica de los diferentes potenciómetros, así como la lectura lógica de el conmutador entre modo planta y modo pc, por otro lado, tendremos que escribir valores digitales de salida, y controlar una señal PWM, que a su vez controla la acción sobre un motor eléctrico.

A continuación, vamos a ver alguno de los comandos más importantes y su función:

- "analogRead()": lectura del valor de un puerto analógico.
- "digitalRead()": lectura del valor de un puerto digital.
- "digitalWrite()": escritura del estado digital del puerto, valor alto (1) o bajo (0)
- "analogWrite()": Escritura de la señal PWM

5.2. ENTRADAS ANALÓGICAS

Nuestro dispositivo Arduino consta de 6 entradas analógicas ([11] y [12]), las cuales se encuentran en la parte inferior derecha, de la *Figura 5-9*, vienen indicadas por el término “ANALOG IN”, se tratan de las entradas A0-A5.



Figura 5-9 Salidas analógicas Arduino

El rango de trabajo de estas entradas analógicas vendrán comprendidas entre lo 0 V y los 5 V, cualquier alimentación superior podría dañar la placa. En este proyecto no existirá dicho peligro, debido a que los potenciómetros de los cuales se va a realizar una lectura y el conmutador que van a ir conectados a los pines analógicos, van a estar alimentados por la fuente de alimentación, la cual nos entregará 5 Voltios. Por otro lado, el pin restante será para la conexión en modo PC, el cual en ningún momento entregará fuera del rango 0-5 Voltios.

En el propio microcontrolador dichas entradas analógicas se van a transformar en una medición digital codificada con un número concreto de bits, para ello la placa utiliza un conversor analógico digital (ADC). Estos puertos en realidad no miden el valor analógico con todos sus decimales de sus conexiones si no que se clasifican dentro de 2^n niveles, que definen 2^N-1 intervalos. El ancho de este intervalo, medido en mV, es la precisión de la señal, por lo que, cuanto mayor sea el numero de bits, mayor será el número de intervalos, menor será el ancho del intervalo y tendrá una mayor precisión de medida.

En el caso de nuestro microcontrolador Arduino UNO Rev3, las entradas analógicas disponen de 10 bits de resolución, por lo que tenemos disponibles 1024 niveles lógicos, con una precisión de la medición de ± 2.44 mV. En las señales utilizadas el valor máximo de las entradas analógicas van a ser 5V ocupando toda la resolución.

Sustituyendo en la ecuación 5.1, en el caso de que la señal analógica nos este dando 5 voltios en el momento de lectura, obtendremos que su lectura en la placa será igual a 1023, el máximo valor, en el caso de que la señal analógica se tratase de una señal de 0 V, habría una lectura 0.

$$Lectura = \frac{V * 1023}{5V} \tag{5.1}$$

Donde:

- V = la tensión de entrada del pin analógico en voltios.
- Lectura = lectura de la placa (0-1023).

Para leer las señales analógicas de los puertos se utilizará el comando "analogRead()", indicando el puerto entre los paréntesis. Por ejemplo, "analogRead(A0)".

La distribución de los pines analógicos para la lectura de todas las variables va a ser la siguiente:

- Pin A0: Se conectará la señal proveniente del potenciómetro de la entrada, realizando la lectura de la señal de entrada.
- Pin A1: Se conectará la señal proveniente del potenciómetro de la salida, realizando la lectura de la señal de entrada.
- Pin A2: se conectará la señal proveniente del potenciómetro de la ganancia proporcional (Kp), realizando la lectura de dicha ganancia.
- Pin A3: se conectará la señal proveniente del potenciómetro de la ganancia integrativa (Ki), realizando la lectura de dicha ganancia.
- Pin A4: se conectará la señal proveniente del potenciómetro de la ganancia derivativa (Kd), realizando la lectura de dicha ganancia.
- Pin A5: se conectará la señal proveniente del convertidor de tensiones, que convierte la señal de la tarjeta de adquisición de datos.

5.3. PUERTOS DIGITALES

Las señales analógicas son aquellas señales las cuales tienen únicamente dos estados [13], un estado "ON", es decir, encendido, el cual se representa a través de un 1 o un estado "OFF", o apagado, el cual se representa con un cero (*Figura 5-10*).



Figura 5-10 Señal digital

La placa Arduino nos dará o reconocerá para los valores 1, un valor de 5V de tensión, por el otro lado en caso de ser un valor de 0 se corresponderá con un valor de 0V.

Este tipo de puertos les daremos varios usos, por un lado los utilizaremos para determinar el sentido de giro de nuestro motor en función de las indicaciones del código, por otro lado, también utilizaremos la señal PWM, que nos permite generar algunos de estos puertos digitales, estos son los que aparecen señalados y finalmente conmutaremos entre el modo PC o el modo planta, a través de una resistencia pull up.

Los pines digitales de nuestra placa Arduino se encuentran en la parte superior de la *Figura 5-11*, se tratan de 14 pines, enumerados del 0 al 13, algunos de esos pines tienen la capacidad de ser señales PWM, son los pines que tiene el símbolo "~" (pines: 3, 5, 6, 9, 10, 11).



Figura 5-11 Salidas Digitales Arduino

5.3.1. Conmutación entre el modo PC o el modo planta

Esta parte consiste en generar un conmutador [14], para poder seleccionar en el modo que vamos a utilizar la planta, modo pc, aportando y simulando el regulador desde un ordenador o el modo planta, para ello utilizamos la resistencia Pull Up.

Este método de creación de un conmutador constará de una resistencia de $4.7K\Omega$ y de un interruptor y alimentaremos el sistema con 5V, conectándolo al pin digital número "2" (Figura 5-12).

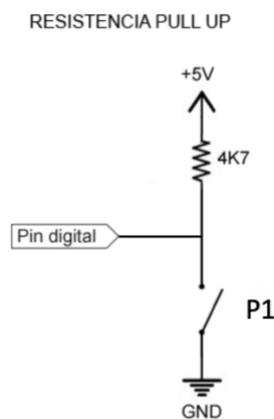


Figura 5-12 Conmutador

Cuando el circuito se encuentra en reposo, como podemos observar en la imagen superior (*Figura 5-12*), la caída de tensión es de 5V, es decir, tiene un valor alto (1), en el caso de que conmutemos la posición de P1, entonces la caída de tensión es 0V, aportando al pin digital un valor bajo.

Programaremos Arduino para que sea capaz de leer el valor de la entrada digital del puerto 2, a través del comando "digitalRead(2)", a continuación generaremos una condición en caso de que el valor del pin digital sea 1, se ejecutará modo planta leyendo los valores de las entradas analógicas A0-A4, obteniendo el valor de entrada y de salida, así como el valor de las ganancias seleccionadas, y se ejecutará con esos datos el regulador PID, por el caso contrario, si el pin digital 2, valiese 0, entonces se ejecutaría modo PC, habilitando únicamente la lectura del pin analógico A5, el cual, aportará la señal correspondiente para accionar el motor.

5.3.2. Sentido de giro del motor

Se utilizaran las señales digitales para cambiar el sentido de giro del motor de salida, el sentido de giro vendrá en función del valor de el regulador.

Debido a que el regulador va a estar normalizado, es decir va a tener un valor comprendido entre -1 y 1, el sentido de giro del motor dependerá de dicho valor, para ello se crearan unas condiciones, siempre que el valor del PID sea superior a cero ($PID > 0$), entonces el motor se le hará girar hacia la izquierda, en el caso contrario, en el caso de que el valor de PID sea inferior a cero ($PID < 0$), el motor girará en sentido contrario, hacia la derecha.

Esto se consigue conectando dos salidas digitales hacia un driver controlador de motores. Estos dos puertos digitales de salida van a ser el puerto 7 y el puerto 8, siendo el pin 7 correspondiente al sentido de giro izquierda y el pin 8 correspondiente al sentido de giro derecha.

Siempre que se vaya a hacer girar en un sentido u otro el motor, se le deberá de dar un valor alto, es decir 1, a el sentido de giro que se quiera girar el motor, por otro lado, se le deberá de dar un valor bajo al otro pin de salida. Por ejemplo, si queremos hacer girar el motor a la derecha se le deberá de dar un valor alto al pin digital 8, y deberemos darle un valor bajo al pin número 7, correspondiente a la izquierda.

Tabla 3 Valor digital de los puertos para determinar el sentido de giro del motor CC

	Puerto 8	Puerto 7
Derecha	HIGH	LOW
Izquierda	LOW	HIGH

Escribimos las señales digitales utilizando el comando "digitalWrite()", seleccionamos el puerto al cual queremos enviar esa señal y por otro lado seleccionamos el valor de la señal. Por ejemplo en el caso de que quisiéramos representar el giro derecha:

digitalWrite(8,HIGH)

digitalWrite(7,LOW)

Po el caso contrario, si el giro que se desea es hacia la izquierda:

digitalWrite(8,LOW)

digitalWrite(7,HIGH)

5.3.3. Señal PWM

La señal PWM ([15] y [16]) se utilizará para transmitir la señal del regulador al motor y hacerlo girar, por otro lado, también transmitirá la señal proveniente del modo PC.

Es un método para generar una señal analógica a raíz de la utilización de una señal digital. Las señales PWM constan de dos componentes que definen su comportamiento, por un lado el ciclo de trabajo, el cual describe la cantidad de tiempo que la señal esta en un estado alto (HIGH) y lo define como un porcentaje del tiempo total que se tarda en completar el ciclo, es decir es el ancho del pulso. El otro componente que define nuestra señal PWM es la frecuencia, la cual determina la rapidez con la que la señal PWM completa el ciclo, y por otro lado también, determina como de rápido cambia entre los dos estados, entre el estado alto y el bajo. Al realizar el apagado y encendido de una señal digital a una velocidad lo suficientemente alta, y con un

cierto ciclo de trabajo, la salida parecerá comportarse como una señal analógica de voltaje constante.

La ecuación que define la proporción de tiempo que está encendida la señal, respecto al total del ciclo de trabajo, se la denomina duty cycle y se expresa en tanto por ciento (5.2). Nuestro sistema estará comprendido entre $\pm V_{cc}$, en nuestro caso la placa Arduino nos aporta señales comprendidas entre los 0 y los 5 voltios. Por lo que la ecuación será igual:

$$VPromedio = (V_{cc(+)} - V_{cc(-)}) * \frac{DutyCycle}{100} \tag{5.2}$$

Despejando el ciclo de trabajo (DutyCycle) de la ecuación 5.2, obtenemos la ecuación 5.3.

$$DutyCycle = 100 * \frac{VPromedio}{V_{cc(+)} - V_{cc(-)}} \tag{5.3}$$

Donde:

- $V_{cc(+)} = 5$ voltios
- $V_{cc(-)} = 0$ voltios
- DutyCycle = ciclo de trabajo
- VPromedio = tensión de salida deseada

Por ejemplo, se desea obtener una señal de 0.5 voltios, por lo que nuestro Duty Cycle para obtener dicha señal será, sustituyendo en la ecuación 5.3, igual a un 10%.

$$DutyCycle = 100 * \frac{VPromedio}{V_{cc(+)} - V_{cc(-)}} = 100 * \frac{0.5 V}{5 V - 0 V} = 10\%$$

El ciclo de trabajo de nuestra señal será de un 10%, lo que quiere decir que solo estará encendido un 10% el 90% restante se encontrará apagado. En la *Figura 5-13*, podemos observar la representación de dicha señal, entre otros ejemplos, donde podemos observar el tiempo que está la señal encendida y apagada en función del tiempo de trabajo.

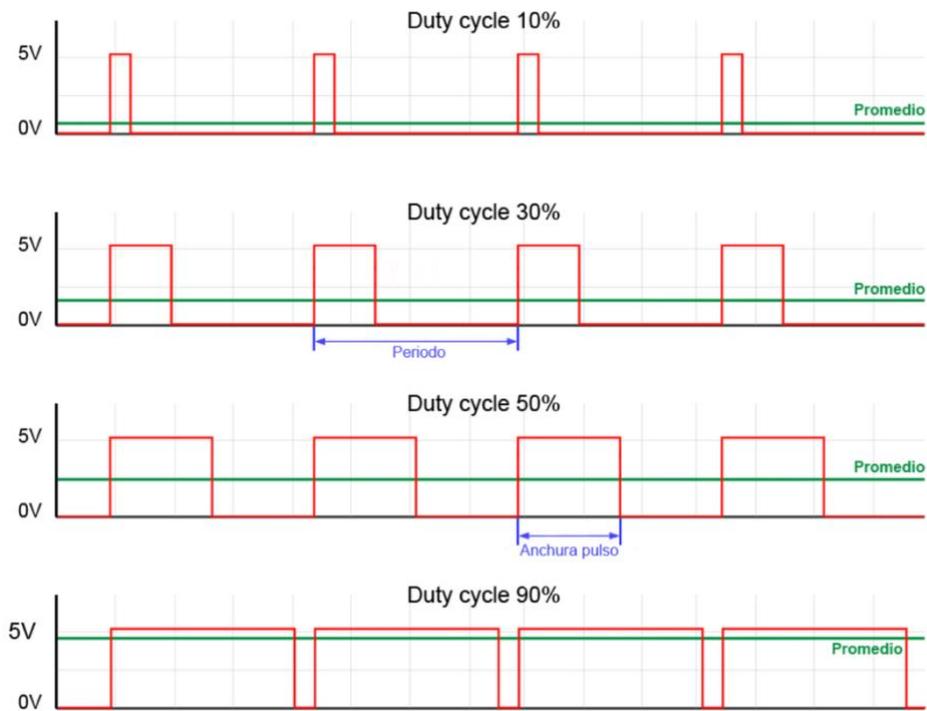


Figura 5-13 Señal PWM

Arduino implementa por hardware salidas PWM en varios de sus pines, que aparecen con la identificación " ~ " junto al número del pin, en nuestra placa Arduino Uno Rev3, son los pines 3, 5, 6, 9, 10 y 11 (*Figura 5-14*). En caso que se necesitase, se podría emular señales PWM a través de software, pero supone una carga de trabajo, además de que no nos es necesario.

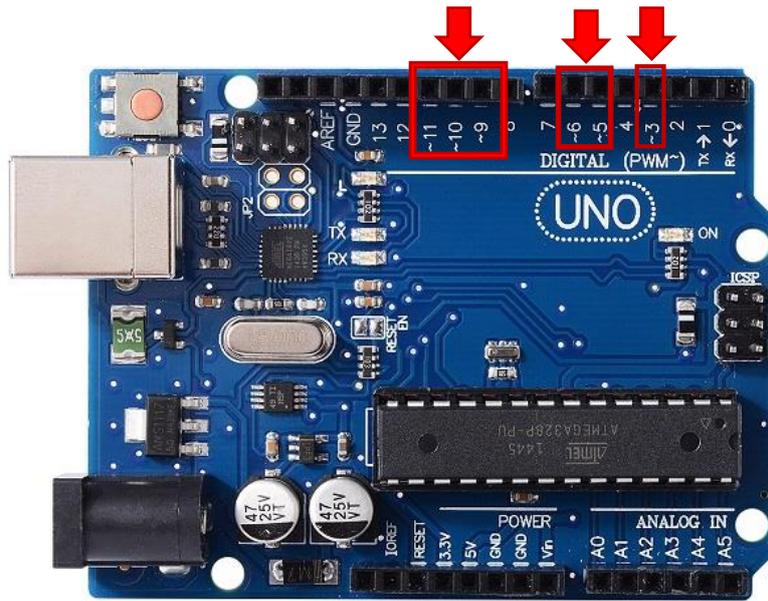


Figura 5-14 Salidas PWM Arduino

Arduino Uno Rev3 cuenta con 8bits en sus salidas PWM, significa que tenemos 256 niveles de señal, es decir, que el ciclo de trabajo o Duty cycle, va a estar comprendido entre un valor de 0 y un valor de 255, correspondiendo el valor 0 a un valor de ciclo de trabajo del 0% y el 255 a un valor de trabajo del 100%.

El valor de la señal PWM, vendrá definida por le ecuación 5.4.

$$SeñalPWM = 255 * \frac{DutyCycle(\%)}{100\%} \tag{5.4}$$

La frecuencia de las salidas PWM de la placa Arduino Uno Rev3, tienen un valor de 490Hz, exceptuando los pines digitales 5 y 6, los cuales tienen una frecuencia de 980Hz, en nuestro programa utilizaremos el pin digital número 6.

6. PUENTE EN H

El puente en H [17] se utilizará para controlar el motor eléctrico de corriente continua, tanto el sentido de su giro como la velocidad de giro, a través de las salidas digitales de la placa Arduino y de la señal PWM que nos aporta. Para ello utilizaremos el controlador de motores L298N.

Los puentes en H están disponibles como circuitos integrados, o se pueden construir con componentes comunes.

6.1. ESTRUCTURA

El término puente en H se deriva de la representación gráfica típica del circuito. Un puente en H se construye con cuatro interruptores, los cuales serán los encargados de permitir el paso de la corriente, para que el motor gire en un sentido o en otro. Su estructura la podemos observar en la *Figura 6-1*.

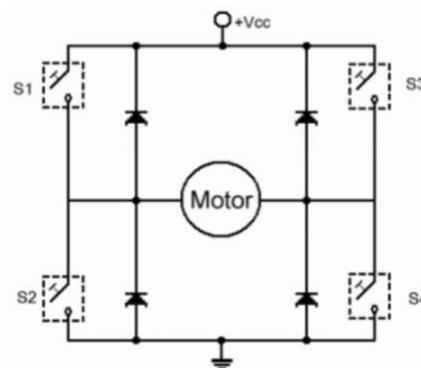


Figura 6-1 Puente en H

Tienen tres posiciones posibles de funcionamiento, cuando los interruptores S1 y S4, están cerrados y los otros dos interruptores se encuentran abiertos (caso A de la *Figura 6-2*), en ese instante el motor se ve a travésado por la tensión en una de las direcciones, haciéndole girar en un sentido. En el caso contrario (caso B de la *Figura 6-2*), en el caso que sean los interruptores S2 y S3, los que se encuentran cerrados y los otros dos abiertos, la tensión de alimentación del motor cambia de sentido, por lo que el motor cambia el sentido de giro.

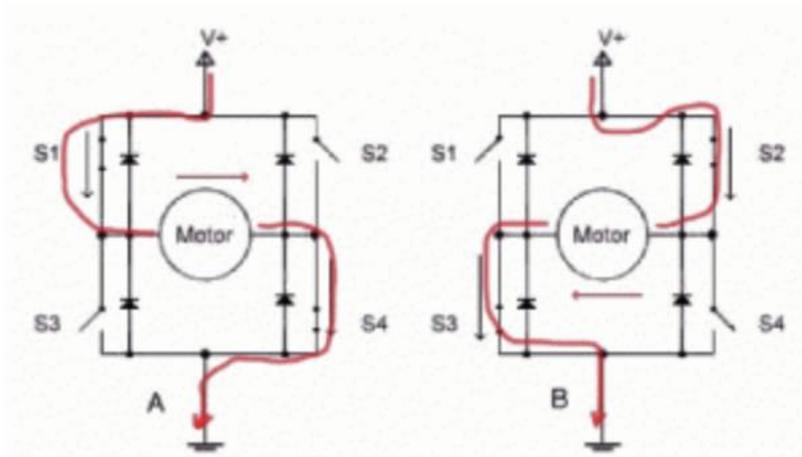


Figura 6-2 Estados de funcionamiento del puente en H

La otra posición de funcionamiento es cuando todos los interruptores se encuentran en la posición de abierto, no alimentando con tensión el motor y no provocando ningún tipo de movimiento.

De ningún modo los interruptores S1 y S3 o S2 y S4, deberán de estar en la posición cerrados a la vez, debido a que causaría un corto circuito en la fuente de tensión, este efecto se conoce como "Shoot-through".

6.2. PUENTE EN H CON TRANSISTORES

Una de las maneras de implementar un puente en H es utilizando un transistor como interruptor (Figura 6-3), es una opción rápida, robusta, barata y fácil de diseñar y controlar, sus principales inconvenientes son la baja potencia que soportan y su muy baja eficiencia.

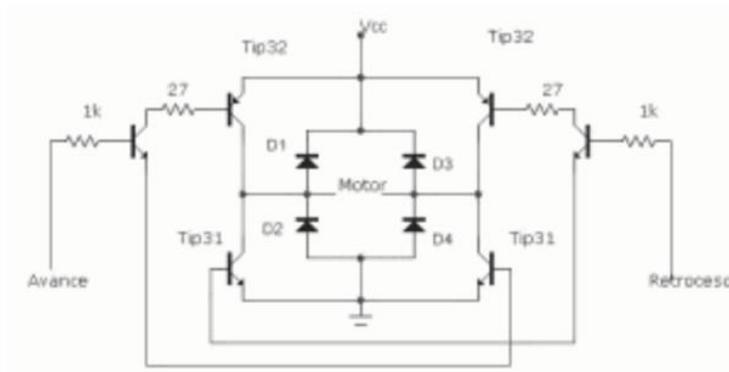


Figura 6-3 Puente en H con transistores

En este caso se puede observar como los transistores son equivalentes a los interruptores, hay un circuito alimentando el motor, interrumpido por la conexión al colector y al emisor del transistor, por otro lado, podemos observar como la fuente alimenta a la base del colector.

El transistor actúa como una puerta desde un lado a otro, en este caso desde el colector al emisor, la base si no recibe electricidad, la interrupción se mantendrá cerrada en el caso contrario, que reciba una tensión se abrirá.

Las características de los componentes vendrán determinada por las capacidades del puente en H, su tensión y amperaje de trabajo, las características de la señal con la que se trabaja etc. También se debe tener en cuenta las características del motor que se va a controlar.

6.3. PUENTE EN H CON CIRCUITOS INTEGRADOS

A día de hoy el uso de estos circuitos integrados esta muy extendido en el control de los motores eléctricos, en nuestro caso vamos a utilizar el controlador de motores L298N, el cuál utiliza el circuito integrado L298.

El circuito integrado L298 (*Figura 6-4*) contiene dos puentes en H que son capaces de trabajar hasta con 2 A de corriente, por lo que nos es muy útil para controlar motores de bajo o media potencia, por otro lado tiene dos etapas de salida de potencia.

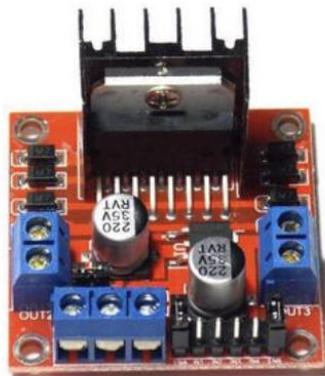


Figura 6-4 Controlador de motores L298N

Esta placa lleva integrada el chip L298N y nos permitirá el control de dos motores de corriente continua, aunque solo necesitamos el control de uno de ellos

Las principales características de esta placa son:

- Contiene el circuito integrado L298N
- Doble puente en H
- Tiene un buen rendimiento ante posibles interferencias
- Intensidad de trabajo de 2 A con un máximo de 3 A
- Máxima potencia de 25 W
- Incorpora un regulador 78M05, es un estabilizador de tensión.
- Incluye diodos de protección y filtros por condensador
- Dos tipos de alimentación 5-12 V o 12 –35 V, en función de las características del motor
- Peso de 26 gr.

6.3.1. Conexiones del driver controlador L298N

La placa del driver controlador tiene diferentes conexiones. Por un lado, estás las entradas digitales IN1, IN2, IN3 e IN4, estas entradas se van a utilizar para controlar el sentido de giro del motor, por otro lado están las salidas OUT1, OUT2, OUT3 y OUT4, estas salidas estarán conectadas directamente con el motor de corriente continua, las entradas digitales IN1 e IN2, vienen relacionadas directamente con la salida OUT1 y OUT2, van a ser las encargadas de determinar el giro de dicho motor, por otro lado ocurre exactamente lo mismo con IN3 e IN4, que determinaran el giro del motor de las salidas OUT3 y OUT4.

Por otro lado, encontramos los puertos de entrada ENA y ENB, el primero controlara la velocidad del giro del motor de las salidas OUT1 y OUT2 y el otro pin (ENB) controlará la velocidad de giro del otro motor conectado a las otras salidas. Estos se encuentran cortocircuitados con un jumper, en nuestro caso queremos controlar la velocidad de giro de nuestro motor, por lo que lo deberemos de quitar, para poder conectarlo con nuestra señal PWM que se origina en nuestro Arduino.

En nuestro caso vamos a utilizar los pines IN1 e IN2 para conectar nuestras señales digitales que nos va a aportar Arduino, con ello controlaremos el sentido de giro de motor, por otro lado,

conectaremos ENA, para controlar la velocidad de giro del motor, quitando previamente el jumper que la cortocircuita.

Finalmente, existen las conexiones de alimentación, hay dos modos de alimentar esta placa que vamos a explicar a continuación.

En la *Figura 6-5*, podemos observar las diferentes partes que componen nuestro controlador.

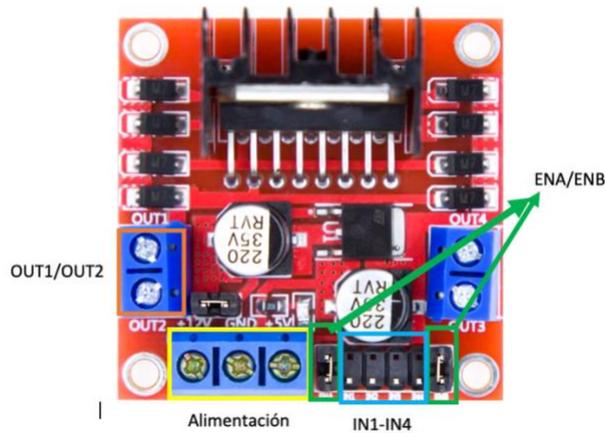


Figura 6-5 Pines controlador de motores L298N

6.3.2. Alimentación del driver

Como ya comentábamos anteriormente hay dos maneras de alimentar esta placa que va a controlar nuestro motor, en función de la tensión de alimentación que necesite nuestro motor.

La primera opción (*Figura 6-6*) consiste en alimentar nuestro motor con una fuente de alimentación de entre los 5 y los 12 voltios. Para ello conectaremos la alimentación positiva a nuestro pin de alimentación con el símbolo +12V, y por otro lado conectaremos la tierra o el polo negativo de la fuente de alimentación a el puerto GND. Nos deberemos de asegurar que el jumper que se encuentra justo en la parte superior de los puertos de alimentación se encuentre conectado. Esta forma de alimentar el circuito nos aporta una tensión de 5 voltios de salida, con la cual podemos alimentar otro dispositivo, a través del puerto +5V.

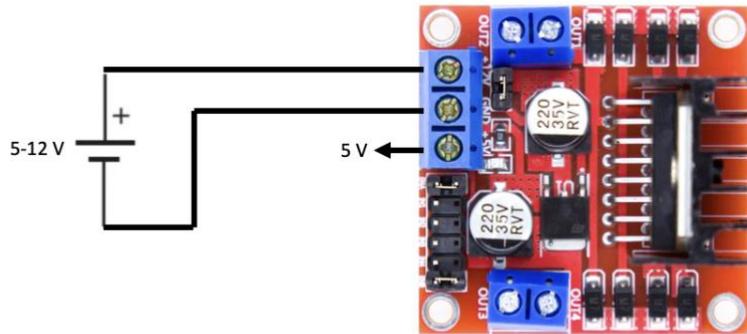


Figura 6-6 Alimentación tipo 1 controlador de motores L298N

La segunda opción (Figura 6-7) para alimentar nuestra placa consiste en utilizar fuentes de alimentación entre los 12-35 V, esta alimentación se realizará para los motores con tensiones de alimentación superiores. Para llevar a cabo esta segunda opción nos deberemos de asegurar que el jumper superior se encuentra desconectado, debido a que dañaríamos el sistema. Alimentamos por un lado el puerto +12V con nuestra fuente de alimentación que está entre los 12-35V, por otro lado conectamos la tierra y también procedemos a conectar el puerto de +5V con una fuente de 5V para alimentar la parte lógica del circuito.

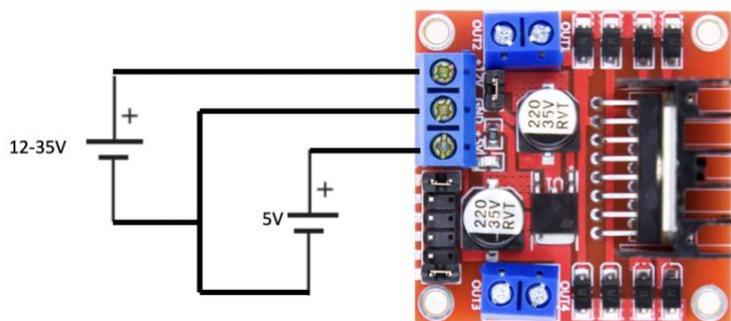


Figura 6-7 Alimentación tipo 2 controlador de motores L298N

Esta segunda alimentación es la que vamos a utilizar, debido a que nuestro motor se trata de un motor de 12V de corriente continua y teniendo en cuenta las posibles caídas de tensión de la placa, y por otro lado que nuestra fuente de alimentación tiene alimentaciones de 12V y de 5V disponibles, alimentaremos el circuito lógico de la placa con la conexión de 5V y el resto con la tensión 12 V, con lo que podemos asegurar que al motor le van a llegar cerca de 12V, para que pueda funcionar con normalidad.

7. MOTOR CORRIENTE CONTINUA

El motor de corriente continua [18] consiste en una máquina que transforma la energía eléctrica en mecánica, transformándola en un movimiento rotatorio transmitido por un eje de salida.

Este fue descubierto por el científico Hans Christian Oesterd, que comprobó su teoría colocando un espiral doble alrededor de una aguja, hizo que la atravesará una corriente y la aguja unida a un imán se movió y así consiguió demostrar la relación entre el magnetismo y la electricidad.

Los motores de corriente continua están compuestos por dos parte: por el rotor y por el estator, una de las principales desventajas es que el mantenimiento en grandes motores es muy costoso, por otro lado las principales ventajas son: limpieza, comodidad, fiabilidad, diferentes tamaños y velocidades, además de poder encontrar motores con tensiones de alimentación muy variadas.

El motor a utilizar en este proyecto consiste en un motor con una reducción den la velocidad de salida a 200 RPM (*Figura 7-1*), para que la salida no sea excesivamente rápida, este estará alimentado por 12 V, irá conectado directamente a nuestro driver controlador de motores.



Figura 7-1 Motor CC

Algunas características más del motor:

- Alimentación: 12 V
- Velocidad con reducción a 200 RPM
- Par de torsión de la carga: 10 Kg.cm
- Potencia eléctrica: 2 W
- Potencia en caballos: 2 CV
- Peso: 81.6 gr

- Carcasa metálica
- Diámetro exterior: 20 mm
- Diámetro del eje: 10 mm

7.1. PARTES DEL MOTOR CC

El motor de corriente continua está formado por diferentes partes [19]:

- Estátor: es la parte fija del motor responsable del establecimiento del campo magnético de excitación. Este está formado por un número par de polos inductores, los cuales están constituidos por un núcleo y por unas expansiones en sus extremos y se encuentran en la carcasa. El devanado del inductor son unas bobinas que se encuentran alrededor de los polos, generalmente se usa hilo de cobre, estos al ser alimentados por una corriente continua, generan un campo inductor, presentando polaridades norte y sur.
- Rotor: consiste en la parte móvil del motor, que proporciona el par para mover la carga. Este está formado por un conjunto de bobinas denominadas bobinas inducidas o inducido, que van arrolladas sobre las ranuras de un núcleo de hierro que recibe el nombre de inducido.
- Colector delgas: son un conjunto de láminas de cobre, las cuales se encuentran aisladas entre sí, formando el colector y a las cuales se sueldan los extremos de las bobinas inducidas. Esta parte establece la conexión eléctrica entre la parte fija o estator y las bobinas de la parte móvil o rotor.
- Escobillas: se encuentran montadas sobre los portaescobillas, están en contacto permanente con el colector y suministran la corriente eléctrica a las bobinas inducidas.
- Entrehierro: es el espacio que existe entre el estator y el rotor, es por donde el flujo magnético pasa de uno a otro.

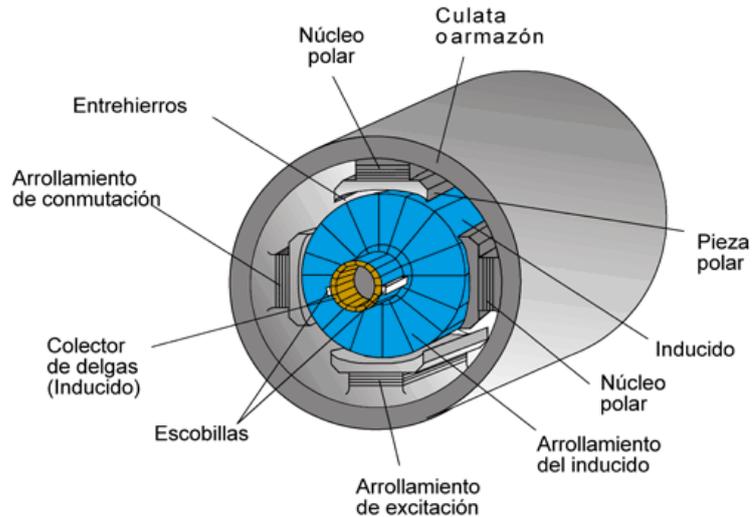


Figura 7-2 Partes motor CC

7.2. FUNCIONAMIENTO DEL MOTOR CC

El motor CC funciona de la siguiente manera [20], el estator suministra un campo magnético constante y el rotor que es la parte rotaria es una simple bobina, el rotor está conectado a una fuente de energía continua a través de un par de anillos de conmutación. Cuando la corriente fluye a través de la bobina una fuerza electromagnética es inducida sobre esta de acuerdo a la ley de Lorentz, por lo que la bobina comienza a girar, a medida que la bobina gira los anillos del colector se conectan a la fuente de alimentación, con polaridad invertida, como resultado en el lado izquierdo de la bobina, la electricidad siempre fluirá alejándose, y en el lado derecho la electricidad fluirá acercándose. Esto asegura que la acción del par es siempre en el mismo sentido durante el movimiento, de este modo la bobina continuará girando.

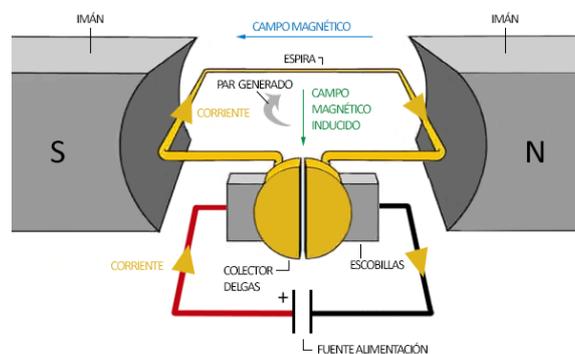


Figura 7-3 Funcionamiento motor CC

El motor tendrá dos estados de funcionamiento con carga o en vacío. Un motor funciona con carga cuando este arrastra cualquier objeto o elemento, es decir, cuando debe de soportar cualquier resistencia externa, lo que le obliga a absorber energía mecánica. En este caso el par resistente, que es la oposición al movimiento del eje, se debe a factores internos y externos. Este será nuestro caso, pues nuestro motor deberá de pilotar una salida que esta en su propio eje y no le aporta apenas carga, pero por otro lado deberá de ejercer la transmisión a través de engranajes del giro, para que un potenciómetro sea capaz de determinar la posición actual que nos proporciona el motor.

El otro estado de los motores de corriente continua es el funcionamiento en vacío, a diferencia del anterior, es cuando no tenemos ningún objeto pilotado por el giro del eje, es decir, el eje está girando libremente y en este caso el par resistente se debe únicamente a factores internos como rozamientos.

7.3. TIPOS DE MOTORES CC

7.3.1. Motor serie

En este tipo de motores las bobinas inductoras y las inducidas están conectadas en serie, la conexión forma un circuito en serie en el que la intensidad absorbida por el motor al conectarlo a la red es la misma, tanto por el estator como por el rotor.

Este tipo de motor puede desarrollar un elevado par motor de arranque, si se disminuye la carga del motor, disminuye la intensidad de corriente absorbida y aumentando la velocidad del motor, este proceso puede ser peligroso, pues la velocidad aumenta de forma brusca y finalmente otra característica de este tipo de motor es que sus bobinas tienen pocas espiras, pero una gran sección.

7.3.2. Motor Shunt o de derivación en paralelo

En este caso las bobinas inductoras van conectadas en paralelo con la inducidas. La corriente absorbida por el motor, se dividirá en dos, una de las corrientes circulará por las bobinas inducidas y otra de las corrientes por la bobina inductora. El circuito de excitación se encuentra a la misma tensión que el inducido.

Algunas de las características de estos motores son que en el arranque, el par motor es menor que el del motor serie, por otro lado, si la intensidad de corriente absorbida disminuye y el motor se encuentra en vacío, la velocidad de giro apenas varía, es una construcción bastante estable y finalmente si el par motor aumenta, la velocidad de giro apenas se ve afectada, disminuyendo muy poco.

7.3.3. Motor Compound o compuesto

Este tipo de motor resulta de la combinación del motor serie y el motor Shunt, debido que una de las bobinas inductoras está en serie con el inducido y la otra se encuentra en paralelo a él.

Una parte de la intensidad de corriente absorbida circula por la bobinas inducidas y por una de las inductoras, mientras que el resto de la corriente recorre la otra bobina inductora.

Este motor se caracteriza por tener un elevado par de arranque, pero no corre el peligro de ser inestable cuando trabaja en vacío, como ocurre con el motor serie, aunque puede llegar a alcanzar un número de revoluciones muy alto.

Finalmente vamos a observar una imagen de los tres tipos de motores:

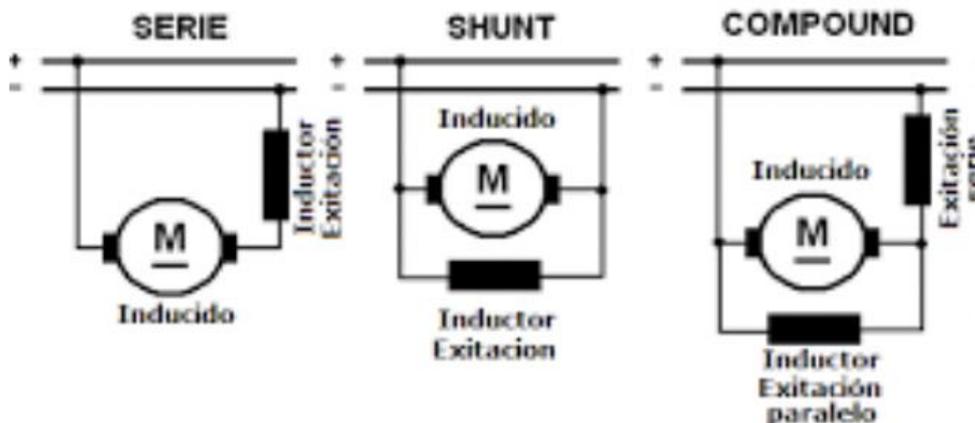


Figura 7-4 Tipos de motores corriente continua

8. TRANSMISIÓN

Esta parte del proyecto vamos a analizar la manera de transmitir el giro del motor al potenciómetro. Esta parte del proyecto es esencial, debido a que para poder llegar a analizar el comportamiento de nuestro sistema deberemos de comprobar continuamente la posición inicial de entrada respecto de la posición final de salida, para poder comprobar la diferencia entre ellas, es decir el error, y poder aplicar los diferentes tipos de reguladores, para hacer el sistema lo más eficiente posible.

Para determinar la posición de entrada y salida se utilizan unos potenciómetros giratorios continuos-vueltas, en el caso de la entrada seremos nosotros los que pilotaremos directamente dicho potenciómetro, pero en el caso de la salida, el potenciómetro deberá de indicar la posición en la que se encuentra el eje del motor, por lo que, a través de un juego de engranajes se conseguirá trasladar dicho giro, obteniendo en cada momento la medición de salida.

8.1. ¿QUÉ SON LOS ENGRANAJES?

Los engranajes son juegos de ruedas dentadas [22], los cuales encajan entre si, de manera que una rueda, conocida como rueda motriz, arrastra a otra rueda, conocida como rueda conducida, transmitiéndola movimiento, esta rueda conducida se mueve en sentido opuesto a la conductora.

El uso de los engranajes es variado, desde aumentar a disminuir la transmisión de velocidades o incluso mantener la transmisión de velocidades, una de las grandes peculiaridades es que podemos conseguir grandes reducciones de velocidades o por el contrario grandes aumentos, de una manera muy sencilla.

La principal condición para que las ruedas dentadas engranen entre si y se puede transmitir el movimiento, es que tengan los mismos parámetros y dimensiones de sus dientes, así conseguimos un buen acople para facilitar la buena transmisión de giro.

8.2. PARTES DE LOS ENGRANAJES

Los engranajes están compuestos por diferentes partes (*Figura 8-1*), los cuales van a definir el propio engranaje:

- Diente del engranaje: son los que se van a encajar entre si, transmitiendo el movimiento a través de aplicar fuerza de empuje, estos tiene un perfil característico además de que deben de ser iguales para que queden perfectamente engranados.
- Circunferencia exterior (R_e): es el radio de la circunferencia total del engranaje.
- Circunferencia interior (R_i); es el radio de la circunferencia del engranaje hasta la base del diente del engranaje.
- Circunferencia primitiva (R): es la circunferencia a lo largo de la cual engranan los dientes.
- Arco AB: espesor circular del diente.
- Arco BC: anchura del hueco entre los dientes.
- $AC = AB + BC$: paso circular.
- Addendum: distancia radial entre la circunferencia primitiva (R_i) y la exterior (R_e).

$$a = R_e - R_i$$

- Dedendum: distancia radial entre la circunferencia interior (R_i) y la primitiva (R).

$$d = R - R_i$$

- Altura del diente: distancia entre la circunferencia de fondo y exterior:

$$ht = a + d$$

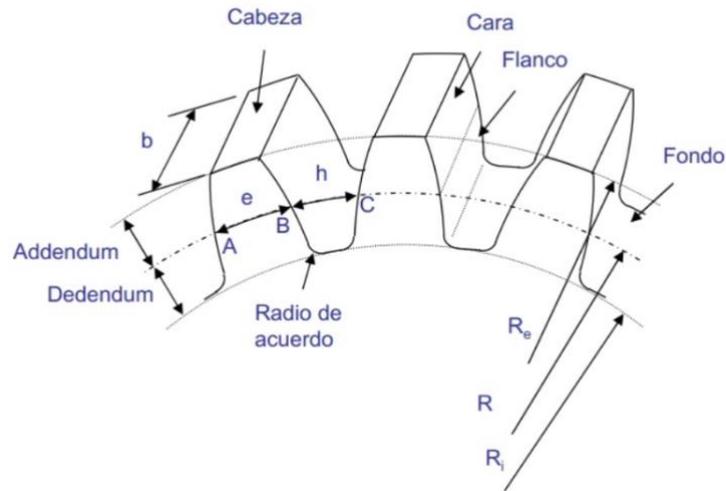


Figura 8-1 Partes del engranaje

8.3. TIPOS DE ENGRANAJES

Los engranajes se pueden clasificar en función de la forma de sus dientes y por otro lado en función de la posición de sus ruedas dentadas.

En función de la forma de sus dientes:

- Engranajes rectos (*Figura 8-2*): son los mecanismos más utilizados, consiste en la transmisión de dos ejes colocados de forma paralela sus ruedas dentadas están formadas por dientes completamente rectos. En nuestro caso utilizaremos este tipo de engranajes, debido a que tiene un uso muy extendido, son más sencillos de obtener y más baratos.

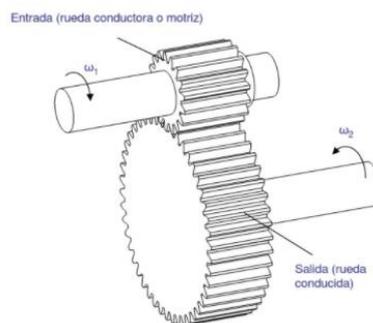


Figura 8-2 Engranajes rectos

- Engranajes helicoidales (*Figura 8-3*): a diferencia de los engranajes rectos, estos están dispuestos siguiendo una trayectoria de hélices paralelas alrededor de un cilindro. Por otro lado, son capaces de transmitir potencia a ejes paralelos o ejes que se encuentran cruzados.

La principal ventaja de este tipo de engranajes es que entran en contacto un mayor número de dientes que en el caso de los engranajes rectos, por lo que proporciona una transmisión de fuerza y de movimiento más uniforme y robusta, también es un sistema mucho más silencioso.

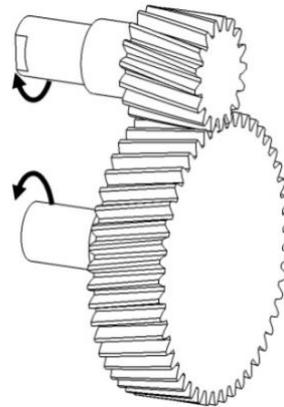


Figura 8-3 Engranajes helicoidales

- Engranajes cónicos (*Figura 8-4*): este tipo de engranajes se utilizan para transmitir movimiento entre ejes perpendiculares o en ejes con ángulos no rectos. Consiste en unas ruedas dentadas en forma de cono, estos dientes pueden ser rectos o helicoidales.

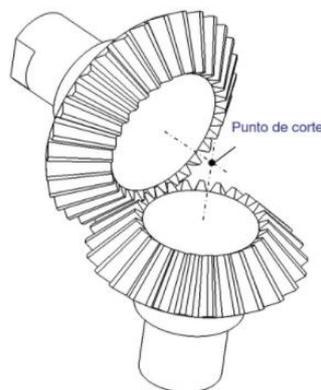


Figura 8-4 Engranajes cónicos

Por otro lado, lo podemos clasificar en función de la posición de las ruedas:

- Engranajes exteriores (8-5): cuando la transmisión de potencia se realiza a través de los dientes situados en la parte exterior de cada una de las ruedas dentadas.



Figura 8-5 Engranajes exteriores

- Engranajes interiores (8-6): a diferencia del anterior una de las ruedas dentadas utiliza su parte interior.



Figura 8-6 Engranajes interiores

8.4. RELACIÓN DE TRANSMISIÓN

La relación de transmisión entre dos engranajes es la relación que existe entre el número de dientes que componen a la rueda transmisora de movimiento y a la que se le transmite el movimiento, con la velocidad de giro de cada una de ellas (8.1).

$$\frac{W2}{W1} = \frac{Z1}{Z2}$$

(8.1)

Donde:

- W1: la velocidad angular de entrada.
- W2: la velocidad angular de salida.
- Z1: el número de dientes de la rueda dentada de entrada.
- Z2: el número de dientes de la rueda dentada de salida.

En nuestro sistema vamos a utilizar dos engranajes con el mismo número de dientes para que pueda transmitir la misma velocidad, se utilizarán engranajes de 30 dientes (*Figura 8-7*).

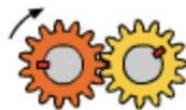


Figura 8-7 Engranajes de igual número de dientes

Procedemos a sustituir, en la ecuación 8.1, los valores conocidos, en este caso, sabemos que los engranajes tienen el mismo número de dientes. Por lo que, las velocidades serán iguales.

$$Z1 = Z2$$

$$\frac{W2}{W1} = 1$$

$$W2 = W1$$

Obtenemos que la relación de transmisión es igual a 1, por lo que la velocidad de entrada será exactamente igual a la velocidad de salida.

8.5. VENTAJAS E INCONVENIENTES

El uso de este tipo de engranajes supone numerosas ventajas respecto a otros sistemas de transmisión como pueden ser las correas y poleas, que pueden llegar a resbalar y no transmitir el 100% del movimiento.

Algunas ventajas de la utilización de engranajes son:

- Ocupan un espacio reducido.
- A diferencia de las poleas y correas, no tiene la posibilidad de deslizarse.
- Tiene una gran capacidad de transmisión.
- Tienen un bajo y sencillo mantenimiento.
- Gran capacidad reductora o amplificadora de velocidad.

Por otro lado, existen inconvenientes de este mecanismo de transmisión:

- Son más costosos que otros mecanismo de transmisión.
- Llevan un proceso de fabricación más difícil.
- En su transmisión se produce ruido, debido al rozamiento de sus dientes.

9. FUENTE DE ALIMENTACIÓN

Para alimentar el sistema se utilizara una fuente de alimentación PC, esta fuente será capaz de aportarnos las diferentes tensiones para nuestro circuito, con tensiones de: $\pm 12V$, 5V y 3.3V (Figura 9-1).

Contando con 5 salidas de 5V, una salida de -12V y 2 tensiones de 12 V, además de 8 tierras comunes.

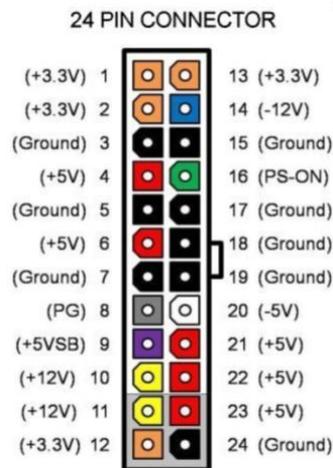


Figura 9-1 Pin Out fuente ATX

Para que la fuente de alimentación funcione y nos aporte las diferentes tensiones será necesario cortocircuitar, el pin número 16, conocido como PS-ON, representado con el color verde, con una de las tierras. Cuando la fuente está en standby ,, este pin tiene un voltaje comprendido entre los 3 y los 5 V, cuando lo cortocircuitamos, genera una señal de 0 V, produciendo el encendido de la fuente.

10. IMPLEMENTACIÓN DEL PROYECTO

Una vez ya conocemos todos los conceptos básicos de los procesos que vamos a implementar en nuestro proyecto para llevarlo a cabo, vamos a concretar aún más los diferentes partes que lo componen tanto a nivel software como a nivel hardware.

10.1. FUNCIONAMIENTO DEL SISTEMA

Dicho sistema consistirá en controlar la posición y la velocidad de una salida respecto de una entrada, con el sistema se podrán utilizar dos modos completamente diferentes para poder lograr el control de la salida. Dichos modos se podrán conmutar a través de un interruptor.

El primer modo consistirá en un modo planta, donde podremos simular directamente sobre el sistema de entrenamiento, se podrá variar la señal de entrada, y observar la respuesta del sistema en la salida, por otro lado, se podrán variar los valores de las ganancias del regulador, formando diferente tipos de reguladores y se podrá observar la respuesta del sistema, aumento o disminución de las oscilaciones, velocidad, error o incluso la inestabilidad del sistema.

En este primer método, en todo momento se podrán obtener datos a del valor de la entrada y de la salida a través de un par de conectores, estos transformarán la señal analógica en la cual trabaja nuestro Arduino (0-5 V) a una señal de ± 5 V, para obtener la lectura de señales con la tarjeta de adquisición de datos con el mayor número de bits posible,.

Por otro lado, existirá un segundo método de utilización del sistema de entrenamiento, este consistirá en la utilización de nuestro ordenador para ejercer como regulador. Para ello conectaremos la señal de la tarjeta de adquisición a nuestro puerto de entrada de señales analógicas, el cuál transformará la señal proveniente de la tarjeta de adquisición de datos, la cual tendrá valores entre ± 5 V, en una señal legible para Arduino comprendida entre 0-5 V. Por otro lado, conectaremos a la tarjeta de adquisición de datos al puerto de la respuesta del sistema, para poder obtener la señal de salida del sistema y poder calcular el error con nuestro ordenador, para poder aplicar así nuestro regulador.

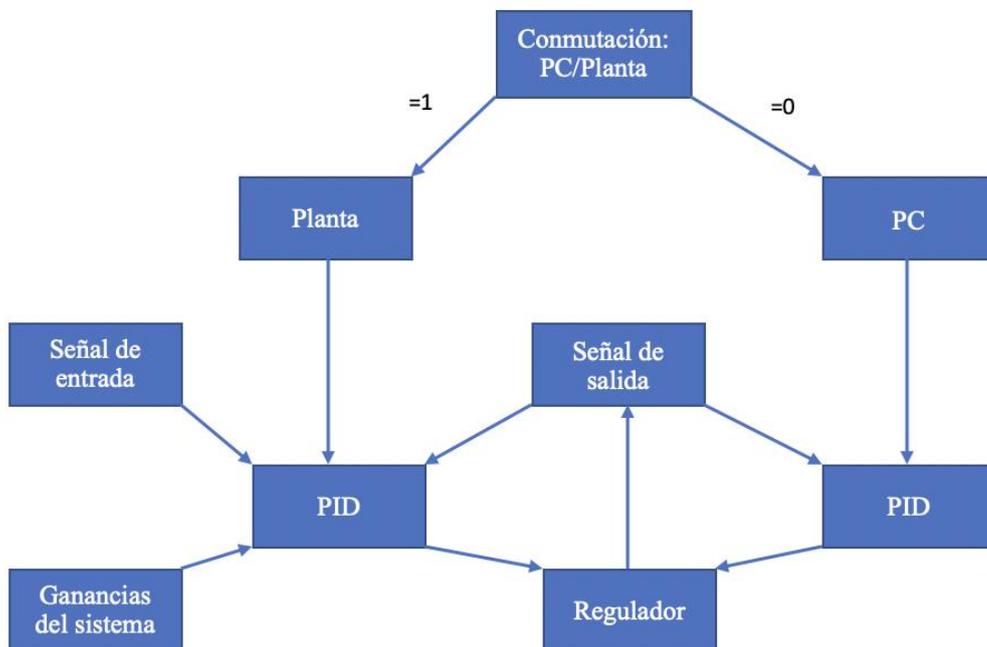
En este último método, quedará anulada la señal de entrada de la planta, pues vendrá directamente del PC, además de las señales provenientes de los potenciómetros que

representan las tres ganancias del regulador, pues dichos valores serán aportados directamente por el ordenador.

A continuación, vamos a analizar el código que va a tener programado nuestra placa Arduino, además analizaremos las conexiones del sistema, y el montaje del sistema en general.

10.2. FLUJOGRAMA

Para ayudar la comprensión del código vamos a observar el siguiente flujograma.



10.3. ALIMENTACIÓN ARDUINO

Como ya se ha mencionado anteriormente en las características de nuestra placa (5. *Microcontrolador*), nuestra placa puede ser alimentada a través del pin Vin, por una tensión entre los 6 – 20 V, aunque se recomienda alimentar la placa con una tensión que vaya desde los 7 – 12 V, como en nuestro caso, vamos a utilizar una fuente de alimentación, capaz de aportarnos diferentes tensiones, para alimentar la placa, utilizaremos la salida de tensión que

nos aporta de 12 V, conectándola directamente a la tensión V_{in} , por otro lado, conectaremos el pin GND a la tierra común de la fuente de alimentación (*Figura 10-1*).

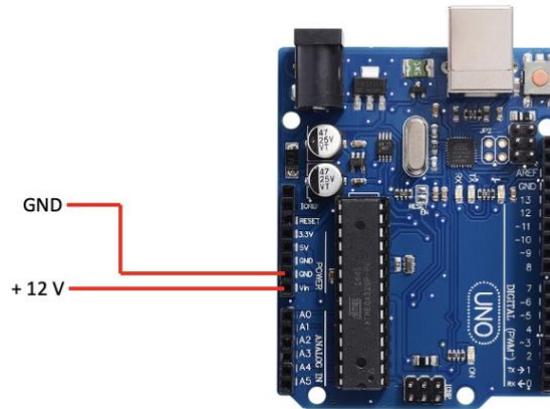


Figura 10-1 Alimentación Arduino

10.4. CONMUTACIÓN ENTRE LOS MODOS PC/PLANTA

En este apartado vamos a hablar sobre la conmutación entre ambos modos, para ello vamos a utilizar el método de la resistencia Pull-Up, explicado en el apartado: *5.3.1. Conmutación entre el modo PC o el modo planta.*

Para ello utilizaremos una resistencia de un valor de $4.7\text{ K}\Omega$ además de un interruptor para seleccionar los distintos modos, por otro lado, utilizaremos el puerto digital 2 de Arduino (*Figura 10-2*). Alimentando el sistema a través de los +5v que nos aporta la fuente de alimentación.

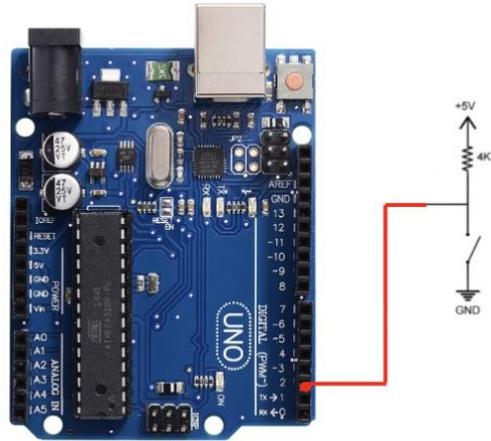


Figura 10-2 Conexión conmutador

A la hora de implementarlo en Arduino, el primer paso consistirá en realizar la lectura del puerto digital número 2, para ello utilizamos el comando `digitalRead()`, previamente deberemos de haber asignado el puerto a una variable, en este caso asignamos que el puerto número 2, es denominado "Conmutacion". Procedemos a realizar la lectura del valor del puerto, asignándolo a la variable "Modo":

```
Modo = digitalRead(Conmutacion)
```

Una vez hemos leído el valor del puerto digital procedemos a realizar una condición, siempre y cuando el valor del puerto sea alto (HIGH), es decir valga 1, vas a actuar en modo planta, sin embargo, si el valor del puerto digital es cero, es decir, un valor bajo (Low), entonces actuarás en modo PC.

Tabla 4 Estado del sistema en función del conmutador

Estado del puerto	Modo
HIGH	PLANTA
LOW	PC

10.5. LECTURAS ANALÓGICAS

Las entradas analógicas representan la mayoría de nuestras entradas, por un lado, tenemos las señales analógicas de entrada y de salida, por otro lado, los valores de las señales analógicas provenientes de las ganancias del regulador y finalmente la señal analógica proveniente del ordenador. El funcionamiento de las señales analógicas de Arduino se encuentra explicado en el apartado: *5.2. Entradas analógicas*.

En primer lugar debemos recordar que nuestros potenciómetros se encuentran alimentados con 0-5 V, por lo que el valor de salida estará comprendido entre esos valores, lo alimentamos a través de la fuente de alimentación, por otro lado, al señal proveniente del ordenador también estará entre los 0-5 V, tras sufrir una conversión.

La conexión de las señales analógicas es la siguiente:

- Pin A0: Se conectará la señal proveniente del potenciómetro de la entrada, realizando la lectura de la señal de entrada.
- Pin A1: Se conectará la señal proveniente del potenciómetro de la salida, realizando la lectura de la señal de entrada.
- Pin A2: se conectará la señal proveniente del potenciómetro de la ganancia proporcional (K_p), realizando la lectura de dicha ganancia.
- Pin A3: se conectará la señal proveniente del potenciómetro de la ganancia integrativa (K_i), realizando la lectura de dicha ganancia.
- Pin A4: se conectará la señal proveniente del potenciómetro de la ganancia derivativa (K_d), realizando la lectura de dicha ganancia.
- Pin A5: se conectará la señal proveniente del convertidor de tensiones, que convierte la señal de la tarjeta de adquisición de datos.

Para la lectura de dichas señales analógicas, se utilizará el comando "`analogRead()`", con el puerto que se desee realizar la lectura.

El conexionado de las diferentes entradas analógicas se puede observar en la *Figura 10-3*.

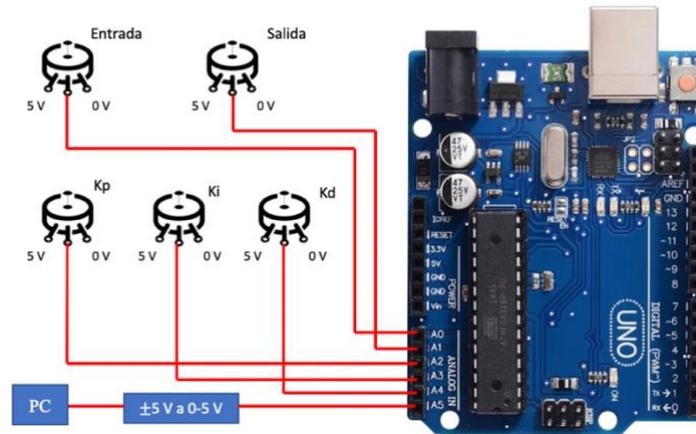


Figura 10-3 Conexión Señales analógicas

10.6. CONEXIONADO DEL DRIVER MOTOR

Finalmente en este proceso de montaje, procedemos a realizar la conexión del controlador de motores y del motor, el funcionamiento del controlador lo hemos explicado en el apartado: 6. *Puente en H*, por otro lado, el motor de corriente continua está explicado en el apartado: 7. *Motor corriente continua*. Por otro lado, haremos uso de la señal PWM, para poder controlar el driver del motor (5.3.3. *Señal PWM*), además de dos salidas digitales, para determinar el sentido de giro del motor (5.3.2. *Sentido de giro del motor*).

Para poder controlar el motor deberemos de realizar las siguientes conexiones (Figura 10-4):

- El puerto digital número 8 de Arduino lo conectaremos con el puerto IN1 del driver motor.
- El puerto digital número 7 de Arduino lo conectaremos con el puerto IN2 del driver motor.
- El puerto digital con salida PWM número 6 de Arduino lo conectaremos con el puerto ENA, para controlar la velocidad de giro.
- Los puertos de salida OUT1 y OUT2, del driver controlador los conectaremos cada uno a cada patilla del motor.

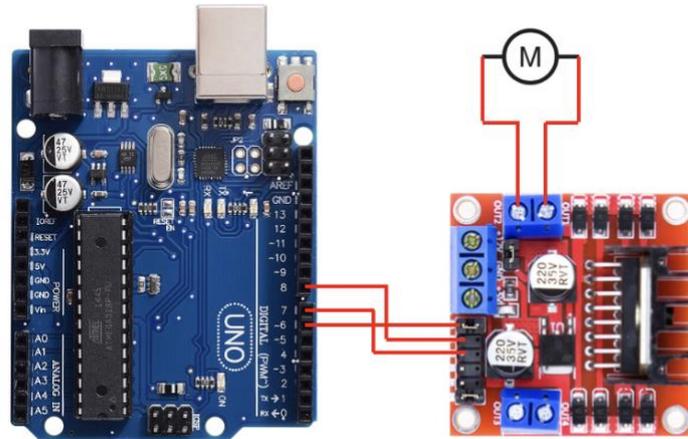


Figura 10-4 Conexión control del motor

La alimentación de nuestro driver motor como ya se explicó en apartados anteriores (6.3.2. *Alimentación del driver*) será la segunda opción utilizando una fuente de 12 V y otra de 5 V, ambas provenientes de nuestra fuente de alimentación (Figura 10-5).

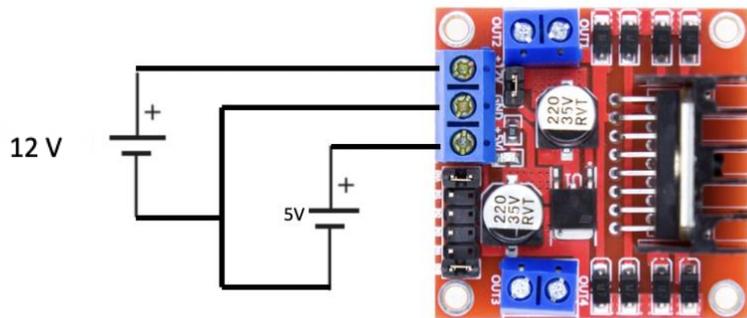


Figura 10-5 Conexión alimentación driver

10.7. CÓDIGO

Este código será el encargado de conmutar entre ambos modos, modo planta y el modo PC, por otro lado, deberá de actuar diferente en cada uno de los casos.

El primer de los pasos consistirá en asignar cada uno de las variables a cada uno de los puertos, como ya hemos podido observar en los apartados previos, la asignación de puertos completa será la siguiente:

- Pin A0 = Entrada .
- Pin A1 = Salida .

- Pin A2 = Kp .
- Pin A3 = Ki .
- Pin A4 = Kd .
- Pin A5 = EntradaOrdenador .
- Pin 2 = Conmutador .
- Pin 6 = Control.
- Pin 7 = Izquierda.
- Pin 8 = Derecha.

Una vez hemos asignado todos los puertos, procedemos a nombrar las diferentes variables:

- Variables del tiempo: TiempoTranscurrido, TiempoActual y TiempoAnterior.
- Variables del regulador: Error, ErrorAnterior, DerError, IntError, GKp, GKl, GKd, Respuesta regulador.
- Variables de entrada y salida: EntradaSistema y SalidaSistema.

Por otro lado, generamos el "void setup()", explicado en el apartado 5.1.2. *estructura de programación*, en el asignaremos el modo de los pines digitales, en este caso de salida, serán los pines que controlen nuestro motor.

pinMode(Control, OUTPUT)

pinMode(Izquierda, OUTPUT)

pinMode(Derecha, OUTPUT)

pinMode(Conmutador, INPUT)

10.7.1. Modo Planta

El primero de los modos que vamos a analizar es el modo planta, la primera condición que se debe de cumplir para que el modo planta se ejecute, es que el conmutador esté en modo alto, una vez se cumpla dicha condición se ejecutará el código.

Esta parte del código se va a realizar en el "void loop", el primer paso consistirá en realizar la lectura de las señales analógicas de entrada y salida. Dichas señales van a estar normalizadas

entre un valor 0 y de 1, para ello las dividimos entre 1023, el cuál será su valor más alto (10.1 y 10.2).

$$EntradaSistema = \frac{analogRead(Entrada)}{1023} \quad (10.1)$$

$$SalidaSistema = \frac{analogRead(Salida)}{1023} \quad (10.2)$$

A continuación, se procede a la lectura de las ganancias, esta parte, debido a que el sistema no se ha conseguido implementar al completo y realizar un análisis realista de él, vamos a suponer el rango de ganancias que utilizamos en los apartados anteriores como ejemplo, el cuál podemos encontrar en el apartado (3.2. *Diferentes tipos de controladores*). Por lo que, nuestras ganancias tendrán un valor máximo de 100. Por otro lado, el valor de las ganancias también se debe de normalizar entre un valor de 0 y 1 (10.3, 10.4 y 10.5).

$$GKp = \frac{analogRead(Kp)}{1023} \quad (10.3)$$

$$GKi = \frac{analogRead(Ki)}{1023} \quad (10.4)$$

$$GKd = \frac{analogRead(Kd)}{1023} \quad (10.5)$$

El siguiente paso consiste en determinar el tiempo actual en el cual está nuestro sistema, se medirá en milisegundos, con ello podremos obtener el tiempo transcurrido (10.6), a raíz de la resta del tiempo actual menos el tiempo pasado, inicialmente el tiempo pasado es cero, se

deberá de determinar previamente, una vez se ejecuta el código, el tiempo actual pasa a ser el tiempo pasado (10.7).

$$TiempoTranscurrido = TiempoActual - TiempoPasado \quad (10.6)$$

$$TiempoPasado = TiempoActual \quad (10.7)$$

El siguiente paso consistirá en obtener el valor del error (10.8), este se obtendrá de la diferencia entre la salida del sistema y de la entrada, como ambas lecturas están comprendidas entre 0 y 1, el error nos va a quedar con un valor normalizado de entre -1 y 1.

$$Error = SalidaSistema - EntradaSistema \quad (10.8)$$

El siguiente parámetro a calcular es el valor de la parte integral de nuestro regulador (10.9), esta parte es acumulativa, por lo que, en el código deberemos de poner el símbolo (+).

$$IntError += Error * TiempoTranscurrido \quad (10.9)$$

Como esta parte es acumulativa, puede darse el caso que la señal supere los valores normalizados, el valor debe de estar comprendida siempre entre ± 1 . Por lo que saturamos el

valor de la parte integral del regulador. Siempre y cuando el valor absoluto de la integral sea menor que 1 (10.10).

$$IntError = IntError + Error * TiempoTranscurrido \quad (10.10)$$

A continuación, la última parte que compone nuestro regulador, la parte derivativa (10.11). Se obtiene de la división del error actual menos el error anterior, todo ello dividido entre el tiempo transcurrido, dicha parte nuevamente estará normalizada entre un valor de ± 1 . El error anterior

inicialmente tendrá un valor cero, el cual deberemos determinar al inicializar las variables, a medida que se ejecuta el código el error anterior será el mismo que el error actual (10.12) y así sucesivamente.

$$DerError = \frac{Error - ErrorAnterior}{TiempoTranscurrido}$$

(10.11)

$$ErrorAnterior = Error$$

(10.12)

Finalmente, obtenemos el valor de nuestro regulador (10.13), el cual, se obtendrá de la multiplicación de cada una de las ganancias con la parte que le corresponde.

$$PID = GKp * Error + GKd * DerError + GK_i * IntError$$

(10.13)

Es posible que el valor de nuestro regulador (PID), no esté normalizado, por ello, lo normalizaremos, todo valor superior a 1, será igual a 1 y todo valor inferior a -1, será igual a -1. Con ello conseguimos que nuestra salida PID, se encuentre normalizada entre ± 1 .

Seguidamente determinamos el sentido de giro del motor, para valores de PID mayores que 0 el motor girara hacia la izquierda, en el caso contrario, en el caso que el valor de PID, sea menor que cero, el motor girará a la derecha.

Tabla 5 Sentido de giro en función del valor del PID

	Derecha	Izquierda
PID > 0	digitalWrite(derecha,LOW)	digitalWrite(Izquierda,HIGH)
PID < 0	digitalWrite(derecha,HIGH)	digitalWrite(Izquierda,LOW)

Finalmente desnormalizamos la respuesta del regulador (10.14), para ello la multiplicamos el valor absoluto de la variable PID por 255, que es el tamaño de nuestra señal PWM, la cuál se encuentra explicada en el apartado: 5.3.3. *señal PWM*. Obteniendo una señal de 0-255.

$$RespuestaRegulador = 255 * |PID| \tag{10.14}$$

Hemos comprobado dicha señal en el motor de corriente continua, y para valores PWM menores de 120, el motor no se mueve, debido a que no tiene la potencia suficiente para los engranajes, por lo que, creamos una condición, para valores de la respuesta del regulador mayores que cero y menores que 120, la respuesta será igual a 120 (10.15).

$$0 < RespuestaRegulador < 120 \rightarrow RespuestaRegulador = 120 \tag{10.15}$$

10.7.2. Modo PC

Este segundo modo, se ejecutará siempre y cuando el conmutador valga cero, es decir, esté en modo bajo, por otro lado, será necesario aportar una señal externa de entrada, la cual proviene del PC, esta señal pasará un proceso de transformación para trabajar en el rango de la placa Arduino, 0-5 V.

Dicha señal analógica de entrada se transformará en la señal de nuestro PID, de Arduino.

Para ello utilizamos la ecuación de la recta, para determinar el valor normalizado de la señal del regulador entre ± 1 , en función de la señal analógica de entrada, la cuál va a tener un valor comprendido entre 0 y 1023.

Conocemos que para el valor PID 1, nuestra señal analógica deberá valer 1023, y que para nuestra señal de regulador 0, la señal analógica valdrá 0, que corresponden a los 0 voltios de dicha señal, que previamente de la conversión valieron -5V.

La *Figura 10-6* representa la ecuación de la recta.

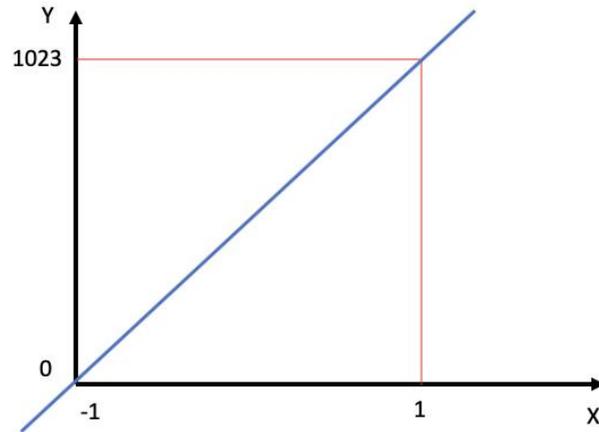


Figura 10-6 Representación de la recta

Calculamos la pendiente que define nuestra recta (10.16)

$$m = \frac{1023}{1 - (-1)} = \frac{1023}{2} = 511.5$$

(10.16)

A través de la ecuación de la recta y sustituyendo el valor de la pendiente (10.16), determinamos el valor de X (10.17), comprendido entre ± 1 , en función del valor de lectura de la entrada analógica.

$$Y - Y_0 = m * (X - X_0)$$

$$Y - 0 = 511.5 * (X - (-1))$$

$$X = \frac{Y}{511.5} - 1$$

(10.17)

Una vez ya hemos obtenido la ecuación que va a determinar el valor normalizado de la señal PID modo PC, sustituimos los valores, obteniendo la ecuación 10.18.

$$PID = \frac{analogRead(EntradaOrdenador)}{511.5} - 1$$

(10.18)

Por ejemplo, en el caso de que la señal proveniente de la tarjeta de adquisición de datos fueran 5V, entonces a Arduino le llegarían 5V, siendo el valor de la entrada analógica igual a 1023, sustituyendo en la ecuación 10.18, nos quedaría un valor PID de 1.

$$PID = \frac{1023}{511.5} - 1 = 2 - 1 = 1$$

En el caso de que la señal proveniente de la tarjeta de adquisición de datos se tratase de una señal de -5V, entonces al Arduino le llegarían 0V, debido a la transformación sufrida. Por lo que, nuestra lectura analógica sería 0, sustituyendo en la ecuación 10.18, obtendríamos una señal PID de -1.

$$PID = \frac{0}{511.5} - 1 = 0 - 1 = -1$$

En el caso de que nuestra tarjeta de adquisición de datos fuera cero, nos llegaría a Arduino una tensión de 2.5 V, debido a la transformación de tensión sufrida, siendo igual a 511.5, lo cual haría que nuestro PID, sustituyendo en la ecuación 10.18, nos da que el PID tiene un valor de cero, es decir, no actúa.

$$PID = \frac{511.5}{511.5} - 1 = 1 - 1 = 0$$

A continuación, seleccionamos el sentido de giro exactamente igual que del modo planta, para valores de PID mayores que 0 el motor girara hacia la izquierda, en el caso contrario, en el caso que el valor de PID, sea menor que cero, el motor girará a la derecha.

Finalmente desnormalizamos nuestra señal PID (10.19), para que tenga un valor comprendido entre los 0 y 255. Además de hacer que la señal PWM siempre que se accione, es decir, siempre que no valga cero esté comprendida entre los 120-255 (10.20).

$$RespuestaRegulador = 255 * |PID| \tag{10.19}$$

$$0 < RespuestaRegulador < 120 \rightarrow RespuestaRegulador = 120 \tag{10.20}$$

10.8. REPRESENTACIÓN DEL CONEXIONADO DEL SISTEMA

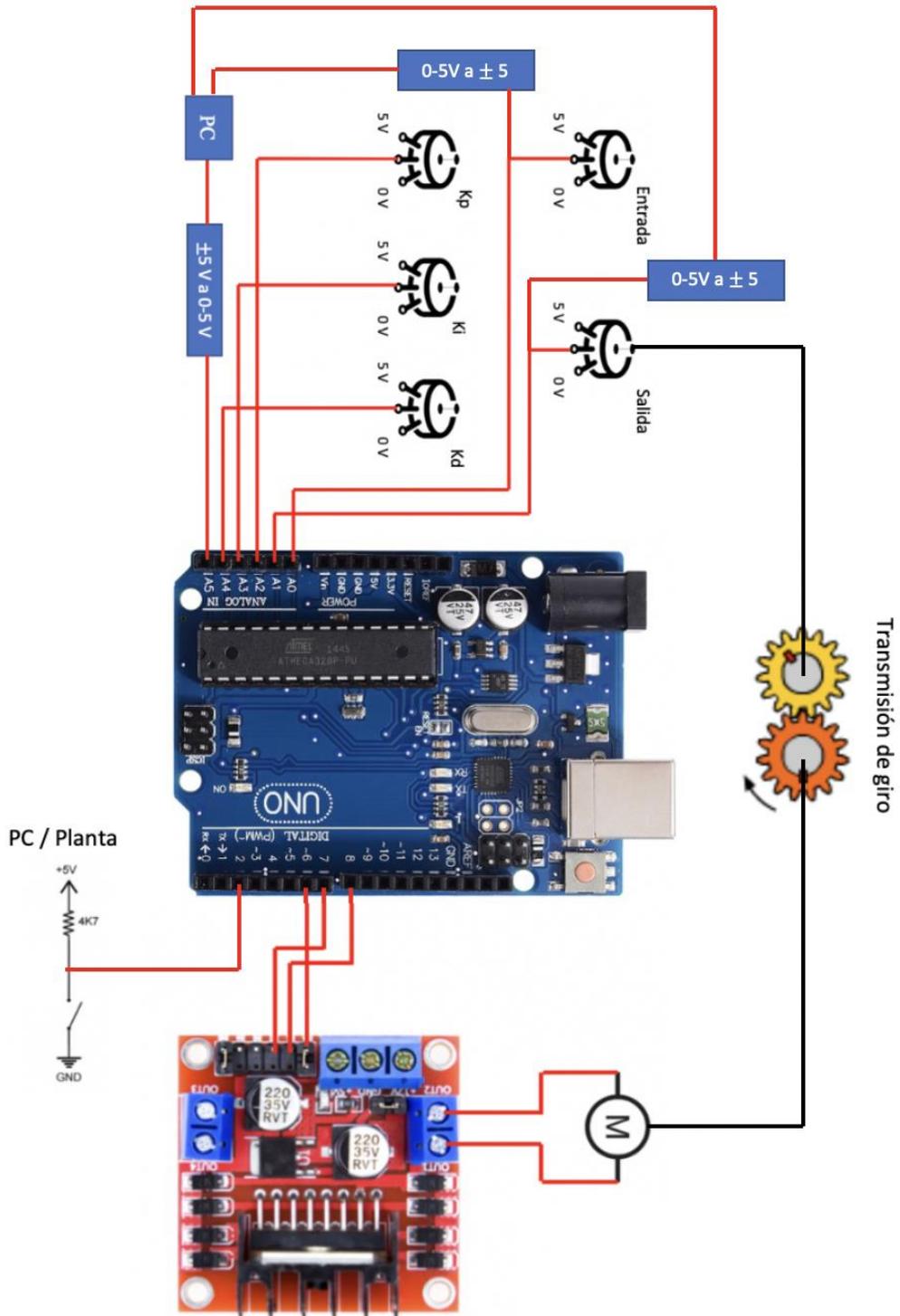


Figura 10-7 Conexionado completo

10.9. ESTRUCTURA DE LA PLANTA

En este apartado vamos a observar la estructura metálica que se va a utilizar a la hora de montar nuestra planta.

Se van a utilizar barras de aluminio de 2Cm x 2Cm, formando la estructura que podemos observar en la *Figura 10-8*.

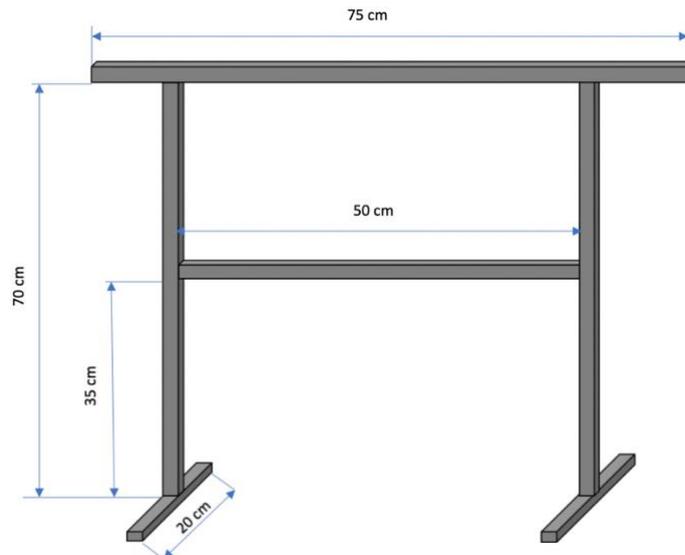


Figura 10-8 Estructura metálica

Por otro lado, constará de una tabla de madera de okumen con las siguientes medidas (*Figura 10-9*).

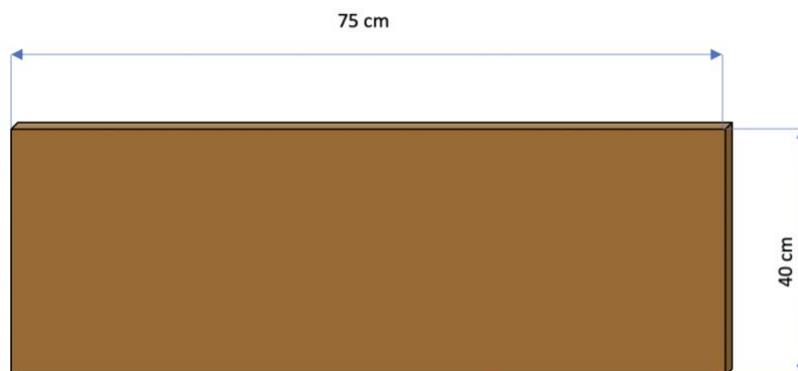


Figura 10-9 Estructura okumen

Finalmente en la *Figura 10-10*, podemos observar la representación final de la planta.

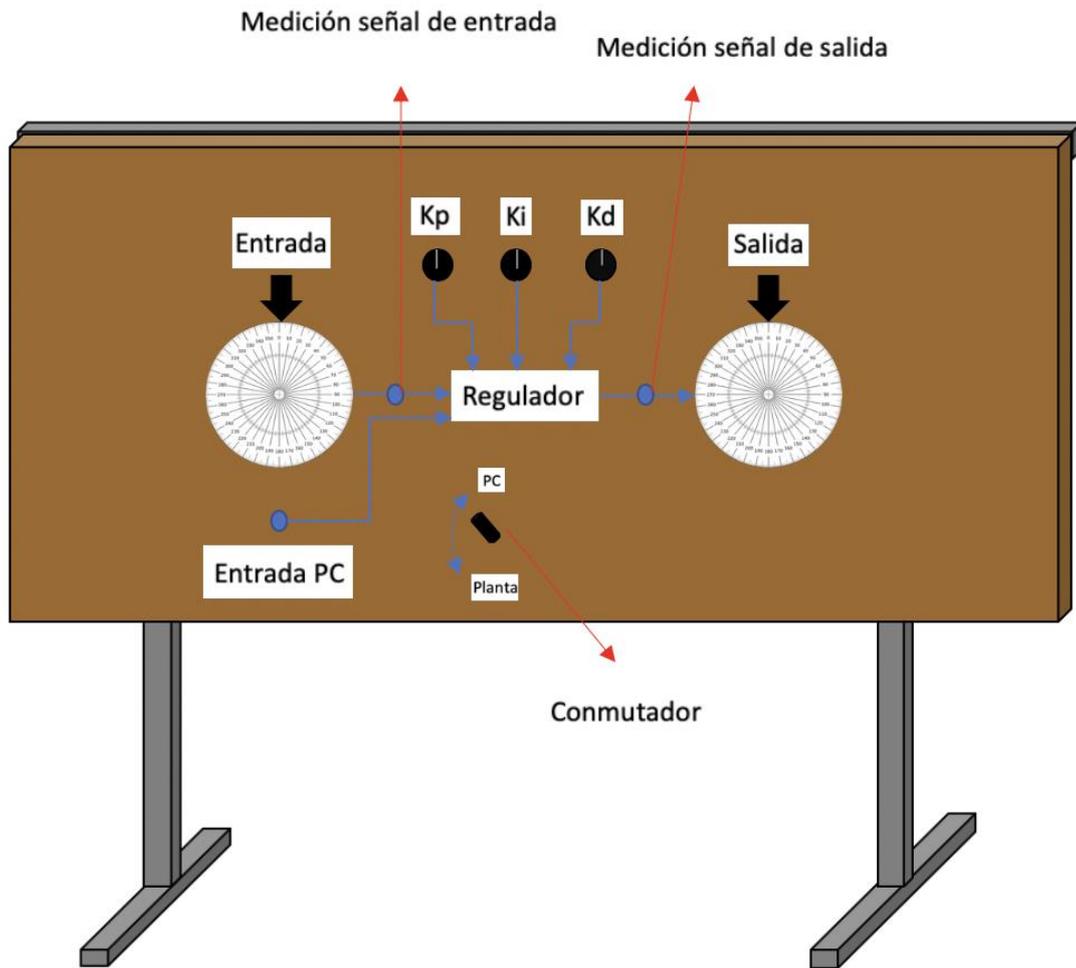


Figura 10-10 Estructura física final

11. ANEXOS

A continuación, veremos el código implementado de las diferentes partes de nuestro proyecto.

11.1. CÓDIGO ARDUINO

En este apartado podemos observar el código que va a ejecutar nuestra tarjeta Arduino, en el se pueden observar los dos modos de funcionamiento, así como la lectura de las diferentes entradas analógicas y entrada digital, también podemos observar el control de nuestro motor, de los dos modos posibles.

```
//Codigo creado por: Rubén Zabala de Cos

//Inicializo variables
const int Entrada = A0, Salida = A1;

const int Derecha = 8, Izquierda = 7, Control = 6;

const int Kp = A2, Ki = A3, Kd = A4;

const int EntradaOrdenador = A5;

const int Conmutador = 2;
int Modo = 0;

unsigned long TiempoActual;
double TiempoTranscurrido;

int TiempoPasado=0, ErrorAnterior=0;

int EntradaSistema, SalidaSistema, RespuestaRegulador;
double Pid, Error, DerError, IntError, GKp, GKd;

void setup() {
  Serial.begin(9600);
  pinMode(Control, OUTPUT);
  pinMode(Izquierda, OUTPUT);
  pinMode(Derecha, OUTPUT);
  pinMode(Conmutador, INPUT);
}

void loop(){
  Modo = digitalRead(Conmutador); //Leo el estado del conmutador
  if (Modo == HIGH) { //En caso de que sea 1, se ejecuta el modo planta

    //Leo y normalizo las señales provenientes de los potenciómetros
    EntradaSistema = analogRead(Entrada)/1023;
    SalidaSistema = analogRead(Salida)/1023;

    GKp = analogRead(Kp)/1023;
    GKd = analogRead(Kd)/1023;

    TiempoActual = millis(); //Mido el tiempo desde que se encendió la placa
    TiempoTranscurrido = (double)(TiempoActual - TiempoPasado); //calculo el tiempo transcurrido

    Error = SalidaSistema - EntradaSistema; //Calculo el error que solo va de -1 a 1
    IntError += Error*TiempoTranscurrido; //Calculo la parte integral del sistema
    DerError = (Error-ErrorAnterior) / TiempoTranscurrido; //Calculo la parte derivativa del sistema
```

```
    if (abs(IntError)<1){ //Me aseguro de saturar el valor de la integral
        IntError = IntError + Error * TiempoTranscurrido;
    }

    Pid = GKp*Error + GKd*DerError + GKi*IntError; //Calculo la acción del regulador

    if (Pid > 0) //Determino el sentido de giro del motor
    {
        digitalWrite(Izquierda, HIGH);
        digitalWrite(Derecha, LOW);
    }else if (Pid < 0){
        digitalWrite(Izquierda, LOW);
        digitalWrite(Derecha, HIGH);
    }
}

if (Pid > 1) //Me aseguro de que la acción del regulador este dentro de los valores
{
    Pid = 1;
}else if (Pid<-1){
    Pid = -1;
}

RespuestaRegulador = 255*abs(Pid); //Desnormalizo la acción del regulador

if (RespuestaRegulador<120 && RespuestaRegulador>0){ //Garantizo el movimiento del motor
    RespuestaRegulador = 120;
}

ErrorAnterior = Error; //Actualizo el error anterior
TiempoPasado = TiempoActual; // Actualizo el tiempo pasado

analogWrite(Control,RespuestaRegulador); //Proporciono la señal PWM al motor
}
else {
    Pid = (analogRead(EntradaOrdenador)/511.5)-1; //Calculo el valor PID normalizado en función de la entrada

    if (Pid > 0) //Determino el sentido de giro
    {
        digitalWrite(Izquierda, HIGH);
        digitalWrite(Derecha, LOW);
    }else if (Pid < 0){
        digitalWrite(Izquierda, LOW);
        digitalWrite(Derecha, HIGH);
    }
}

RespuestaRegulador = 255*abs(Pid); //Desnormalizo la señal del regulador

if (RespuestaRegulador<120 && RespuestaRegulador>0){ //Aseguro el movimiento del motor
    RespuestaRegulador = 120;
}

    analogWrite(Control,RespuestaRegulador); //Proporciono la señal PWM al motor
}
}
```

11.2. CÓDIGO ZIEGLER-NICHOLS Y APLICACIÓN DEL REGULADOR

Este código se utilizará para calcular las ganancias de nuestro sistema desde el ordenador utilizando el programa MATLAB. Deberemos de utilizar nuestra tarjeta de adquisición de datos, conectando la señal de salida con la entrada PC, que observamos en la planta, por otro lado, deberemos de conectar la señal de salida de nuestra planta, para realizar su lectura.

```
% Código creado por Rubén Zabala de Cos

clear; close all; clc;

daqreset;

%crear y configurar sesion

s=daq.createSession('ni');

%añadir y configurar canales

co=addAnalogOutputChannel(s,'Dev1',[0],'Voltage');

%canal cero de entrada cableado con el canal de salida

ci=addAnalogInputChannel(s,'Dev1',[0],'Voltage');

set(ci,'TerminalConfig','SingleEnded');

% creamos una entrada variada

tfin=15; Ts=0.01; %Definimos el tiempo total y el periodo de muestreo

t=(0:Ts:tfin)'; N=length(t); %Definimos el vector del tiempo

u=ones(size(t));y=zeros(size(t)); yf=y; % Para que la entrada sea un escalon, metemos una
matriz de unos

%inicializamos variables

e=y;

v=y;

d=y;

i=y;
```

```
p=y;

figure();

plot(t,u,'linewidth',2)%Representamos la entrada

xlabel('t'), ylabel('u(t)')

h = plot(t, [u y]);

drawnow;

outputSingleScan(s,0);

pause(2)

a=0.9;

outputSingleScan(s,1);

for k=2:N

    tic;

    y(k)=inputSingleScan(s); %Se lee la respuesta del sistema ante la entrada escalon

    yf(k)= a*yf(k-1)+(1-a)*y(k); %Filtramos la respuesta del sistema, para reducir el ruido

    set(h(1),'ydata', u);

    set(h(2),'ydata', yf);

    drawnow; %Se representa la entrada y la salida

    while toc < Ts end;

end;

% Procedemos a obtener la ecuacion de la recta

dy=diff(yf)/Ts; %obtenemos la derivada de la salida filtrada

dy=[dy; dy(end)];

figure, plot(t, dy)

%Buscamos el maximo valor de la derivada, y buscamos para que tiempo es
```

```
n=find(dy==max(dy(t>0.5)),1,'first');

tm=t(n) % Tiempo en el cual el valor es maximo

ymax=dy(n) % Punto maximo de la derivada. PENDIENTE

ym=yf(n) % Valor medio para el tm

yr=ym+ymax*(t-tm); %Ecuacion de la recta

%Representamos la respuesta con la recta y su derivada

figure()

plot(t,yr)

hold on

plot(t,yf)

plot(t,dy)

legend('recta','respuesta','derivada');

% calculamos los puntos de t0, t1 y t2, para los cuales corta la recta

n1=find(u>0,1,'first');

t0=t(n1)

n2=find(yr>=yf(1),1,'first');

t1=t(n2)

n3=find(yr>yf(end),1,'first');

t2=t(n3)

% Calculamos los valores de L y T

L=t1-t0

T=t2-t1

% A continuacion, aplicamos los diferentes reguladores y observamos su respuesta
```

```
% Obtenemos regulador P
```

```
tini=tic;
```

```
kp=T/L
```

```
ti=inf;
```

```
td=0;
```

```
ki=kp/ti
```

```
kd=kp*td
```

```
figure()
```

```
h = plot(t, [u yf e]);
```

```
legend('Entrada','Salida filtrada','error','Regulador P')
```

```
title('regulador P')
```

```
for k=2:N
```

```
    tic;
```

```
    y(k)=inputSingleScan(s);% Leemos la señal de la respuesta
```

```
    yf(k)= a*yf(k-1)+(1-a)*y(k); % Obtenemos la respuesta filtrada
```

```
    e(k)=u(k)-yf(k); %Calculamos el error
```

```
    p(k)=kp*e(k); %Obtenemos el regulador proporcional
```

```
    %Limitamos la salida entre 5 y -5 voltios
```

```
    if p(k)>5
```

```
        p(k)=5;
```

```
    end;
```

```
    if p(k)<-5
```

```
        p(k)=-5;
```

```
    end;
```

```
outputSingleScan(s,p(k)); % Lo aplicamos a la salida

set(h(1),'ydata', u); %Entrada

set(h(2),'ydata', yf); %Salida filtrada

set(h(3),'ydata', e); %Error
```

```
set(h(4),'ydata', p); %Regulador

drawnow;

while toc < Ts end;

end;

% Regulador PI

kp=0.9*(T/L);

ti=3.33*L;

td=0;

ki=kp/ti;

kd=kp*td;

figure()

h = plot(t, [u yf e]);

legend('Entrada','Salida filtrada','error','Regulador PI')

title('regulador PI')

for k=2:N

tic;

y(k)=inputSingleScan(s); %Leemos la señal de la respuesta

yf(k)= a*yf(k-1)+(1-a)*y(k); % Filtramos la señal de la respuesta del sistema

e(k)=u(k)-yf(k); % Calculamos el error

i(k)=i(k-1)+Ts*e(k); % la integral del error
```

```
p(k)=0.3*e(k)+ki*i(k); % Calcular el regulador

%Limitamos la salida entre 5 y -5 voltios

if p(k)>5

    p(k)=5;

end;

if p(k)<-5

    p(k)=-5;

end;

outputSingleScan(s,p(k)); % Aportamos el valor de la salida

set(h(1),'ydata', u); %entrada

set(h(2),'ydata', yf); %salida filtrada

set(h(3),'ydata', e); %error

set(h(4),'ydata', p); %regulador

drawnow;

while toc < Ts end;

end;

% Regulador PID

kp=1.2*(T/L);

ti=2*L;

td=0.5*L;

ki=kp/ti;

kd=kp*td;

figure()

h = plot(t, [u yf e]);
```

```
legend('Entrada','Salida filtrada','error')  
  
title('regulador PID')  
  
for k=2:N  
  
    tic;  
  
    y(k)=inputSingleScan(s); %Leemos la señal de respuesta del sistema  
  
    yf(k)= a*yf(k-1)+(1-a)*y(k); % La filtramos
```

```
e(k)=u(k)-yf(k); % Calculamos el error  
  
d(k)=(e(k)-e(k-1))/Ts; %Calculamos la derivada del error  
  
i(k)=i(k-1)+Ts*e(k); % Calculamos la integral del error  
  
p(k)=0.3*e(k)+kd*d(k)+ki*i(k); %Obtenemos el valor del regulador  
  
%Llimitamos la salida del sistema entre 5 y -5 Voltios  
  
if p(k)>5  
  
    p(k)=5;  
  
end;  
  
if p(k)<-5  
  
    p(k)=-5;  
  
end;  
  
outputSingleScan(s,p(k)); %Aplicamos la salida  
  
set(h(1),'ydata', u); %Entrada  
  
set(h(2),'ydata', yf); %Salida filtrada  
  
set(h(3),'ydata', e); %Error  
  
set(h(4),'ydata', p); %Regulador  
  
drawnow;  
  
while toc < Ts end;
```

end;

11.3. HOJAS DE CARACTERÍSTICAS DE LOS COMPONENTES

- Hoja características amplificador operacional UA741:
<http://www.ti.com/lit/ds/slos094g/slos094g.pdf?ts=1589534545471>

- Hoja de características de la tarjeta de adquisición de datos:
<https://www.ni.com/pdf/manuals/374369a.pdf>

- Hoja características potenciómetros continuo vueltas:
<https://docs.rs-online.com/c0c7/0900766b80030319.pdf>

- Hoja características potenciómetro 4.7 K Ω :
<https://es.rs-online.com/web/p/potenciometros/0387739/>

- Hoja características Arduino Uno Rev3:
<http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>

<https://components101.com/microcontrollers/arduino-uno>

- Manual de programación Arduino:
<https://arduinobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>

12. PRESUPUESTO

En este apartado se especificarán los costes relacionados con el desarrollo del proyecto, así como los costes relacionados con la futura implementación del proyecto.

Los costes se verán desglosados en dos grupos, por un lado analizaremos los costes derivados de la mano de obra, por otro lado, los costes derivados de la obtención del material necesario, par poder llevar a cabo el montaje, finalmente se ha añadido un beneficio industrial del 6%, y unos gastos generales del 13%.

En la tabla presupuesto material podemos observar el presupuesto a invertir para adquirir los componentes y materiales necesarios para poder llevar a cabo el proyecto, en el mostramos la cantidad, la descripción del producto, la referencia del fabricante, su precio con IVA incluido y donde se han adquirido los diferentes productos. Finalmente se muestra la cuantía total de los costes derivados únicamente de la adquisición material.

Presupuesto Material

Cantidad	Descripción del producto	Referencia del fabricante	Precio (Con IVA)	Lugar de compra
3	Potenciómetro Giratorio	1624139-2	5,51 €	RSComponents
2	Potenciómetro continuo vueltas	6639S-1-202	18,68 €	RSComponents
1	Arduino Uno Rev 3	A000066	11,00 €	Cadesa Santander
5	Amplificador Operacional UA741	UA741CP	0,53 €	RSComponents
1	Fuente alimentación 460W ATX	-	14,90 €	Cadesa Santander
1	Cableado de cobre	-	8,00 €	Cadesa Santander
1	Motor corriente continua 200RPM 12V	-	14,30 €	Amazon
40	Resistencias variadas	-	0,04 €	Cadesa Santander
3	Hembrilla para realizar conexiones	-	0,20 €	Cadesa Santander
1	Conmutador	-	0,80 €	Cadesa Santander
1	Hilo de estaño de 4mm	10528791	7,99 €	Leroy Merlín
1	Barra de cola termofusible BOSCH	18020611	3,59 €	Leroy Merlín
1	Soldador de estaño	13545434	9,99 €	Leroy Merlín
1	Pistola termofusible SALKI 150 W	82205066	14,99 €	Leroy Merlín
1	Driver controlador motores	-	9,60 €	Amazon
1	Tablero contrachapado de okume 60x120 Cm	11034254	7,79 €	Leroy Merlín
1	Barras de aluminio 2X2 Cm	-	30,00 €	Leroy Merlín
1	Remaches aluminio y acero	13244273	4,29 €	Leroy Merlín
1	Remachadora COMERSIM	13867581	9,49 €	Leroy Merlín
2	Engranaje rueda sin fin de bronce 30 dientes	521-6957	32,96 €	RSComponents
	TOTAL		271,39 €	

PRESUPUESTO

A continuación, procedemos a obtener un presupuesto sobre el coste de la mano de obra necesaria para poder realizar el proyecto. Para realizar este coste por hora de los operarios nos hemos basado en el BOC, publicado el 21 de marzo de 2017 y que está vigente para el periodo 2017-2021, para la empresa Sidenor Aceros Especiales. Se puede acceder al BOC en el siguiente enlace: <https://boc.cantabria.es/boces/verAnuncioAction.do?idAnuBlob=311031>.

Se ha considerado que los procesos de investigación y diseño del sistema, diseño de los circuitos, programación de Arduino y la creación de la documentación han sido desarrollados por un técnico especialista, por otro lado, los procesos de montaje han sido desarrollados por un operador. Para obtener el coste por hora, se ha realizado una media entre los diferentes salarios diarios de cada nivel y se han dividido entre 8 horas que se ha supuesto que tiene la jornada laboral. No se han tenido en cuenta horarios nocturnos, trabajo en días festivos y demás situaciones que pudieran aumentar dicho precio por hora.

Presupuesto mano de obra

Descripción del proceso	Horas de trabajo	Precio por hora	Precio Total
Investigación y diseño del sistema	100	8,96 €	896,00 €
Diseño circuitos	10	8,96 €	89,60 €
Montaje circuitos	10	7,3 €	73,00 €
Montaje estructura	10	7,3 €	73,00 €
Conexión del sistema	10	7,3 €	73,00 €
Programación de Arduino	10	8,96 €	89,00 €
Creación de la documentación	70	8,96 €	627,20 €
TOTAL	220 horas		1920,20 €

El siguiente paso consiste en calcular el coste del TFG, sumando los dos presupuestos.

PRESUPUESTO TOTAL

Presupuesto	Cantidad
Presupuesto mano de obra	1920,20 €
Presupuesto Material	271,39 €
TOTAL	2191,59 €

PRESUPUESTO

Finalmente añadimos al presupuesto total, los gastos derivados al beneficio industrial (6% del presupuesto total) y también añadimos los gastos generales (13% del presupuesto total).

PRESUPUESTO FINAL

Presupuesto	Cantidad
Presupuesto total	2191,59 €
Beneficio industrial (6% del presupuesto total)	131,50 €
Gastos generales (13% del presupuesto total)	284,91 €
TOTAL	2608,00 €

13. BIBLIOGRAFÍA

- [1] Área tecnología: Potenciómetro. Disponible en:
<https://www.areatecnologia.com/electronica/potenciometro.html>
- [2] Fernando Morilla García Dpto. de Informática y Automática ETSI de Informática, UNED: el controlador PID. Disponible en:
<http://www.dia.uned.es/~fmorilla/MaterialDidactico/El%20controlador%20PID.pdf>
- [3] Picuino: Controlador PID. Disponible en:
<https://www.picuino.com/es/arduprog/control-pid.html>
- [4] Luis Llamas: Ajuste manual PID. Disponible en:
<https://www.luisllamas.es/como-ajustar-un-controlador-pid-en-arduino/>
- [5] Picuino: Método Ziegler-Nichols. Disponible en:
<https://www.picuino.com/es/arduprog/control-ziegler-nichols.html>
- [6] Control automático educación: Método Ziegler-Nichols. Disponible en:
<https://controlautomaticoeducacion.com/control-realimentado/ziegler-nichols-sintonia-de-control-pid/>
- [7] Xataka: ¿Qué es Arduino?. Disponible en:
<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- [8] Arduino: Características principales de la placa. Disponible en:
<https://store.arduino.cc/arduino-uno-rev3c>
- [9] Aprendiendo Arduino: Entorno de programación de Arduino IDE. Disponible en:
<https://aprendiendoarduino.wordpress.com/2016/03/29/entorno-de-programacion-de-arduino-ide/>
- [10] Aprendiendo Arduino: Estructura sketch Arduino. Disponible en:
<https://aprendiendoarduino.wordpress.com/2017/06/19/estructura-sketch-arduino/>

- [11] Creatividad codificada: Entradas y salidas analógicas (PWM) en Arduino. Disponible en:
<https://creatividadcodificada.com/arduino/entradas-y-salidas-analogicas-en-arduino/>
- [12] Luis Llamas: Entradas analógicas Arduino. Disponible en:
<https://www.luisllamas.es/entradas-analogicas-en-arduino/>
- [13] Luis Llamas: Entradas digitales Arduino. Disponible en:
<https://www.luisllamas.es/entradas-digitales-en-arduino/>
- [14] Programa fácil: Resistencia Pull Up y Pull Down. Disponible en:
<https://programarfácil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/>
- [15] Luis Llamas: Salidas analógicas PWM en Arduino. Disponible en:
<https://www.luisllamas.es/salidas-analogicas-pwm-en-arduino/>
- [16] National Instruments: ¿Qué es una señal de modulación de ancho de pulso (PWM) y para qué se utiliza?. Disponible en:
<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019OkFSAU&l=es-ES>
- [17] Plataforma arte y medios: Controlar motores corriente continua con puente en H. Disponible en:
<http://arteymedios.org/tutoriales/item/76-controlar-motores-de-corriente-continua-con-puente-h>
- [18] Tercesa: Motor de corriente continua: tipos y partes. Disponible en:
<https://tercesa.com/noticias/motor-de-corriente-continua-tipos-y-partes/>
- [19] Máquinas eléctricas: Partes de un motor de corriente continua. Disponible en:
<https://maquinaselectricastecnind.wordpress.com/2015/04/27/partes-de-un-motor-de-corriente-continua/>
- [20] Así funciona el motor de corriente continua o directo. Disponible en:
http://www.asifunciona.com/electrotecnia/af_motor_cd/af_motor_cd_6.htm

[21] IES Villalba Hervas Tecnología: Tipos motores corriente continua. Disponible en:
<https://iesvillalbahervastecnologia.files.wordpress.com/2010/02/motores-electricos-parte-ii1.pdf>

[22] Edu Xunta: Engranajes. Disponible en:
https://www.edu.xunta.gal/espazoAbalar/sites/espazoAbalar/files/datos/1464947673/contido/43_engranajes.html