

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



## **DISEÑO CONCEPTUAL DE UN SISTEMA DE CONTROL PARA UN MOTOR DC DE IMANES PERMANENTES ADHERENTE A CONCEPTOS DIGITALES**

**CARLOS ANDRES SALGADO MORENO**

**UNIVERSIDAD DE PAMPLONA  
FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
PROGRAMA INGENIERIA MECATRONICA  
PAMPLONA  
2022**

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



## **DISEÑO CONCEPTUAL DE UN SISTEMA DE CONTROL PARA UN MOTOR DC DE IMANES PERMANENTES ADHERENTE A CONCEPTOS DIGITALES**

**CARLOS ANDRES SALGADO MORENO**

**ARTICULO MONOGRAFIA**

**PhD. CRISTHIAN IVAN RIAÑO JAIMES (Tutor)  
DOCENTE UNIVERSIDAD DE PAMPLONA**

**UNIVERSIDAD DE PAMPLONA  
FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
PROGRAMA INGENIERIA MECATRONICA  
PAMPLONA  
2022**

# Formando **líderes** para la construcción de un nuevo país en paz



## CONTENIDO

|                                                                                                      |    |
|------------------------------------------------------------------------------------------------------|----|
| Lista De Ilustraciones .....                                                                         | 6  |
| Lista De Tablas .....                                                                                | 8  |
| Resumen.....                                                                                         | 9  |
| Introducción .....                                                                                   | 10 |
| Capítulo I .....                                                                                     | 11 |
| Generalidades.....                                                                                   | 11 |
| Planteamiento del problema y Justificación. ....                                                     | 11 |
| Problema científico. ....                                                                            | 11 |
| Objetivo General.....                                                                                | 12 |
| Objetivos Específicos.....                                                                           | 12 |
| Estado del arte o antecedentes. ....                                                                 | 13 |
| Implementación de dos estrategias de control para la velocidad de un motor DC. ....                  | 13 |
| Diseño de un controlador PID análogo para la velocidad de un motor DC mediante Matlab y pspice. .... | 13 |
| Control PID de la velocidad de una banda transportadora para la clasificación de objetos. ....       | 14 |
| Control de velocidad y dirección de un robot de carrera autónomo. ....                               | 14 |
| Capítulo II .....                                                                                    | 15 |
| Desarrollo Objetivo 1.....                                                                           | 15 |
| 2.1. Técnicas de Control PWM.....                                                                    | 15 |
| 2.1.1 Conceptualización de la señal PWM Arduino .....                                                | 15 |
| 2.1.2 Señal PWM con integrado 555 .....                                                              | 18 |
| 2.2 Identificación de control PWM.....                                                               | 19 |
| 2.3 Representación de un motor DC Modelado.....                                                      | 19 |
| 2.3.1 Modelado de una planta para un Motor DC.....                                                   | 20 |
| 2.3.2 Obtención del sistema a partir de ecuaciones. ....                                             | 23 |
| 2.3.3 Espacio de Estados.....                                                                        | 25 |

# Formando **líderes** para la construcción de un nuevo **país en paz**



|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| 2.3.4 Parametrización del modelo matemático del motor DC.....                      | 26 |
| 2.3.5 Medición de parámetros para la obtención del modelo. ....                    | 27 |
| 2.3.6 Simulación a través de los parámetros obtenidos del modelo del sistema. .... | 32 |
| 3. Controlador PID.....                                                            | 34 |
| 3.1 Implementación sistemas de control.....                                        | 34 |
| 3.2 Diseño del controlador por medio del Método Ziegler – Nichols. ....            | 35 |
| 3.3 Primer método Ziegler – Nichols.....                                           | 36 |
| 3.4 Método para el control de velocidad/ posición. ....                            | 40 |
| 4. Tipos de sintonizaciones que permita la flexibilidad del sistema. ....          | 40 |
| 4.1 Acción proporcional: .....                                                     | 41 |
| 4.2 Acción derivativa:.....                                                        | 42 |
| 4.3 Acción integral .....                                                          | 42 |
| 4.4 Aplicación de un controlador PI. ....                                          | 42 |
| 4.5 Aplicación de un controlador PD. ....                                          | 43 |
| 4.6 Aplicación de un controlador PID. ....                                         | 44 |
| 5. Sistema de control por ecuaciones diferenciales lazo cerrado.....               | 44 |
| 6. Componentes para el diseño de un controlador digital.....                       | 47 |
| 6.1 Identificación del Motor DC.....                                               | 47 |
| 6.2 Modulo Sensor Tipo Herradura. ....                                             | 48 |
| 6.3 Arduino UNO.....                                                               | 49 |
| 6.4 Puente H.....                                                                  | 50 |
| 6.5 Potenciómetro .....                                                            | 51 |
| 6.6 Pantalla LCD con adaptador I2C. ....                                           | 51 |
| 6.7 Conexión de la placa Arduino USB.....                                          | 52 |
| 6.8 Cableado .....                                                                 | 53 |
| 6.9 74hc04.....                                                                    | 53 |
| 6.10 Adaptador de corriente.....                                                   | 54 |
| 6.11 Conectores tipo Jack banana.....                                              | 55 |

# Formando **líderes** para la construcción de un nuevo país en paz



|                                                                   |    |
|-------------------------------------------------------------------|----|
| 6.12 Placa de prototipos.....                                     | 55 |
| 7. Programación para el diseño del controlador del motor Dc. .... | 56 |
| 7.2 Driver .....                                                  | 56 |
| 7.3 Pantalla LCD.....                                             | 56 |
| 7.4 Sensor encoder. ....                                          | 57 |
| 7.5 Potenciómetro .....                                           | 57 |
| 7.6 Funciones al controlador de la posición y de velocidad. ....  | 57 |
| 7.7 Implementación del controlador PID.....                       | 59 |
| 7.8 Resultados de la implementación en programa Arduino .....     | 59 |
| 8. Construcción del controlador digital del motor CD.....         | 63 |
| Conclusiones .....                                                | 65 |
| Bibliografía .....                                                | 66 |
| ANEXOS .....                                                      | 67 |



## Lista De Ilustraciones

|                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------|----|
| Ilustración 1: Señal de salida PWM Fuente: [7] .....                                                                           | 15 |
| Ilustración 2 Representación del circuito con imán permanente. Fuente Srdanovic, V., Henao, O. B., & Ruiz, S. H. (2017). ..... | 21 |
| Ilustración 3 Modelo del diagrama de bloques representado en el programa Simulink. Fuente: Carlos salgado. ....                | 24 |
| Ilustración 4 Subsistema del motor DC en el programa Simulink en el software Simulink. Fuente: Carlos salgado. ....            | 25 |
| Ilustración 5 Motor 12V DC. ....                                                                                               | 28 |
| Ilustración 6 Obtención de datos por medio de Matlab Fuente: Carlos s. ....                                                    | 33 |
| Ilustración 7 Respuesta mediante el software Matlab.....                                                                       | 34 |
| Ilustración 8 Sistema de control con controlador PID. [11] .....                                                               | 35 |
| Ilustración 9 Sistema de escalón unitario.....                                                                                 | 36 |
| Ilustración 10 Recta tangente al punto de inflexión.....                                                                       | 36 |
| Ilustración 11 Forma Ziegler y Nichols [11] .....                                                                              | 37 |
| Ilustración 12 Implementación de un controlador PID.....                                                                       | 38 |
| Ilustración 13 Parámetros para el PID Fuente: Elaboración propia.....                                                          | 39 |
| Ilustración 14 Respuesta del controlador PID. Fuente: Elaboración propia .....                                                 | 40 |
| Ilustración 15 Respuesta de un sistema de control proporcional Fuente: Sergio Andrés Castaño Giraldo.....                      | 41 |
| Ilustración 16 Respuesta de un sistema de control proporcional integral. Fuente: Sergio Andrés Castaño Giraldo .....           | 43 |
| Ilustración 17 Respuesta de un sistema de control proporcional integral derivativa Fuente: Sergio Andrés Castaño Giraldo ..... | 44 |
| Ilustración 18 La ecuación del control PID. ....                                                                               | 45 |
| Ilustración 19 Retenedor orden cero .....                                                                                      | 45 |
| Ilustración 20 Motor Dc.....                                                                                                   | 48 |
| Ilustración 21 Módulo Sensor Tipo Herradura. [13] .....                                                                        | 48 |
| Ilustración 22 Placa Arduino UNO. [14] .....                                                                                   | 49 |
| Ilustración 23 Puente H. [15].....                                                                                             | 50 |
| Ilustración 24 Potenciómetro. Fuente SSDIELECT ELECTRONICA SAS .....                                                           | 51 |
| Ilustración 25 LCD con adaptador I2C. [11] .....                                                                               | 51 |
| Ilustración 26 Cable Arduino USB. [11] .....                                                                                   | 52 |
| Ilustración 27 Cables. [2].....                                                                                                | 53 |
| Ilustración 28 Integrado 74hc04 Fuente: Vistronica.....                                                                        | 53 |
| Ilustración 29 Adaptador de corriente [11].....                                                                                | 54 |

# Formando **líderes** para la construcción de un nuevo **país en paz**



|                                                                                                               |    |
|---------------------------------------------------------------------------------------------------------------|----|
| Ilustración 30 Conectores tipo Jack banana. Fuente Ferretronica. ....                                         | 55 |
| Ilustración 31 Placa perforada. [11].....                                                                     | 55 |
| Ilustración 32 Relación de pulsos encoder.....                                                                | 58 |
| Ilustración 33 Controlador PID. Fuente: Elaboración propia. ....                                              | 60 |
| Ilustración 34 Respuesta a diferentes cambios de posición Fuente elaboración propia. ....                     | 61 |
| Ilustración 35 Respuesta a cambio de la constante de Kp. Fuente elaboración propia. ....                      | 62 |
| Ilustración 36 Respuesta de la velocidad con el controlador PID implementado. Fuente: Elaboración propia..... | 63 |
| Ilustración 37 Diseño del concepto del controlador didáctico digital. Fuente elaboración propia. ....         | 64 |
| Ilustración 38 Visualización interna de los componentes del controlador. Fuente elaboración propia. ....      | 64 |
| Ilustración 39 Diseño cableado del controlador en Proteus.....                                                | 67 |
| Ilustración 40 Programación Arduino del control de posición. ....                                             | 73 |
| Ilustración 41 Visualización de Inicio del sistema de Control. ....                                           | 74 |
| Ilustración 42 Visualización de los parámetros del controlador PID. ....                                      | 74 |
| Ilustración 43 Visualización del control de posición con pantalla Oled.....                                   | 75 |
| Ilustración 44 Programación Arduino del control de velocidad. ....                                            | 78 |
| Ilustración 45 Demostración Lcd del control de velocidad. ....                                                | 79 |
| Ilustración 46 Visualización en el monitor serial del control de velocidad.....                               | 79 |
| Ilustración 47 Diseño de construcción del sistema de control en milímetros.....                               | 80 |

# Formando **líderes** para la construcción de un nuevo **país en paz**



## Lista De Tablas

|                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------|----|
| Tabla 1 <i>En la tabla siguiente podemos observar la relación de pines de cada Arduino.</i> .....     | 16 |
| Tabla 2 <i>Relación porcentaje del ciclo de trabajo.</i> .....                                        | 17 |
| Tabla 3 <i>Parámetros del motor DC con imán permanente.</i> .....                                     | 26 |
| Tabla 4 <i>Parámetros obtenidos.</i> .....                                                            | 29 |
| Tabla 5 <i>Parámetros obtenidos para la fuerza electromotriz.</i> .....                               | 30 |
| Tabla 6 <i>Resultados sobre las pruebas de inercia.</i> .....                                         | 31 |
| Tabla 7 <i>Parámetros obtenidos del motor.</i> .....                                                  | 32 |
| Tabla 8 <i>Fórmulas para los diferentes tipos de controlador por el método Ziegler -Nichols</i> ..... | 37 |
| Tabla 9 <i>Datos obtenidos para los controladores P, PI y PID</i> .....                               | 37 |

## Formando **líderes** para la construcción de un nuevo **país en paz**



### Resumen

En este proyecto de diseño de investigación, se conceptualizó un sistema de control para un motor DC de imanes permanentes adherente a conceptos digitales. En el diseño cuenta con un motor DC este cuenta con un encoder acoplado y se adapta un sensor tipo encoder que sirve para medir la velocidad y posición de este motor DC, en el diseño contamos con una pantalla lcd o display que nos permite visualizar los parámetros que ingresemos para controlar la velocidad o la posición. En el diseño también se cuenta con fundamentos matemáticos para el diseño del controlador del motor DC. En esta investigación podemos conocer las diferentes técnicas de control y mediante los modelos matemáticos del motor DC podemos implementar el tipo de control que deseamos. En el diseño contamos con la opción de adquisición de datos para un motor DC ya que el usuario puede utilizar motores de imanes permanentes de diferentes parámetros ya que con la adquisición de datos podemos obtener las funciones de transferencias para modelar y así obtener un controlador con una respuesta optima y precisa.

Los parámetros de las ganancias del PID se permiten controlar mediante potenciómetros para que el controlador nos permita verificar cómo funciona el control de un motor mediante el cambio de uno de sus parámetros ya sea manual o por medio del teclado se puede modificar el código de programación. Este trabajo se realizó como respuesta a las necesidades de los estudiantes universitarios de Pamplona ya que requiere como un diseño didáctico de un controlador para ver el comportamiento del sistema PID ya en un futuro realizar este trabajo en una herramienta del laboratorio, de la Universidad de Pamplona, y así poner a prueba los conocimientos del área de control y diseño mecatrónico. En el diseño del controlador se desea construir un dispositivo que sea fácil de usar y que las dimensiones sean mínimas para que los estudiantes puedan transportarlo fácilmente. En este diseño conceptual podemos en un futuro

## Formando **líderes** para la construcción de un nuevo **país en paz**



obtener mejoras ya que todos los días podemos evidenciar las nuevas mejoras que se incluyen en los dispositivos de medición ya que con su finalidad de estrás adicionales los controladores de para motores DC van a ser más precisos y mejorados.

### **Abstract**

In this investigative work, the conceptual design of a control system for a permanent magnet DC motor adhering to digital concepts is presented. In the design it has a DC motor, it has a coupled encoder and an encoder type sensor is adapted that is implemented to measure positions and velocities this DC motor, in the design we have an lcd screen or display that allows us to visualize the parameters that we enter to control speed or position. The design also has mathematical foundations for the conceptualization of the DC controller. We can know in this investigation the different controls techniques and through the mathematical models of the DC motor we can implement the type of control we want. In the design we have the option of data acquisition for a DC motor since the user can use permanent magnet motors of different parameters since with the data acquisition, we can obtain the transfer functions to model and thus obtain a controller with a optimal and accurate response.

### **Introducción**

Para el diseño y el desarrollo de la conceptualización del sistema de control DC son la alternativa perfecta en aplicaciones que necesitan alta fiabilidad, alta eficiencia y una buena interacción potencia-tamaño, dada su capacidad de proporcionar un bastante buen torque sobre un extenso rango de velocidades. Nos permite un amplio uso educativo con fines de que los estudiantes refuercen en la parte de control y automatización de sistemas por medio de controladores PID.

# Formando **líderes** para la construcción de un nuevo país en paz



Sin embargo, es muy importante que los sectores industriales existen procesos que es necesario del funcionamiento de un sistema de control para motores, por lo que se utilizan dispositivos que tengan un control cómo por ejemplo que son los variadores de velocidad y frecuencia.

Para la realización del modelado y diseño del controlador es esencial el uso de instrumentos computacionales ya que podemos analizar los resultados del sistema de control a partir de un estudio físico.

## **Capítulo I** **Generalidades.**

### **Planteamiento del problema y Justificación.**

Con la utilización de este diseño los laboratorios de la Universidad de Pamplona se va a tener un aporte bastante relevante para los estudiantes ya que podrían aplicar de manera practica y así comprender mejor la implementación del control PID a modelos físicos reales.

Así como además la conceptualización de este sistema de control busca plantear una idea para los alumnos de continuar aportando con proyectos que enriquezcan el razonamiento colectivo en trabajos investigativos o espacios que permitan a los alumnos el entendimiento.

### **Problema científico.**

Es aquí donde se recomienda que el resultado de esta indagación sea añadir un nuevo módulo de pruebas, para los estudiantes y laboratorios, que tengan un rango más extenso de experimentación, ósea que posibilite a los estudiantes cambiar y probar con el dispositivo de forma completa a partir de su hardware hasta su programa, en el que ellos logren de forma segura manipular y cambiar límites para poder hacer llevar a cabo sus controladores en un sistema que simula de forma más cercana a la verdad.

## Formando **líderes** para la construcción de un nuevo **país en paz**



La teoría del control para el diseño de controladores es aplicada a problemas mayormente frecuentados para el control de motores DC, la precisión y calidad que requieren los sistemas industriales es muy importante ya que hoy en día es obtener la exactitud deseada para que los procesos se ejecuten de manera autónoma y estable, la posición debe ser rápida y precisa a la hora de programar el controlador debemos de verificar los límites de cada variable para así desarrollar el control automático. Dada a las condiciones de rapidez y posición de los motores como ya se ha aclarado debe ser lo más instantáneo posible y viable para la utilización en las prácticas o clases de los estudiantes y obtener la capacidad de llevar a cabo esta clase de control y saber en qué momento y en qué proceso es necesario hacerlo.

### **Objetivo General**

Diseñar un sistema de control de manera conceptual para un motor DC de imanes permanentes adherente a conceptos digitales

### **Objetivos Específicos**

- Investigar los métodos de PWM para un sistema de control de motores DC.
- Reconocer o identificar las variables de control para un motor DC.
- Conocer la ingeniería de control de velocidad y posición para implementarla en el sistema.
- Diseñar un modelo de control para evaluar el grado de confiabilidad de la respuesta del controlador digital con respecto al modelo de caracterización de la respuesta de velocidad del motor DC.
- Describir la construcción del controlador digital del motor DC y cómo serán acoplados para un funcionamiento conjunto.

## Formando líderes para la construcción de un nuevo país en paz



- Presentar las conclusiones y contribuciones del trabajo de investigación, así como los posibles trabajos futuros que puedan derivar de esta investigación.

### **Estado del arte o antecedentes.**

#### **Implementación de dos estrategias de control para la velocidad de un motor DC.**

Este artículo muestra el proceso de sintonización y reajuste de un control PID y un control con base en un modelo de alusión, los cuales trabajan sobre un motor DC de imán persistente.

Hasta este instante se ha justificado el trabajo con el motor DC y con el control PID; ahora se explica la configuración en lazo cerrado, la cual se muestra en la figura. Son relevantes los bloques de compra de datos, la fase de potencia, el sensor, el motor y el algoritmo de control, ya que corresponden a las secciones de este artículo.

En el artículo se dedican ambas secciones iniciales a la configuración del experimento, de forma que logre ser replicado por los estudiosos interesados. Después, en otra parte, se recibe el modelo del motor, como una funcionalidad de transferencia, la cual es imprescindible para sintonizar los controladores. Se dedican 2 secciones más a los algoritmos de control PID y al aquí nombrado modelo de alusión; el resultado es la selección de las constantes de ajuste de cada controlador.

[1]

#### **Diseño de un controlador PID análogo para la velocidad de un motor DC mediante Matlab y pspice.**

El motor se divide en 2 piezas primordiales la parte eléctrica de la armadura y la parte mecánica. La armadura se modela como un circuito con resistencia  $R$  conectada en serie a un conductor  $L$ , una fuente de voltaje  $V$  y constante  $K_b$  de fuerza contra electromotriz ( $K=K_i=K_b$ ) en la

## Formando líderes para la construcción de un nuevo país en paz



armadura, donde  $K_i$  es la constante del par. El controlador PID se ocupa de arreglar el error que obtiene de la diferencia de la señal de acceso y la señal de salida del tacómetro con ganancia 0,0023, el error enmendado entra a una fase de ampliación anterior a entrar a la planta. [2]

### **Control PID de la velocidad de una banda transportadora para la clasificación de objetos.**

El objetivo de este plan de tesis es llevar a cabo un sistema de control PID para un motor de corriente directa con taco generador el cual se ocupa de desplazar una banda transportadora que preserva una rapidez constante dependiendo del costo esperado de objetos a trasladar. Como el motor es de un voltaje de 0 a 24 volts y la salida del PLC solo nos entrega 10 volts, se añadió una fase de potencia que da como consecuencia los 24 volts. Además, se le integró un convertidor de frecuencia a voltaje debido a que el motor entrega frecuencia y lo cual es necesario para el plan es voltaje. [3]

### **Control de velocidad y dirección de un robot de carrera autónomo.**

Los dos controles usan el algoritmo de control PID. El primer modelo de carrera está formado por 3 subsistema, un grupo de sensores, unidad de control y un sistema de actuadores y es un diminuto móvil de 4 ruedas, 2 de ellas determinan la dirección que seguirá el móvil. Permanecen ubicadas en la parte frontal del coche. Los motores son alimentados por un arreglo de transistores conocidos como puente H, que sirve como interfaz entre la unidad de control y los motores. Se ha seleccionado como unidad de control de un microcontrolador que procesara las señales generadas por un grupo de sensores instalados en el transporte para tomar elecciones de control implementando el algoritmo de control PID. [4]

Formando **líderes** para la construcción de un nuevo **país en paz**



## Capítulo II

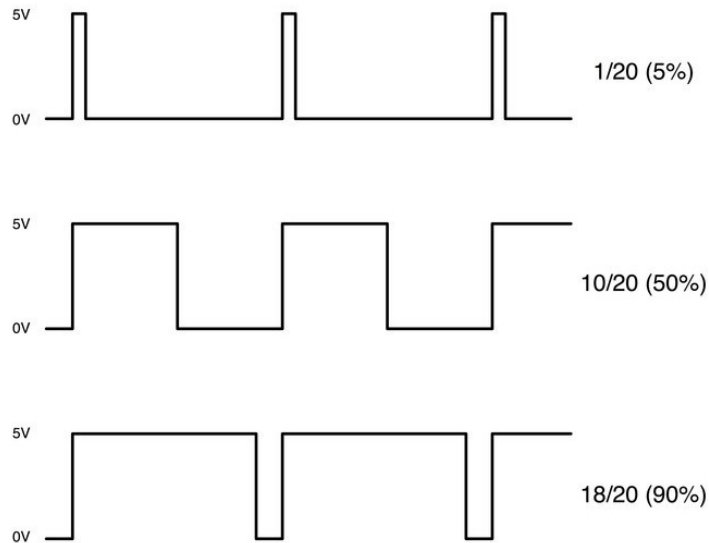
### Desarrollo Objetivo 1.

#### 2.1. Técnicas de Control PWM.

##### 2.1.1 Conceptualización de la señal PWM Arduino

La señal del Arduino es una señal por ancho de pulso se inspira de manera de onda cuadrada que se obtiene de manera proporcional del motor por los flancos de subida en el periodo de la onda cuadrada. Con esta relación se varia la velocidad de manera óptima con la señal Arduino. [5]

Lo que logramos es generar una señal de nivel bajo y alto de aquí obtenemos dos conceptos fundamentales. El periodo y la frecuencia (Hz). Los Hertz son los ciclos por segundo. [6]



*Ilustración 1: Señal de salida PWM Fuente: [7]*

Por las consideraciones anteriores la placa es capaz de producir una señal analógica modulada por ancho de pulso es la técnica llamada PWM. Esta técnica es utilizada por los usuarios para la

## Formando **líderes** para la construcción de un nuevo país en paz



implementación en circuitos digitales que necesiten señales analógicas producidas por el Arduino. Estas señales se pueden varias modificando el ancho relativo respecto al periodo y el resultado es el cambio de ciclo de trabajo lo cual están representadas por unidad de porcentaje. [7]

Cada Arduino cuenta con sus pines PWM, pero no todos con la misma relación de pines y las salidas analógicas toman valores entre 0 y 255. Según la hoja técnica del Arduino los pines marcados con ~ son pines que generan la señal PWM. [6]

**Tabla 1**

*En la tabla siguiente podemos observar la relación de pines de cada Arduino.*

| Arduino              | Pines PWM              | Frecuencia                    |
|----------------------|------------------------|-------------------------------|
| Mini, nano y Uno     | 3, 5, 6, 9, 10, 11     | 490 Hz pin 5 y 980 Hz pin 6.  |
| Mega adk             | 2 – 13, 44 – 46        | 490 Hz pin 4 y 980 Hz pin 13. |
| Leonardo, Micro, Yún | 3, 5, 6, 9, 10, 11, 13 | 490 Hz pin 3 y 980 Hz pin 11. |

*Nota.* \*De esta manera se puede verificar las relaciones de las salidas PWM con su frecuencia.

Fuente: Programar fácil. [6]

## Formando líderes para la construcción de un nuevo país en paz



Con respecto al programa Arduino podemos generar las señales PWM con el Arduino el usuario debe utilizar la función `analogWrite ()`. Es una función de Arduino que utiliza dos parámetros por donde podemos generar la señal PWM. [6]

`analogWrite (pin, valor)`

Cabe decir que el valor (pin) va a ser el número de la salida de la señal, el usuario puede escoger el pin de salida de su preferencia, cada una de estas salidas están marcadas con una señal ~.

El resultado es el número del ciclo que esta denotado por valores entre 0 y 255.

Eso significa que la función va a trabajar con un valor promedio de 0 a 5V es el resultado de un ciclo del 100%. Esta señal es muy útil para generar señales analógicas utilizadas para controlar motores y en fuentes de alimentación. [6]

Se observa en la siguiente tabla los resultados según la medida de una entrada analógica y salida del voltaje promedio.

**Tabla 2**

*Relación porcentaje del ciclo de trabajo.*

| Argumento <code>analogWrite ()</code> | Ciclo de trabajo | Voltaje promedio |
|---------------------------------------|------------------|------------------|
| 0                                     | 0%               | 0 v              |
| 63                                    | 25%              | 1.25v            |
| 76                                    | 30%              | 1.5v             |
| 127                                   | 50%              | 2.5v             |
| 178                                   | 70%              | 3.5v             |

## Formando líderes para la construcción de un nuevo país en paz



|     |      |       |
|-----|------|-------|
| 191 | 75%  | 3.75v |
| 229 | 90%  | 4.5v  |
| 255 | 100% | 5v    |

*Nota.* \*De esta manera estas cifras se deben a que las salidas PWM solo dan un valor entre 0 y

255. Fuente: Herramientas tecnológicas profesionales (hetpro) [6]

Para la respuesta de un ciclo de trabajo de 10% este indica que esta un 10 de un 100% es el nivel más alto. [8]

$$\text{Ciclo Dutty} = \frac{t}{T}$$

Es el tiempo sobre el periodo.

El valor de la tensión de salida al pasar el tiempo también puede ser igual a un valor deseado analógico. Con esta técnica un usuario puede controlar el brillo de un led o ya sea la velocidad de un motor.

### 2.1.2 Señal PWM con integrado 555

Es un circuito integrado mundialmente conocido 555 que es un temporizador en configuración astable en la cual a través de un MOSFET que nos ayuda entregar la corriente necesaria para poder manejar una carga inductiva que es un motor DC. Realiza el giro de un motor Dc y controla la velocidad. Este circuito es muy práctico no solamente podemos conectar un motor podemos conectar cargas para la intensidad de un led.

## Formando **líderes** para la construcción de un nuevo **país en paz**



El dispositivo 555 puede funcionar como oscilador, este temporizador es muy estable ya que puede producir pulsos con gran precisión algunas de sus características del 555 son [9]

- Se usa como un temporizador de horas hasta microsegundos.
- Trabaja en modo estable o monoestable.
- Se aplica como un oscilador.
- Trabaja como un divisor de frecuencia.
- Sirve para modular la frecuencia.
- También puede generar señales triangulares. [9]

### **2.2 Identificación de control PWM.**

Sobre las bases de las consideraciones anteriores la metodología aplicada incluye la técnica PWM de la salida del Arduino ya que transmite de una señal analógica de modulación para transmitir y mantener el control de la proporción de energía que se envía sin perder la potencia o padecer alguna distorsión al motor DC.

### **2.3 Representación de un motor DC Modelado.**

## Formando líderes para la construcción de un nuevo país en paz



Para la representación existen campos para el desarrollo del modelo matemático de una planta física que es muy primordial; hablamos de parametrizar la conducta dinámica de una planta mediante un grupo de representaciones matemáticas. Una representación matemática de un modelo o una planta es importante para el control de un sistema y podemos visualizar su comportamiento y mirar el equilibrio o precisión de la planta, hacer este diseño de controladores, examinar la contestación del sistema ante las diversas señales de prueba. La importancia de utilizar herramientas computacionales como el software Matlab, como el Simulink y Simscape es muy necesario en la participación para analizar la respuesta del controlador.

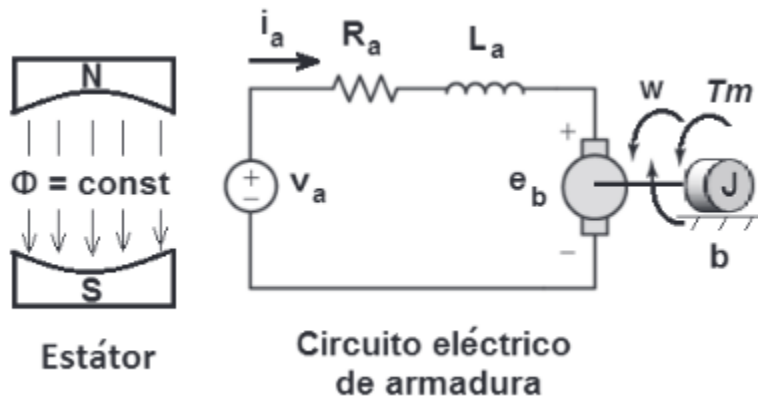
En referencia a la clasificación del motor DC se define las dimensiones pequeñas para la experimentación y son utilizados frecuentemente en la mecatrónica y robótica o equipos que se controlan remotamente.

### 2.3.1 Modelado de una planta para un Motor DC.

El modelado de una planta se puede obtener mediante criterios de selección de elementos primarios como es el flujo de corriente, velocidad y posición a través de ecuaciones diferenciales y sus funciones de transferencia. Los estudiantes aprenderán a la representación de un modelo lineal del motor DC a través de ecuaciones diferenciales y a partir de ello se obtienen sus funciones de transferencia.

La representación de un sistema con motor DC de imán permanente esta dado por el estator donde se encuentra el imán del motor y está representado por un par de polos magnéticos y el circuito eléctrico se representa como la armadura del motor DC.

Formando **líderes** para la construcción de un nuevo **país en paz**



*Ilustración 2 Representación del circuito con imán permanente. Fuente Srdanovic, V., Henao, O. B., & Ruiz, S. H. (2017).*

En la ilustración 3 podemos observar los elementos importantes del modelo de un motor Dc. Tenemos la resistencia con una inductancia de la bobina L y una tensión que genera a todo el circuito de la armadura.

Las ecuaciones representadas en el circuito se representan a continuación. [10]

$$V(t) = Ri(t) + L \frac{di(t)}{dt} + Ea(t)$$

$$L \frac{di(t)}{dt} = V(t) - Ri(t) - Ea(t) \quad (1)$$

La sección mecánica está dada por:

$$Tm(t) = J \frac{dm(t)}{dt} + Bm(t)$$

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



$$J \frac{dw(t)}{dt} = Tm(t) - Bm(t) \quad (2)$$

La ecuación del momento de inercia del motor está dada por el torque del motor y el coeficiente de fricción y la derivada para el cálculo de la velocidad. Luego se asume la constante proporcional  $K_a$ .

$$Ea(t) = K_a w(t) \quad (3)$$

Luego la relación electromecánica que circula en  $K_m$  es una corriente que evidencia claramente en el circuito.

$$Tm(t) = K_m i(t) \quad (4)$$

Ya obtenemos las cuatro ecuaciones para luego aplicar la transformada en las ecuaciones.

$$Ls i(s) = V(s) - Ri(s) - Ea(s) \quad (5)$$

$$Js w(s) = Tm(s) - Bw(s) \quad (6)$$

$$Ea(s) = K_a w(s) \quad (7)$$

$$Tm(s) = K_m i(s) \quad (8)$$

Se sustituye ecuación 7 y ecuación 8 en la ecuación 5.

$$Ls \frac{Tm(s)}{K_m} = V(s) - R \frac{Tm(s)}{K_m} - K_a w(s)$$

$$V(s) = \frac{(R + Ls) Tm(s)}{K_m} + K_a w(s) \quad (9)$$

Podemos obtener la velocidad la podemos sacar de la ecuación 6.

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



$$W(s) = \frac{Tm(s)}{Ls + B} \quad (10)$$

Hacemos sustitución.

$$V(s) = \left( \frac{R + Ls}{Km} + \frac{Ka}{Js + B} \right) * Tm(s)$$

$$V(s) = \frac{(R + Ls)(Js + B) + Ka Km}{Km(Js + B)} * Tm(s)$$

Obtenemos la ecuación de salida con la de la entrada. [10]

$$\frac{Tm(s)}{V(s)} = \frac{Km(Js + B)}{LJs^2 + (RJ + LB)s + RB + KmKa} \quad (11)$$

### 2.3.2 Obtención del sistema a partir de ecuaciones.

Para poder realizar el diseño del diagrama en el software Simulink debemos saber implementar las ecuaciones diferenciales en el programa para poder visualizar el diagrama de bloques con sus respectivas ecuaciones ya que con ellas podemos calcular su velocidad exacta con un transmisor o sensor igualmente su posición del motor. Caracterizamos cada una de las ecuaciones y luego agregamos a un subsistema para implementarlo en el control PID.

$$L \frac{di(t)}{dt} = V(t) - Ri(t) - Ea(t) \quad (1)$$

$$J \frac{dw(t)}{dt} = Tm(t) - Bm(t) \quad (2)$$

$$Ea(t) = Ka w(t) \quad (3)$$

Formando **líderes** para la construcción de un nuevo **país en paz**



$$T_m(t) = K_m i(t) \quad (4)$$

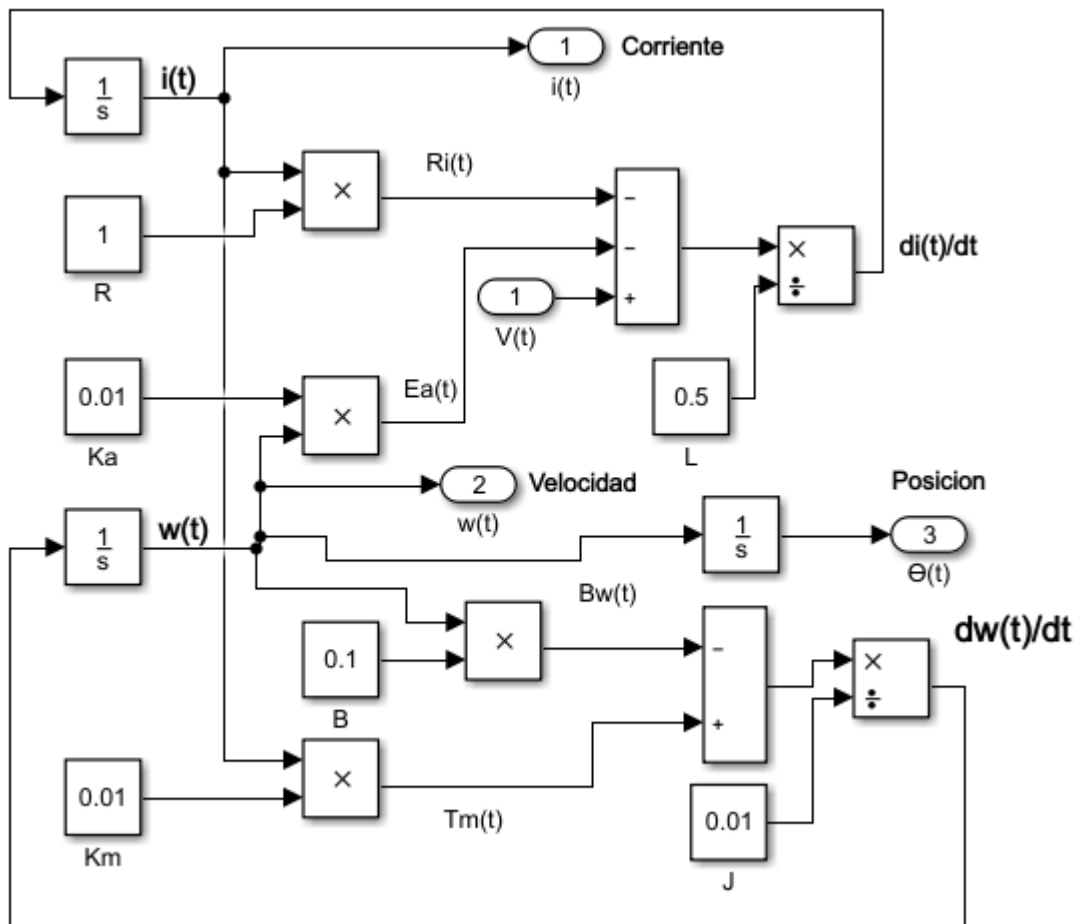


Ilustración 3 Modelo del diagrama de bloques representado en el programa Simulink. Fuente: Carlos salgado.

Especificamos cada componente con su respectiva ecuación de función de transferencia:

$$\text{Torque} - \text{Voltaje} \quad \frac{T_m(s)}{V(s)} = \frac{K_m(Js + B)}{LJs^2 + (RJ + LB)s + RB + K_mKa} \quad (12)$$

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**

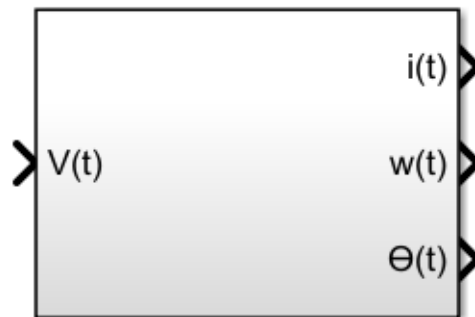


$$\text{Fuerza Contraelectromotriz} - \text{Voltaje} \frac{Ea(s)}{V(s)} = \frac{Km Ka}{LJs^2 + (RJ + LB)s + RB + KmKa} \quad (13)$$

$$\text{Corriente armadura} - \text{Voltaje} \frac{i(s)}{V(s)} = \frac{Js + B}{LJs^2 + (RJ + LB)s + RB + KmKa} \quad (14)$$

$$\text{Velocidad angular} - \text{Voltaje} \frac{w(s)}{V(s)} = \frac{Km}{LJs^2 + (RJ + LB)s + RB + KmKa} \quad (15)$$

$$\text{Posición} - \text{Voltaje} \frac{\Theta(s)}{V(s)} = \frac{Km}{s(LJs^2 + (RJ + LB)s + RB + KmKa)} \quad (16)$$



**Motor Dc**

*Ilustración 4 Subsistema del motor DC en el programa Simulink en el software Simulink.*

*Fuente: Carlos salgado.*

### 2.3.3 Espacio de Estados.

En el desarrollo de espacios de estados se puede obtener partiendo de las ecuaciones obtenidas anteriormente. [10]

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



$$x1 = \omega$$

$$= \dot{\omega}$$

$$x2 = i$$

$$\dot{x}2 = i$$

Reescribiendo se obtiene la siguiente ecuación.

$$\begin{aligned}\dot{x}1 &= -\frac{B}{J}x1 + \frac{Km}{J}x2 \\ \dot{x}2 &= -\frac{R}{L}x2 + \frac{Ka}{L}x1 + \frac{1}{L}v\end{aligned}$$

La representación del modelo es: [10]

$$\begin{aligned}\begin{bmatrix} \dot{x}1 \\ \dot{x}2 \end{bmatrix} &= \begin{bmatrix} -\frac{B}{J} & \frac{Km}{J} \\ -\frac{Ka}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v \\ \begin{bmatrix} y1 \\ y2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix}\end{aligned}$$

#### 2.3.4 Parametrización del modelo matemático del motor DC.

En la tabla podemos observar los datos de los parámetros respectivos de un motor DC y así poder visualizar la respuesta del sistema de acuerdo a los parámetros del motor.

**Tabla 3**

*Parámetros del motor DC con imán permanente.*

| Parámetro      | Descripción                    |
|----------------|--------------------------------|
| R [ $\Omega$ ] | Resistencia armadura del motor |

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**




---

|                        |                                            |
|------------------------|--------------------------------------------|
| L [H]                  | Inductancia armadura del motor             |
| Ka [V/(rad/s)]         | Constante de la fuerza contraelectromotriz |
| J [kg.m <sup>2</sup> ] | Momento de inercia de armadura             |
| Km [N.m/A]             | Constante de torque electromecánico        |
| b [(N.m) /(rad/s)]     | Coeficiente de fricción viscosa            |

---

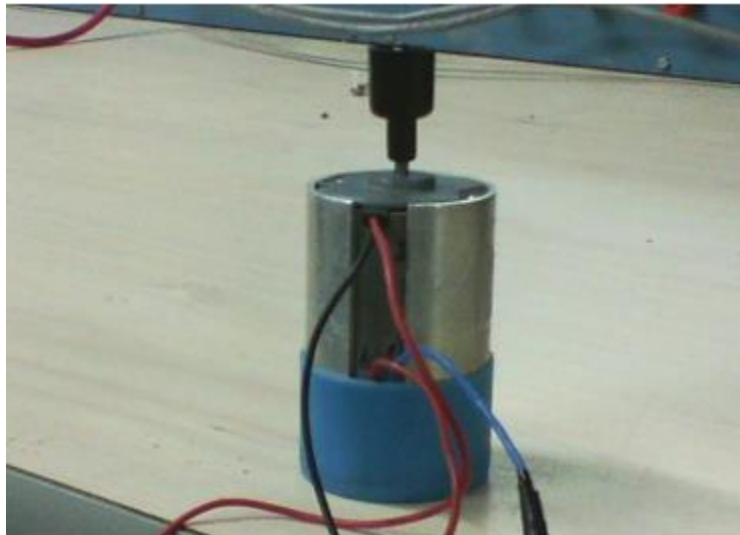
Nota\* Para el desarrollo de un modelo simulado es necesario los parámetros del motor DC.

### 2.3.5 Medición de parámetros para la obtención del modelo.

Para la obtención de los parámetros se debe utilizar los siguientes materiales o equipos para realizar las pruebas pertinentes.

- Motor DC
- Osciloscopio
- Amperímetros
- Voltímetro
- Tacómetro

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



*Ilustración 5 Motor 12V DC.*

Para obtener los parámetros de un motor Dc utilizamos como ejemplo un motor de 12 voltios de corriente continua que posee un tacómetro que nos ayuda a obtener el parámetro de la velocidad.

#### **2.3.5.1 Obtención parámetro de la resistencia armadura ( $R_a$ ).**

Se obtiene el parámetro utilizando un multímetro y se mide directamente en la armadura y el valor promedio de las muestras fue de  $5.2 \Omega$  y el factor de correlación de dicha armadura será de un 10% así que el valor total de la armadura será de  $R_a 0.52 \Omega$ . [10]

#### **2.3.5.2 Obtención parámetro de la inductancia de armadura ( $L_a$ )**

Se obtiene el parámetro de la inductancia conectando el amperímetro y con el reóstato o potenciómetro. Luego se conecta a la tensión y variamos la tensión a diferentes valores para obtener los datos de la tensión de la corriente. Para obtener la inductancia hacemos el uso de la ley de ohm. [10]

Formando **líderes** para la construcción de un nuevo **país en paz**



Ley de ohm  $Z = \frac{v}{I} \quad L = \frac{\sqrt{Z^2 - R}}{2\pi F}$

**Tabla 4**  
*Parámetros obtenidos.*

| Tensión | Corriente | Impedancia |
|---------|-----------|------------|
| 5,2     | 0,58      | 9,25       |
| 6       | 0,66      | 9,33       |
| 7,1     | 0,82      | 8,79       |
| 8,1     | 0,93      | 8,89       |
| 9       | 1         | 3          |
| 10,4    | 1,25      | 8,6        |
| 11,5    | 1,3       | 7,75       |
| 12      | 1,55      | 8,1        |

Nota\* Se saca el valor promedio de  $Z_a = 8.6 \, \Omega$  y se utiliza el valor de la resistencia  $R = 0.52 \, \Omega$  y una frecuencia de 60Hz.

$$L = \frac{\sqrt{8.6^2 - 0.52}}{2\pi * 60} = L = 0.021H$$

### 2.3.5.3 Obtención parámetro de la de fuerza contra electromotriz.

La obtención del parámetro de la fuerza electromotriz se realizan mediciones bajo diferentes tensiones y velocidad del motor para la determinar del valor de la constante  $K_b$  se realiza mediante la ecuación: [10]

$$K_b = \frac{V_{in} - R_a I_a}{\omega n}$$

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



**Tabla 5**  
*Parámetros obtenidos para la fuerza electromotriz.*

| Voltaje | Corriente | Velocidad<br>(RPM) | Velocidad<br>(Rad/s) | Kb(V/rad/s) |
|---------|-----------|--------------------|----------------------|-------------|
| 5       | 0,13      | 1475               | 164,33               | 0,03181     |
| 6       | 0,14      | 2100               | 210,2                | 0,02826     |
| 7       | 0,15      | 2600               | 292,79               | 0,02438     |
| 8       | 0,16      | 3500               | 315,2                | 0,02526     |
| 9       | 0,16      | 3700               | 356,55               | 0,02436     |
| 10      | 0,16      | 4100               | 428,7                | 0,02347     |
| 11      | 0,17      | 4700               | 471,52               | 0,02256     |
| 12      | 0,17      | 5600               | 565,7                | 0,02078     |

Nota\*El promedio. Kb= 0,02535 [V/rad/s].

La ecuación de la constante está dada de la siguiente manera:

$$Kb = Km \left[ \frac{N * m}{A}; \frac{V}{rad/s} \right]$$

#### 2.3.5.4 Obtención del parámetro de la fricción viscosa.

Se obtiene partiendo de una ecuación:

$$J \frac{d\omega n}{dt} = Kb * Ia - B\omega n$$

Pasando al estado estable:

$$J \frac{d\omega n}{dt} = 0$$

El resultado despejando B de la ecuación:

Formando **líderes** para la construcción de un nuevo país en paz



$$B = \frac{Kb * Ia}{\omega n}$$

Se reemplazan los valores y se halla la constante de fricción: [10]

$$B = \frac{(0,02515)(0,155)}{350,405} = 0,011126^{-3} \frac{Nm}{rad/s}$$

#### 2.3.5.5 Pruebas y parámetros de la constante de inercia.

En el mismo sentido anteriormente las pruebas se deben realizar elaborando varias simulaciones de pruebas en el motor para poder obtener los parámetros correspondientes el tiempo de parada lo debemos obtener y la constante de tiempo y así obtenemos todos los parámetros del motor para obtener la constante de inercia. [10]

**Tabla 6**

*Resultados sobre las pruebas de inercia.*

| Tiempo de parada (s) | Tiempo de parada(ms) | $\tau m$ (ms) |
|----------------------|----------------------|---------------|
| 0,97                 | 980                  | 320           |
| 0,98                 | 982                  | 321           |
| 0,84                 | 860                  | 273,43        |
| 0,65                 | 680                  | 233,53        |
| 0,55                 | 570                  | 183,43        |
| 0,75                 | 860                  | 273,43        |
| 0,68                 | 570                  | 183,443       |
| 0,75                 | 840                  | 273,53        |
| 0,84                 | 930                  | 320,43        |
| 0,89                 | 980                  | 320           |

Nota\* el promedio  $T_m = 276,33ms$  la inercia se debe hallar mediante una ecuación.

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



$$Tm = \frac{J}{B}$$

Como ya se conocen algunos valores se despeja la inercia del motor:

$$J = Tm * B$$

$$J = (276,33)(0,011125^{-3}) = 3,075^{-3} \text{ Kg m}^2$$

2.3.5.6 Parámetros del motor obtenidos.

**Tabla 7**

*Parámetros obtenidos del motor.*

| Nombre                      | Parámetros | Unidades                  |
|-----------------------------|------------|---------------------------|
| Resistencia de la armadura  | Ra         | 0,52                      |
| Inductancia de la armadura  | La         | 0,021H                    |
| Fuerza contra electromotriz | Kb         | 0,02535V/rad/s            |
| Fricción viscosa            | B          | 0,011126E-3N-<br>m/rad/s  |
| Inercia del motor           | J          | 3,075E-3Kg-m <sup>2</sup> |

### 2.3.6 Simulación a través de los parámetros obtenidos del modelo del sistema.

La simulación se obtiene por medio del software Matlab, se ingresan los parámetros obtenidos del motor Dc y reemplazamos en la función de transferencia de la ecuación (15) de la velocidad angular sacada anteriormente.

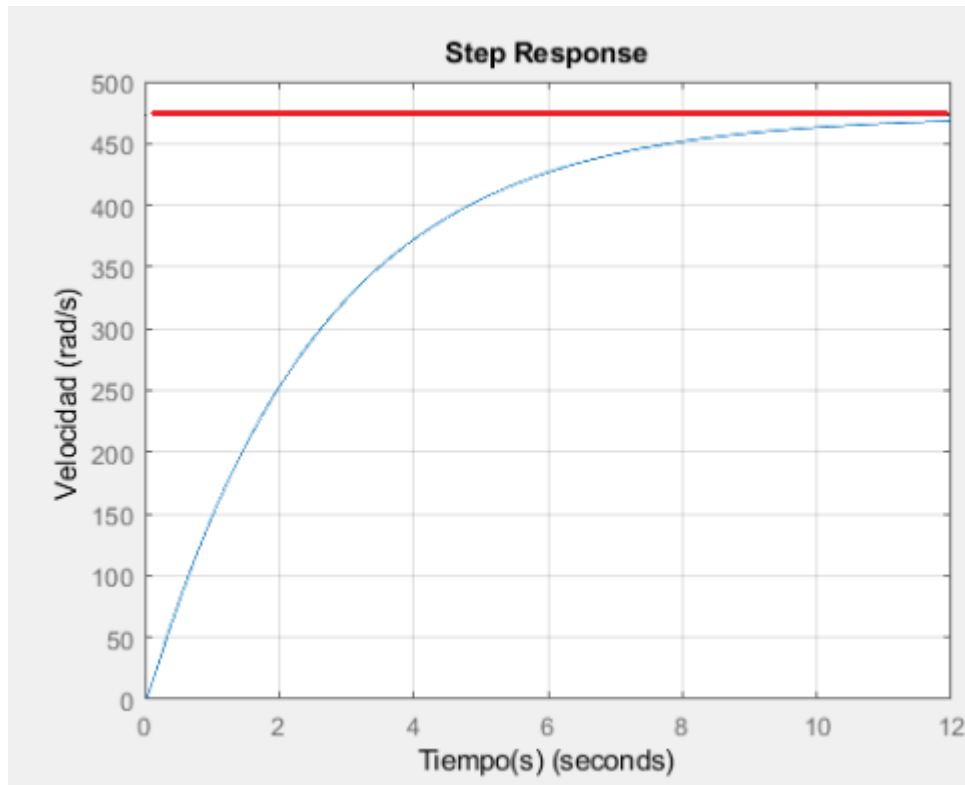
Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



```
ParametrosMotor.m  x +
-
-   Va=12;
-   J=0.003074; %Inercia mecánica
-   B=0.000011125; %Fricción viscosa de los rodamientos
-   K=0.02515; %Constante mecánica del motor
-   R=0.54; %Resistencia del circuito de armadura
-   L=0.023; %Inductancia del circuito de armadura
-   num=K;
-   den=[ (J*L) ((J*R)+(L*b)) ((b*R)+K^2) ];
-   t=0:0.001:12;
-   step (Va*num,den,t)
-   xlabel('Tiempo(s)');
-   ylabel('Velocidad (rad/s)')
-   grid on
```

*Ilustración 6 Obtención de datos por medio de Matlab Fuente: Carlos s.*

Formando **líderes** para la construcción de un nuevo país en paz



*Ilustración 7 Respuesta mediante el software Matlab*

Tal como se observa en la gráfica el motor se estabiliza en a una velocidad de 475 rad/s y un tiempo estimado de 12 segundos, se realizó mediante la situación planteada y or medio de la parametrización del motor.

### **3. Controlador PID.**

#### **3.1 Implementación sistemas de control**

A lo largo de los planteamientos hechos un motor DC con la aplicación de un controlador PID el motor puede controlarse de manera estable a través de diferentes técnicas analíticas y

Formando **líderes** para la construcción de un nuevo país en paz



computacionales. Se puede determinar un sistema de control por medio de amplificadores operacionales donde se puede ver cada configuración y determinar cuál es la más óptima en este caso se puede utilizar el software para simular sistemas por medio de sus ecuaciones y ver claramente su respuesta de tal manera que en la ilustración 8 observamos como sería un controlador con PID con un accionador y el sistema del motor como ya se ha aclarado anteriormente y contamos también con el transmisor todo esto se encierra en un circuito de lazo cerrado.

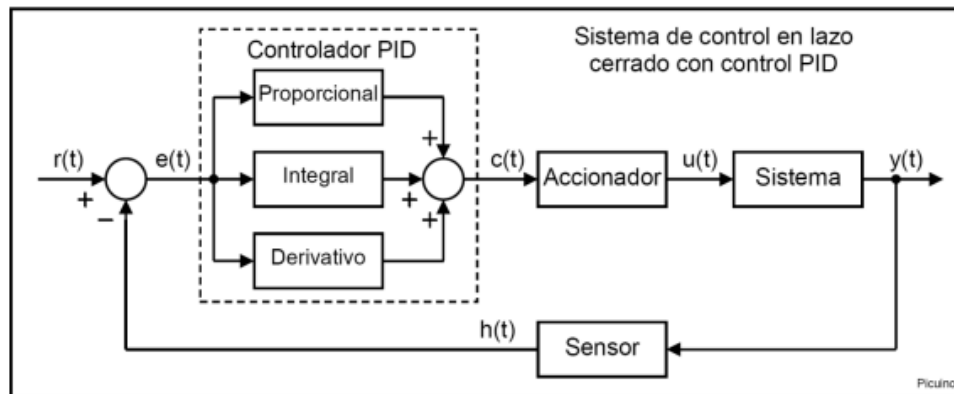
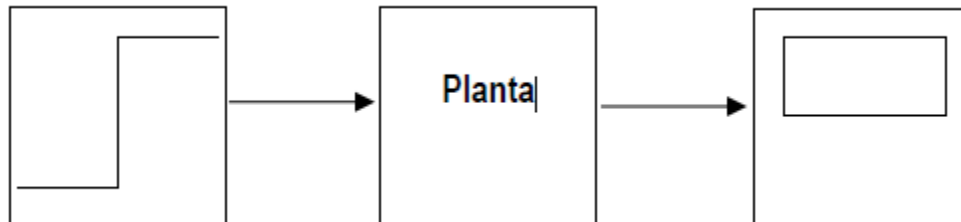


Ilustración 8 Sistema de control con controlador PID. [11]

### 3.2 Diseño del controlador por medio del Método Ziegler – Nichols.

En este diseño e implementación del método se propone por medio de una entrada unitaria que se le ingresa a la planta para obtener el punto de inflexión por medio de Matlab.

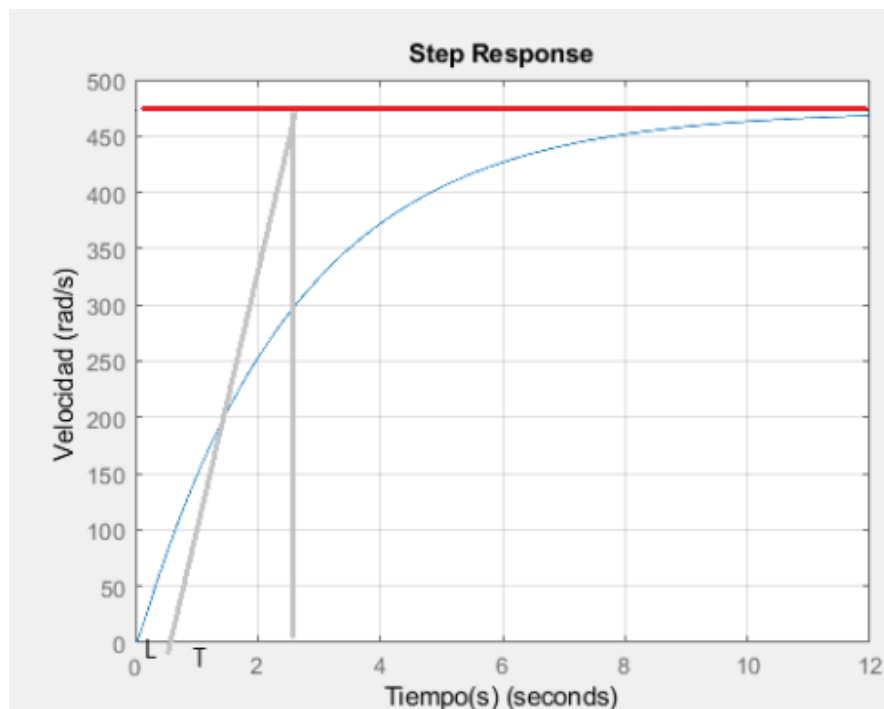
Formando **líderes** para la construcción de un nuevo **país en paz**



*Ilustración 9 Sistema de escalón unitario*

Se realiza mediante los parámetros físicos del motor implementados en la planta para obtener la respuesta de acuerdo al motor deseado.

### 3.3 Primer método Ziegler – Nichols.



*Ilustración 10 Recta tangente al punto de inflexión*

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



**Tabla 8**

*Fórmulas para los diferentes tipos de controlador por el método Ziegler -Nichols*

| Tipo de controlador     | Kp                  | Ti              | Td    |
|-------------------------|---------------------|-----------------|-------|
| <b>P</b>                | $\frac{T}{K L}$     | $\infty$        | 0     |
| <b>PI</b>               | $0.9 \frac{T}{K L}$ | $\frac{L}{0.3}$ | 0     |
| <b>PID</b>              | $1.2 \frac{T}{K L}$ | 2 L             | 0.5 L |
| $0.1 \leq L/T \leq 0.3$ |                     |                 |       |

*Ilustración 11 Forma Ziegler y Nichols [11]*

$$L = 0.5 \quad T = 2.1 \quad K = 475 \quad \frac{L}{T} = \frac{0.5}{2.1} = 0.238 \quad \text{SI CUMPLE}$$

**Tabla 9**

*Datos obtenidos para los controladores P, PI y PID*

| <b>P</b>   | <b>KP</b> | <b>TI</b> | <b>TD</b> |
|------------|-----------|-----------|-----------|
|            | 0.00884   | $\infty$  | 0         |
| <b>PI</b>  | 0.00795   | 0.16      | 0         |
| <b>PID</b> | 0.0106    | 1         | 0.25      |

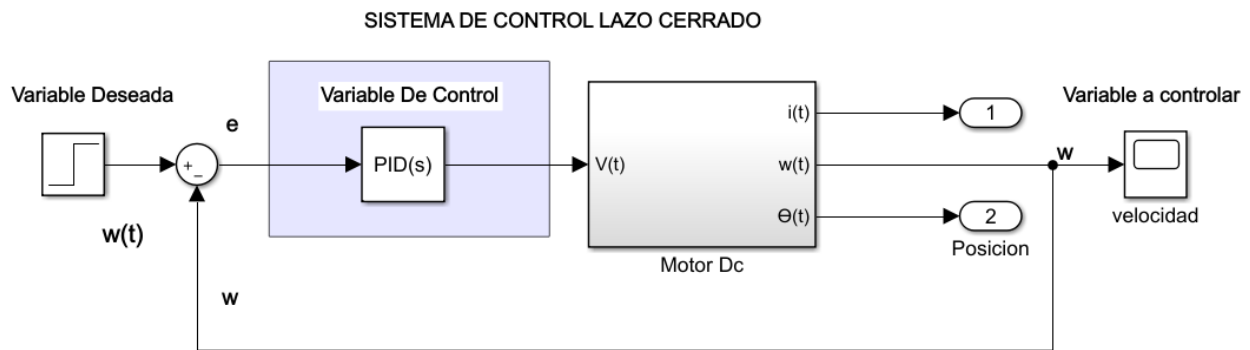
Nota\* Parámetros obtenidos de las constantes Kp, Ti y Td.

Formando **líderes** para la construcción de un nuevo **país en paz**



$$G_c(S) = K_p \left( 1 + \frac{1}{T_i S} + T_d s \right)$$

Para ver la respuesta se puede obtener de un sistema continuo que le implementamos un controlador PID



*Ilustración 12 Implementación de un controlador PID.*

Las variables del PID fueron calculadas matemáticamente y son ingresadas al software Matlab.

Formando **líderes** para la construcción de un nuevo **país en paz**



Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced windup, external reset, and signal tracking. You can tune the PID gains automatically using the Simulink Control Design.

Controller: PID Form: Parallel

Time domain:

☒ Continuous-time  
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Source: internal

Proportional (P): 0.0106

Integral (I): 1

Derivative (D): 0.5

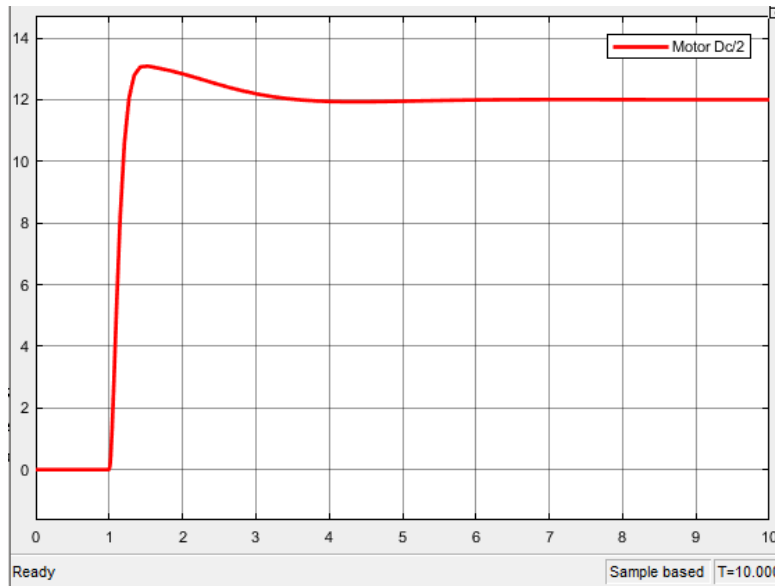
Filter coefficient (N): 1

Select Tuning Method: Transfer Function Based (PID Tuner App) Tune...

*Ilustración 13 Parámetros para el PID Fuente: Elaboración propia.*

Podemos ver la respuesta del controlador PID.

Formando **líderes** para la construcción de un nuevo país en paz



*Ilustración 14 Respuesta del controlador PID. Fuente: Elaboración propia*

Respuesta del lazo de control cerrado para un motor Dc, se evidencia que tiene un arranque rápido y el controlador se estabiliza en un step de 12v. la respuesta tiene una reacción rápida y con un sobre pico y se estabiliza.

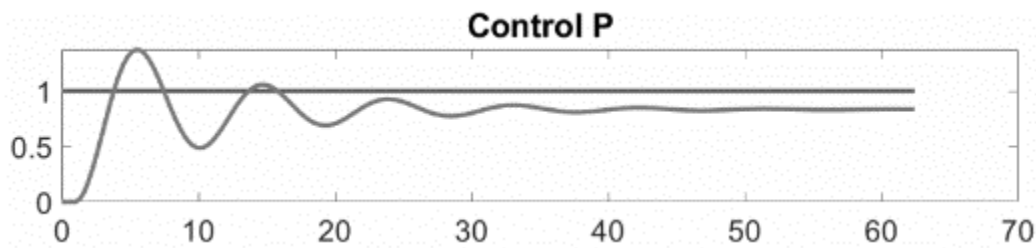
### **3.4 Método para el control de velocidad/ posición.**

El método óptimo para analizar la velocidad y posición se emplea un modelo matemático de la parametrización del motor. Se realiza otros métodos para la identificación del sistema o planta para así obtener sus funciones y ecuaciones del motor. La herramienta a definir es Matlab y Simulink para realizar el proceso pertinente de verificación de las ecuaciones para el diseño del controlador PID.

### **4. Tipos de sintonizaciones que permita la flexibilidad del sistema.**

La sintonización se busca para mejorar la respuesta del controlador PID, en este proceso se mejora gran parte de la forma de respuesta que se quiere en esta situación se logra una rapidez o postura del sistema solicitado por los estudiantes o también esta sintonización se puede llamar por prueba y error en las constantes del controlador PID y además va a ser capaz de arreglar probables perturbaciones en el sistema.

#### 4.1 Acción proporcional:



*Ilustración 15 Respuesta de un sistema de control proporcional Fuente: Sergio Andrés Castaño Giraldo*

La acción proporcional funciona como una señal tiende a tener una respuesta rápida y oscilatoria esta señal de control no funciona de manera óptima ya que la acción de incremento en la variable proceso que es una acción directa. También funciona para un proceso de control de respuesta rápida en el proceso. El controlador proporcional esta determinado por el transmisor. En este controlador la acción que se necesita es directa. [12]

Ecuación continua:

$$U_p(t) = K_p * e(t)$$

Ecuación discreta:

$$U_p(t) = K_p * e(k)$$

## Formando **líderes** para la construcción de un nuevo **país en paz**



### 4.2 Acción derivativa:

La acción derivativa es el cambio de la salida del controlador respecto al valor del proceso. En cuanto al cambio de error la acción derivativa puede producir un cambio significativo de gran magnitud por lo tanto puede producir respuestas rápidas al iniciar. El comportamiento de este controlador el cambio de error es cero. [12]

Ecuación continua

$$U_i(t) = K_d * \frac{de(t)}{dt}$$

Ecuación discreta:

$$U_i(t) = \frac{e(K) - e(K - 1)}{T} * K_p * T_d$$

### 4.3 Acción integral

La acción integral como su nombre lo dice integra el valor de proceso hasta llegar al valor deseado, este sistema va tomando los datos anteriores del error y se va integrando hasta llegar a un equilibrio en el sistema.

Ecuación continua:

$$U_i(t) = K_i * \int_0^t e(t) dt$$

Ecuación discreta:

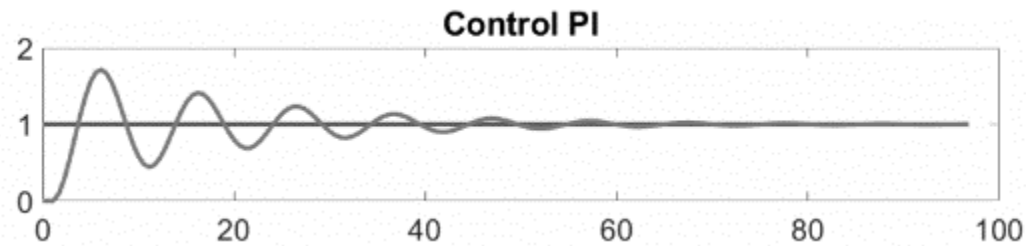
$$U_i(t) = \frac{K_p * T}{T_i} * e(k) + U_i(k - 1)$$

### 4.4 Aplicación de un controlador PI.

Formando **líderes** para la construcción de un nuevo **país en paz**



Esta aplicación se encarga de combinar la proporcional e integral también se encarga de acelerar la respuesta del proceso hasta llegar al valor de referencia utilizando la acción integral que se encarga para ajustar la variable de error del sistema para que este tenga un error nulo y así funcione más rápidamente sin provocar oscilaciones en el sistema.



*Ilustración 16 Respuesta de un sistema de control proporcional integral. Fuente: Sergio Andrés Castaño Giraldo*

La ecuación de la acción proporcional integral está dada por:

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt$$

#### 4.5 Aplicación de un controlador PD.

Al establecer esta acción es muy esencial contar con la acción proporcional ya que nos ayuda a llevar el valor del proceso mucho más rápido y la acción derivativa nos ayuda a mantener un error constante variante en el tiempo este control no es casi utilizado ya que no tiene buena precisión y la respuesta tiende a oscilar y el sistema es inestable.

Y la ecuación de esta acción está dada por:

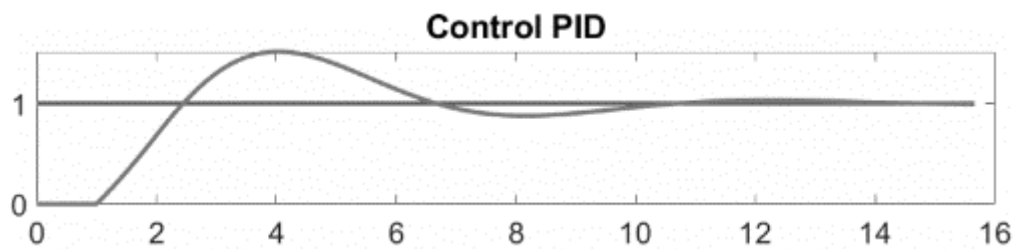
$$U(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

Formando **líderes** para la construcción de un nuevo **país en paz**



#### 4.6 Aplicación de un controlador PID.

Se identifica que es una acción que combina los tres controladores, es una acción que podemos obtener buenos resultados y por lo tanto es un controlador muy estable ya que con las tres ganancias puede mejorar el sistema o no ya que pueden ser muy sensibles a errores ya depende a cada sistema.



*Ilustración 17 Respuesta de un sistema de control proporcional integral derivativa Fuente: Sergio Andrés Castaño Giraldo*

$$U(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

#### 5. Sistema de control por ecuaciones diferenciales lazo cerrado.

La forma de sistema de control discretos se basa por medio de ecuaciones diferenciales para cada una de sus acciones de control y su ecuación del controlador lo podemos ver en la figura 18.

Formando **líderes** para la construcción de un nuevo **país en paz**

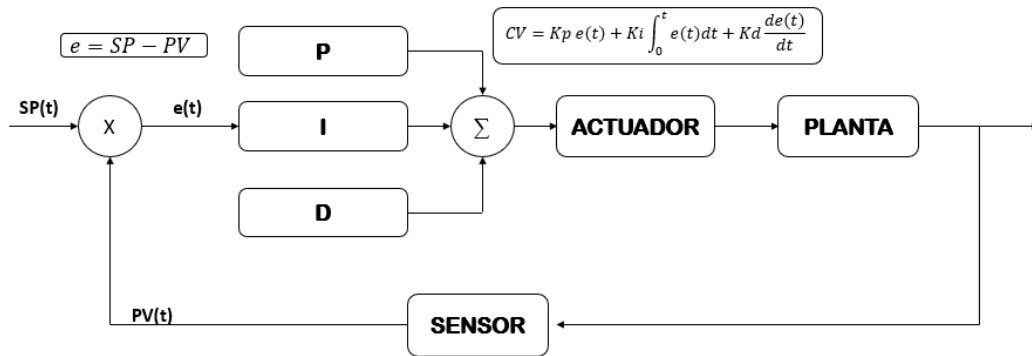
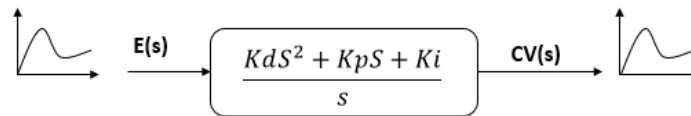


Ilustración 18 La ecuación del control PID.



El método para realizar la implementación de un PID discreto de velocidad es mediante la aproximación rectangular, en la cual varia la señal de control, debemos usar un retenedor a la entrada en este caso es un retenedor de orden cero

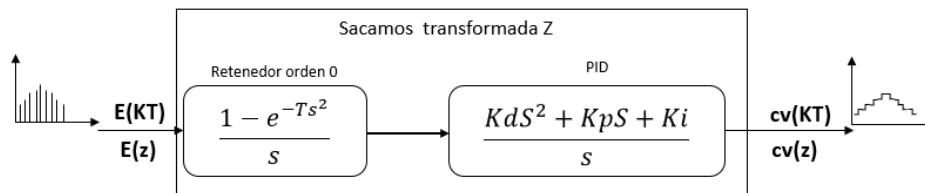


Ilustración 19 Retenedor orden cero

Desarrollamos la transformada z:

$$\begin{aligned}
 PID &= \frac{CV(z)}{E(z)} = Z \left[ \frac{1 - e^{-Ts}}{s} \left( \frac{KdS^2 + KpS + Ki}{s} \right) \right] \\
 &= Z \left[ \frac{1 - e^{-Ts}}{s} \left( Kp + \frac{Ki}{s} + KdS \right) \right]
 \end{aligned}$$

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



$$\begin{aligned}
 &= (1 - z^{-1})Z \left[ \frac{Kp}{s} + \frac{Ki}{s} + Kd \right] \\
 &= (1 - z^{-1}) \left[ \frac{Kp}{(1 - z^{-1})} + \frac{KiTs z^{-1}}{(1 - z^{-1})^2} + \frac{Kd}{Ts} \right] \\
 \frac{Cv(z)}{E(z)} &= \frac{Kp(1 - z^{-1}) + KiTs z^{-1} + \frac{Kd}{Ts} (1 - z^{-1})^2}{(1 - z^{-1})} \\
 \frac{Cv(z)}{E(z)} &= Kp + KiTs \frac{1}{z - 1} + \frac{Kd}{Ts} \frac{z - 1}{z}
 \end{aligned}$$

Las constantes son obtenidas de un sistema continuo para la ecuación de diferencia de un PID discreto.

$$\begin{aligned}
 cv(z) - cv(z)z^{-1} &= KpE(z) - KpE(z)z^{-1} + KiTsE(z)z^{-1} + \frac{Kd}{Ts}E(z) + \frac{Kd}{Ts}E(z)z^{-2} - 2\frac{Kd}{Ts}E(z)z^{-1} \\
 cv(z) - cv(z)z^{-1} &= \left( Kp + \frac{Kd}{Ts} \right) E(z) + \left( -Kp + KiTs - 2\frac{Kd}{Ts} \right) E(z)z^{-1} + \frac{Kd}{Ts} E(z)z^{-2}
 \end{aligned}$$

La implementación digital del PID será:

$$cv(n) = cv(n - 1) + \left( Kp + \frac{Kd}{Ts} \right) e(n) + \left( -Kp + KiTs - 2\frac{Kd}{Ts} \right) e(n - 1) + \frac{Kd}{Ts} e(n - 2)$$

En donde:

- Kp: Constante proporcional.
- Ki: Constante Integral.
- Kd: Constante derivativa.
- Ts: Tiempo de muestreo.
- Cv(n): Acción de control actual.

## Formando **líderes** para la construcción de un nuevo **país en paz**



- $Cv(n-1)$ : Acción de control de hace una muestra.
- $e(n)$ : error actual.
- $e(n-1)$ : error de hace una muestra.
- $e(n-2)$ : error de hace dos muestras.

Ya obtenidas estas consideraciones, se procede a seleccionar el sistema embebido, se tienen en cuenta los tiempos de muestreo, procedimientos matemáticos. Se debe considerar las condiciones físicas de la planta.

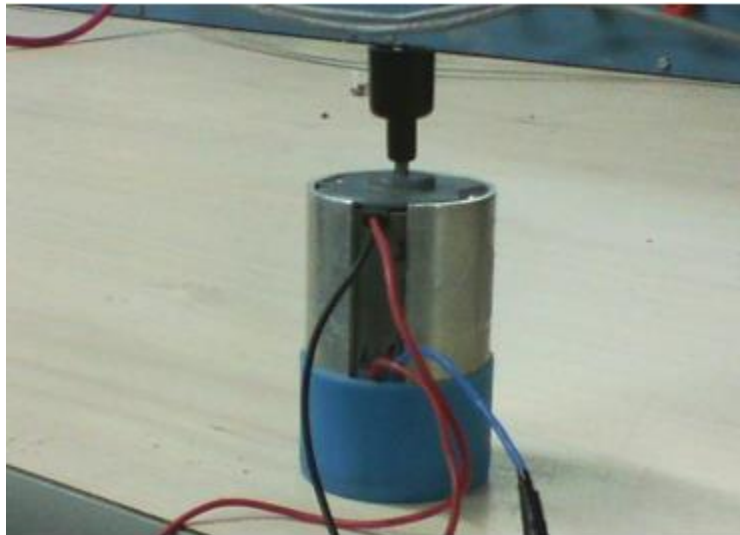
### **6. Componentes para el diseño de un controlador digital**

#### **6.1 Identificación del Motor DC.**

Para la determinación del controlador se propone usar un motor de imán permanente. Éste va a ser el factor a mantener el control de (como se verá después, no deberá ser este motor en específico,

Las características de este tipo de motor 12V- 4600 RPM, especialmente diseñado para prototipos donde necesitemos censar las RPM (Revoluciones por Minuto) de nuestro motor, se puede calcular la posición o dirección.

Formando **líderes** para la construcción de un nuevo **país en paz**



*Ilustración 20 Motor Dc*

Para el cálculo de velocidad y posición se acopla en el motor DC de imanes permanentes un encoder ranurado que nos permite medir con el sensor las revoluciones por minuto y los grados para la posición.

## **6.2 Modulo Sensor Tipo Herradura.**



*Ilustración 21 Modulo Sensor Tipo Herradura. [13]*

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



Modulo Sensor Tipo Herradura KLH-512 cuenta con 2 canales A y B de cuadratura, los cuales nos pueden ayudar visualizar la velocidad, así como la posición, el sensor de herradura puede ser alimentado en un rango de 1.8V ~ 5V. [13]

Características del sensor:

- Modelo: F249 (FC-03)
- Sensor: KLH 512
- Voltaje de entrada 5V
- La salida es de tipo analógica y digital
- Tiene un circuito integrado LM393 [13]

### 6.3 Arduino UNO.



*Ilustración 22 Placa Arduino UNO. [14]*

El controlador principal a utilizar en este trabajo investigativo es el Arduino UNO ya que es de bajo costo es muy fácil de manejar y de realizar la programación deseada ya que cuenta con muchas ayudas para programarla y cuenta con la salida PWM y la potencia deseada para el control del motor DC.

## Formando líderes para la construcción de un nuevo país en paz



También podemos contar con otras tarjetas programables y algunas de ellas son:

- **Arduino MEGA 2560:** Esta tarjeta es una alternativa a utilizar, pero es de mayor costo que la tarjeta Arduino UNO, pero las características van a ser las mismas, pero con mayores números de pines a utilizar y tiene un tamaño mucho más grande y robusta. Esta tarjeta es más viable para aplicaciones de simulaciones reales con software ya que posee mayor grado de precisión en las lecturas de datos.
- **Rasberry Pi:** La placa Rasberry es otra alternativa para poder realizar el proyecto ya que se puede programar y utilizar para otras funciones más. Pero es una placa que básicamente es un ordenador ya que tenemos que instalar un sistema operativo para poder trabajarla y pues es mucho más trabajo reprogramarla, esta placa es muy utilizada para proyectos mucho más complejos y su precio es el más elevado del mercado a comparación de las otras placas.

### 6.4 Puente H



*Ilustración 23 Puente H. [15]*

Comercialmente podemos obtener en el mercado puente H para motores de baja y alta potencia. El Driver L298N sirve para el control de un motor de baja potencia y muy usado para proyectos

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



didácticos en la electrónica. El puente H de tipo IGBT con circuito 555, que también funciona como variador de velocidad. Mediante el cual se pueden controlar motores de alta potencia con el fin de regular la velocidad angular y la posición del motor DC.

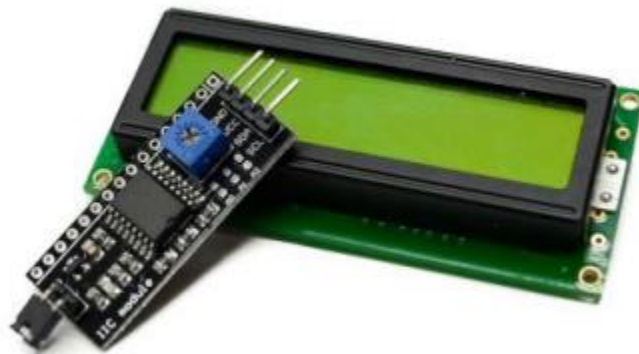
### 6.5 Potenciómetro



*Ilustración 24 Potenciómetro. Fuente SSDIELECT ELECTRONICA SAS*

Un potenciómetro rotativo que permite variar su resistencia de forma manual como para poder controlar el Setpoint de la posición y/o velocidad del sistema de control.

### 6.6 Pantalla LCD con adaptador I2C.



*Ilustración 25 LCD con adaptador I2C. [11]*

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



La pantalla LCD nos facilita mucho en los proyectos ya que nos permite visualizar lo que queremos ver. En esta imagen anterior podemos ver que tiene un módulo llamado I2C que nos permite reducir las conexiones normales que tiene la LCD a solo cuatro conexiones de salida. La LCD se debe alimentar a 5v del Arduino y a gnd y también tiene dos pines para la comunicación de ella. Estos pines se deben conectar a A4 y A5 de las entradas del Arduino.

#### **6.7 Conexión de la placa Arduino USB.**



*Ilustración 26 Cable Arduino USB. [11]*

Tenemos el cable de alimentación del Arduino que es conectado a la computadora que también sirve para el envío de datos de la programación.

Formando **líderes** para la construcción de un nuevo **país en paz**



## 6.8 Cableado



*Ilustración 27 Cables. [2]*

Tenemos el tipo de cable que se va a llevar a cabo las conexiones con el Arduino este tipo de cable es muy utilizado para intensidades de corriente bajas.

## 6.9 74hc04



*Ilustración 28 Integrado 74hc04 Fuente: Vistronica.*

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



Es de la familia de las compuertas lógicas llamada compuerta NOT o negadora nos sirve para la implementación de nuestro controlador.

#### **6.10 Adaptador de corriente**



*Ilustración 29 Adaptador de corriente [11]*

Es un convertidor de corriente alterna 120vAc a corriente directa 12vDc sirve para la alimentación de la placa Arduino.

Formando **líderes** para la construcción de un nuevo país en paz



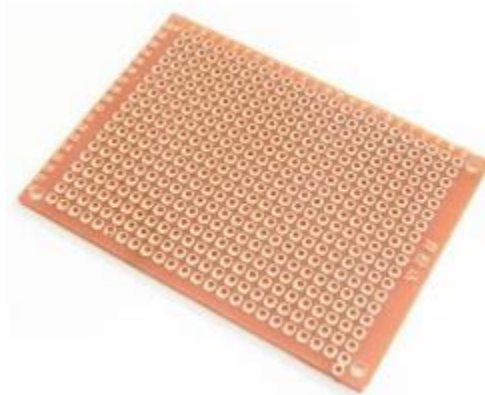
### 6.11 Conectores tipo Jack banana



*Ilustración 30 Conectores tipo Jack banana. Fuente Ferretronica.*

Los conectores banana son un tipo de Jack muy usados en instrumentos

### 6.12 Placa de prototipos.



*Ilustración 31 Placa perforada. [11]*

Es una placa fácil de implementar ya que en ella estarán las conexiones de los componentes electrónicos.

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



## 7. Programación para el diseño del controlador del motor Dc.

La definición de la programación de nuestro controlador puede ser implementada en el software Arduino, es un software libre y fácil de usar ya que se puede obtener mucha información para la implementación de nuestro controlador.

### 7.2 Driver

Se declaran como salidas digitales sus pines IN1 e IN2 si el estado de cada uno es bajo o alto dependiendo de sus estados y su tercer pin será ENB donde será conectada el pin digital PWM que controla el motor. [11]

La declaración en el programa Arduino será:

- `pinMode (pinA, INPUT);`
- `pinMode(pinB, INPUT);`
- `pinMode(PWM_salida, OUTPUT);`

### 7.3 Pantalla LCD

Para la programación de nuestra pantalla LCD con el módulo I2C se deben utilizar las librerías. [11]

- `#include <LiquidCrystal_I2C.h>`
- `LiquidCrystal_I2C lcd (0x27, 16, 2);`

Para hacer el llamado en la pantalla ubicamos la dirección y luego hacemos la impresión en la LCD.

- `lcd.setCursor(0, 0);`

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



- `lcd.print("");`

#### 7.4 Sensor encoder.

Se usa en la programación una interrupción tipo CHANGE para cualquier flanco ascendente o descendente. En la salida de los pines digital 2 y 3 del Arduino actúa la interrupción.

- `attachInterrupt(digitalPinToInterrupt(encA), encoder, CHANGE);`
- `attachInterrupt(digitalPinToInterrupt(encB), encoder, CHANGE);`

#### 7.5 Potenciómetro

El potenciómetro nos sirve para poder regular la velocidad o posición en nuestra programación lo implementamos como un valor del Setpoint y se conecta en el pin de entradas análogas del Arduino y el llamado en la programación se hace con `analogWrite ()`;

- `sp = analogRead (POT_sp)*(RPM/1023);`

De esta manera podemos calcular el valor de las revoluciones del motor DC y con otras funciones podemos variar las ganancias de las constantes P, I y D.

#### 7.6 Funciones al controlador de la posición y de velocidad.

La medición a la velocidad y posición debemos de tener en cuenta que existen motores que tienen acoplado el encoder y otros no. para el caso de que el motor no traiga encoder debemos buscar uno en el mercado y así acolarlo al rotor del motor y así configurar el Arduino para este caso la lectura.

Formando **líderes** para la construcción de un nuevo país en paz



Para motores que tienen cajas reductoras se debe aplicar una formula ara hallar el número de pulsos final en el eje del rotor.

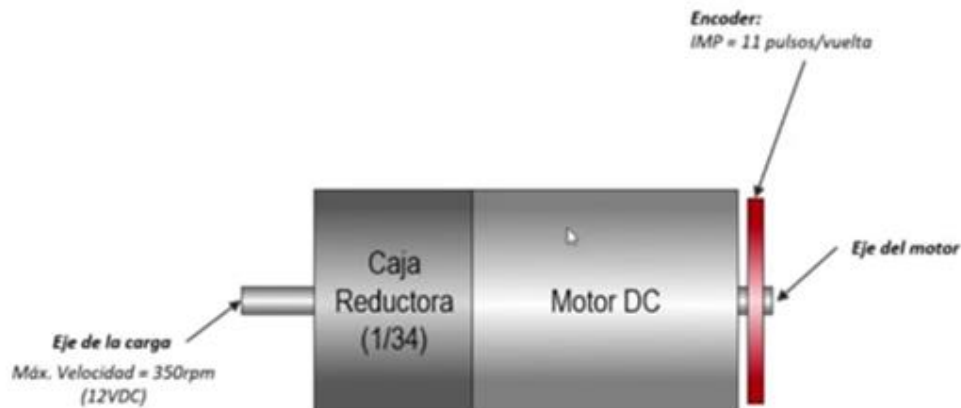


Ilustración 32 Relación de pulsos encoder

$$Vel_{ejecarga(Rpm)} = \text{frec encoder, pulsos/s} * \left( \frac{1 \text{ Rev}}{IMP * \text{Relacion reducción}} \right) * \left( \frac{1 \text{ min}}{60 \text{ seg}} \right)$$

Así podemos determinar el control de velocidad y posición

### 7.6.1 Ecuación empleada al controlador de la posición

Para la determinación de la ecuación del controlador del motor se implementa en la programación mediante la fórmula que tiene en cuenta los numero de pulsos del encoder por los grados que esta implementado como un contador en la programación. [11]

$$\text{Posición} = 360 * \text{cuenta} / (4.0 * \text{PPR})$$

## Formando **líderes** para la construcción de un nuevo **país en paz**



### 7.6.2 Funciones empleadas al control de la velocidad

Para la determinación de la ecuación del controlador simplemente se emplea una fórmula de velocidad y también se emplea un tiempo de muestreo en la programación para obtener un comportamiento más rápido y estable en el tiempo de respuesta de nuestro controlador. [11]

- $pv = contador * (60.0 / PPR);$
- $contador = 0;$

Al calcular la función de la velocidad se implementa en el PID y se evalúa el error y la velocidad deseada.

### 7.7 Implementación del controlador PID

En este propósito se transcribe las ecuaciones diferenciales calculadas a la programación Arduino y con las interrupciones planteada la programación llega a tener un error mínimo para el sistema de control.

- $cv = cv1 + (kp + kd/Tm) * error + (-kp + ki * Tm - 2 * kd/Tm) * error1 + (kd/Tm) * error2;$
- $cv1 = cv;$  // recursividad escalera
- $error2 = error1;$
- $error1 = error;$

### 7.8 Resultados de la implementación en programa Arduino

A lo largo de los planteamientos hechos en este trabajo investigativo podemos controlar un motor siempre y cuando tenga el driver y el modelado del motor adecuado para el control del motor.

Formando **líderes** para la construcción de un nuevo país en paz



### 7.8.1 PID posición

Tal como se observa la respuesta de la posición a un Angulo de 90 grados podemos observar que el controlador se comporta de manera eficiente ante los anteriores planteamientos se observa claramente en la ilustración 32.



*Ilustración 33 Controlador PID. Fuente: Elaboración propia.*

Si la respuesta la sometemos a diferentes cambios de posición deberá cumplir con las expectativas de la implementación de un controlador PID comportándose de manera precisa.

Formando **líderes** para la construcción de un nuevo país en paz



*Ilustración 34 Respuesta a diferentes cambios de posición Fuente elaboración propia.*

Si al controlador le hacemos un pequeño cambio a la constante proporcional este tiende a ser una respuesta mucho más rápida, pero con un sobre impulso mayor lo podemos ver en la ilustración 34.

Formando **líderes** para la construcción de un nuevo **país en paz**

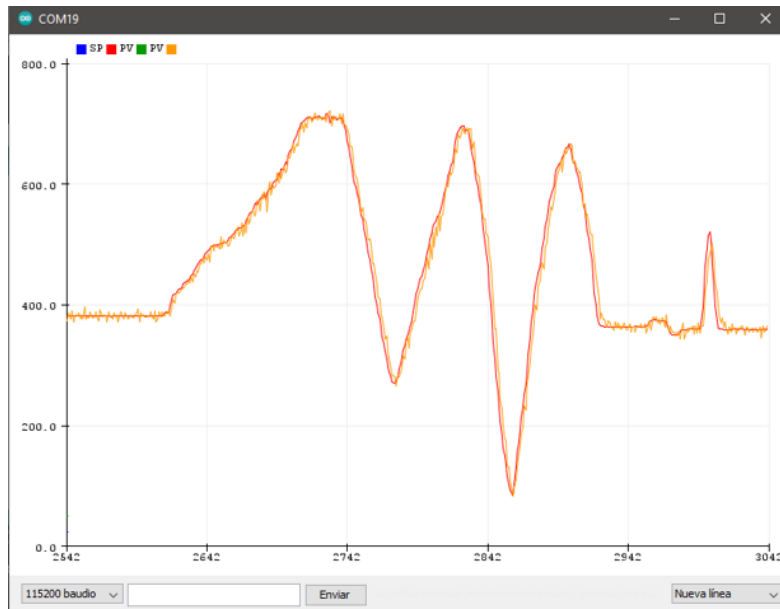


*Ilustración 35 Respuesta a cambio de la constante de  $K_p$ . Fuente elaboración propia.*

### 7.8.2 PID velocidad

En esta respuesta variamos la velocidad a diferentes unts para ver la reacción de respuesta. La línea roja es el valor del SetPoint y la naranja de la velocidad comportamiento del sistema.

Formando **líderes** para la construcción de un nuevo país en paz



*Ilustración 36 Respuesta de la velocidad con el controlador PID implementado. Fuente: Elaboración propia*

En la respuesta del controlador de velocidad podemos observar que el sistema trata de trabajar de manera estable pero no alcanza ser muy preciso.

## 8. Construcción del controlador digital del motor CD

Para el concepto del controlador se utilizó el software SolidWorks ya que es una herramienta que se puede utilizar muy fácil gracias a que podemos encontrar tutoriales que nos permite manejarlo fácilmente para la creación de cada pieza del controlador y el resultado final lo podemos observar en la ilustración 36.

Formando **líderes** para la construcción de un nuevo país en paz



Ilustración 37 Diseño del concepto del controlador didáctico digital. Fuente elaboración propia.

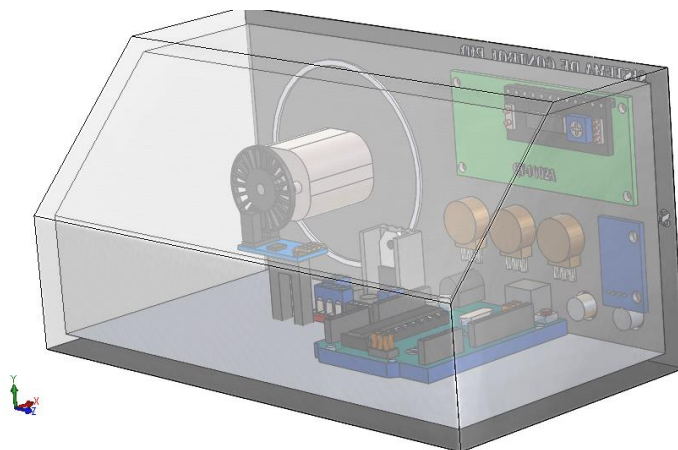


Ilustración 38 Visualización interna de los componentes del controlador. Fuente elaboración propia.

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



## Conclusiones

La precisión de un sistema de control depende de la precisión que se obtenga en los parámetros del motor en nuestra resistencia, inductancia, constante de inercia y fricción etc. Es importante utilizar los métodos correctos de medición y parametrización de cualquier sistema.

En este artículo se demostró el proceso del modelado físico de un motor DC de imanes permanentes en la cual obtuvimos los parámetros que caracterizan el modelo con sus respectivas ecuaciones para que las personas o estudiantes de la universidad puedan ponerlo en práctica de manera experimental la adquisición de datos de un motor de corriente continua.

En la metodología se incluye la técnica PWM o modulación por ancho o de pulso del Arduino ya que transmite de una señal analógica con controlar la velocidad de un motor de corriente continua con imán permanente sin perder potencia o sufrir distorsión por interferencias.

Es muy esencial la utilización de las herramientas computacionales para el diseño de un sistema de control de motor de corriente continua y con la herramienta que ofrece Simulink y Matlab podemos observar cada una de las respuestas del sistema de control.

Según el diseño del controlador de velocidad y posición las simulaciones realizadas en el software de Matlab y ambos sistemas de control responden satisfactoriamente. En la siguiente etapa está prevista la implementación física de sistema de control PID diseñada.

# Formando líderes para la construcción de un nuevo país en paz



## Bibliografía

- [1] S. N. & M. R. Rairán Antolines, «Implementación de dos estrategias de control para la velocidad de un motor DC.,» Ingeniería e Investigación, 29(2), 100-106., 2009.
- [2] R. G. Gil Garcés, «Control PID para el control de velocidad de un motor DC.,» Universidad Tecnológica de Pereira, Pereira:, 2014.
- [3] A. M. D. C. JAIME, «Control PID De La Velocidad De Una Banda Transportadora Para La Clasificación De Objetos,» Tesis Doctoral. Universidad Autónoma de Zacatecas., Zacatecas., 2008.
- [4] Á. H. FERNANDO, «Control de velocidad y dirección de un robot de carreras autónomo.,» Instituto Politécnico Nacional, MÉXICO D.F., 2011.
- [5] S. O. H. S. Cardona Herrera, «Construcción de módulo convertidor CC-CC elevador, para prácticas en el laboratorio del programa ingeniería mecatrónica,» Universidad Tecnológica de Pereira, Pereira , 2015.
- [6] J. G. Carmenate, «programarfacil,» 2022. [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/pwm-con-arduino-analogico/>.
- [7] M. Á. Abellán, «programoergosum,» 09 julio 2019. [En línea]. Available: <https://www.programoergosum.es/tutoriales/salidas-analogicas-con-arduino/>.
- [8] Spain, «Solectroshop,» 26 Agosto 2020. [En línea]. Available: <https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38>.
- [9] marinjl, «El 555,» [En línea]. Available: <https://www.uv.es/marinjl/electro/555.htm>.
- [10] S. A. CASTAÑO, «controlautomaticoeducacion,» [En línea]. Available: <https://controlautomaticoeducacion.com/analisis-de-sistemas/modelo-de-motor-dc/#:~:text=Description-,La%20armadura%20del%20motor%20DC%20se%20modela%20como%20si%20tuviera,tensi%C3%B3n%20generada%20en%20la%20armadura..>
- [11] I. Iranzo Juan, «Diseño e implementación de un sistema de control para motores de continua mediante un microcontrolador Arduino,» Universitat Politècnica de València, 2021.
- [12] C. P. Martín, «picuino,» 10 12 2018. [En línea]. Available: <https://www.picuino.com/es/control-pid.html>.
- [13] Ferretrónica, «Ferretrónica,» [En línea]. Available: <https://ferretronica.com/products/modulo-sensor-de-velocidad-f249-tipo-herradura-klh-512>.
- [14] Servotronic, «Servotronic,» [En línea]. Available:

Formando **líderes** para la construcción de un nuevo **país en paz**



<https://www.servotronik.com.co/index.php/producto/arduino-uno-r3-compatible-atmega-328p/>.

- [15] M. G. & G. López, «Control De Posición PID Para Motor De Cd Con Raspberry Pi 3 Sin Carga Mecánica (Pid Position Control For Dc Motor With Raspberry Pi 3 Without Mechanical Load,» Pistas Educativas, 42(137)., Mexico, 2020.

## ANEXOS

### Anexo 1. Conexiones del controlador en el Software Proteus.

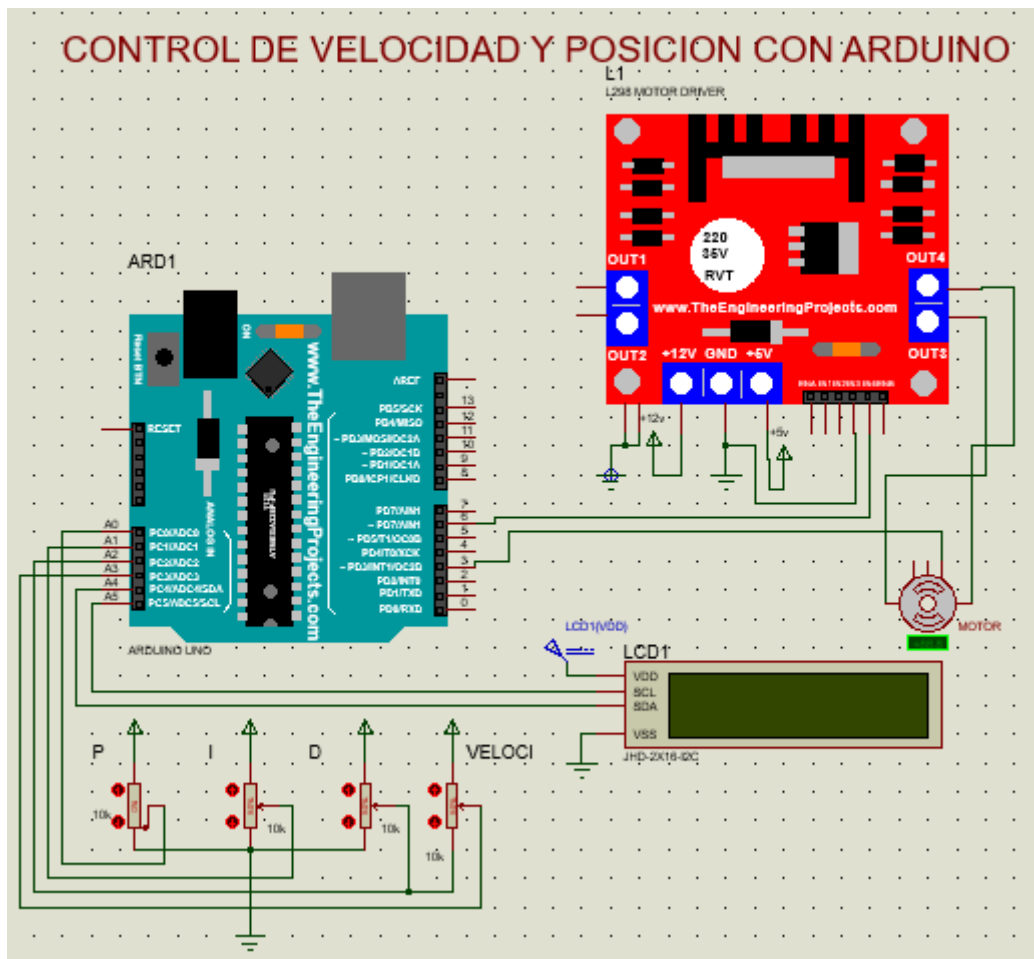


Ilustración 39 Diseño cableado del controlador en Proteus.

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



## Anexo 2. Programación de Arduino del controlador posición.

```

Posición$
# include <Wire.h> //Esto es para i2C
# include <SSD1306Ascii.h> //i2C OLED
# include <SSD1306AsciiWire.h> //i2C OLED
# define I2C_ADDRESS 0x3C
# define RST_PIN - 1
SSD1306AsciiWire oled;
float OLEDTimer = 0; //Temporizador para el intervalo de actualización de la pantalla
//Pines I2C:
//Arduino: SDA: A4 SCL: A5
//-----
const byte encA = 2; // Entrada de la señal A del encoder.
const byte encB = 3; // Entrada de la señal B del encoder.
const byte PWMA = 5; // Salida PWM a la primera patilla del motor a través de un puente H.
const byte PWMB = 6 ; // Salida PWM a la segunda patilla del motor a través de un puente H.
const int Ppot = 0; //pin 0
const int Ipot = 1; //Pot controls Ki
const int Dpot = 2; //Pot controls Kd
int kP = 0; // contiene el parámetro P
int kI = 0; // contiene el parámetro I
int kD = 0; // contiene el parámetro D
int motorCommand = 0; // mantiene el comando final para motor
int tick=0; // mantiene el desplazamiento del tick del codificador.
int currentError = 0; // error de desplazamiento. Viene de tampling tick
int previousError = 0; // contiene el error del paso de tiempo anterior

```

## Formando **líderes** para la construcción de un nuevo país en paz



```
int deltaError = 0;           // contiene la derivada del error
float accumulatedError = 0; // retiene el error acumulado (a lo largo del tiempo)
// siguientes variables utilizadas para controlar el tiempo de ciclo
int loopTimeMs = 20; // tiempo de bucle deseado en milisegundos
long nowish = 0;
long timeDelta = 0;
long timeMarker = 0;

void encoderEvent() {
  if(digitalRead(encA) == digitalRead(encB)){
    tick++;
  } else {
    tick--;
  }
}

void setup(){
  Serial.begin(115200);      // Configura la velocidad en baudios del terminal serie.
  Wire.begin(); //iniciar I2C
  Wire.setClock(800000L); //reloj mas rapido

  //Referencia OLED
  # if RST_PIN >= 0
    oled.begin(&Adafruit128x32 , I2C_ADDRESS , RST_PIN);
```

## Formando **líderes** para la construcción de un nuevo **país en paz**



```
# else // RST_PIN >= 0
  oled.begin(&Adafruit128x32 , I2C_ADDRESS);
# endif // RST_PIN >= 0
  oled.setFont(Adafruit5x7);
  oled.clear(); //borrar pantalla
  oled.set2X(); //
  oled.println("SISTEMA.C  "); // imprime un mensaje de bienvenida
  oled.println("  PID");
  oled.set1X();
  delay(90000);
  OLEDTimer = millis(); //iniciar el temporizador
  oled.clear();
  displayPermanentItems();
| //---interrupción de inicio
  attachInterrupt(digitalPinToInterrupt(encA), encoderEvent, CHANGE);
}
int fetchAnalog(int potPin){
  int potVal = analogRead(potPin); // devuelve el valor en el rango [0,1023]
  return int(potVal * .008797654); // normalizar al valor en el rango [0,9]
}
// dividiendo por 10 porque no necesitamos valores mayores de 0-9.
void updatePIDValues(){
  kP = fetchAnalog(Ppot);
```

---

## Formando **líderes** para la construcción de un nuevo país en paz



```
kI = fetchAnalog(Ipot);
kD = fetchAnalog(Dpot);
}
void calculateAccumulatedError(){
    // una vez que cruzamos la marca de error 0, el error pasado acumulado
    // ya no nos ayuda. Restablecer a 0.
    if(currentError == 0){
        accumulatedError = 0;
        return;
    }
    accumulatedError += currentError;
    // ayuda con la liquidación (sobreacumulación del error)
    accumulatedError = constrain(accumulatedError, -20, 20);
}
// deltaError se usa para el componente D del controlador.
// para eso necesitamos la derivada del error.
// dado que esto solo se llama una vez por ciclo, no necesitamos dividir por tiempo
void calculateDeltaError(){
    previousError = currentError;
    currentError = tick;
    deltaError = currentError - previousError;
}
void generateMotorCommand(){
    // este es el quid del controlador PID: agregar los componentes P, I, D
```

---

## Formando **líderes** para la construcción de un nuevo país en paz



```
int finalCommand = kP * currentError + kI * accumulatedError + kD * deltaError;
// nos aseguramos de permanecer dentro de los límites del controlador del motor.
motorCommand = constrain(finalCommand, -255, 255);
}
void updateMotor() {
  if(motorCommand >= 0) {
    digitalWrite(PWMA, HIGH);
  }else{
    digitalWrite(PWMA, LOW);
  }
  analogWrite(PWMB, abs(motorCommand));
}
void displayPermanentItems()
{
  //imprime los elementos permanentes en la pantalla
  oled.setCursor(0,0); //(x [píxeles], y[líneas])
  oled.print("P: ");
  oled.print(kP);
  oled.setCursor(0,1);
  oled.print(", I: ");
  oled.print(kI);
  oled.setCursor(0,2);
  oled.print(", D: ");
  oled.println(kD);
}
```

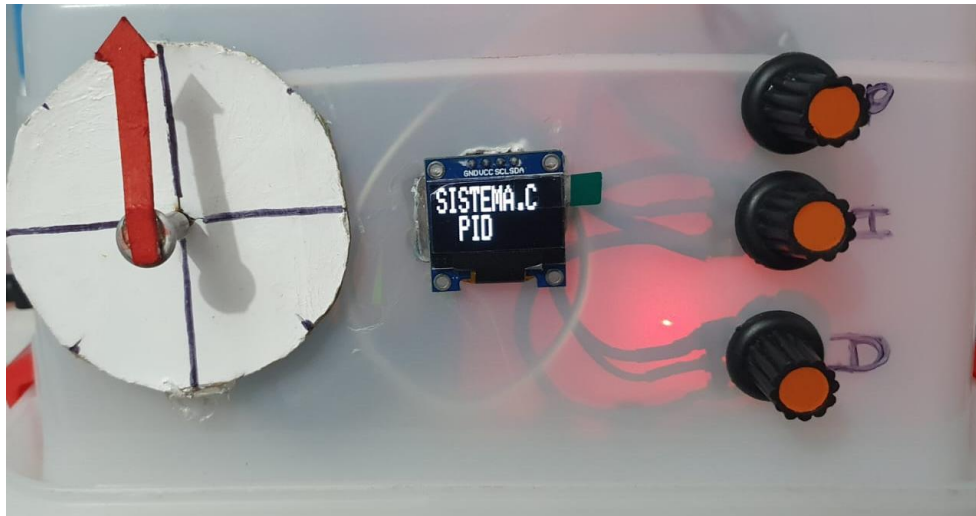
## Formando **líderes** para la construcción de un nuevo país en paz



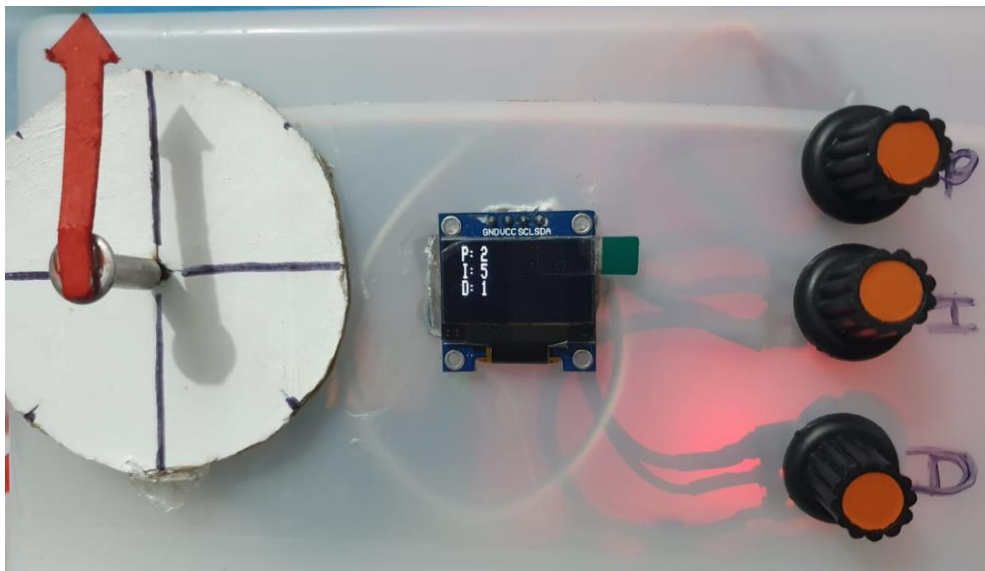
```
    }  
// asegúrese de que el bucle principal tenga una duración de <loopTimeMs> milisegundos  
void wait_enough_time() {  
    while(true) {  
        nowish = millis();  
        timeDelta = nowish - timeMarker;  
        if(timeDelta < loopTimeMs) {return;}  
        timeMarker = nowish;  
    }  
}  
  
void loop() {  
    wait_enough_time();  
    updatePIDValues();  
    calculateAccumulatedError();  
    calculateDeltaError();  
    generateMotorCommand();  
    updateMotor();  
    displayPermanentItems();  
}
```

*Ilustración 40 Programación Arduino del control de posición.*

Formando **líderes** para la construcción de un nuevo país en paz

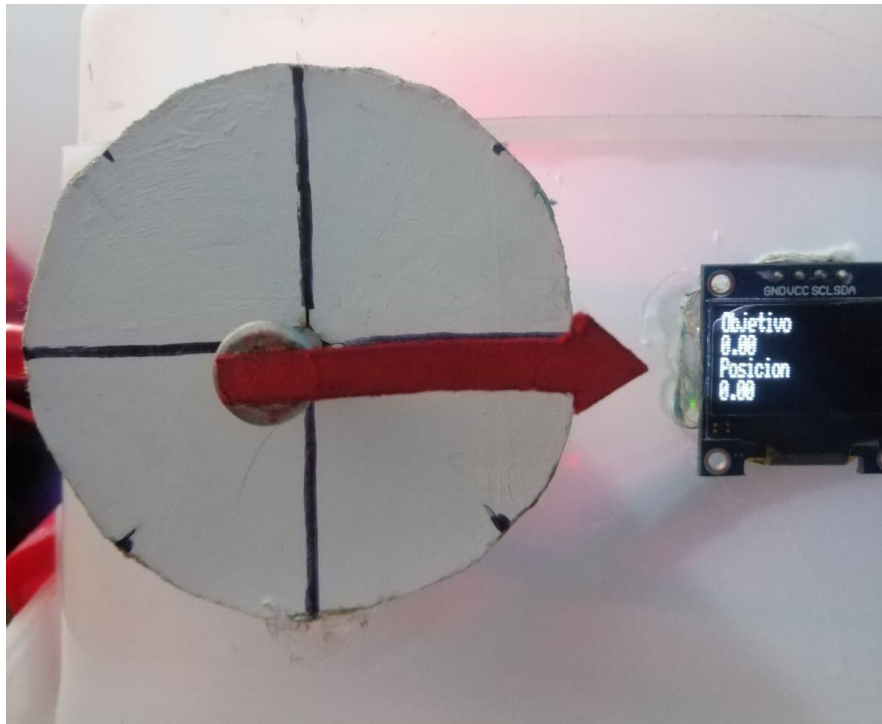


*Ilustración 41 Visualización de Inicio del sistema de Control.*



*Ilustración 42 Visualización de los parámetros del controlador PID.*

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



*Ilustración 43 Visualización del control de posición con pantalla Oled.*

### **Anexo 3. Programación de Arduino del controlador Velocidad.**

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



#### Control\_Velocidad\_PID \$

```
// SISTEMA CONTROL PID PARA UN MOTOR DC
// MONITOREADO DESDE LA COMPUTADORA Y LCD
// ECUACIONES DE DIFERENCIAS DE UN PID DISCRETO
//-----
// Voltaje de alimentacion: 6v-12v
// Velocidad 3800 RPM (6v) / 7800 RPM (12v)
// Consumo corriente: 40 mA
//-----
// Resolucion Encoder: 30 PPR(Pulsos por revolucion)
// Cuadratura "A" y "B"
int POT_sp = 1;
float sp;
int PWM_salida = 9;
float pv; // VARIABLE DEL PROCESO
int pinA= 3;
volatile int contador = 0;
unsigned long previousMillis = 0;
long interval = 10; //INTERVALO RPM CADA 100 MILI SEGUNDO
float cv;
float cv1; // cv(n-1)
float error; // variables de error
float error1; // error e(n-1)
float error2; // error e(n-2)
float kp = 1;
```

Formando **líderes** para la  
construcción de un nuevo  
**país en paz**



```
float ki = 7;    // Hace que el sistema responda mas rapido
float kd = 0.01;
float Tm = 0.1; // tiempo de muestreo
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x27, 16, 2);
void setup() {
  pinMode(pinA, INPUT);
  pinMode(PWM_salida, OUTPUT);
  Serial.begin(115200);
  attachInterrupt(1,interrupcion,RISING); //Flanco de subida Pin3
  Wire.begin();
  lcd.begin(16, 2);
  lcd.backlight();
}
void loop() {
  unsigned long currentMillis = millis();
  if ((currentMillis - previousMillis) >= interval)
  {
    previousMillis = currentMillis;
    pv = contador*(60.0/30.0); //RPM DEL EJE DEL MOTOR-----
    contador = 0;
  }
  // MOTOR MOVIMIENO A TRAVEZ DEL POTENCIOMETRO
```

## Formando **líderes** para la construcción de un nuevo país en paz



```

sp = analogRead(POT_sp)*(800.0/1023); //0-800 RPM MOTOR SE MODIFICA DEPENDIENDO EL MOTOR QUE SE USE
error = sp - pv; // error set point menos la variable del proceso
//-----ECUACION DE DIFERENCIAS-----
cv = cv1 + ( kp + kd/Tm)*error + (-kp + ki*Tm - 2*kd/Tm)*error1 + (kd/Tm)*error2;
cv1 = cv; // recursividad escalera
error2 = error1;
error1 = error;
//-----SATURAMOS LA SALIDA DEL PID-----
if (cv > 800.0){
    cv = 800.0;
}
if (cv < 10.0){ // Apartir de 10.0 el motor se empieza a mover
    cv = 10.0; // sino se pone 0
}
analogWrite(PWM_salida,cv*(255.0/800.0)); //0-255
// VER EN EL MONITOR SERIAL LO QUE SALGA
Serial.print("SP: ");
Serial.print(sp);
Serial.print(" PV: ");
Serial.println(pv);
delay(100);
// IMPRESION LCD
    lcd.setCursor(0, 0);
    lcd.print("SP(rpm): ");
    lcd.print(sp);
    lcd.setCursor(0, 1);
    lcd.print("PV(rpm): ");
    lcd.print(pv);
    delay(0.1);
}
void interrupcion(){
    contador++;
}

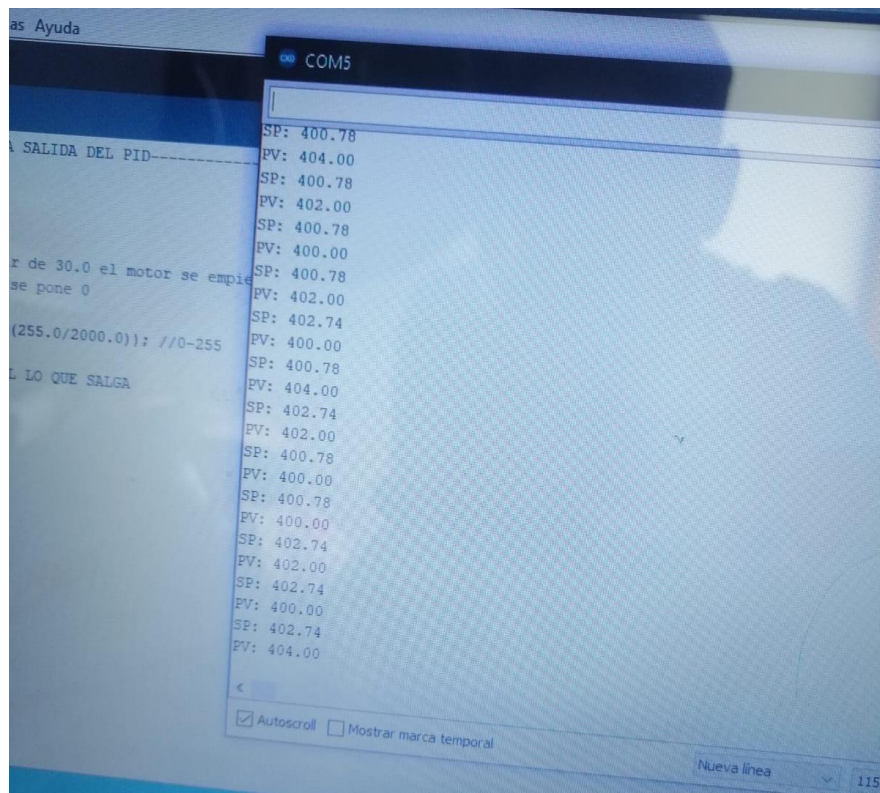
```

*Ilustración 44 Programación Arduino del control de velocidad.*

Formando **líderes** para la construcción de un nuevo país en paz



*Ilustración 45 Demostración Lcd del control de velocidad.*



*Ilustración 46 Visualización en el monitor serial del control de velocidad.*

Formando **líderes** para la construcción de un nuevo país en paz



#### Anexo 4. Diseño del prototipo utilizando el software SolidWorks.

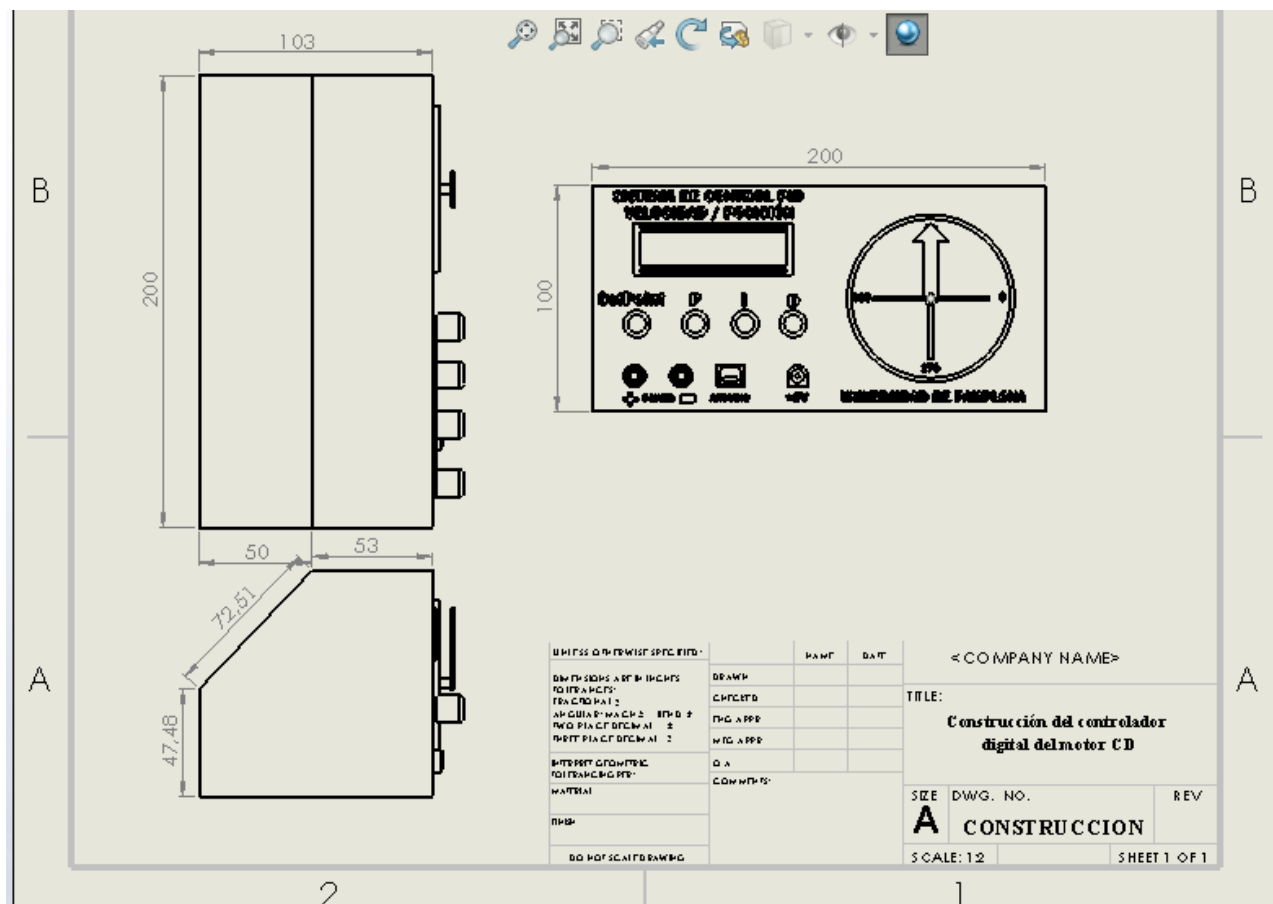


Ilustración 47 Diseño de construcción del sistema de control en milímetros.