

PARTE 1-PROYECTO FINAL: CONTROL DE VELOCIDAD ANGULAR SOBRE UN MOTOR DC Y ENCONDER DE POSICIÓN

1st Oscar Javier Macías Pizo
Universidad del Cauca
Popayán, Cauca
ojmacias@unicauca.edu.co

2nd Juan Sebastian Ijaji Zuniga
Universidad del Cauca
Popayán, Cauca
jsijaji@unicauca.edu.co

Abstract—Para este informe se tendrá en cuenta principalmente la investigación, estudio e implementación de un sistema de control de velocidad angular sobre un motor DC y encoder de posición, que en esta primera parte conllevará a la respectiva construcción del montaje, sus modelos matemáticos y su respectiva implementación en el programa Simulink

Index Terms—modelos, linealizar, montaje, velocidad, encoder, DC, Arduino

I. INTRODUCCIÓN

En esta primera parte del proyecto final de la asignatura se abordarán diferentes factores que se destacan por ser los pasos necesarios antes de llegar a la parte de control aplicada al sistema, por lo tanto se tratará como un prelude antes de llegar al que se podría considerar el tema prioritario para el trabajo final.

Dado lo anterior, para esta primera parte se seguirán algunos pasos que se basan en comenzar con la respectiva construcción del montaje del sistema en cuestión, el planteo de un modelo matemático para el sistema, representar el sistema mediante espacio de estados y función de transferencia, la implementación de un diagrama de bloques con el aplicativo Simulink y por ultimo, obtener la curva de respuesta real del sistema mediante la aplicación de una señal escalón de entrada al sistema real.

Es de mencionar que para esta primera parte del proyecto final la adquisición de datos se realizará sobre el aplicativo Simulink y el microprocesador Arduino tomará la función de puente de comunicación entre el mismo aplicativo Simulink controlado por el computador y la misma tarjeta Arduino ya mencionada descrita como el hardware del sistema.

Asimismo, si se indaga más en la parte del aplicativo Simulink se destaca que su función para esta primera parte del proyecto final se basa en la creación de un diagrama de bloques que permite dar con la identificación de los datos obtenidos por el sensor utilizado dentro del montaje del sistema (explicado más adelante) evidenciando de igual manera las gráficas generadas por este mismo.

Igualmente, es de aclarar que se siguieron todas las indicaciones ofrecidas por el ingeniero a cargo de la asignatura y se siguieron algunas fuentes externas para obtener más

información respecto al tema a estudiar para esta primera parte del proyecto final

II. OBJETIVOS

A. OBJETIVO GENERAL

Construir un sistema que permita el control de velocidad angular sobre un motor DC y encoder de posición a escala y asimismo, sobre el cual sea posible implementar un controlador embebido en microprocesador este caso será Arduino.

B. OBJETIVOS ESPECIFICOS

- Obtener un modelo matematico para el sistema dinamico en cuestión
- Representar el sistema mediante espacio de estados
- Representar el sistema mediante función de transferencia
- Implementar en Simulink un diagrama de bloques que permita recibir y graficar los datos leídos por el sensor de la planta, asimismo incluir una serie de bloques de Simulink que permita establecer valores a la variable de entrada del sistema elegido. Lo anterior mediante comunicación serial entre Matlab/Simulink y el microprocesador Arduino
- Obtener la curva de respuesta real del sistema elegido mediante la aplicación de una señal escalón de entrada al sistema real

III. MONTAJE DE LA PLANTA

Para poder realizar el respectivo montaje del sistema que permita el control de velocidad angular sobre un motor DC y encoder de posición es necesario hacer uso de los siguientes materiales que permitan la respectiva construcción del sistema de control en cuestión

- Placa de pruebas (Protoboard)
- Arduino Nano
- Puente H L298n
- Sensor Encoder de velocidad con herradura
- Motor DC
- Modulo de expansión de Arduino

Asimismo, se sigue unos respectivos pasos que permitan dar con la conexión de cada uno de los componentes necesarios

para cumplir con la construcción del sistema control escogido para el proyecto final. Primero se realizan las diferentes conexiones entre el microprocesador Arduino con el respectivo sensor de velocidad y el puente H L298n. Igualmente es de aclarar que se tomará el modulo de expansión de Arduino para que las conexiones sean más organizadas y fáciles de manipular Por lo tanto se tomarán los pines del Arduino Nano escogidos por el grupo de trabajo y se conectarán debidamente tanto al sensor como al puente H de la siguiente manera

- El pin D7 del Arduino Nano irá conectado al pin INT1 del puente H L298n
- El pin D6 del Arduino Nano irá conectado al pin INT2 del puente H L298n
- El pin D5 del Arduino Nano irá conectado al pin Enable del puente H L298n
- El pin D4 del Arduino irá conectado a una de las dos salidas del sensor de velocidad, para este caso D0

Con las conexiones de cada uno de los pines del Arduino tanto al sensor como al puente H L298n, ahora se mostrará como irán conectados tanto los pines de fuente de voltaje del Arduino como el pin de tierra GND

- La fuente de 5 voltios del Arduino Nano irá conectado a la entrada del sensor de velocidad
- Por ultimo, se conectará la tierra GND del Arduino Nano a la misma tierra tanto del sensor de velocidad como del puente H L298n

Por ultimo, es de destacar que la única conexión que no estará directamente conectada al Arduino será el motor DC que fue conectado a la entrada A del puente H L298n como es posible observar en la imagen 1

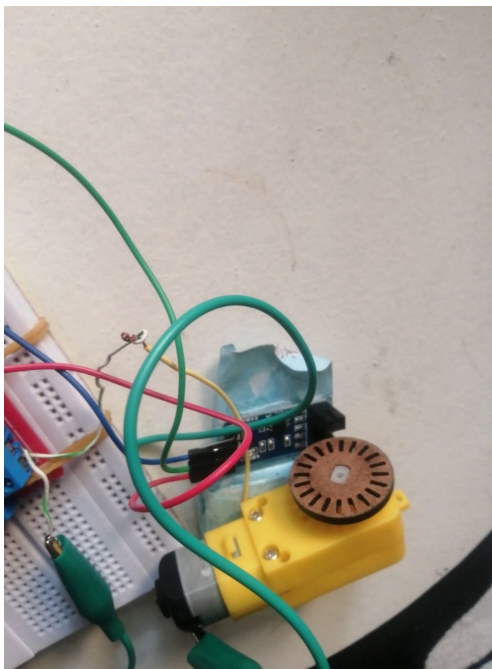


Fig. 1. Motor DC

Por lo tanto el montaje completo del sistema de control en cuestión estaría dado por la siguiente imagen

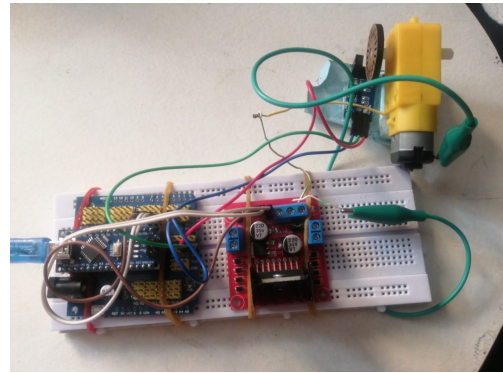


Fig. 2. Montaje de planta completo

IV. MODELADO MATEMÁTICO

Se tiene en cuenta que para conseguir el modelado matemático del sistema que permite el control de velocidad angular sobre un motor DC y encoder de posición, es necesario seguir el proceso tradicional para encontrar el modelado de este mismo sistema haciendo uso de las leyes de la física. Dado la anterior, es necesario destacar que esta clase de sistema se divide en dos partes, debido a que el modelo en cuestión rige una aparte eléctrica dado que el motor DC es aquel que va a aportar movimiento rotatorio producido en este caso por un voltaje y por otro lado, el modelo también se compone por una parte mecánica debido a la rotación que tomará el motor, teniendo en cuenta sus respectivos valores dependiendo de las leyes de la física.

A. Ecuación eléctrica del motor

Como se dijo anteriormente el motor DC aportará movimiento rotatorio producido por una fuente de voltaje, por lo tanto, este modelo matemático estaría dado por la inductancia del motor L , la resistencia del motor R , la corriente eléctrica producida por el motor $I(t)$, la fuerza contraelectromotriz que aumentará proporcionalmente a la velocidad angular del motor $emf(t)$ y por ultimo, el voltaje de entrada $v(t)$. Siguiendo las leyes de la física el modelo para el funcionamiento eléctrico del sistema estaría dado por

$$v(t) = LI'(t) + RI(t) + emf(t) \quad (1)$$

B. Ecuación mecánica del motor

Ahora, como se dijo anteriormente el sistema en cuestión también tiene un modelado mecánico que representará mediante las leyes de la física la rotación del motor. Por lo tanto, el respectivo modelado para esta parte mecánica estaría dada por por el momento de fuerza que ejercer el motor dado por $T(t)$, J que será su respectiva inercia, B que será la constante de fricción viscosa y por ultimo tanto la velocidad angular como la aceleración angular que estarían expresadas por

$$\theta(t) \quad (2)$$

$$\ddot{\theta}(t) \quad (3)$$

Por lo tanto, siguiendo el procedimiento tradicional y teniendo en cuenta las leyes de la física, el modelado para el sistema en su componente mecánico quedaría de la siguiente manera

$$T(t) = J\ddot{\theta}(t) + B\dot{\theta}(t) \quad (4)$$

C. Ecuaciones de acople

Ahora, para poder encontrar la relación entre los modelados mecánico y eléctrico es necesario proponer dos ecuaciones de acople que permitan conectar de manera directa la fuerza contraelectromotriz $emf(t)$ y la velocidad angular, y asimismo, el momento de fuerza del motor con la corriente del motor, ambos términos ya explicados anteriormente. Teniendo en cuenta lo anterior, se propone una constante contraelectromotriz K_e que esté relacionada con la respectiva fuerza de esta misma índole y la velocidad angular. Asimismo, también se aplicará una constante K_i que represente la constante de torque del motor y esté relacionada con el respectiva fuerza de torsión del motor y su corriente. Obteniendo las siguientes ecuaciones de acople

$$emf(t) = K_e\dot{\theta}(t) \quad (5)$$

$$T(t) = K_i I(t) \quad (6)$$

Ahora se reemplazan estas mismas ecuaciones de acople en cada uno de los modelados, tanto eléctrico como mecánico

$$v(t) = LI(t) + RI(t) + K_e\dot{\theta}(t) \quad (7)$$

$$k_i I(t) = J\ddot{\theta}(t) + B\dot{\theta}(t) \quad (8)$$

V. REPRESENTACIÓN EN ESPACIO DE ESTADOS

Con las ecuaciones (7) y (8) obtenidas es posible obtener la representación del sistema en espacio de estados. Para realizar este proceso se encuentran la X_n que será nuevas variables de estado y serán igualadas con su respectiva variable a representar de las ecuaciones diferenciales ya mostradas anteriormente. Por lo tanto las nuevas variables X_n quedarían de la siguiente manera

$$x_1 = \theta(t) \quad (9)$$

$$\dot{x}_1 = \dot{\theta}(t) \quad (10)$$

$$x_2 = I(t) \quad (11)$$

$$\dot{x}_2 = \dot{I}(t) \quad (12)$$

Ahora se reemplazará estas mismas variables X_n en las ecuaciones (7) y (8)

$$v(t) = L\dot{x}_2(t) + Rx_2 + K_ex_1 \quad (13)$$

$$K_ix_2 = J\dot{x}_1 + Bx_1 \quad (14)$$

Ahora ya es posible representar mediante espacio de estados el sistema en cuestión

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{B}{J} & \frac{K_i}{J} \\ -\frac{K_e}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v(t) \quad (15)$$

VI. REPRESENTACIÓN EN FUNCIÓN DE TRANSFERENCIA

El principal objetivo para poder encontrar la representación en función de transferencia del sistema es encontrar la relación entre la velocidad angular y el voltaje de entrada del sistema ya explicados anteriormente, por lo tanto, se sigue el debido proceso y se deberá aplicar la transformada de Laplace al respectivo sistema expresado en ecuaciones diferenciales, obteniendo de tal manera las siguientes ecuaciones tanto para el modelado eléctrico como mecánico

$$v(s) = LSI(s) + RI(s) + k_e S\theta(s) \quad (16)$$

$$k_i I(s) = Js^2\theta(s) + BS\theta(s) \quad (17)$$

Ahora se despeja la respectiva salida del sistema, que para este caso sería $I(s)$ de la ecuación (17)

$$I(s) = \frac{JS^2\theta(s) + BS\theta(s)}{K_i} \quad (18)$$

Ahora se reemplazará en la ecuación (16) el resultado de $I(s)$ factorizando este mismo valor de salida como se muestra a continuación

$$v(s) = I(s)(LS + R) + K_e S\theta(s) \quad (19)$$

$$v(s) = \frac{(JS^2\theta(s) + BS\theta(s))(LS + R)}{K_i} + K_e S\theta(s) \quad (20)$$

Por ultimo, se muestra la función de transferencia despejando la velocidad angular y el voltaje de entrada del sistema de la siguiente

$$\frac{S\theta(s)}{v(s)} = \frac{K_i}{JLS^2 + (BL + JR)S + BR + K_i K_e} \quad (21)$$

VII. CONTROL

Para realizar el control del motor utilizamos el controlador PID el cual tiene como objetivo mantener la velocidad del controlador igual a la de la referencia. Este controlador ajusta la salida en función de tres términos, termino de proporcionalidad, integral y derivativa, esto para minimizar el error y mejorar la estabilidad del sistema. Para implementar este controlador en nuestro proyecto utilizamos el método heurístico o normal, hallando las constantes de los términos para que el control funcione de la mejor forma posible.

Para poder lograr el objetivo que tiene el controlador PID y encontrar las constantes de los términos ya mencionados se realizaron varias pruebas que permitieran encontrar los valores ideales tanto para la parte proporcional, integral y derivativa. Por lo tanto, primero que todo se le dio como valor igual cero a cada una de las constantes, empezando con la variación de la constante proporcional, aumentando su respectivo valor hasta 1.9 donde obtuvimos el mejor resultado, posteriormente procedimos a variar los valores de las constante derivada e integral siguiendo este mismo método, encontrando como valores 1.1 y 1.155 respectivamente, lo que permitió obtener valores que estaban muy acercados al valor de referencia de propuesto.

VIII. CÓDIGO

Para la implementación del control utilizamos el siguiente código

A. Declaración de pines

```
#include <util/atomic.h>
#define Flanco 2
#define PWM 5
#define IN2 6
#define IN1 7
```

Fig. 3. Declaración de pines

B. Declaración de variables

```
//Declaración de variables
int cont=0;
volatile float vel=0;
int rev=0;
long tAnt=0;
long t1=0;
float eprev=0;
float eintegral=0;
```

Fig. 4. Declaración de variables

C. Declaración de pines del Arduino

```
void setup() {
  //Declaración de pines del arduino
  Serial.begin(9600);
  pinMode(Flanco, INPUT);
  pinMode(PWM, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN1, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(Flanco), tiempoFlancos, RISING);
}
```

Fig. 5. Pines Arduino

D. Voltaje de referencia, constantes proporcional, derivativa e integral

```
void loop() {
  // Voltaje de referencia
  //constantes proporcional, derivativa e integral
  float referencia=130;
  float kp=1.9;
  float kd=1.1;
  float ki=1.155;
  float RPM=0;
  // Se utiliza la funcion Atomic block para que el codigo se ejecute sin interrupciones
  ATOMIC_BLOCK(ATOMIC_RESTORESTATE){
    RPM=vel*60;
```

Fig. 6. Constantes proporcional, derivativa e integral

E. Calculo de error kp, kd, ki

```
long t2=micros();
float deltaT=((float)(t2-t1))/1.0e6;
t1=t2;
float e=referencia-RPM;
//derivativo
float dedt=(e-eprev)/deltaT;
//integral
eintegral=eintegral+e*deltaT;
//CONTROL
float u=kp*e+kd*dedt+ki*eintegral;
```

Fig. 7. Error kp, kd, ki

F. RPM maximas y minimas

```
if(pwr>200){
  pwr=200;
}

if(pwr<100){
  pwr=100;
}
```

Fig. 8. RPM maximas y minimas

G. Tiempo de flancos

```
void tiempoFlancos() {
  int b=digitalRead(Flanco);
  if(b==HIGH){
    long tActual=micros();
    cont++;
    if(cont>=40){
      rev++;
      cont=0;
      float dT=((float)(tActual-tAnt))/1.0e6;
      tAnt=tActual;
      vel=1/dT;
    }
  }
}
```

Fig. 9. Tiempo de flancos

H. Controlador del motor

```
analogWrite(pwm, valorPWM);
if (dir==1){
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
}
else if (dir==-1){
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
}
else{
  digitalWrite(in1,LOW);
  digitalWrite(in2,LOW);
}
}
```

Fig. 10. Controlador de motor

IX. RESULTADOS

Como se puede ver en la siguiente imagen comparamos los resultados de las RPM de referencia con los que el motor nos muestra, apreciando que la variación del motor es proporcional a la de la referencia con un error aceptable debido al tipo de motor y el sensor

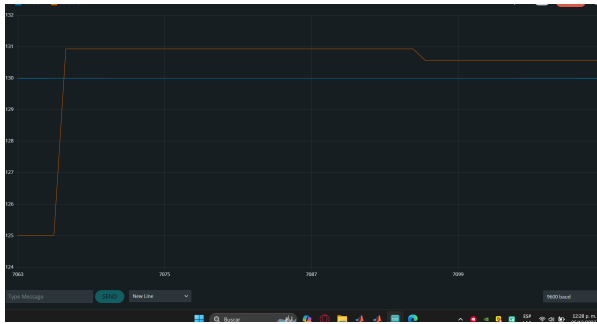


Fig. 11. Controlador de motor

CONCLUSIONES

- Un sistema dinámico que utiliza un motor DC es considerado como un modelo electromecánico debido a que cuenta con dos modelos para dos campos diferentes, que en este caso son el eléctrico y el mecánico
- El programa Simulink es utilizado para poder realizar diagramas de bloques que pueden ser utilizados en sistemas montados de manera física, evidenciando que mediante este mismo es posible encontrar diferentes valores necesarios para el buen desarrollo del proyecto en cuestión como lo fue por ejemplo, encontrar el valor en RPM para el encoder del motor
- Es posible encontrar la representación tanto de espacio de estados como función de transferencia a un modelado matemático para un sistema que está compuesto por dos distintas ramas de la física, en este caso como lo fueron la electrónica y la mecánica
- El controlador PID es realmente eficiente debido a que aunque usáramos instrumentos realmente básico este cumpliría su función de manera aceptable
- Las revoluciones se encuentran limitadas por un intervalo debido a que si estas se salen del intervalo, el error es demasiado grande y el resultado no es el esperado
- El controlador permite que aunque se interfiera en el movimiento del motor este genere más voltaje para que las revoluciones sean las de referencia
- El método heurístico nos permite hallar las constantes que nos permitirán el correcto funcionamiento del controlador de manera fácil y eficiente, obteniendo un resultados satisfactorio

REFERENCIAS

- 1 Javier Ramirez Perez de Inestrosa. Introducción a Matlab y Simulink
- 2 Angel Valera Fernandez. Modelado y control en el espacio de estados
- 3 Eduardo Zurek Valera Soni Vadalá. Aplicación de Matlab para el análisis de sistemas LTI
- 4 Universidad Nacional de Tucuman. Facultad de ciencias exactas y tecnología. Departamento de Ingeniería mecánica. Función de transferencia