Controlador PID de un Motor/Generador DC.

Cesar Patricio Vizhñay Tuza

cvizhnayt@est.ups.edu.ec

Edgar Vladimir Calle García.

ecalleg@est.ups.edu.ec

Universidad Politécnica Salesiana.

Cuenca-Ecuador

Resumen— En este documento se aborda la implementación de un controlador PID para la gestión de un motor DC, utilizando los softwares Matlab y LabVIEW, junto con el dispositivo Arduino. Inicialmente, se presenta una breve descripción de los componentes necesarios. Posteriormente, se deriva la ecuación PID y se diseña el controlador. Finalmente, se detallan la implementación del sistema y la verificación de los resultados obtenidos.

Palabras clave: Controlador PID, Motor DC, Matlab, Arduino.

Abstract— This document addresses the implementation of a PID controller for the management of a DC motor, using the Matlab and LabVIEW software, together with the Arduino device. Initially, a brief description of the necessary components is presented. Subsequently, the PID equation is derived and the controller is designed. Finally, the implementation of the system and the verification of the results obtained are detailed.

Keywords: PID Controller, DC Motor, Matlab, Arduino.

INTRODUCCIÓN

Se presentará la respuesta de un motor de corriente continua al implementarle un controlador PID digital diseñado en el software LabVIEW. Para ello, se realizará una explicación teórica sobre controladores y el comportamiento de los motores de corriente continua. Además, se describirá el proceso de adquisición de datos, que consiste en tomar muestras del mundo real, convertirlas en tensiones eléctricas y digitalizarlas para ser procesadas por un ordenador. Cabe resaltar que para este proyecto se utilizaron dos softwares principales: LabVIEW y Matlab.

MARCO TEORICO

A. *Principios Básicos de un Motor de Corriente Continua.*

Cuando un conductor de longitud LLL se mueve dentro de un campo magnético de inducción BBB, cortando las líneas del campo, se genera en él una fuerza electromotriz o contra-electromotriz e, [1] de tal manera que:

$$\boldsymbol{e = k * n * \emptyset}$$

Donde:

$\boldsymbol{n = n^{\circ}\ de\ revoliciones\ por\ minuto}$.

$\boldsymbol{k = cte.que\ depende\ de\ la\ maquina}$

$\boldsymbol{\theta = flujo\ magnetico}$

Cuando un conductor de longitud L, a través del cual fluye una corriente I, se encuentra dentro de un campo magnético, se ejerce una fuerza sobre dicho conductor, la cual es:

$$\boldsymbol{F = U * L * B}$$

Como se muestra en la figura 1, una máquina de corriente continua tiene un devanado inducido que consiste en una espira, mientras que el campo magnético es generado por el devanado inductor con un par de polos. Si se aplica una tensión continua al devanado inducido, circulará una corriente continua a través de él, generando una fuerza que hará girar el motor. Por otro lado, si se aplica un movimiento de rotación al devanado inducido con una velocidad angular, se formará una fuerza electromotriz en los bornes de las espiras, la cual será rectificada y extraída al exterior de la máquina a través de las escobillas, funcionando, así como un generador. [1]

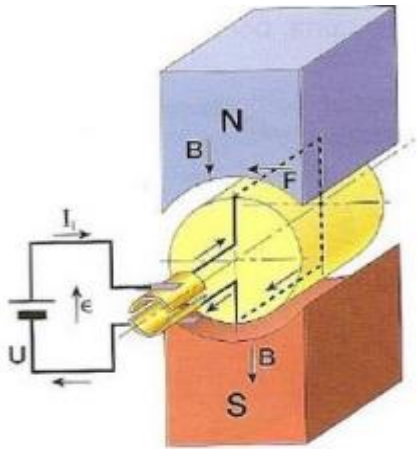
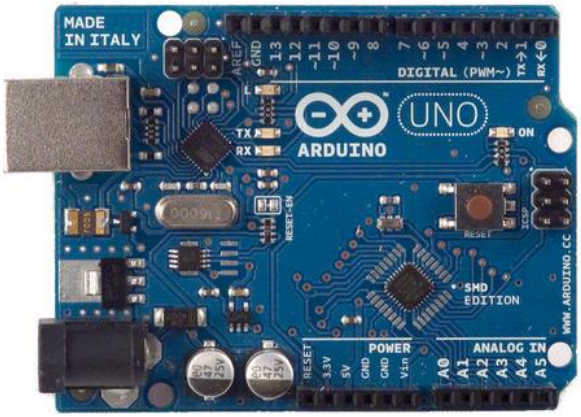


Figura 1: Principio de funcionamiento de un motor generador de corriente continua. [1]

B. ***Arduino:*** Es una plataforma de electrónica «open-source» o de código abierto cuyos principios son contar con software y hardware fáciles de usar. Básicamente lo que permite esta herramienta es la generación de infinidad de tipos de microordenadores de una sola placa, que luego pueden tener una amplia variedad de usos según la necesidad de la persona que lo cree.



C. ***Transistor TIP41C:*** Convertidor, Reductor de voltaje 12VDC - 9VDC. Circuito que permite obtener 9 VDC reduciendo los 12V o más que da la batería de un auto. [2]

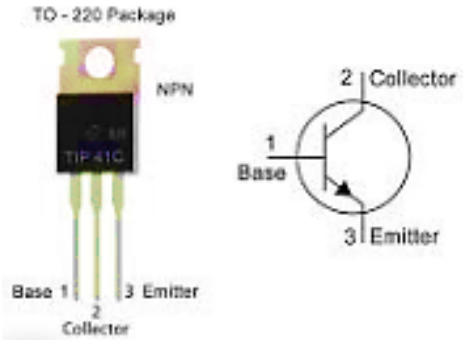


Figura 2: Arduino.

D. ***Teoría del Motor:*** Las Máquinas DC son generadores que convierten energía mecánica en energía eléctrica DC y motores que convierten energía eléctrica DC en energía mecánica. Las máquinas DC tienen una salida DC solo porque existen un mecanismo que convierte los voltajes AC internos en voltajes DC en sus terminales.

Figura 3: Stephen Chapman S.J., Máquinas Eléctricas.

Considere el motor CD controlado por el voltaje de armadura, el diagrama de bloques donde I es inercia del motor, C es el coeficiente de fricción, Ra y La son la resistencia de la inductancia de armadura, $K_T = K_b$ son las constantes del motor.

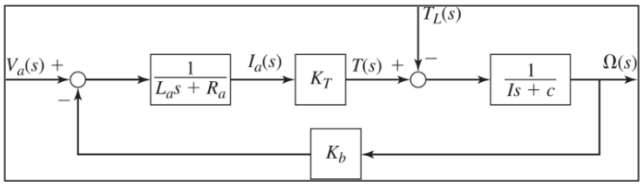


Figura 4: Diagrama de bloque.

Se muestra el esquema del motor donde el generador hace las veces de un sensor de velocidad con resolución lineal de 7V por 1000 rpm. El eje del motor tiene acoplado en su eje un sistema de engranajes hacia la carga y otro hacia el generador. Es importante conocer la relación de engranajes, donde $\omega_1 \, y \, \omega_2$ son la velocidad, $r_1 \, y \, r_2$ son los radios de los engranajes y $n_1 \, y \, n_2$ son los números de dientes.

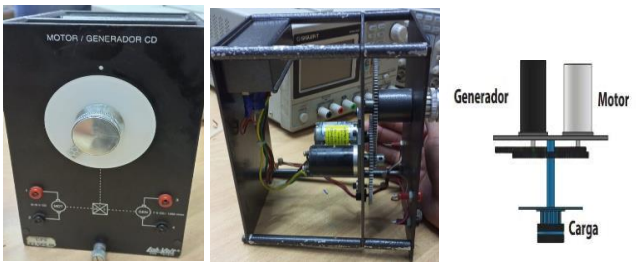


Figura 5: Vista frontal, lateral y superior del motor/ generador.

E. ***PID:*** Es un controlador o regulador, es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. Este compuesto por la acción Proporcional, Integral y Derivativa.

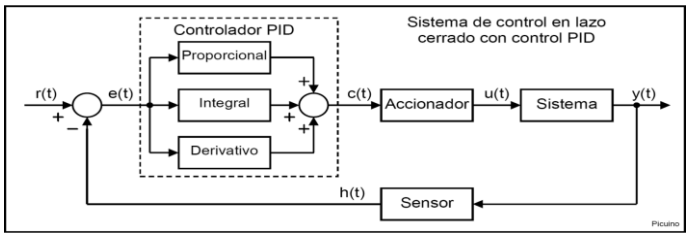


Figura 6: Sistema de control en lazo cerrado con control PID

F. ***LabVIEW***: Labview es un desarrollo de programación interactivo y un sistema de ejecución diseñado para personas como científicos o ingenieros que sin ser informáticos necesitan programar como parte de su trabajo. El ambiente de desarrollo de Labview trabaja sobre computadoras Windows, Mac OS X, o Linux, además se pueden crear programas que corren en una variedad de plataformas embebidas como FPGAs (Field Programable Gate Arrays), DSPs (Digital Signal Processors) y microprocesadores.

Figura 7: LabView

f. ***PID Tuner:*** La app PID Tuner ajusta automáticamente las ganancias de un controlador PID para una planta SISO con el fin de lograr un equilibrio entre el rendimiento y la robustez. Puede especificar el tipo de controlador, como controladores PI, PID con filtro derivativo o PID con dos grados de libertad (2-DOF).(4)

DESARROLLO

Mediante LabVIEW precedemos a implementar y a declaras los valores para la interfaz ente el Arduino y motor, con la finalidad de obtener datos para un posterior análisis.

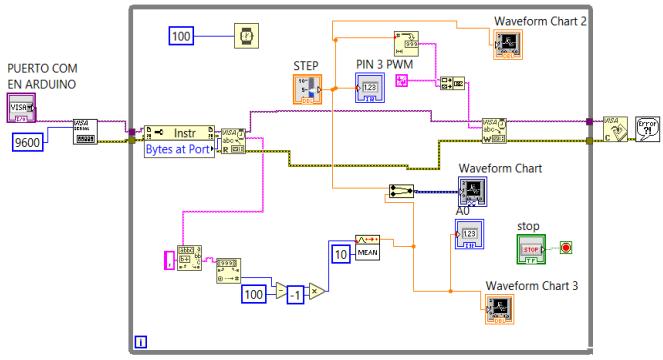


Figura 8: Programación en LabVIEW

Para la obtención de la curva se procede a determinar mediante LabVIEW el cual se puede observar en la primera imagen los tres valores, que en este caso son el escalón, la curva característica y el tiempo de muestreo. Mientras que en la imagen 2 se puede visualizar solo el escalón con el tiempo de muestreo y finalmente en la figura 3 se puede observar la curva característica con el tiempo de maestreó.

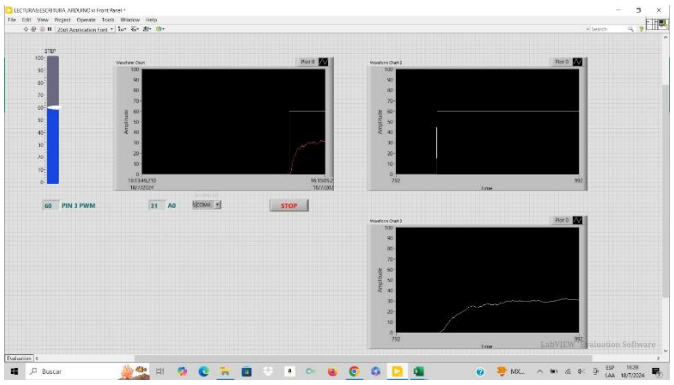


Figura 9: Respuesta al escalón

Para ello se ara clic en el apartado **plot** donde encuentra un aportado para exportas los datos a Excel, obteniendo a si los datos representados a continuación.

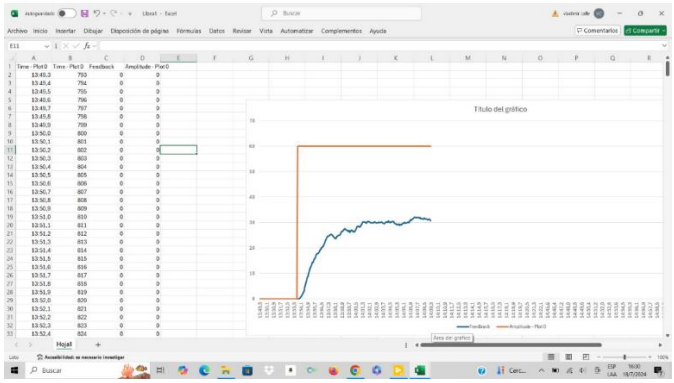


Figura 10: Tabla de datos

Una vez obtenido estos datos se procede a importa el Excel, al **Command Window** de Matlapa, tal y como se indica en la siguiente figura.

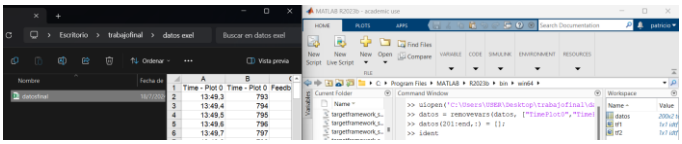


Figura 11: Exportación de datos a Matlap

Por consiguiente, se exporta los datos al mediante el comando **ident,** con la finalidad de realizar iteraciones de primer, segundo orden y así sucesivamente hasta encontrar un el porcentaje más alto.

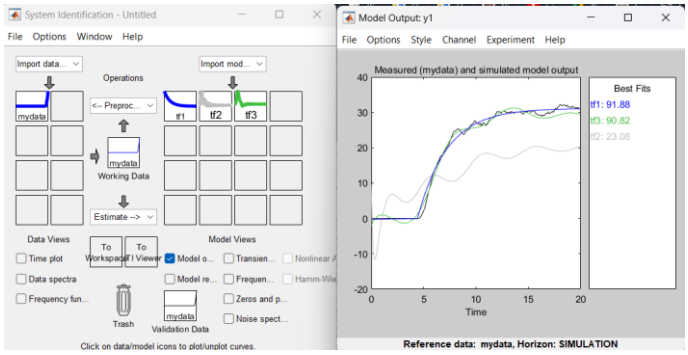


Figura 12: Iteracción de funciones.

De esta forma siendo el más alto la ecuación de primer orden con un **91.88%**

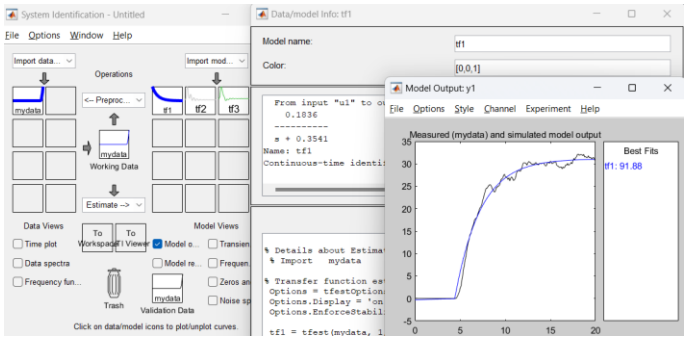


Figura 13: Ecuación de primer orden

A continuación, se procede a exporta la función (tf1) al **Workspace**. Una vez obtenido la fusión de transferencia o planta, se exporta mediante el comando **pidTuner,** donde se selecciona en el apartado **Type-PID.** Y mediante la calibración en **Response Time** y **Transiente Behavior** se encuentra los datos: **kp=0.50482, ki=0.04239 y kd=1.5** [2]

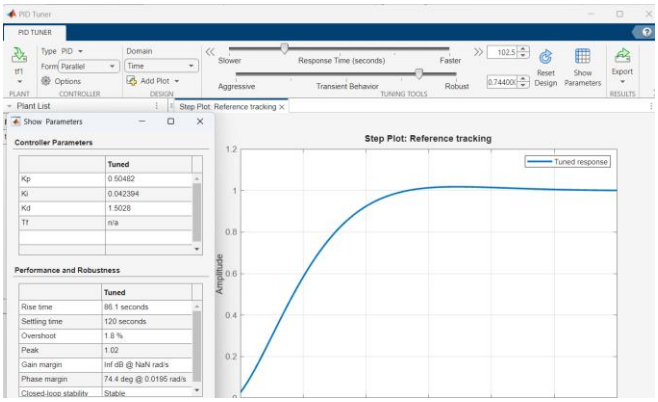


Figura 13: Obtención de kp, ki y kd

Después de verificar el funcionamiento del controlador en la simulación, procedemos a implementar el control PID en lazo cerrado en LabVIEW. Finalmente, se integra el motor y se realizan las mediciones correspondientes.

De esta forma obtenemos la el siguiente controlador:

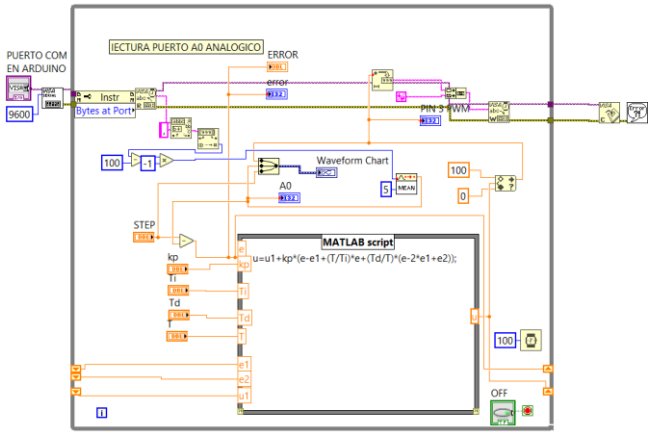


Figura 14: Programación de la planta en lazo cerrado con controlador PID en Labview

Resultados.

Por consiguiente, se obtiene los siguientes resultados. En primera instancia se visualiza la panta en cero, cabe recalcar que el tiempo de muestre ya carre.

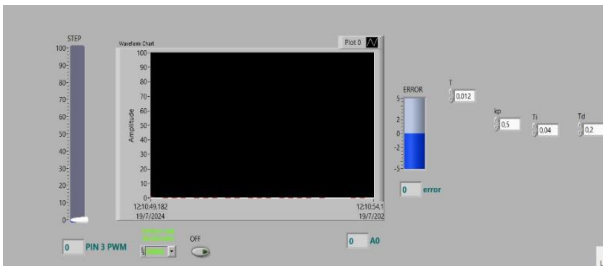


Figura 15: Encendido del controlador.

Cave recalcar que la línea de color verde es el escalo, mientras que la línea roja es la curca característica y finalmente la línea blanca indica el voltaje de referencia.

a) Estabulación al escalón

Al dar inicio al escalón con un 70% se visualiza como la curva se estabiliza, aunque el tiempo de estabilización es lento, logra estabilizarse.

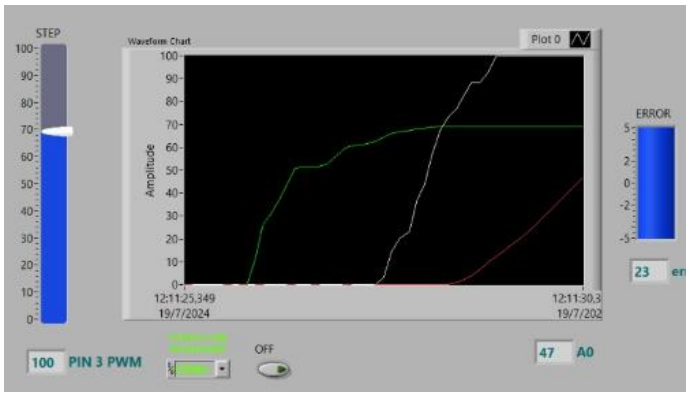


Figura 16: Escalón al 70%

b) Perturbación

Ya que las capturas fueron tomadas de forma continua, se visualiza la estabilización al escalón y una vez estable se procede a realizar la perturbación la cual de la misma forma precede a estabilizarse tal y como lo indica las presentes figuras.

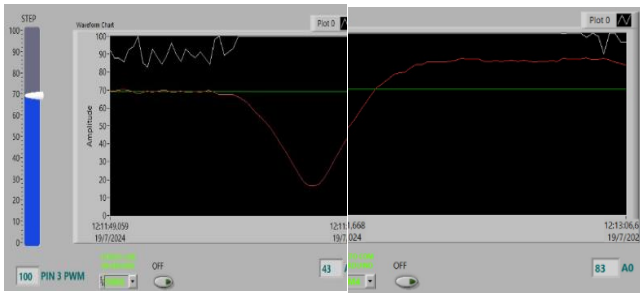


Figura 17: Respuesta al escalón y perturbación.

Finalmente, una vez establecido la perturbación se procede a dar un escalón del 30% y al 50%, obteniendo en los dos casos la estabilización.

CONCLUSIONES.

- Para implementar el controlador PID digital en lazo cerrado, fue necesario construir tanto el circuito físico como el circuito diseñado en software. Estos circuitos deben interactuar con el motor, que actúa como la planta del sistema.
- Al encontrar la respuesta al escalón unitario de la planta (motor), es importante considerar que puede haber un margen de error al tomar los datos de la gráfica. Por lo tanto, es necesario ajustar las variables del controlador PID al implementar el circuito final de control.
- Concluimos que el software LabVIEW es muy versátil y eficiente, debido a la abundancia de información, tutoriales y el soporte proporcionado por la inteligencia artificial.

ANEXOS

```
const int sesnorPin=A0;   // feedback o generador electrico
const int controlPin=3;  // pwm tipo1 para amplificar
int sensorValue=0;       // inicializa en cero
int pv=0;                // en cero
int cv=0;                // en cero

/// comunicacion serial

String inputString ="";        // lectura feedback para enviar por el puerto serial
bool stringComplete=false;     // si recibimos o no datos enviamos un true y un false para no

unsigned long sampleTime =100, lastTime=0;       // tiempo de muestreo 100ms



void setup() {                 // metodo para inicializar las configutaciones como entradas o salidas y velocidad de transmision
  // put your setup code here, to run once:
Serial.begin (9600);
}

void loop() {              // metodo de ejecucion

  // put your main code here, to run repeatedly:
if((millis()-lastTime)>sampleTime||lastTime==0)
{
lastTime =millis();

sensorValue=analogRead(sesnorPin);   // lwctura del feedback
pv=map(sensorValue,0,1023,100,0);    // conversion de 1024 a 100 feedback
Serial.print(',');                   // delimitador
Serial.print(pv);                    // valores del feedback de 0-100
}
```

```
//  sentencia  0 - 255 pwm salida para el motor dc

if(stringComplete)
{
cv=inputString.toInt();
cv=map(cv,0,100,0,255);    // yo recibo 0 y la salida es 0pwm si recibo 100 la salida es de 255pwm
analogWrite(controlPin,cv);


inputString ="";    // espacio en blanco
stringComplete =false;



}



}

// comunicacion serial  lectura
void serialEvent(){
  while(Serial.available()){
    char inChar =(char)Serial.read();
    inputString+=inChar;
    if(inChar=='\n'){
    stringComplete=true;
    }
}

}
```

Figura 18: Código del Arduino

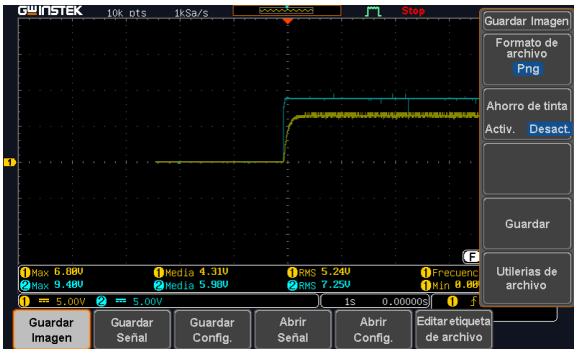


Figura 19: grafica obtenida por el osciloscopio

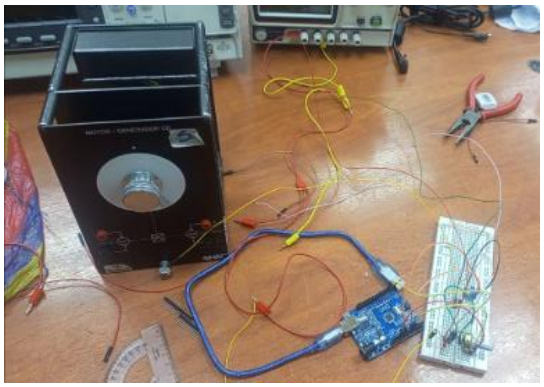


Figura 20: Primer implementación fallida del controlador PID

BIBLIOGRAFIA

Bibliografía

[1] P. Abril, «Motores Eléctricos de Corriente Continua,» 2024.

[2] Electrónica Unicrom, «Pinterest,» [En línea]. Available: https://www.pinterest.com/pin/664210645022716570/. [Último acceso: 19 Julio 2024].

[3]

[4] J. Correa, «YouTuve,» 28 Octubre 2020. [En línea]. Available: https://www.youtube.com/watch?v=yIQzwpZhyQM&t=1840s. [Último acceso: 19 Julio 2024].

[5] C. Dordt, Sistemas de control moderno, vol. 10, Martin Romo, 2005, p. 928.