UADE

Universidad Argentina de la Empresa

Facultad de Ingeniería y Ciencias Exactas

Trabajo práctico Obligatorio

Modelado y construcción de un sistema de control

Autores:
Aguirre, Emanuel
Fernandez, Gonzalo
Portugal Rios, Julian
Russo, Pablo

2021

Contenidos

1 Introducción

Como soporte para el trabajo final de ingeniería ”REINGENIERÍA DE UN SISTEMA DE RESCATE DE PERSONAS EN AMBIENTES ACUÁTICOS PARA EMBARCACIONES (Aguirre, Russo)”, se diseñará un sistema capaz de ser utilizado en el ensayo de las turbinas diseñadas y construidas como parte del mencionado trabajo.

El algoritmo de control será responsable de recibir un valor de velocidad angular (ω_0 [RPM]) almacenado en memoria del MCU, comenzará el giro del motor desde un duty cicle del 0% para la señal de control modulada por ancho de pulsos, acelerando el motor hasta la velocidad fijada. Luego de un cierto tiempo t se actualizará la velocidad objetivo, y el sistema deberá actualizar el ciclo de trabajo hasta un cierto valor.

Mediante el uso de los conocimientos adquiridos en el curso de Instrumentación y Control, se pretende:

- Pre-fijar en el sistema de control una serie de valores de velocidad angular en RPM.
- Implementar la medición del parámetro: Velocidad Angular Medida [ω_m]
-

2 Sistema

Se dispone de un motor 775-24v, de escobillas, el cual posee un rotor, que será nuestra planta y su velocidad la variable a controlar. A continuación se mostrará el motor, con sus medidas respectivas:

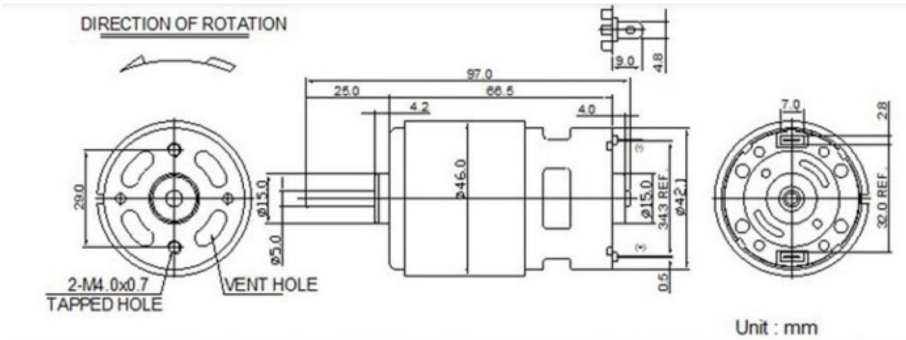

Figure 1: Esquema del motor CC

Se dibujo en SolidWorks un rotor similar al original para calcular el peso y el momento de inercia del mismo. Se obtuvieron los siguientes datos:

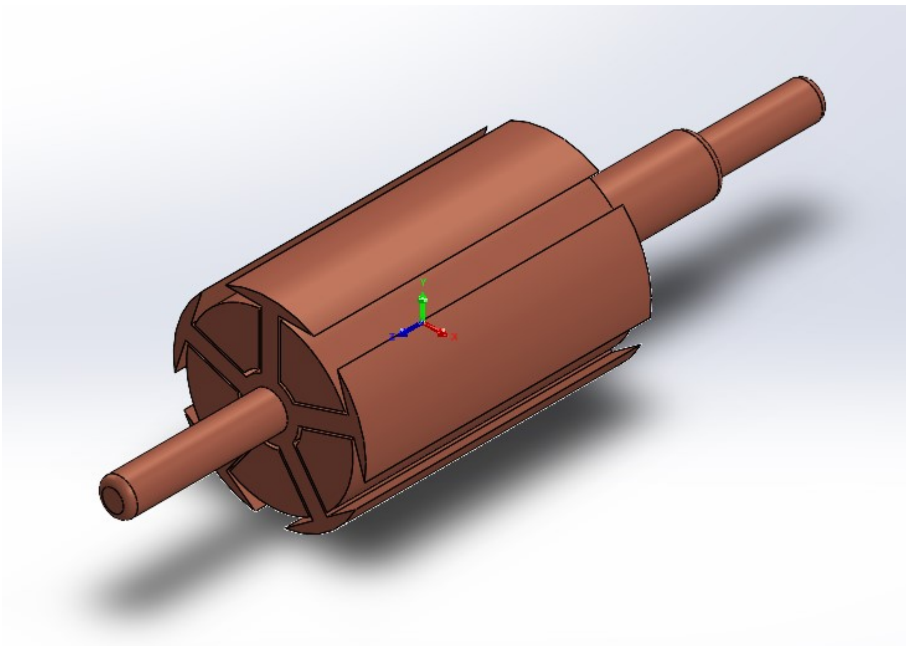

Figure 2: Rotor del motor CC

```
Propiedades de masa de rotor abad
    Configuración: Predeterminado
    Sistema de coordenadas: -- predeterminado --

Densidad = 8900.00000000 kilogramos por metro cúbico

Masa = 0.18142007 kilogramos

Volumen = 0.00002038 metros cúbicos

Área de superficie = 0.01615723 metros cuadrados

Centro de masa: ( metros )
     X = 0.00000000
     Y = 0.00000000
     Z = 0.05444967

Ejes principales de inercia y momentos principales de inercia: ( kilogramos * metros cuadrados )
Medido desde el centro de masa.
     Ix = ( 0.00000000,  0.00000000,  1.00000000)      Px = 0.00001382
     Iy = ( 0.00000000, -1.00000000,  0.00000000)      Py = 0.00004710
     Iz = ( 1.00000000,  0.00000000,  0.00000000)      Pz = 0.00004710

Momentos de inercia: ( kilogramos * metros cuadrados )
Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados.
     Lxx = 0.00004710      Lxy = 0.00000000      Lxz = 0.00000000
     Lyx = 0.00000000      Lyy = 0.00004710      Lyz = 0.00000000
     Lzx = 0.00000000      Lzy = 0.00000000      Lzz = 0.00001382

Momentos de inercia: ( kilogramos * metros cuadrados)
Medido desde el sistema de coordenadas de salida.
     Ixx = 0.00058497      Ixy = 0.00000000      Ixz = 0.00000000
     Iyx = 0.00000000      Iyy = 0.00058497      Iyz = 0.00000000
     Izx = 0.00000000      Izy = 0.00000000      Izz = 0.00001382
```

Figure 3: Datos obtenidos en SolidWorks

3 Modelado del sistema

A continuación se presentará un circuito equivalente del sistema a partir del cual se calcularan las variables, con el fin de obtener la función de transferencia del sistema. El circuito eléctrico de armadura y el diagrama mecánico rotacional se muestran a continuación:

Donde los parámetros de mis sistema son:

- Momento de inercia (J)= 0.00005 $kg.m^2$
- Coeficiente de roce (b)= $1.5319e-4$ $N.m.s$
- Constante de fuerza electromotriz (K=Ke=Kt)= 0.0034 $N.m/A$
- Resistencia de Armadura (R)= $0.0710 Ohm$
- Inductancia de armadura (L)= 0.0127 H
- Tensión de entrada (Ea=V)= Fuente de Tensión
- Tensión del rotor (Eb=e)
- Posición del eje= θ

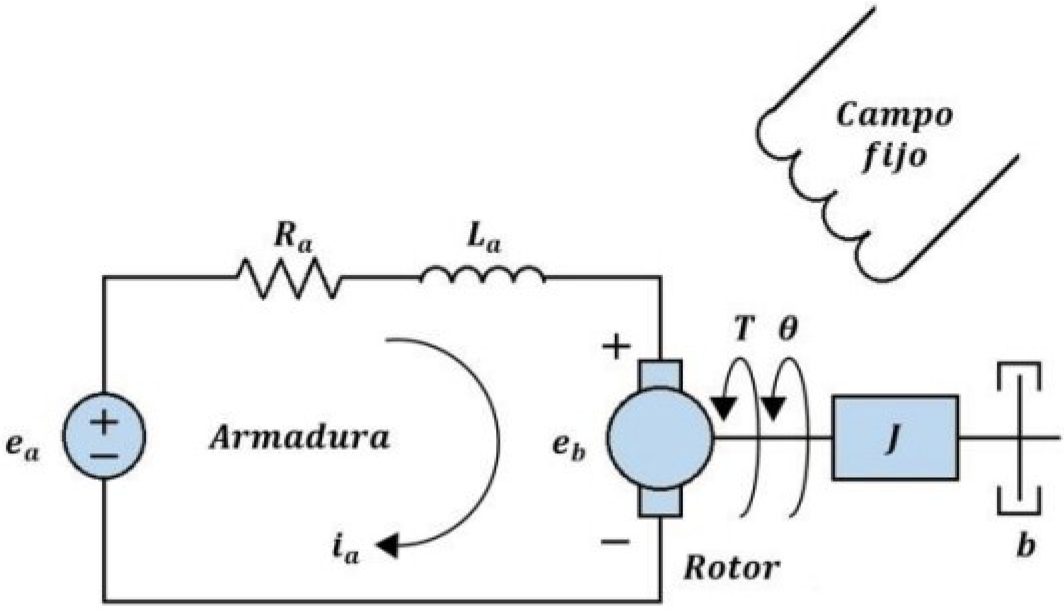

Figure 4: Circuito equivalente de una motor DC

El circuito equivalente del motor se puede descomponer en dos partes, una parte mecánica y otra eléctrica, resultando en las siguientes ecuaciones:

- **Parte Eléctrica**

$$e = k_e\dot{\theta}(t) \tag{1}$$

$$v(t) = Ri + L\frac{di}{dt} + e \tag{2}$$

Reemplazado la ecuación 1 en la ecuación 2 se obtiene:

$$v(t) = Ri + L\frac{di}{dt} + k_e\dot{\theta}(t) \tag{3}$$

Aplicando Laplace a 3:

$$V(s) = RI + LIs + k_e\Theta(s)s \tag{4}$$

$$V(s) = I(R + Ls) + k_e\Theta(s)s \tag{5}$$

$$V(s) = I(R + Ls) + k_e \Theta(s) s \tag{6}$$

$$I = \frac{V(s) - K_e \Theta(s) s}{R + Ls} \tag{7}$$

- **Parte Mecánica**

$$T(t) = K_t i \tag{8}$$

$$T(t) = J \frac{d\dot{\theta}(t)}{dt} + b\dot{\theta}(t) \tag{9}$$

Reemplazando la ecuación 8 en la ecuación 9 se obtiene lo siguiente:

$$K_t i = J\frac{d\dot{\theta}(t)}{dt} + b\dot{\theta}(t) \tag{10}$$

Aplicando la transformada de Laplace a la ecuación anterior:

$$K_t I = J\Theta(s)s^2 + b\Theta(s)s \tag{11}$$

Reemplazando la ecuación 7 en la ecuación 11:

$$K_t[\frac{V(s) - K_e\Theta(s)s}{R + Ls}] = J\Theta(s)s^2 + b\Theta(s)s \tag{12}$$

$$K_t V(s) - K_t K_e \Theta(s)s = (J\Theta(s)s^2 + b\Theta(s)s)(R + Ls) \tag{13}$$

$$K_t V(s) = (J\Theta(s)s^2 + b\Theta(s)s)(R + Ls) + K_t K_e \Theta(s)s \tag{14}$$

$$K_t V(s) = \Theta(s)[(Js^2 + bs)(R + Ls) + K_t K_e s] \tag{15}$$

La función transferencia de un motor de CC sera de la forma:

$$G(s) = \frac{\Theta(s)}{V(s)} \tag{16}$$

Por lo que operamos con la ecuación 15 para obtener G(s):

$$\frac{V(s)}{\Theta(s)} = \frac{(Js^2 + bs)(R + Ls) + K_t K_e s}{K_t} \tag{17}$$

$$G(s) = \frac{\Theta(s)}{V(s)} = \frac{K_t}{(Js^2 + bs)(R + Ls) + K_t K_e s} \tag{18}$$

Nota: Se desconocen las constantes Ke y Kt, lo único que se conoce es que son aproximadas en valores, por lo que se considerará Kt=Ke=K.

$$G(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{(Js^2 + bs)(R + Ls) + K^2 s} \tag{19}$$

Siendo la anterior la función de transferencia de nuestro sistema, la cual se utilizará para constituir un sistema controlado por un PID.

4 Representación en MatLab

En la representación de la función transferencia se consideraron los polinomios del numerador y denominador:

$$num = K$$

$$den = (Js^2 + bs)(R + Ls) + K^2 s$$

El archivo de MatLab se encuentra en el anexo, el cual da una respuesta a lazo abierto, como se muestra en la figura de abajo:

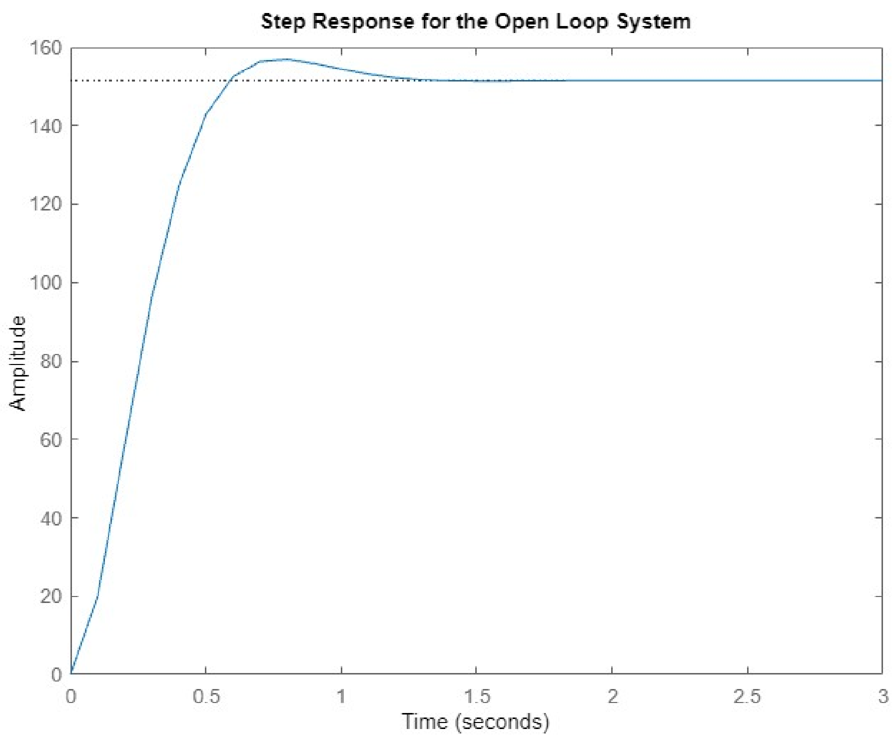

Figure 5: Respuesta a lazo abierto

Observar que a lazo abierto se obtiene un tiempo de establecimiento es aproximadamente de 2 segundos, lo cual para nuestro sistema es un tiempo excesivo y que deseamos mejorar. Por lo que a continuación se verá como responde el sistema con un controlador proporcional y otro PID.

5 Diseño del sistema de control de velocidad de un motor CC

Recordando las ecuaciones halladas anteriormente:

- **Función de transferencia:**

$$G(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{(Js^2 + bs)(R + Ls) + K^2 s} \tag{20}$$

- **Ecuación eléctrica de mi planta:**

$$V(s) = I(R + Ls) + k_e \Theta(s)s \tag{21}$$

- **Ecuación mecánica de mi planta:**

$$K_t I = J\Theta(s)s^2 + b\Theta(s)s \tag{22}$$

Siendo el diagrama en bloques del sistema el siguiente:

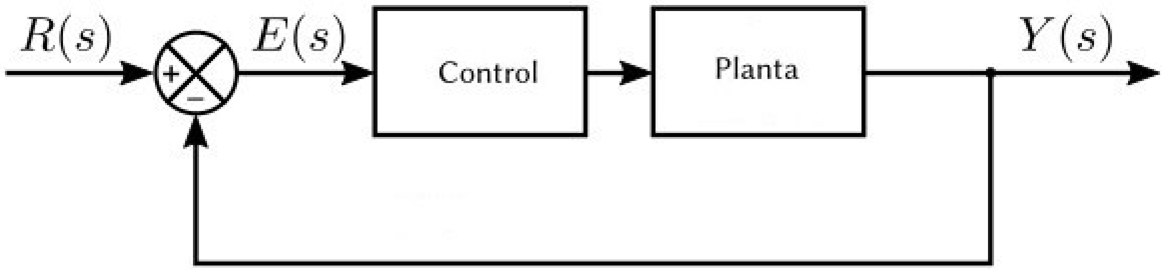

Figure 6: Diagrama en bloques

Antes de empezar con los tipos de controles debemos recordar que la transferencia de un PID es:

$$K_P + \frac{K_I}{s} + K_D.s = \frac{K_D.s^2 + K_P.s + K_I}{s}$$

1. **Control Proporcional**

   Lo primero que se nos ocurre es hacer que la señal de control sea proporcional al error entre la salida y el valor de referencia. Ejecutando el sistema en MatLab nos queda:

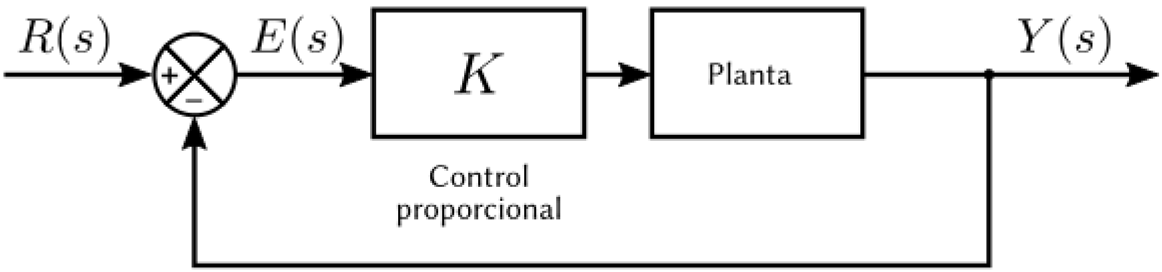

Figure 7: Diagrama en bloques del controlador proporcional

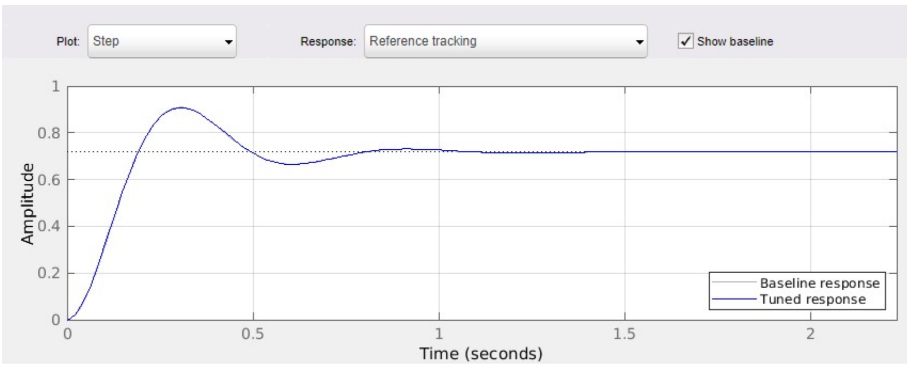

Figure 8: Respuesta con Kp=0.016735

Podemos observar que el tiempo de en llegar al estado estacionario ahora es de aproximadamente 1.4 segundo, logramos reducirlo, pero no se llegó al un valor óptimo para nuestro motor. Para este gráfico MatLab nos dio un valor de Kp=0.016735, el mismo programa con la función pidTuner, nos permite variar este valor para obtener un mejor tiempo de respuesta, como se mostrará a continuación:

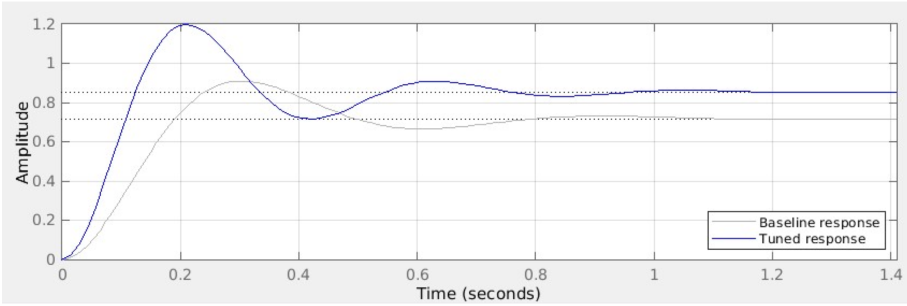

Figure 9: Respuesta con Kp=0.038539

Aquí se aumento el valor de a Kp=0.038539, obteniéndose un tiempo de estado estacionario de aproximadamente 1.15 segundos, se redujo el valor anterior pero no lo suficiente, por lo que si seguimos aumentando el valor de kp el tiempo disminuirá, pero las perturbaciones en la amplitud aumentarán, por lo que un control proporcional no es un tipo de controlador adecuado para nuestro sistema, también se puede observar en gris, la respuesta del sistema con el Kp anterior. A continuación se harán los cálculos para un control PID, con el fin de obtener mejores resultados.

2. **Control PID**

Para solucionar el problema anterior se agrega un termino integral, para eliminar el error de estado estacionario, mientras que el termino derivativo, sintonizado de forma adecuada, se puede utilizar para reducir el sobre paso. Al igual que con el control proporcional, con MatLab, podemos obtener a partir de la planta los valores de Kp, Ki,Kd. Dichos valores son los siguientes, con su correspondiente gráfico:

$$K_P = 0.0145$$

$$K_I = 0.0606$$

$$K_D = 0.000867$$

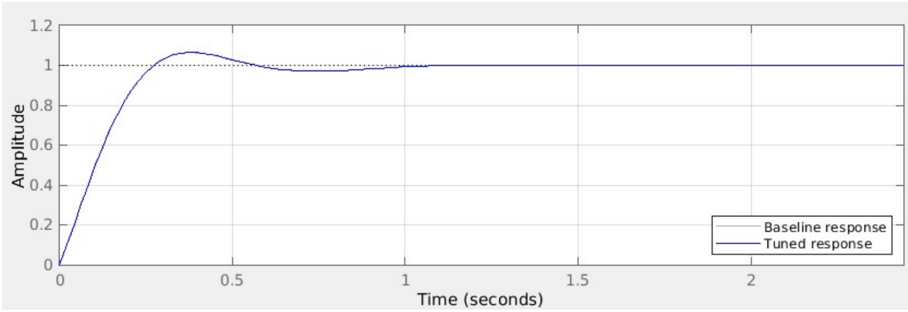

Figure 10: Respuesta del PID

Con la misma herramienta utilizada anteriormente, pidTuner, podemos variar los valores de las constantes, para mejorar la respuesta y asi obtener un tiempo de respuesta mejor. Se puede observar que el tipo de respuesta es mucho menor que el del controlador proporcional modificado, es decir, inicialmente el controlador PID es mas rapido para llevar mi planta al estado estacionario. Ahora modificaremos los valores del PID para ver si es posible obtener una mejor respuesta:

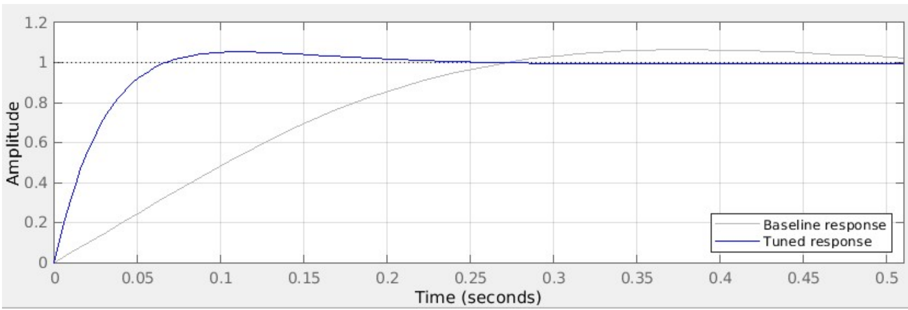

Figure 11: Respuesta del PID con sus valores modificados

Como se ve en la figura anterior, podemos observar que el tiempo se redujo bastante, ahora nuestra respuesta de estado estacionario esta en aproximadamente 0.25 segundos, lo cual es un valor deseado y util para controlar nuestra planta. Los valor de Kp, Ki,Kd resultantes del grafico anterior son los siguientes:

Controller Parameters

	Tuned	Baseline
Kp	0.10097	0.014505
Ki	0.35887	0.060634
Kd	0.0071028	0.00086748
Tf		

Figure 12: Respuesta del PID con sus valores modificados

6 Anexos

(a) Función de transferencia (MatLab)

$J = 0.00005;$

$b = 0.00015319;$

$K = 0.0034;$

$R = 0.071;$

$L = 0.0127;$

$num = K;$

$den = [(J * L)((J * R) + (L * b))((b * R) + K^2)];$

$step(num, den, 0 : 0.1 : 3)$

$title('StepResponsefortheOpenLoopSystem')$

(b) Control Proporcional (MatLab)

$J = 0.00005;$

$b = 0.00015319;$

$K = 0.0034;$

$R = 0.071;$

$L = 0.0127;$

$num = K;$

$den = [(J * L)((J * R) + (L * b))((b * R) + K^2)];$

$Gp = tf((num), (den));$

$Gc = pidtune(Gp,' P');$

$pidTuner(Gp, Gc)$

(c) Control PID (MatLab)

$J = 0.00005;$

$b = 0.00015319;$

$K = 0.0034;$

$R = 0.071;$

$L = 0.0127;$

$num = K;$

$den = [(J * L)((J * R) + (L * b))((b * R) + K^2)];$

$Gp = tf((num), (den));$

$Gc = pidtune(Gp,' PID');$

$pidTuner(Gp, Gc)$