Sistemas de control I

Monografía

Universidad Nacional de Córdoba

Facultad de Ciencias Exactas, Físicas y Naturales

Integrantes:

* Colazo, Agustín (38.986.764)
* Passaglia, Nicolás (38.987.149)

nicopassaglia@hotmail.com

[agustin.colazo@gmail.com](mailto:agustin.colazo@gmail.com)

Córdoba, 2016

Diseño de un Sistema de Control lineal

# Introducción

En este trabajo se busca desarrollar y simular un sistema de control que represente un sistema físico real, utilizando para esto lo aprendido durante el curso Sistemas de Control I. La situación planteada es la de un pequeño auto al cual a través de un potenciómetro angular se le mide la distancia recorrida, la idea es que luego de un cierto valor determinado y predefinido de distancia el auto se frene teniendo una inmejorable precisión. La intención del proyecto sería que este auto pueda ser utilizable en expediciones de geología y biología para la captura de imágenes en lugares inalcanzables por el hombre o para el acercamiento a muestras en los que la mano del hombre podría traer resultados catastróficos.

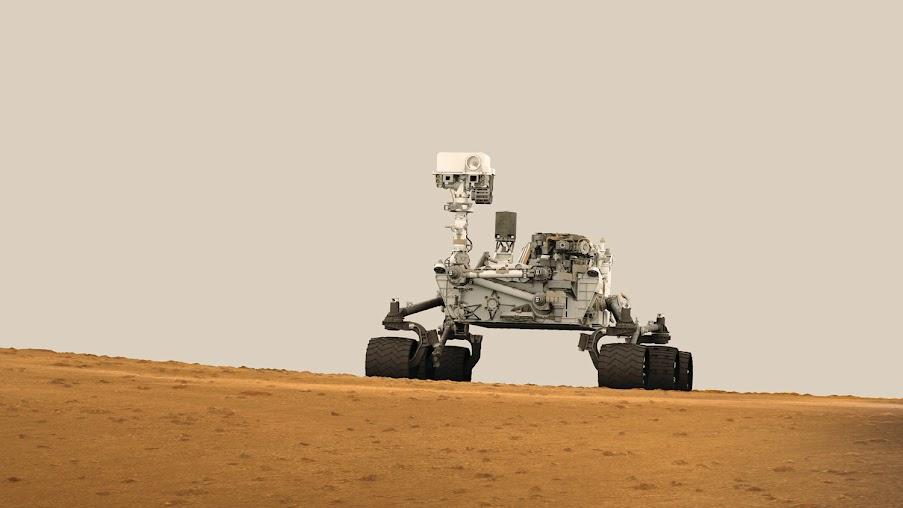


Fig. 1

Para modelar esta situación se hicieron varias simplificaciones físicas para evitar alinealidades y disminuir la complejidad mecánica.

# Problemática y especificaciones

Por lo tanto la problemática planteada es la siguiente. Necesitamos que logre detenerse con total precisión luego de una determinada distancia recorrida por lo tanto el sobrepaso debe ser menor a 0.1%, además también es deseable que el sistema no sea muy lento por lo que fijamos un tiempo de levantamiento menor a 1.4 segundos.

Resumiendo, los requerimientos de diseño son:

* Sobrepaso < 0.1%
* Tiempo de Levantamiento < 1.4 Segundos.
* Error de rampa < 10%

Modelado Matemático

## Obtención de la función de transferencia

Se va a implementar el modelado de un motor de corriente continua de imán permanente, se ha optado por éstos ya que no requieren una fuente externa que genere flujo magnético lo que simplifica el control y además de esto suelen ser muy eficientes y tener una buena relación par-peso.

Para comenzar con el modelado podemos pensar al motor como un circuito donde tenemos una tensión de entrada, una resistencia propia de la armadura y una inductancia presente en los terminales y una fuerza contra electromotriz.

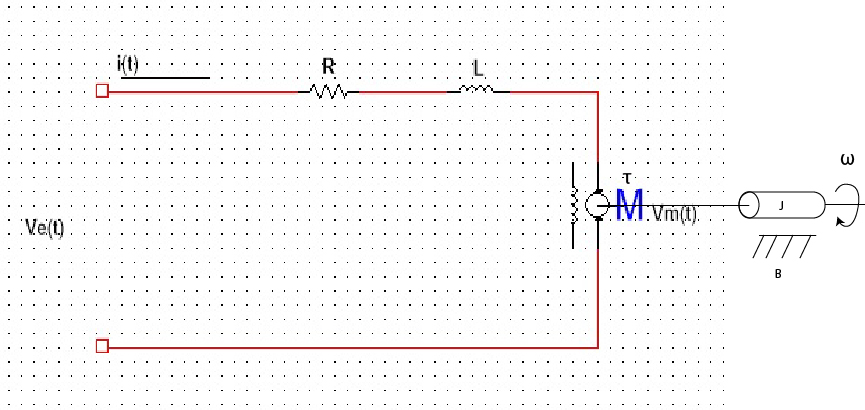


Fig. 2

A partir de este planteo podemos plantear la conocida ley de Kirchhoff que indica que la tensión de entrada es igual a la suma de caída de potencial en cada componente, a lo cual obtenemos:

Donde

* es la tensión de entrada
* R es la resistencia en ohms de la armadura del motor
* es el valor de inductancia del bobinado medido en Henrios
* i(t) es la corriente que circula
* es la F.E.M inducida en el bobinado por el efecto de Faraday.

Podemos decir también que:

Donde es la constante de la fuerza contra electromotriz [V.s/rad], la cual el fabricante en nuestro caso la indica a la inversa como constante de velocidad [V/rpm]. Y es la velocidad angular del motor en rad/seg.

Y con respecto al torque del motor podemos afirmar que:

Y:

Donde en la primera ecuación es la constante de par y la segunda ecuación sale de las ecuaciones diferenciales de la mecánica newtoniana, cuyas constantes son , indicando momento de inercia del motor y que representa el coeficiente de rozamiento viscoso .

Corresponde al torque de carga reflejado al motor la cual se obtiene a partir masa del auto y las fricciones involucradas multiplicadas por la relación de reducción al cuadrado.

Una vez obtenidas estas ecuaciones procedemos a realizar la Transformada de Laplace en ambos miembros de todas las ecuaciones obteniendo las siguientes igualdades:

Suponiendo que todas las fuerzas de fricción son nulas, entonces .

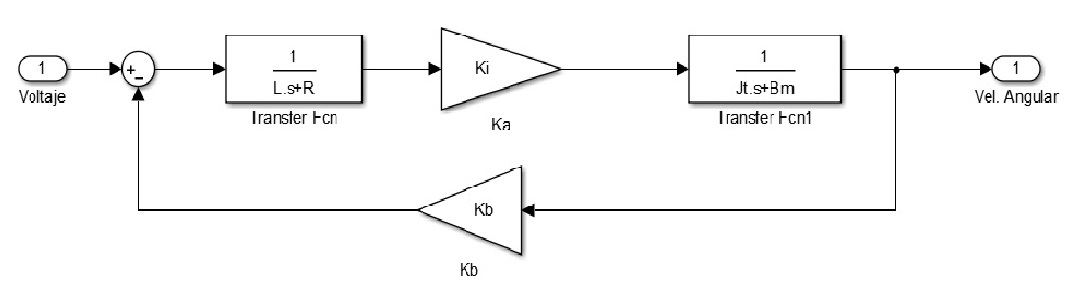
Una vez obtenidas estas ecuaciones ya es posible armar un diagrama de bloques

Fig. 3

Ahora procederemos a obtener la función de transferencia, pero antes debemos considerar que el motor va a estar sometido a una carga, lo cual producirá una variación en el momento de inercia y el coeficiente de rozamiento viscoso.

Para el momento de inercia total al propio del motor hay que adicionarle el de las ruedas y el del eje. Colocando el motor en la parte trasera del auto sobre el eje, para simplificar, **suponemos que toda la masa del móvil se encuentra concentrada en las ruedas traseras del mismo, ya que es aquí donde colocamos el motor.**

Ahora procedemos a aplicar la regla de Mason para obtener la FT del motor.

Cantidad de caminos directos = 1

Cantidad de Lazos = 1

M1 =

\*Datos obtenidos de la hoja de datos del motor ubicada al final del trabajo

Por lo tanto la función de transferencia quedaría:

(Polos calculados en Matlab)

**Motor:**

\*Datos obtenidos de la ficha técnica del motor

**Auto:**

Suponiendo que la masa total del vehículo se encuentra concentrada en ambas ruedas traseras. Luego:

(Caso de un cilindro macizo)

Como consideramos dos ruedas:

**Ejes:**

Los valores obtenidos de masa de los ejes, momento de inercia y demás son despreciables con respecto a las demás variables por lo tanto no se consideran en los cálculos.

**Engranaje:**

Entre el motor y el eje de la rueda se le acoplo un tren de engranajes con una relación 226:639, ya que con esto podemos limitar la velocidad final del auto de 113km/h a 40km/h teniendo en cuenta la velocidad angular de salida del motor y el radio de las ruedas del auto. La velocidad máxima del auto es 11,11 m/s.

Para obtener esta relación se hicieron los siguientes cálculos:

Deducimos a partir de la velocidad angular máxima del motor extraída de la hoja de datos, 10000 rpm, o lo que es lo mismo 1047 rad/s, que la velocidad máxima que alcanzaría el auto serian 31,4 m/s (113km/h) ya que v = .R. Esto nos pareció una velocidad excesiva por lo tanto decidimos reducirla a 40km/h. Por lo tanto la relación de reducción de los engranajes será 226:639.

Una vez establecido esto, podemos calcular el momento de inercia total:

Para obtener la posición angular del móvil, luego del diagrama de bloques del motor es preciso insertar un bloque integrador “1/s” ya que:

Tomando transformada de Laplace:

Así obtendríamos la posición angular del auto.

A continuación en el siguiente bloque se coloca el tren de engranajes que logra la reducción.

Por último para ya obtener el desplazamiento final del móvil, hacemos el producto entre la posición angular y el radio de las ruedas puesto que:

Donde se encuentra en radianes.

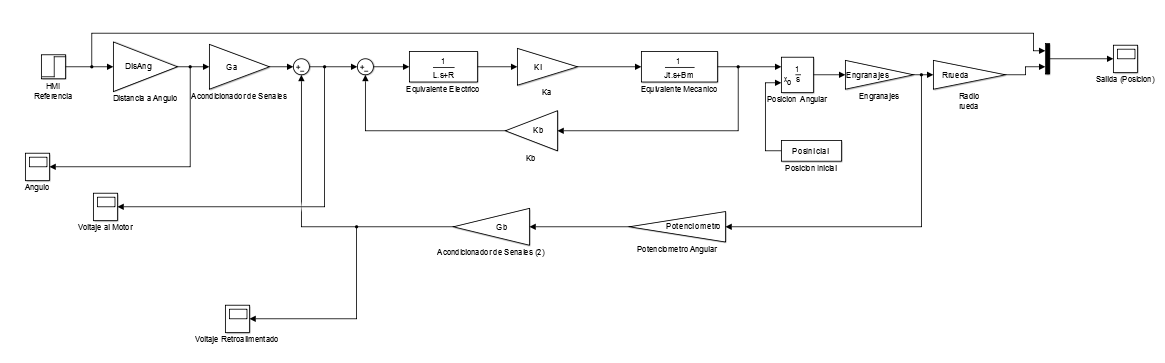
A continuación colocamos el diagrama de bloques del sistema:

Fig. 4

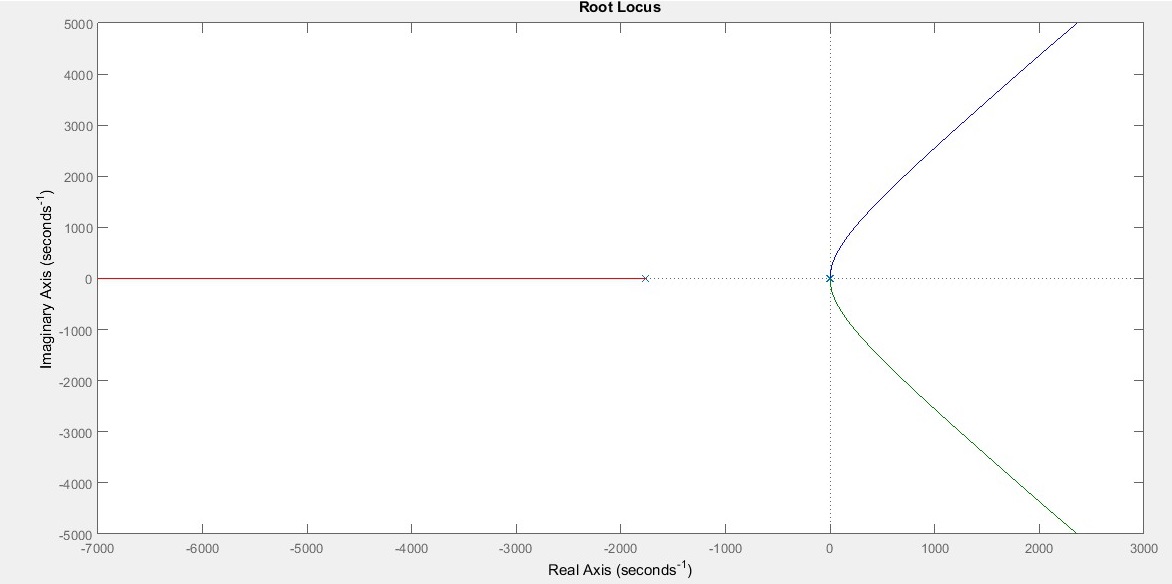
En HMI Referencia ingresamos la distancia que deseamos que el vehículo recorra. DisAng es una función respecto del radio de las ruedas que calculara el ángulo de giro requerido para recorrer dicha distancia.

Luego este ángulo ingresa al acondicionador de señales el cual consiste en una ganancia que entregara 0.5 voltios cada 360º. Hacemos esto para independizarnos de la distancia recorrida en el lazo cerrado. Ya que es más fácil conseguir sensores de desplazamiento angular que sensores de desplazamiento lineal para esta aplicación. Podríamos usar sensores de distancia pero los mismos no son útiles para el caso. Ya que se usan para medir distancia y requeriríamos de un objeto de referencia desde el que podamos medir. El objetivo de este sistema es poder desplazarse una determinada distancia sin depender de objetos de referencia. Podemos notar que la distancia lineal volverá a formar parte del sistema al final cuando se multiplica por el radio de las ruedas antes de la salida. De esta manera obtendremos la distancia final recorrida.

Una vez realizado esto, se retroalimenta esta posición angular obtenida a través de la implementación de un potenciómetro angular, el cual registra la variación de ángulo de las ruedas del auto para convertir esta señal en un voltaje. El potenciómetro otorga 10 voltios cada 3500º o lo que es lo mismo, 1 voltio cada 350º. Este voltio luego será introducido en un acondicionador de señales, básicamente una ganancia, el cual ajustara el voltaje para que entregue 0.5 voltios cada 360º según la siguiente función.

Esta señal luego entra a un sumador que llega a la entrada del bloque del motor.

# Función de Transferencia de Lazo Abierto

Lugar de raíces de la FTLA:

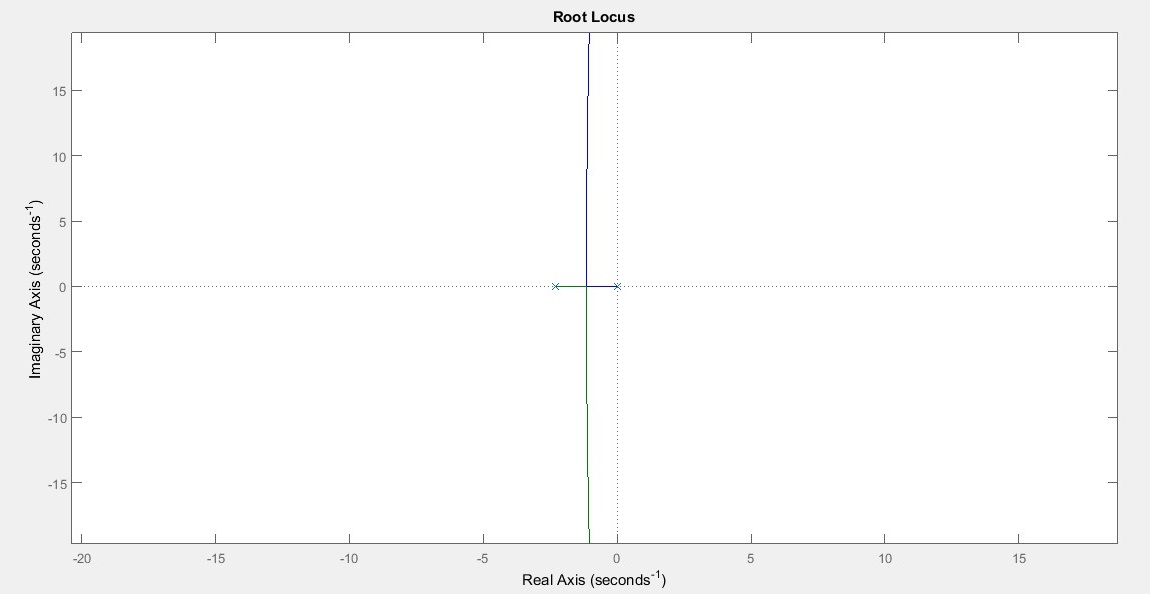
Fig. 5

Fig. 6

El lugar de raíces que se ve anteriormente corresponde a las raíces de la ecuación característica 1+kGH para

# Función de transferencia a lazo cerrado

Notamos que los polos son complejos conjugados, lo que indica que el sistema tiene sobrepaso.

# Respuesta del Sistema

# Análisis de estabilidad

Para analizar la estabilidad utilizaremos el método de Routh-Hurwitz.

Lo primero que hacemos al calcular Routh-Hurwitz es buscar la ecuación característica.

Donde:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | 0 |
|  |  |  |

Para lograr la estabilidad, no debe haber cambios de signos en la primera columna de la tabla, por lo tanto, α1 y β1 deben ser mayores que cero. De esta manera la ganancia variable del lazo directo debe ser:

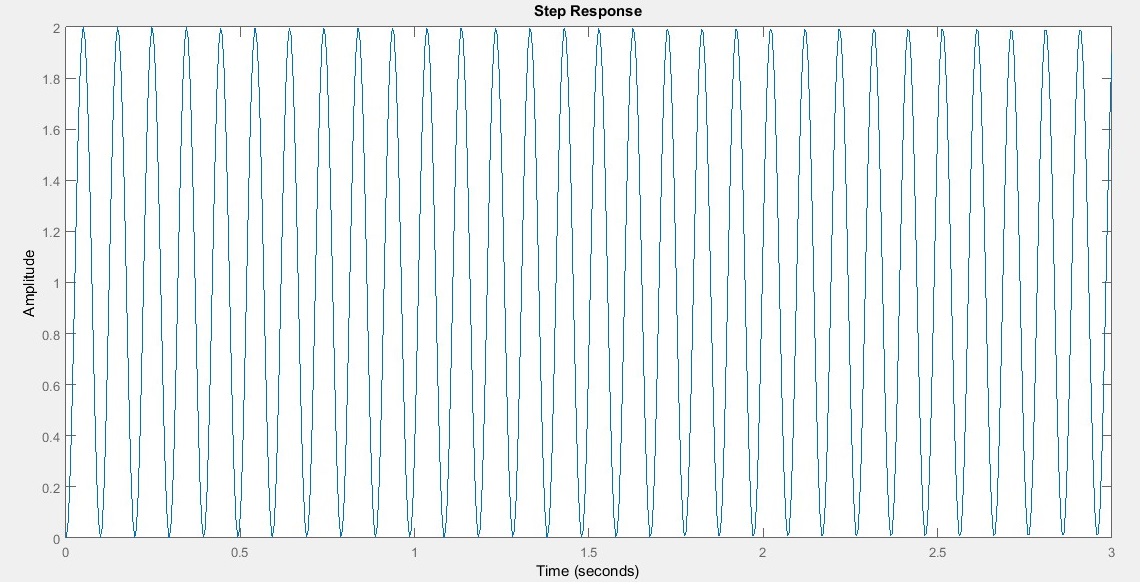
Podemos ver el sistema críticamente estable cuando se aplica una ganancia de . Aunque no se podrá llegar en la práctica hasta este punto ya que la ganancia necesaria es excesivamente grande. Es interesante en el caso teórico para confirmar los cálculos hechos anteriormente.

Fig. 6

Fig. 7

Por el momento hemos confirmado que el sistema en cuestión se volverá inestable para una ganancia de lazo muy grande, pero no hemos visto que tan próximo está a nuestras especificaciones de diseño previamente establecidas.

* Sobrepaso < 0.1%
* Tiempo de levantamiento < 1.4 Segundos

# Error en Régimen Permanente

A continuación se procede a determinar los errores en régimen permanente de nuestro sistema para diferentes entradas (escalón, rampa y parábola), a partir de lo cual se podrá determinar de qué tipo es el sistema.

Para una entrada R(s), el error en régimen permanente de nuestro sistema estará dado por:

Entrada Escalón ()

Entrada Rampa ()

El error es para un entrada rampa es del 77%.

Entrada Parábola ()

Esto deja en evidencia que el sistema en cuestión es de ***tipo 1***.

También podemos observar que el sistema no cumple con el requerimiento de diseño de que el error de rampa sea menor al 10%.

A continuación podemos ver las graficas de Simulink donde se comparan las entradas y las salidas del sistema sin compensar. La próxima imagen muestra la respuesta para una entrada escalón. Se puede verificar que para este caso no hay error, lo cual concuerda con el cálculo hecho anteriormente.

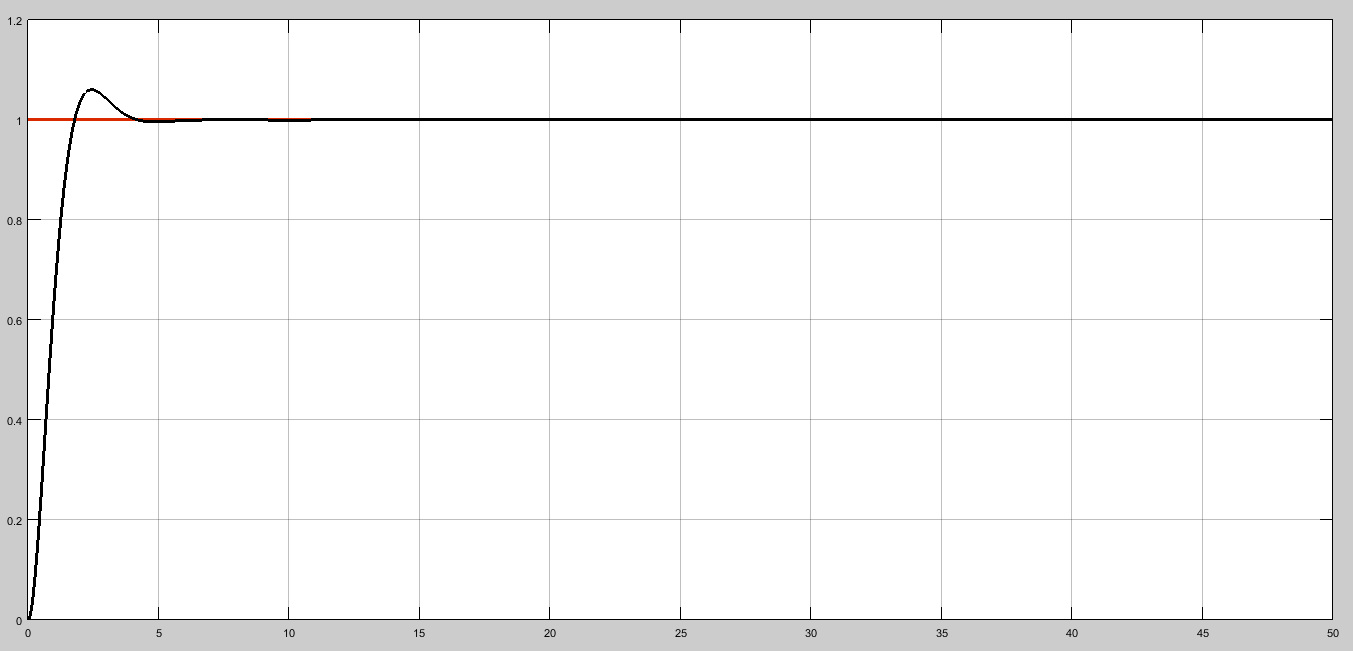


Fig. 8

Y en la figura que se muestra a continuación se observa la respuesta del sistema ante una entrada rampa. En este caso, se observa que si hay error en régimen permanente, lo cual concuerda con el cálculo resuelto anteriormente.

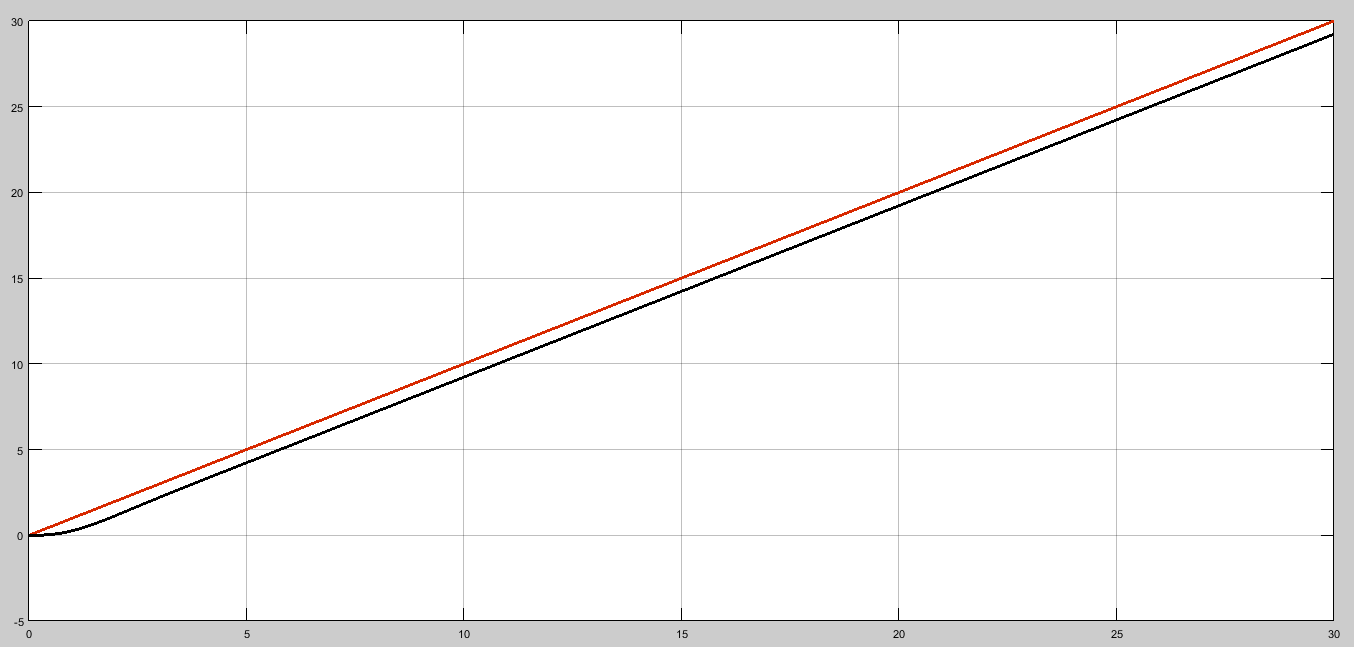


Fig. 11

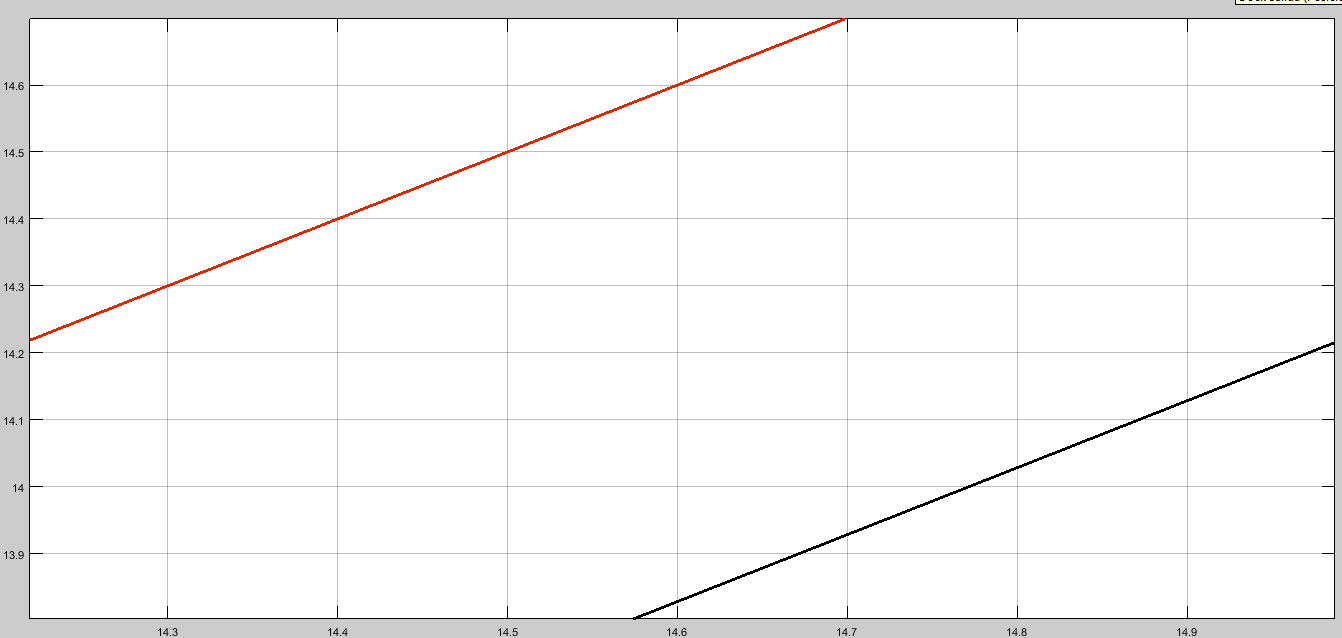


Fig. 9

La división del eje vertical es de 0.1.

## Sobrepaso

Para el análisis del sobrepaso se analizan los polos de la FTLC, los cuales son (-1.7645, -0.0012 + 0.0013i, -0.0012 - 0.0013i). Notamos que dos de los polos tienen parte imaginaria, lo cual indica que el sistema tiene sobrepaso.

Utilizaremos la función de MATLAB “stepinfo” para calcular el sobrepaso del sistema y el tiempo de levantamiento. En las imágenes a continuación notamos que se cumple el criterio del tiempo de levantamiento ya que este es menor al estipulado, pero el sobrepaso es mayor al especificado en el diseño del sistema (6%).

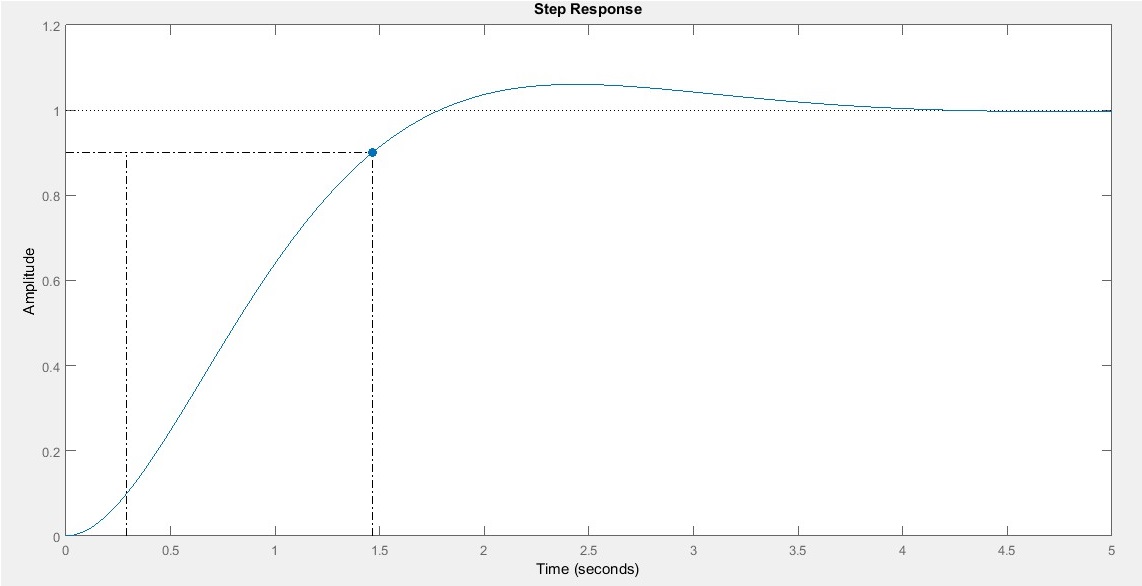
Graficaremos la respuesta del sistema a un escalón y podemos notar el sobrepaso en la figura.

Fig. 10

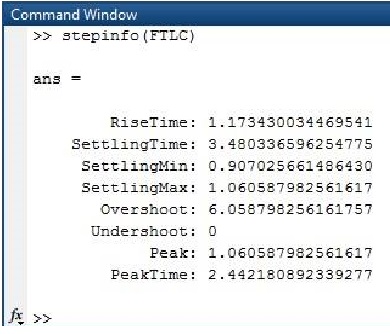


Fig.11

A continuación se tratara de compensar el sistema utilizando el método de lugar de raíces. Se buscara acercar los polos al eje real y que de esta manera el sobrepaso será menor.

# Compensación por el método de lugar de raíces

Como primer paso para calcular el compensador debemos primero calcular los polos deseados a partir de las especificaciones de diseño. Para lograr esto se utilizará la función prototipo de segundo orden. Por lo tanto a partir del sobrepaso vamos a calcular xita, y luego junto a este xita y el tiempo de levantamiento obtendremos el valor de ωn.

Por lo tanto si el sobrepaso =0.1% y si su fórmula es:

entonces xita vale

Luego como se habia definido el tiempo de levantamiento tr como 1.4 segundos. Se puede despejar ωn de la siguiente formula aproximada. Obteniendo asi el valor de wn.

Luego a partir de la ecuación característica de segundo orden

Podemos calcular el valor de los polos deseados. Estos fueron calculados en matlab con la siguiente fórmula:

Obteniendo así los siguientes polos complejos conjugados p1 y p2:

Para calcular el compensador requerido se utilizara el método de la bisectriz

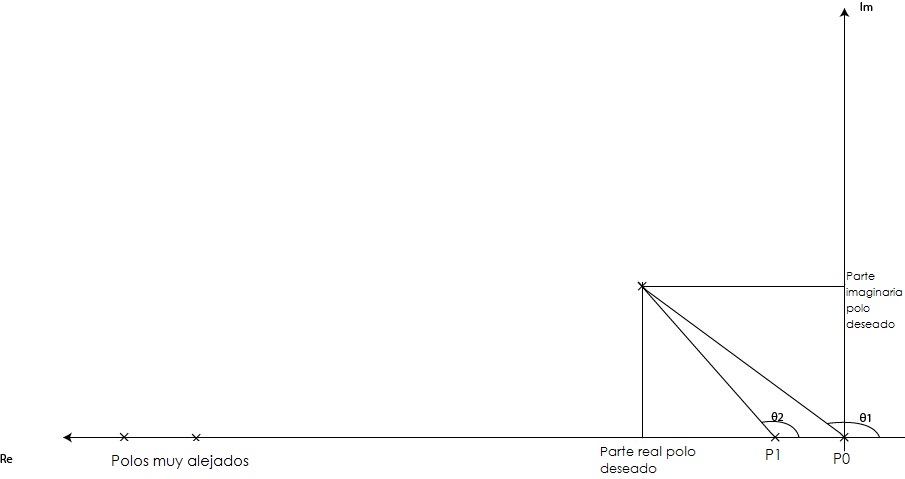


Fig. 12

Esta compensación se toma a partir de la función a lazo abierto.

Se consideran solamente los ángulos aportados por los polos dominantes (los más cercanos al origen). La aportación de los polos que están muy lejos del origen se considera despreciable ya que el ángulo es de valor muy pequeño.

Por lo tanto, el ángulo del compensador será:

se compara con el múltiplo impar de 180º más cercano al mismo. Así se obtiene el ángulo que se utilizara para determinar el polo y cero del compensador.

El compensador a diseñar será uno en adelanto (Cero más cercano al origen y polo más alejado) el cual mejora la respuesta transitoria.

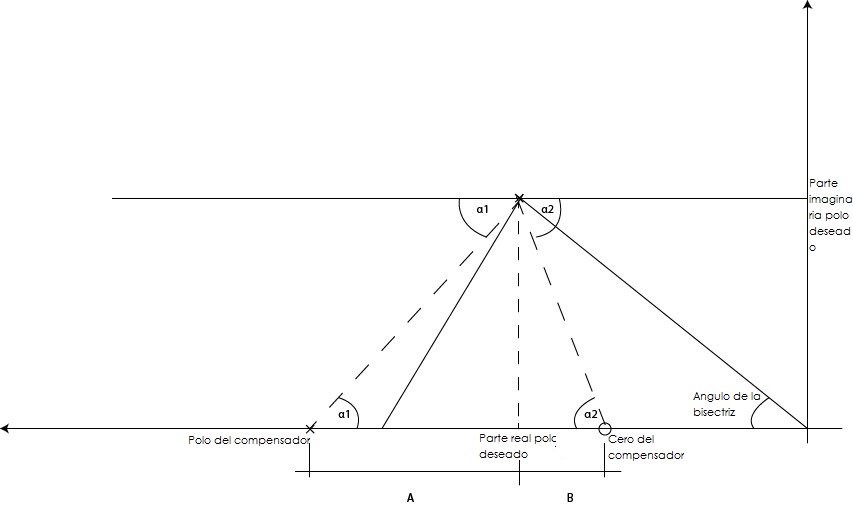
La siguiente imagen indica el cero y polo del compensador y como se calcularon:

Fig. 13

Luego, se traza una recta paralela al eje horizontal que pasa por el punto que corresponde a la parte imaginaria del polo deseado. Se traza una segunda recta que pasa por el origen del eje cartesiano y que corta a la recta trazada anteriormente justo por arriba de la parte real del polo deseado. Como se ve en la figura anterior.

Después, se traza la bisectriz del ángulo obtuso entre ambas rectas. Finalmente, se trazan dos rectas para ambos lados (las rectas punteadas) que forman un ángulo de con la bisectriz. El polo y el cero del compensador corresponden al cruce de las rectas punteadas con el eje real del sistema cartesiano como se puede observar en la imagen.

Por último es necesario calcular la ganancia de dicho compensador. Para esto hacemos uso de la siguiente fórmula:

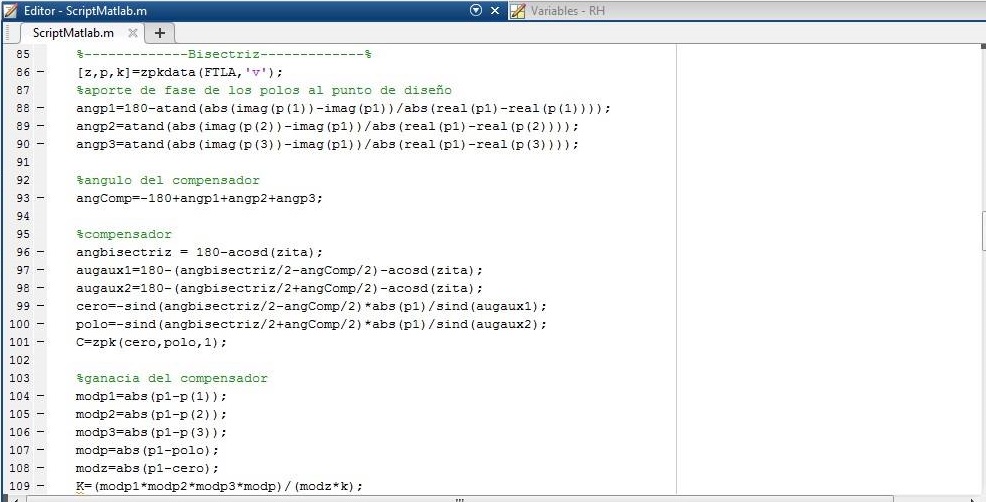
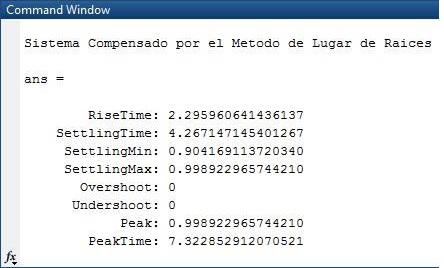
Donde es el valor de ganancia que deseamos obtener y cuyo valor es:

Ya con todo esto definido podemos adquirir la función de transferencia del compensador la cual es la siguiente:

Los cálculos se automatizaron desarrollando un script en Matlab que calcula el polo, el cero y la ganancia del compensador.

La ecuación característica del sistema compensado es:

Las raíces obtenidas son **-2 + 0.9i** y **-2 – 0.9i** que se asemejan a los polos deseados.

 Fig. 14Fig.15

En la figura 15 es posible ver la información sobre la respuesta del sistema donde podemos notar que no hay sobrepaso, no obstante el tiempo de levantamiento es mayor al deseado (2.30 > 1.40). Luego, se va a probar ajustar la ganancia del compensador observando si es posible llegar a los criterios de diseño con este compensador.

La ganancia del compensador es de 0.85 y se probó ajustándola a 1.3

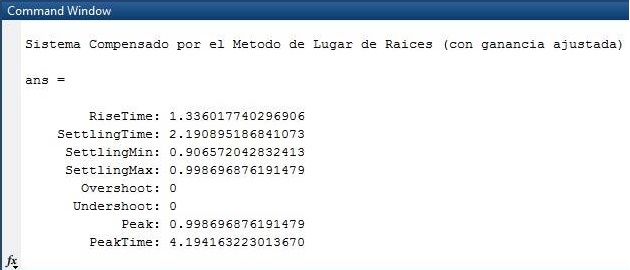


Fig. 16

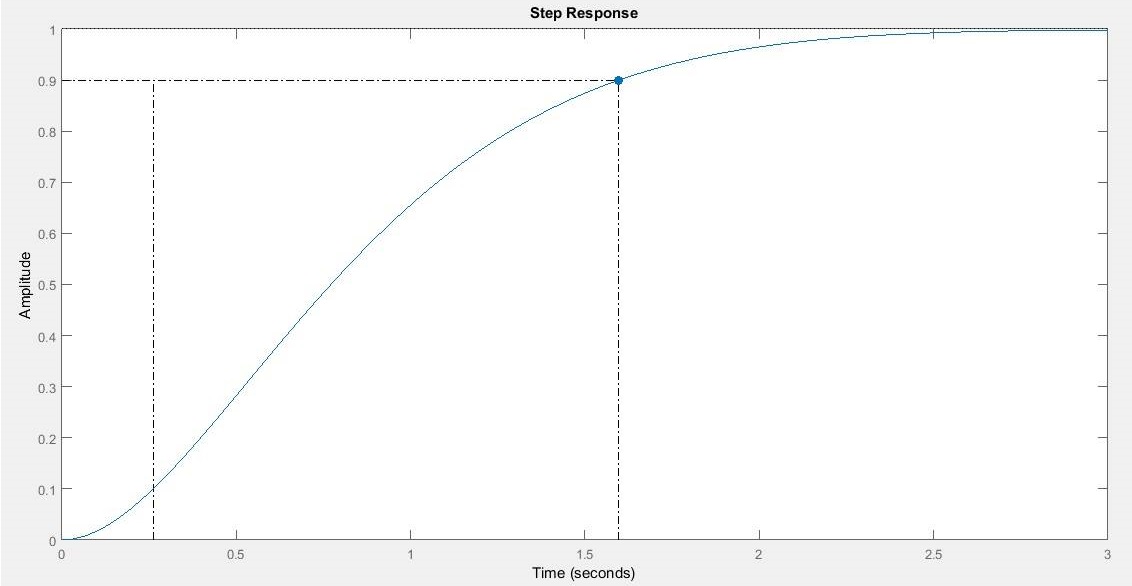


Fig. 17

Como podemos ver, al ajustar la ganancia el sistema cumple con los requisitos de diseño.

# Análisis de frecuencia

En la siguiente imagen observaremos la respuesta en frecuencia de la función de transferencia a lazo abierto sin compensar.

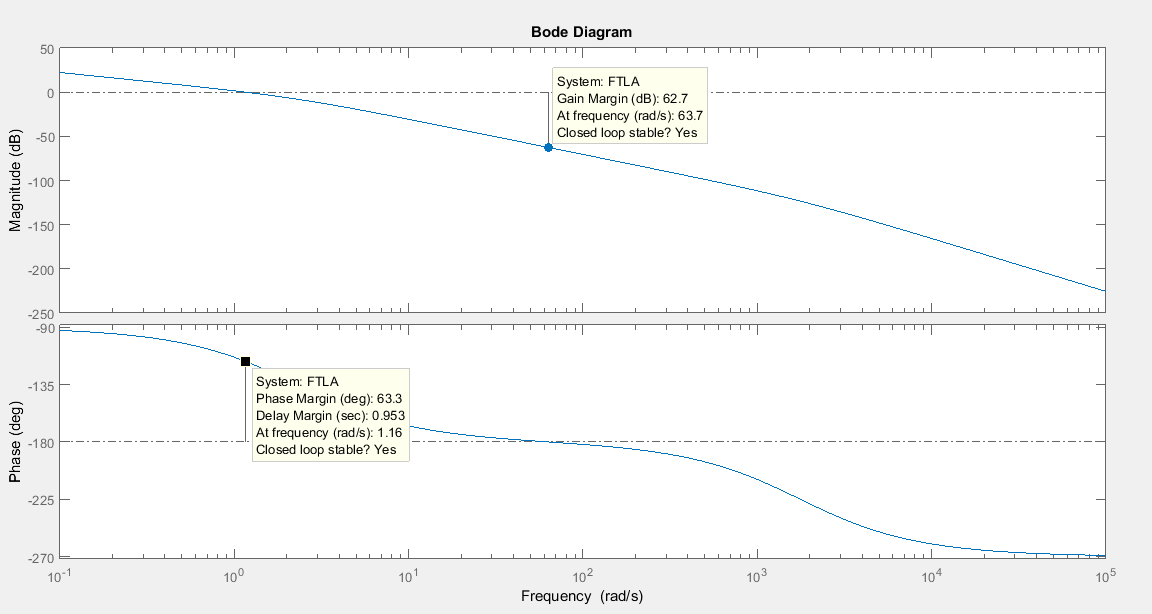


Fig. 18

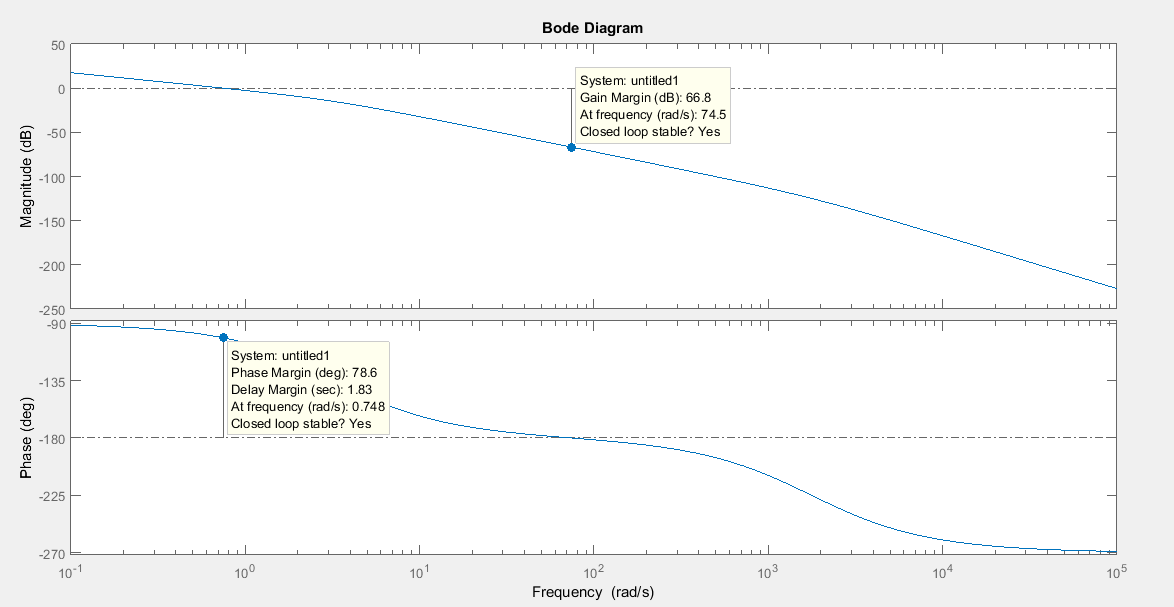
Podemos notar viendo el diagrama de fase que el sistema es estable y que tiene un margen de fase de 63.3º. También podemos notar que el margen de ganancia es de 62.7 dB. Siendo esto la ganancia que se le puede agregar al sistema antes de que se vuelva inestable. Convirtiendo a veces:

Lo cual concuerda con lo calculado anteriormente usando Ruth-Hurwitz.

Siendo MF=63.3, podemos determinar que correspondiendo un que es similar a lo obtenido.

Luego podemos ver la respuesta en frecuencia de la función de transferencia a lazo abierto después de haber compensado.

Fig. 19



Siendo MF=78.6, podemos determinar que correspondiendo un que es similar a lo obtenido.

Podemos notar que el margen de fase aumento, significando esto un aumento en la estabilidad del sistema y una disminución en el sobrepaso.

La siguiente imagen muestra ambas curvas de bode.

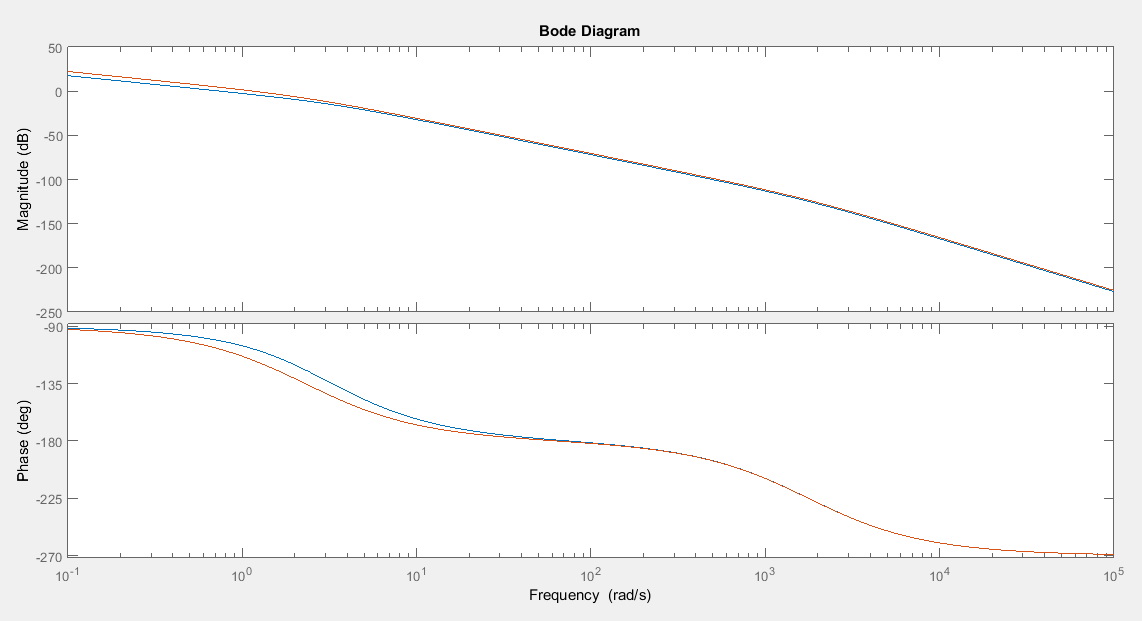


Fig. 20

# Compensador en atraso

### Marco Teórico

Error ante una entrada rampa para el sistema compensado:

Se pretende obtener Ev<10%, por lo tanto Kv=10.

Se busca no alterar la característica de la respuesta transitoria, por lo tanto Kc=1, luego:

Para no afectar la respuesta transitoria los ángulos que aportan el polo y el cero deben ser mínimos. Luego se elige un polo en 0.02. Entonces:

El compensador en adelanto es:

A continuación se muestran imágenes de Simulink mostrando el error en régimen antes de compensar y después. Se hizo un acercamiento en las imágenes para poder verificar los valores calculados con los valores de la simulación.

Antes de compensar:

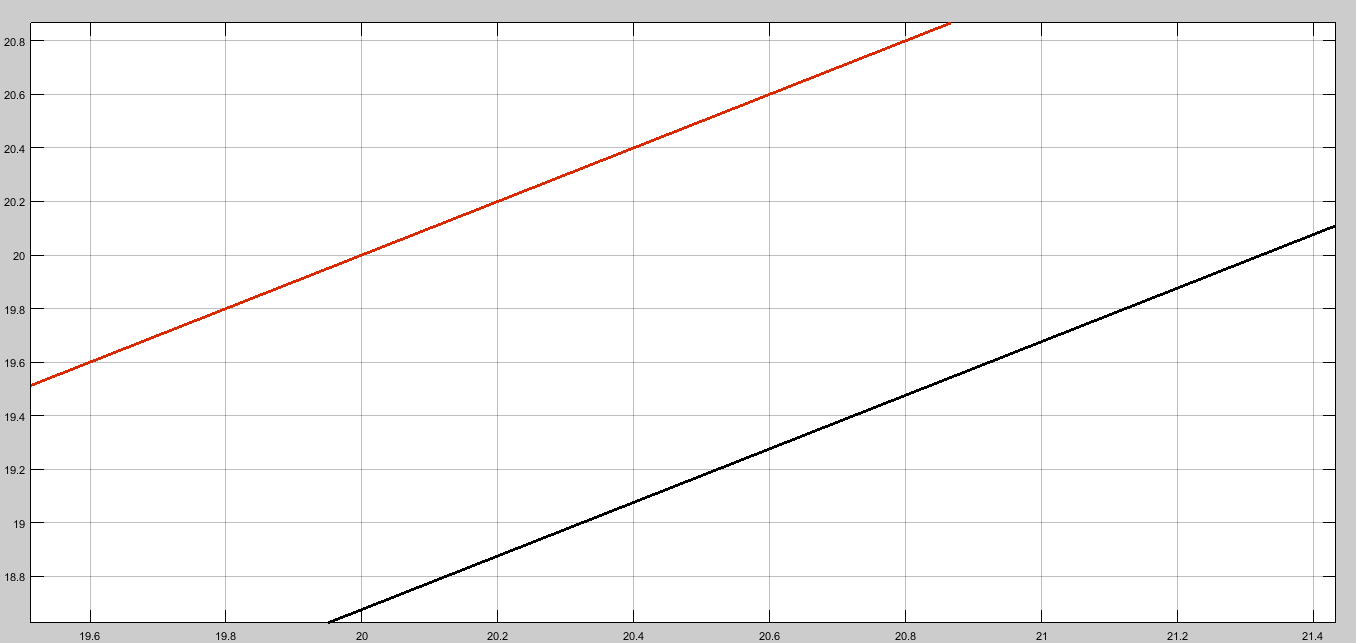


Fig. 21

La división del eje vertical es de 0.2.

Después de compensar:

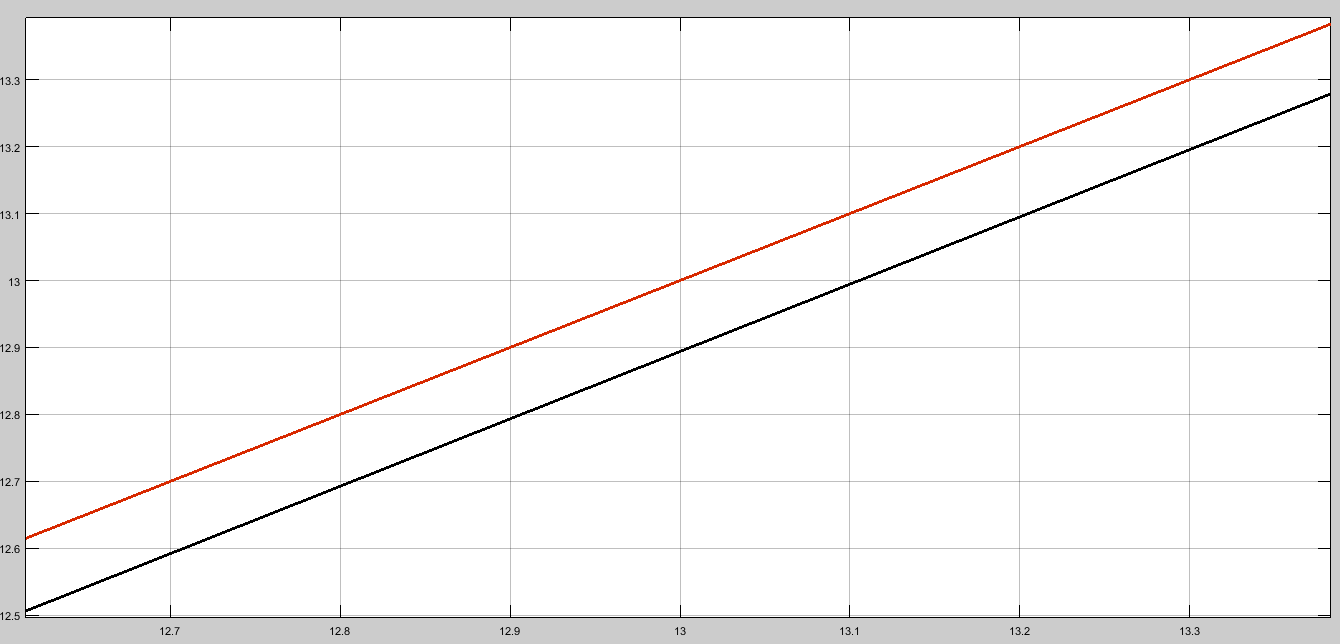


Fig. 22

La división del eje vertical es de 0.1.

Se puede observar que los valores en las graficas coinciden con los valores calculados.

# Script de MatLab

Clear all

clc

% % Motor de Corriente Continua con Carga % %

% ParametrosTecnicos del Motor

Vnom=9;

R=0.113;

L= 0.064e-3;

Bm=30/(pi\*9.19\*1000);

Kb = 1/(881\*2\*pi/60);

Ki=10.8e-3;

Jm = 12.8e-7;

Gain=5;

%Acondicionador de Senales

VoltajeOtorgado=0.5;

Ga=VoltajeOtorgado/(360);

Gb=3500/10\*VoltajeOtorgado/360;

%Carga%

%Ruedas

Rrueda=(0.03); %diametro de 10 cm

MasaTotal=16; %Masa de ruedas mas la masa del auto y del motor.

Jc=(MasaTotal/2)\*(Rrueda^2); %momento de inercia de la carga

Bc=0;

%Engranajes

MAXVelocidadAuto=40; %km/hora

wmotor=10000\*2\*pi/60;

wauto=MAXVelocidadAuto/3.6/Rrueda;

%RelacionEngranajes=1; %Esta relacion es de prueba.

RelacionEngranajes = wauto/wmotor;

%Ejes

Reje = 0.005; %diametro de un centimetro

Meje=0.02; %Pesa 20 gramos

Je = Meje \* (Reje^2); %momento de inercia de los ejes

% Momento de Inercia y Rozamiento Viscoso con Carga

Jt=Jm+Jc\*(RelacionEngranajes^2); %Momento de inercia total

Bt=Bm+Bc;

%Problematica (en metros)

PosInicial=0;

Objetivo=1;

% Especificaciones Transistorio

tr=1.4; % Tiempo de Levantamiento (lo que tarda en llegar del 10% al 90%)

Mp=0.1/100; %El sobre paso no puede ser 0!

zita=1/sqrt(((-pi/log(Mp))^2)+1);

%if(zita<0.69)

% wn=3.2/(zita\*ts);

%else

% wn=4.5\*zita/ts;

%end

wn=(0.8+2.5\*zita)/tr;

F\_Ideal = tf(wn^2, [1, 2\*zita\*wn, wn^2]);

display('Sistema "Ideal"')

stepinfo(F\_Ideal)

polosDeseados = [-zita\*wn+1i\*wn\*sqrt(1-zita^2);-zita\*wn-1i\*wn\*sqrt(1-zita^2)];

p1 = polosDeseados(1);

p2 = polosDeseados(2);

%-------------Sistema Sin Compensar-------------%

DisAng = 360/(2\*pi\*Rrueda);

Potenciometro=10/(pi\*3500/180);

I = tf([1],[1 0]); %Integrador

Gmotor = tf([Ki],[L\*Jt R\*Jt+Bm\*L R\*Bm+Ki\*Kb]); %FT del motor

Gaux = Gmotor \* I \* RelacionEngranajes; %G

%SensorLaser=tf([10/5],[0.002 1]);

Sensor = Potenciometro \* Gb; %H

FTLA = Gaux\*Sensor;

FTLC = DisAng\*Ga\*feedback(Gaux, Sensor)\*Rrueda;

F=DisAng\*Ga\*feedback(Gaux\*Gain, Sensor)\*Rrueda;

%-------------Bisectriz-------------%

[z,p,k]=zpkdata(FTLA,'v');

%aporte de fase de los polos al punto de diseño

angp1=180-atand(abs(imag(p(1))-imag(p1))/abs(real(p1)-real(p(1))));

angp2=atand(abs(imag(p(2))-imag(p1))/abs(real(p1)-real(p(2))));

angp3=atand(abs(imag(p(3))-imag(p1))/abs(real(p1)-real(p(3))));

%angulo del compensador

angComp=-180+angp1+angp2+angp3;

%compensador

angbisectriz = 180-acosd(zita);

augaux1=180-(angbisectriz/2-angComp/2)-acosd(zita);

augaux2=180-(angbisectriz/2+angComp/2)-acosd(zita);

cero=-sind(angbisectriz/2-angComp/2)\*abs(p1)/sind(augaux1);

polo=-sind(angbisectriz/2+angComp/2)\*abs(p1)/sind(augaux2);

C=zpk(cero,polo,1);

%ganacia del compensador

modp1=abs(p1-p(1));

modp2=abs(p1-p(2));

modp3=abs(p1-p(3));

modp=abs(p1-polo);

modz=abs(p1-cero);

K=(modp1\*modp2\*modp3\*modp)/(modz\*k);

K=1.3;

Comp = K\*C;

%-------------respuesta en frecuencia-------------

FTLC\_Compensada = DisAng\*Ga\*feedback(Gaux\*Comp, Sensor)\*Rrueda;

display('Sistema sin Compensar')

stepinfo(FTLC)

%Tabla Routh-Hurwitz

EcuacionCaracteristica = 1+Gmotor \* tf(1, [1, 0]) \* RelacionEngranajes\*Potenciometro\*Gb;

x = sym ('x');

A=L\*Jt;

B=R\*Jt+Bm\*L;

C=Ki\*Kb+R\*Bm;

RH(1).Col1=A;

RH(1).Col2=C;

RH(2).Col1=B;

RH(2).Col2=(Sensor\*Ki\*RelacionEngranajes); %\*k

RH(3).Col1= ((B\*C)-(A\*Sensor\*Ki\*RelacionEngranajes\*x))/B; %\*k

RH(3).Col2=0;

RH(4).Col1=Sensor\*Ki\*RelacionEngranajes; %\*k

LimiteSuperior=(B\*C)/(A\*Sensor\*Ki\*RelacionEngranajes);

FCritica=DisAng\*Ga\*feedback(Gaux\*LimiteSuperior, Sensor)\*Rrueda;

%Imprimir Tabla

lalala=1;

while(lalala<4)

RH(lalala).Col1;

RH(lalala).Col2;

lalala=lalala+1;

end

display('Sistema Compensado por el Metodo de Lugar de Raices (con ganancia ajustada)')

stepinfo(FTLC\_Compensada)

%%%%Compensacion por variables de estado%%%%%

display('Variables de estado')

[n, d] = tfdata(FTLC, 'v'); %Coeficientes del numerador y el denominador de la FTLC

[E1, E2, E3, E4] = tf2ss(n, d); %Obtengo el sistema en espacio de estados

%Determino la Observabilidad del sistema

Observabilidad = [E3; E3\*E1; E3\*(E1^2)];

ifdet(Observabilidad)~=0

display('El sistema es observable.')

Observable=true;

else

display('El sistema no es observable.')

Observable=false;

end

%Determino la Controlabilidad del sistema

Controlabilidad = [E2 E1\*E2 (E1^2)\*E2];

ifdet(Controlabilidad)~=0

display('El sistema es controlable.');

Controlable=true;

else

display('Cuidado! El sistema no es controlable.')

Controlable=false;

end

if(Controlable && Observable) %Si el sistema es controlable, sera compensado por variables de estado.

alpha=5;

polosExtras= -(alpha\*wn);

%Agregamos un polo lo suficientemente

%alejados para que no interfieran con la

%respuesta del sistema.

vectorK = place(E1, E2, [transpose(polosDeseados), polosExtras]); %Creo el vector k para el calculo del compensador

Sistema\_Compensado\_Variables\_Estado=ss(E1-E2\*vectorK, E2, E3, E4); %Creo el sistema compensado en variables de estado

[n\_compensado, d\_compensado] = ss2tf(E1-E2\*vectorK, E2, E3, E4); %Lo transformo a TF

FT\_Compensado\_Variables\_Estado = tf(n\_compensado, d\_compensado);

display('Sistema compensado por Variables de Estado')

stepinfo(FT\_Compensado\_Variables\_Estado)

end

# Hoja de Datos del Motor:

