

Sistema de Control Automático.

**Práctica 2: Control por Retroalimentación de Estados para Motor de CD**

Luis Fernando Rodriguez Gutierrez ie705694

Profesor:

Dr. Luis Enrique Gonzalez Jimenez

29/03/20

INDICE

Introducción…………………………………………………………………2

Objetivo….…………….……………………………………………………4

Desarrollo práctico …………………………………………………………5

Conclusión.…………………………………………………………………15

Código Matlab LQR...………………………………………………………16

Código Matlab LQR...………………………………………………………19

1. Introducción

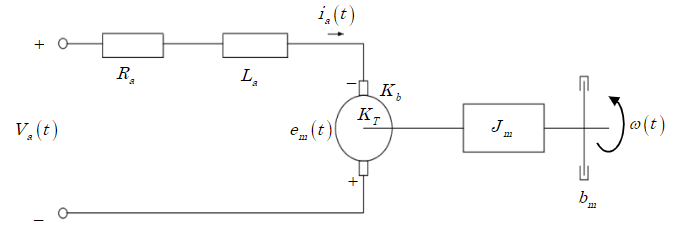
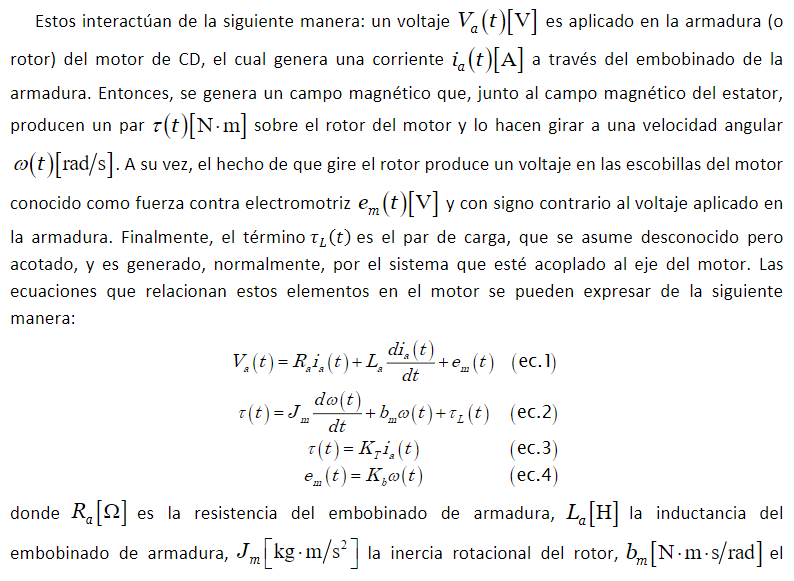
La siguiente figura muestra un diagrama de los elementos eléctricos y mecánicos que componen a un motor de CD. 

Ilustración 1. Elementos de un motor de CD.



Coeficiente de fricción entre el rotor y el estator, constante de par del motor y la constante de tensión.

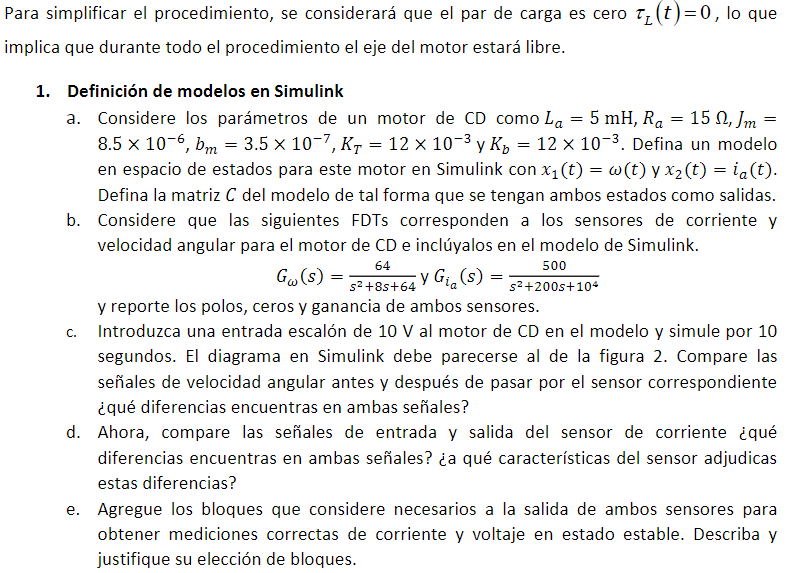
1. Objetivo

Simular modelos en espacio de estados en lazo abierto usando SIMULINK.

1. Material

* PC
* Matlab
* Simulink

1. Procedimiento



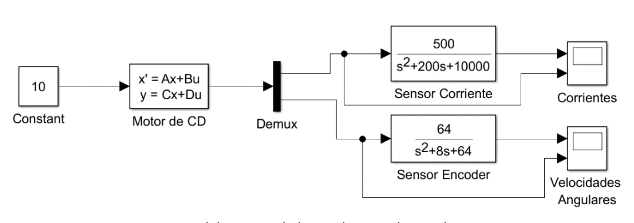
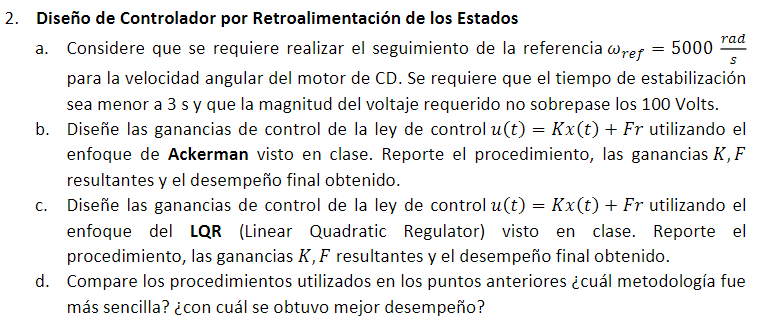


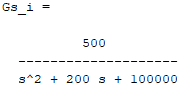
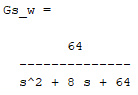
Ilustración 2. Modelo en Simulink con el motor de CD y los sensores.



1. Desarrollo práctico.

DEFINICION DE MODELOS EN SIMULINK

En primera instancia la diferencia a él como se desarrolló esta práctica, lo cual no afecta el resultado, sino que solamente la posicion de los valores en las matrices. Es que se tomó como . De manera a que las ecuaciones varían un poco. Pero como primera parte lo que se hizo fue calcular los polos, esto se hizo por medio de la función poles de Matlab y así mismo comprobando matemáticamente. Para cerciorar que los valores fueran correctos. De manera que se obtuvieron los siguientes polos.



Ceros;

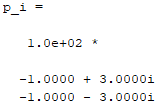
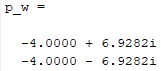
Gs\_w = 0; Gs\_i = 0;

Polos;

Gs\_w = -4 +- 6.9282i Gs\_i = -100 +- 300i

Ganancias;

Ks\_w = 1; Ks\_i = 5;



Lo siguiente que se hizo fue diseñar el sistema en simulink, para lograr analizar los efectos en el sistema de los sensores agregados. Se planteo desde en la practica que fuera en simulink, dado que integrar los sensores al sistema seria mas tardado y complicado que en simulink.

Por lo que obtenemos el siguiente sistema.

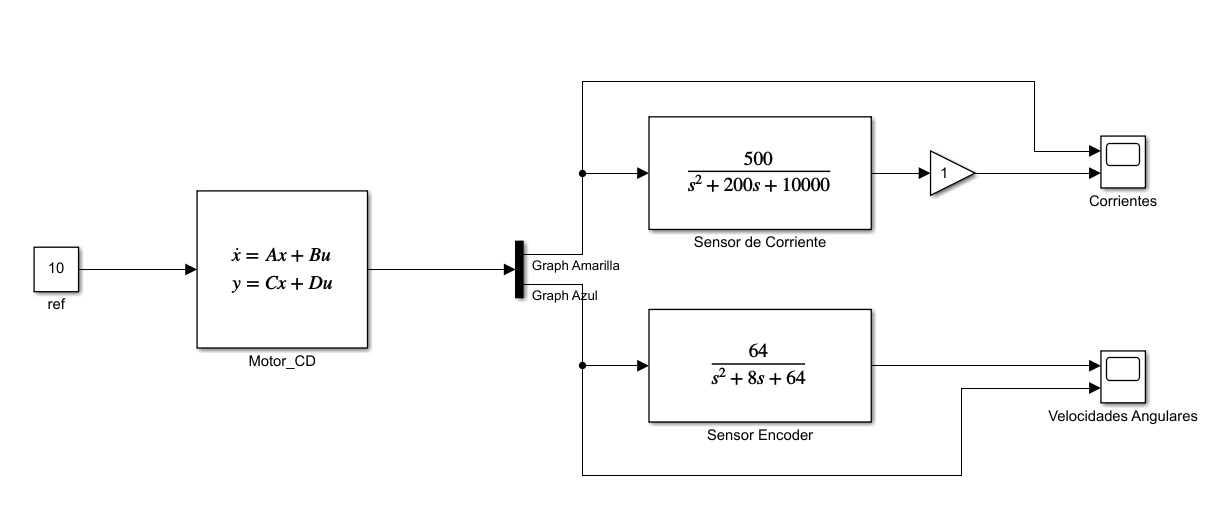


Ilustración 3. Sistema Simulink sin ganancia.

Las conexiones que se logran apreciar antes y después de los sensores con para poder analizar el comportamiento del sistema antes y después, de manera que se obtienen las siguientes graficas.

* Corrientes

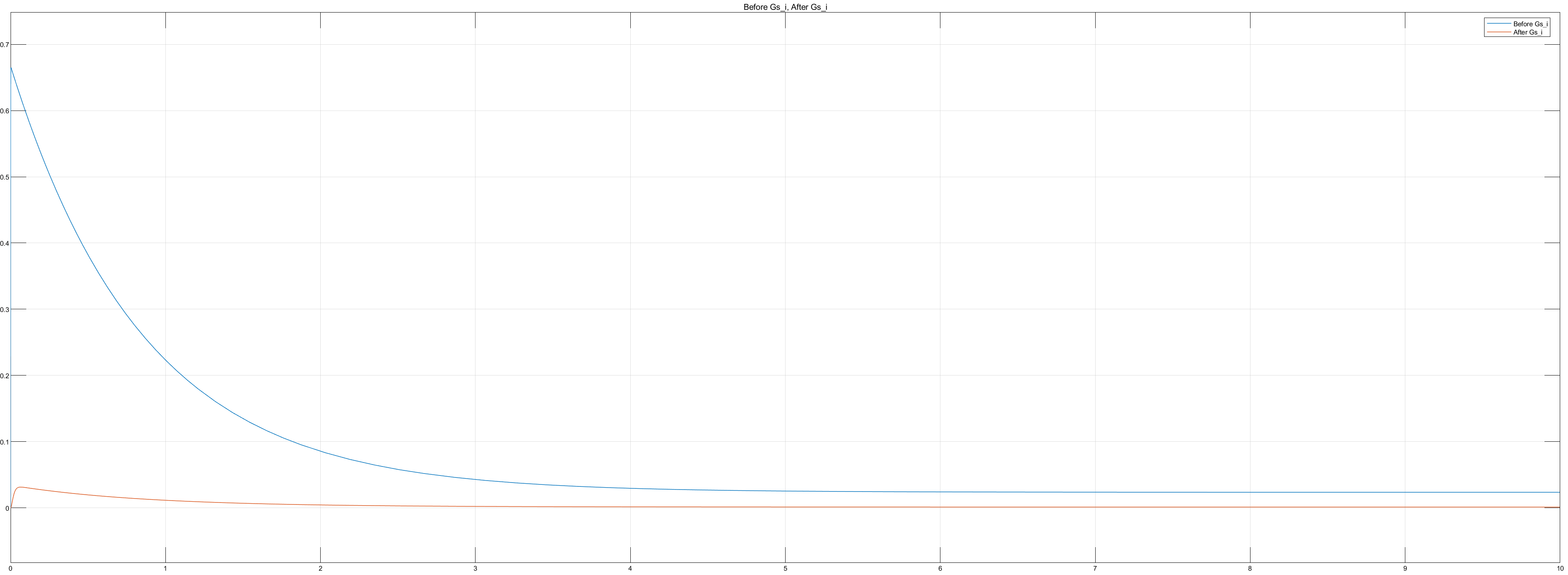


Ilustración 4. Grafica de la respuesta del sensor de corriente

La grafica que se aprecia es la respuesta en cuanto al sensor de corriente. La parte más baja es después del sensor, mientras que la de mayor ganancia es antes del sensor. Por lo que se puede apreciar que dicho sensor tiene un efecto de atenuar la respuesta, de manera que se requiere una amplificación. La cual se muestra en la siguiente figura.

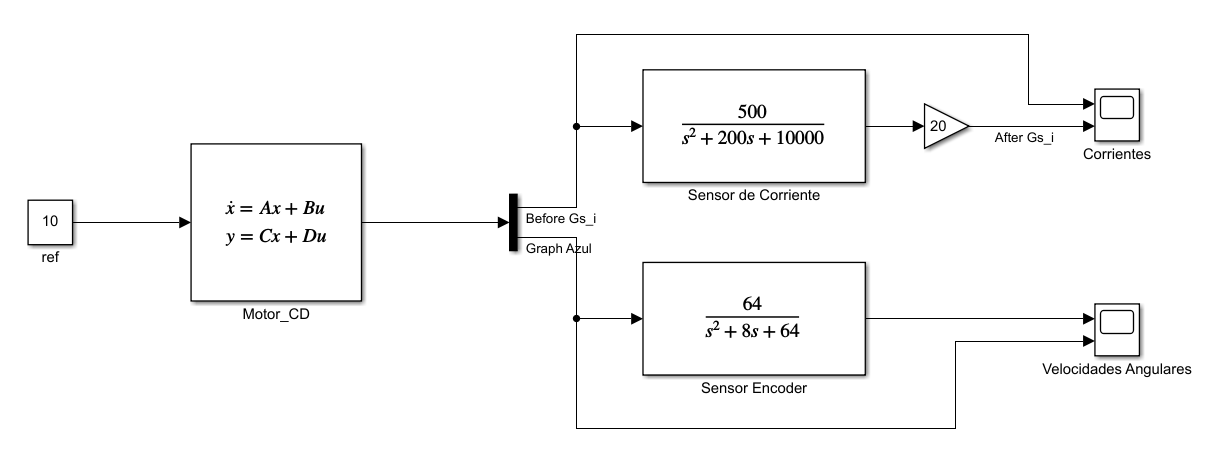


Ilustración 5. Sistema simulink con ganancia.

En este sistema lo que se hizo fue ir ajustando las ganancias por medio de prueba y error para hacer que la respuesta después del sensor se aproximara en lo que se pudiera a la señal original. De manera que tanto el máximo sobre impulso y, su tiempo así como su magnitud de estabilización sean semejantes.

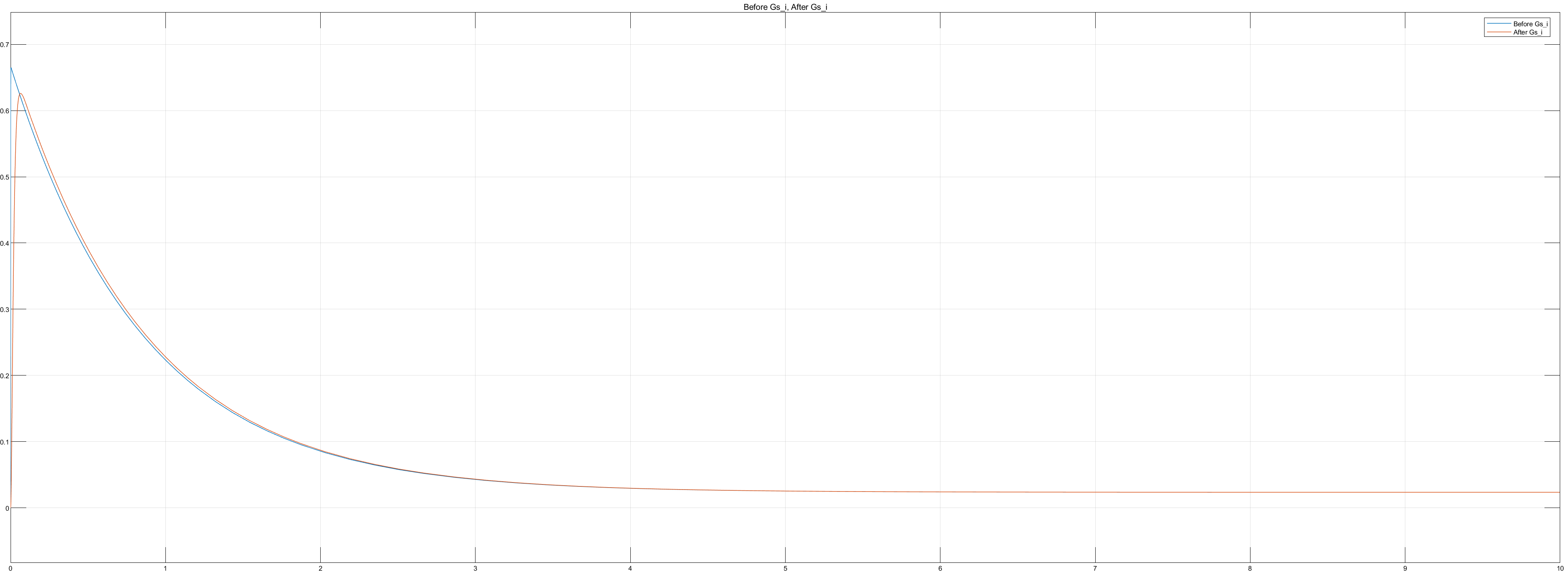


Ilustración 6.Grafica de la respuesta del sensor de corriente

* Encoder

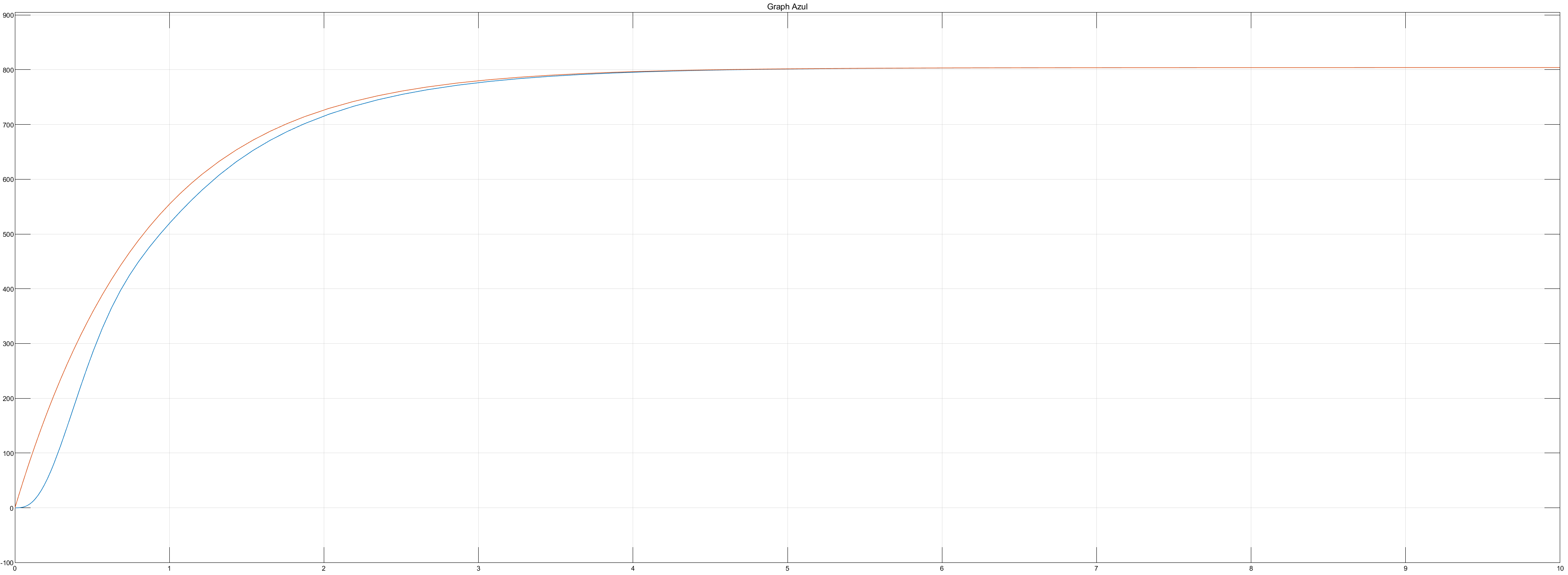


Ilustración 7. Grafica de la respuesta del encoder

La grafica que tenemos es la respuesta al sensor correspondiente al encoder. De manera a como se visualiza, este sensor no fue necesario llegar a hacerle alguna modificación. Dado que este mismo logra el punto de estabilización aproximadamente al mismo momento que la señal original. El buscar optimizar dicha señal solo seria un gasto extra si fuera llevado a la vida real.

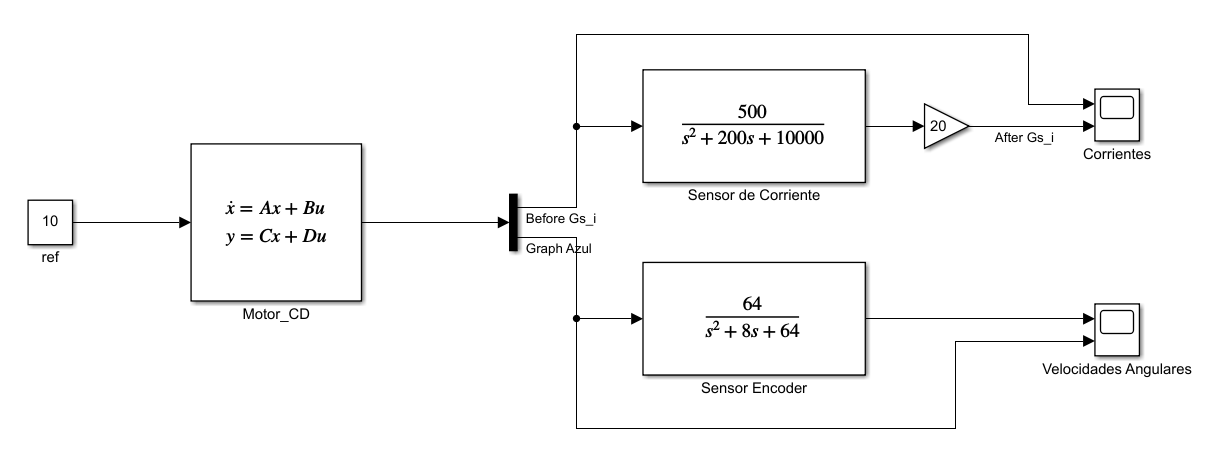


Ilustración8. Sistema simulink con ganancia.

DISEÑO DE CONTROLADOR POR RETROALIMENTACION DE LOS ESTADOS.

* ACKERMAN

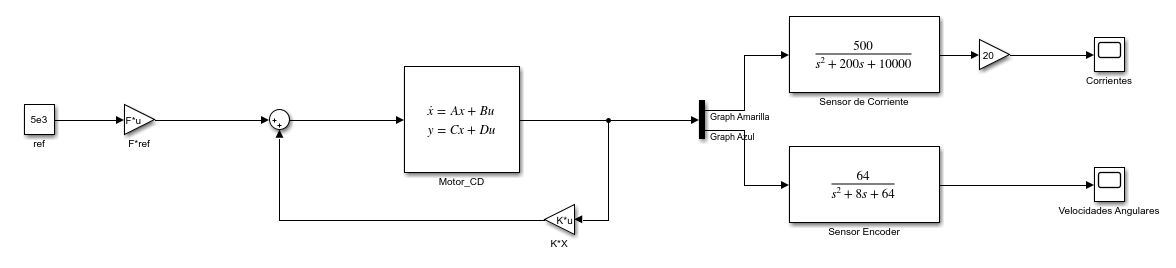
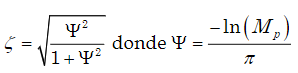
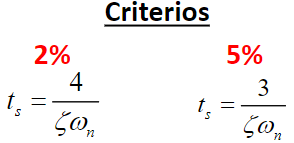


Ilustración 7. Sistema simulink ACKERMAN

De manera a como se pide en el ejercicio 2, lo que se busco fue que ahora el sistema lograra una referencia de . De manera que buscamos que el encoder logre registrar dicha pero en un lapso menor a 3 seg. Mientras se busca que el máximo sobre impulso no sobrepase los 100V. Donde buscamos los parámetros tengan los siguientes polos.

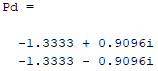


Implementando el siguiente sistemas de ecuaciones, obtenemos el valor que nombraremos dseda y así mismo buscamos la Wn usando el criterio del 2%.



= 1.6140

De manera que ahora obtenemos los siguientes polos del sistema para ingresar al sistema ackerman.



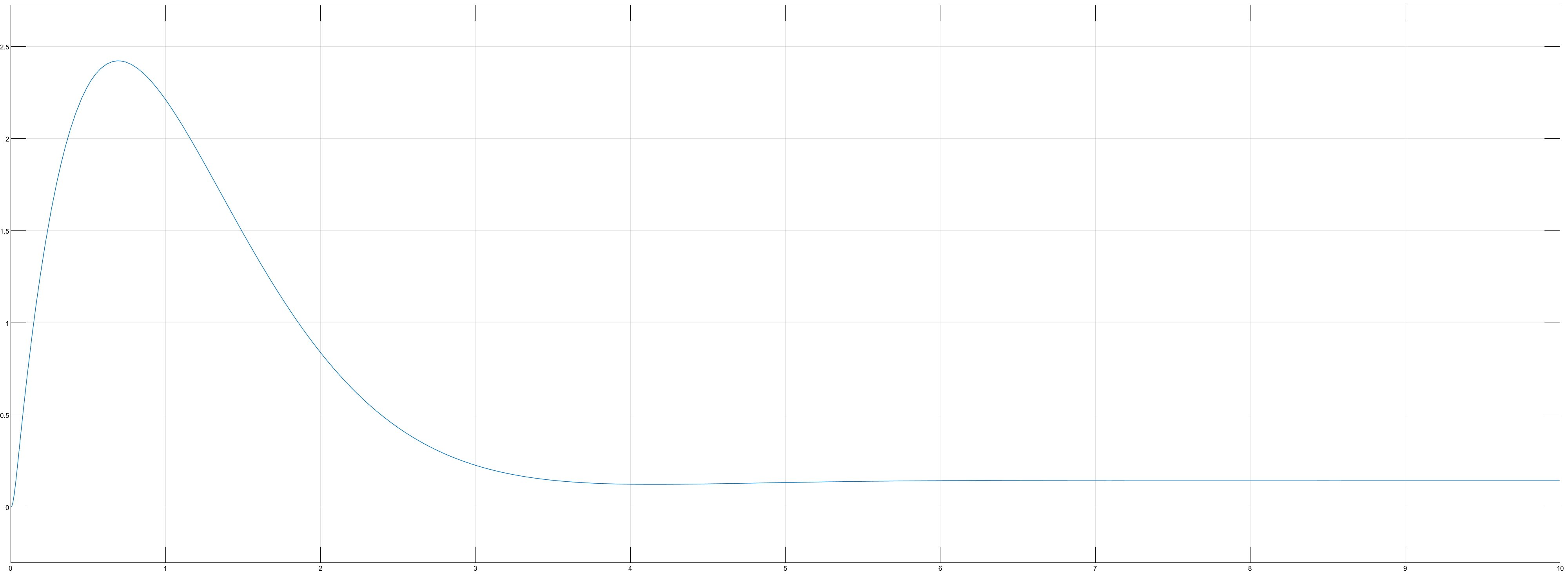


Ilustración 8. Respuesta del sensor de corriente, Ackerman

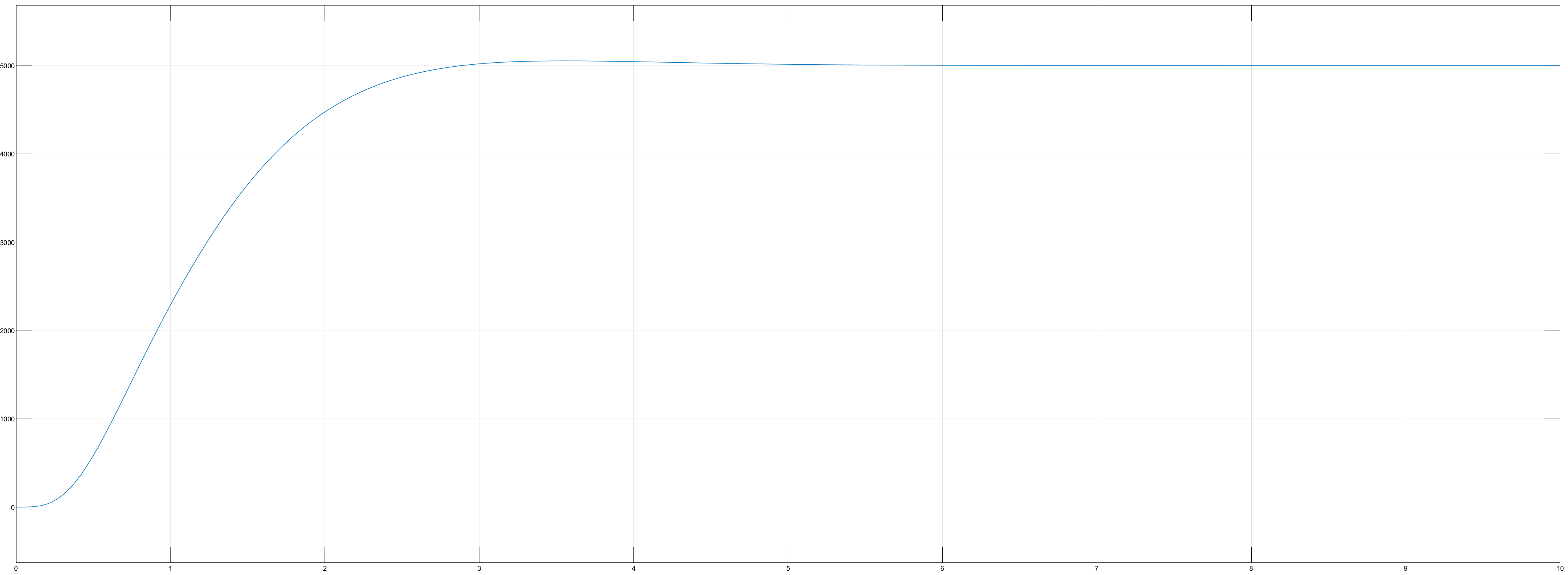


Ilustración 9. Respuesta del encoder, Ackerman

De manera a como se observa, solo por comprobación que se diseño correctamente en simulink, a su vez se hizo la simulación por medio de Matlab, pero sin los sensores.

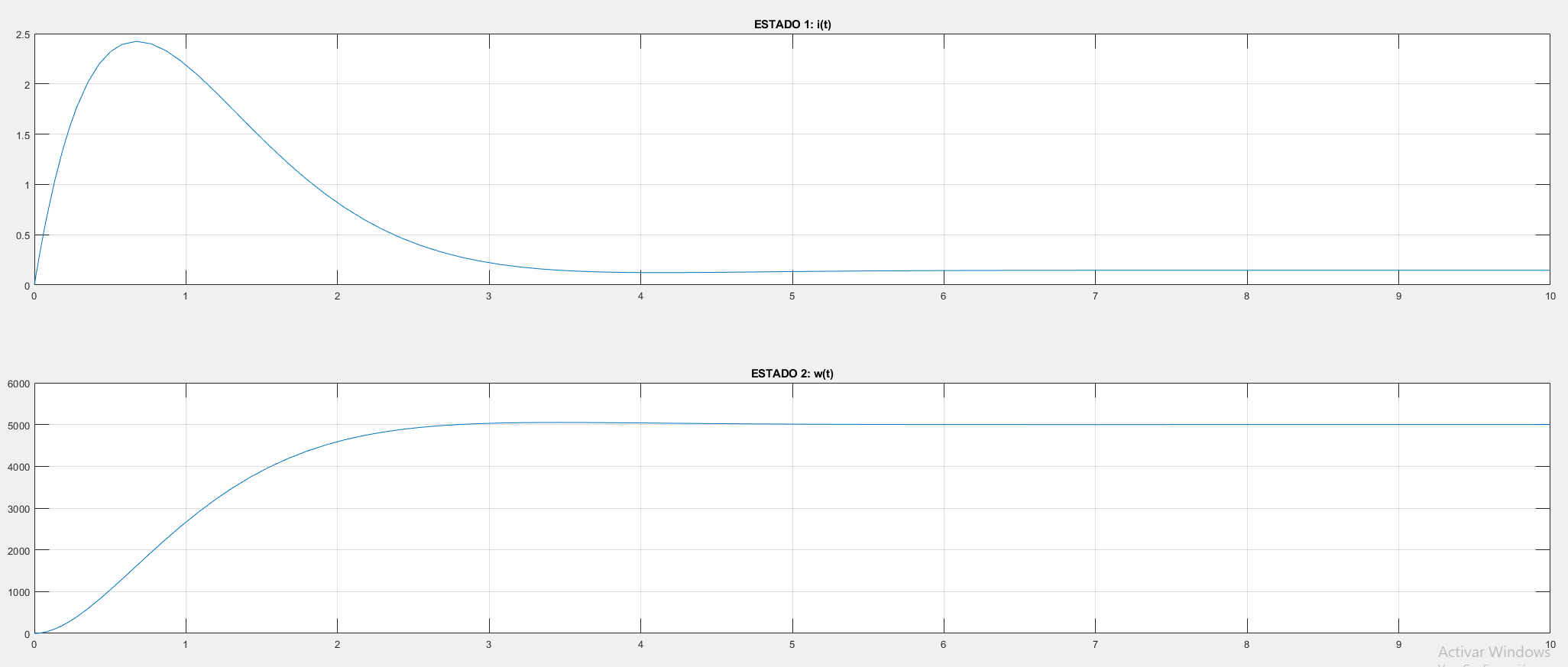


Ilustración 10. Simulacion en matlab, Ackerman

* LQR

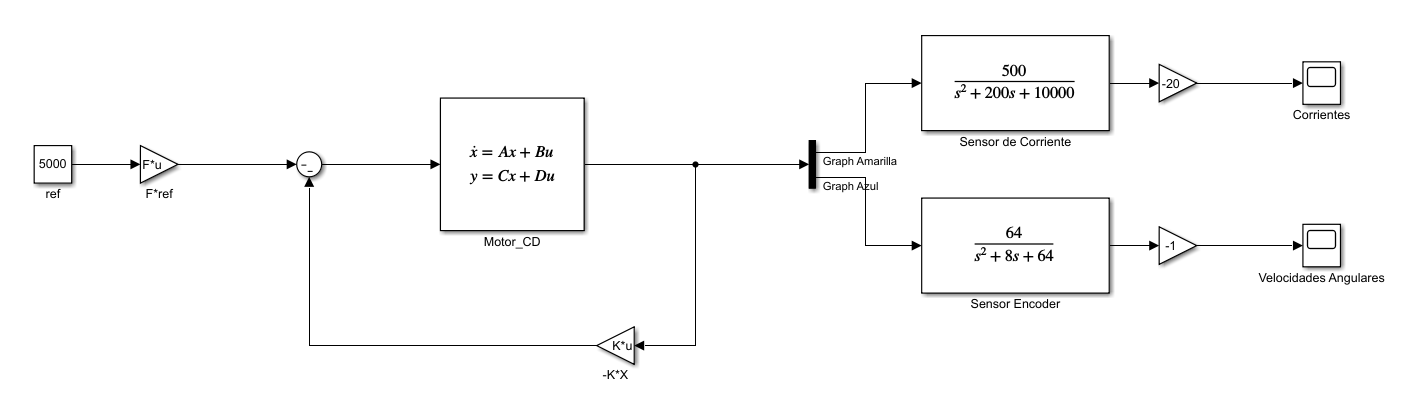


Ilustración 11. Sistema simulink, LQR

De manera a como se pide en el ejercicio 2, lo que se busco fue que ahora el sistema lograra una referencia de . De manera que buscamos que el encoder logre registrar dicha pero en un lapso menor a 3 seg. Mientras se busca que el máximo sobre impulso no sobrepase los 100V. Donde buscamos los parámetros tengan los siguientes polos.

La gran diferencia a lo que es ackerman, en el como lo desarrolle fue que estos valores (Q y R) no se calculan como tal, sino que son por medio del dichoso tanteo metro. Pero no a ese punto, sino, que analizamos que es lo que afecta Q y que es lo que afecta R. De manera que tomo la siguiente imagen como base para un mayor entendimiento de como afecta R y Q al sistema.

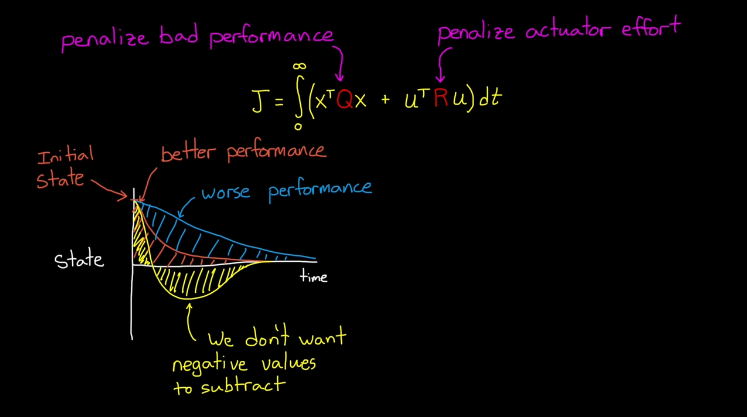
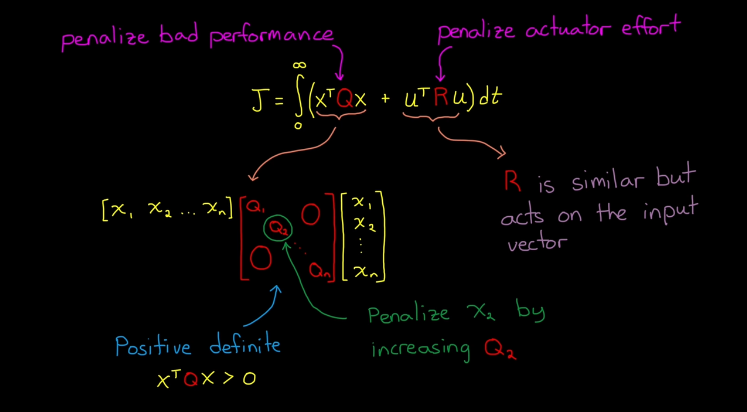


Ilustración . Explicacion sacada de matlab

Ilustración 13. Explicacion sacada de matlab

En cuanto ya la experimentación, primero se empezó con unas matrices base, para de ese punto ir tanteando terreno para saber hacia donde mover los valores de Q y R.

Obtenemos las siguientes respuestas.

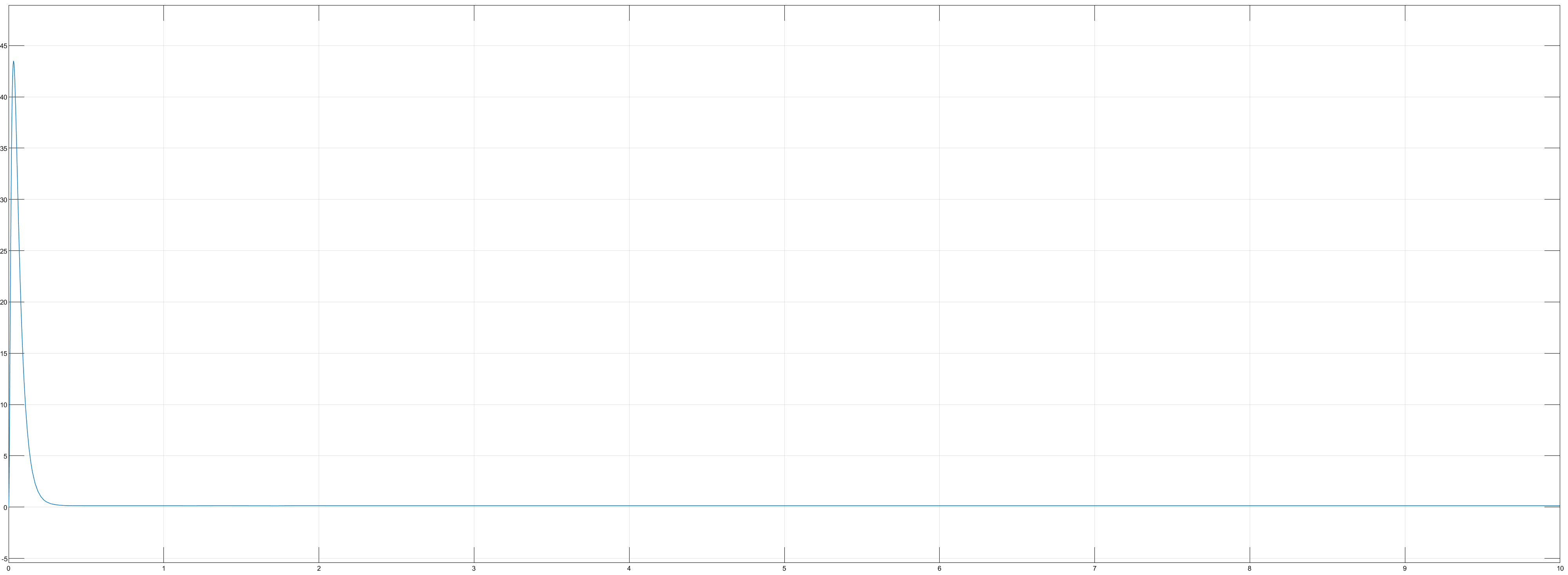


Ilustración 14.Respuesta del sensor de corriente, LQR

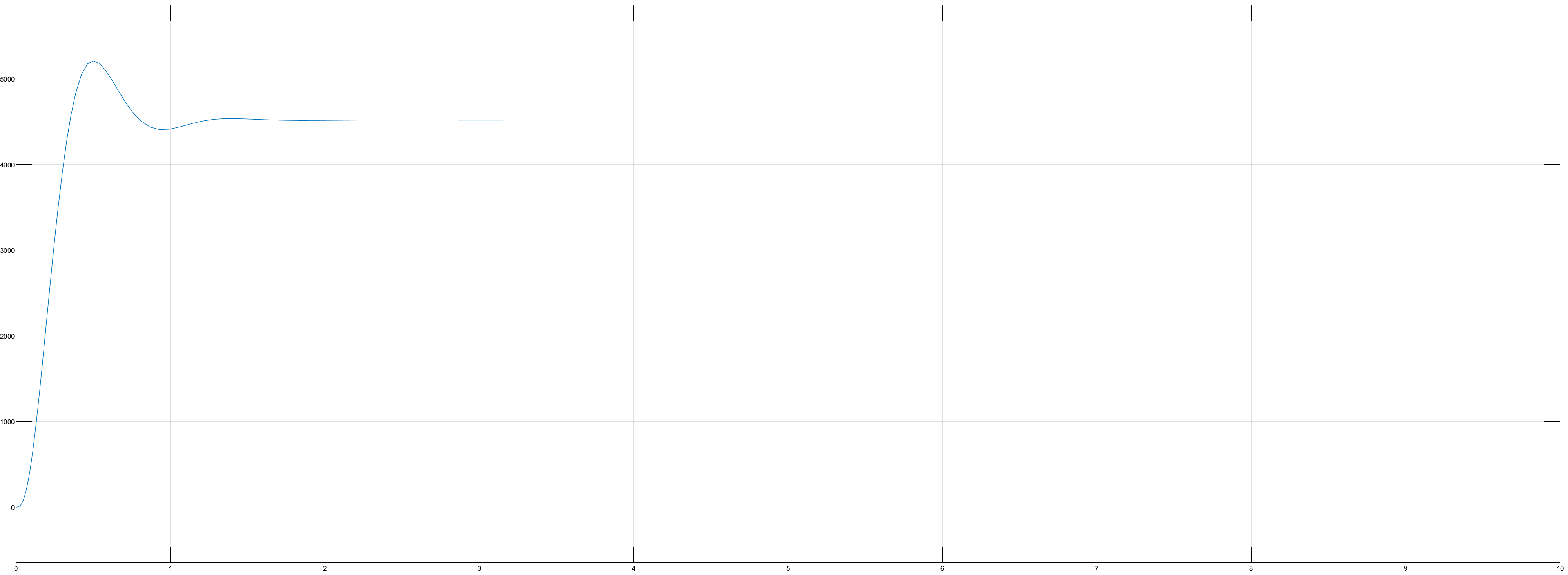


Ilustración 15.Respuesta del encoder, LQR

De manera a que, a como están las matrices inicialmente, primero, no tenemos suficiente energía en el actuador (motor) para llegar a la referencia que buscamos, si logra el tiempo de estabilización. Pero a un costo demasiado grande de energía. Por lo que ahora movemos los polos entendiendo un poco mas que variable acentuar, o que variable castigar del sistema. Por ejemplo, aquí ahora nos fuimos a otro extremo en donde Q = 500 y R = 0.01. El valor obtenido del encoder no aumenta de 5000, por lo que funciona dicho controlador, en donde busca mantener su valor máximo. Mientras que este ya empieza a estabilizarse en el valor deseado con Q = 80, esto incluyendo si estabilización siendo menor a 3 segundos.

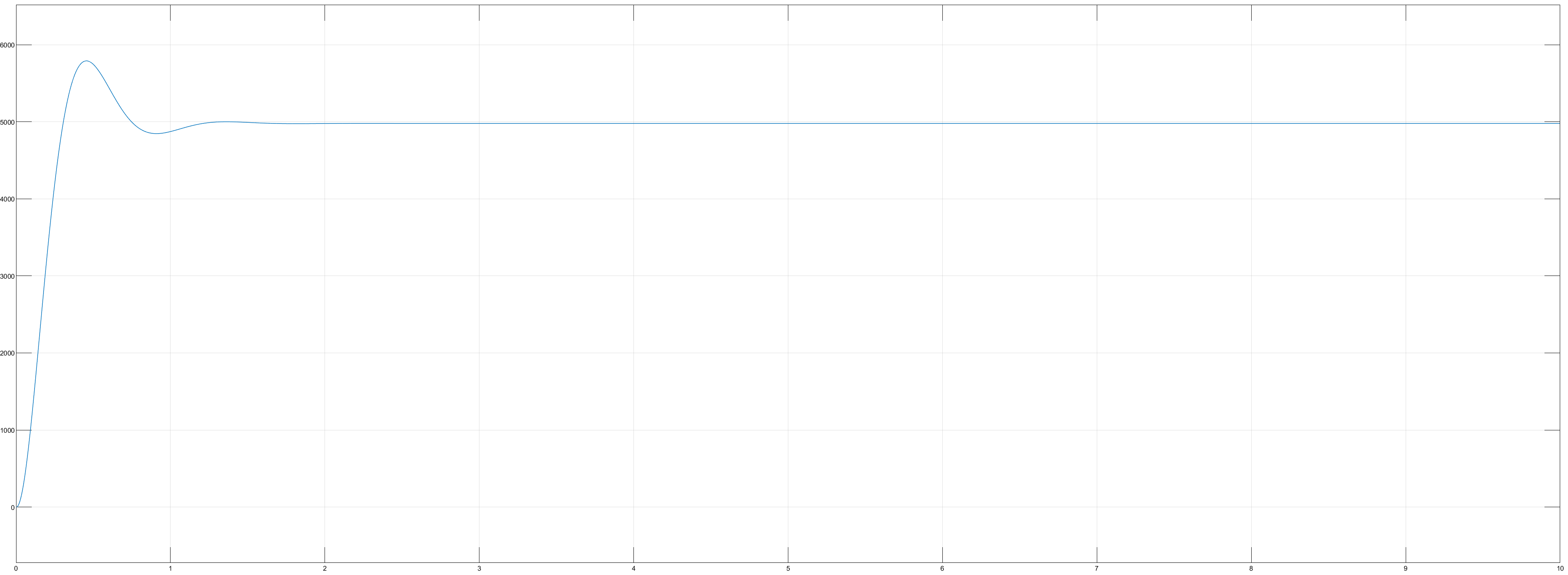
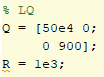


Ilustración 16.Respuesta del encoder, LQR

Teniendo dicha ventaja que la estabilización es sumamente corta en cuanto al tiempo, pero tomando un gran costo a la energía consumida para que logre esto.

Se logro la respuesta buscada, aunque hace falta una mayor practica y mejor análisis de LQR para poder modificar los valores con mayor precisión. Dado que esto fue mucho por medio de prueba y error. Pero al final las Q y R obtenidos que logran dicha funcionalidad fueron:



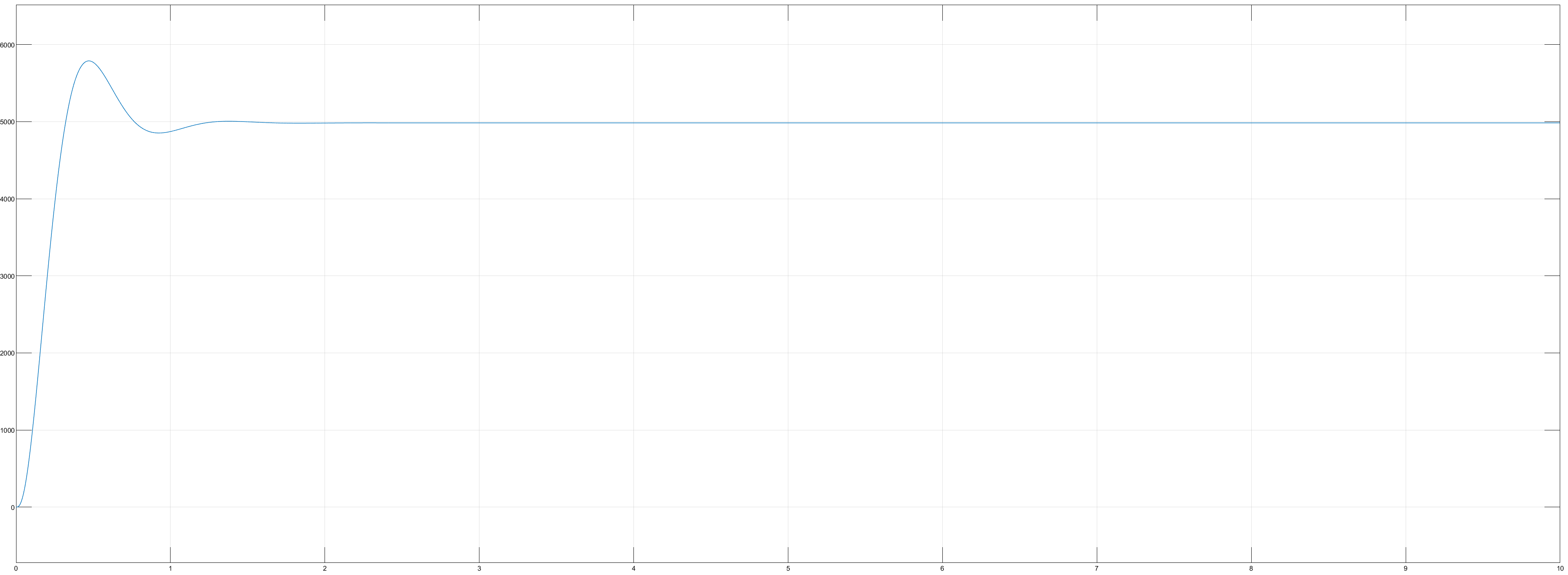


Ilustración 17.Respuesta del encoder, LQR

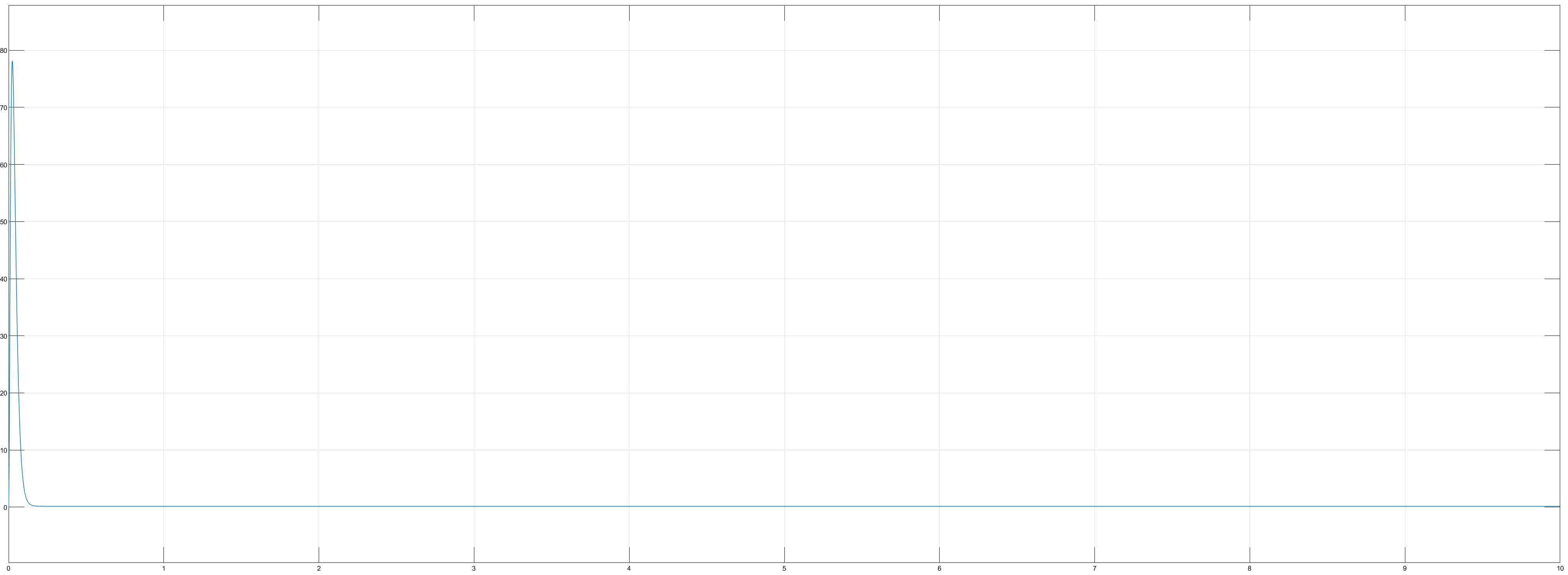


Ilustración 18.Respuesta del sensor de corriente, LQR

CONCLUSION

En lo que respecta de mi punto de vista, preferí el método de ackerman por facilidad y mayor visualización que se obtiene por medio de los cálculos. De misma manera no descarto el poder que tiene LQR. Sin embargo, siento que hace falta mayor análisis, y no me refiero en cuanto a la teoría, sino, al análisis por medio de Matlab que este puede facilitar a ubicación de polos. Ya que con un análisis mas profundo LQR puede ser algo que por medio de dos o tres intentos se logra lo que se busca. Siempre y cuando se haga de la manera correcta.

Código Matlab Ackerman:

close all

clear

clc

% Vector de tiempo para espacio de estados

tspan = [0 10];

% Condiciones iniciales

x0 = [0, 0];

%Polos deseados

Ks = 13.668e3;

sig = -log(100)/pi;

amort = sqrt( (sig.\*sig)/(1+(sig.\*sig)) );

wn = 4/(3\*amort);

G\_s = tf([Ks\*wn\*wn],[1 amort\*wn wn\*wn]);

Pd1 = -amort\*wn + wn\*sqrt((amort.\*amort)-1);

Pd2 = -amort\*wn - wn\*sqrt((amort.\*amort)-1);

Pd = [Pd1;Pd2];

% Llamado de estados para simulink

R = 15;

L = 5\*(10^-3);

Jm = 8.5\*(10^-6);

B = 3.5\*(10^-7);

Kt = 12\*(10^-3);

Kb = 12\*(10^-3);

a = R/L;

b = Kb/L;

c = Kt/Jm;

d = B/Jm;

e = 1/L;

A = [-a, -b;c, -d];

B = [e; 0];

C = [0, 1];

D = 0;

Mc = [B A\*B];

H = ( A - Pd(1)\*eye(2) )\*( A - Pd(2)\*eye(2) );

K = -[0 1]\*inv(Mc)\*H;

F = 1 / ( C\*inv( -A-B\*K ) \* B );

%

motor\_cd\_plot(tspan,x0,Pd);

%% Functions

function motor\_cd\_plot(tspan, x0, Pd)

global A B C D K F

R = 15;

L = 5\*(10^-3);

Jm = 8.5\*(10^-6);

B = 3.5\*(10^-7);

Kt = 12\*(10^-3);

Kb = 12\*(10^-3);

a = R/L;

b = Kb/L;

c = Kt/Jm;

d = B/Jm;

e = 1/L;

A = [-a, -b;c, -d];

B = [e; 0];

C = [0, 1];

D = 0;

Mc = [B A\*B];

H = ( A - Pd(1)\*eye(2) )\*( A - Pd(2)\*eye(2) );

K = -[0 1]\*inv(Mc)\*H;

F = 1 / ( C\*inv( -A-B\*K ) \* B );

[t,X] = ode45(@motor\_cd\_sys,tspan, x0);

ref = 5000; %rad/s

U = X\*K' + F\*ref;

maxU = max( abs(U) )

figure;

subplot(2,1,1); plot(t, X(:,1)); title('ESTADO 1: i(t)'); grid;

subplot(2,1,2); plot(t, X(:,2)); title('ESTADO 2: w(t)'); grid;

end

function dX = motor\_cd\_sys (t,X)

global A B C D K F

ref = 5000; %rad/s

U = K\*X + F\*ref;

dX = A\*X + B\*U;

end

Código Matlab LQR:

close all

clear

clc

% Vector de tiempo para espacio de estados

tspan = [0 10];

% Condiciones iniciales

x0 = [0, 0];

% LQ

Q = [50e4 0;

0 900];

R = 1e3;

R = 15;

L = 5\*(10^-3);

Jm = 8.5\*(10^-6);

B = 3.5\*(10^-7);

Kt = 12\*(10^-3);

Kb = 12\*(10^-3);

a = R/L;

b = Kb/L;

c = Kt/Jm;

d = B/Jm;

e = 1/L;

A = [-a, -b;c, -d];

B = [e; 0];

C = [0, 1];

Mc = [B A\*B];

K = lqr(A,B,Q,R);

F = 1 / ( C\*inv( -A-B\*K ) \* B );

%LQR\_motor\_cd\_plot(tspan, x0, Q, R)

%% Functions

function LQR\_motor\_cd\_plot(tspan, x0, Q, R)

%Q = tiempo

%R = Amplitud

global A B C K F ref

R = 15;

L = 5\*(10^-3);

Jm = 8.5\*(10^-6);

B = 3.5\*(10^-7);

Kt = 12\*(10^-3);

Kb = 12\*(10^-3);

a = R/L;

b = Kb/L;

c = Kt/Jm;

d = B/Jm;

e = 1/L;

A = [-a, -b;c, -d];

B = [e; 0];

C = [0, 1];

Mc = [B A\*B];

K = lqr(A,B,Q,R);

F = 1 / ( C\*inv( -A-B\*K ) \* B );

ref = 5000; %rad/s

eigs(A-B\*K) % Comprobamos ubicacion de los polos en lazo cerrado

[t,X] = ode45(@LQR\_motor\_cd\_sys,tspan, x0);

figure;

subplot(2,1,1); plot(t, X(:,1)); title('ESTADO 1: i(t)'); grid;

subplot(2,1,2); plot(t, X(:,2)); title('ESTADO 2: w(t)'); grid;

figure;

plot(t, -K\*X'); title('SEÑAL DE CONTROL'); grid on;

end

function dX = LQR\_motor\_cd\_sys (t,X)

global A B C K F ref

%U = -K\*X;

U = -K\*X-F\*ref;

dX = A\*X + B\*U;

end