



**ITESO**  
Universidad Jesuita  
de Guadalajara

Sistema de Control Automático.

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

Luis Fernando Rodriguez Gutierrez

ie705694

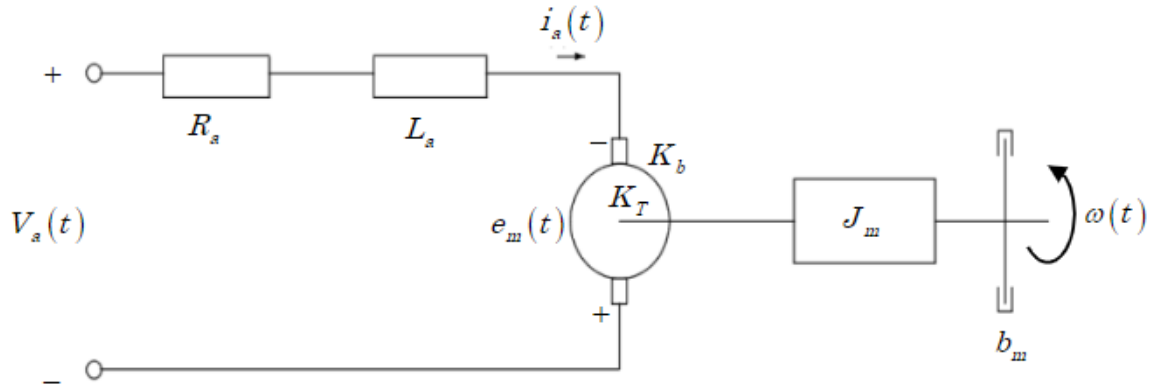
Profesor:

Dr. Luis Enrique Gonzalez Jimenez

29/03/20

## A. 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.*

Estos interactúan de la siguiente manera: un voltaje  $V_a(t)[V]$  es aplicado en la armadura (o rotor) del motor de CD, el cual genera una corriente  $i_a(t)[A]$  a través del embobinado de la armadura. Entonces, se genera un campo magnético que, junto al campo magnético del estator, producen un par  $\tau(t)[N \cdot m]$  sobre el rotor del motor y lo hacen girar a una velocidad angular  $\omega(t)[rad/s]$ . A su vez, el hecho de que gire el rotor produce un voltaje en las escobillas del motor conocido como fuerza contra electromotriz  $e_m(t)[V]$  y con signo contrario al voltaje aplicado en la armadura. Finalmente, el término  $\tau_L(t)$  es el par de carga, que se asume desconocido pero acotado, y es generado, normalmente, por el sistema que esté acoplado al eje del motor. Las ecuaciones que relacionan estos elementos en el motor se pueden expresar de la siguiente manera:

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_m(t) \quad (\text{ec.1})$$

$$\tau(t) = J_m \frac{d\omega(t)}{dt} + b_m \omega(t) + \tau_L(t) \quad (\text{ec.2})$$

$$\tau(t) = K_T i_a(t) \quad (\text{ec.3})$$

$$e_m(t) = K_b \omega(t) \quad (\text{ec.4})$$

donde  $R_a[\Omega]$  es la resistencia del embobinado de armadura,  $L_a[H]$  la inductancia del embobinado de armadura,  $J_m[kg \cdot m/s^2]$  la inercia rotacional del rotor,  $b_m[N \cdot m \cdot s/rad]$  el

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

## B. Objetivo

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

## C. Material

- PC
- Matlab
- Simulink

## D. Procedimiento

Para simplificar el procedimiento, se considerará que el par de carga es cero  $\tau_L(t) = 0$ , lo que implica que durante todo el procedimiento el eje del motor estará libre.

### 1. Definición de modelos en Simulink

- Considere los parámetros de un motor de CD como  $L_a = 5 \text{ mH}$ ,  $R_a = 15 \Omega$ ,  $J_m = 8.5 \times 10^{-6}$ ,  $b_m = 3.5 \times 10^{-7}$ ,  $K_T = 12 \times 10^{-3}$  y  $K_b = 12 \times 10^{-3}$ . Defina un modelo en espacio de estados para este motor en Simulink con  $x_1(t) = \omega(t)$  y  $x_2(t) = i_a(t)$ . Defina la matriz  $C$  del modelo de tal forma que se tengan ambos estados como salidas.
- Considere que las siguientes FDTs corresponden a los sensores de corriente y velocidad angular para el motor de CD e inclúyalos en el modelo de Simulink.

$$G_\omega(s) = \frac{64}{s^2 + 8s + 64} \text{ y } G_{i_a}(s) = \frac{500}{s^2 + 200s + 10^4}$$

y reporte los polos, ceros y ganancia de ambos sensores.

- Introduzca una entrada escalón de 10 V al motor de CD en el modelo y simule por 10 segundos. El diagrama en Simulink debe parecerse al de la figura 2. Compare las señales de velocidad angular antes y después de pasar por el sensor correspondiente ¿qué diferencias encuentras en ambas señales?
- Ahora, compare las señales de entrada y salida del sensor de corriente ¿qué diferencias encuentras en ambas señales? ¿a qué características del sensor adjudicas estas diferencias?
- Agregue los bloques que considere necesarios a la salida de ambos sensores para obtener mediciones correctas de corriente y voltaje en estado estable. Describa y justifique su elección de bloques.

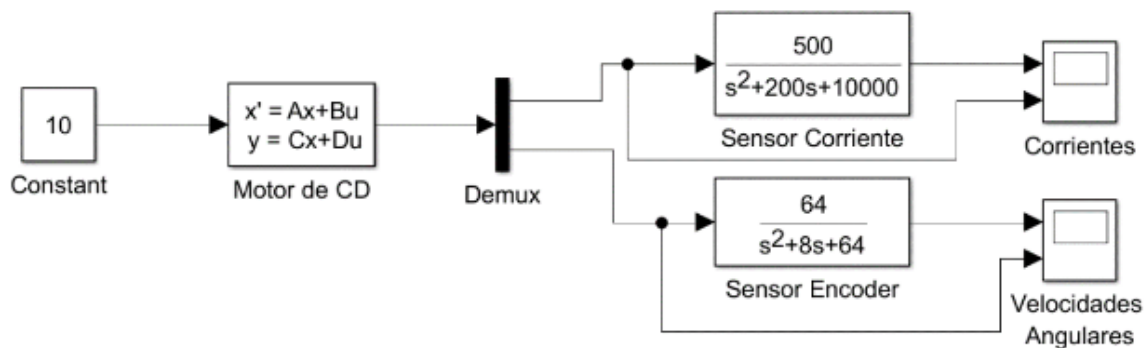


Ilustración 2. Modelo en Simulink con el motor de CD y los sensores.

## 2. Diseño de Controlador por Retroalimentación de los Estados

- Considere que se requiere realizar el seguimiento de la referencia  $\omega_{ref} = 5000 \frac{rad}{s}$  para la velocidad angular del motor de CD. Se requiere que el tiempo de estabilización sea menor a 3 s y que la magnitud del voltaje requerido no sobrepase los 100 Volts.
- Diseñe las ganancias de control de la ley de control  $u(t) = Kx(t) + Fr$  utilizando el enfoque de **Ackerman** visto en clase. Reporte el procedimiento, las ganancias  $K, F$  resultantes y el desempeño final obtenido.
- Diseñe las ganancias de control de la ley de control  $u(t) = Kx(t) + Fr$  utilizando el enfoque del **LQR** (Linear Quadratic Regulator) visto en clase. Reporte el procedimiento, las ganancias  $K, F$  resultantes y el desempeño final obtenido.
- Compare los procedimientos utilizados en los puntos anteriores ¿cuál metodología fue más sencilla? ¿con cuál se obtuvo mejor desempeño?

E. Desarrollo práctico.

### DEFINICION DE MODELOS EN SIMULINK

En primera instancia la diferencia a el 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  $x_1 = i(t)$  y  $x_2 = w(t)$ . 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.

```
Gs_w =          Gs_i =
      64          500
-----
s^2 + 8 s + 64   s^2 + 200 s + 100000

p_i =
      1.0e+02 *
-4.0000 + 6.9282i  -1.0000 + 3.0000i
-4.0000 - 6.9282i  -1.0000 - 3.0000i
```

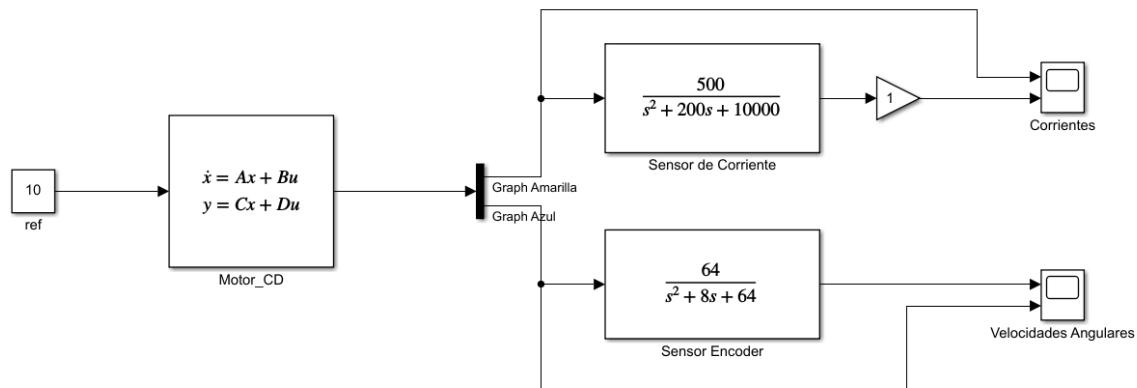
```
Ceros;
Gs_w = 0;          Gs_i = 0;

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.

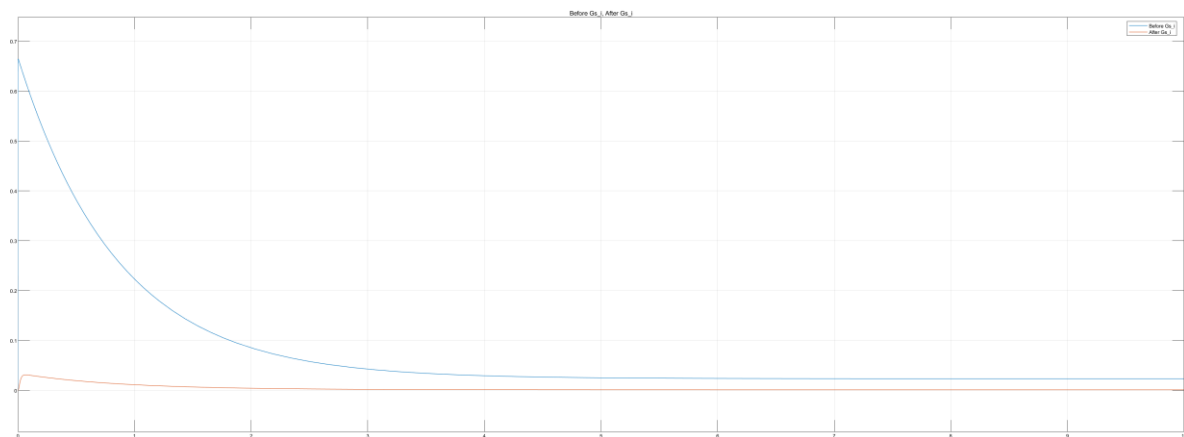
Por lo que obtenemos el siguiente sistema.



*Ilustración 3. Sistema Simulink sin ganancia.*

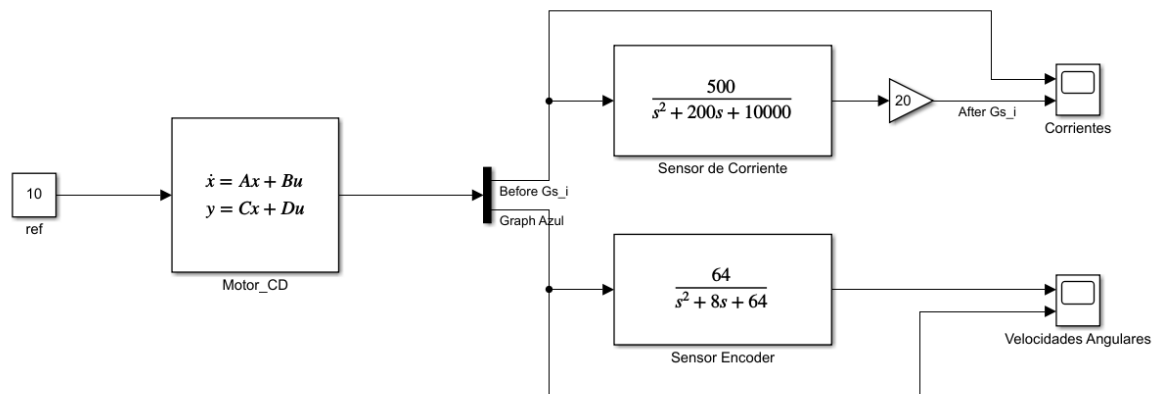
Las conexiones que se logran apreciar antes y después de los sensores con para poder analizar el comportamiento del sistema antes y después, de manera que se obtienen las siguientes graficas.

- Corrientes



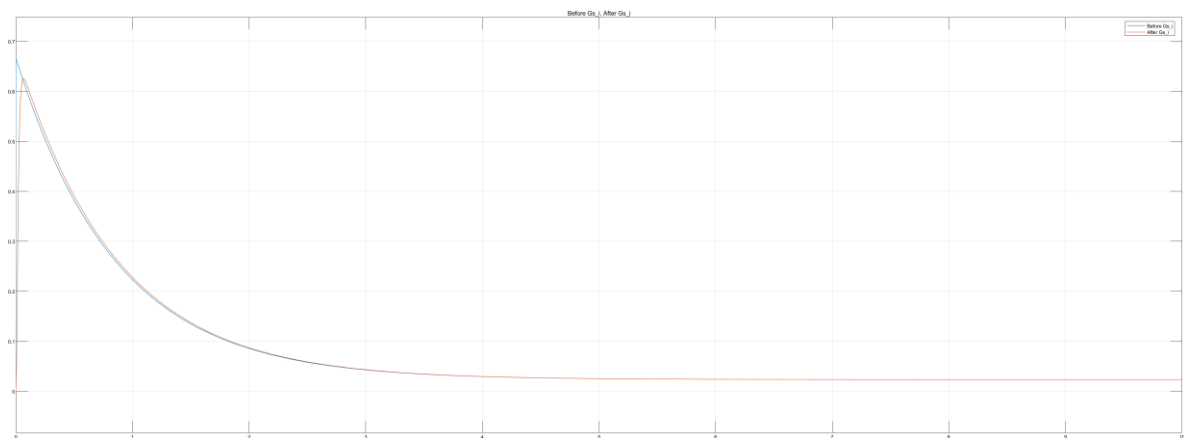
*Ilustración 4. Grafica de la respuesta del sensor de corriente*

La grafica que se aprecia es la respuesta en cuanto al sensor de corriente. La parte mas 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.



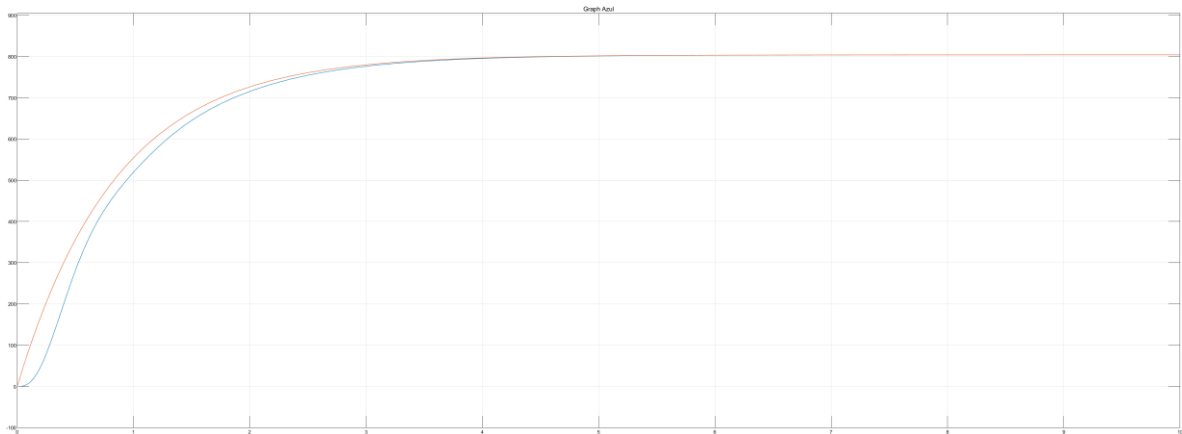
*Ilustración 5. Sistema simulink con ganancia.*

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.



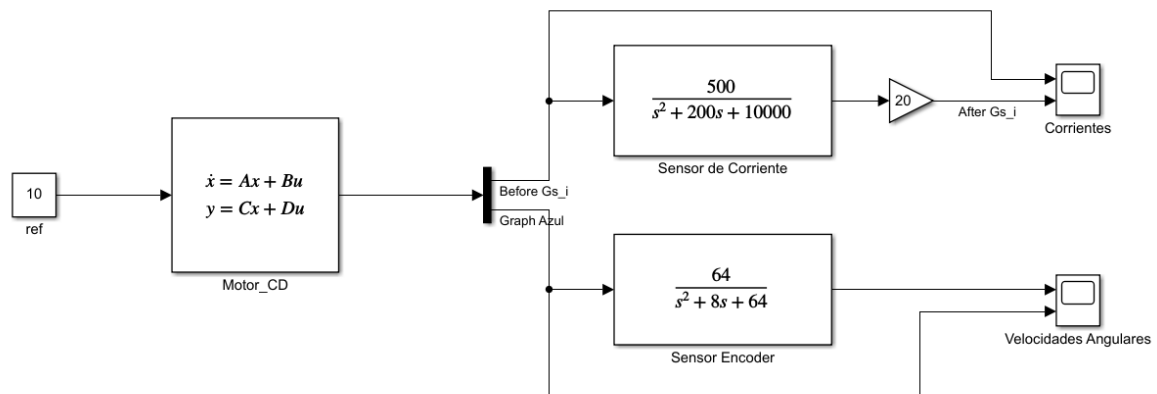
*Ilustración 6. Grafica de la respuesta del sensor de corriente*

- Encoder



*Ilustración 7. Grafica de la respuesta del encoder*

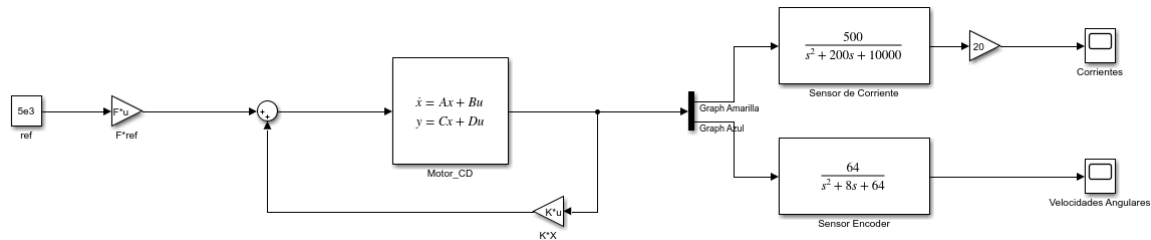
La grafica que tenemos es la respuesta al sensor correspondiente al encoder. De manera a como se visualiza, este sensor no fue necesario llegar a hacerle alguna modificación. Dado que este mismo logra el punto de estabilización aproximadamente al mismo momento que la señal original. El buscar optimizar 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.

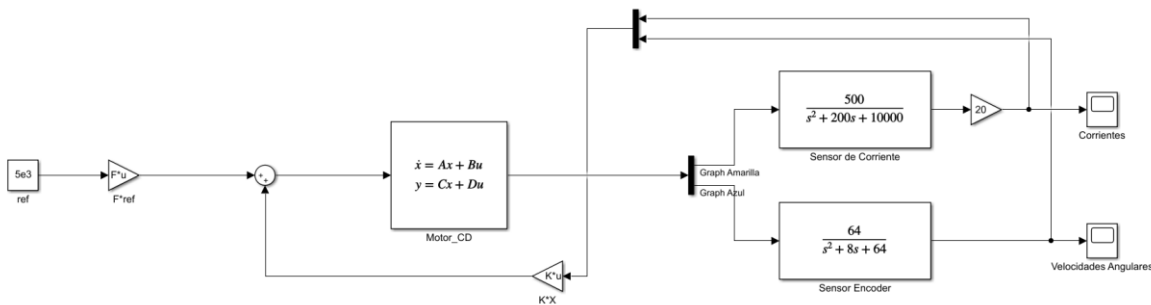
- ACKERMAN



*Ilustración 7. Sistema simulink ACKERMAN*

### UPDATE:

Después de una corrección del sistema en simulink, dado que la retroalimentación del sistema no fue implementada de la manera ideal. Acabo de la siguiente manera.



*Ilustración 8. Sistema simulink CORREGIDO, ACKERMAN*

De manera a como se pide en el ejercicio 2, lo que se busco fue que ahora el sistema lograra una referencia de  $W_{ref} = 5000 \frac{rad}{s}$ . De manera que buscamos que el encoder logre registrar dicha  $W_{ref}$  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.

**POLOS DEL SISTEMA:**

$$p_{1,2} = -\zeta\omega_n \pm \omega_n \sqrt{\zeta^2 - 1}$$

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



## Criterios

$$\zeta = \sqrt{\frac{\Psi^2}{1 + \Psi^2}} \text{ donde } \Psi = \frac{-\ln(M_p)}{\pi}$$

$$\zeta = 0.8261$$

$$w_n = \frac{4}{\zeta * t_s} = 1.6140$$

$$t_s = \frac{4}{\zeta \omega_n}$$

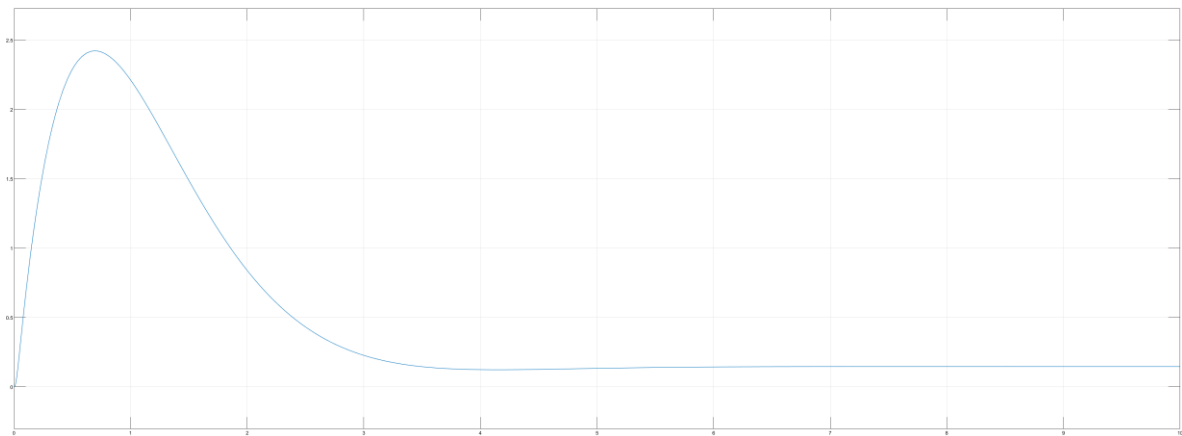
$$t_s = \frac{3}{\zeta \omega_n}$$

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

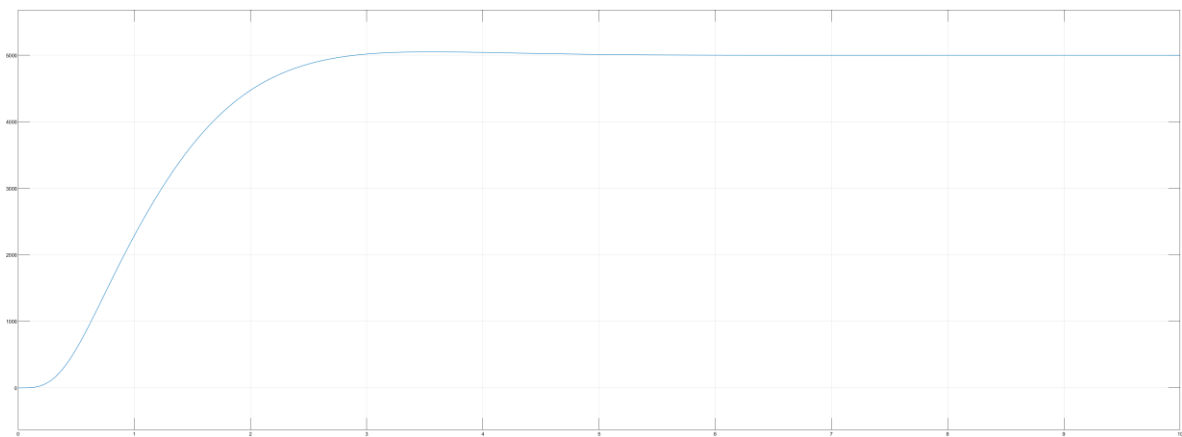
Pd =

$$-1.3333 + 0.9096i$$

$$-1.3333 - 0.9096i$$

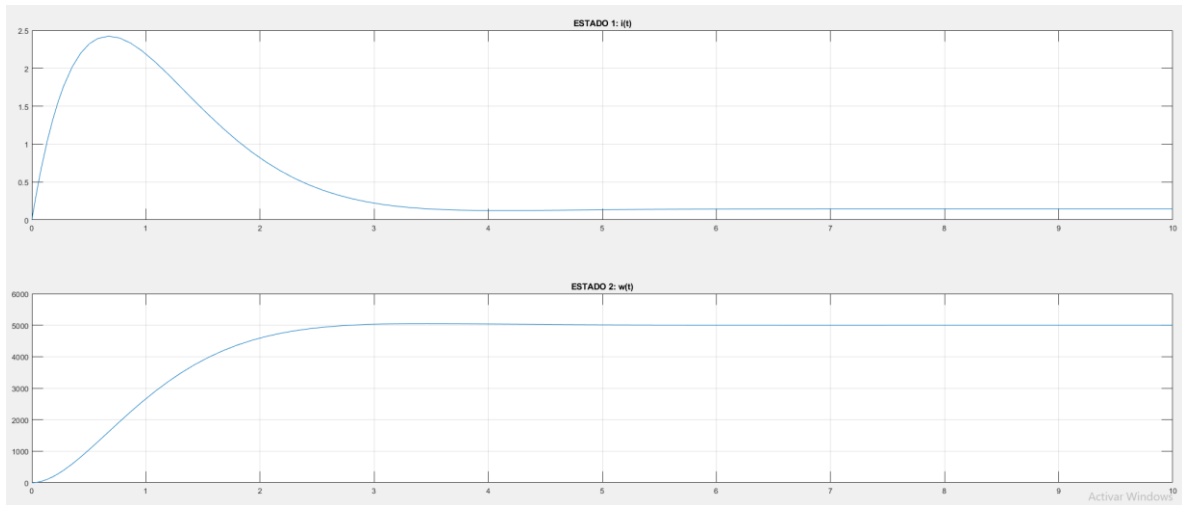


*Ilustración 9. Respuesta del sensor de corriente, Ackerman*



*Ilustración 10. Respuesta del encoder, Ackerman*

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

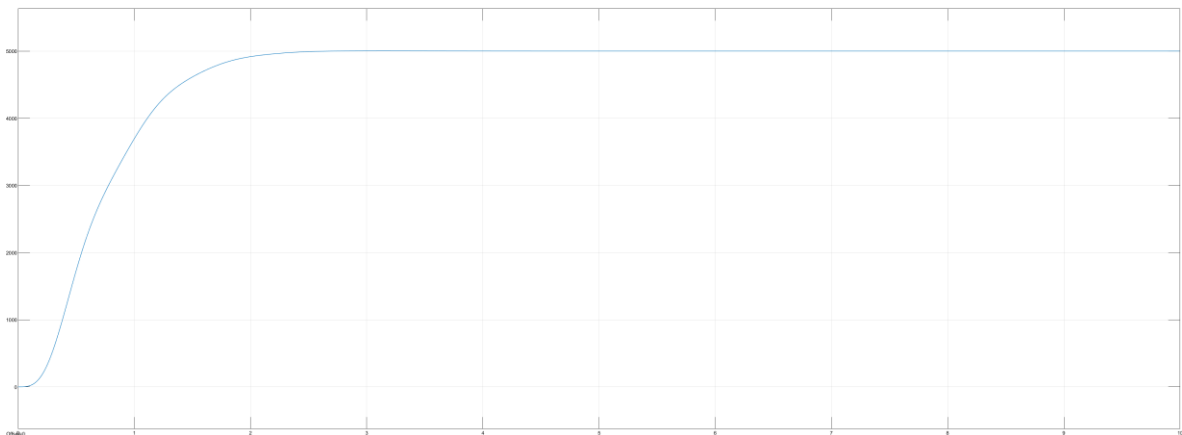


*Ilustración 11. simulación en matlab, Ackerman*

#### **UPDATE: GRAFICAS CORREGIDAR**

En un inicio, se simulo con los polos originales del sistema que calculamos. De manera que los resultados eran que el sistema con esos polos no era suficiente para responder en el tiempo adecuado que se requería en el problema, por lo que se propusieron otros polos. En un inicio, nos fuimos con unos polos muy extremistas en donde el sistema se estabilizaba en el tiempo ideal y de misma manera seguía la referencia, pero tenia un sobre impulso que a mi gusto no era necesario.

Una vez que vimos eso simplemente era cuestión de ajustar los polos para obtener una respuesta idea. De manera que, al obtener unos polos mas aceptables, obtuve la siguiente respuesta.



*Ilustración 12. simulación en matlab CORREGIDA, ACKERMAN*

- LQR

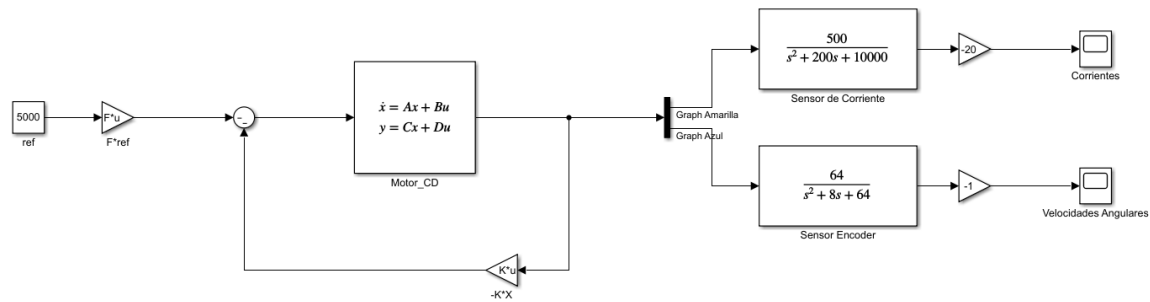


Ilustración 13. Sistema simulink, LQR

#### UPDATE:

Después de una corrección del sistema en simulink, dado que la retroalimentación del sistema no fue implementada de la manera ideal. Acabo de la siguiente manera.

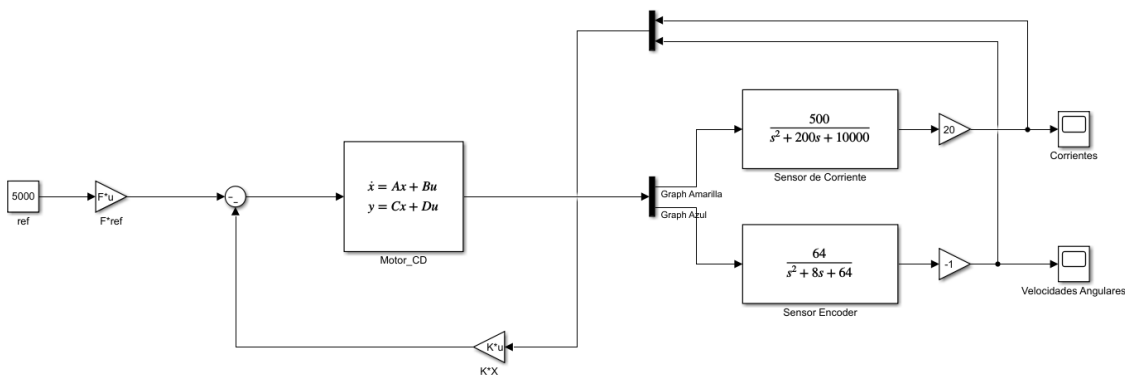


Ilustración 14. Sistema simulink CORREGIDO, ACKERMAN

De manera a como se pide en el ejercicio 2, lo que se busco fue que ahora el sistema lograra una referencia de  $W_{ref} = 5000 \frac{rad}{s}$ . De manera que buscamos que el encoder logre registrar dicha  $W_{ref}$  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.

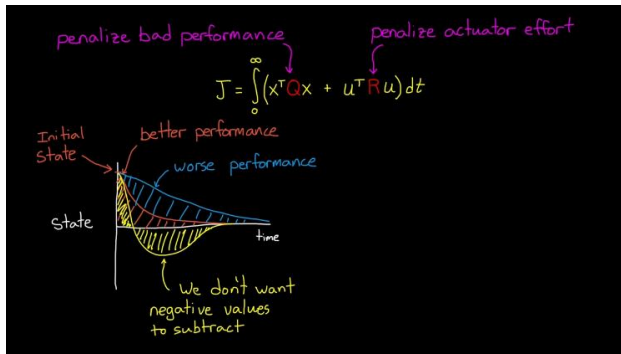


Ilustración 16. explicación sacada de matlab

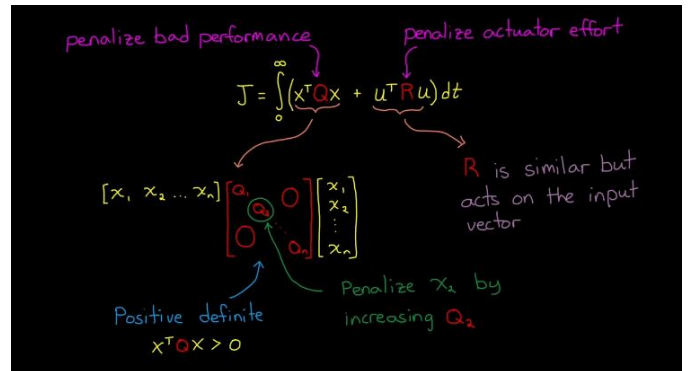


Ilustración 15. explicación sacada de matlab

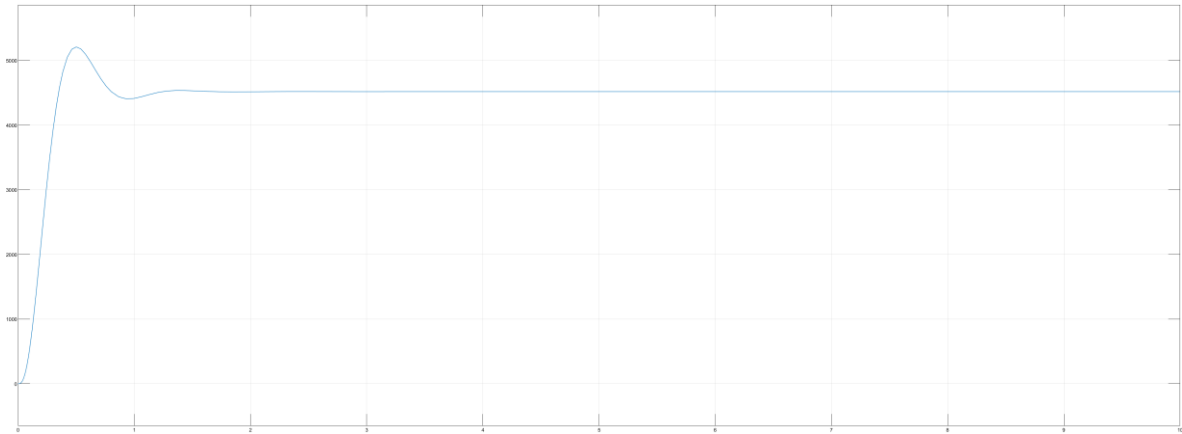
En cuanto ya la experimentación, primero se empezó con unas matrices base, para de ese punto ir tanteando terreno para saber hacia donde mover los valores de Q y R.

```
% LQ
Q = [1 0;
     0 1];
R = 0.01;
```

Obtenemos las siguientes respuestas.

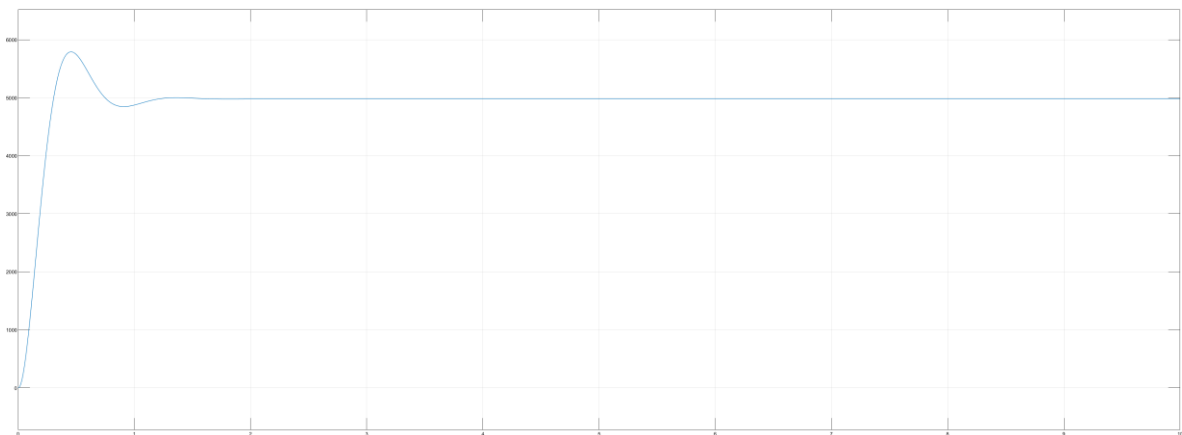


Ilustración 17. Respuesta del sensor de corriente, LQR



*Ilustración 18. 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.

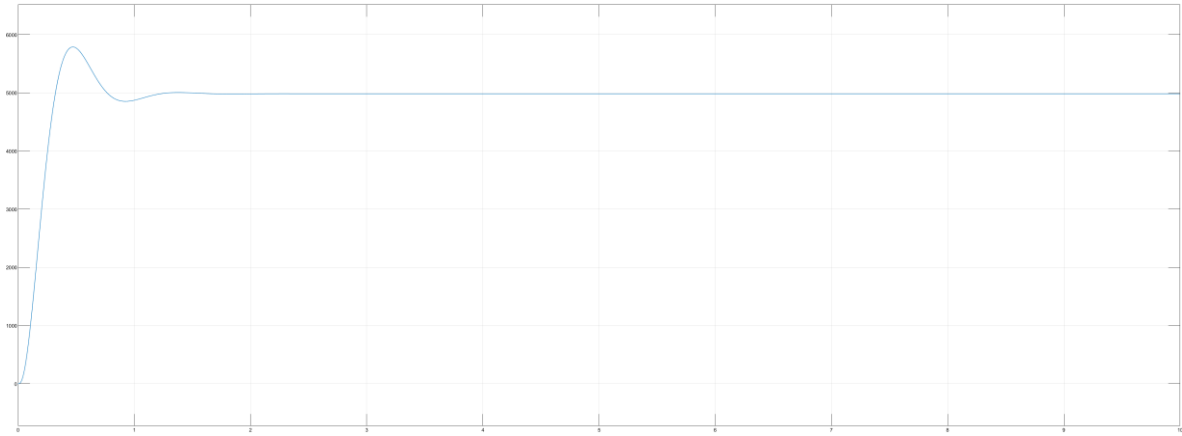


*Ilustración 19. Respuesta del encoder, LQR*

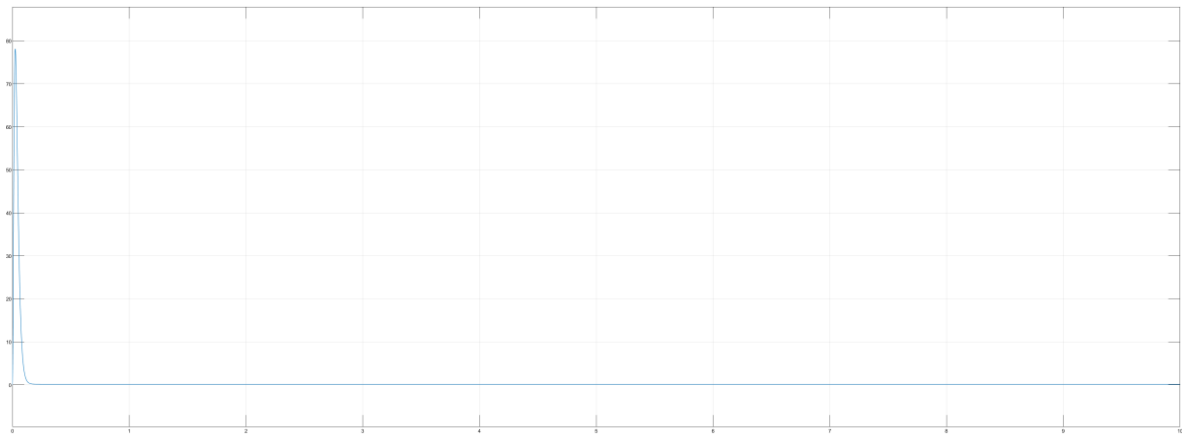
Teniendo dicha ventaja que la estabilización es sumamente corta en cuanto al tiempo, pero tomando un gran costo a la energía consumida para que logre esto.

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:

```
% LQ
Q = [50e4 0;
     0 900];
R = 1e3;
```



*Ilustración 20. Respuesta del encoder, LQR*



*Ilustración 21. Respuesta del sensor de corriente, LQR*

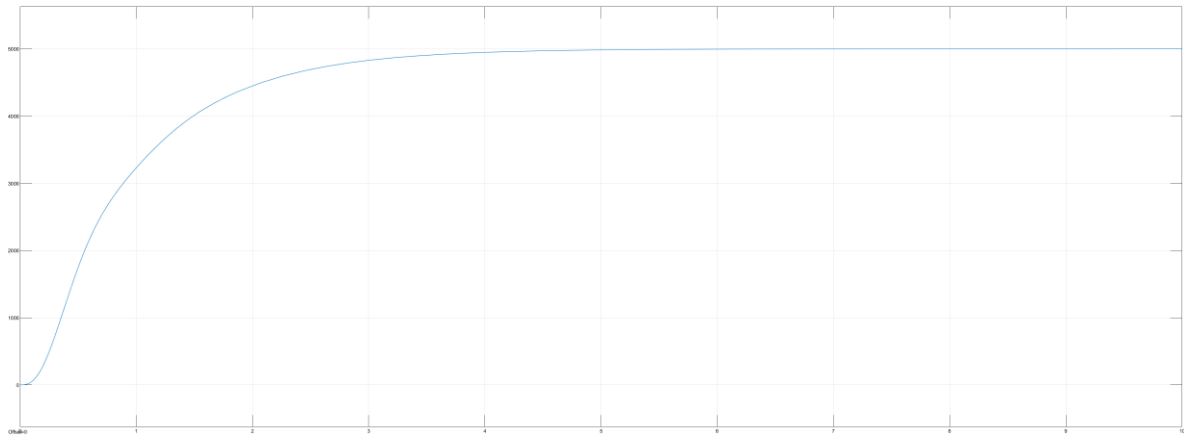
## CONCLUSION

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 más 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.

### UPDATE: GRAFICAS CORREGIDAS

En un inicio se intentaron los valores de Q y R originales y como era de esperarse, la grafica no era para nada la esperada. Sin embargo, a como es el modelo de control de LQR. El cual tiene sus ventajas y desventajas, y en lo personal no prefiero por el que sea por prueba y error (al momento). Después de múltiples intentos en esto mismo, se obtuvieron los siguientes valores de Q y R. En los que se logro obtener el resultado deseado. Sin embargo tras mover los valores, visualice que, al mover R hasta cierto rango, este llega a cambiar la polaridad del resultado.

En cuanto a esto mismo, al intentar múltiples valores de Q y R, llegando hasta los Giga y femto. No se logro que el tiempo de estabilización fuera menor a 3 segundos.



*Ilustración 22. simulación en matlab CORREGIDA, LQR*

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

```