Sistema de control II

Actividad N°1 Representación de Sistemas y Control PID

Docente:

- Julián Pucheta.

Alumno:

- Alaniz Franco.

Caso de estudio 1. Sistema de dos variables de estado

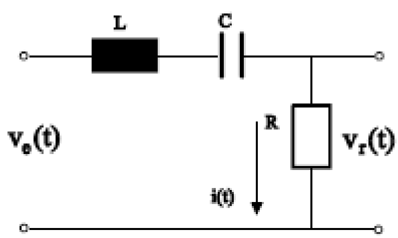


Fig. 1-1. Esquemático del circuito RLC.

Sea el sistema eléctrico de la Fig. 1-1, con las representaciones en variables de estado

$$\dot{x} = A\,x(t) + b\,u(t)$$

$$y = c^T x(t)$$

donde las matrices contienen a los coeficientes del circuito,

$$A = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix},\ b = \begin{bmatrix} 1/L \\ 0 \end{bmatrix},$$

$$c^T = \begin{bmatrix} R & 0 \end{bmatrix}$$

1- Asignar valores a R=2,2KΩ, L=10μHy, y C=100nF. Obtener simulaciones que permitan estudiar la dinámica del sistema, con una entrada de tensión escalón de 12V, que cada 1ms cambia de signo.

Desarrollo:

Para el estudio de la dinámica del circuito primero partí de la ecuación diferencial que describe el comportamiento en el tiempo del circuito:

$$vin(t) = L * \frac{di(t)}{dt} + C * \int i(t)dt + R * i(t)$$

Llevando todo al dominio de Laplace y obteniendo la ecuación de transferencia

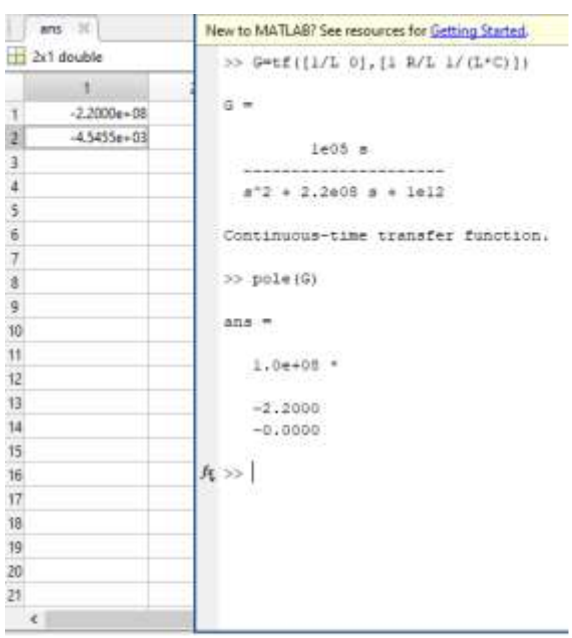
$$\frac{I(s)}{Vin(s)}$$

$$\frac{I(s)}{Vin(s)} = \frac{\frac{s}{L}}{s^2 + \frac{R}{L} * s + \frac{1}{L * C}}$$

Valuando:

$$\frac{I(s)}{Vin(s)} = \frac{\frac{s}{10e^{-6}}}{s^2 + 2e^8 * s + 1e^{12}}$$

Lo cual nos da 2 polos $\lambda_1 = -2.2e^8\ y\ \lambda_2 = -4.545e^3$



Como $\lambda_1$se encuentra más alejado del origen en el plazo S entonces la respuesta en el tiempo que aporta es mucho más rápida que la λ_2. De esto podemos obtener el tiempo que represente el 95% de la dinámica más rápida como

$$e^{\lambda_1 * t} \dashrightarrow t = \frac{\ln(.95)}{\lambda_1} = 2.331e^{-11}$$

Por lo que el tiempo de integración tiene que ser más chico para poder captar esta respuesta, en este caso tome un

$$t = \frac{t}{10} = 2.331e^{-12}$$

(Pero por cuestiones prácticas y menor tiempo de simulación tome opte por un valor de $1e^9$, que si es menor pero todavía puede captar la dinámica del sistema)

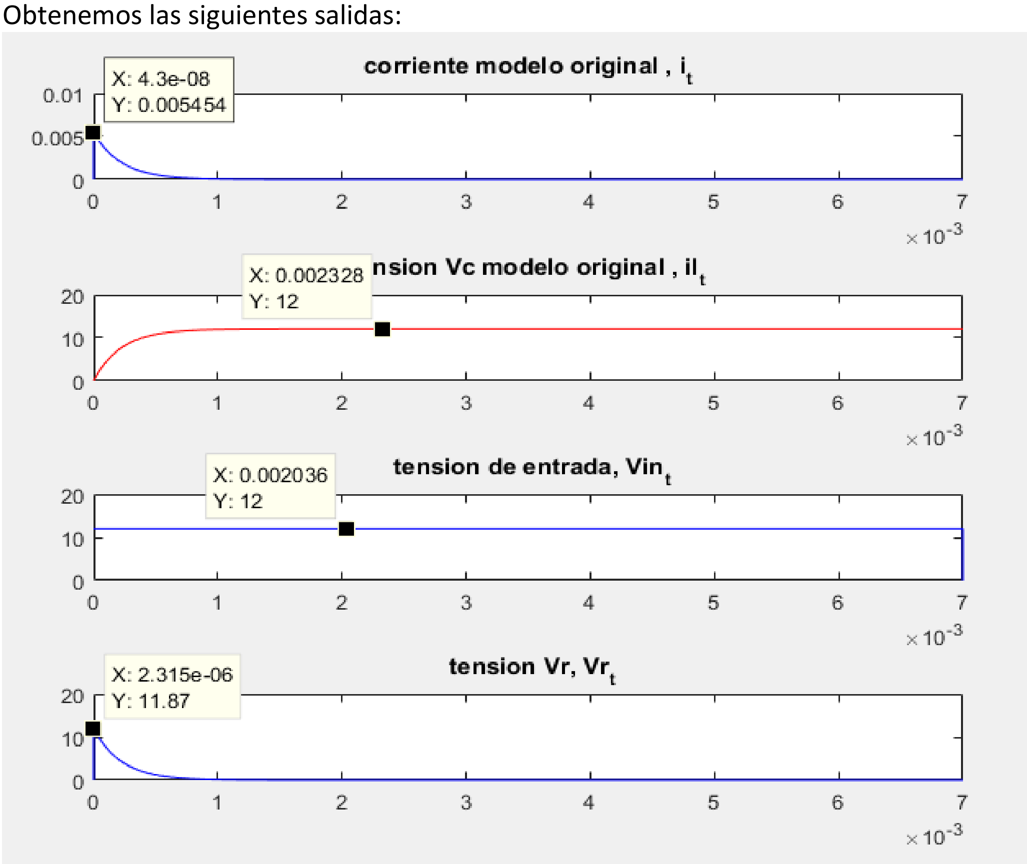
Para el tiempo de simulación tomé la respuesta más lenta y para cuando el tiempo llega a un 5% de su valor (y no hay mas transitorios), obtuve el siguiente valor:

$$e^{\lambda_2 * t} \dashrightarrow t = \frac{\ln(.05)}{\lambda_2} = 6.5905e^{-4}$$

Con el mismo de criterio que para simulación tiene que ser 10 veces más:

$$t = 10 * t \cong 7ms$$

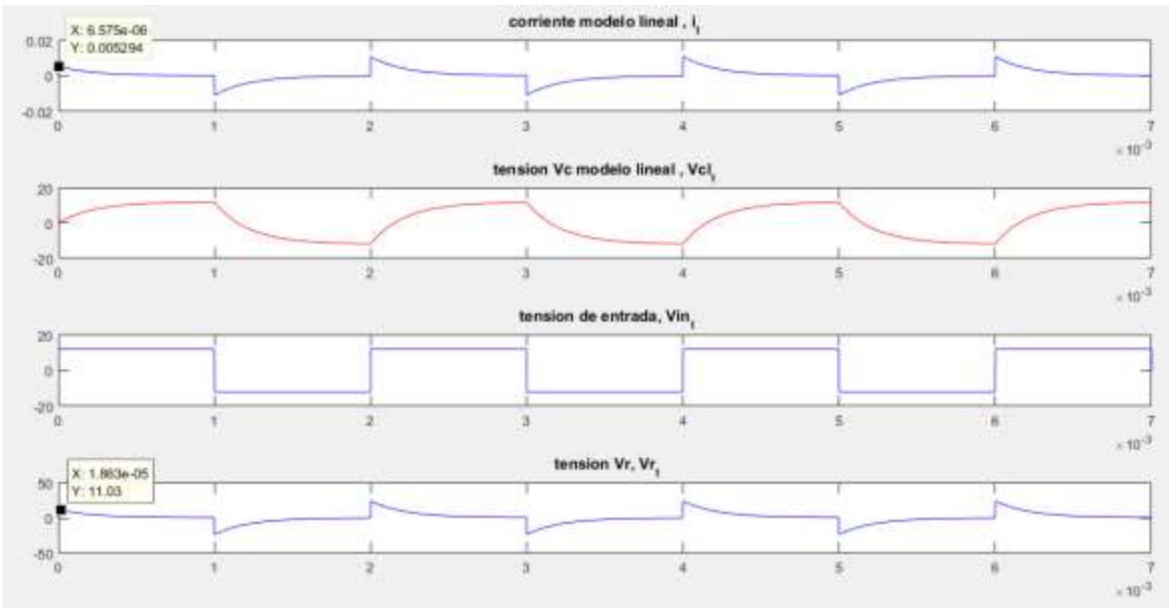
Obtenemos las siguientes salidas:



Vemos como la corriente empieza con un pico y esto lo produce el capacitor dado que no estaba cargado como sabemos:

$$i(t) = \frac{dq(t)}{dt} = \frac{V_{in}}{R} * e^{-\frac{t}{R*C}}$$

Por lo cual para el tiempo t=0 la corriente es máxima y a medida que el tiempo aumenta y el capacitor se va cargando la corriente disminuye.



Se observa como el cambio en la tensión de entrada produce una carga y descarga del capacitor obteniendo picos de corrientes en las transiciones de carga y descarga del capacitor. Igualmente, el modelo lineal funciona bastante bien porque si hacemos del producto de la corriente por la resistencia en algún instante de tiempo corresponde al valor obtenido en la gráfica.

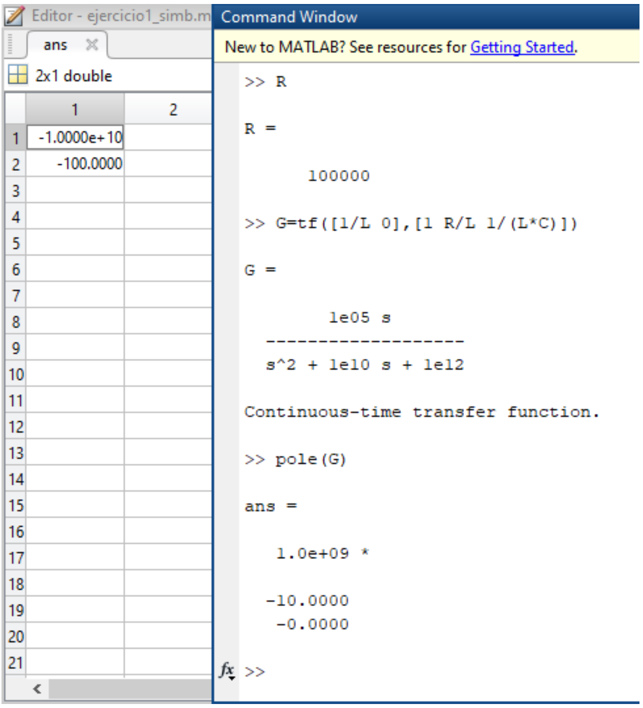
2. Asignar valores a R=100KΩ, L=10μHy, y C=100nF. repetir lo anterior para comparar el resultado y verificar la correcta simulación.

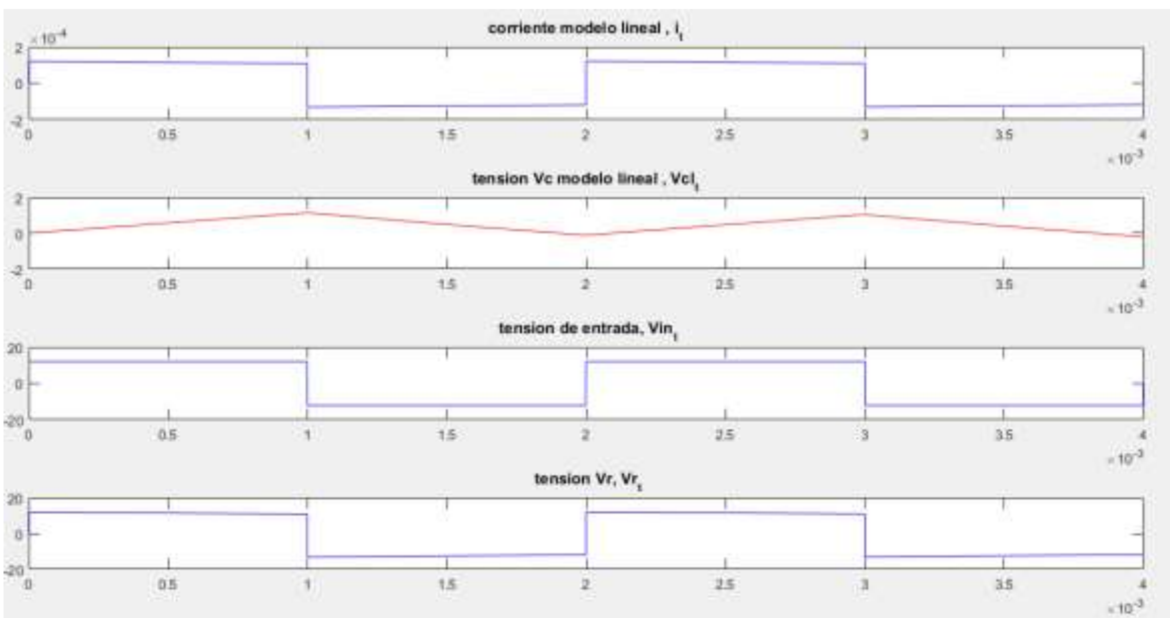
Como la resistencia ahora es mucho más grande vamos a tener un corrimiento de los polos por lo que el tiempo de integración va a tener que cambiar para tener una correcta simulación.
Como ahora los polos son $\lambda_1 = -1 \; y \; \lambda_2 = -100$

$$e^{\lambda_1 * t} \longrightarrow t = \frac{\ln(.95)}{\lambda_1} = 5.1293e^{-12}$$

$$t = \frac{t}{3} = 1.8e^{-12}$$

Opte por tomar un valor más chico, con un tiempo de simulación menor para simplificar la cantidad de cuentas y no demorar demasiado en cada simulación.





Vemos como el capacitor no alcanza a cargarse en una totalidad y la corriente en el circuito no disminuye a cero. En los cambios de polaridad de la tensión de entrada se ven cambios bruscos de corriente dado por la descarga del capacitor. Se nota igual al tener una resistencia más grande la corriente disminuye en modulo por lo que le modelo responde.

3. En el archivo Curvas_Medidas_RLC.xls (datos en la hoja 1 y etiquetas en la hoja 2) encontrarán las series de datos que deberían emplear para deducir los valores de R, L y C del circuito. Emplear el método de la respuesta al escalón, tomando como salida la tensión en el capacitor

Realizando el método de chena para obtener un modelo representativo de un sistema de 2 orden con polo, y tomando un tiempo inicial de muestra como indica la siguiente figura:

```
t_inic =

   1.0000e-04


K =

   11.8820


sys_G_ang =

        -4.999e-07 s + 1
  -----------------------------
  9.891e-06 s^2 + 0.02178 s + 1

Continuous-time transfer function.


sys_G_ang =

       -5.94e-06 s + 11.88
  -----------------------------
  9.891e-06 s^2 + 0.02178 s + 1

Continuous-time transfer function.
```

Se obtiene la sys_G_ang que representa la función de trasferencia obtenido para el caso de

$$\frac{V_c(S)}{V_{in}(S)}$$

A partir de la ecuación característica de sys_G_ang y sacando factor común el término que acompaña a s^2, igualándola con la de:

$$\frac{V_c(S)}{V_{in}(S)} = \frac{1}{L * C} * \frac{1}{s^2 + \left(\frac{R}{L} * s\right) + \left(\frac{1}{L * C}\right)}$$

Se obtiene por igualdad y tomando un valor inicial de resistencia de 10Ω, valores de L, C.

```
Calculo de valores R,L,C

R =

    10


L =

    0.0045


C =

    0.0022
```

Cuya función de transferencia obtenida resulta:

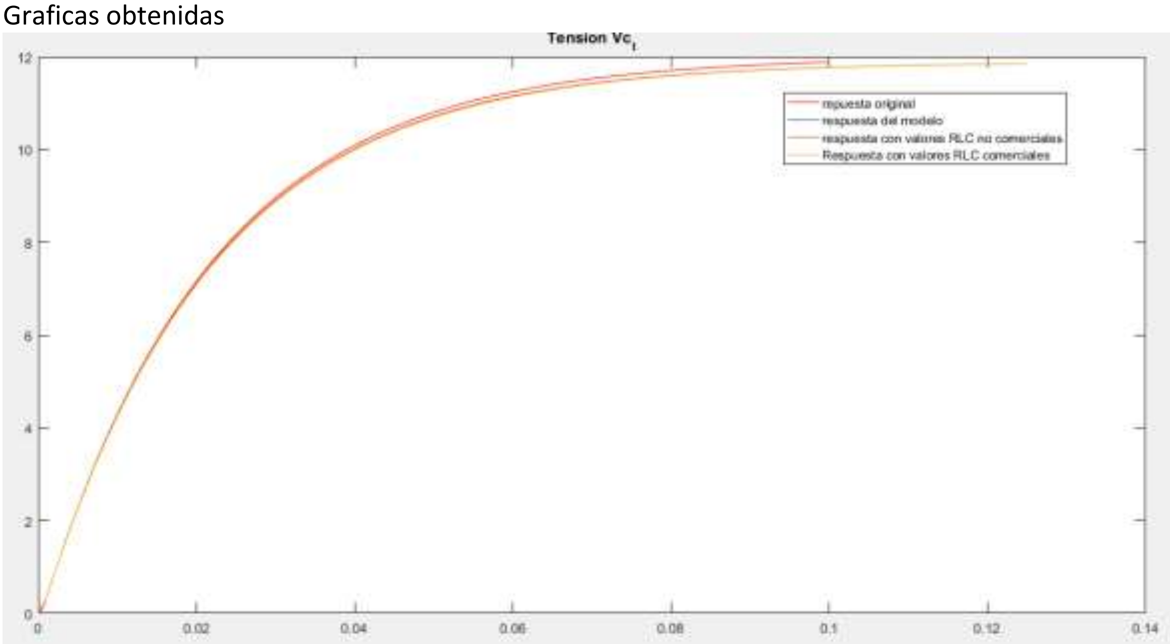
```
sys_G_obte =

        1.201e06
  -----------------------
  s^2 + 2202 s + 1.011e05

Continuous-time transfer function.
```

Y si ahora tomamos valores comerciales de R, L, C, tenemos:

$$R = 10\Omega, L = 4.5mHy\ (esto\ no\ es\ comercial\ pero\ se\ podria\ diseñar), C = 2200uF$$

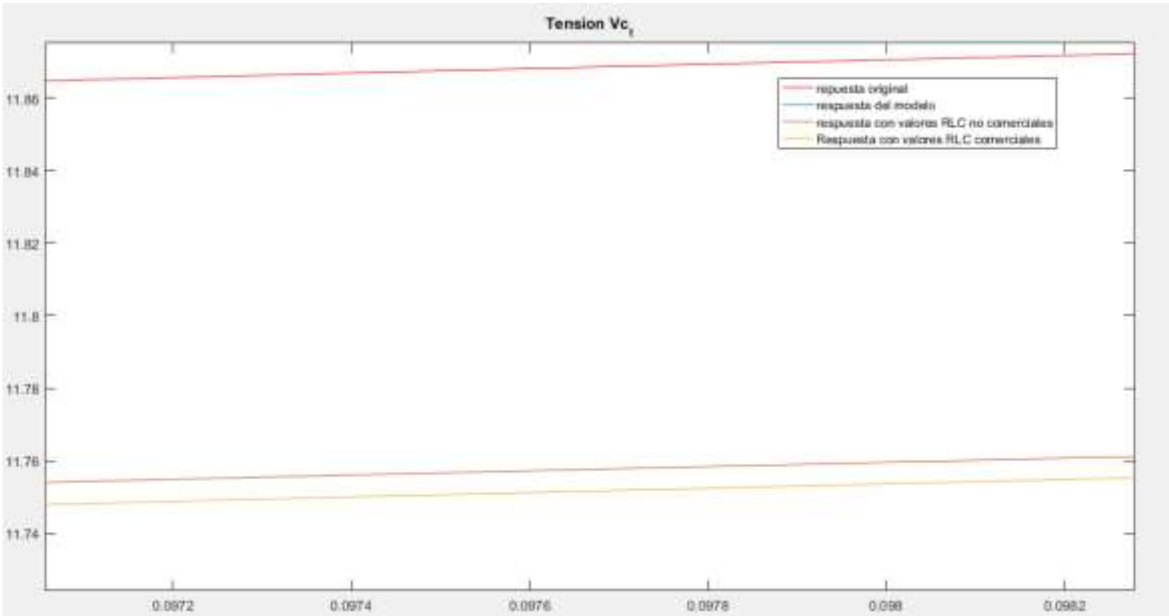
```
sys_G_obte_come =

         1.2e06
  ----------------------
  s^2 + 2222 s + 1.01e05

Continuous-time transfer function.
```

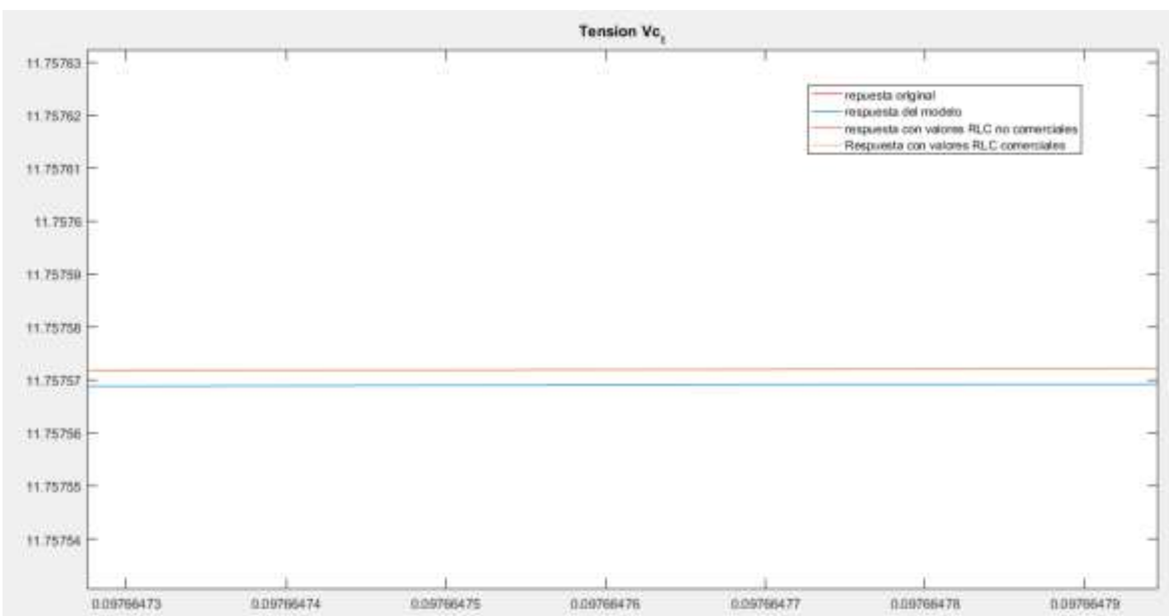
Graficas obtenidas



Las gráficas son casi indistinguibles unas de otras, pero se observan diferencia respecto de la original y la del modelo con valores comerciales, la de no comerciales y sacada por el método de chena están casi superpuestas.

Si hacemos un zoom se manifiestan las diferencias





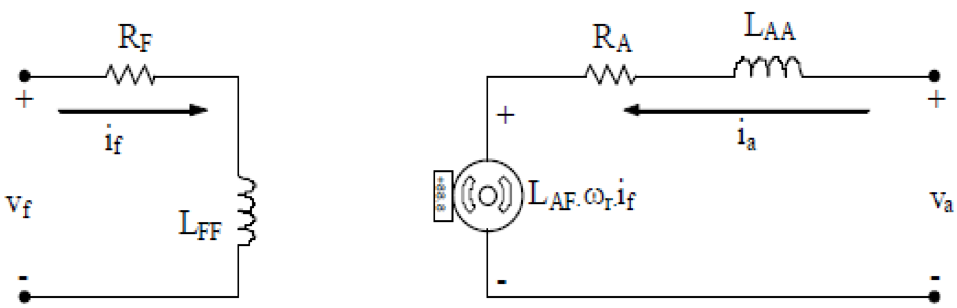
Lecciones aprendidas:

En esta actividad pudimos aprender la importancia de obtener un tiempo de integración adecuado para las simulaciones, este procedimiento se aprendió desarrollando la función de transferencia y por medio de sus polos.

Complicaciones:

- haber tomado un tiempo inicial más grande para la estimación de la función de transferencia y obtenía valores de L, C inviables. Y el método de obtener los parámetros se complicaba o eran valores poco usuales.
- El tiempo de integración para las simulaciones.

Caso de estudio 2



$$\frac{di_a}{dt} = -\frac{R_A}{L_{AA}} i_a - \frac{K_m}{L_{AA}} \omega_r + \frac{1}{L_{AA}} v_a \qquad (5)$$

$$\frac{d\omega_r}{dt} = \frac{K_i}{J} i_a - \frac{B_m}{J} \omega_r - \frac{1}{J} T_L \qquad (6)$$

$$\frac{d\theta_t}{dt} = \omega_r . \qquad (7)$$

Tomando $T_l = 0$.

$$\frac{\omega_r(s)}{v_a(s)} = \frac{K_i}{s^2 L_{AA} J + s(R_A J + L_{AA} B) + (R_A B + K_I K_m)}.$$

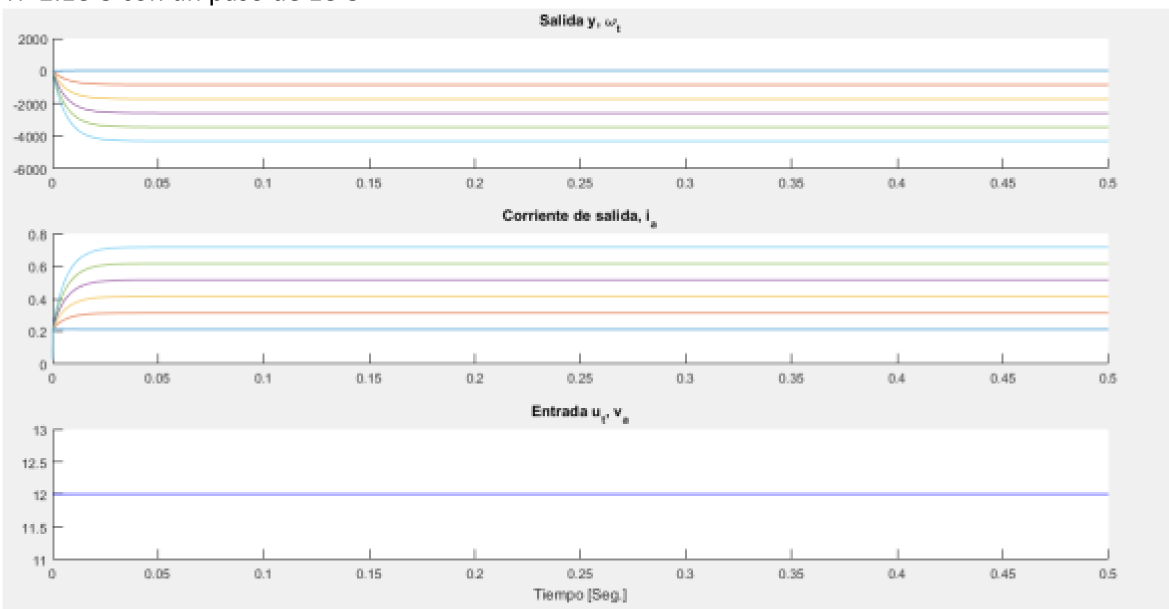
Dadas las ecuaciones del motor de corriente continua con torque de carga T_L no nulo, con los parámetros$L_{AA} = 366\ 10^{-6};\ J = 5\ 10^{-9};\ R_A = 55{,}6;\ B = 0;\ K_i = 6{,}49\ 10^{-3};\ K_m = 6{,}53\ 10^{-3}$:

Implementar un algoritmo de simulación para inferir el comportamiento de las variables interés mediante integración Euler con $\Delta_t = 10^{-7}$segundos para:

1. Obtener el torque máximo que puede soportar el motor modelado mediante las Ecuaciones. (5) (6) y (7) cuando se lo alimenta con 12V, graficando para 5 segundos de tiempo la velocidad angular y corriente i_a.

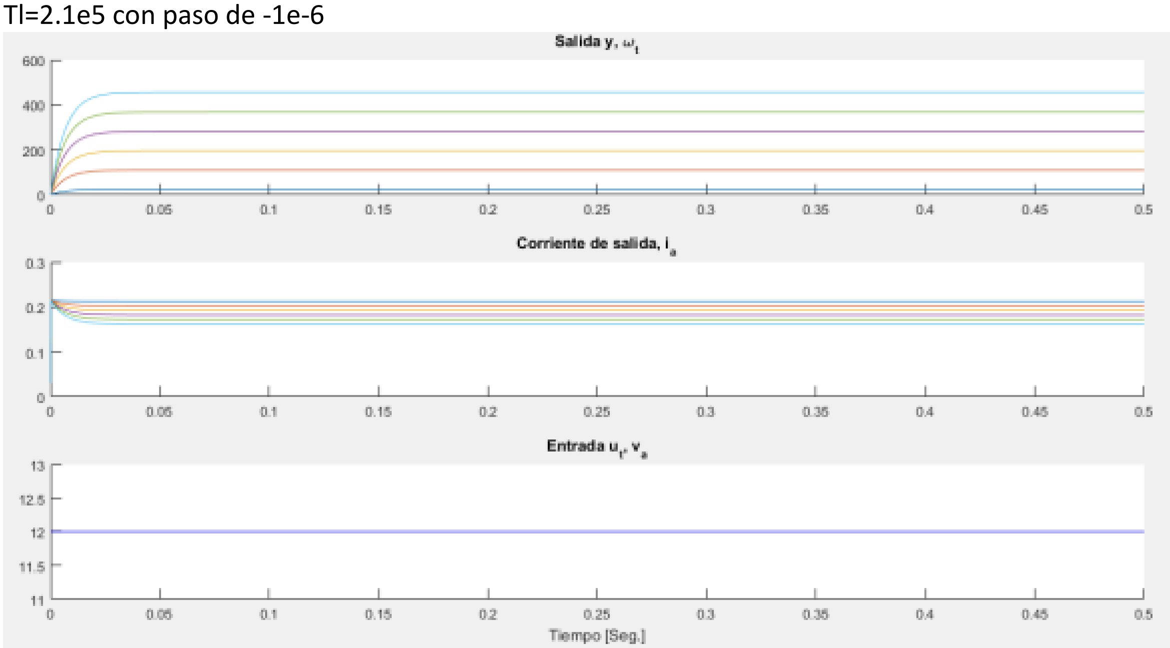
Desarrollo:

Tl=2.1e-5 con un paso de 1e-5



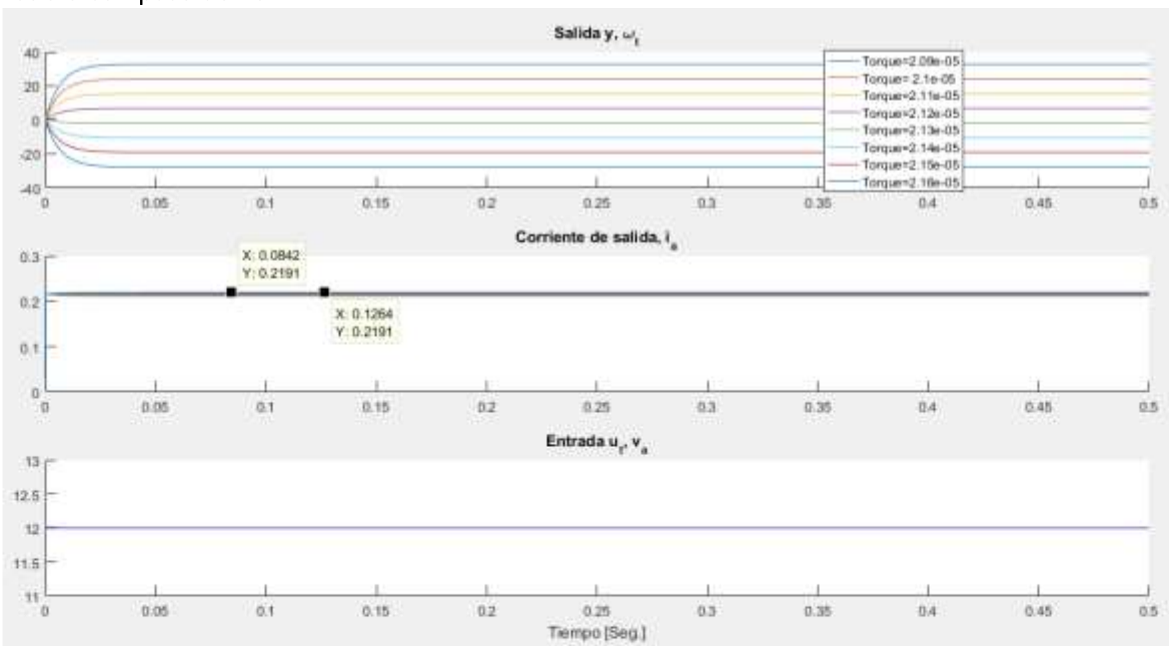
Tenemos valores de corrientes y velocidades angulares que van en aumento a medida que sube el torque.

Tl=2.1e5 con paso de -1e-6



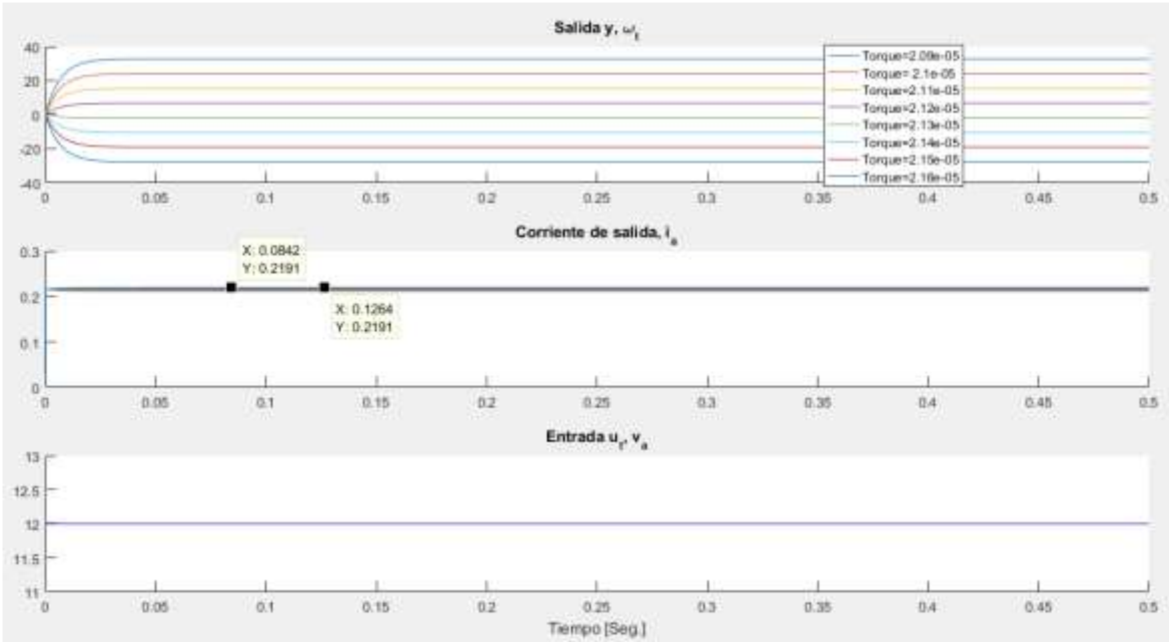
A medida que el torque es más chico la corriente en el motor en disminuye y el motor tiene más velocidad angular con corrientes más chicas.

Tl=2.09e-5 con paso de 1e-7.



Se puede observar que el límite del torque esta entre 2.12e-5 y 2.13e-5, para lo cual la velocidad angular sea cero lo que nos dice que el motor no podría funcionar con esta tensión de 12v de entrada y el torque dado.

2. Mostrar simulaciones de 5 segundos que permitan observar la corriente i_a en todo momento y establecer su valor máximo como para dimensionar dispositivos electrónicos



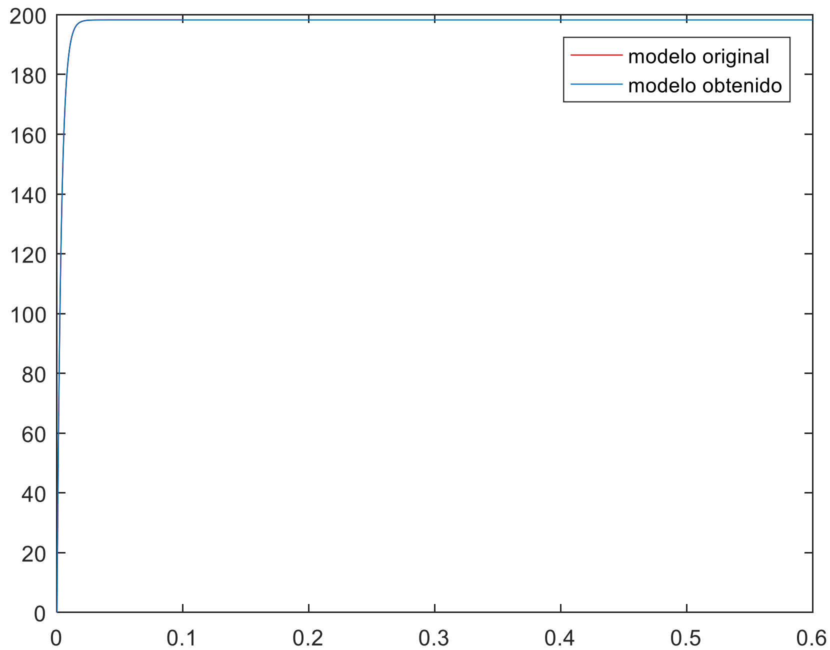
Como se ve en la figura la corriente máxima está dada por el torque máximo, esta corriente es de aproximadamente 291mA.

3. A partir de las curvas de mediciones de las variables graficadas en la Fig. 1-3, se requiere obtener el modelo del sistema considerando como entrada un escalón de 12V, como salida a la velocidad angular, y a partir de 0,1segundo se aplica un T_L aproximado de7,5 10^2 Nm. En el archivo Curvas_Medidas_Motor.xls están las mediciones, en la primera hoja los valores y en la segunda los nombres. Se requiere obtener el modelo dinámico, para establecer las constantes de la corriente.

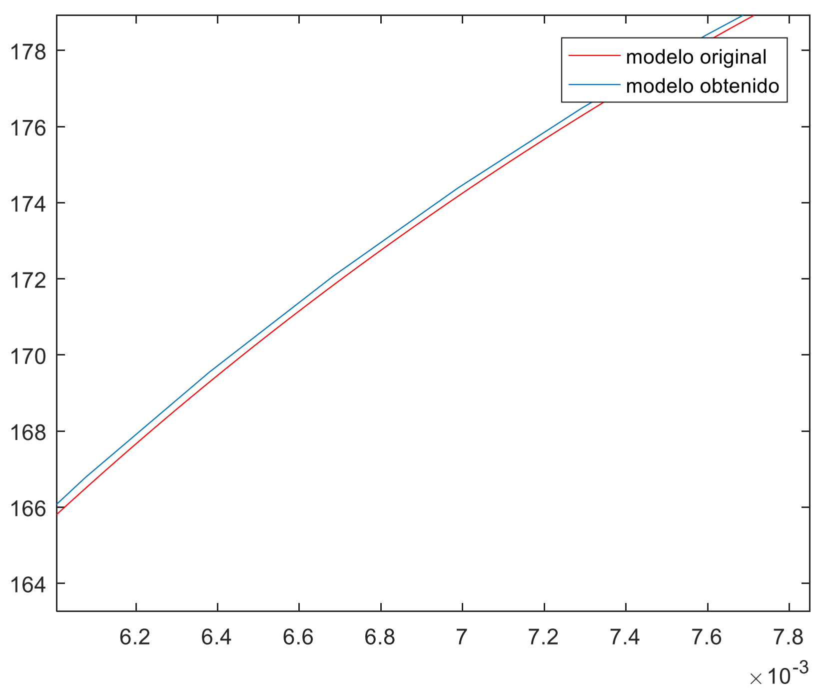
Para obtener el modelo solo tomamos lo valores antes de 0.1seg ya que en ese tiempo ya se puede obtener las muestras necesarias de la dinámica del sistema, por lo que obtenemos

```
sys_G_ang =

       -2.476e-06 s + 16.52
  ------------------------------
  2.049e-08 s^2 + 0.003304 s + 1

Continuous-time transfer function.
```

Cuya respuesta es:

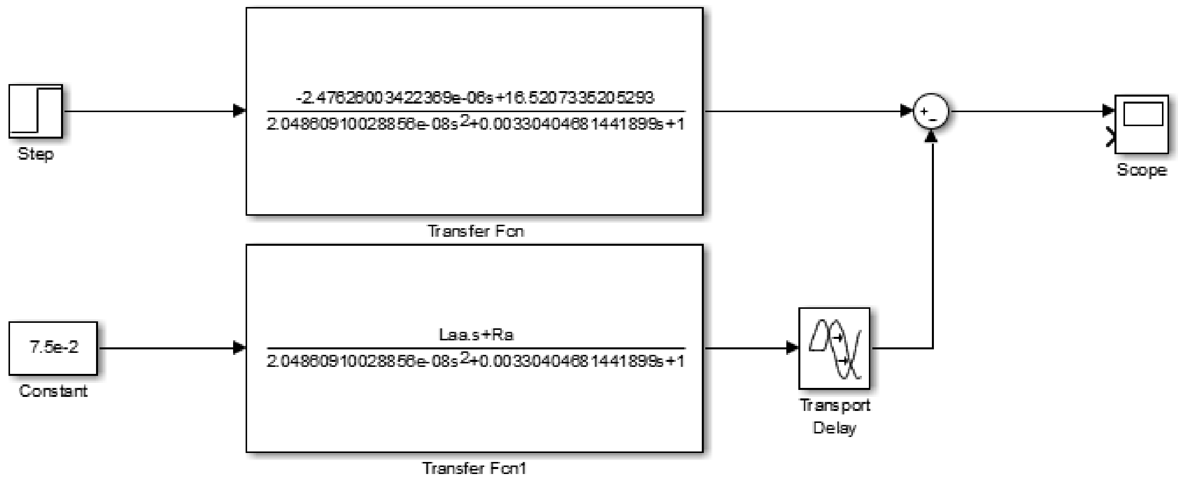


Haciendo un zoom se puede observar la diferencia.



Ahora para corroborar el modelo vamos a realizar una simulación por medio de simulink y la función de transferencia obtenida sin despreciar el Tl:

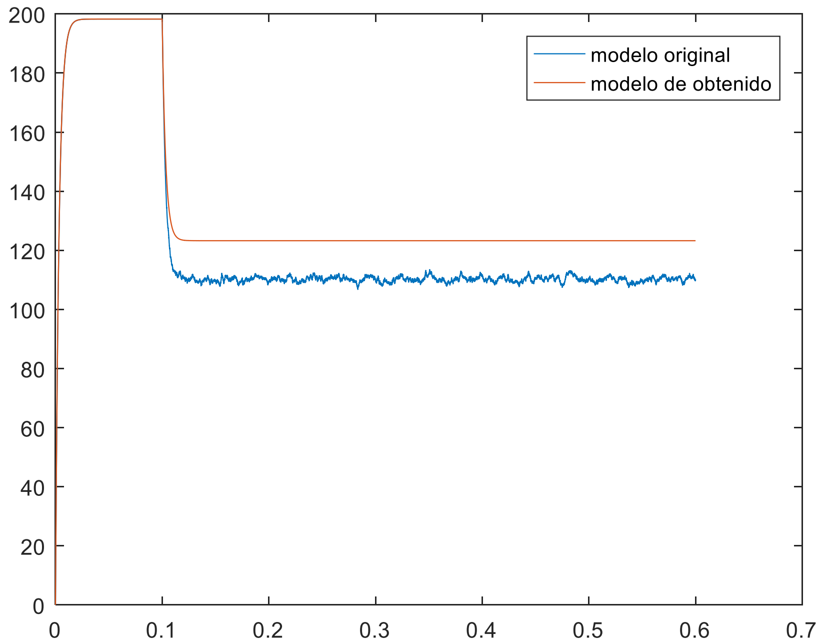
$$W_r(s) = \frac{k_i * V_a(S) - Tl * (L_{aa} * S + R_{aa})}{S^2 * L_{aa} * J + S * (J * R_{aa} + L_{aa} * B_m) + k_i * k_m + B_m * R_a}$$



Los valores de L_{aa} y R_a se tomaron a modo de prueba, con el fin de probar el modelo.

$$L_{aa} = 336e^{-9} \; y \; R_a = 1000\Omega$$

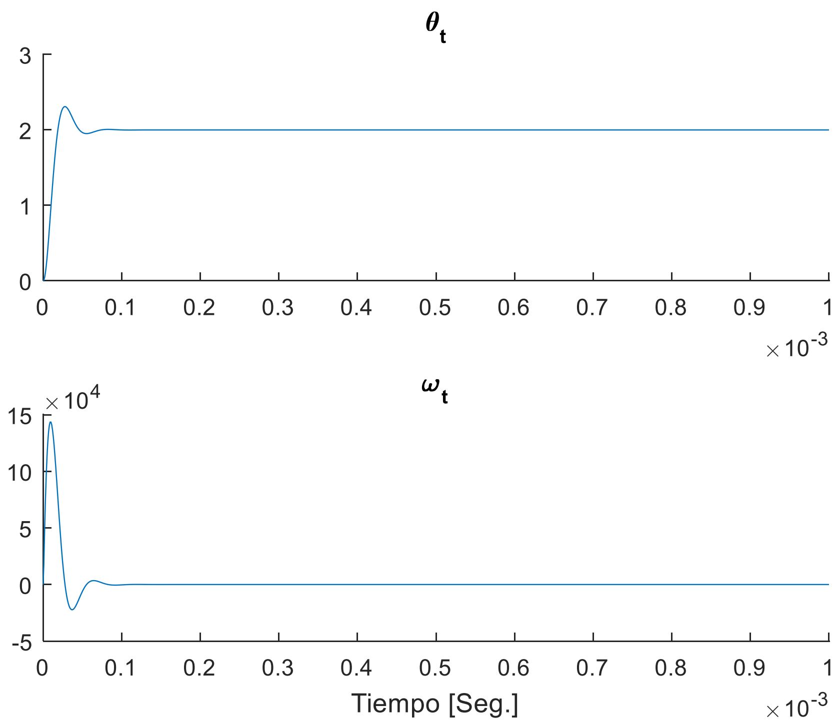
Obtenemos las siguientes figuras una vez exportando los datos desde simulink y ploteando las 2 señales para su comparación:

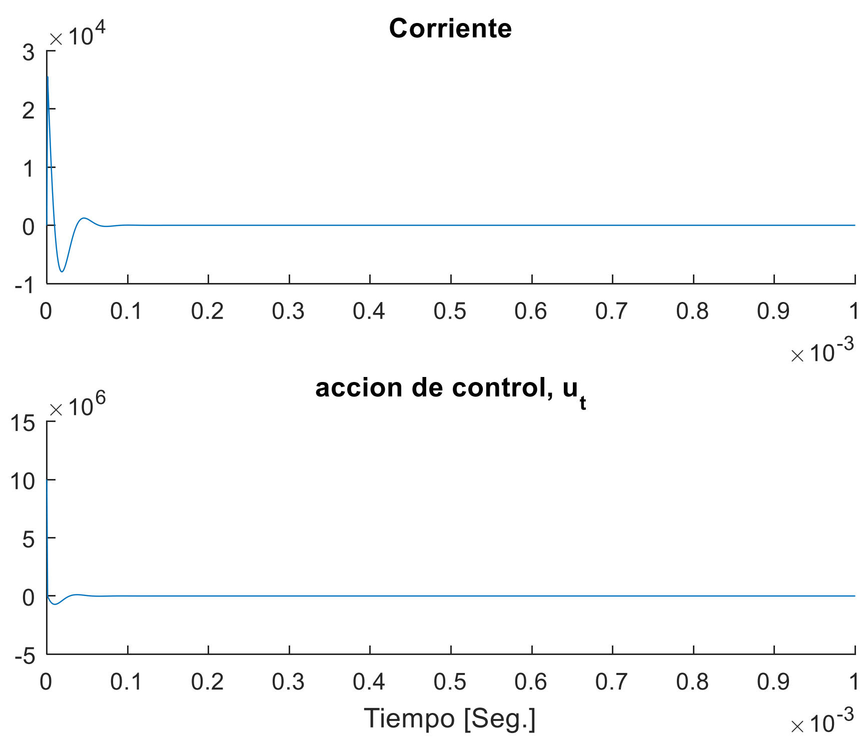


Se puede observar que el modelo obtenido responde de igual manera a la acción del torque (no es el mismo porque los valores de Laa y Ra no son iguales) por lo que se podría concluir que el modelo obtenido es una muy buena aproximación.

4. Implementar un PID en tiempo discreto para que el ángulo del motor permanezca en una referencia dada, por ejemplo 2 radianes. (Tip: partir de KP=0,1; Ki=0,01; KD=5).

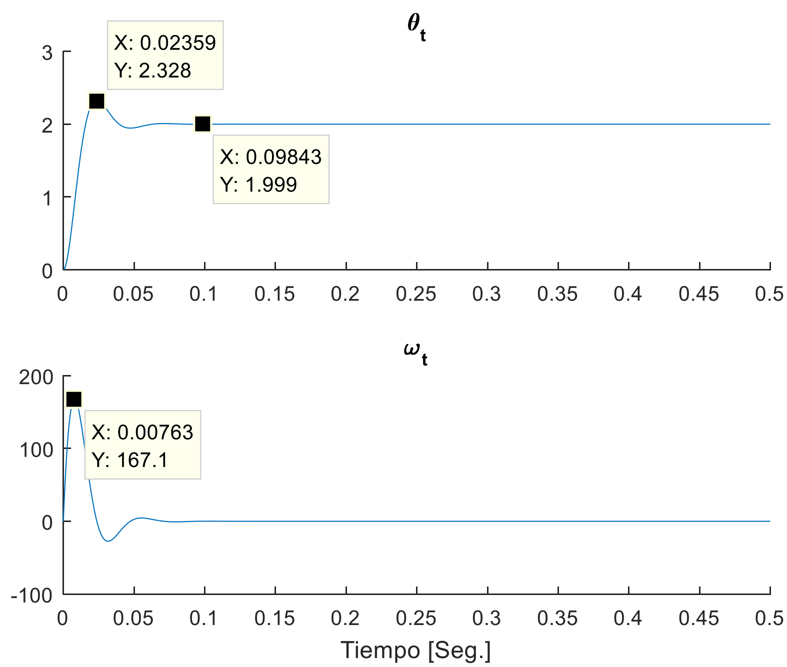
Ejecutando el código para la implementación de un PID obtenemos con los valores de K_p, K_i y K_d dados como tip, además se utilizó un $T_l = 0$, se obtuvo:

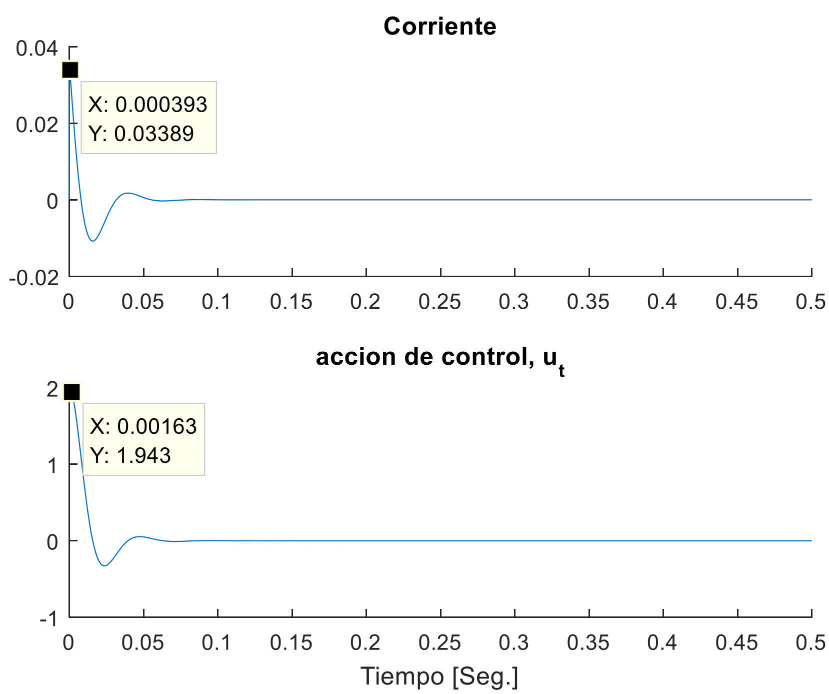




Como vemos con estos valores de $K_p, K_i\ y\ K_d$, tenemos acciones de control, corriente y velocidad angular inviables, por lo que se optó por tomar otros valor de $K_p, K_i\ y\ K_d$.

$Con: \ K_p = 1, K_i = 0 \ y \ K_d = 0$





Estos valores son mucho más razonables ya que las acciones de control, corriente y velocidad angular son más viables para un motor.
Si se quiere aumentar el tiempo de establecimiento y reducir el sobre pasamiento tendríamos que aumentar el K_d pero esto tiene un inconveniente (se explica en el apartado complicaciones).

El error en estado estable es muy chico por lo que agregar el termino integrador no haría falta en este caso, ya que este toma partido para corregir el error en estado estable. Igual mente agregar este término hacer tener un polo doble en el origen por lo que el sistema tendería a ser más inestable.

Lecciones aprendidas:

- Comprender un poco mejor el manejo de un PID.
- Uso de simulink dentro de Matlab, antes me resultaban 2 herramientas distintas ahora las pude unir.
- La importancia de tener en cuenta el torque máximo de un motor, y así dimensionar la corriente presente.
- Interpretación de gráficos para corroborar el correcto funcionamiento de los modelos.

Complicaciones:

- En el método de estimar la función de transferencia resulto complicado corroborarla por medio de una perturbación(torque), simplemente por código. Pero al usar simulink esto resulto mucho más fácil.
- el agregar K_d hace que se tengo un cero en el origen y como para controlar la posición tenemos un integrados por ende un polo en el origen, se cancelaria con el cero del termino derivativo, esto a su vez causa acciones de control muy grandes como se vio en la primera simulación del PID. Por lo tanto, el agregar el termino derivativo no es una muy buena opción en estos casos, se debería optar por un término derivativo desplazado.

Códigos utilizados.

Caso de estudio uno

```
%ejercicio 1 RLC parte simbolica
clc;clear all;
syms R L C i vc vin ip vcp
ip  =(1/L)*vin-(1/L)*vc-(R/L)*i
vcp = 1/C * i
A = [[diff(ip,i) diff(ip,vc)]
     [diff(vcp,i) 0 ]];

B = [[diff(vcp,i)]
     [0]];
 pretty(simplify(A))
 pretty(simplify(B))
clc;clear all;
 T=2e-3; At=1e-9; Kmax=T/At; t=linspace(0,T,Kmax);
 R=2.2e3;L=10e-6;C=100e-9;vin=12;
 Ip=0;Vcp=0;
```

```
I=zeros(1,Kmax);Vc=zeros(1,Kmax);u=linspace(0,0,Kmax);
%Condiciones iniciales
I(1)=0;Vc(1)=0;u(1)=vin;
A=[-R/L -1/L ; 1/C  0]; %eig(A) me da los autovalores de A que corresponden a los polos de la ecuacion caracteristica
B=[1/L; 0];
E=[R 0];
tve(1)=0;Il(1)=0;Vcl(1)=0;x=[I(1) Vc(1)]' ;Vc_t(1)=0;Xop=[0 0]'; %punto de operacion es cero
ii=0;
for i=1:Kmax-1
    ii=ii+At;
    if(ii>=1e-3)
        ii=0;
        vin = vin*-1;
    end
    u(i)=vin;
    %sistema real
    Ip =(1/L)*u(i)-(1/L)*Vc(i)-(R/L)*I(i);
    Vcp = 1/C * I(i);
    I(i+1)=I(i)+Ip*At;
    Vc(i+1)=Vc(i)+Vcp*At;
    %variables de sistema lineal
    xp=A*(x-Xop)+B*u(i);
    x=x+xp*At;
    Y=E*x;
    Vc_t(i+1)=Y(1);
    Il(i+1)=x(1);
    Vcl(i+1)=x(2);
    %tve(i+1)=tve(i)+At;
end
figure(1)
subplot(4,1,1);%hold on;
plot(t,Il,'b');title('corriente modelo lineal , i_t');
subplot(4,1,2);%hold on;
plot(t,Vcl,'r');title('tension Vc modelo lineal , Vcl_t');
subplot(4,1,3);%hold on;
plot(t,u,'b');title(' tension de entrada, Vin_t');
subplot(4,1,4);%hold on;
plot(t,Vc_t,'b');title(' tension Vr, Vr_t');
%  figure(2)
%  subplot(3,1,1);hold on;
%  plot(t,Il,'b');title('corriente modelo lineal , Vcl_t');
%  subplot(3,1,2);hold on;
%  plot(t,Vcl,'b');title('tension Vc modelo lineal , Vcl_t');
%  subplot(3,1,3);hold on;
%  plot(t,u,'b');title(' tension de entrada, Vin_t');
```

```
clear all;close all;
num = xlsread('Curvas_Medidas_RLC.xls');
t=num(1:end,1);
i_t=num(1:end,2);
Vc_t=num(1:end,3);
opt = stepDataOptions;
opt.StepAmplitude = 12;

%t_inic=0.03 si tomo este tiempo obtengo distintos valores en la ecuacion
%caracteristica que hace que los valores de L,C sean dificil de obtener
%comercialmente
t_inic=0.0001
[val, lugar] = min(abs(t_inic-t)); %obtengo punto a punto el valor mas proximo al
t_inic y obtengo el min punto del vector t
y_t1=Vc_t(lugar);
t_t1=t(lugar);
ii=0;
ii=ii+1;
[val, lugar] =min(abs(2*t_inic-t));
t_2t1=t(lugar);
y_2t1=Vc_t(lugar);
[val, lugar] =min(abs(3*t_inic-t));
t_3t1=t(lugar);
y_3t1=Vc_t(lugar);

K=Vc_t(end)/opt.StepAmplitude

k1=(1/opt.StepAmplitude)*y_t1/K-1; %Afecto el valor del Escalon
k2=(1/opt.StepAmplitude)*y_2t1/K-1;
k3=(1/opt.StepAmplitude)*y_3t1/K-1;

be=4*k1^3*k3-3*k1^2*k2^2-4*k2^3+k3^2+6*k1*k2*k3;
alfa1=(k1*k2+k3-sqrt(be))/(2*(k1^2+k2));
alfa2=(k1*k2+k3+sqrt(be))/(2*(k1^2+k2));
beta=(k1+alfa2)/(alfa1-alfa2); %(2*k1^3+3*k1*k2+k3-sqrt(be))/(sqrt(be));
% alfa2= (k3 + k1*k2 + (4*k1^3*k3 - 3*k1^2*k2^2 + 6*k1*k2*k3 - 4*k2^3 +
k3^2)^(1/2))/(2*(k1^2 + k2));
% alfa1= (k3 + k1*k2 - (4*k1^3*k3 - 3*k1^2*k2^2 + 6*k1*k2*k3 - 4*k2^3 +
k3^2)^(1/2))/(2*(k1^2 + k2));
T1_ang=-t_t1/log(alfa1);
T2_ang=-t_t1/log(alfa2);
T3_ang=beta*(T1_ang-T2_ang)+T1_ang;
T1(ii)=T1_ang;
T2(ii)=T2_ang;
T3(ii)=T3_ang;
T3_ang=sum(T3/length(T3));
T2_ang=sum(T2/length(T2));
```

```
T1_ang=sum(T1/length(T1));
%ec_carac= conv([T1_ang 1],[T2_ang 1]);
sys_G_ang = tf([T3_ang 1],conv([T1_ang 1],[T2_ang 1]))
sys_G_ang = sys_G_ang *K
[y,t0] = step(sys_G_ang,opt);
%----Calculo de valores R,L,C
disp('Calculo de valores R,L,C');
ft_denominador = sys_G_ang.Denominator{1,1}/sys_G_ang.Denominator{1,1}(1);
%desnormalizo
% para la corriente para la corriente
% factor_de_escalado = 1;
%
L=1/((sys_G_ang.Numerator{1,1}(2)/sys_G_ang.Denominator{1,1}(1))*factor_de_escala
do)
% R = (ft_denominador(2)*L)*factor_de_escalado
% C = 1/((ft_denominador(3)*L)*factor_de_escalado)

%tomando un valor de resistencia de resitencian
R=10
L=R/(ft_denominador(2))
C=1/(L*ft_denominador(3))
sys_G_obte = tf(K*[1/(L*C)],[1 (R/L) 1/(C*L)])
[y1,t1] = step(sys_G_obte,opt);

L_come = 4.5e-3; %no es un valor comercial pero se podria diseñar
R_come = 10;
C_come = 2200e-6;

sys_G_obte_come = tf(K*[1/(L_come*C_come)],[1 (R_come/L_come) 1/(C_come*L_come)])
[y2,t2] = step(sys_G_obte_come,opt);

figure(1)
plot(t,Vc_t,'r');hold on;
plot(t0,y);hold on;
plot(t1,y1);hold on;
plot(t2,y2);hold on;
title('Tension Vc_t');legend('repuesta original','respuesta del
modelo','respuesta con valores RLC no comerciales','Respuesta con valores RLC
comerciales');
```

Caso de estudio dos:

```
clc;clear;%close all;
ii=0;t_etapa=1e-6;wRef=200;tF=0.5;
tp=0.05;%tiempo de la pertubacion
%Constantes del PID
%Kp=0.500;Ki=0.001;Kd=0.0001;color_='r';
```

```
%Kp=1;Ki=0;Kd=0.0001;color_='k';
Kp=9;Ki=1;Kd=0;color_='b';
Ts=t_etapa;
A1=((2*Kp*Ts)+(Ki*(Ts^2))+(2*Kd))/(2*Ts);
B1=(-2*Kp*Ts+Ki*(Ts^2)-4*Kd)/(2*Ts);
C1=Kd/Ts;
e=zeros(round(tF/t_etapa),1);u=15;max_u=12;
input=zeros(round(tF/t_etapa),1);
delta_Tl=1e-7;
Tl=2.09e-5; %torque maximo de 2.1e-5
for i=0:1:7
    X=-[0; 0;0;0];
    ii=0;
    x1=0;
    x2=0;
    x3=0;
    x4=0;
    for t=0:t_etapa:tF
        ii=ii+1;k=ii+2;
        X=modmotor(t_etapa, X, u,Tl);
    %     e(k)=wRef-X(1); %ERROR
    %      if t>=tp
    %          wRef=2;
    %      end
    %     delta_u = A1*e(k)+B1*e(k-1)+C1*e(k-2);
        delta_u=0;
    %      if delta_u > max_u || delta_u < -max_u
    %          detal_u = 0;
    %      end
        input(ii)=wRef;
        %e(k)=wRef-X(4); %ERROR
        %u=u+A1*e(k)+B1*e(k-1)+C1*e(k-2); %PID
        u=u+delta_u; %PID
        x1(ii)=X(1);%Omega
        x2(ii)=X(2);%wp
        x3(ii)=X(3);%ia
        x4(ii)=X(4);%theta
        %u=max_u * tanh(u/max_u); %para saturar la accion de control
        acc(ii)=u;
    end
    tl(i+1)=Tl;
    Tl=Tl+delta_Tl
    t=0:t_etapa:tF;
    subplot(3,1,1);hold on;
    plot(t,x1);title('Salida y, \omega_t');
    subplot(3,1,2);hold on;
  %plot(t,x4,'r');title('Salida y, \theta_t');
```

```
  plot(t,x3);title('Corriente de salida, i_a');
end
 legend(strcat('Torque= ',num2str(tl')));
 subplot(3,1,3);hold on;
 plot(t,acc,'b');title('Entrada u_t, v_a');
 xlabel('Tiempo [Seg.]');
```

```
%-----------punto 3--------------
clc;clear all;close all;
num = xlsread('Curvas_Medidas_Motor.xls');
limite=6324;
t=num(1:end,1);
omega_t=num(1:end,2);
i_t=num(1:end,3);
opt = stepDataOptions;
opt.StepAmplitude = 12;
% %%%%%%%% metodo chena %%%%%%%%%%%%%
t_inic=0.0001
%t_inic=0.06
[val, lugar] = min(abs(t_inic-t)); %obtengo punto a punto el valor mas proximo al
t_inic y obtengo el min punto del vector t
y_t1=omega_t(lugar);
t_t1=t(lugar);
ii=0;
ii=ii+1;
[val, lugar] =min(abs(2*t_inic-t));
t_2t1=t(lugar);
y_2t1=omega_t(lugar);
[val, lugar] =min(abs(3*t_inic-t));
t_3t1=t(lugar);
y_3t1=omega_t(lugar);

K=omega_t(end)/opt.StepAmplitude

k1=(1/opt.StepAmplitude)*y_t1/K-1; %Afecto el valor del Escalon
k2=(1/opt.StepAmplitude)*y_2t1/K-1;
k3=(1/opt.StepAmplitude)*y_3t1/K-1;

be=4*k1^3*k3-3*k1^2*k2^2-4*k2^3+k3^2+6*k1*k2*k3;
alfa1=(k1*k2+k3-sqrt(be))/(2*(k1^2+k2));
alfa2=(k1*k2+k3+sqrt(be))/(2*(k1^2+k2));
beta=(k1+alfa2)/(alfa1-alfa2); %(2*k1^3+3*k1*k2+k3-sqrt(be))/(sqrt(be));
% alfa2= (k3 + k1*k2 + (4*k1^3*k3 - 3*k1^2*k2^2 + 6*k1*k2*k3 - 4*k2^3 +
k3^2)^(1/2))/(2*(k1^2 + k2));
```

```
% alfa1= (k3 + k1*k2 - (4*k1^3*k3 - 3*k1^2*k2^2 + 6*k1*k2*k3 - 4*k2^3 +
k3^2)^(1/2))/(2*(k1^2 + k2));
T1_ang=-t_t1/log(alfa1);
T2_ang=-t_t1/log(alfa2);
T3_ang=beta*(T1_ang-T2_ang)+T1_ang;
T1(ii)=T1_ang;
T2(ii)=T2_ang;
T3(ii)=T3_ang;
T3_ang=sum(T3/length(T3));
T2_ang=sum(T2/length(T2));
T1_ang=sum(T1/length(T1));
%ec_carac= conv([T1_ang 1],[T2_ang 1]);
sys_G_ang = tf([T3_ang 1],conv([T1_ang 1],[T2_ang 1]))
sys_G_ang = sys_G_ang *K
[y,t0] = step(sys_G_ang,opt,0.6);
%obtengo el los valores de simulacion desde simulink
%correr antes el archivo modelo_motor_con_pertubacion en simulink
t1 = curba_omega.time;
y1 = curba_omega.signals.values;
%FT para la pertubacion TL
Tl=7.5e-2;
Laa=336e-6;
Ra=55.6;
Gt=tf(Tl*[Laa Ra],sys_G_ang.Denominator{1});

figure(1)
plot(t,omega_t,'r');hold on;
plot(t1,y1);
%plot(t0,y);hold on;
legend('modelo original','modelo obtenido');
```

```
%PID
clc;clear all;close all;
ii=0;t_etapa=1e-6;wRef=2;tF=0.5;X=-[0; 0;0;0];
tp=0.05;%tiempo de la pertubacion
%Constantes del PID
%Kp=0.1;Ki=0.01;Kd=5;color_='r';
Kp=1;Ki=1;Kd=0;color_='k';
%Kp=9;Ki=1;Kd=0;color_='b';
Ts=t_etapa;
A1=((2*Kp*Ts)+(Ki*(Ts^2))+(2*Kd))/(2*Ts);
B1=(-2*Kp*Ts+Ki*(Ts^2)-4*Kd)/(2*Ts);

C1=Kd/Ts;
e=zeros(round(tF/t_etapa),1);u=0;max_u=12;
```

```
input=zeros(round(tF/t_etapa),1);
delta_Tl=1e-7;
Tl=0;%2.09e-5; %torque maximo de 2.1e-5
ii=0;
for t=0:t_etapa:tF
    ii=ii+1;k=ii+2;
    X=modmotor(t_etapa, X, u,Tl);
    e(k)=wRef-X(4); %ERROR
    %u=u+A1*e(k)+B1*e(k-1)+C1*e(k-2); %PID
    delta_u = A1*e(k)+B1*e(k-1)+C1*e(k-2);
    %     satura la accion de control
    %     if delta_u > max_u || delta_u < -max_u
    %         detal_u = 0;
    %     end
    u=u+delta_u; %PID
    %u=max_u * tanh(u/max_u); %para saturar la accion de control
    x1(ii)=X(1);%Omega
    x2(ii)=X(2);%wp
    x3(ii)=X(3);%ia
    x4(ii)=X(4);%theta
    acc(ii)=u;
end
t=0:t_etapa:tF;
figure(1)
subplot(2,1,1);hold on;
plot(t,x4);title('\theta_t');
subplot(2,1,2);hold on;
plot(t,x1);title('\omega_t');
xlabel('Tiempo [Seg.]');

figure(2)
subplot(2,1,1);hold on;
plot(t,x3);title('Corriente');
subplot(2,1,2);hold on;
plot(t,acc);title('accion de control, u_t');
xlabel('Tiempo [Seg.]');
```

```
%motor
function [X]=modmotor(t_etapa, xant, accion,Tl)
Laa=366e-6; J=5e-9;Ra=55.6;B=0;Ki=6.49e-3;Km=6.53e-3;
Va=accion;
h=1e-7;TL=Tl;
omega= xant(1);
w= xant(2);
ia=xant(3);
```

```
theta = xant(4);

for ii=1:t_etapa/h
    wpp =(-wp*(Ra*J+Laa*B)-omega*(Ra*B+Ki*Km)+Va*Ki)/(J*Laa);
    iap=(-Ra*ia-Km*omega+Va)/Laa;
    wp=wp+h*wpp;
    wp=wp-((1/J)*TL);%torque
    ia=ia+iap*h;
    omega = omega + h*wp;
    thetap = omega;
    theta = theta + h*thetap;
end
X=[omega,wp,ia,theta];
```

Caso de estudio 3

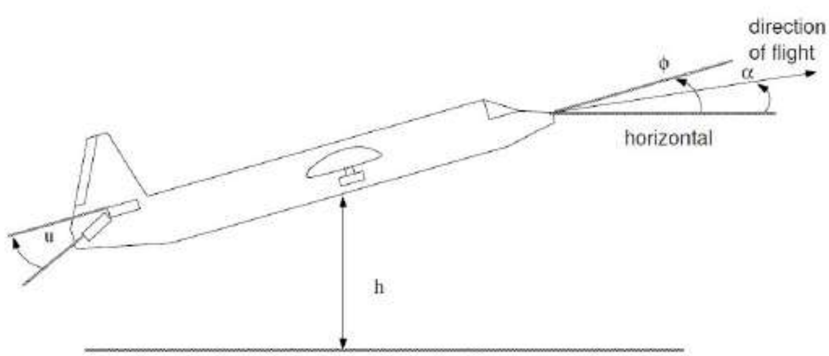


Fig. 1-4. Modelo de sistema de altitud en un avión, extraído de [1].

Para el caso de la Fig. 1-4, modelo válido sólo para pequeños ángulos, se tiene

$$\begin{cases} \dot{\alpha} = a(\phi - \alpha) \\ \ddot{\phi} = -\omega^2(\phi - \alpha - b \cdot u) \\ \dot{h} = c\alpha \end{cases} \qquad (1\text{-}8)$$

ω representa la frecuencia natural, y los coeficientes a b son constantes positivas, u es la variables manipulada (**timón de profundidad**) y es proporcional a la posición de los elevadores, ϕ (**ángulo de cabeceo**) en radianes, vuela a c metros por segundos, su trayectoria de vuelo forma un ángulo α con la horizontal ($\alpha > 0$ $sube, si$ $\alpha <$ 0 $desciende$).

1. Obtener el sistema lineal en variables de estado para el equilibrio $x_{op} = [0 \quad 0 \quad 0 \quad 0]^T$.

Desarrollo:
Tomando como vector de estado $x = [\alpha \quad \phi \quad \dot{\phi} \quad h]^T$.
Aplicando:

$$\begin{cases} \dot{x} = x * A + B * u \\ y = x * C + D * u \end{cases}$$

con:

$$A = \begin{pmatrix} \frac{\partial \dot{\alpha}}{\partial \alpha} & \frac{\partial \dot{\alpha}}{\partial \phi} & \frac{\partial \dot{\alpha}}{\partial \dot{\phi}} & \frac{\partial \dot{\alpha}}{\partial h} \\ \frac{\partial \dot{\phi}}{\partial \alpha} & \frac{\partial \dot{\phi}}{\partial \phi} & \frac{\partial \dot{\phi}}{\partial \dot{\phi}} & \frac{\partial \dot{\phi}}{\partial h} \\ \frac{\partial \ddot{\phi}}{\partial \alpha} & \frac{\partial \ddot{\phi}}{\partial \phi} & \frac{\partial \ddot{\phi}}{\partial \dot{\phi}} & \frac{\partial \ddot{\phi}}{\partial h} \\ \frac{\partial \dot{h}}{\partial \alpha} & \frac{\partial \dot{h}}{\partial \phi} & \frac{\partial \dot{h}}{\partial \dot{\phi}} & \frac{\partial \dot{h}}{\partial h} \end{pmatrix}_{x_{op},u_{op}} \quad ; B = \begin{pmatrix} \frac{\partial \dot{\alpha}}{\partial u} \\ \frac{\partial \dot{\phi}}{\partial u} \\ \frac{\partial \ddot{\phi}}{\partial u} \\ \frac{\partial \dot{h}}{\partial u} \end{pmatrix}_{x_{op},u_{op}}$$

Tomando como salida solo $\alpha, \phi \ y \ h$.

$$C = \begin{pmatrix} \frac{\partial \alpha}{\partial \alpha} & \frac{\partial \alpha}{\partial \phi} & \frac{\partial \alpha}{\partial \dot{\phi}} & \frac{\partial \alpha}{\partial h} \\ \frac{\partial \phi}{\partial \alpha} & \frac{\partial \phi}{\partial \phi} & \frac{\partial \phi}{\partial \dot{\phi}} & \frac{\partial \phi}{\partial h} \\ \frac{\partial h}{\partial \alpha} & \frac{\partial h}{\partial \phi} & \frac{\partial h}{\partial \dot{\phi}} & \frac{\partial h}{\partial h} \end{pmatrix}_{X_{op},u_{op}} \quad ; D = \begin{pmatrix} \frac{\partial \alpha}{\partial u} \\ \frac{\partial \phi}{\partial u} \\ \frac{\partial h}{\partial u} \end{pmatrix}_{x_{op},u_{op}}$$

Desarrollando a través del simbólico de Matlab obtenemos los siguientes resultados:

```
Matriz A :
/ -a,   a,  0, 0 \
|                |
|  0,   0,  1, 0 |
|                |
|  2     2       |
| w , - w , 0, 0 |
|                |
\  c,   0,  0, 0 /

Reemplazando

A =

   -0.0500    0.0500         0         0
         0         0    1.0000         0
    4.0000   -4.0000         0         0
   50.0000         0         0         0

Matriz B :
/   0  \
|      |
|   0  |
|      |
|   0  |
|      |
|    2 |
| b w  |
|      |
\   0  /

Reemplazando

B =

     0
     0
    20
     0
```

Como la matriz C y D salen directo por pura inspección tenemos:

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} ; D = 0.$$

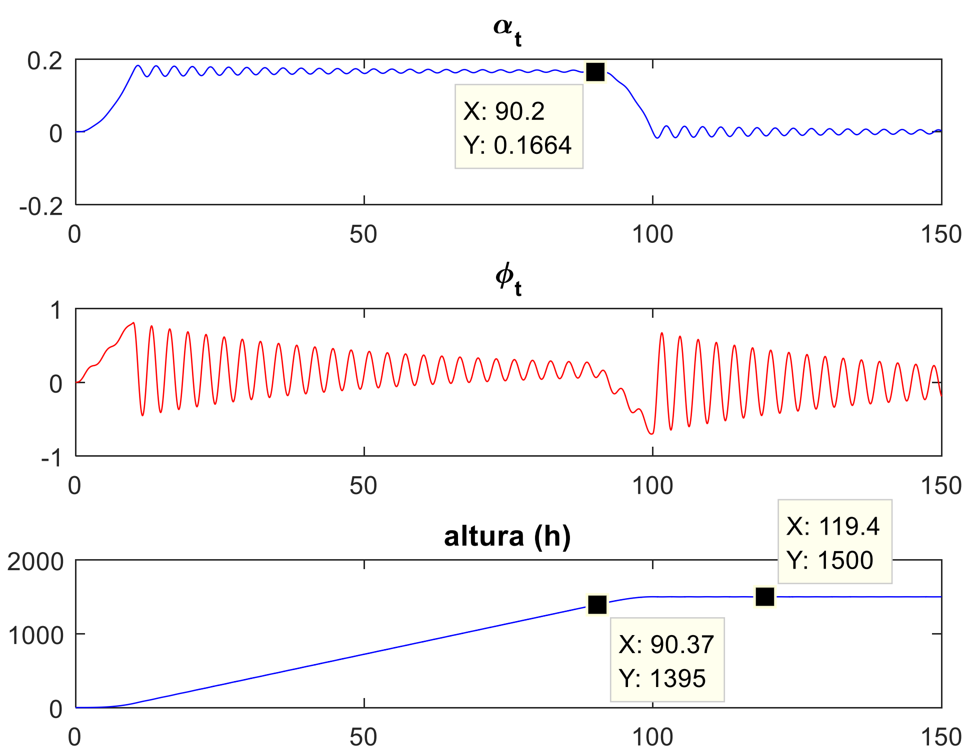
2. Obtener la solución numérica del sistema lineal para evaluar cuantitativamente el comportamiento con intención de verificar el correcto planteo. Para hacerlo, se le asignan los valores siguientes a los parámetros, son ω=2; a=0,05; b=5; c=100 m/s, (es decir, 360Km/h), $\Delta t = 10^{-3}$; y el tiempo de simulación de 5 segundos.

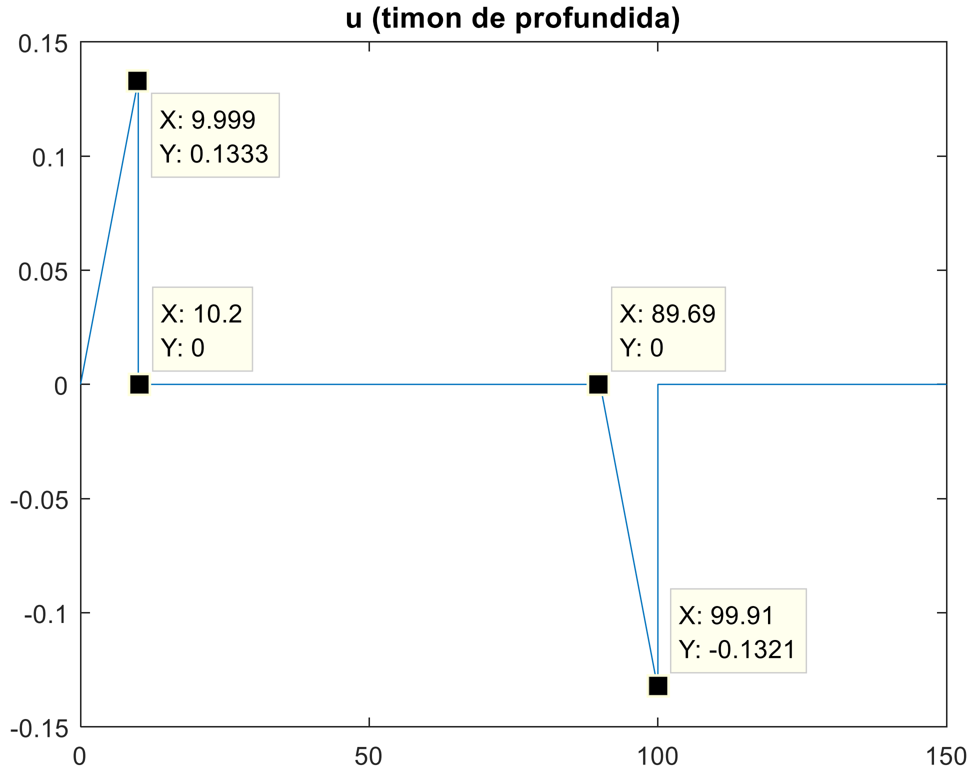
Desarrollo:

Una primera simulación con un tiempo de 150 seg. Vemos que si la variables u (timón de profundidad) arranca en 0 y se va moviendo lentamente hacia arriba hasta los 10 seg, después pasa a 0.

El ángulo α es positivo en aumento durante los 10 seg, por lo que el avión cada vez está a mayor altura. El ángulo de cabeceo ϕ también indica que el avión está subiendo. Igual mente se ven las oscilaciones que se tienen dada por la respuesta, hasta que se estabiliza como se observa se van haciendo cada vez más chica con el tiempo.

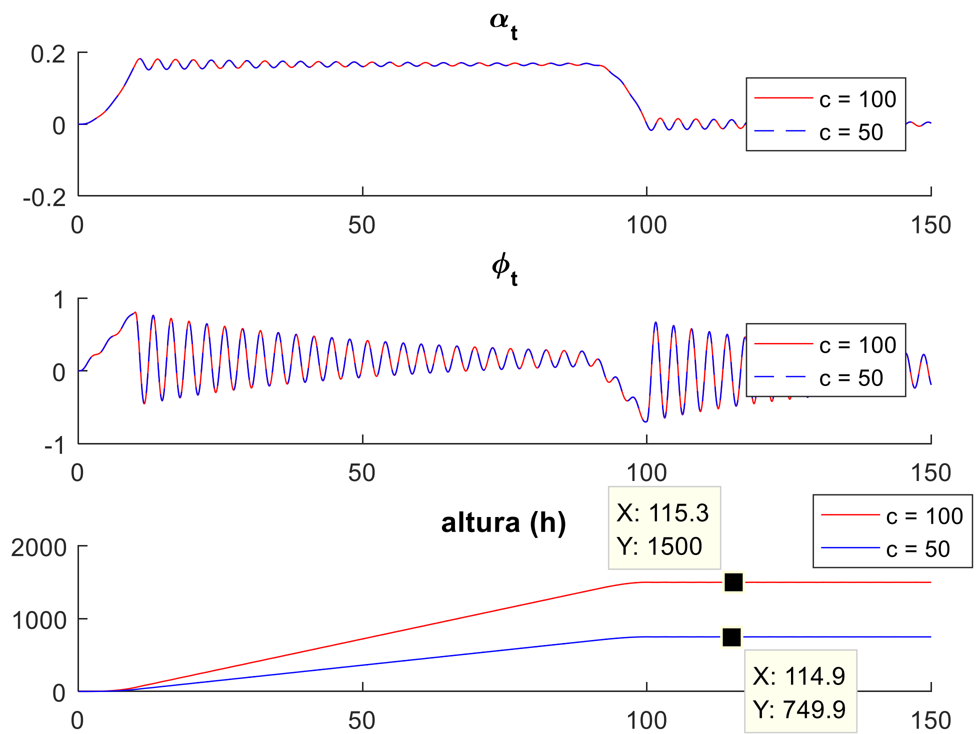
A los 90 seg u empieza a moverse hacia abajo, y se tiene que α empieza a bajar por lo que el avión pierde altura, como se ve que en 90 seg la pendiente en la altura disminuye, hasta que se llega a los 100 seg donde u vuelve a cero y el avión queda a una cierta altura.





3. Obtener la solución numérica del sistema lineal para c=50 m/s, (es decir, 180Km/h), $\Delta t = 10^{-3}$ y el tiempo de simulación de 20 segundos.

Como es de esperarse a disminuir c entonces el avión llega a menos altura en los mismo puntos que en la anterior simulación, y las frecuencias de oscilación siguen igual.



Lecciones aprendidas:

- Entender cómo funciona el timón de profundidad de un avión y un poco mejor sus ejes.
- Mejor manejo de la herramienta simbólica de Matlab.
- Mayor comprensión cuantitativa de los resultados.
- Obtención del modelo en variables de estados.

Complicaciones:

- Entendimiento de las variables de interés del modelo.
  Algunos links útiles fueron:
  https://es.wikipedia.org/wiki/Tim%C3%B3n_de_profundidad
  https://www.sociedadaeronautica.org/control-aviones/
  https://es.wikipedia.org/wiki/Ejes_del_avi%C3%B3n#:~:text=El%20eje%20lateral%20o%20transversal,trav%C3%A9s%20del%20tim%C3%B3n%20de%20profundidad.
- Herramienta simbólica de Matlab.

Código utilizado

```
%simbolico
clc;clear all
syms u alfa alfa_p h h_p fi fi_p fi_pp a b c w
alfa_p = a*(fi-alfa);
```

```
fi_pp = -w^2*(fi - alfa - b*u);
h_p = c*alfa;

A =
[[subs(subs(subs(subs(diff(alfa_p,alfa),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(alfa_p,fi),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(alfa_p,fi_p),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(alfa_p,h),'alfa',0),'fi',0),'fi_p',0),'h',0)];
     [0 0 1 0];
     [subs(subs(subs(subs(diff(fi_pp,alfa),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(fi_pp,fi),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(fi_pp,fi_p),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(fi_pp,h),'alfa',0),'fi',0),'fi_p',0),'h',0)];
     [subs(subs(subs(subs(diff(h_p,alfa),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(h_p,fi),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(h_p,fi_p),'alfa',0),'fi',0),'fi_p',0),'h',0),...
     subs(subs(subs(subs(diff(h_p,h),'alfa',0),'fi',0),'fi_p',0),'h',0)];
    ];

B =[subs(subs(subs(subs(diff(alfa_p,u),'alfa',0),'fi',0),'fi_p',0),'h',0);
    0;
    subs(subs(subs(subs(diff(fi_p,u),'alfa',0),'fi',0),'fi_p',0),'h',0);
    subs(subs(subs(subs(diff(fi_pp,u),'alfa',0),'fi',0),'fi_p',0),'h',0);
    subs(subs(subs(subs(diff(h_p,u),'alfa',0),'fi',0),'fi_p',0),'h',0)];

disp('Matriz A :')
pretty(simplify(A))
disp('Reemplazando')
a=0.05;b=5;c=50;w=2;
A=[-a a 0 0;
    0 0 1 0;
    w^2 -w^2 0 0;
    c 0 0 0]
disp('Matriz B :')
pretty(simplify(B))
disp('Reemplazando')
B=[0;
   0;
   b*w^2;
   0]
```

```
%variables
clc;clear all;close all;
T=150; At=1e-3; Kmax=T/At; t=linspace(0,T,Kmax);
a=0.05;b=5;c=50;w=2;
alfa_p=0;fi_p=0;fi_pp=0;h_p=0;
alfa=zeros(1,int32(Kmax));fi=zeros(1,int32(Kmax));fi_p=zeros(1,int32(Kmax));h=zer
os(1,int32(Kmax));
u=linspace(0,0,int32(Kmax));
A=[-a a 0 0;
    0 0 1 0;
    w^2 -w^2 0 0;
    c 0 0 0];
B=[0;
   0;
   b*w^2;
   0];
% C=[1 0 0 0;
%    0 1 0 0;
%    0 0 0 1]
% D=0;
%condiciones iniciales
alfa(1)=0;fi(1)=0;fi_p(1)=0;h(1)=0;u(1)=0;
Xop=[0 0 0 0]';x=[alfa(1) fi(1) fi_p(1) h(1)]';
alfa_l(1)=0;fi_l(1)=0;fi_p_l(1)=0;h_l(1)=0;
ii = 0;
flag=0;
flag1=0;
for i=1:Kmax-1
     ii=ii+At;
     if(ii>=100)
         u(i)=0;
         flag1 = 1;
     end
     if(ii>=90 && ~flag1)
         u(i) = u(i-1) - 2*(1/Kmax);
%          if(u(i)<0)
%              u(i)=0;
%          end
         flag = 1;
     end
     if(ii>=10 && ~flag)
         u(i) = 0;
     else
         u(i+1)=u(i)+ 2*(1/Kmax);
     end
     alfa_p    = a*(fi(i) - alfa(i));
     fi_pp     = (-w^2)*(fi(i)-alfa(i)-(b*u(i)));
```

```
    h_p        = c*alfa(i);
    alfa(i+1)  = alfa(i) + alfa_p*At;
    fi_p(i+1)  = fi_p(i) + fi_pp*At;
    fi(i+1)    = fi(i) + fi_p(i)*At;
    h(i+1)     = h(i) + h_p*At;
    %variables de sistema lineal
    xp         =A*(x-Xop)+B*u(i);
    x          =x+xp*At;
    %Y=C*x;
    alfa_l(i+1)=x(1);
    fi_l(i+1)  =x(2);
    fi_p_l(i+1)=x(3);
    h_l(i+1)   =x(4);
end
figure(1)
subplot(3,1,1);%hold on;
plot(t,alfa_l,'b');title('\alpha_t');
subplot(3,1,2);%hold on;
plot(t,fi_l,'r');title('\phi_t');
% subplot(4,1,3);%hold on;
% plot(t,fi_p_l,'b');title('fi_p');
subplot(3,1,3);%hold on;
plot(t,h_l,'b');title('altura (h)');
figure(2)
plot(t,u);title('u (timon de profundida)');
```

Caso de estudio 4

Caso de estudio 4. Sistema no lineal de cuatro variables de estado

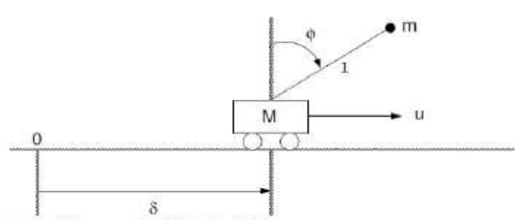


Fig. 1-5. Sistemas para modelar, extraído de [1].

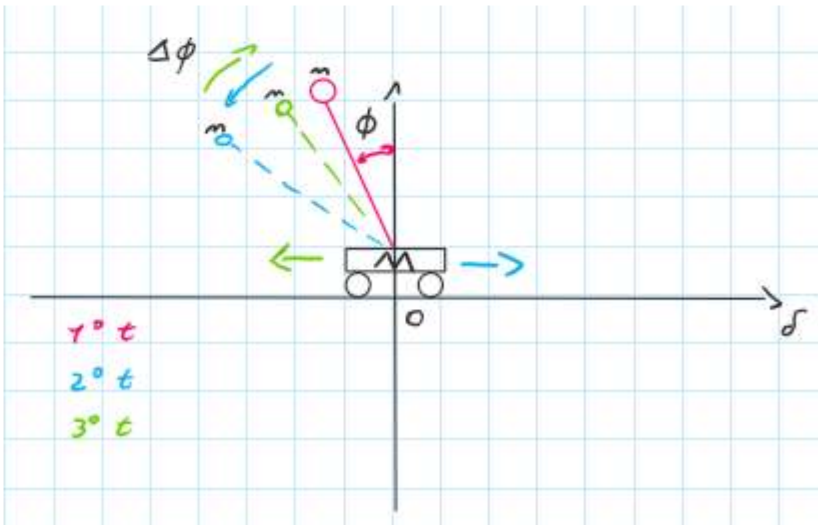
Para el caso del esquema del péndulo invertido de la Fig. 1-5 se tiene,

$$\begin{cases}(M+m)\ddot{\delta}+ml\ddot{\phi}\cos\phi-ml\dot{\phi}^2\operatorname{sen}\phi+F\dot{\delta}=u \\ l\ddot{\phi}-g\operatorname{sen}\phi+\ddot{\delta}\cos\phi=0\end{cases} \tag{9}$$

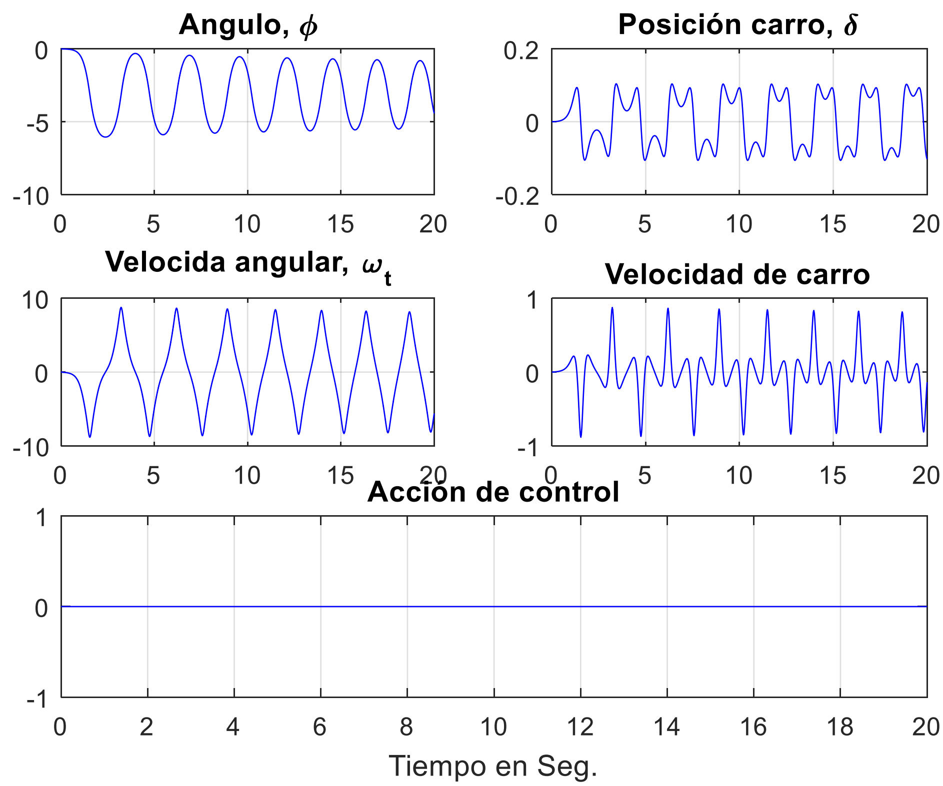
donde el sistema lineal en variables de estado $x=[\delta \quad \dot{\delta} \quad \phi \quad \dot{\phi}]^T$, con los valores de los coeficientes de m=0,1; F=0,1; l=0,6; g=9,8; M=0,5 y $\Delta t=10^{-4}$seg, tomando un tiempo de simulación de 5 segundos con u=0.

1. Obtener simulaciones del sistema (9) en las condiciones iniciales $x_{op} = (0 \quad 0 \quad -0.01 \quad 0)^T$ y $x_{op} = (0 \quad 0 \quad 3.2 \quad 0)^T$, empleando una integración Euler con $\Delta t = 10^{-4}$. El tiempo de simulación será de 10 segundos en cada caso, con u=0.
   Desarrollo:

$$Con: \mathrm{x_{op}} = (0 \quad 0 \quad -0.01 \quad 0)^{\mathrm{T}}$$

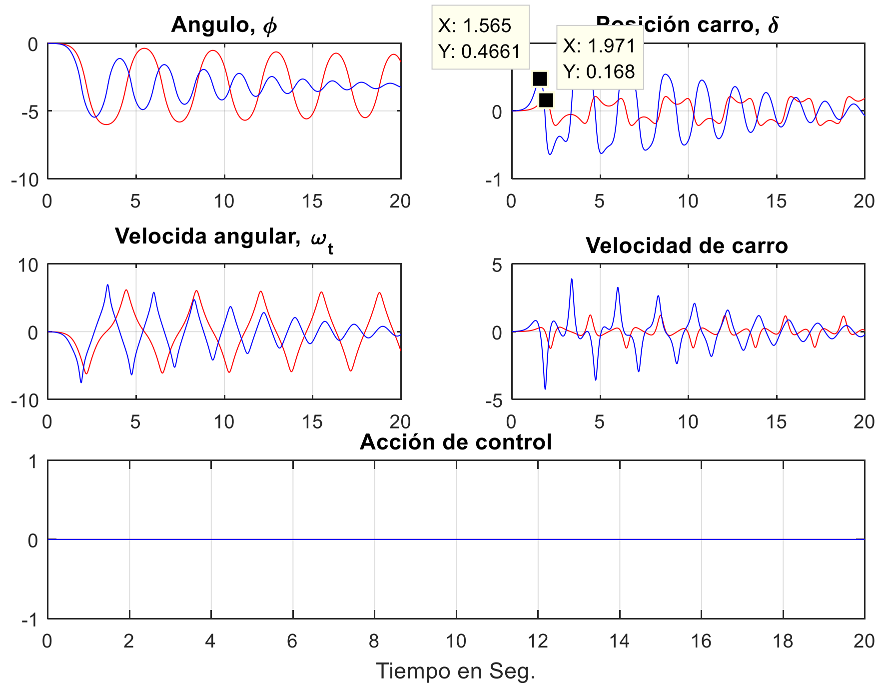


Esquema simplificado con tres tiempos distintos del desplazamiento angular.

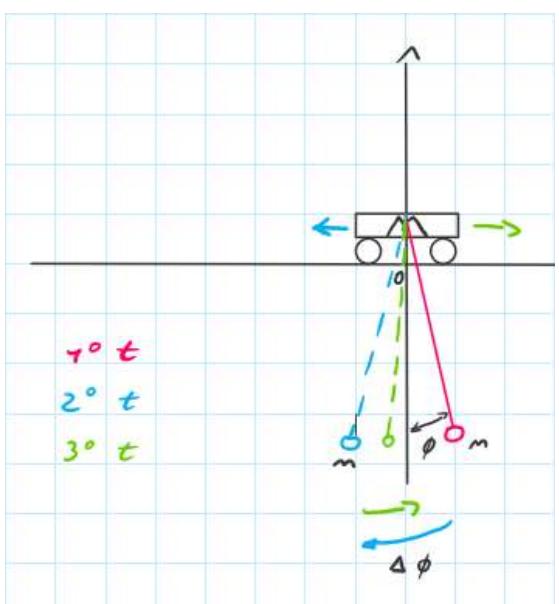


Vemos que debido a que la masa chica empieza a caer por lo que el ángulo crece negativamente, debido a sus fuerzas que produce sobre el eje horizontalmente hace que la masa grande (carro) se desplace un $\Delta\delta$ hacia la derecha, después se ve como el carro trata de estabilizar el péndulo haciendo un cambio de dirección rápido y el ángulo empieza a decrementar, hasta que se vuelve a repetir lo anterior y la posición del carro nunca se estabiliza.

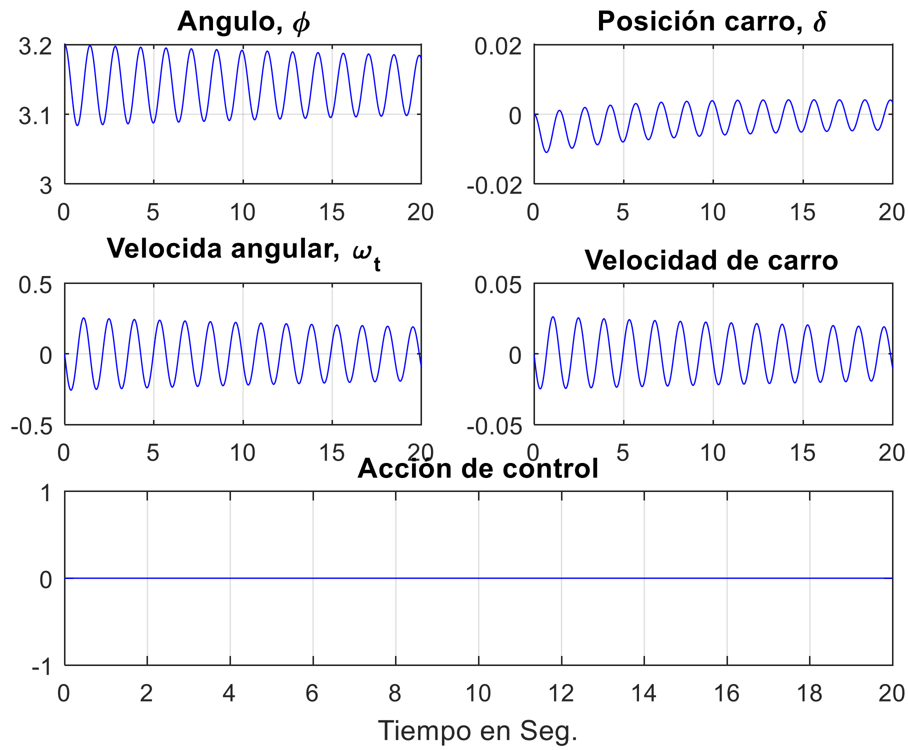
Podemos hacer una simulación con una M más chica y ver como se obtiene un desplazamiento más grande al inicio, visto en la siguiente figura la gráfica azul.



$$Con: \; x_{op} = (0 \quad 0 \quad 3.2 \quad 0)^T$$



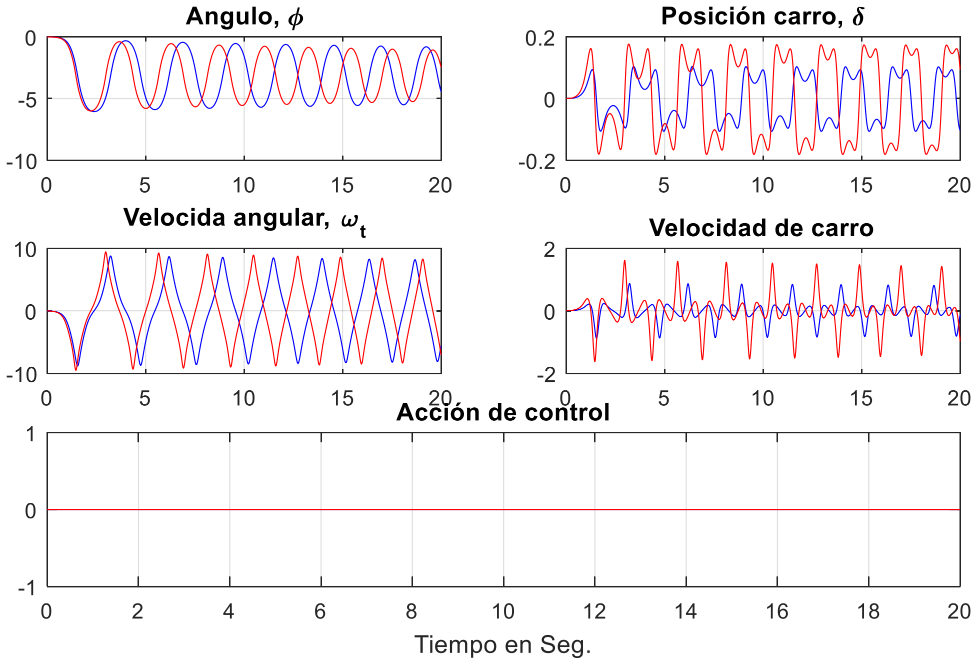
Esquema simplificado con 3 tiempos distintos del desplazamiento angular.



Con un ángulo inicial más grande se observa una oscilación más suave, con menores recorridos de la posición del carro y el ángulo esto se debe a que está más cerca del equilibrio.

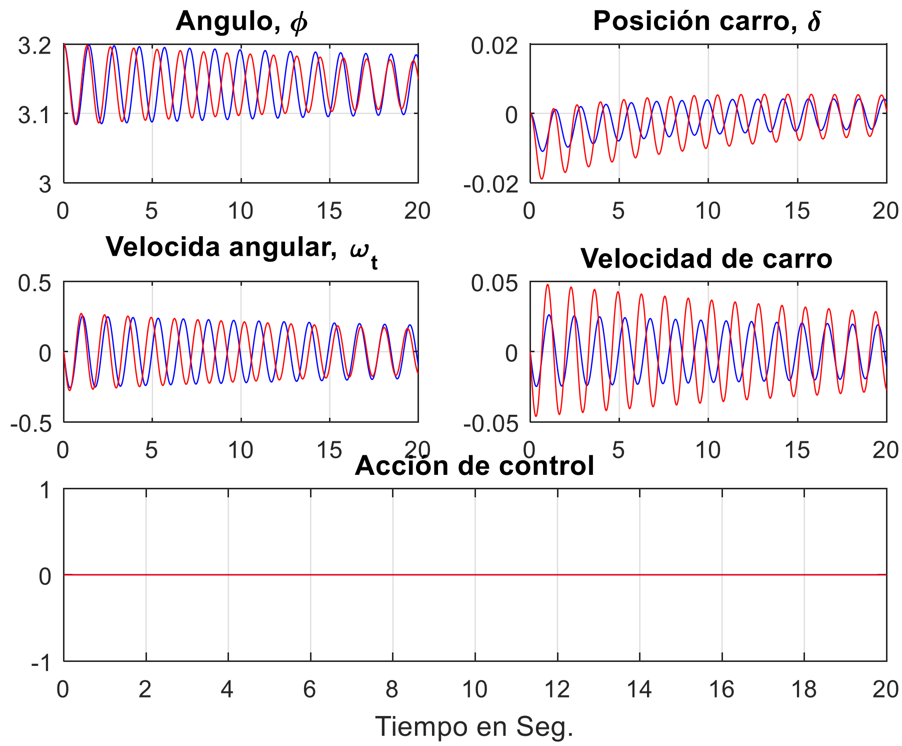
2. Modificar la masa *m* al doble y repetir la operación.

$$Con: \mathrm{x_{op}} = (0 \quad 0 \quad -0.01 \quad 0)^{\mathrm{T}}$$



Tenemos en rojo un mayor desplazamiento de la masa grande (carro), un aumento de la frecuencia de oscilación, esto porque al ser la masa más grande genera una mayor aceleración en el carro.

$$Con: \mathrm{x_{op}} = (0 \quad 0 \quad 3.2 \quad 0)^{\mathrm{T}}$$



Se obtienen los mismos resultados anterior mente.

3. Obtener la representación lineal en variables de estado para el equilibrio inestable.

Tomando como vector de estado $x = [\delta \quad \dot{\delta} \quad \phi \quad \dot{\phi}]^T, x_{op} = [0 \quad 0 \quad 0 \quad 0]$.
Aplicando:

$$\begin{cases} \dot{x} = x * A + B * u \\ y = x * C + D * u \end{cases}$$

$$A = \begin{pmatrix} \frac{\partial \dot{\delta}}{\partial \delta} & \frac{\partial \dot{\delta}}{\partial \dot{\delta}} & \frac{\partial \dot{\delta}}{\partial \phi} & \frac{\partial \dot{\delta}}{\partial \dot{\phi}} \\ \frac{\partial \ddot{\delta}}{\partial \delta} & \frac{\partial \ddot{\delta}}{\partial \dot{\delta}} & \frac{\partial \ddot{\delta}}{\partial \phi} & \frac{\partial \ddot{\delta}}{\partial \dot{\phi}} \\ \frac{\partial \dot{\phi}}{\partial \delta} & \frac{\partial \dot{\phi}}{\partial \dot{\delta}} & \frac{\partial \dot{\phi}}{\partial \phi} & \frac{\partial \dot{\phi}}{\partial \dot{\phi}} \\ \frac{\partial \ddot{\phi}}{\partial \delta} & \frac{\partial \ddot{\phi}}{\partial \dot{\delta}} & \frac{\partial \ddot{\phi}}{\partial \phi} & \frac{\partial \ddot{\phi}}{\partial \dot{\phi}} \end{pmatrix}_{x_{op},u_{op}} \quad ; B = \begin{pmatrix} \frac{\partial \dot{\delta}}{\partial u} \\ \frac{\partial \ddot{\delta}}{\partial u} \\ \frac{\partial \dot{\phi}}{\partial u} \\ \frac{\partial \ddot{\phi}}{\partial u} \end{pmatrix}_{x_{op},u_{op}}$$

$$C = \begin{pmatrix} \frac{\partial \delta}{\partial \delta} & \frac{\partial \dot{\delta}}{\partial \delta} & \frac{\partial \phi}{\partial \delta} & \frac{\partial \dot{\phi}}{\partial \delta} \\ \frac{\partial \dot{\delta}}{\partial \delta} & \frac{\partial \dot{\delta}}{\partial \dot{\delta}} & \frac{\partial \dot{\delta}}{\partial \phi} & \frac{\partial \dot{\delta}}{\partial \dot{\phi}} \\ \frac{\partial \phi}{\partial \delta} & \frac{\partial \phi}{\partial \dot{\delta}} & \frac{\partial \phi}{\partial \phi} & \frac{\partial \phi}{\partial \dot{\phi}} \\ \frac{\partial \ddot{\phi}}{\partial \delta} & \frac{\partial \ddot{\phi}}{\partial \dot{\delta}} & \frac{\partial \ddot{\phi}}{\partial \phi} & \frac{\partial \ddot{\phi}}{\partial \dot{\phi}} \end{pmatrix}_{X_{op},u_{op}} \qquad ; D = \begin{pmatrix} \frac{\partial \delta}{\partial u} \\ \frac{\partial \dot{\delta}}{\partial u} \\ \frac{\partial \phi}{\partial u} \\ \frac{\partial \dot{\phi}}{\partial u} \end{pmatrix}_{x_{op},u_{op}}$$

Obtenemos atreves del simbólico de Matlab[1]:

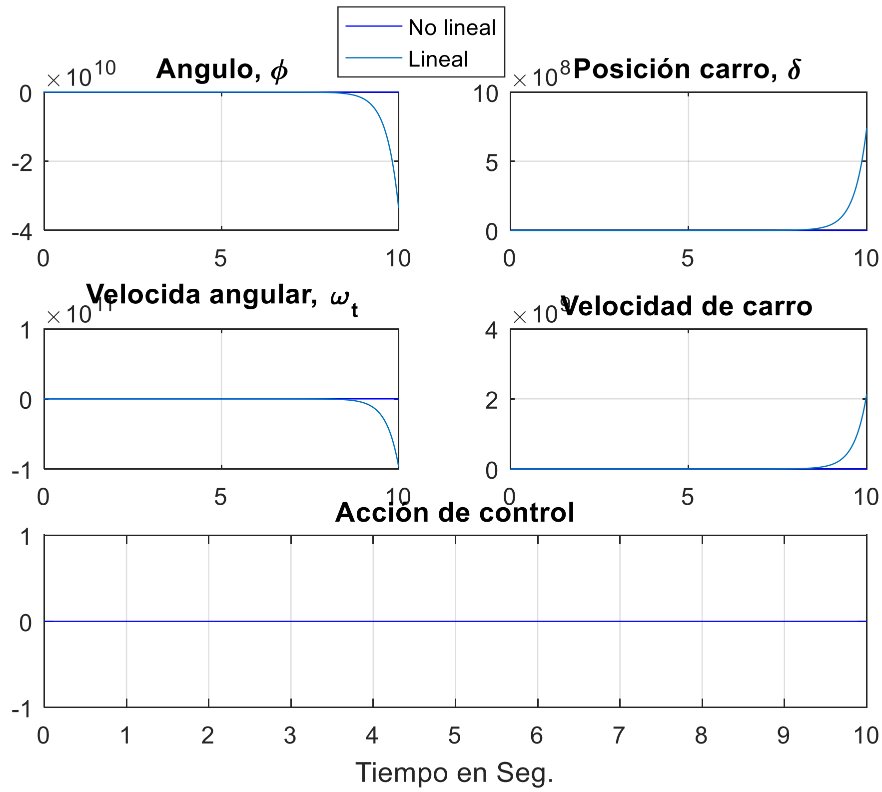
```
Matriz A:
/ 0,  1,       0,      0 \
|                        |
|       F      g m       |
| 0, - -,   - ---,     0 |
|       M       M        |
|                        |
| 0,  0,       0,      1 |
|                        |
|      F   g  (M + m)    |
| 0, ---,  ---------,  0 |
\     L M     L M        /

Matriz B:
/    0    \
|         |
|    1    |
|    -    |
|    M    |
|         |
|    0    |
|         |
|     1   |
|  - ---  |
\    L M  /
```

C y D salen de pura inspección.

[1] Los desarrollos de las ecuaciones, están en papel y lápiz, en caso de que haga falta se integran al trabajo.

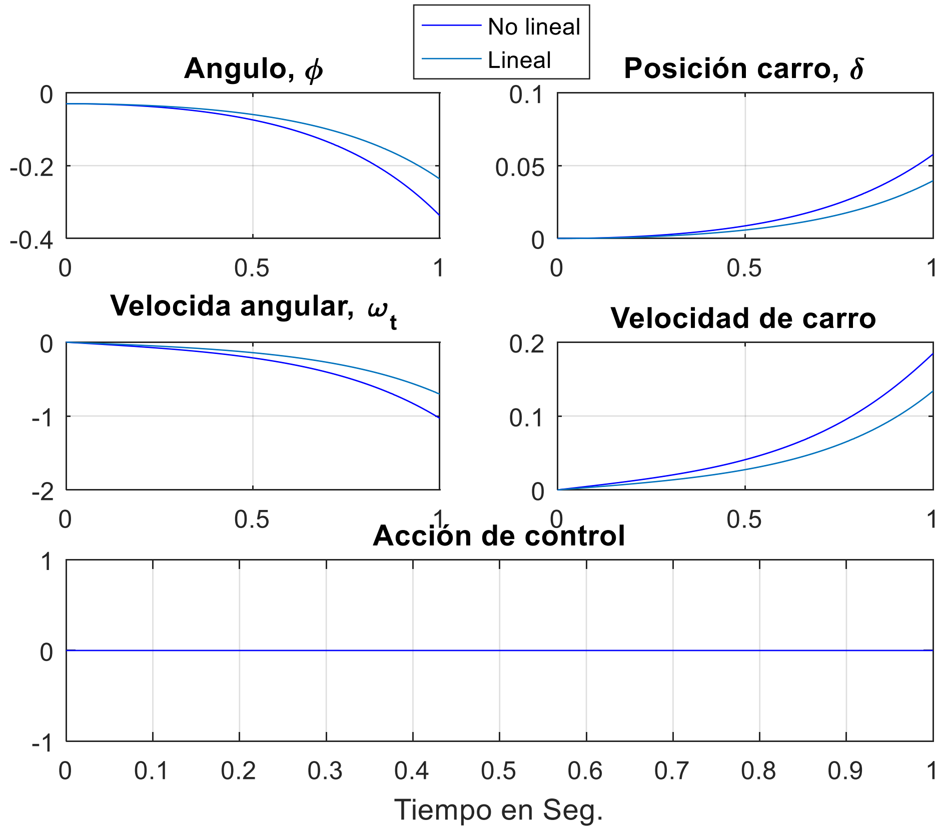
$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad ; D = (0)$$

4. Obtener la solución numérica de los dos sistemas, del lineal y del no lineal para evaluar cuantitativamente la equivalencia, modificando *m* de 0,1 a 0,01 y la longitud (long) a 1,2m.

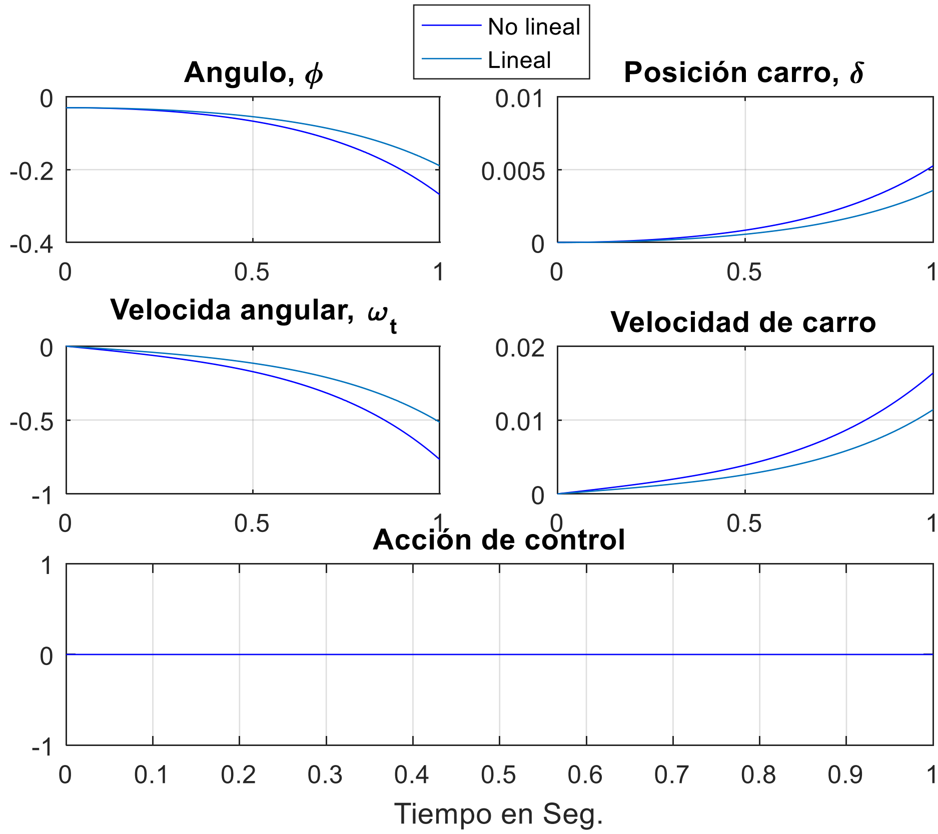


Como vemos el modelo lineal para el equilibrio inestable no responde muy bien si vemos el rango de valides haciendo el tiempo de simulación más corto tenemos:

$$m = 0.1$$


No lineal
Lineal
Angulo, ϕ
Posición carro, δ
Velocida angular, ω_t
Velocidad de carro
Acción de control
Tiempo en Seg.

$m = 0.01$



Como se ve el rango de valides a medida que pasa el tiempo es muy chico. Y con las disminuciones de la masa los valores de velocidad y posición se ven influenciados en igual medida.

5. Obtener el sistema lineal para el equilibrio estable.
   Siguiendo el mismo procedimiento que en el punto 3, peor con:

$$x_{op} = [0 \quad 0 \quad \pi \quad 0].$$

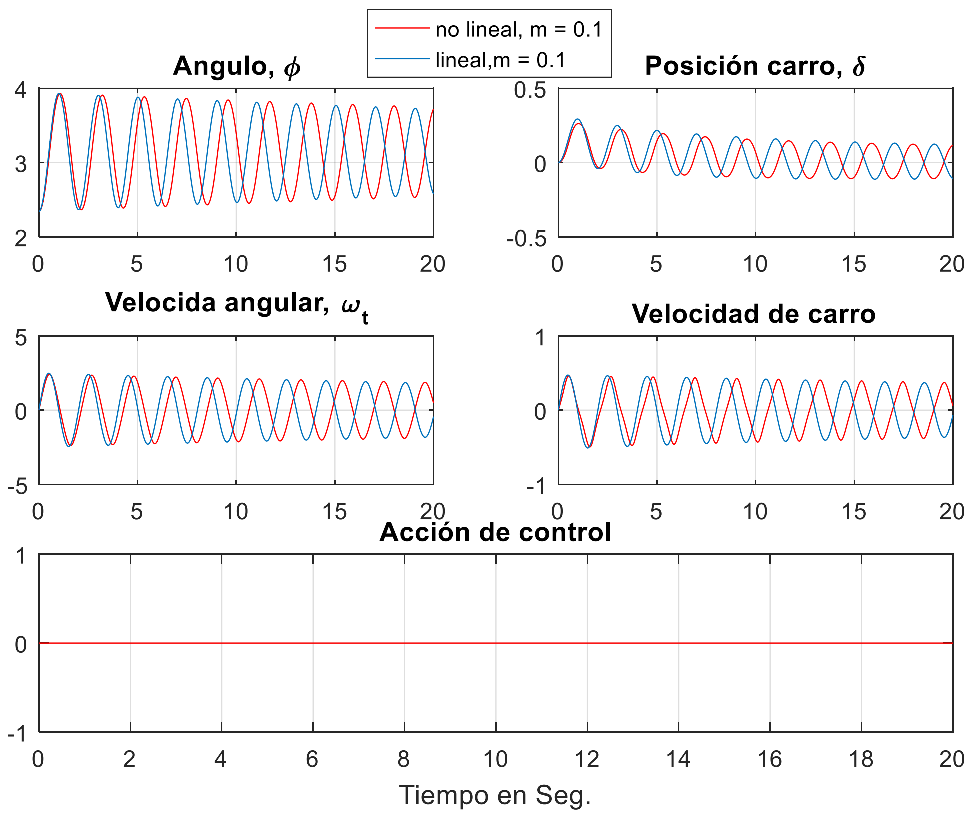
Matriz A:

$$\begin{pmatrix} 0, & 1, & 0, & 0 \\ 0, & -\frac{F}{M}, & -\frac{g\,m}{M}, & 0 \\ 0, & 0, & 0, & 1 \\ 0, & -\frac{F}{L\,M}, & -\frac{g\,(M+m)}{L\,M}, & 0 \end{pmatrix}$$

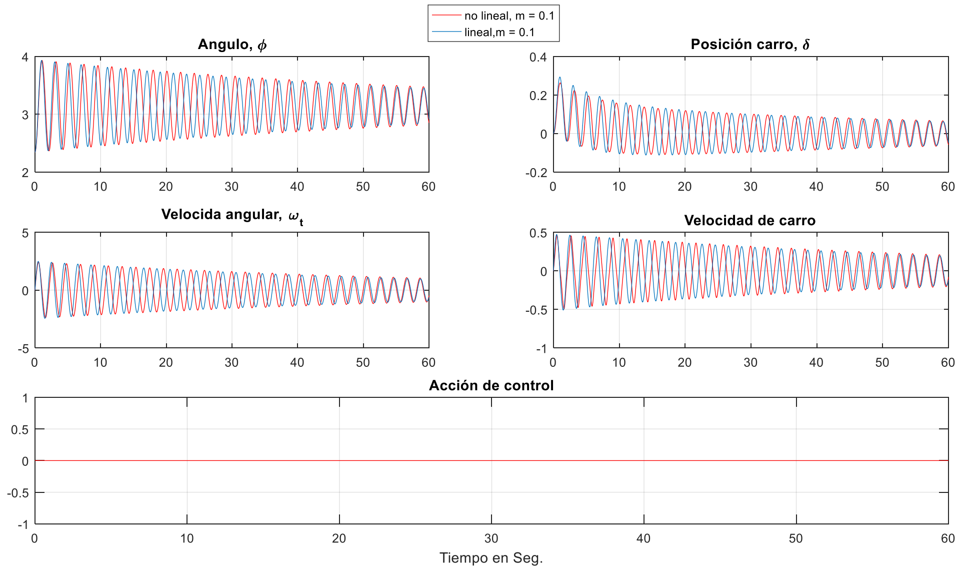
```
Matriz B:
/   0   \
|       |
|   1   |
|   -   |
|   M   |
|       |
|   0   |
|       |
|   1   |
|  ---  |
\  L M  /
```

6. Obtener la solución numérica de los dos sistemas, del lineal y del no lineal para evaluar cuantitativamente la equivalencia en el equilibrio estable modificando *m* de 0,1 a 0,01 y cambiar la longitud l a 1,2m.

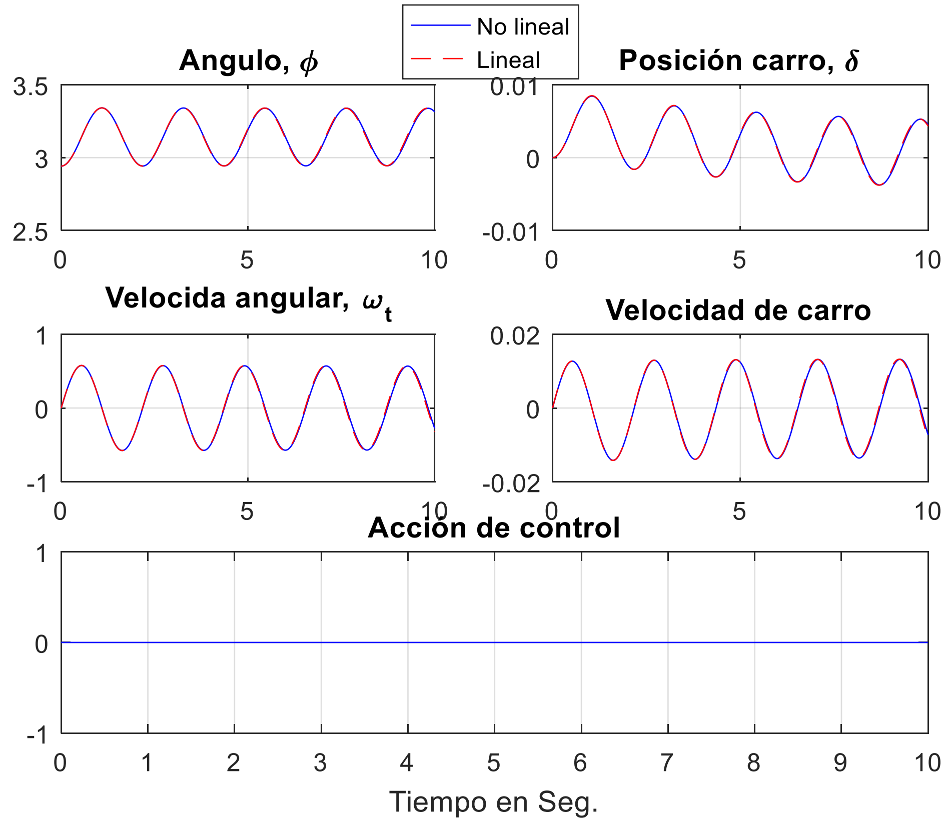
$$m = 0.1\,,con\ un\ angulo\ incial\ pi - 0.8$$



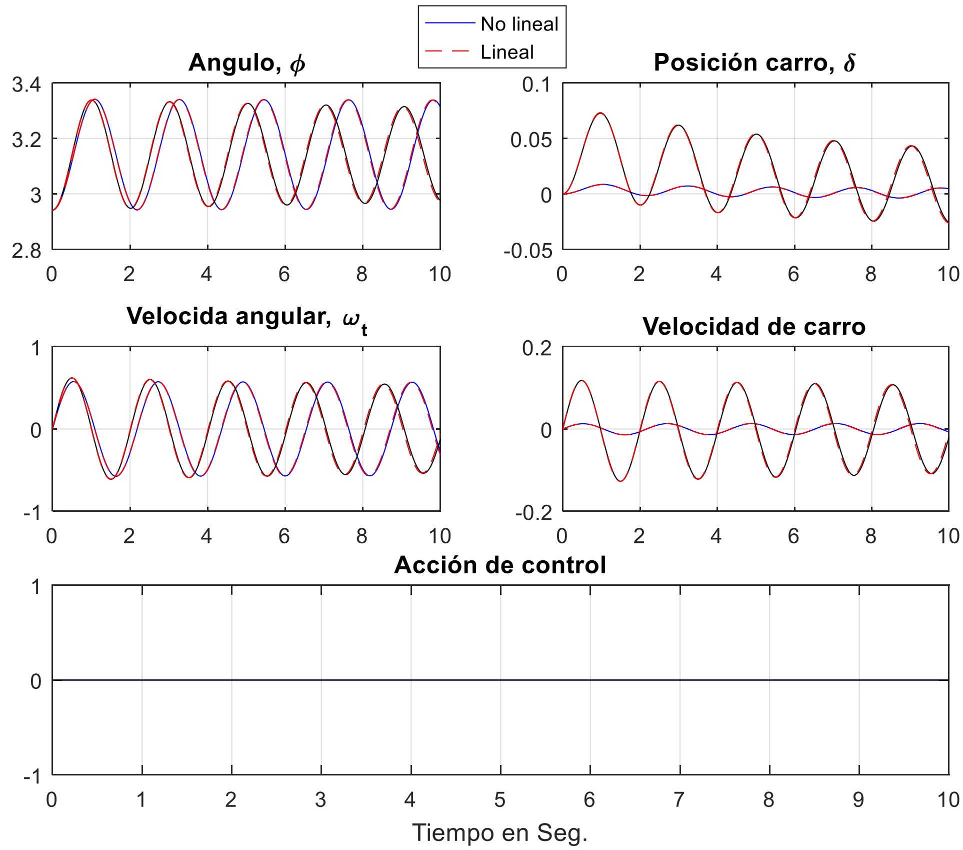
Como se ve el modelo lineal, responde bastante bien hasta cierto tiempo (antes de los 5 seg.), después ya presenta diferencias en su frecuencia de oscilación, esto es porque el ángulo en el que oscila es grande, pero a medida que el ángulo va disminuyendo el modelo lineal empieza a respondes de la misma manera que él no lineal, como se ve en la siguiente figura para un tiempo de simulación de 60 segundos.



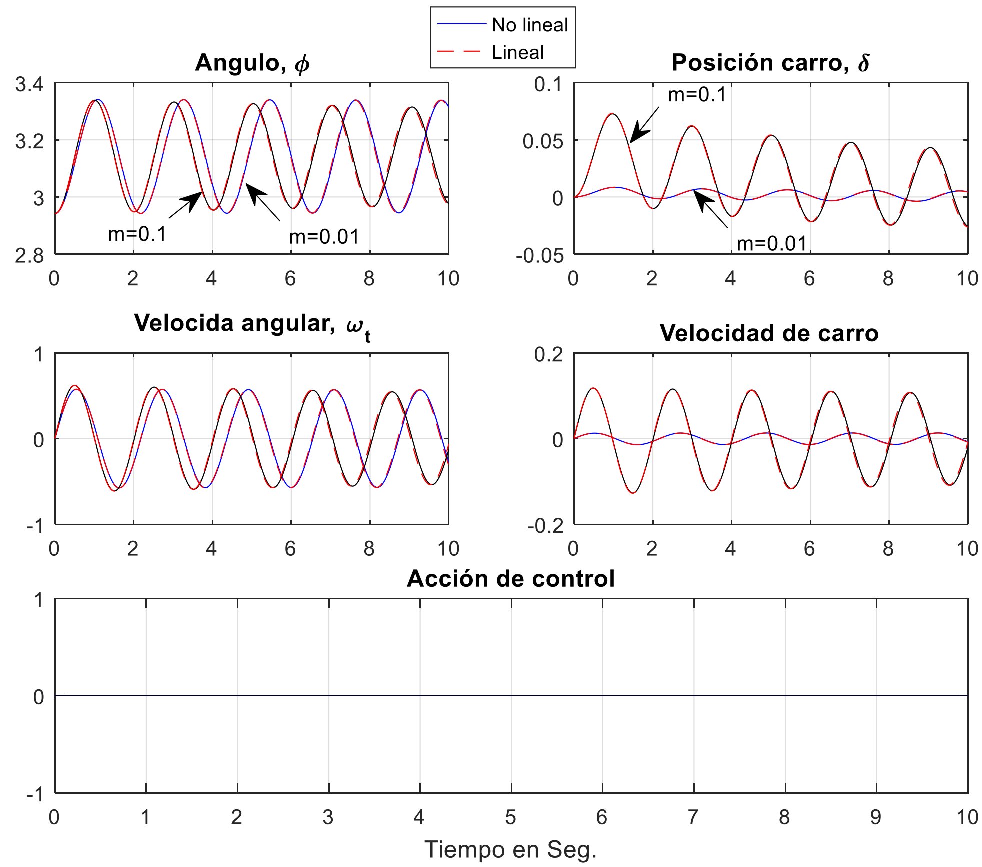
$$m = 0.01, con\ alguno\ de\ inicial\ pi - 0.2$$



Vemos como las oscilaciones son más chicas, y los modelos están superpuestos ya que el ángulo respecto del equilibrio es muy chico.



En la siguiente figura se observan las gráficas superpuestas para las distintas masas y los modelos superpuestas:



Lecciones aprendidas:

- Mayor entendimiento cualitativo de los resultados
- Practica en obtener el sistema lineal en variable de estados.
- Entendimiento de las limitaciones del modelo lineal.
- Distintas posibilidades de equilibrio de un sistema.

Complicaciones:

- Entendimiento del modelo no lineal, para ángulos negativos chicos.
- Verificación del sistema, en base a sus resultados
- Link útil para entendimiento del modelo:
  https://www.youtube.com/watch?v=9DF-R4wxpqA
  https://www.youtube.com/watch?v=PAtqTvc7lll&t=375s
  https://riunet.upv.es/bitstream/handle/10251/10296/Memoria.pdf
  https://www.youtube.com/watch?v=qjhAAQexzLg

Código utilizado:

```
clc;clear all;
syms M m dpp dp d fipp fip fi L F u g
%M=0.5;m=.1;F=.1;l=.6;g=9.8;
ang_inicial=0;
disp('Equilibrio iniestable')
ec1=(M+m)*dpp+m*L*fipp*cos(fi)-m*L*fip^2*sin(fi)+F*dp==u;
ec2 = L*fipp - g*sin(fi) + dpp*cos(fi) == 0;
%para el equilibrio inestable fi=0; ==> cos(0)=1, sin(0)=fi
%reemplaznado
ec1_aux= subs(subs(ec1,cos(fi),round(cos(ang_inicial))),sin(fi),fi);
ec2_aux= subs(subs(ec2,cos(fi),round(cos(ang_inicial))),sin(fi),fi);
%encontrar fipp y dpp
fipp_aux=solve(ec2_aux,fipp);
ec1_aux=subs(ec1_aux,'fipp',fipp_aux);
dpp_aux=solve(ec1_aux,dpp);
fipp_aux=subs(fipp_aux,'dpp',dpp_aux);
disp('dpp es igual a :');pretty(simplify(dpp_aux));
disp('fipp es igual a :');pretty(simplify(fipp_aux));
%linealizando para le equilibro inestable
%haciendo taylor para cada ecuacion y evaluando para el punto de operacion Xe=[0
0 0 0]
mat_A=[ [0 1 0 0];

[subs(subs(subs(subs(diff(dpp_aux,d),'d',0),dp,0),'fi',ang_inicial),'fip',0),...

subs(subs(subs(subs(diff(dpp_aux,dp),'d',0),'dp',0),'fi',ang_inicial),'fip',0),..
.

subs(subs(subs(subs(diff(dpp_aux,fi),'d',0),'dp',0),'fi',ang_inicial),'fip',0),..
.

subs(subs(subs(subs(diff(dpp_aux,fip),'d',0),'dp',0),'fi',ang_inicial),'fip',0)];
         [0 0 0 1];

[subs(subs(subs(subs(diff(fipp_aux,d),'d',0),dp,0),'fi',ang_inicial),'fip',0),...

subs(subs(subs(subs(diff(fipp_aux,dp),'d',0),'dp',0),'fi',ang_inicial),'fip',0),.
..

subs(subs(subs(subs(diff(fipp_aux,fi),'d',0),'dp',0),'fi',ang_inicial),'fip',0),.
..

subs(subs(subs(subs(diff(fipp_aux,fip),'d',0),'dp',0),'fi',ang_inicial),'fip',0)]
];
 mat_B[ 0;
```

```
subs(subs(subs(subs(diff(dpp_aux,u),'d',0),dp,0),'fi',ang_inicial),'fip',0);
        0;
subs(subs(subs(subs(diff(fipp_aux,u),'d',0),dp,0),'fi',ang_inicial),'fip',0)];
 disp('Matriz A:')
 pretty(simplify(mat_A))
 disp('Matriz B:')
 pretty(simplify(mat_B))
 %-------------
disp('equilibrio estable')
ang_inicial=pi;
%fi=pi ==> cos(fi)=-1 y sin(fi)=-fi
ec1_aux=subs(subs(ec1,cos(fi),-1),sin(fi),-fi);
ec2_aux=subs(subs(ec2,cos(fi),-1),sin(fi),-fi);
%encontramos dpp y fpp
fipp_aux=solve(ec2_aux,fipp);
ec1_aux=subs(ec1_aux,'fipp',fipp_aux);
dpp_aux=solve(ec1_aux,dpp);
fipp_aux=subs(fipp_aux,'dpp',dpp_aux);
disp('dpp es igual a :');pretty(simplify(dpp_aux))
disp('fipp es igual a :');pretty(simplify(fipp_aux))
%linealizando para le equilibro estable
%haciendo taylor para cada ecuacion y evaluando para el punto de operacion Xe=[0
0 pi 0]
mat_A=[ [0 1 0 0];

[subs(subs(subs(subs(diff(dpp_aux,d),'d',0),dp,0),'fi',ang_inicial),'fip',0),...

subs(subs(subs(subs(diff(dpp_aux,dp),'d',0),'dp',0),'fi',ang_inicial),'fip',0),..
.

subs(subs(subs(subs(diff(dpp_aux,fi),'d',0),'dp',0),'fi',ang_inicial),'fip',0),..
.

subs(subs(subs(subs(diff(dpp_aux,fip),'d',0),'dp',0),'fi',ang_inicial),'fip',0)];
         [0 0 0 1];

[subs(subs(subs(subs(diff(fipp_aux,d),'d',0),dp,0),'fi',ang_inicial),'fip',0),...

subs(subs(subs(subs(diff(fipp_aux,dp),'d',0),'dp',0),'fi',ang_inicial),'fip',0),.
..

subs(subs(subs(subs(diff(fipp_aux,fi),'d',0),'dp',0),'fi',ang_inicial),'fip',0),.
..

subs(subs(subs(subs(diff(fipp_aux,fip),'d',0),'dp',0),'fi',ang_inicial),'fip',0)]
];
```

```
 mat_B=[ 0;
subs(subs(subs(subs(diff(dpp_aux,u),'d',0),dp,0),'fi',ang_inicial),'fip',0);
        0;
 subs(subs(subs(subs(diff(fipp_aux,u),'d',0),dp,0),'fi',ang_inicial),'fip',0)];
 disp('Matriz A:')
 pretty(simplify(mat_A))
 disp('Matriz B:')
 pretty(simplify(mat_B))
```

```
clc;clear all;
%variables
T=10; At=1e-4; Kmax=T/At; t=linspace(0,T,Kmax);
m=0.1;F=0.1;long=1.2;g=9.8;M=0.5;color='k';angulo_inicial=pi;
d=zeros(1,int32(Kmax));dp=zeros(1,int32(Kmax));fi=zeros(1,int32(Kmax));fip=zeros(
1,int32(Kmax));
u=linspace(0,0,int32(Kmax));

%equilibrio inestable
A=[ 0 1 0 0;
    0 -F/M -(g*m)/M 0;
    0 0 0 1;
    0 F/(M*long) g*(M+m)/(M*long) 0]
B=[0;
   1/M;
   -1/(M*long);
   0]

%equilibrio estable
A1=[ 0 1 0 0;
     0 -F/M -(g*m)/M 0;
     0 0 0 1;
     0 -F/(M*long) -g*(M+m)/(M*long) 0]
B1=[ 0;
     1/M;
     1/(M*long);
     0]

%condiciones iniciales
d(1)=0;dp(1)=0;
dp(1)=0;fi(1)=angulo_inicial-0.2;fip(1)=0;dpp=0;fipp=0;
u(1)=0;
Xop=[0 0 angulo_inicial 0]';x=[d(1) dp(1) fi(1) fip(1)]';
dl(1)=0;dpl(1)=0;fil(1)=0;fipl(1)=0;
for i=1:Kmax-1
dpp      = (long*m*sin(fi(i))*fip(i)^2+ u(i) - F * dp(i) - fipp * long * m *
cos(fi(i)))/(M+m);
```

```
      fipp      = (-dpp * cos(fi(i))+g*sin(fi(i)))*(1/long);
      dp(i+1)   = dp(i)    + dpp *At;
      d(i+1)    = d(i)     + dp(i)*At;
      fip(i+1)  = fip(i)   + fipp*At;
      fi(i+1)   = fi(i)    + fip(i)*At;

      %variables de sistema lineal
       xp         = A1*(x-Xop)+B1*u(i);
       x          = x+xp*At;
       dl(i)      = x(1);
       dpl(i)     = x(2);
       fil(i)     = x(3);
       fipl(i)    = x(4);
      %Y=C*x;

end
dl(i+1)= x(1);dpl(i+1)= x(2);fil(i+1)= x(3);fipl(i+1)= x(4);
figure(1)
subplot(3,2,1);
plot(t,fi,color);hold on;%plot(t,pi*ones(size(t)),'k');hold on;
plot(t,fil,'r--');hold on;legend('No lineal','Lineal');
grid on;title('Angulo, \phi');
subplot(3,2,3);
plot(t,fip,color);hold on;%
plot(t,fipl,'r--');hold on;
grid on;title('Velocida angular, \omega_t');
subplot(3,2,2);
plot(t,d,color);hold on;%
plot(t,dl,'r--');hold on;
grid on;title('Posición carro, \delta');
subplot(3,2,4);
plot(t,dp,color);hold on;%
plot(t,dpl,'r--');hold on;
grid on;title('Velocidad de carro');
subplot(3,1,3);
plot(t,u,color);
grid on;title('Acción de control');xlabel('Tiempo en Seg.');hold on;
```