ESCUELA DE POSGRADO

Curso:

CONTROL ÓPTIMO

Tema:

Control de posicionamiento hidráulico

Presentado por:

CONTRERAS MARTINEZ, DIMEL ARTURO

Docente:

DR. ANTONIO MORÁN

2016

1. Modificar y resolver para t = finito

Sea el sistema:

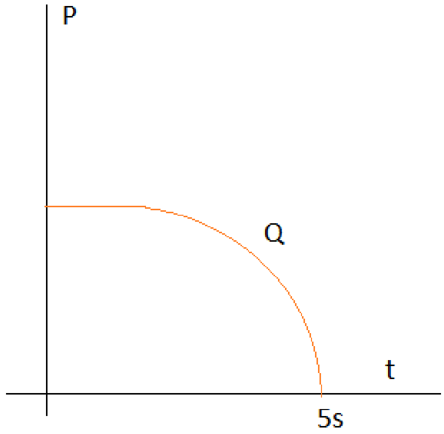
$$A = \begin{bmatrix} 2 & 1 & 0 \\ 3 & -10 & 0 \\ 2 & 1 & 2 \end{bmatrix}$$
$$B = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

Sea la función de costo.

$$J = X_{(tf)}{}^{T} P_{(tf)} X_{(tf)} + \int_{0}^{tf} (X^T Q X + u^T R u) dt$$

Se pide:
Hallar P

Comparar J mínimo, para tiempo infinito y finito.



Solución:

$$J_{min} = X_o{}^T P X_0$$

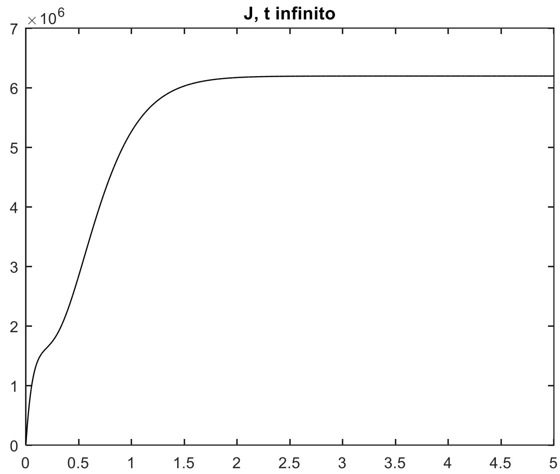
a. **Para tiempo infinito:**

Se utiliza para el cálculo de “P” la ecuación de Riccati.

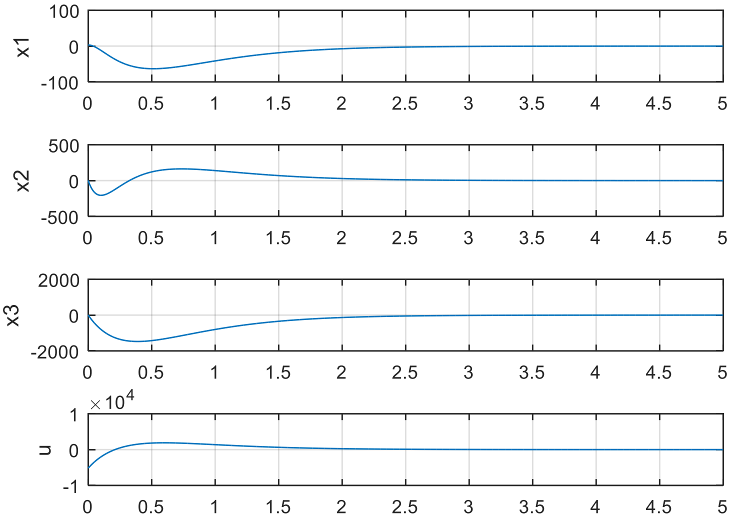
$$P = are(A, BR^{-1}B^T, Q)$$

Script en Matlab:

```
A = [ 2    1     0
        3   -10   0
        2    1     2 ];
B = [ 0
        1
        2 ];
q1 = 1e2;  q2 = 1e1;     q3 = 1e0;
Q = diag([ q1 q2 q3 ]);
R = [ 1 ];
P = are(A,B*inv(R)*B',Q);
K = inv(R)*B'*P;

ti = 0;    dt = 0.001;     tf = 5;
t = ti:dt:tf;      t = t';
[Ak Bk] = c2d(A,B,dt);

x0 = [ 3; 1; 5];
x = x0;
J = 0;
k =1;
for tt = ti:dt:tf

   x1(k,1) = x(1,1);
   x2(k,1) = x(2,1);
   x3(k,1) = x(3,1);
   u = -K*x;
   uu(k,1) = u;
   J = J + (x'*Q*x + u'*R*u)*dt;
   J1(k,1) = J;
   x = Ak*x + Bk*u;
   k = k + 1;
end
Jmin = x0'*P*x0;
disp('  ');
disp('Jmin Sumatoria - Jimin Estado Inicial');
[ J  Jmin ]

figure(1);
subplot(4,1,1);    plot(t,x1);    grid;   ylabel('x1');
subplot(4,1,2);    plot(t,x2);    grid;   ylabel('x2');
subplot(4,1,3);    plot(t,x3);    grid;   ylabel('x3');
subplot(4,1,4);    plot(t,uu);    grid;   ylabel('u');

figure(2)
plot(t,J1,'k');title('J, t infinito');
```



Se obtiene Jmin = 6.1951e+06



b. <u>Para tiempo finito:</u>

Se calcula el P mediante la ecuación matricial diferencial de Riccati.

$$\dot{\mathrm{P}} = -A^T\mathrm{P} - \mathrm{PA} + \mathrm{PB}R^{-1}B^T\mathrm{P} - \mathrm{Q}$$

$$\mathrm{P_N} = \mathrm{Q}$$

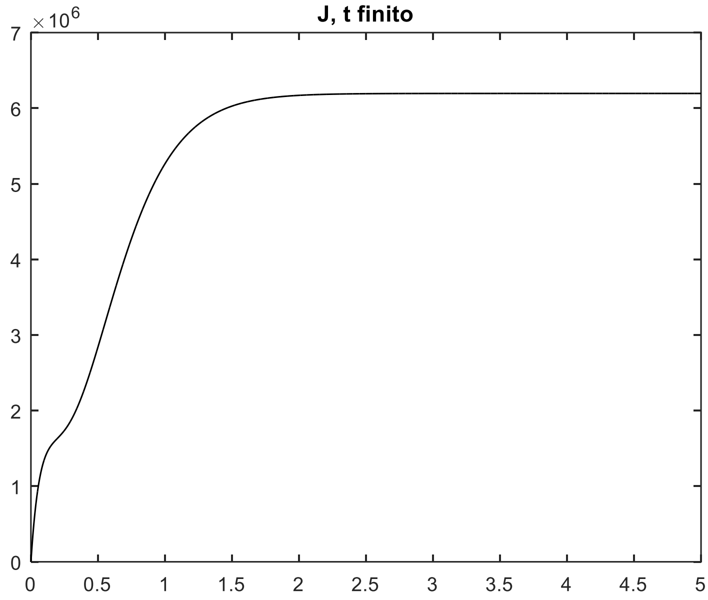
Y además mediante Euler:

$$\dot{P} = \frac{P_k - P_{k-1}}{\Delta t}$$

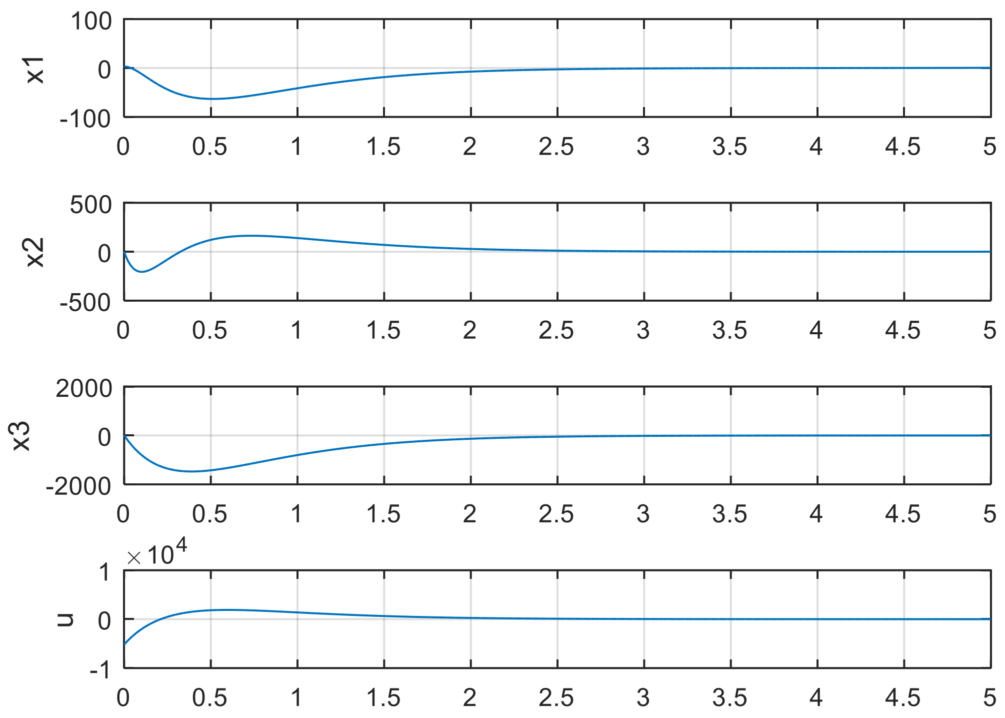
$$\Delta t \rightarrow 0$$

Script en Matlab

```
q1 = 1e2;   q2 = 1e1;     q3 = 1e0;
Q  = diag([ q1 q2 q3 ]);
R  = [ 1 ];

Pn = Q;

ti = 0;    dt = 0.001;     tf = 5;
t = ti:dt:tf;      t = t';
nt = length(t);
[Ak Bk] = c2d(A,B,dt);

x0 = [ 3; 1; 5];
x = x0;
J = 0;
k =1;

P(:,:,nt) = Pn;
k = nt;
for tt = tf:-dt:(ti+dt)
    Pk = P(:,:,k);
    Pp = -A'*Pk - Pk*A + Pk*B*inv(R)*B'*Pk - Q;
    P(:,:,k-1) = Pk - dt*Pp;
    %P11(k,1) = P(1,1,k);
    k = k - 1;
end

x0 = [ 3; 1; 5];
x  = x0;
J  = 0;
k  = 1;
for tt = ti:dt:tf
   x1(k,1) = x(1,1);
   x2(k,1) = x(2,1);
   x3(k,1) = x(3,1);
   Pk = P(:,:,k);
   u(k,1) = -inv(R)*B'*Pk*x; %(x-xast);
   %uu(k,1) = u;
   Jo(k,1)=J;
   J = J + (x'*Q*x + u(k,1)'*R*u(k,1))*dt;
   x = Ak*x + Bk*u(k,1);
   k = k + 1;
end
Jmin = x0'*P(:,:,1)*x0;
disp('  ');
disp('Jmin Sumatoria - Jimin Estado Inicial');
[ J  Jmin ]

figure(1);
subplot(4,1,1);    plot(t,x1);    grid;   ylabel('x1');
subplot(4,1,2);    plot(t,x2);    grid;   ylabel('x2');
subplot(4,1,3);    plot(t,x3);    grid;   ylabel('x3');
subplot(4,1,4);    plot(t,u);    grid;   ylabel('u');

figure(2)
plot(t,Jo,'k');title('J, t infinito');
```



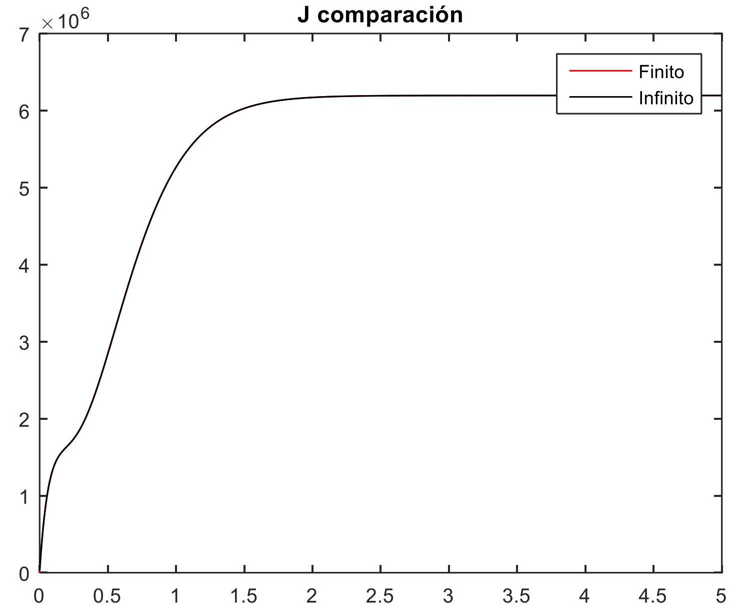
Se obtiene Jmin = 6.1951e+06



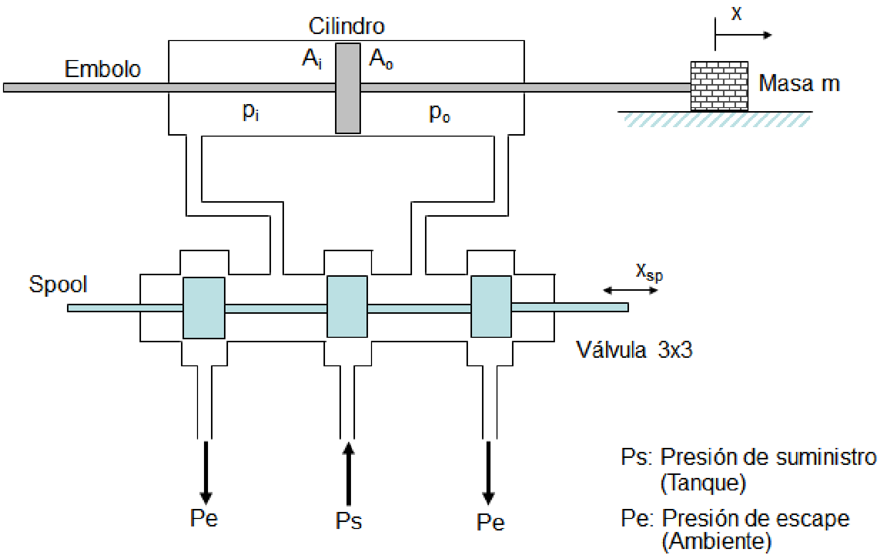
Comparando:

Se obtiene el valor de Jmin = 6.1951e+06 iguales en ambos casos.

La gráfica de comparación del valor de J:



2. Control de posicionamiento hidráulico



Se pide reducir el sistema a orden 3 y calcular el controlador para el sistema linealizado.

Solución:

Linealización

Se considera el vector de estados X:

$$X = \begin{bmatrix} x \\ \dot{x} \\ Pi - Po \end{bmatrix}$$

Y la derivada de X:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dfrac{d(Pi - Po)}{dt} \end{bmatrix}$$

De la ecuación de movimiento del bloque de masa “m”, obtenemos $\ddot{x}$.

$$m\ddot{x} = A(Pi - Po) - c\dot{x} - Fs$$

$$\ddot{x} = \frac{A}{m}(Pi - Po) - \frac{c}{m}\dot{x} - \frac{Fs}{m}$$

Para obtener las ecuaciones de linealización:

Se considera punto de operación :

$$Vi = Vo = V$$

$$Xsp = \frac{Xspmax}{2}$$

A partir de los flujos de entrada y salida se obtiene:

$$a_i x_{sp} + b_i p_i = A_i \dot{x} + \frac{V}{\beta}\frac{dPi}{dt} \ldots .(I)$$

$$a_o x_{sp} + b_o p_o = A_o \dot{x} - \frac{V}{\beta}\frac{dPo}{dt} \ldots (II)$$

De (I) y (II) se obtiene:

$$(a_i + a_o)x_{sp} + b_i P_i + boPo = (A_i + Ao)\dot{x} + \frac{V}{\beta}\frac{d(Pi - Po)}{dt}$$

Se reemplaza $b_i P_i = -b_o P_i$

$$(a_i + a_o)x_{sp} - bo(Pi - Po) = (A_i + Ao)\dot{x} + \frac{V}{\beta}\frac{d(Pi - Po)}{dt}$$

$$\frac{d(Pi - Po)}{dt} = \frac{(a_i + a_o)}{V}\beta x_{sp} - \frac{bo}{V}\beta(Pi - Po) - \frac{(A_i + Ao)}{V}\beta\dot{x}$$

Se considera :

$$A_i = Ao = Area = A$$

El sistema de 3° Orden resulta:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dfrac{d(Pi - Po)}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -c/m & A/m \\ 0 & -\dfrac{2A\beta}{V} & -\dfrac{bo}{V}\beta \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ Pi - Po \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{(a_i + a_o)\beta}{V} \end{bmatrix} Xsp + \begin{bmatrix} 0 \\ -1/m \\ 0 \end{bmatrix} Fs$$

<u>Escalamiento z para balanceo de las matrices:</u>

$$X = \begin{bmatrix} x \\ \dot{x} \\ Pi - Po \end{bmatrix}$$

Debido a que los valores de presión son altos:

$$PPi = zPi$$

$$PPo = zPo$$

$$z\frac{d(Pi - Po)}{dt} = z\frac{(a_i + a_o)}{V}\beta x_{sp} - z\frac{bo}{V}\beta(Pi - Po) - z\frac{(A_i + Ao)}{V}\beta\dot{x}$$

$$\frac{d(PPi - PPo)}{dt} = \frac{(a_i + a_o)}{V}\beta\, z\, x_{sp} - \frac{bo}{V}\beta z(Pi - Po) - \frac{2Az}{V}\beta\dot{x}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dfrac{d(PPi - PPo)}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -c/m & A/zm \\ 0 & -\dfrac{2A\beta z}{V} & -\dfrac{bo}{V}\beta z \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ PPi - PPo \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{(a_i + a_o)\beta z}{V} \end{bmatrix} Xsp + \begin{bmatrix} 0 \\ -1/m \\ 0 \end{bmatrix} Fs$$

Cálculo del controlador

$$u = -K \begin{bmatrix} x - r \\ \dot{x} \\ PPi - PPo \end{bmatrix}$$

Simulación

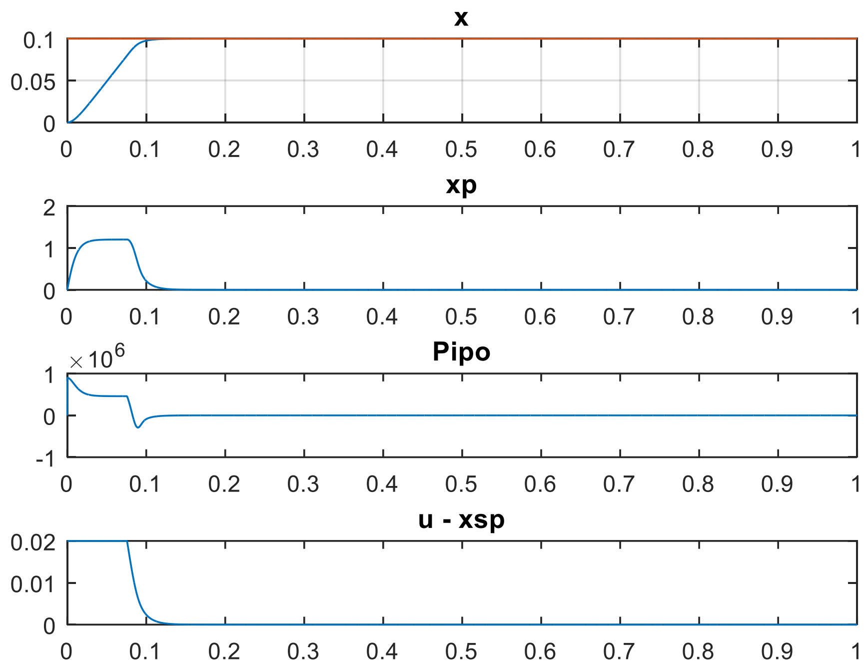
Script en Matlab:

```
Area = 1.18E-3;      % D = 0.04     d = 0.01
Ai = Area;
Ao = Area;
maxelon = 0.20;      % Elongacion maxima
Vol = Area*maxelon;
beta = 1.25E9;
rho = 900;
cd = 16E-2;
w = 0.02;
c = 450;
m = 10;
Fseca =  0*400;    % Variar el coeficente de 0 a 2.75
Pe = 1E5;            % Presion de escape
Ps = 10E5;           % Presion del tanque

xspmax = 0.02;
xmax = maxelon*0.80;     % 80% de elongacion máxima

%Punto de operación:
Pio = Ps/2;        % Probar valores
Poo = 2*Pe;
Pio = (Ps+Pe)/2;
Poo = (Ps+Pe)/2;
xspo = xspmax/2;

ai = cd*w*sqrt(2/rho*(Ps-Pio));
bi = -cd*w*xspo/sqrt(2*rho*(Ps-Pio));
ao cd*w*sqrt(2/rho*(Poo-Pe));
bo = cd*w*xspo/sqrt(2*rho*(Poo-Pe));
aa = (ai+ao)/2; %aproximación

a22 = -c/m;
a23 =  Area/m;
a32 = -2*Area*beta/Vol;
a33 = -bo*beta/Vol;

b3  =  2*aa*beta/Vol;

w2 = -1/m;
A = [  0   1       0
       0   a22    a23
       0   a32    a33 ];

B = [  0
       0
       b3];
```

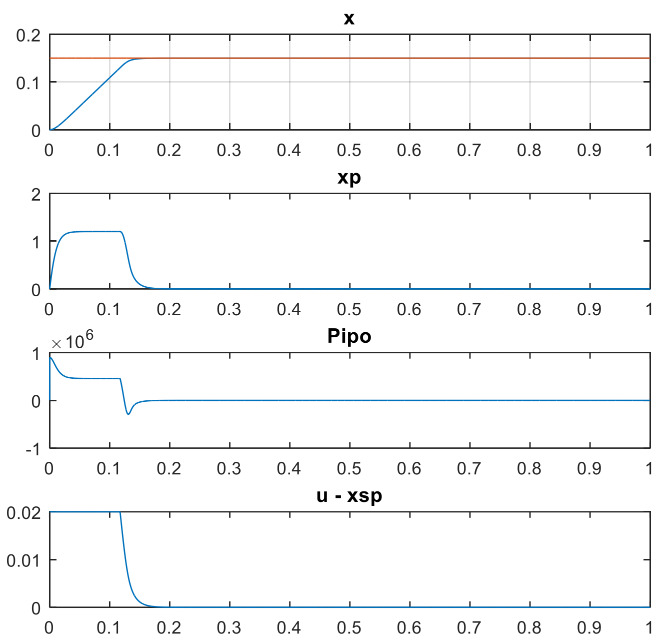
```
z = 1e-8;   % Analizar efecto
A = [  0    1       0
       0   a22      a23/z
       0   a32*z    a33*z ];
B = [  0
       0
       b3*z];
qx  = input('Introducir qx  [1e-1,1,10,1E2,1E3] : ');
qv  = input('Introducir qv  [0] : ');
qpipo = input('Introducir qpipo [0] : ');
Q = diag([qx  qv  qpipo]);
R = [ 1 ];
Pric = are(A,B*inv(R)*B',Q);
K = inv(R)*B'*Pric;
ti = 0;
dt = 0.00001;
tf = 1;
t = ti:dt:tf;
t = t';
nt = length(t);
%Condiciones iniciales
x  = 0.0;
xp = 0;
Pi = 1*Pe;
Po = 1*Pe;
Pipo = 0*Pe;
ampxast = input('Introducir xast [-0.15 0.15 ] : ');
xast = ampxast*ones(nt,1);
k = 1;
for tt = ti:dt:tf
  pos(k,1)     = x;
  pos2(k,1)     = x;
  vel(k,1)     = xp;
  Preio(k,1)    = Pipo;
  error = x - xast(k,1);
  xsp = -K*[ error; xp;  z*(Pipo)];
  if(xsp > xspmax)
     xsp = xspmax;
  elseif(xsp < -xspmax)
     xsp = -xspmax;
  end
  if(abs(x) >= xmax)
     xsp = 0;
  end
  u(k,1) = xsp;
  Vi = Vol + Ai*x;
  Vo = Vol - Ao*x;
  Volo(k,1) = Vo;
  if(xp >= 0)
     Ff = Fseca;
  elseif( xp < 0 )
     Ff = -Fseca;
  elseif( xp == 0 )
    Ff = Area*Pipo;
  end
  x2p = Area*Pipo/m - c/m*xp -Ff/m;
  if(xsp > 0)
     qi = cd*w*xsp*sqrt(2*(Ps-Pi)/rho);
     qo = cd*w*xsp*sqrt(2*(Po-Pe)/rho);
  elseif(xsp < 0)
     qi = cd*w*xsp*sqrt(2*(Pi-Pe)/rho);
     qo = cd*w*xsp*sqrt(2*(Ps-Po)/rho);
  elseif(xsp == 0)
     qi = 0;
     qo = 0;
  end
  Pip = -Ai*beta/Vi*xp + beta/Vi*qi;
  Pop =  Ao*beta/Vo*xp - beta/Vo*qo;
  x  = x + xp*dt;
  xp = xp + x2p*dt;
  Pi = Pi + Pip*dt;
  Po = Po + Pop*dt;
  Pipo = Pi - Po ;
  if(Pi > Ps)
     Pi = Ps;
  elseif(Pi < Pe)
     Pi = Pe;
  end
  if(Po > Ps)
     Po = Ps;
  elseif(Po < Pe)
     Po = Pe;
  end
```

```
figure(1);
subplot(4,1,1);  plot(t,pos,t,xast);title('x'); grid on;
subplot(4,1,2);  plot(t,vel);;title('xp');
subplot(4,1,3);  plot(t,Preio);;title('Pipo');
subplot(4,1,4);  plot(t,u);title('u - xsp');
```

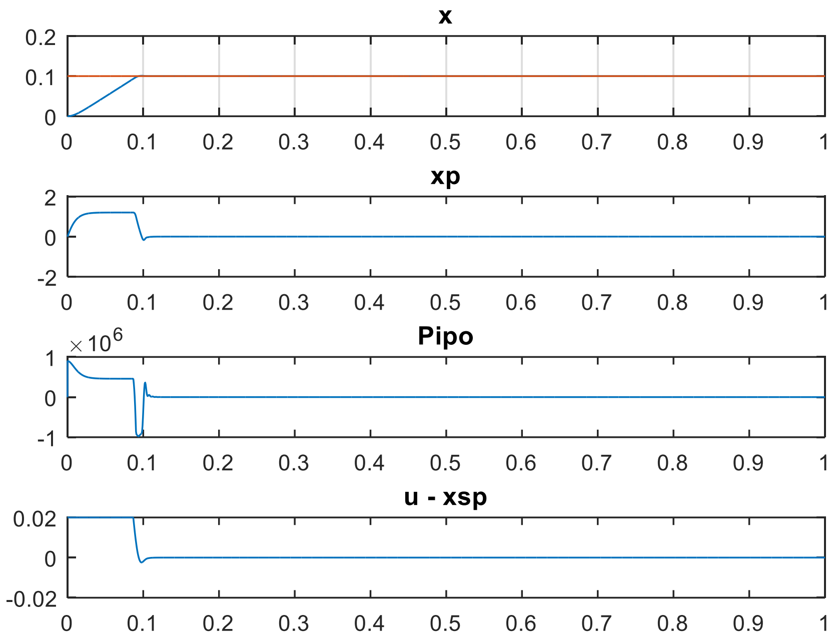
a. Prueba para qx = 1 ; qv = 0 ; qpipo = 0 ; x* = 0.1



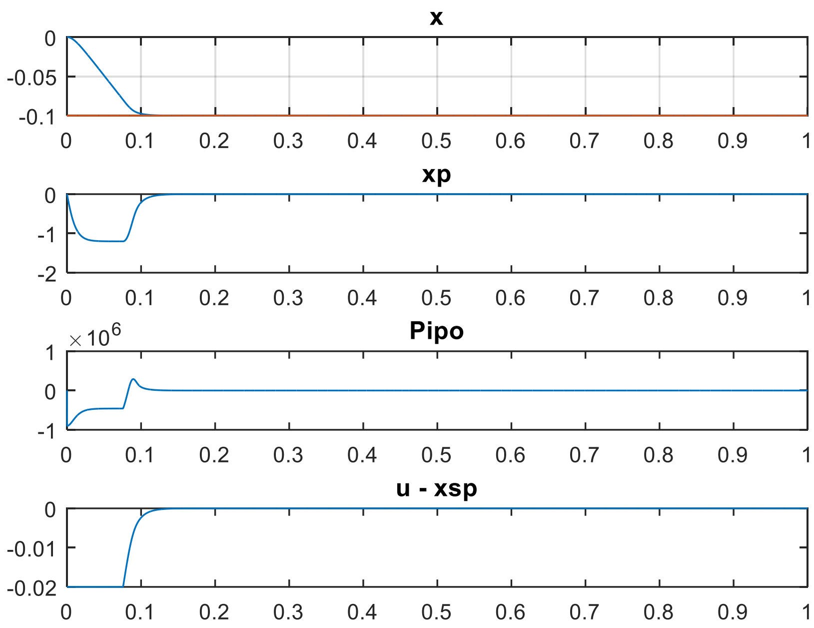
b. Prueba para qx = 1 ; qv = 0 ; qpipo = 0 ; x* = 0.15



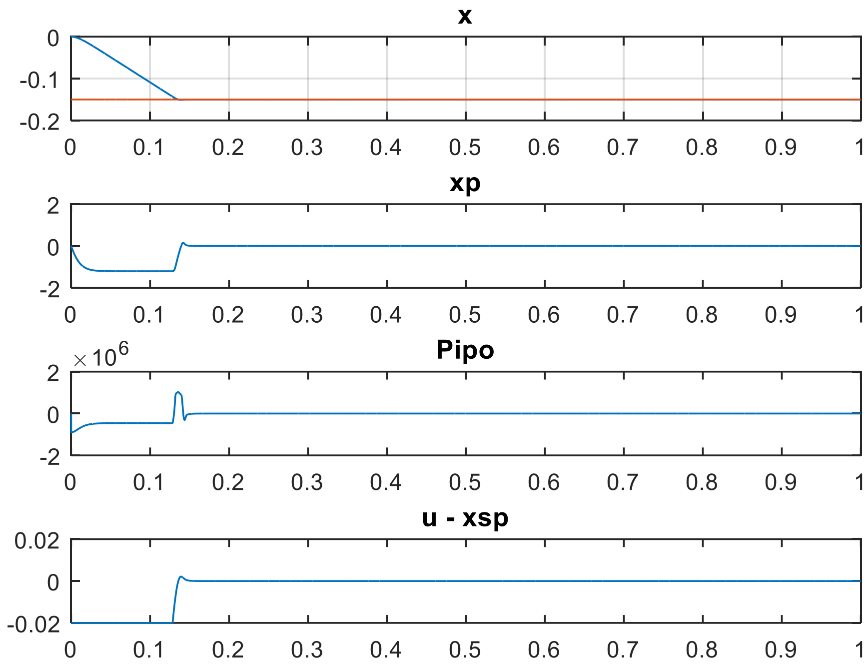
c. **Prueba para qx = 10 ; qv = 0 ; qpipo = 0 ; x* = 0.1**



d. **Prueba para qx = 1 ; qv = 0 ; qpipo = 0 ; x* = -0.1**



e. Prueba para qx = 10 ; qv = 0 ; qpipo = 0 ; x* = -0.15



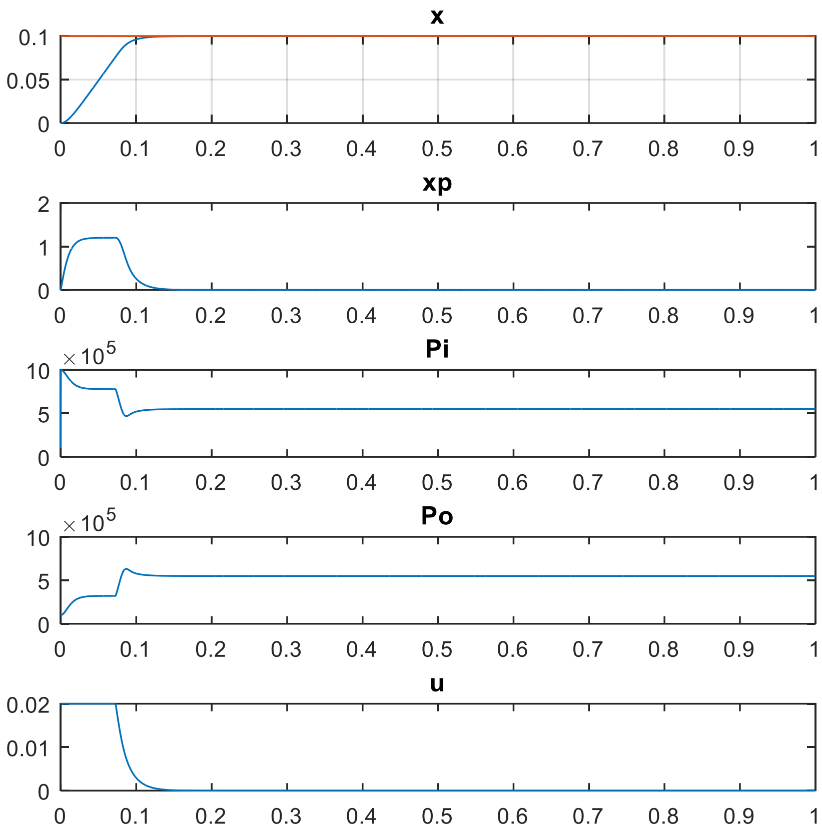
Comparación con el sistema de orden 4

El sistema de orden 4 desarrollado en clase se implementó en el siguiente SCRIPT MATLAB:

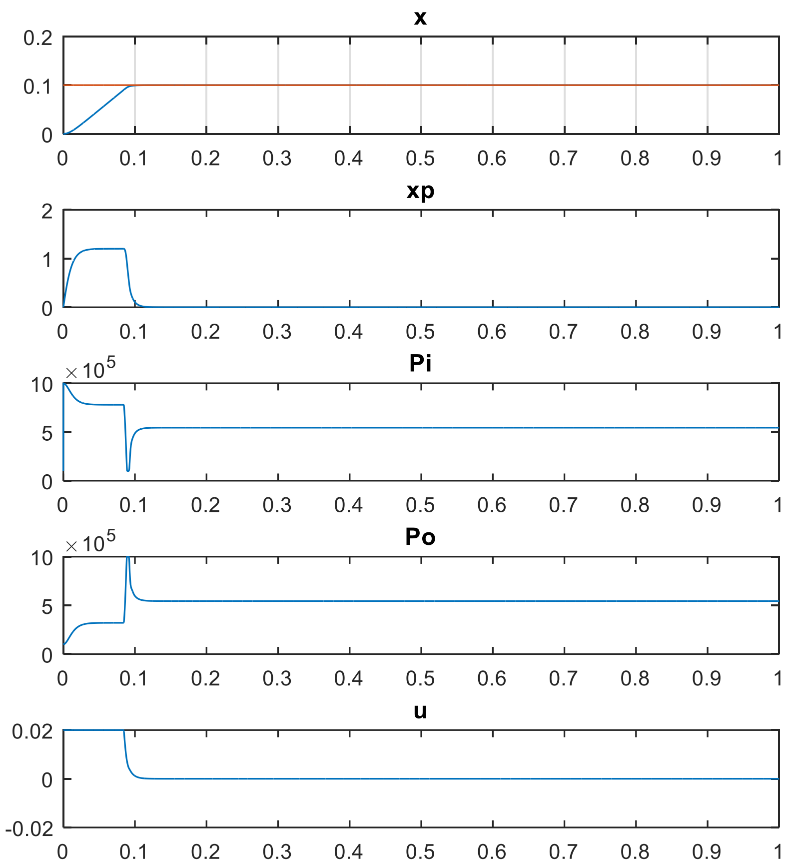
```
%Punto de operación:
Pio = Ps/2;       % Probar valores
Poo = 2*Pe;
Pio = (Ps+Pe)/2;
Poo = (Ps+Pe)/2;
xspo = xspmax/2;

ai = cd*w*sqrt(2/rho*(Ps-Pio));
bi = -cd*w*xspo/sqrt(2*rho*(Ps-Pio));
ao = cd*w*sqrt(2/rho*(Poo-Pe));
bo = cd*w*xspo/sqrt(2*rho*(Poo-Pe));

a22 = -c/m;
a23 =  Area/m;
a24 = -Area/m;
a32 = -Area*beta/Vol;
a33 =  bi*beta/Vol;
a42 =  Area*beta/Vol;
a44 = -bo*beta/Vol;
b3  =  ai*beta/Vol;
b4  = -ao*beta/Vol;
w2 = -1/m;
A = [  0   1      0     0
         0   a22    a23   a24
         0   a32    a33    0
         0   a42    0     a44 ];

B = [  0
         0
         b3
         b4 ];

z = 1E-8;    % Analizar efecto
A = [  0    1          0         0
         0   a22       a23/z    a24/z
         0   a32*z    a33        0
         0   a42*z      0       a44 ];

B = [  0
         0
       b3*z
       b4*z ];

qx  = input('Introducir qx  [1e-1,1,10,1E2,1E3] : ');
qv  = input('Introducir qv  [0] : ');
qpi = input('Introducir qpi [0] : ');
qpo = input('Introducir qpo [0] : ');

Q = diag([qx  qv  qpi  qpo]);
R = [ 1 ];

Pric = are(A,B*inv(R)*B',Q);
K = inv(R)*B'*Pric;

ti = 0;dt = 0.00001;
tf = 1;
t = ti:dt:tf;
t = t';
nt = length(t);

x  = 0.0;
xp = 0;
Pi = 1*Pe;
Po = 1*Pe;
ampxast = input('Introducir xast [-0.15 0.15 ] : ');
xast = ampxast*ones(nt,1);
```

```
k = 1;
for tt = ti:dt:tf
  pos(k,1)    = x;
  pos1(k,1)    = x;
  vel(k,1)    = xp;
  Prei(k,1)   = Pi;
  Preo(k,1)   = Po;
  error = x - xast(k,1);
  xsp = -K*[ error; xp;  z*(Pi-0*1*Pio); z*(Po - 0*1*Poo) ];
  if(xsp > xspmax)
     xsp = xspmax;
  elseif(xsp < -xspmax)
     xsp = -xspmax;
  end

  if(abs(x) >= xmax)
     xsp = 0;
  end
  u(k,1) = xsp;
  Vi = Vol + Ai*x;
  Vo = Vol - Ao*x;
  Volo(k,1) = Vo;
  if(xp >= 0)
     Ff = Fseca;
  elseif( xp < 0 )
     Ff = -Fseca;
  elseif( xp == 0 )
    Ff = Ai*Pi - Ao*Po;
  end
  x2p = Ai/m*Pi - Ao/m*Po - c/m*xp -Ff/m;
  if(xsp > 0)
     qi = cd*w*xsp*sqrt(2*(Ps-Pi)/rho);
     qo = cd*w*xsp*sqrt(2*(Po-Pe)/rho);
  elseif(xsp < 0)
     qi = cd*w*xsp*sqrt(2*(Pi-Pe)/rho);
     qo = cd*w*xsp*sqrt(2*(Ps-Po)/rho);
  elseif(xsp == 0)
     qi = 0;
     qo = 0;
  end

  Pip = -Ai*beta/Vi*xp + beta/Vi*qi;
  Pop =  Ao*beta/Vo*xp - beta/Vo*qo;
  x  = x + xp*dt;
  xp = xp + x2p*dt;
  Pi = Pi + Pip*dt;
  Po = Po + Pop*dt;
  if(Pi > Ps)
     Pi = Ps;
  elseif(Pi < Pe)
     Pi = Pe;
  end
  if(Po > Ps)
     Po = Ps;
  elseif(Po < Pe)
     Po = Pe;
  end
  k = k + 1;
end

figure(1);
subplot(5,1,1);  plot(t,pos,t,xast);title('x');grid on;
subplot(5,1,2);  plot(t,vel);;title('xp');
subplot(5,1,3);  plot(t,Prei);;title('Pi');
subplot(5,1,4);  plot(t,Preo);;title('Po');
subplot(5,1,5);  plot(t,u);title('u');
```

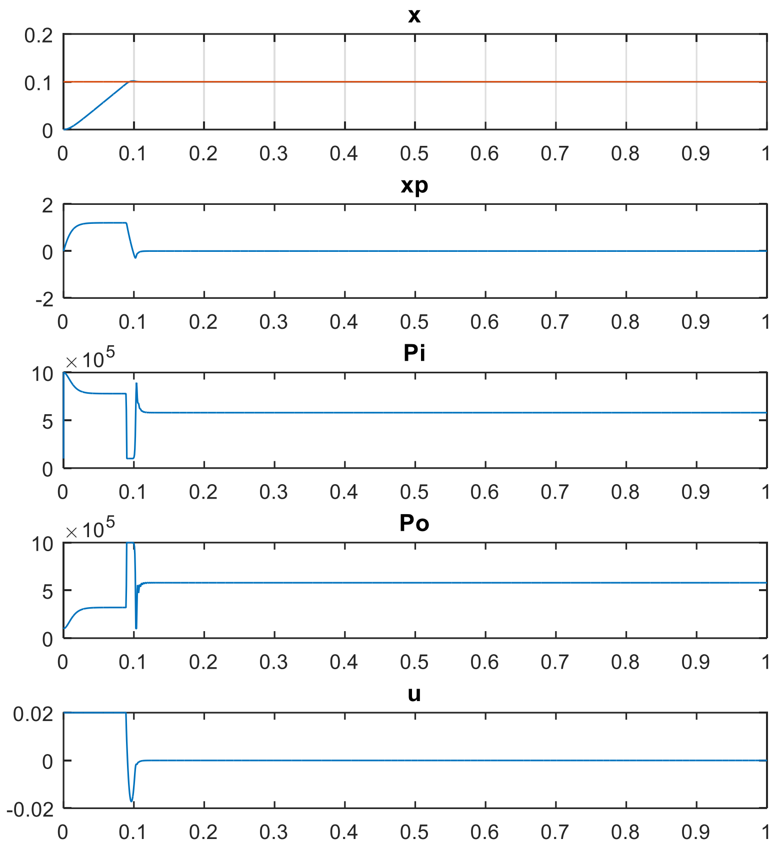
a. **Prueba para qx = 1 ; qv = 0 ; qpi= 0 ;qpo = 0 ; x* = 0.1**



b. **Prueba para qx = 10 ; qv = 0 ; qpi= 0 ;qpo = 0 ; x* = 0.1**



c. **Prueba para qx = 100 ; qv = 0 ; qpi= 0 ;qpo = 0 ; x* = 0.1**

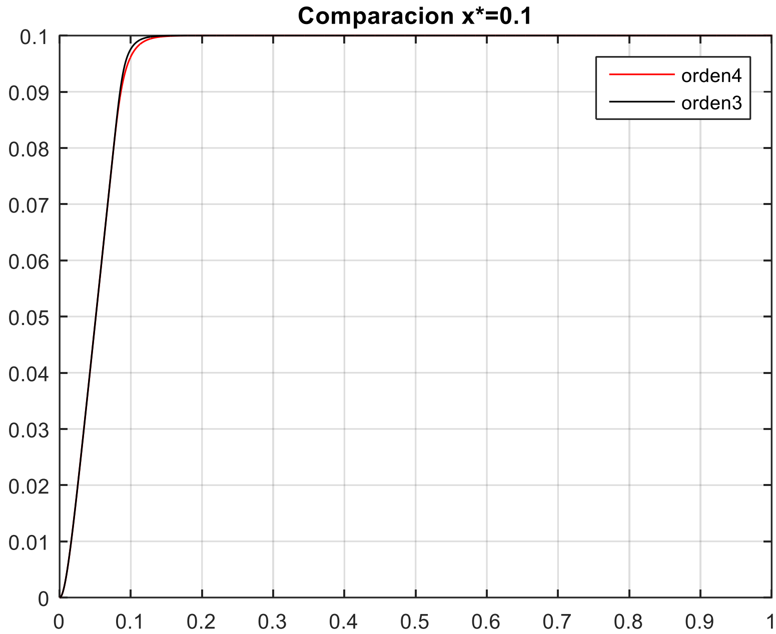


Comparaciones entre el sistema de orden 3 y orden 4

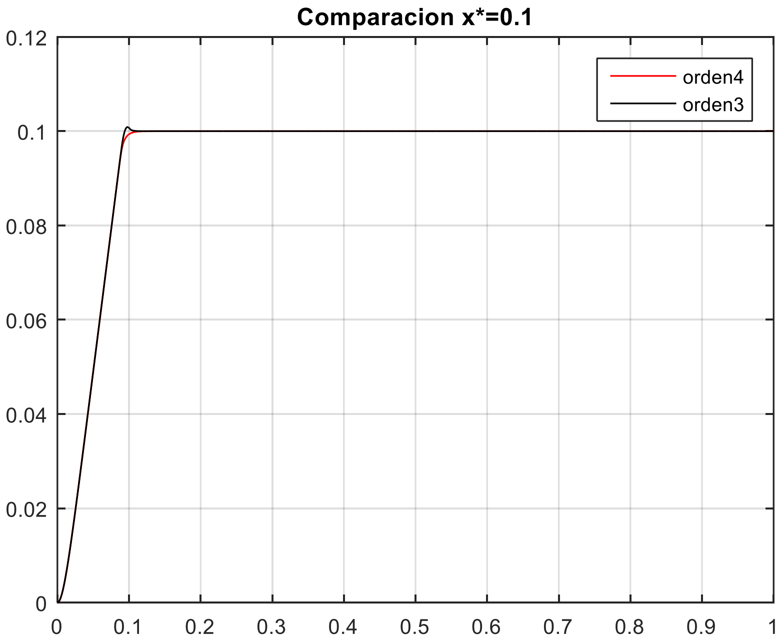
Con el punto de operación considerado:

```
Pio = Ps/2;
Poo = 2*Pe;
Pio = (Ps+Pe)/2;
Poo = (Ps+Pe)/2;
xspo = xspmax/2;
```

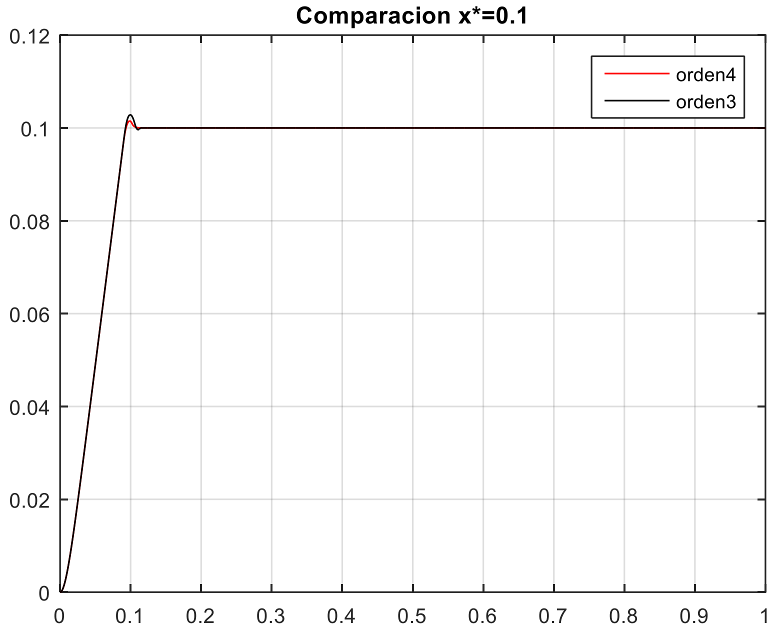
a. **Prueba para qx = 1 ; qv = 0 ; qpipo = 0 [orden3] ; x* = 0.1 y qpi= 0 ; qpo = 0 [orden4]**



b. **Prueba para qx = 10 ; qv = 0 ; qpipo = 0 [orden3] ; x* = 0.1 y qpi= 0 ; qpo = 0 [orden4]**



a. **Prueba para qx = 100 ; qv = 0 ; qpipo = 0 [orden3] ; x* = 0.1 y qpi= 0 ; qpo = 0 [orden4]**



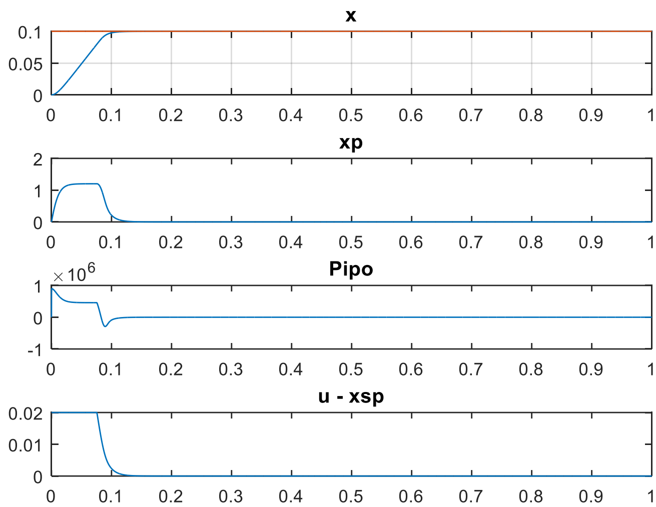
<u>Prueba variando los valores medios de operación</u>

Para el sistema de orden 3, se variará el punto de operación

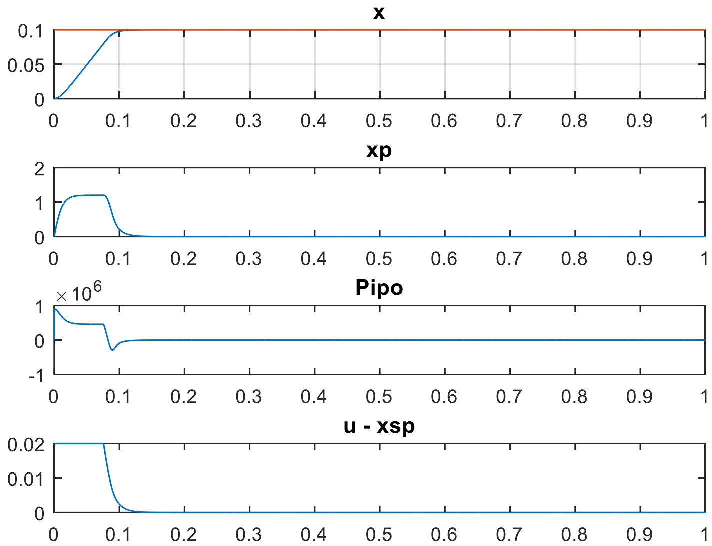
a. **Prueba para qx = 1 ; qv = 0 ; qpipo = 0; x* = 0.1**

<u>Variando el xspo:</u>

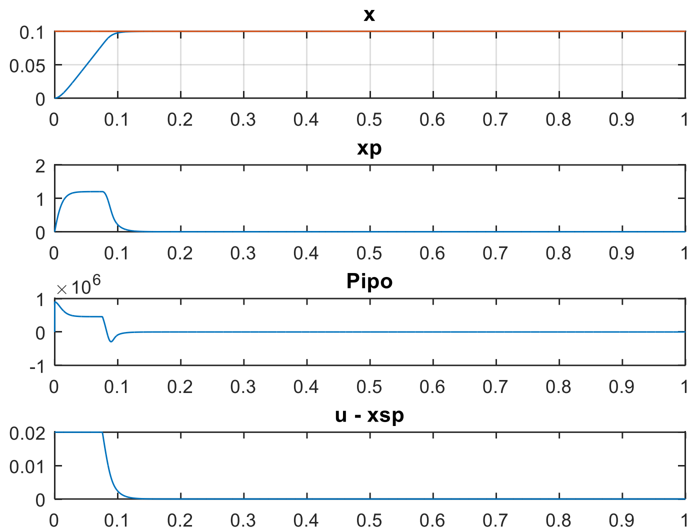
$$\boldsymbol{xspo} = \frac{\text{xspmax}}{\mathbf{3}}$$



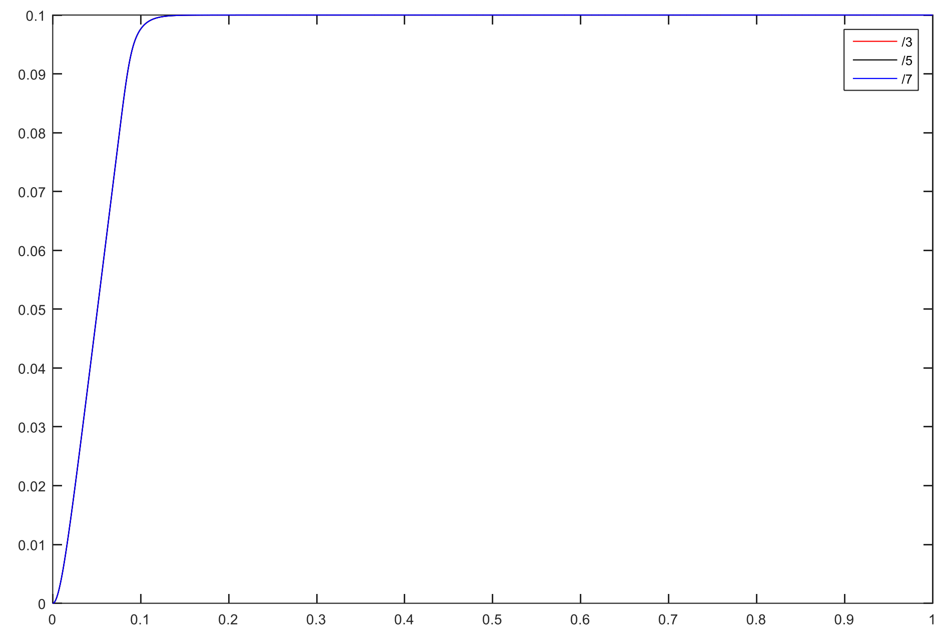
$$\boldsymbol{xspo} = \frac{\text{xspmax}}{\mathbf{5}}$$

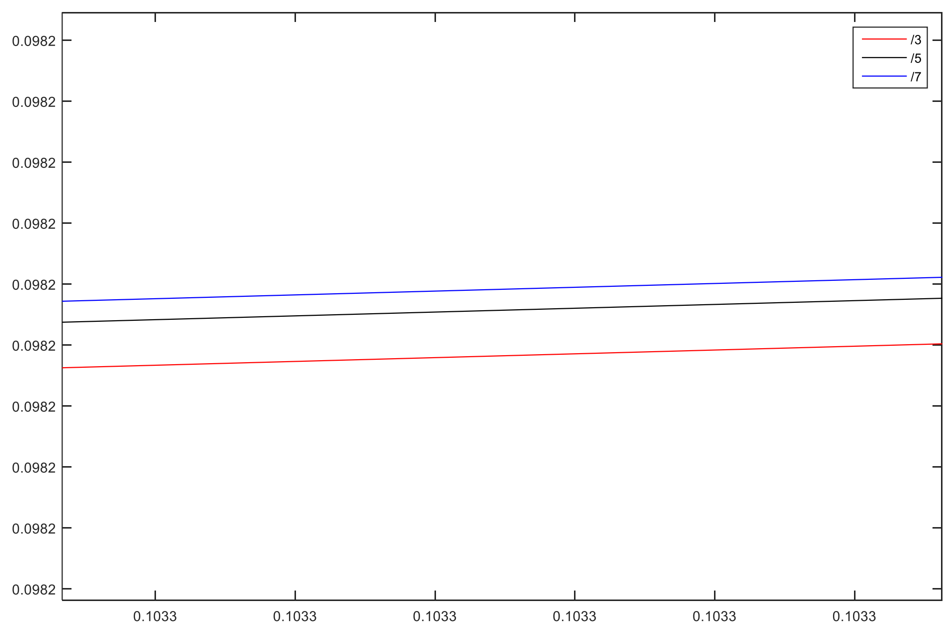


$$xspo = \frac{\text{xspmax}}{7}$$



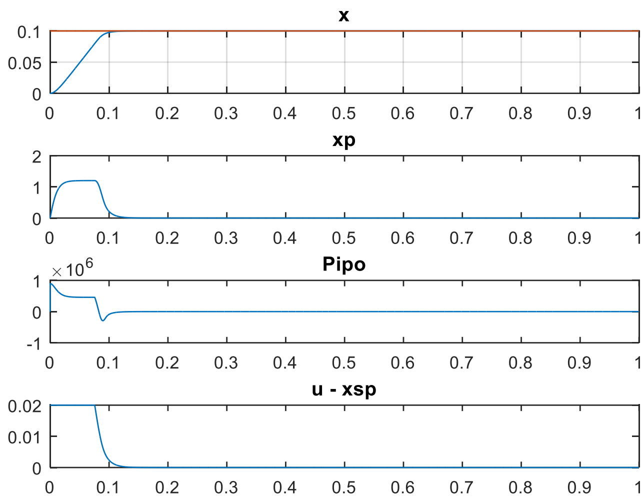
Resumen de los casos:



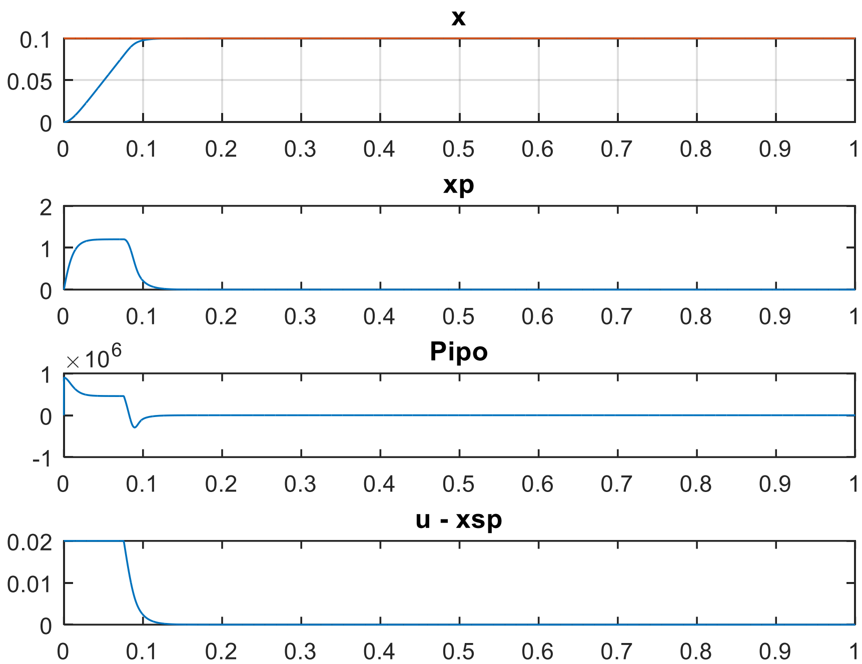


Variando Pio y Poo:

$$\boldsymbol{Pio} = \frac{\mathrm{Ps} + \mathrm{Pe}}{\mathbf{3}}$$
$$\boldsymbol{Poo} = \frac{\mathrm{Ps} + \mathrm{Pe}}{\mathbf{3}}$$

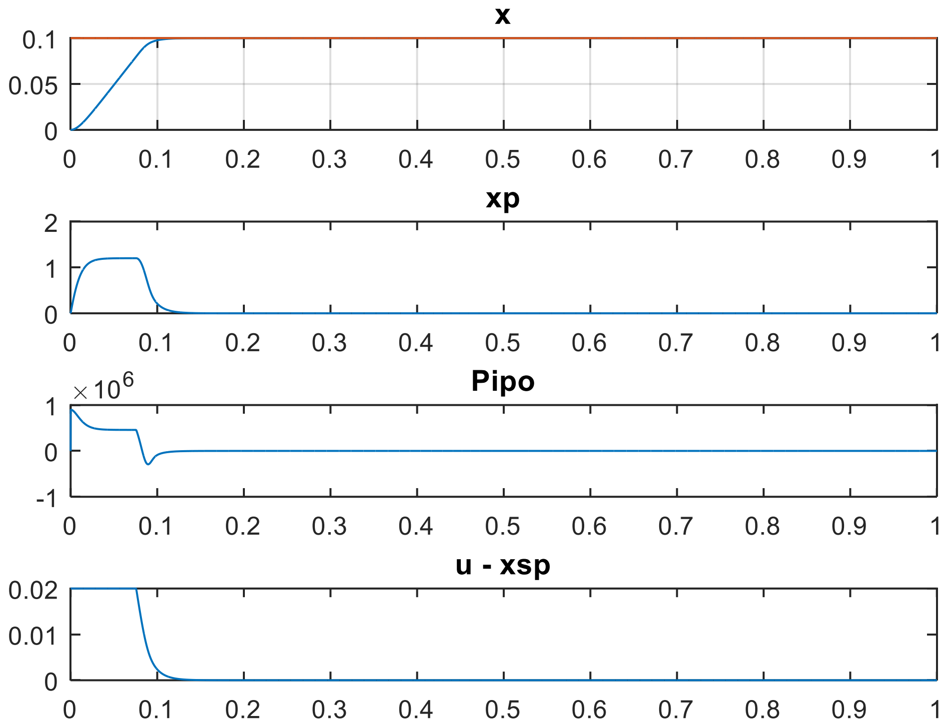


$$\boldsymbol{Pio} = \frac{\mathrm{Ps} + \mathrm{Pe}}{\mathbf{5}}$$
$$\boldsymbol{Poo} = \frac{\mathrm{Ps} + \mathrm{Pe}}{\mathbf{5}}$$

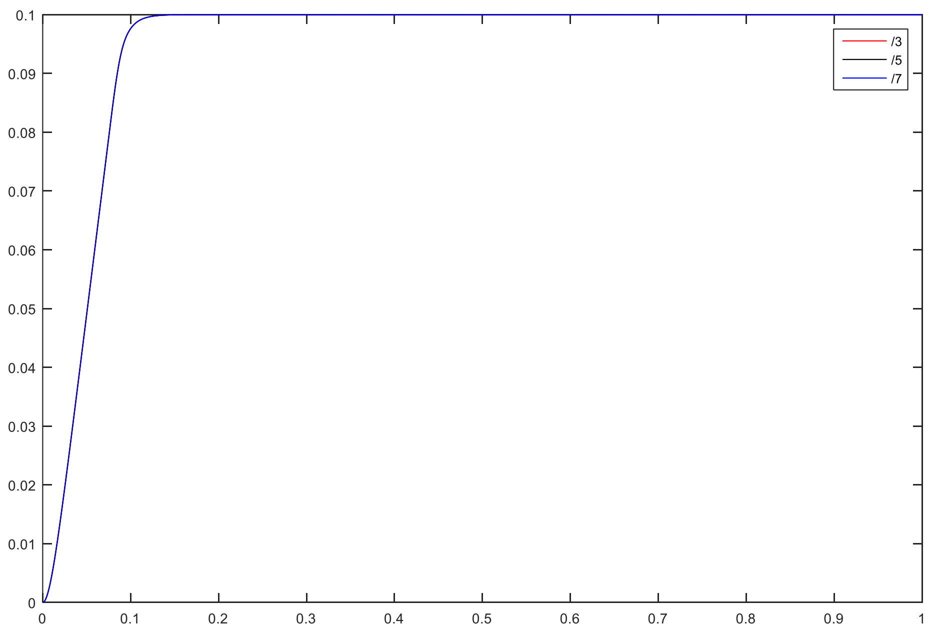


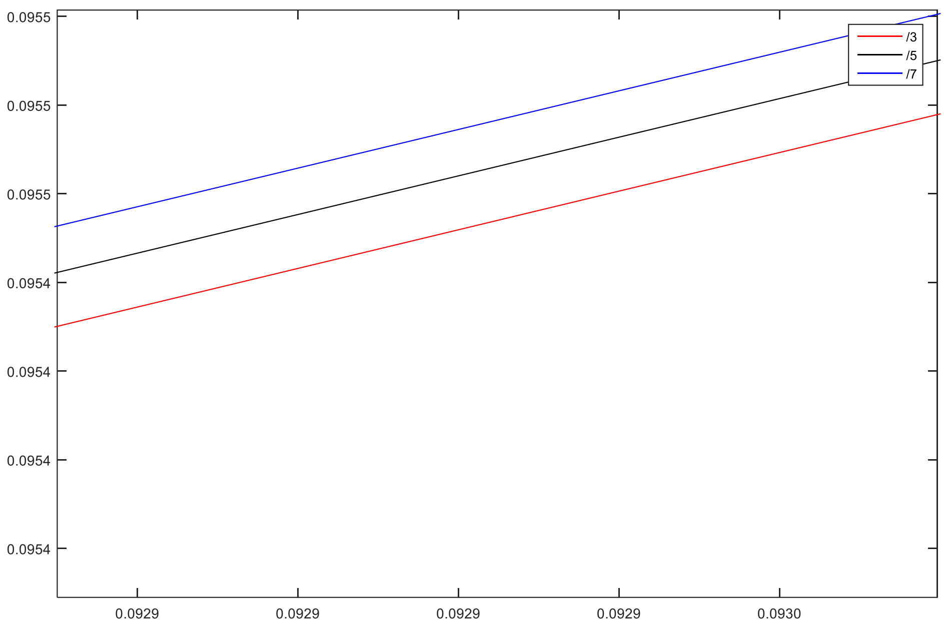
$$Pio = \frac{\mathrm{Ps} + \mathrm{Pe}}{7}$$

$$Poo = \frac{\mathrm{Ps} + \mathrm{Pe}}{7}$$



Resumen de los casos:



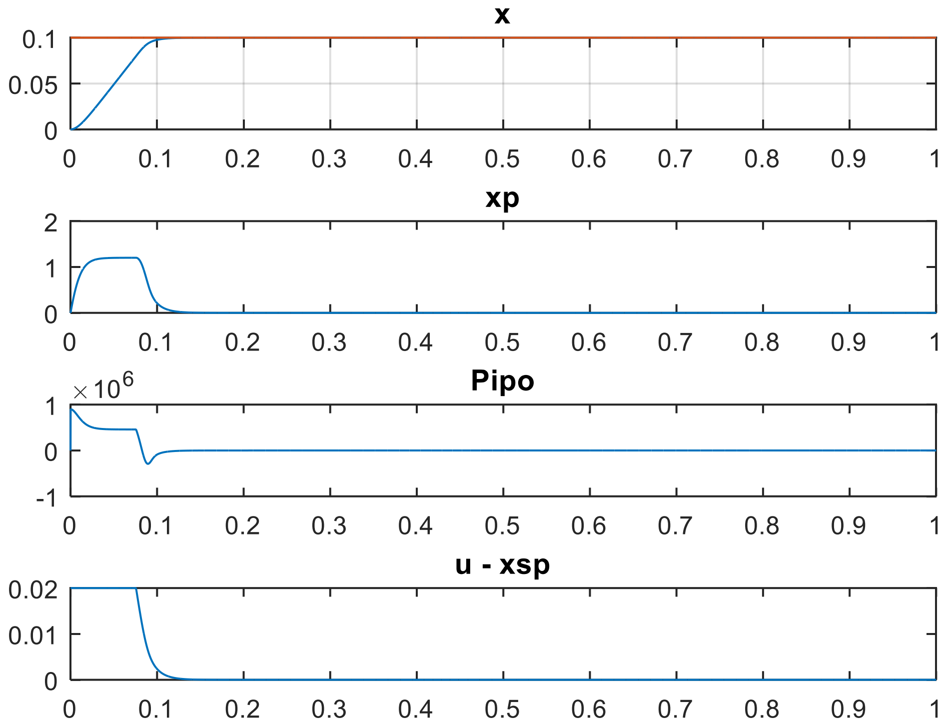


0.0955
0.0955
0.0955
0.0954
0.0954
0.0954
0.0954
0.0929
0.0929
0.0929
0.0929
0.0930
/3
/5
/7

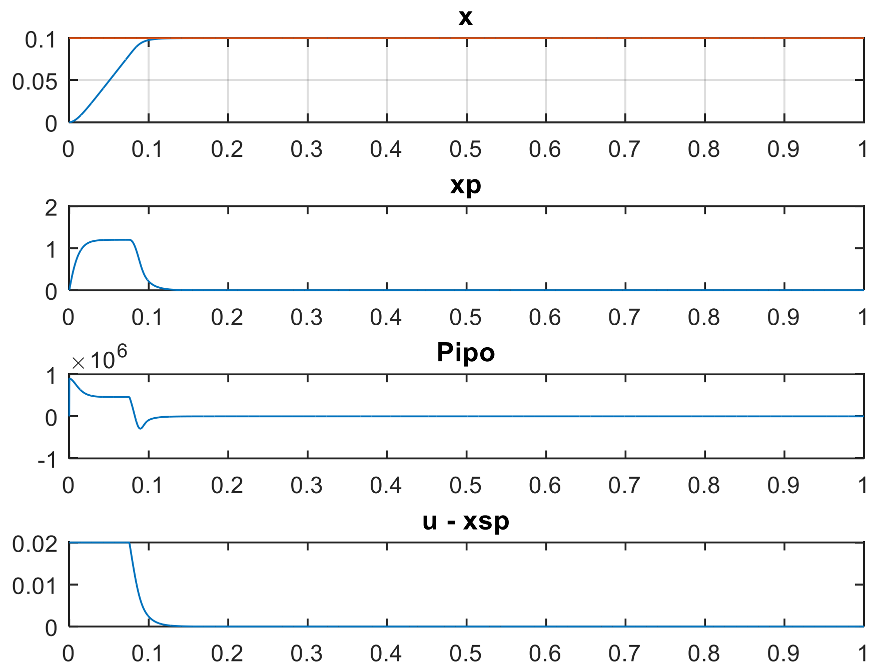
Análisis del efecto de "z" para el balanceo de las matrices A y B

Se realizan las pruebas variando el valor de z:

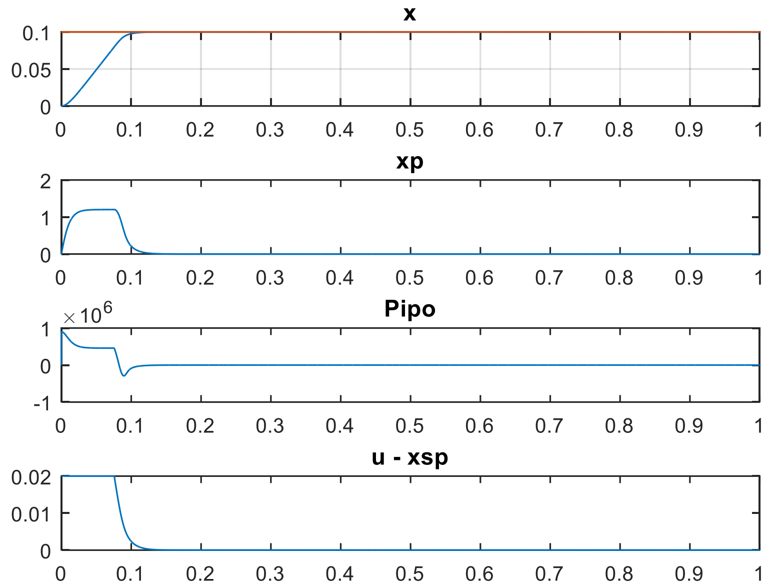
a. Prueba para qx = 1 ; qv = 0 ; qpipo = 0; x* = 0.1   z = 1e-8



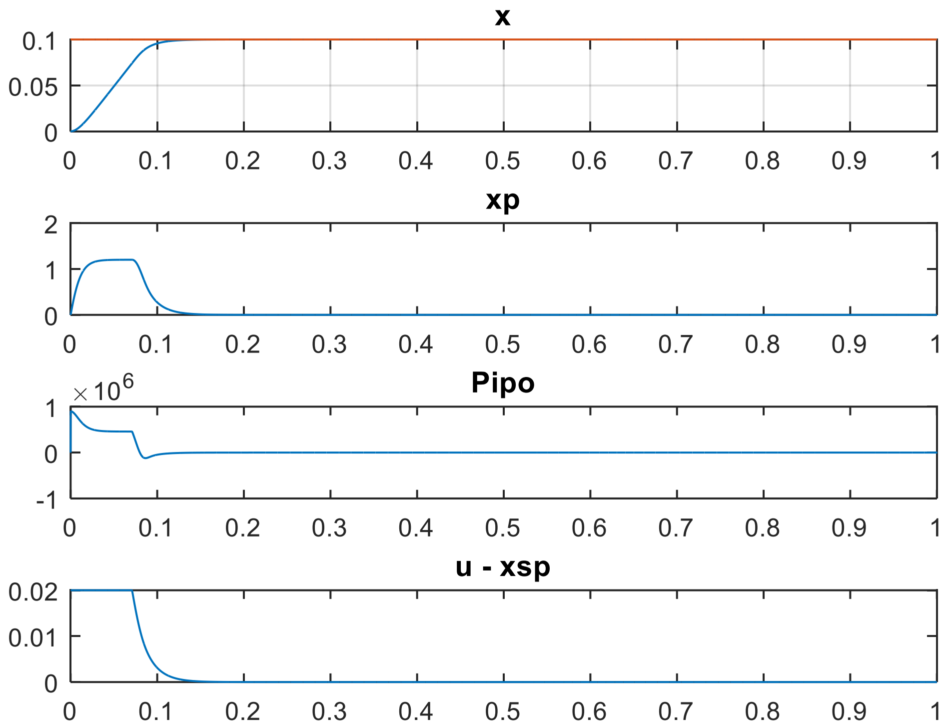
b. Prueba para qx = 1 ; qv = 0 ; qpipo = 0; x* = 0.1   z = 1e-6



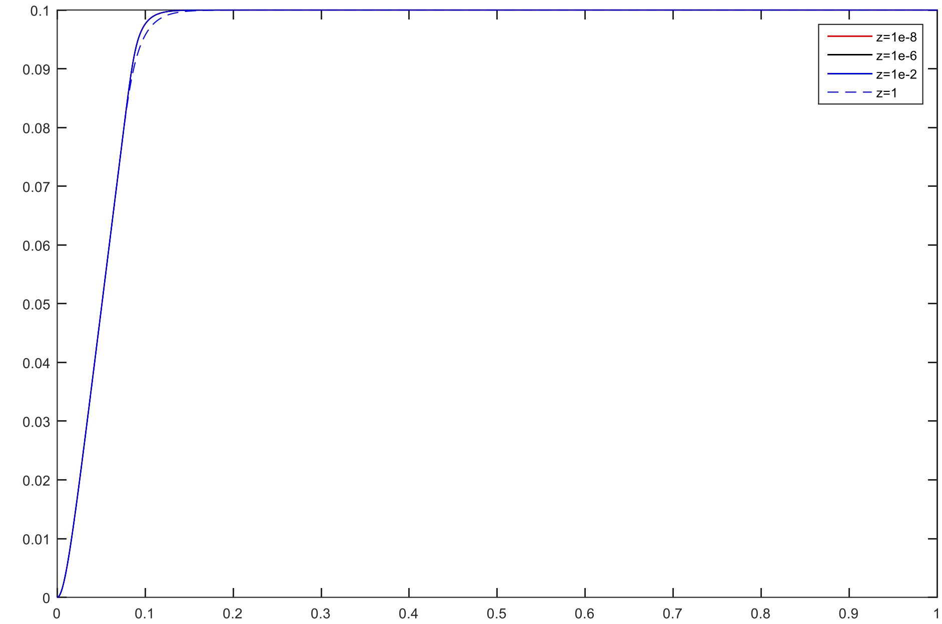
c. Prueba para qx = 1 ; qv = 0 ; qpipo = 0; x* = 0.1   z = 1e-2

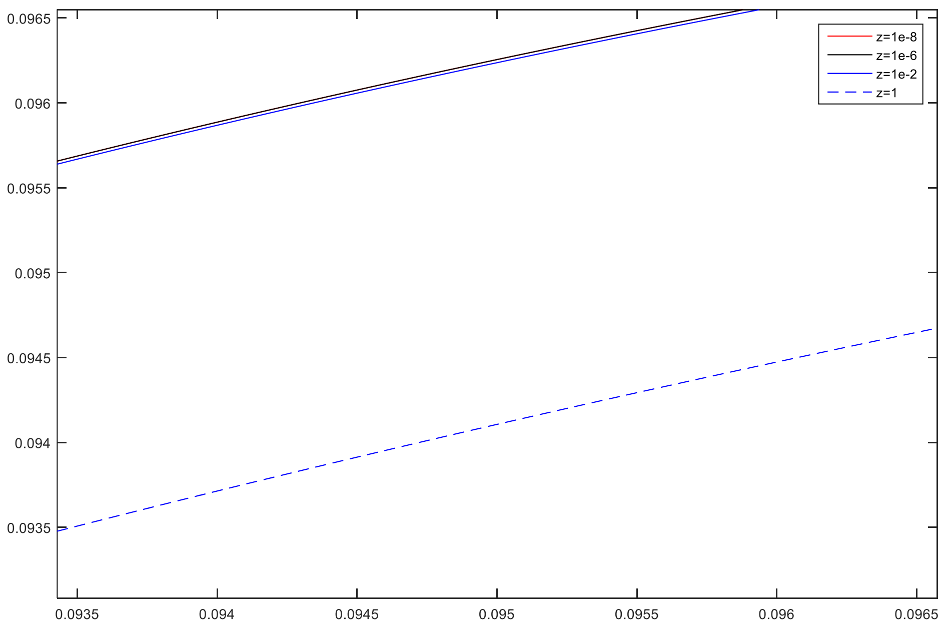


d. Prueba para qx = 1 ; qv = 0 ; qpipo = 0; x* = 0.1   z = 1



Resumen de los casos:





Conlusiones