# Primer Parcial

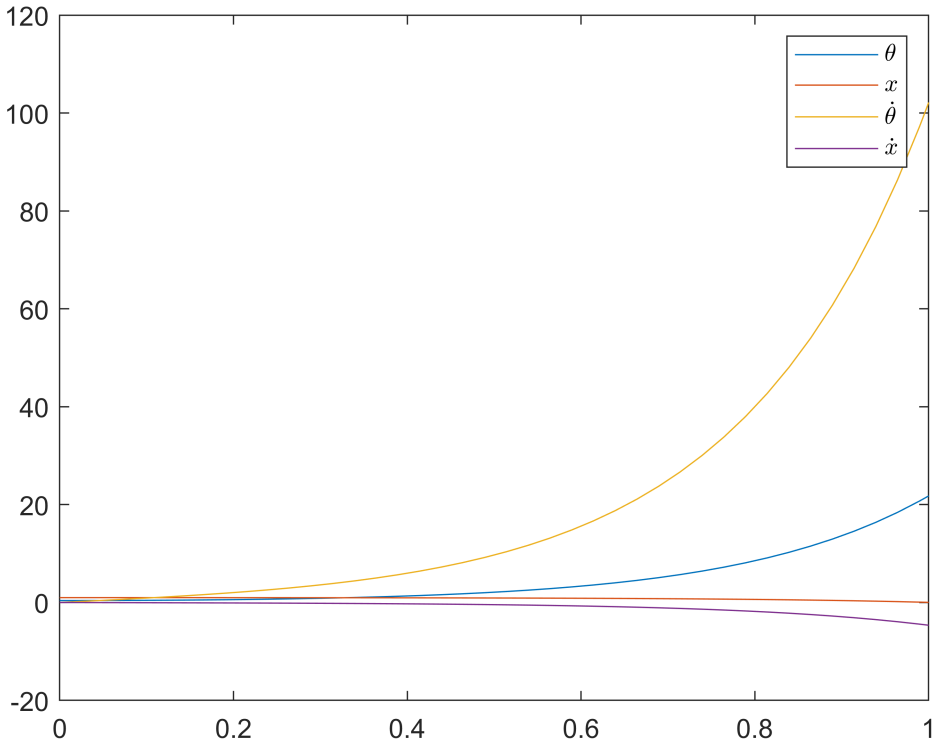
```
%problema del pendulo invertido
%la ecuacion completa seria:
%(M+n)*x''(t) + m*L*theta''(t)*cos(theta(t))+
%- m*L*[theta'(t)]^2*sin(theta(t)) = f(t)
%m*x''(t)*cos(theta(t)) + m*L*theta''(t) - m*g*sin(theta(t)) = 0
% x'' es la aceleración del carro
%x' su velocidad
%theta'' es la aceleración angular del pédulo
%theta' es la velocidad angular
%theta es el angulo que forma con la vertical

%definicion de la variables de estados
%x(1) angulo del pendulo theta
%x(2) posicion del carro x
%x(3) velocidad angular de pendulo theta'
%x(4) velocidad lineal del carro

%parametros del problema
close all
clear all
M = 1; %masa del carro
m = 0.1; %masa del pendulo
L = 0.5; %longitud del pendulo
g = 10; % aceleracion de la gravedad

%Si linearizamos en torno a theta = 0, theta' = 0: cos(theta) = 1,
% sin(theta) = theta, y theta'^2 * sin(theta) = 0 y despejamos

A = [0 0 1 0; 0 0 0 1; g*(M+m)/(L*M) 0 0 0; -m*g/M 0 0 0];
B = [0;0;-1/(M*L);1/M];
C = [1 0  0 0;0 1 0 0];
D = [0;0];
```

podemos comprobar que el sistema es asquerosamente inestable

```
lambda = eig(A)
```

```
lambda = 4×1
         0
         0
    4.6904
   -4.6904
```

```
%simulamos en lazo abierto a ver que pasa
fun =@(t,x)carro(t,x,A);
[t,x] = ode45(fun,[0,1],[0.4;1;0;0]);
plot(t,x)
legend('$\theta$','$x$','$\dot{\theta}$','$\dot{x}$','Interpreter','latex')
```




podemos realimentarlo a ver si mejora...

lo primero que hacemos como siempre es comprobar si es controlable

```
dim = rank(ctrb(A,B))
```

```
dim = 4
```

como resulta que si, podemos emplear acker o place para colocar los polos

```
K = place(A,B,[-1 -2+0.5j -2-0.5j -3])
```

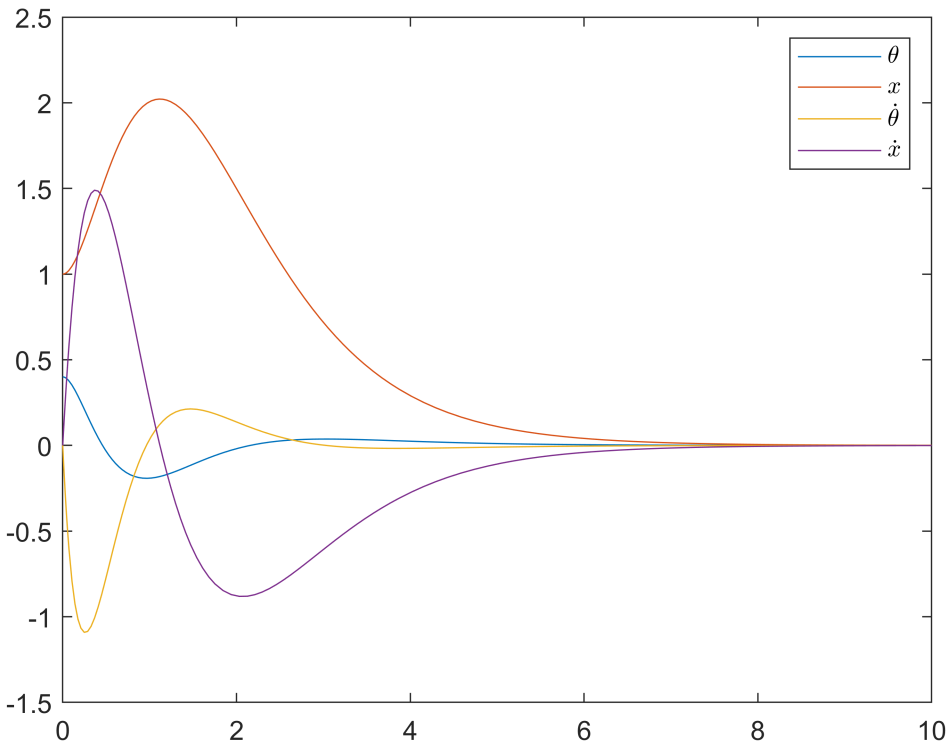
```
K = 1×4
  -22.9437   -0.6375   -4.7250   -1.4500
```

```
%Calculamos la matriz ampliada,
Are = A-B*K
```

```
Are = 4×4
         0         0    1.0000         0
         0         0         0    1.0000
  -23.8875   -1.2750   -9.4500   -2.9000
   21.9437    0.6375    4.7250    1.4500
```

```
% el resto de matrices las podemos dejar como estan
fun =@(t,x)carro(t,x,Are);
```

```
[t,x] = ode45(fun,[0,10],[0.4;1;0;0]);
plot(t,x)
legend('$\theta$','$x$','$\dot{\theta}$','$\dot{x}$','Interpreter','latex')
```




Se estabiliza, Gott sei dank!

Podemos entonces tratar de diseñar un observador, puesto que hay dos variables que no conocemos

Lo primero que hay que comprobar es si nuestro bonito sistema es

Observable. Elegimos como unica variable de salida la posicion del carro a

```
dimo = rank(obsv(A,C(2,:)))
```

```
dimo = 4
```

o welch ein jubel! Es observable debemos elegir unos polos para el observador. Los cogemos rapidillos para que el pendulo no se nos desequilibre mucho...y además reales

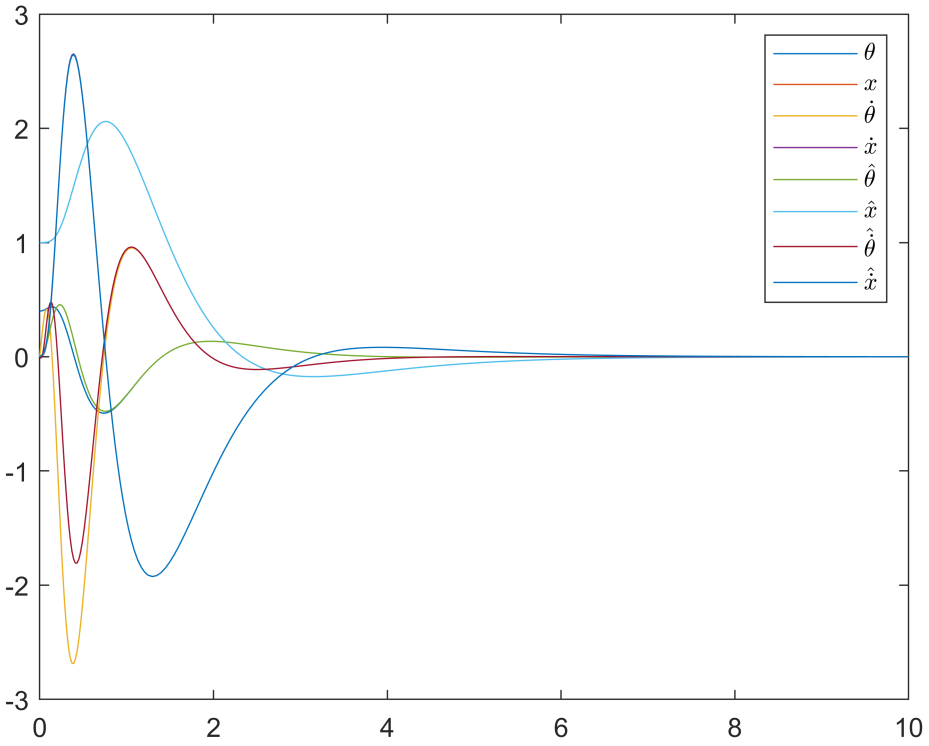
```
L = place(A',C(2,:)',[-10,-12,-13,-14])'
```

```
L = 4×1
10^4 ×
   -0.8322
    0.0049
   -4.2036
    0.0918
```

```
%podemos comprobar que pasa si montamos un sistema con realimentacion de
%estados estimados.

Aest = [A -B*K
        L*C(2,:) A-B*K-L*C(2,:)]
```

```
Aest = 8×8
10^4 ×
         0         0    0.0001         0         0         0         0         0
         0         0         0    0.0001         0         0         0         0
    0.0022         0         0         0   -0.0046   -0.0001   -0.0009   -0.0003
   -0.0001         0         0         0    0.0023    0.0001    0.0005    0.0001
         0   -0.8322         0         0         0    0.8322    0.0001         0
         0    0.0049         0         0         0   -0.0049         0    0.0001
         0   -4.2036         0         0   -0.0024    4.2035   -0.0009   -0.0003
         0    0.0918         0         0    0.0022   -0.0917    0.0005    0.0001
```

```
Best = [B;B];
fun =@(t,x)carro(t,x,Aest);
[t,x] = ode45(fun,[0,10],[0.4;1;0;0;0;1;0;0]);
plot(t,x)
legend('$\theta$','$x$','$\dot{\theta}$','$\dot{x}$','$\hat{\theta}$','$\hat{x}$','$\hat{\dot{\
```




```
%introducimos accion integral
%Considerando que solo podemos controlar la posición
```

```
Amp = [A zeros(4,1); C(2,:) 0]
```

```
Amp = 5×5
     0     0     1     0     0
     0     0     0     1     0
    22     0     0     0     0
    -1     0     0     0     0
     0     1     0     0     0
```

```
Bamp =[B;-1];
%vamos a comprobar que la cosa es controlable
dim = rank(ctrb(Amp,Bamp))
```

```
dim = 5
```

El sistemam añadiendo la componente del error integral es también controlable. Añadimos un  polo para la acción integral.

```
Ka = acker(Amp,Bamp,[ -1 -2+0.5j -2-0.5j -3 -10])
```

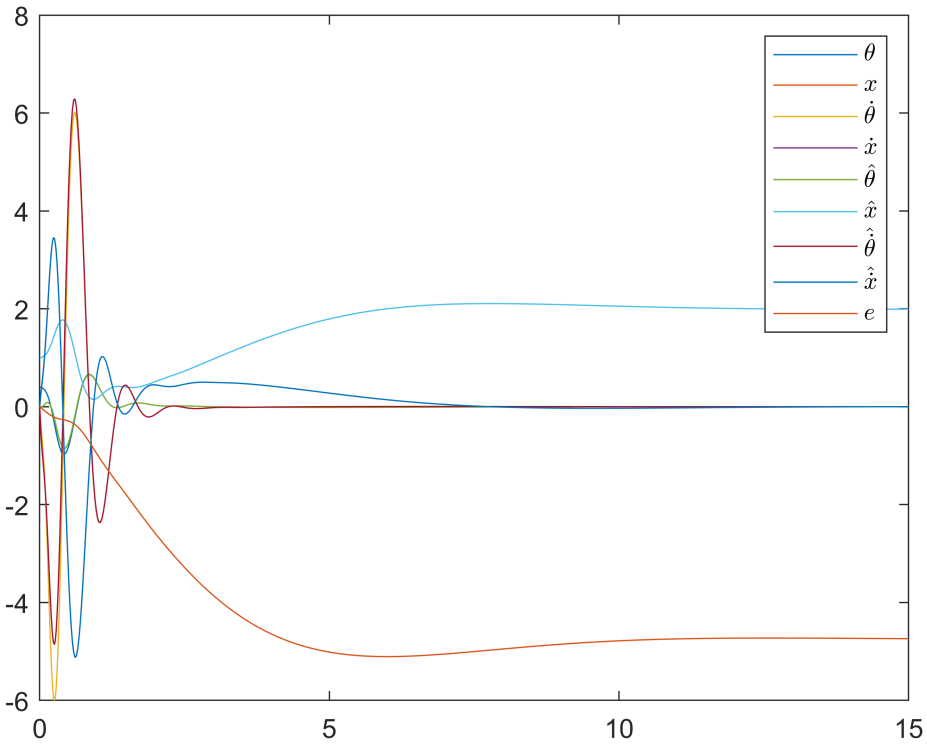
```
Ka = 1×5
  -70.1937  -15.1375  -16.0156  -20.4062   -6.3750
```

Construimos la matriz de estados con la realimentación calculada, mantenemos los polos del observador calculados antes

```
Aint = [A -B*Ka(1:4) -B*Ka(5)
        L*C(2,:) A-B*Ka(1:4)-L*C(2,:) -B*Ka(5)
        zeros(1,4) C(2,:) 0]
```

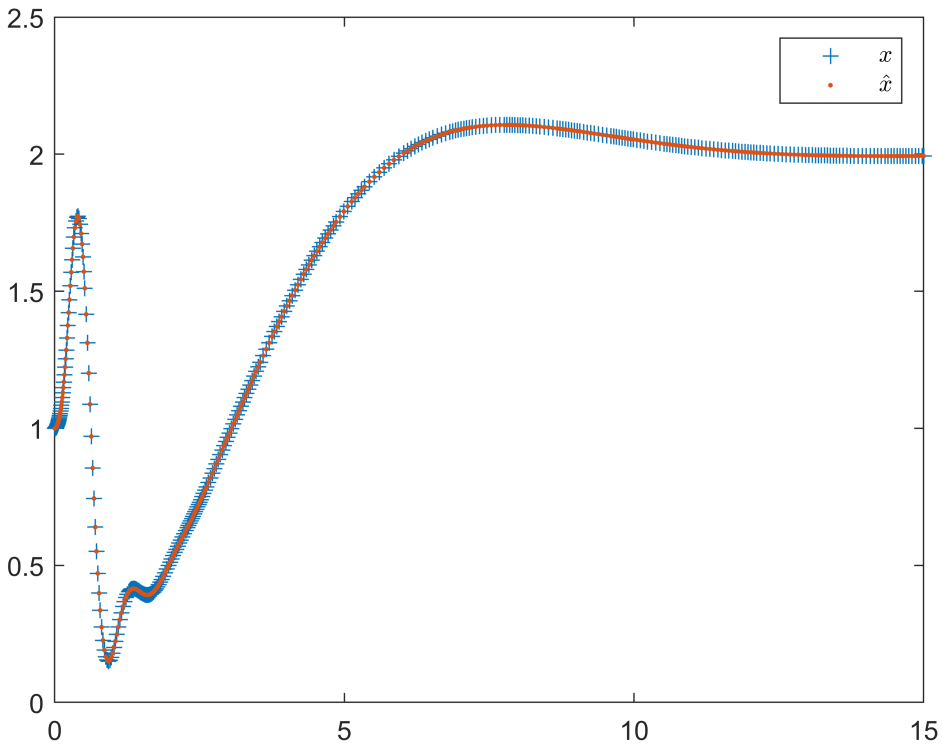
```
Aint = 9×9
10^4 ×
         0         0    0.0001         0         0         0         0         0 ···
         0         0         0    0.0001         0         0         0         0
    0.0022         0         0         0   -0.0140   -0.0030   -0.0032   -0.0041
   -0.0001         0         0         0    0.0070    0.0015    0.0016    0.0020
         0   -0.8322         0         0         0    0.8322    0.0001         0
         0    0.0049         0         0         0   -0.0049         0    0.0001
         0   -4.2036         0         0   -0.0118    4.2006   -0.0032   -0.0041
         0    0.0918         0         0    0.0069   -0.0903    0.0016    0.0020
         0         0         0         0         0    0.0001         0         0
```

```
fun =@(t,x)carroint(t,x,Aint,2);
[t,x] = ode45(fun,[0,15],[0.4;1;0;0;0;1;0;0;0]);
plot(t,x)
legend('$\theta$','$x$','$\dot{\theta}$','$\dot{x}$','$\hat{\theta}$','$\hat{x}$','$\hat{\dot{\
```




Como queda un poco confuso, vamos a dibujar de nuevo solo $x$ y $\widehat{x}$

```
plot(t,x(:,2),'+')
hold on
plot(t,x(:,6),'.')
legend('$x$','$\hat{x}$','interpreter','latex')
```




Si intentamos aplicar control integral a la vez a la posición del bloque y al ángulo del péndulo, obtenemos,

```
Amp2 = [A zeros(4,2); C zeros(2,2)]
```

```
Amp2 = 6×6
     0     0     1     0     0     0
     0     0     0     1     0     0
    22     0     0     0     0     0
    -1     0     0     0     0     0
     1     0     0     0     0     0
     0     1     0     0     0     0
```

```
Bamp2 =[B;-1;-1]
```

```
Bamp2 = 6×1
     0
     0
    -2
     1
    -1
    -1
```

```
dim =rank(ctrb(Amp2,Bamp2))
```

```
dim = 5
```

Necesitaríamos que el rango fuera 6 para que el sistema completo fuera controlable

## Segundo Parcial

### Ejercicio 1

Creamos las variables simbólicas del sistema

```
syms x1 x2 real
f =[1/x2-x1^2;x1/x2-1]
```

```
f =
```

$$\begin{pmatrix} \frac{1}{x_2} - {x_1}^2 \\ \frac{x_1}{x_2} - 1 \end{pmatrix}$$

Buscamos los puntos de equilibrio

```
roots = solve(f,[x1,x2])
```

```
roots = struct with fields:
    x1: 1
    x2: 1
```

Solo hay un punto de equilibrio en $(1, 1)$

Linealizamos en torno a dicho punto de equilibrio,

```
J = jacobian(f,[x1,x2])
```

```
J =
```

$$\begin{pmatrix} -2\,x_1 & -\frac{1}{{x_2}^2} \\ \frac{1}{x_2} & -\frac{x_1}{{x_2}^2} \end{pmatrix}$$

```
Jn = subs(J,[x1,x2],[1,1])
```

```
Jn =
```

$$\begin{pmatrix} -2 & -1 \\ 1 & -1 \end{pmatrix}$$

```
lambda = eig(Jn)
```

```
lambda =
```

$$\begin{pmatrix} -\frac{3}{2} - \frac{\sqrt{3}\,\mathrm{i}}{2} \\ -\frac{3}{2} + \frac{\sqrt{3}\,\mathrm{i}}{2} \end{pmatrix}$$

Así que resulta que al menos localmente el punto de equilibrio es estable

```
V = (x1-1)^2/2 + (x2-1)^2/2
```

```
V =
```

$$\frac{(x_1-1)^2}{2}+\frac{(x_2-1)^2}{2}$$

```
dotV = expand(gradient(V)'*f)
```

```
dotV =
```

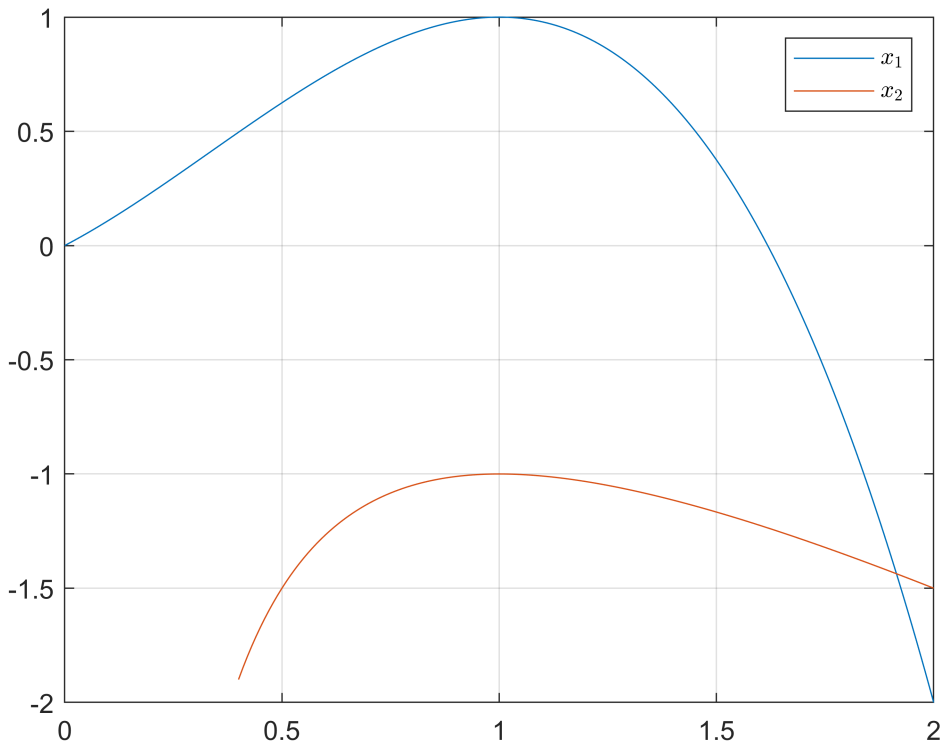
$$x_1 - x_2 + x_1^2 - \frac{1}{x_2} - x_1^3 + 1$$

Podemos reordenar un poco la contribución de las dos variables, eso nos permite dibujarlas juntas y ver qué pasa

$$\dot{V} = (-x_1^3 + x_1^2 + x_1) + \left(1 - x_2 - \frac{1}{x_2}\right)$$

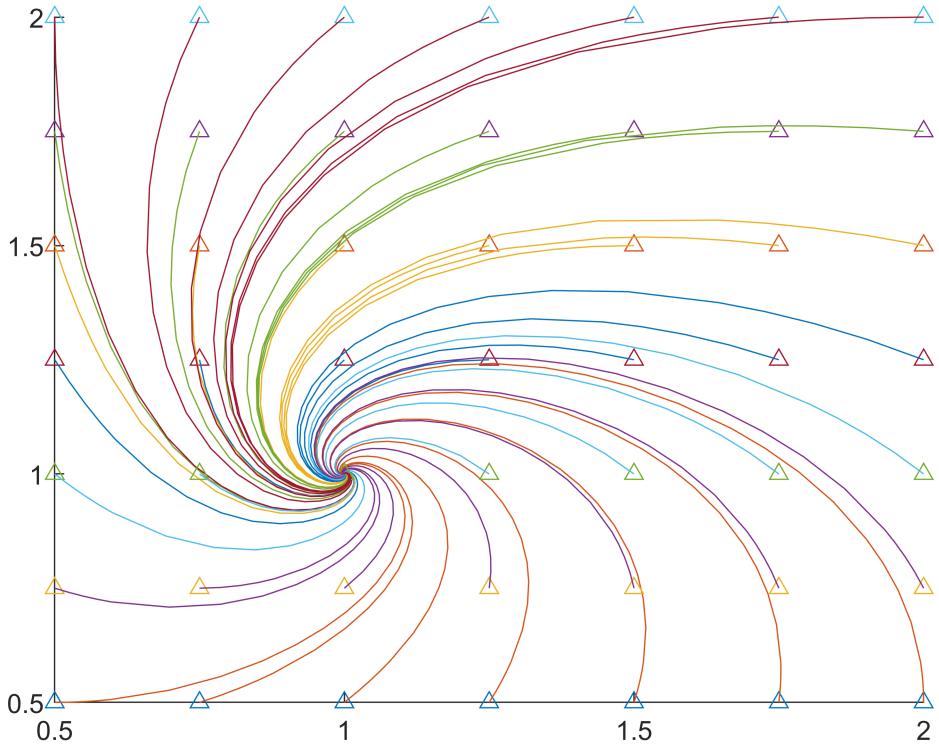
Es fácil ver que $x_2$ va a ser siempre negativa, su valor máximo es $-1$ precísamente en el punto de equilibrio. Para $x_1$ es fácil ver que su valor máximo es 1 precísamente en el mismo punto. por tanto obtenemos que el sistema es estable. de hecho converge para todo valor $x_1, x_2 \in \mathbb{R}_+$

```
x1n = 0:0.01:2;
x2n = 0.4:0.01:2;
dotVx1 = -x1n.^3+x1n.^2+x1n;
dotVx2 = -x2n +1 -1./x2n;
figure()
plot(x1n,dotVx1)
hold on
plot(x2n,dotVx2)
grid()
legend('$x_1$','$x_2$','interpreter','latex')
```




```
fun =@(t,x)chisme(t,x);
x1= 0.5:0.25:2;
x2 = 0.5:0.25:2;
figure()
for i = 1:length(x1)
    for j =1: length(x2)
    [t,x] = ode45(fun,[0,10],[x1(i);x2(j)]);
```

```
    hold on
    plot(x1(i),x2(j),'^')
    plot(x(:,1),(x(:,2)))
    end
end
```




**Ejercicio 2.**

Creamos variables simbólicas para analizar el sistema

```
syms x1 x2 b u k real
f = [x2;-x1-x2*(x1^2+b)]
```

```
f =
```

$$\begin{pmatrix} x_2 \\ -x_1 - x_2\left(x_1^2 + b\right) \end{pmatrix}$$

```
V = (x1^2+x2^2)/2
```

```
V =
```

$$\frac{x_1^2}{2} + \frac{x_2^2}{2}$$

```
dotV = simplify(gradient(V)'*f)
```

```
dotV =
```

$$-x_2^2\left(x_1^2 + b\right)$$

Si $b \geq 0$ Esto es estable para todo $\mathbb{R}^2$. Si $b < 0$ entonces la región de estabilidad esta acotada, dentro de una bola de radio $r = \sqrt{b}$

Si añadimos realimetación

```
fr = [x2;-x1-x2*(x1^2+b)-k*x2]
```

fr =

$$\begin{pmatrix} x_2 \\ -x_1 - k\,x_2 - x_2\ \left(x_1{}^2 + b\right) \end{pmatrix}$$

```
dotV = simplify(gradient(V)'*fr)
```

dotV = $-x_2{}^2\ \left(x_1{}^2 + b + k\right)$

Si elegimos $k \geq -b$, para $b < 0$ el sistema es estable empecemos donde empezemos. Además, podemos limitar la región de estabilidad jugando con el valor de $k$. La región de estabilidad tendra radio $k + b$.

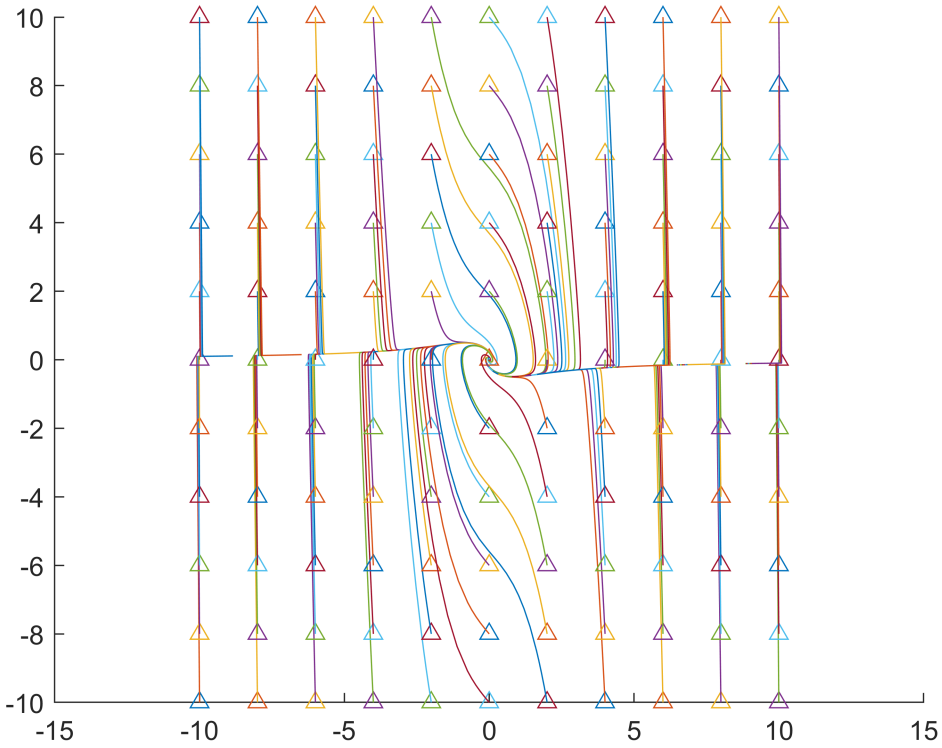
```
k = 0
```

k = 0

```
b = 1 %estable en lazo abierto
```

b = 1

```
fun =@(t,x)eje2(t,x,b,k);
x1= -10:2:10;
x2 = -10:2:10;
figure()
for i = 1:length(x1)
    for j =1: length(x2)
    [t,x] = ode45(fun,[0,10],[x1(i);x2(j)]);
    hold on
    plot(x1(i),x2(j),'^')
    plot(x(:,1),(x(:,2)))
    end
end
```



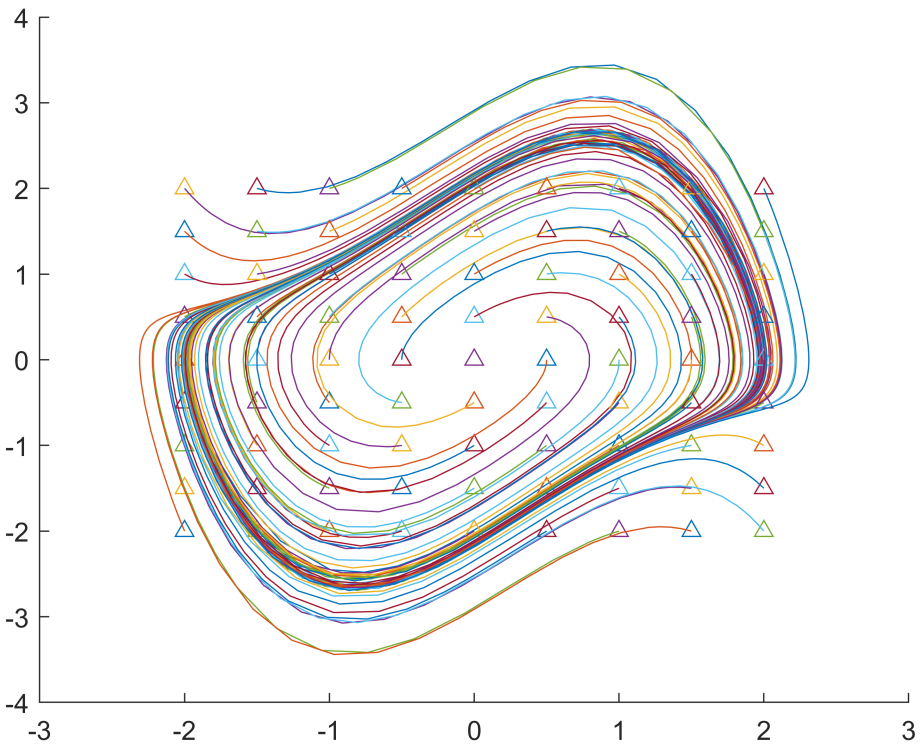

```
k = 0
```

```
k = 0
```

```
b = -1 %Inestable, aparece un cíclo límite
```

```
b = -1
```

```
fun =@(t,x)eje2(t,x,b,k);
x1= -2:0.5:2;
x2 = -2:0.5:2;
figure()
for i = 1:length(x1)
    for j =1: length(x2)
    [t,x] = ode45(fun,[0,10],[x1(i);x2(j)]);
    hold on
    plot(x1(i),x2(j),'^')
    plot(x(:,1),(x(:,2)))
    end
end
```



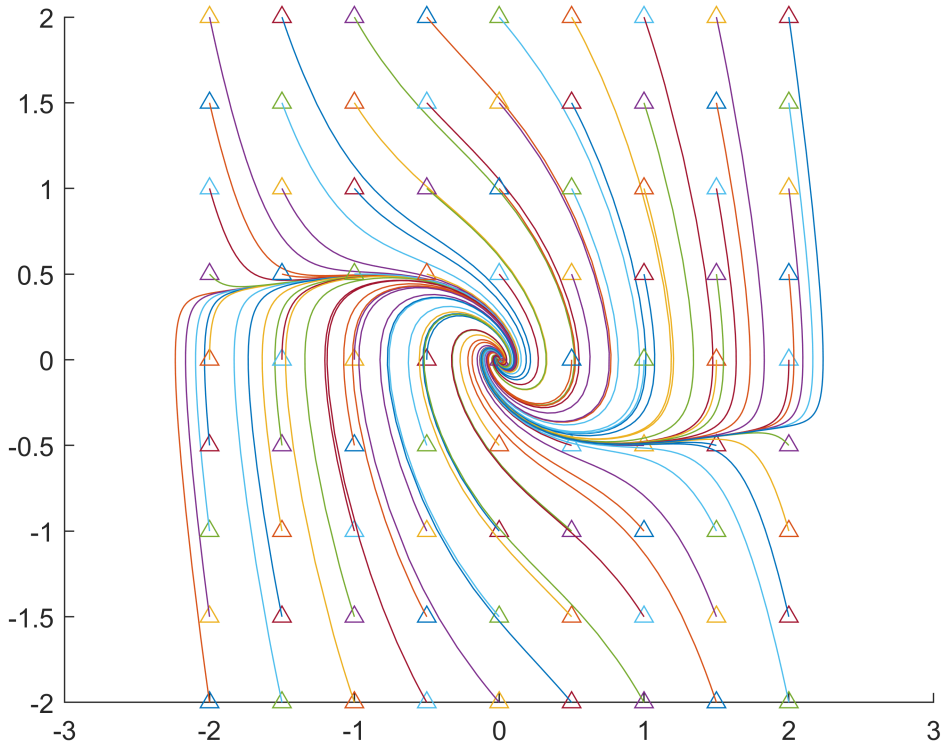

```
k = 2 %restabilizamos el sistema eligiendo una ganancia k > -b
```

```
k = 2
```

```
b = -1
```

```
b = -1
```

```
fun =@(t,x)eje2(t,x,b,k);
x1= -2:0.5:2;
x2 = -2:0.5:2;
figure()
for i = 1:length(x1)
    for j =1: length(x2)
    [t,x] = ode45(fun,[0,10],[x1(i);x2(j)]);
    hold on
    plot(x1(i),x2(j),'^')
    plot(x(:,1),(x(:,2)))
    end
end
```




```
function dotx = carro(t,x,A)
%siempre vamos a realimentar
dotx = A*x;
end
function dotx = carroint(t,x,A,yd)
%añadimos la entrada correspondiente al valor deseado de la salida
dotx = A*x + [zeros(8,1);-1]*yd;
end
function dotx = eje2(t,x,b,k)
%tomar k cero para el sistema sin realimentación
dotx(1,1) = x(2);
dotx(2,1) = -x(1)-x(2)*(x(1)^2+b+k);
end

function dotx = chisme(t,x)
dotx(1,1) = -x(1) + 1/x(2);
dotx(2,1) =  x(1)/x(2) - 1;
end
```