# EECS 562 & AEROSP 551
# Inverted Pendulum on a Cart Project

**Due date:** Tuesday, February 16, 2023 by 23:59 EST on Canvas
**Firm Deadline:** No extensions will be allowed.


Swayampakula
*LAST NAME* (PRINTED), 

Rahul Kashyap
*FIRST NAME*


**HONOR PLEDGE:** Copy and SIGN: I understand that this is an individual assignment and not a group exercise. I have neither given nor received aid on this project nor have I concealed any violation of the Honor Code. I have neither borrowed nor shared computer code for simulating my models and creating my plots.

s.rahul
SIGNATURE

**This Should Be The Cover Page Of Your Project**

Project EECS 565

A. Using linear state variable feedback and with the nonlinear model initialized at x2(0) = 0, x3(0) = 0, x4(0) = 0, the largest I could make x1(0) before the closed loop system went unstable was **0.69 (radians)**.

   The linear state variable feedback that I used was (give the feedback gains, upto 2 digits after the decimal point is sufficient for all answers. )

```
K =

   16.7956   18.7852    1.2563    4.3654
```

B. Using linear state variable feedback plus an observer, with the observer initialized at the origin and the nonlinear model initialized at x2(0) = 0, x3(0) = 0, x4(0) = 0, the largest I could make x1(0) before the closed-loop system went unstable was **0.32 (radians).**

   The observer that I used to obtain the best response was **nonlinear** and the observer gain was .

```
L =

   10.8143    0.9908
   29.6105    5.4448
    0.9745   11.1857
    4.2416   30.5145
```

**Question 1:**

1)

$$\frac{d^2\phi}{d\bar{t}^2} = \frac{d}{d\bar{t}}\Big(\underbrace{\frac{d}{d\bar{t}}\phi}_{\varphi}\Big) = \frac{d\varphi}{d\bar{t}} = \frac{d\varphi}{dt}\cdot\frac{dt}{d\bar{t}}$$

$$= \frac{d}{dt}\left(\frac{d}{d\bar{t}}\phi\right)\cdot\frac{dt}{d\bar{t}}$$

$$= \frac{d}{dt}\left(\left(\frac{d}{dt}\phi\right)\cdot\frac{dt}{d\bar{t}}\right)\cdot\frac{dt}{d\bar{t}}$$

$$= \frac{d^2\phi}{dt^2}\cdot\left(\frac{dt}{d\bar{t}}\right)^2 = \frac{d^2\phi}{dt^2}\left(\frac{d}{d\bar{t}}(\bar{t}T)\right)^2$$

$$= \left(\frac{d^2\phi}{dt^2}\right)\cdot T^2$$

2)

$$\frac{d^2\phi}{dt^2} = \frac{1}{T^2}\left(\frac{d^2\phi}{d\bar{t}^2}\right)$$

$$\frac{d^2\phi}{d\bar{t}} = \frac{-c\dot{\phi}^2\sin(\phi)\cos(\phi) + \sin\phi - \cos\phi\,\bar{u}}{1 + c\sin^2\phi}$$

$$= \frac{-C\dot{\phi}^2 \sin\phi\cos\phi + \sin\phi - \cos\phi\,\bar{u}}{1 + \left(\frac{m^2L^2}{J(m+M) + mML^2}\right)\sin^2\phi}$$

$$= \frac{(J(m+M) + mML^2)(-C\dot{\phi}^2 \sin\phi\cos\phi + \sin\phi - \cos\phi\,\bar{u})}{(J(m+M) + mML^2) + m^2L^2\sin^2\phi}$$

$$\therefore \frac{(J(m+M) + mML^2)(-C\dot{\phi}^2 \sin\phi\cos\phi + \sin\phi - \cos\phi\,\bar{u})}{J(m+M) + mML^2 + m^2L^2 - m^2L^2\cos^2\phi}$$

$$\therefore \frac{(J(m+M) + mML^2)(-C\dot{\phi}^2 \sin\phi\cos\phi + \sin\phi - \cos\phi\,\bar{u})}{(J+mL^2)(M+m) - m^2L^2\cos^2\phi}$$

let $k = (J+mL^2)(M+m) - m^2L^2\cos^2\phi$

$P = (J(m+M) + mML^2)$

$$\therefore \left(\frac{\cancel{P}}{k}\right)\left(-\frac{m^2L^2\dot{\phi}^2 \sin\phi\cos\phi}{\cancel{P}} + P\sin\phi - P\cos\phi\,\bar{u}\right)$$

$\therefore$

$$\frac{\left(-(M+m)g\,m^2L^2\dot{\phi}^2\sin\phi\cos\phi\right) \cancel{+P\sin\phi} + P\sin\phi\,(M+m)g - P\cos\phi\, u}{K\quad (M+m)g}$$

$$\frac{d^2\phi}{dt^2} = \frac{1}{T^2}\frac{d^2\phi}{d\bar{t}^2}$$

$$= \frac{m\,\bar{t}^2 (m+M)\,g}{K\left(J(m+M) + Mm\,L^2\right)}\frac{d^2\phi}{d\bar{t}}$$

$$= \frac{m\,L\,\cancel{g\,(m+M)}}{P}\left(\frac{\left(-(m+M)g\,m^2L^2\dot{\phi}^2\sin\phi\cos\phi\right) + P\sin\phi\,(m+M)g - P\cos\phi\, u}{K\,\cancel{(M+m)\,g}}\right)$$

$$= \frac{1}{K}\left\{\left(mL\sin\phi\,(m+M)g\right) - \left(mL\cos\phi\, u\right) - \left(\frac{(m+M)g\,m^2L^3\dot{\phi}^2\sin\phi\cos\phi}{P}\right)\right\}$$

converting $\dot{\phi}$ w.r.t $\bar{t}$ to $t$

$$\Rightarrow \frac{mL\sin\phi(m+M)g - mL\cos\phi\, u - (mL\dot{\phi})^2\sin\phi\cos\phi}{K}$$

where

$$K = (J + mL^2)(m+M) - m^2L^2\cos^2\phi$$

**Question 2:**

2)

$$x = [\,\phi \;\; \dot{\phi} \;\; \bar{s} \;\; \dot{\bar{s}}\,]^T$$

$$\dot{x} = [\,\dot{\phi} \;\; \ddot{\phi} \;\; \dot{\bar{s}} \;\; \ddot{\bar{s}}\,]^T$$

$$\Rightarrow \dot{x} = \begin{bmatrix} \dot{\phi} \\ \frac{1}{T^2}\left( \frac{-c\dot{\phi}^2 \sin\phi\cos\phi + \sin\phi}{1 + c\sin^2\phi} - \frac{\cos\phi}{1 + c\sin^2\phi}\,\bar{u} \right) \\ \dot{\bar{s}} \\ \frac{1}{T^2}\left( \frac{d\dot{\phi}^2 \sin\phi - \cos\phi\sin\phi}{1 + c\sin^2\phi} + \frac{b}{1 + c\sin^2\phi}\,\bar{u} \right) \end{bmatrix}$$

2) for jacobian

$x = [\phi \ \dot\phi \ s \ \dot s]^T$

$\Rightarrow \dot x = [\dot\phi \ \ddot\phi \ \dot s \ \ddot s]^T$

$\frac{\partial \dot\phi}{\partial x} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}^T \qquad \frac{\partial \dot s}{\partial x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T$

$$\frac{\partial \ddot\phi}{\partial \phi} = \frac{\partial}{\partial \phi}\left(\left(\frac{-c\dot\phi^2 \sin\phi \cos\phi + \sin\phi}{1 + c\sin^2\phi}\right) - \frac{\cos\phi}{(1 + c\sin^2\phi)}\bar u\right) \cdot \frac{1}{T^2}$$

$$= \cancel{\frac{(1+c\sin^2\phi)(2c\dot\phi^2(\cos^2\phi - \sin^2\phi) + \sin\phi) - (2c\sin\phi\cos\phi)(-u)}{(1+c\sin^2\phi)^2}}$$

$$= \left(\frac{(1+c\sin^2\phi)(-c\dot\phi^2(\cos^2\phi - \sin^2\phi) + \cos\phi) - (2c\sin\phi\cos\phi)(-c\dot\phi^2\sin\phi\cos\phi + \sin\phi)}{(1+c\sin^2\phi)^2}\right.$$

$$\left. - \frac{\partial}{\partial\phi}\left(\left(\frac{\cos\phi}{(1+c\sin^2\phi)}\right)\bar u\right)\right) \cdot \frac{1}{T^2}$$

= Substituting at $\phi = 0$

$$\Rightarrow \frac{\partial \ddot\phi}{\partial \phi} = \left(1 - \left[\frac{\partial}{\partial\phi}\left(\frac{\cos\phi}{1+c\sin^2\phi}\right)\bar u + \left(\frac{\cos\phi}{1+c\sin^2\phi}\right)\frac{\partial \bar u}{\partial \phi}\right]\right)\frac{1}{T^2}$$

$\bar u = 0 \qquad \frac{\partial \bar u}{\partial \phi} = 0$ assuming $\bar u$ is independent

$$\Rightarrow \left(\frac{\partial \ddot{\phi}}{\partial \phi}\right) = \frac{1}{T^2}$$

The $\frac{1}{T^2}$ term is due to conversion from derivative w.r.t $\bar{t}$ to $t$

$$\frac{\partial \ddot{\phi}}{\partial \dot{\phi}} = \frac{-2c\dot{\phi}\sin\phi\cos\phi}{1+c\sin^2\phi} + 0 + 0.$$

at $\dot{\phi} = 0$, $\phi = 0$

$$\Rightarrow \frac{\partial \ddot{\phi}}{\partial \dot{\phi}} = 0.$$

$$\frac{\partial \ddot{s}}{\partial s} = 0 \qquad \frac{\partial \ddot{s}}{\partial \dot{s}} = 0$$

$$\frac{\partial \ddot{s}}{\partial \phi} = \frac{1}{T^2}\left(\frac{\partial}{\partial \phi}\left(\frac{d\dot{\phi}^2\sin\phi - \cos\phi\sin\phi}{1+c\sin^2\phi}\right) + \cancel{\frac{\partial}{\partial \phi}\left(\frac{b}{1+c\sin^2\phi}\bar{u}\right)}^{0}\right)$$

$$= \frac{1}{T^2}\left(\frac{(1+c\sin^2\phi)\left(d\dot{\phi}^2\cos\phi - (-\sin^2\phi + \cos^2\phi)\right) - (2c\sin\phi\cos\phi)^{0}(\ldots}{(1+c\sin^2\phi)^2}\right.$$

Substituting $x = 0$.

$$\Rightarrow \frac{\partial \ddot{s}}{\partial \phi} = -\frac{1}{T^2}$$

$$\frac{\partial \ddot{s}}{\partial \dot{\phi}} = \frac{2\dot{\phi}\, d\sin\phi}{1+c\sin^2\phi} + 0 + 0$$

$x \to 0$

$$\Rightarrow \frac{\partial \ddot{s}}{\partial \dot{\phi}} = 0$$

$$\therefore \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1/T^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1/T^2 & 0 & 0 & 0 \end{bmatrix}$$

for B

$$\frac{\partial \dot{\phi}}{\partial u} = 0$$

$$\frac{\partial \ddot{\phi}}{\partial u} = \frac{1}{T^2}\left(\frac{-\cos\phi}{1+c\sin^2\phi}\,\frac{\partial u}{\partial u}\right) \quad \text{at } x=0$$

$$\boxed{\frac{\partial \ddot{\phi}}{\partial u} = \frac{-1}{T^2}}$$

$$\frac{\partial \dot{s}}{\partial u} = 0 \qquad \frac{\partial \ddot{s}}{\partial u} = \left(\frac{b}{1+c\sin^2\phi}\right)\frac{\partial u}{\partial u}\,\frac{1}{T^2}$$

at $x=0$

$$\Rightarrow \frac{\partial \ddot{s}}{\partial u} = \frac{b}{T^2}$$

$$B = \begin{bmatrix} 0 \\ -1/T^2 \\ 0 \\ b/T^2 \end{bmatrix}$$

**Code Problem 2**:

```
%% Problem 2 - Values from the calculation

A = [0,1,0,0;
    1/const,0,0,0;
    0,0,0,1;
    -1/const,0,0,0]; % Linearized value of A over origin

B = [0;-1/const;0;b/const]; % Linearized value of B over origin

p = [-3,-2, -0.7+((1j)*0.2),-0.7-((1j)*0.2)]; % Desired eigen values
```

—-------------------------------------------------------------------

```
function x_dot = nonlin_sys(t,x,b,c,d,const,K,u,check)

%% The file is for the nonlinear dynamics model

x_dot = zeros(4,1);
term = 1/(1+(c*sin(x(1))*sin(x(1))));


phi = x(1);
phi_dot = x(2);
s = x(3);
s_dot = x(4);

B = [0;(-cos(phi)*(term/const));0;(b*(term/const))];

x_dot(1) = phi_dot;
x_dot(2) = ((-c*phi_dot*phi_dot*sin(phi)*cos(phi))+sin(phi))*(term/const);
x_dot(3) = s_dot;
x_dot(4) = ((d*phi_dot*phi_dot*sin(phi))-(sin(phi)*cos(phi)))*(term/const);

if (check==1)
    u = K*x;
end
x_dot = x_dot + (B*u);

End
```

—---------------------------------------------------------------------------

```
function x_dot = lin_sys(t,x,A,B,K,u,check)
%% The file is for the linear dynamics model formed over origin

if (check==1)
    u = K*x;
end
x_dot = (A*x+(B*u));
end
```

## Question 3:

Defined the controllability matrix as shown :

```
controlability = [B,A*B, A*A*B, A*A*A*B];
```

And this matrix should have a rank = number of states i.e.rank = 4. And this has been verified using the matlab:

```
rank_controlability = rank(controlability) % Problem 3  - Confirming the controlability of the system
```

So this confirms the controllability of the system.

```
rank_controlability =

     4
```

To find the K matrix, place command is used with user defined pole positions. This gives us the required gain matrix.

```
K = -place(A,B,p); % Problem 3
```

## Code :

```
%% Problem 3: Place poles %%

controlability = [B,A*B, A*A*B, A*A*A*B];
rank_controlability = rank(controlability); % Problem 3  - Confirming the controlability of the system
K = -place(A,B,p); % Problem 3
```

## Question 4:

4)

in every case compared to prev there will be additional term due to $u$ term

$$\frac{\partial}{\partial x} f(x,u) \quad \text{let} \quad f(x,u) = g(x) + h(u)$$

$$\Rightarrow \frac{\partial}{\partial x} f(x,u) = \frac{\partial}{\partial x}(g(x)) + \frac{\partial}{\partial x} h(u)$$

$$= A + \underbrace{\left(\frac{\partial}{\partial u} h(u)\right)}_{B} \frac{\partial u}{\partial x}$$

$$= A + B\frac{\partial u}{\partial x}$$

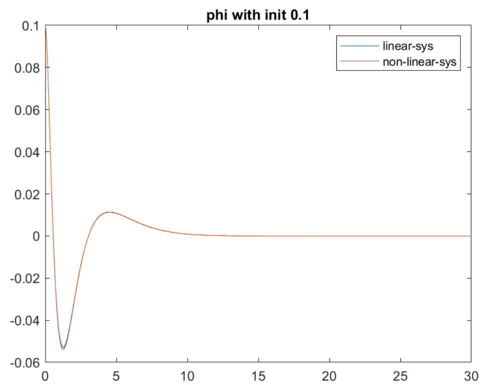
here $\frac{\partial u}{\partial x} = \frac{\partial (kx)}{\partial x} = k$

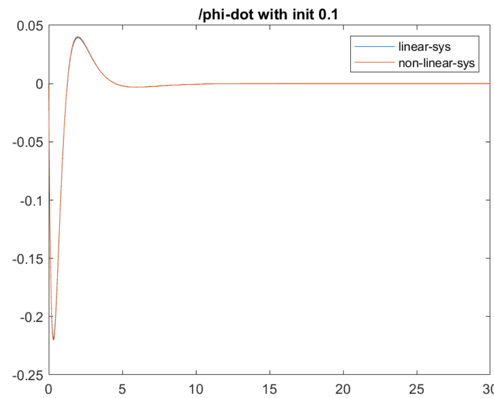
$$\Rightarrow \frac{\partial}{\partial x} f(x,u) = A + Bk \quad \text{where } u = kx$$

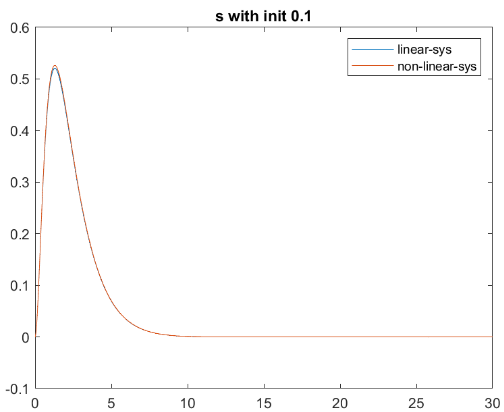
Adding the K computed in problem 3 to the system makes the real part of the eigen values are negative so according to Theorem 4.7, the system is stable around origin.

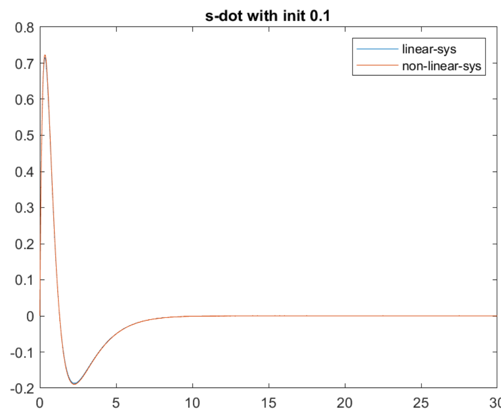
# Question 5:

## For init = 0.1:

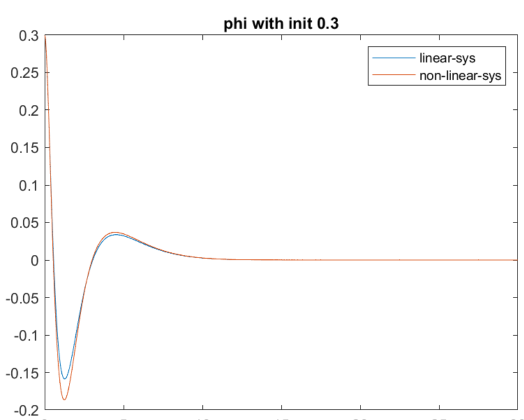


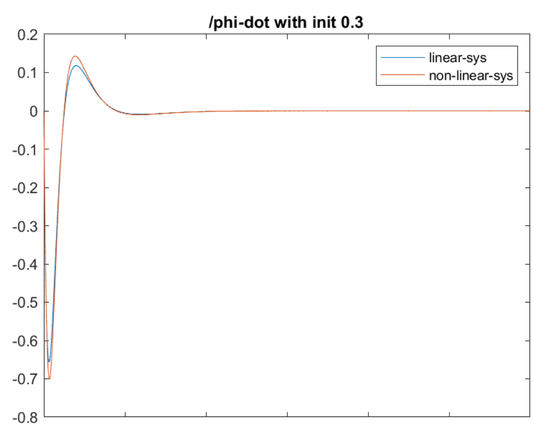



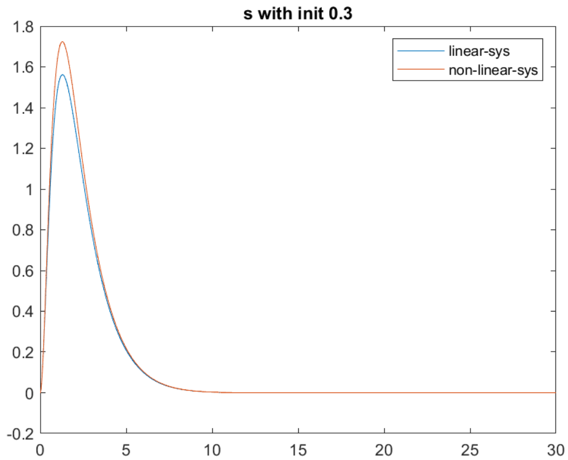



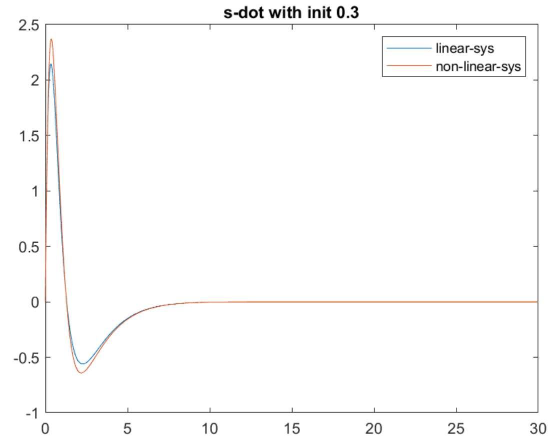




## For init = 0.3:

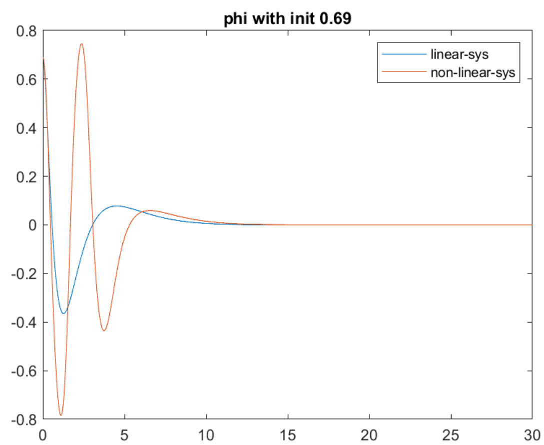


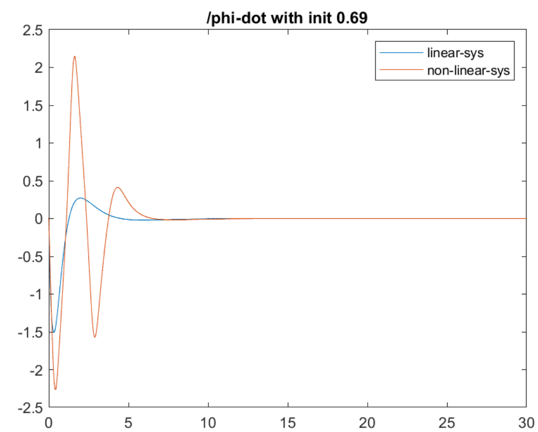



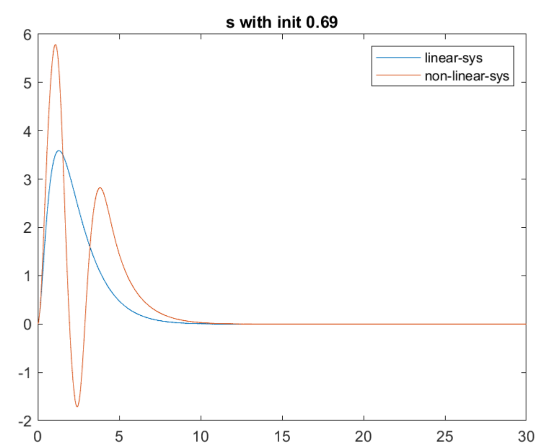

s with init 0.3
linear-sys
non-linear-sys

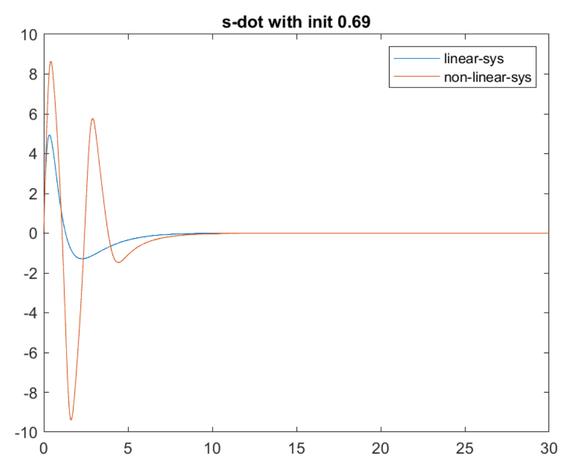

s-dot with init 0.3
linear-sys
non-linear-sys

## For init = 0.69:

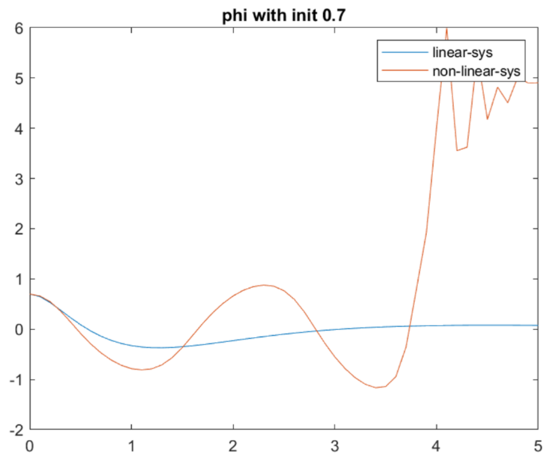

phi with init 0.69
linear-sys
non-linear-sys

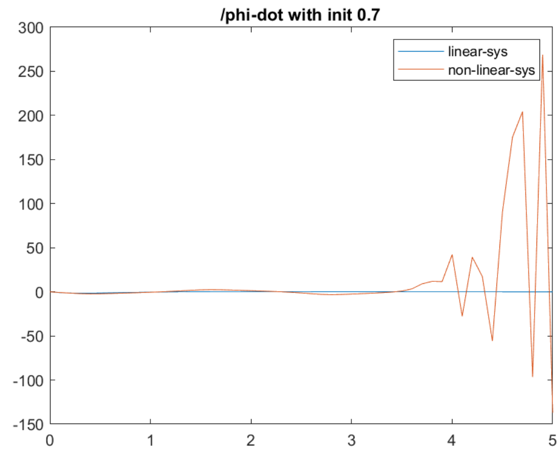

/phi-dot with init 0.69
linear-sys
non-linear-sys

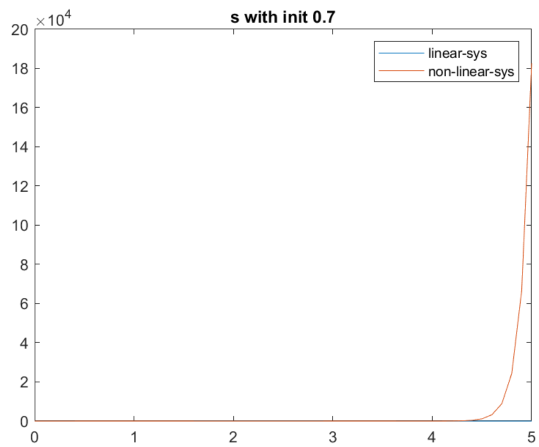

s with init 0.69
linear-sys
non-linear-sys

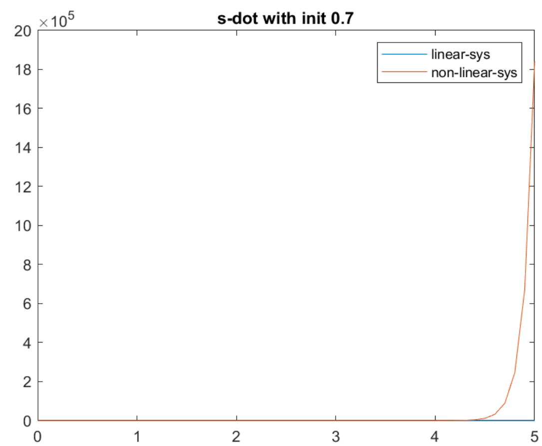

s-dot with init 0.69
linear-sys
non-linear-sys

**For init = 0.7: Unstable**




**Code:**

```
%% Problem 5: Simuation of Linear and Non-linear system with K gain %%

t = 0:0.01:30;
x_init = 0.69;
x0 = [x_init;0;0;0]; % 0.69 is the limit
[t_lin,y_lin] = ode45(@(t,x) lin_sys(t,x,A,B,K,0,1), t, x0); % output of linear system
[t_nonlin,y_nonlin] = ode45(@(t,x) nonlin_sys(t,x,b,c,d,const,K,0,1), t, x0);
% simulation for Linear vs Non-Linear
txt_1 = 'linear-sys';
txt_2 = 'non-linear-sys';
init = num2str(x_init);
saver(t_lin,y_lin,y_nonlin, txt_1, txt_2,init)
```

## Question 6:

Yes the system is observable. This is checked in matlab by checking the rank of *[C CA].* And the rank of this matrix is 4 in our case which is equal to the number of states we have. So this can conclude that the system is observable.

```
observability = [C;C*A];
rank_observability = rank(observability); % Problem 6  - Confirming the observability of the system
```

To design the observer, the eigen values took in this case are :

```
p_new = [-7,-6, -5,-4]; % Observer eigen values
L = place(A',C',p_new)'; % Problem 6  - Designing the observer
```

This is chosen because we want our estimate to converge faster than our controller. From Problem 3 we have designed the controller to converge at the desired rate. So we choose the eigen values for the observer now to make sure that the observer converges faster so that we will get better estimates. And also having very high rates of convergence can lead to larger values of  L and this can amplify the small noise in the system which is one factor we need to consider while designing these poles.

**Code:**

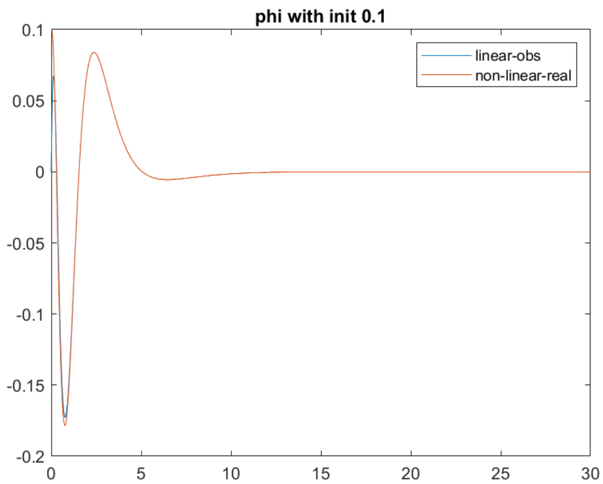
```
C = [1,0,0,0;
   0,0,1,0;]; % Observer matrix

observability = [C;C*A];
rank_observability = rank(observability); % Problem 6  - Confirming the observability of the system

p_new = [-7,-6, -5,-4]; % Observer eigen values
L = place(A',C',p_new)'; % Problem 6  - Designing the observer
```

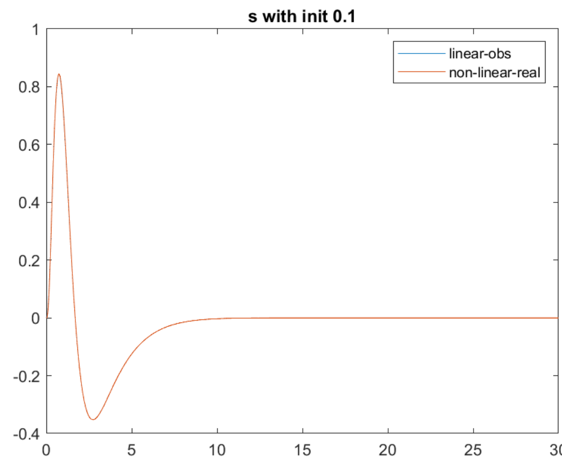
## Question 7:

The correction term helps a lot in estimating the system correctly. If the initial conditions are [x_init,0,0,0] and if the controller's initial conditions are [0,0,0,0]. There is no way we can recover the state of the system even with a small disturbance. So the correction term with the observer really helps a lot.

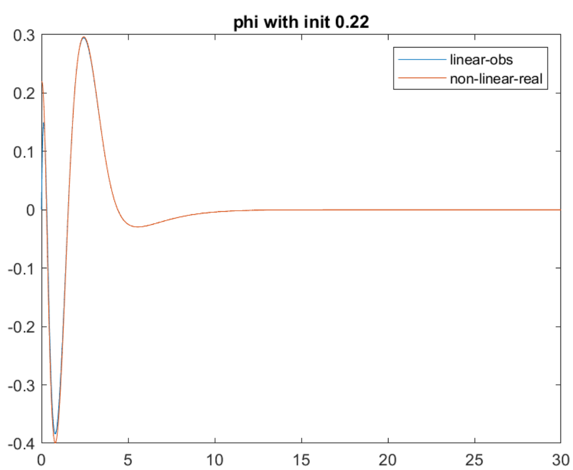
**With observer:**

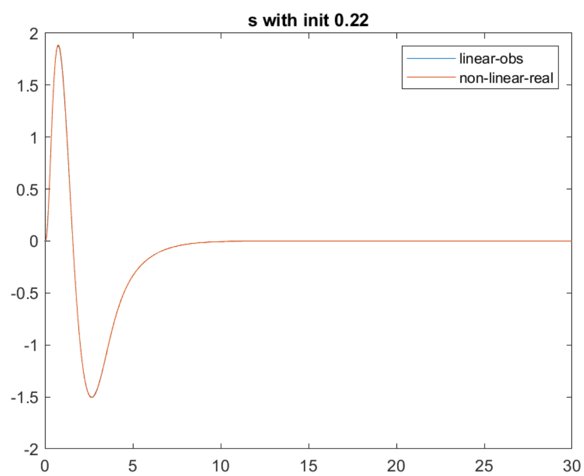
### Wrong init with X= 0.1:



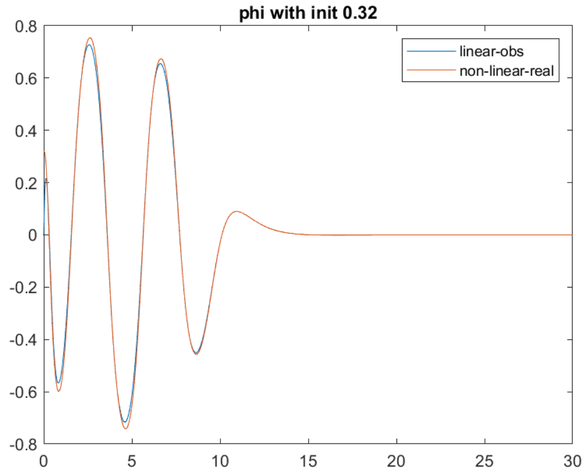



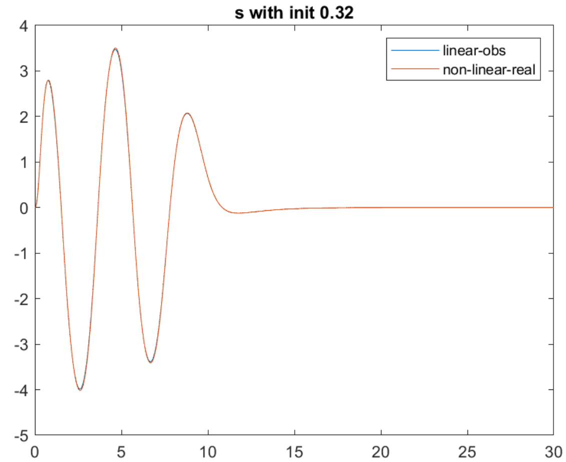

### Wrong init with X= 0.22:



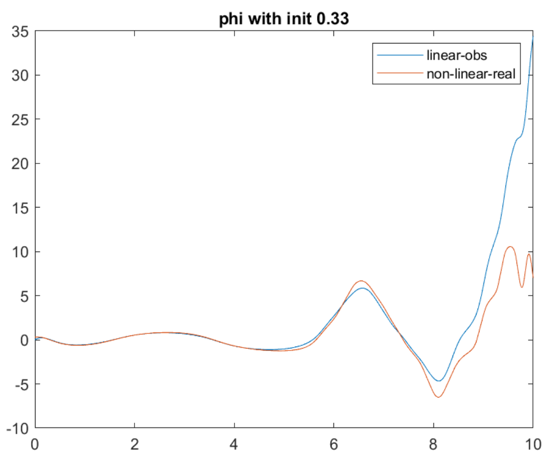



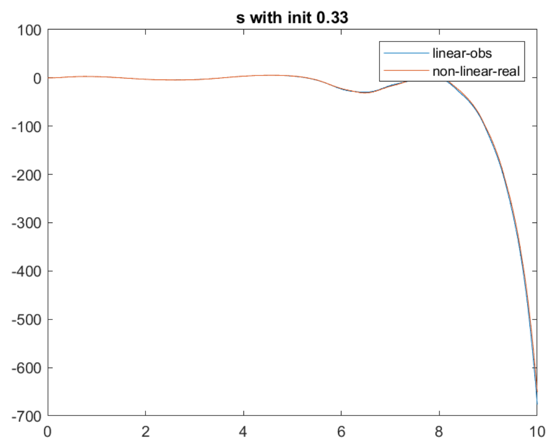
**Wrong init with X= 0.32:**



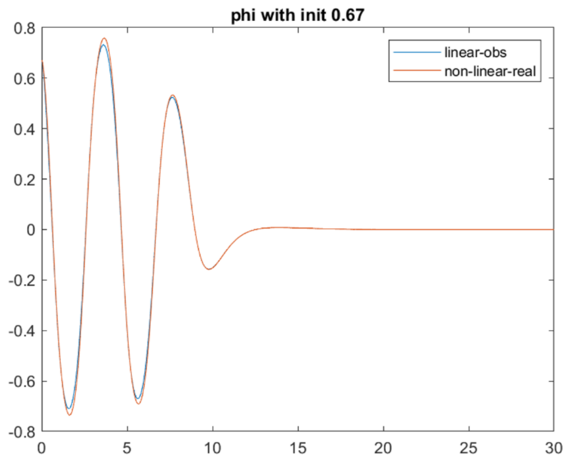



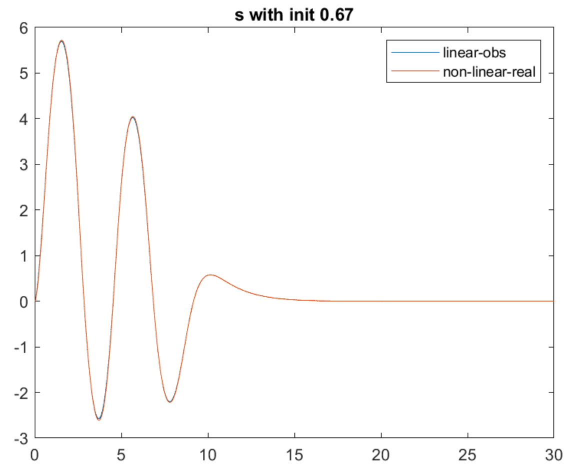

**Wrong init with X= 0.33: unstable**



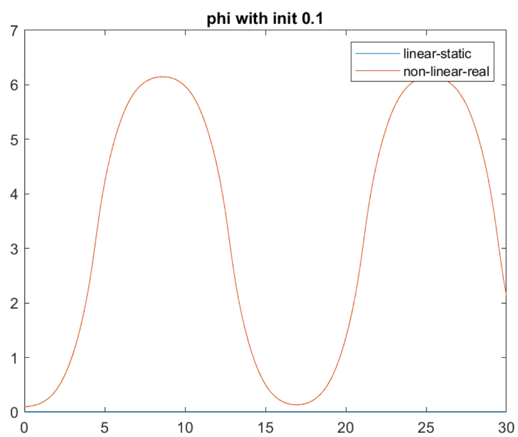



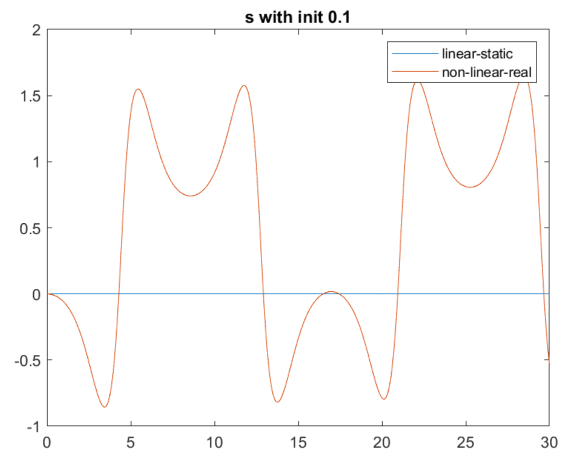
## Without observer correct initialization: unstable after 0.68



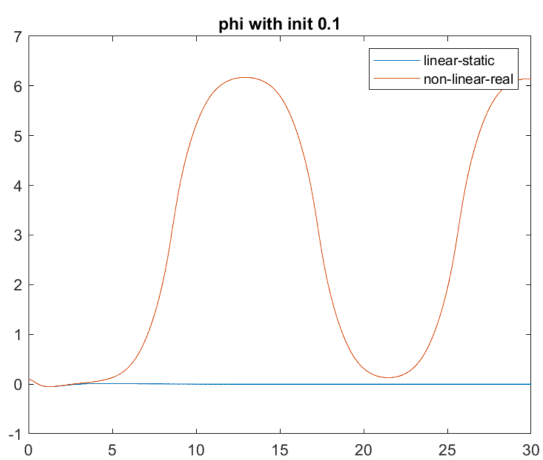



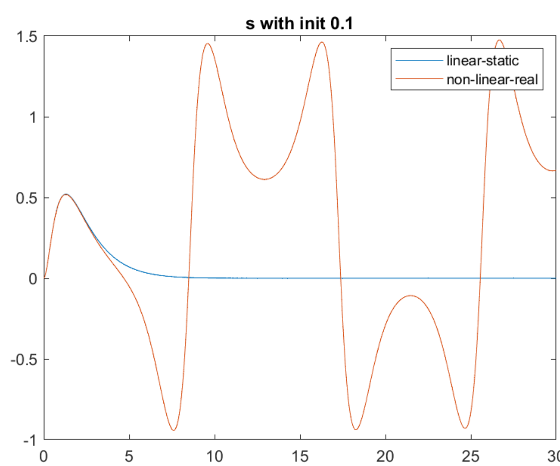

## Without observer wrong initialization: unstable with even 0.1




## Without observer correct initialization: unstable with even 0.1



**Code problem 7:**

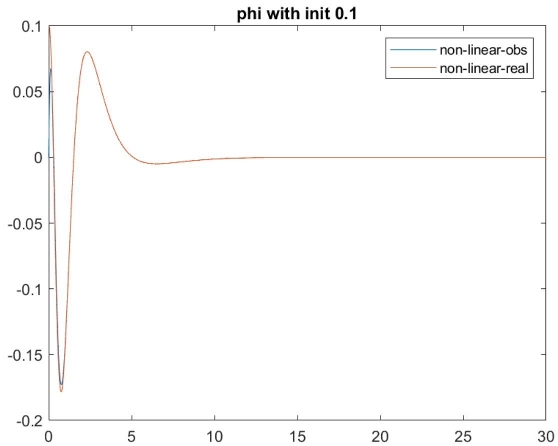
```
%% Problem 7 - observer design with linear system %%

x_init = 0.67; % 0.32 is the limit
t = 0:0.01:30;
x0 = [x_init;0;0;0;x_init;0;0;0];

% output of linear-observer and non-linear-real systems

[t_final,y_final] = ode45(@(t,x) q7(t,x,A,B,C,b,c,d,const,K,L), t, x0);
[t_final_nothing,y_final_nothing] = ode45(@(t,x) q7(t,x,A,B,C,b,c,d,const,K,L*0), t, x0);

% % Dividing the output
y_lin_hat = y_final(:,1:4);
y_nonlin_real = y_final(:,5:8);

% plotting for Linear estimate vs real
txt_1 = 'linear-obs';
txt_2 = 'non-linear-real';
init = num2str(x_init);
saver(t_final,y_lin_hat,y_nonlin_real, txt_1, txt_2,init);
%

% Dividing the output
y_lin_hat = y_final_nothing(:,1:4);
y_nonlin_real = y_final_nothing(:,5:8);

% plotting for Linear estimate vs real
txt_1 = 'linear-static';
txt_2 = 'non-linear-real';
init = num2str(x_init);
saver(t_final_nothing,y_lin_hat,y_nonlin_real, txt_1, txt_2,init);
```

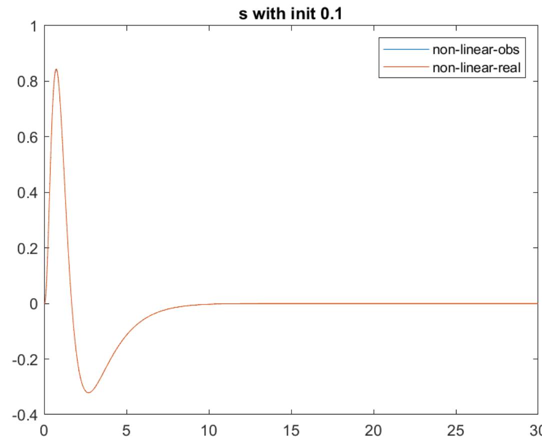
## Question 8:

The performance of non-linear observer &
linear observer are very much similar
The reason for origin to be asymptotically stable
because if the estimates converge then
the $L(y-\hat{y}) \to 0$.
& then $\dot{\hat{x}} \to f(\hat{x}, u)$
as $\hat{x} \to 0$
then $f(\hat{x}, u) \to$ ~~A~~ $(A+BK)x$
as the real eigenvalues of $A+BK$ are -ve
we can find $B_r(0)$ around origin where
this can happen
so origin is asymptotically stable.

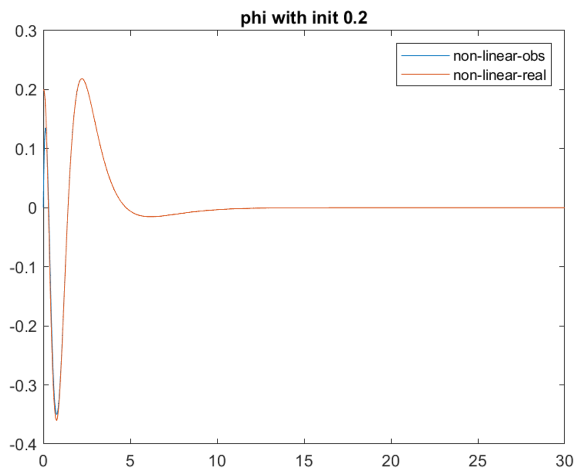
And we can see that the system is going unstable to the initial value which is similar to the linear system. So the performance of the system is similar. And we can see that even nonlinear system with some slight disturbance in the initialization can lead to drastic disaster of estimates.

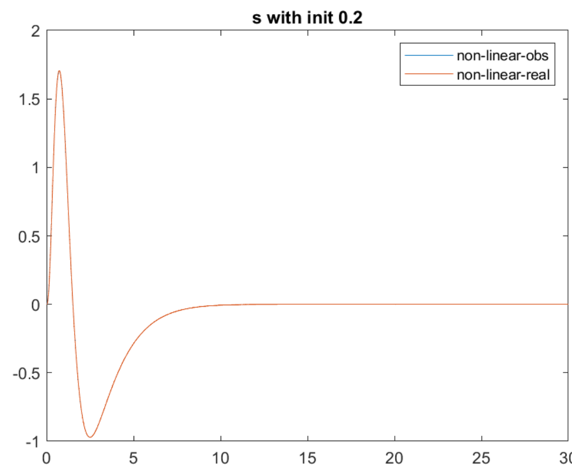
## Wrong init with X= 0.1:



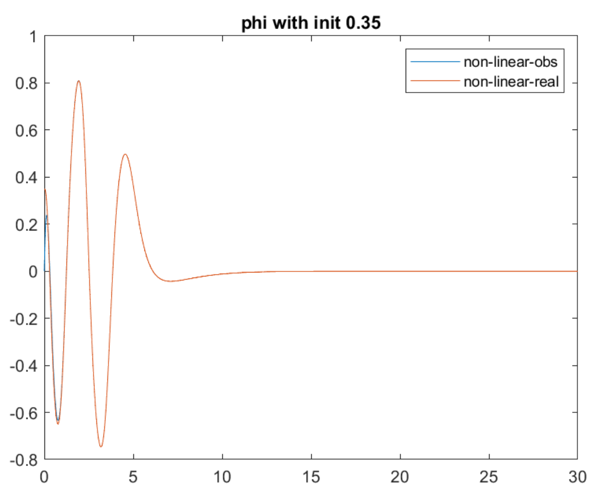



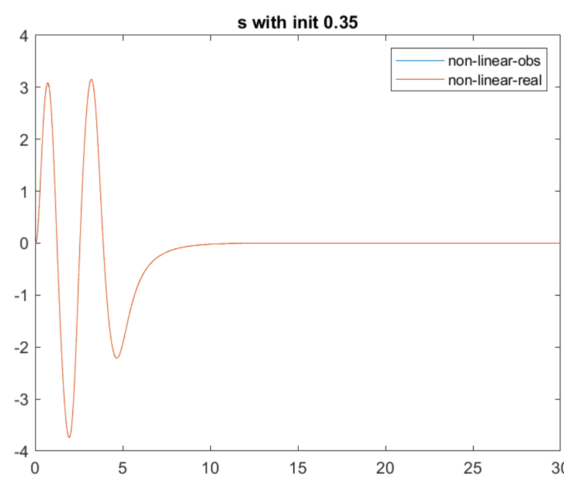

## Wrong init with X= 0.2:



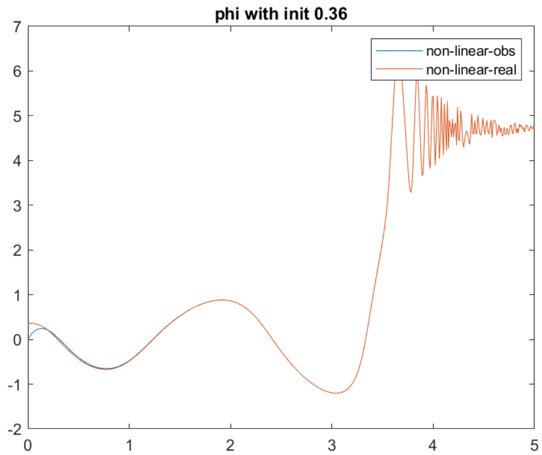



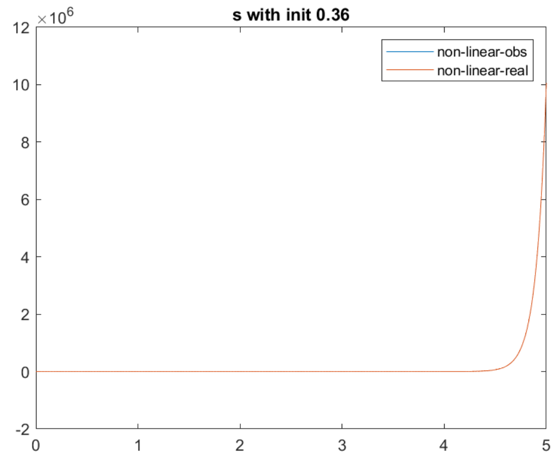

## Wrong init with X= 0.35:



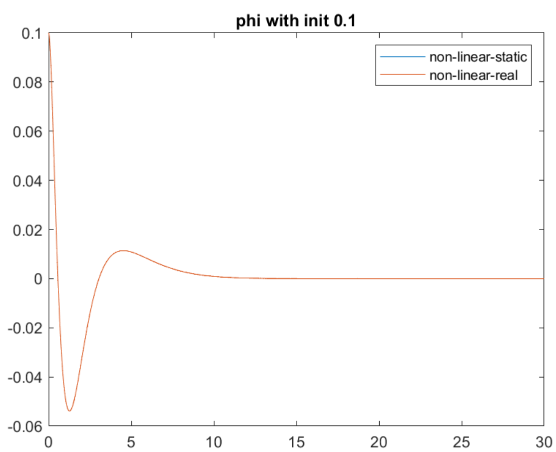



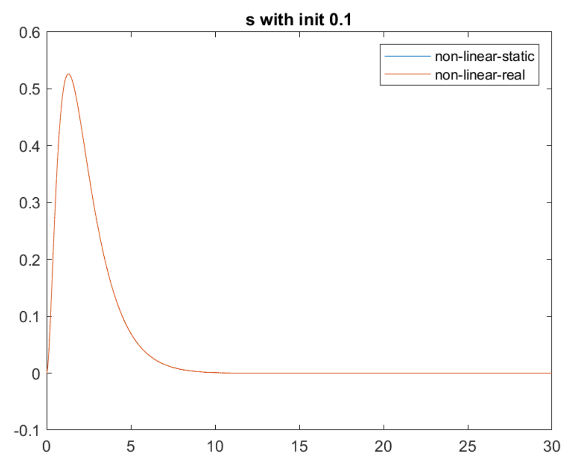
## Wrong init with X=0.36: unstable



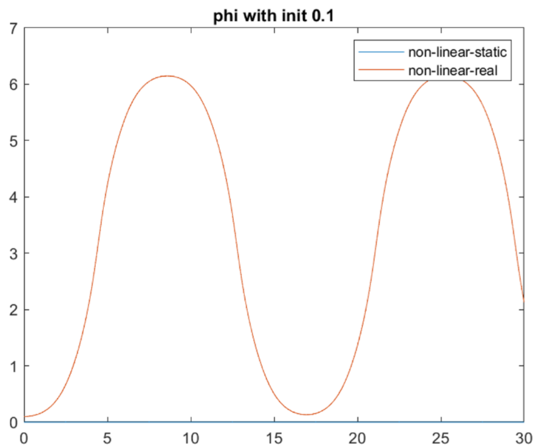



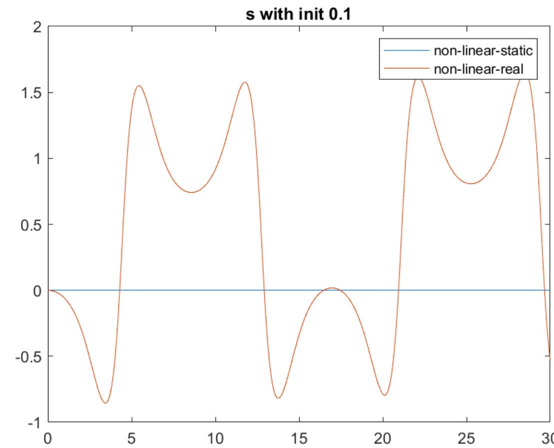

## Non - linear static : correct initialization



**Non-linear static with wrong initialization:**




**Code Problem 8**:

```
x_init = 0.1; % 0.35 is the limit
t = 0:0.01:30;
x0 = [0;0;0;0;x_init;0;0;0]; % 0.96

% output of non-linear-observer and non-linear-real systems
%
[t_final,y_final] = ode45(@(t,x) q8(t,x,A,B,C,b,c,d,const,K,L), t, x0);

y_nonlin_hat = y_final(:,1:4);
y_nonlin_real = y_final(:,5:8);

% plotting  for Non linear estimate vs real
txt_1 = 'non-linear-obs';
txt_2 = 'non-linear-real';
init = num2str(x_init);
saver(t_final,y_nonlin_hat,y_nonlin_real, txt_1, txt_2, init);

% output of non-linear-observer and non-linear-real systems

[t_final,y_final_nothing] = ode45(@(t,x) q8(t,x,A,B,C,b,c,d,const,K,L*0), t, x0);

y_nonlin_hat = y_final_nothing(:,1:4);
y_nonlin_real = y_final_nothing(:,5:8);

% plotting  for Non linear estimate vs real
```

```
txt_1 = 'non-linear-static';
txt_2 = 'non-linear-real';
init = num2str(x_init);
saver(t_final,y_nonlin_hat,y_nonlin_real, txt_1, txt_2, init);
```

## **Conclusion**:

- From the observed simulation results, we can conclude that the performance of the linear system and non linear system is very much similar, which is why the approximation of the linear system is useful.

- And we can observe the importance of correction terms in the system. Even with a perfect nonlinear system, the system can go unstable without the correct initialization. This case can occur not only with imperfect initializations, even a slight noise that is caused even in one step can lead to failure of the controller immediately. This shows the importance of *sensor-feedback* in the controllers.

- The system is performing almost the same with linear approximations formed around origin and even the case of dynamics controller the performance similar. This can help us to use the linear model of system, *for a range of inputs (*may vary for different cases *)*which can help in reducing the computation and complex terms in the modeling (may not be the case always )

## Appendix:

**Code for plots used in all cases:**

```
function saver(t_lin,y_lin,y_nonlin, txt_1, txt_2,init)

%% this file is a helper function to save the results
dir = './results/';
dir = strcat(dir,txt_1,'_',txt_2);
t_nonlin = t_lin;

% plotting and saving phi
f = figure('visible','off');
plot(t_lin,y_lin(:,1));
hold on
plot(t_nonlin,y_nonlin(:,1));
hold off
legend(txt_1, txt_2);
title(strcat("phi with init ",init));
saveas(f,strcat(dir,'phi.png'));

% plotting and saving phi_dot
f = figure('visible','off');
plot(t_lin,y_lin(:,2));
hold on
plot(t_nonlin,y_nonlin(:,2));
hold off
legend(txt_1, txt_2);
title(strcat("/phi-{dot} with init ",init));
saveas(f,strcat(dir,'phi_dot.png'));

% plotting and saving s
f = figure('visible','off');
plot(t_lin,y_lin(:,3));
hold on
plot(t_nonlin,y_nonlin(:,3));
hold off
legend(txt_1, txt_2);
title(strcat("s with init ",init));
saveas(f,strcat(dir,'s.png'));
```

```
% plotting and saving s_dot
f = figure('visible','off');
plot(t_lin,y_lin(:,4));
hold on
plot(t_nonlin,y_nonlin(:,4));
hold off
legend(txt_1, txt_2);
title(strcat("s-dot with init ",init));
saveas(f,strcat(dir,'s_dot.png'));

end
```