State Space model and finding the Open loop TF

```
%Given constants and values
M = 0.5; m = 0.23; g = 9.8; l = 0.321; a = 1/(m + M);

s = tf('s');
%State Space Model Matrices
D = 0;
C = [1 0 0 0];
A = [      0              1  0  0;
    (3*g)/(4*l-3*m*l*a)  0  0  0;
           0              0  0  1;
    -(3*m*a*g)/(4-3*m*a) 0  0  0]
```

```
A = 4×4
         0    1.0000         0         0
   29.9820         0         0         0
         0         0         0    1.0000
   -3.0323         0         0         0
```

```
B = [      0;
    (3*a)/(3*m*l*a-4*l);
           0;
    4*a/(4 - 3*m*a)]
```

```
B = 4×1
         0
   -4.1909
         0
    1.7937
```

```
states = {'phi' 'phi_dot' 'x' 'x_dot'};
inputs = {'u'};
outputs = {'phi'};

sys_ss = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
%open loop transfer function
P_pend = tf(sys_ss)
```

```
P_pend =

  From input "u" to output "phi":
    -4.191
  -----------
  s^2 - 29.98

Continuous-time transfer function.
```

```
%or can be done by the matices, the earlier is more convenient
otf = (C/(s*eye(4) - A))*B
```

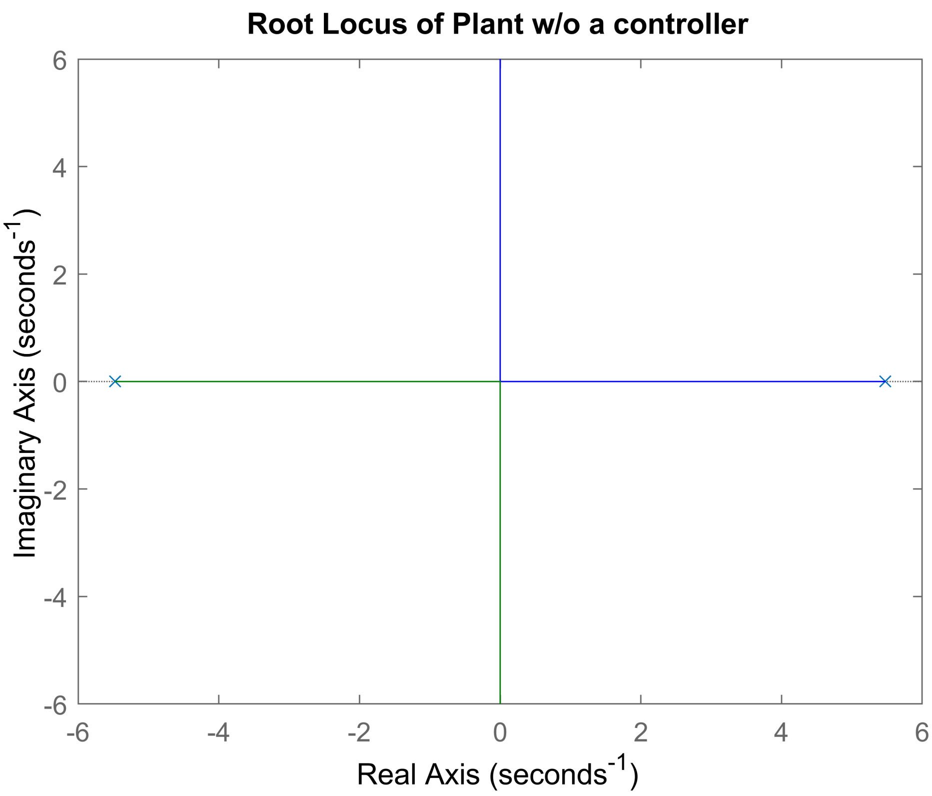
```
otf =

           -4.191
  -------------------------
  s^2 - 3.553e-15 s - 29.98
```

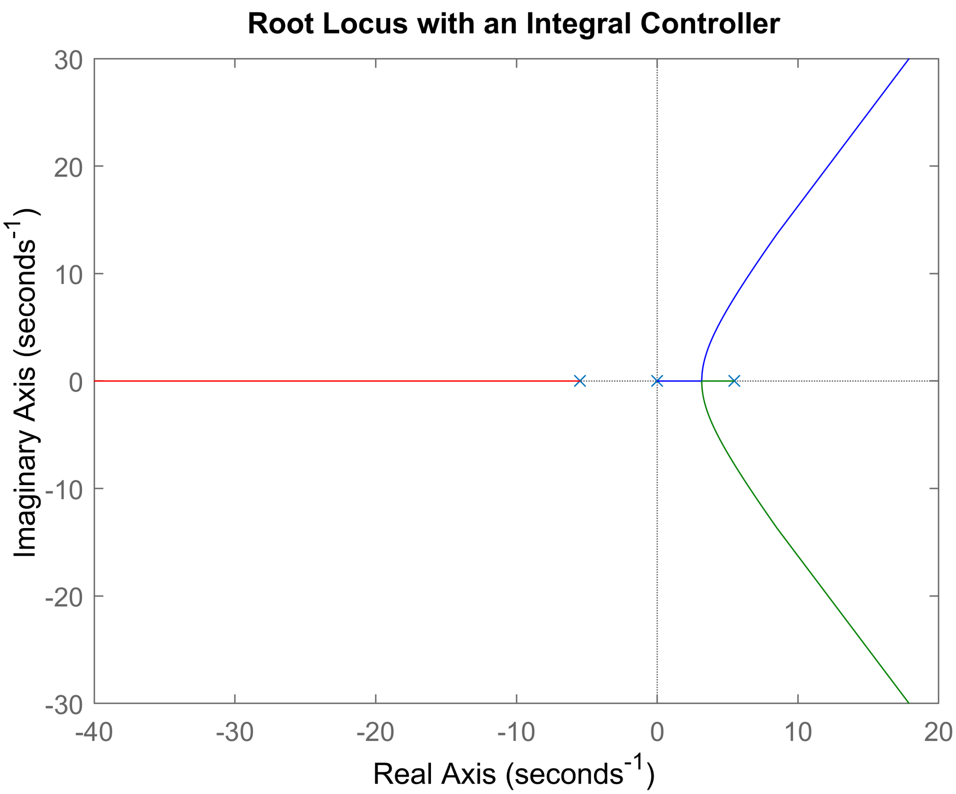
```
Continuous-time transfer function.
```

Finding the positive root locus; negative root locus doesn't provide any stability since there is no zero in RHP

```
P_pend = -P_pend;
rlocus(P_pend)
title('Root Locus of Plant w/o a controller')
```




```
%Finding the root locus for an Integral Controller.
s = tf('s');
C = 1/s;
rlocus(C*P_pend)
title('Root Locus with an Integral Controller')
```




```
zeros = zero(C*P_pend)
```

```
zeros =

  0×1 empty double column vector
```

```
poles = pole(C*P_pend)
```

```
poles = 3×1
         0
    5.4756
   -5.4756
```

As one can see that 2 branches in Integral controller are in RHP. They need to be pulled into the LHP, this can be done by adding two zeros and two poles in the LHP (i.e designing a PID controller)

Angle Criterion

Taking pole at -0.1 and zero at -3(as a lag compensator) and finding the zero to satisfy angle criterion

```
sum_a_p = angle(-4,8,5.48,0) + angle(-4,8,-5.48,0) + angle(-4,8,-0.1,0)
```

```
sum_a_p = 335.3476
```

```
sum_a_z = angle(-4,8,-3,0)
```

```
sum_a_z = 97.1250
```

```
a_new_z = -180 - sum_a_z + sum_a_p
```

```
a_new_z = 58.2226
```

```
x_z = -1*( 4 + (8/tand(a_new_z)))
```

```
x_z = -8.9558
```

Thus the new zero is at -8.9, now we will find the pole to again satisfy the angle criterion, which we desire to be at -inf, and it truly is at farther distance from origin

```
sum_a_p = angle(-4,8,5.48,0) + angle(-4,8,-5.48,0) + angle(-4,8,-0.1,0)
```

```
sum_a_p = 335.3476
```

```
sum_a_z = angle(-4,8,-3,0) + angle(-4,8,-8.9,0)
```

```
sum_a_z = 155.6375
```

```
a_new_p = 180 + sum_a_z - sum_a_p
```

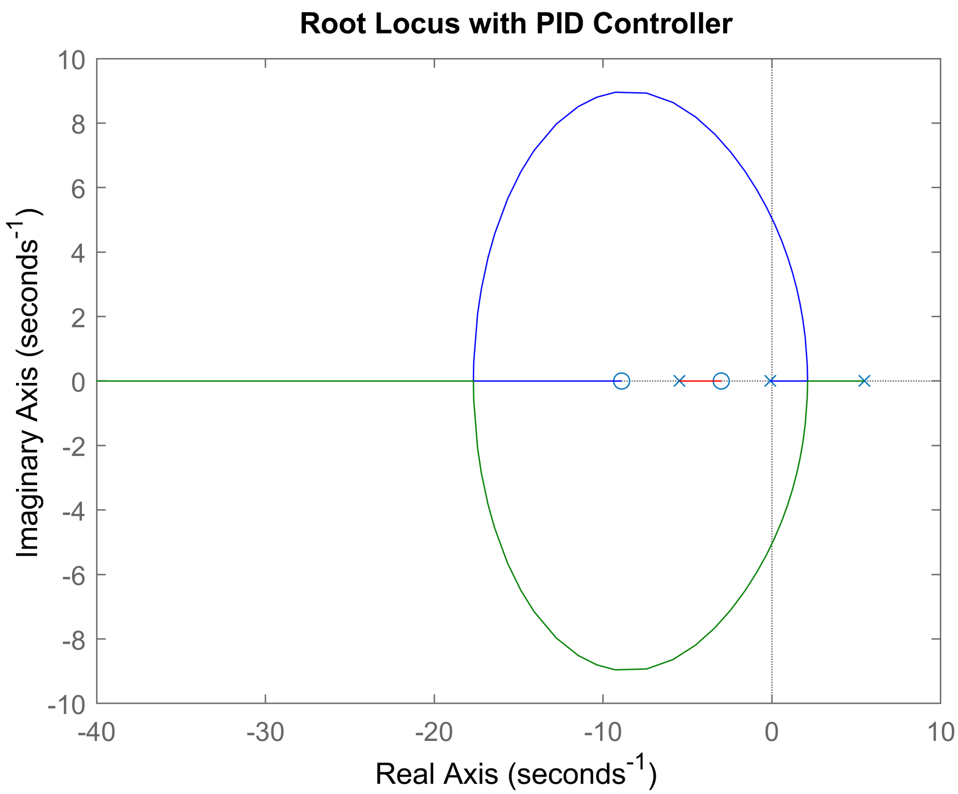
```
a_new_p = 0.2899
```

```
%x_p = -1.5849e+03, so it is farther
x_p = -1*( 4 + (8/tand(a_new_p)))
```

```
x_p = -1.5849e+03
```

Plotting the root locus with the controller having poles and zeros computed above.

```
z = [-3 -8.9];
p = -0.1;
k = 1;
C = zpk(z,p,k);
rlocus(C*P_pend)
title('Root Locus with PID Controller')
```



One can find the gain(for desired characteristics) from root locus plot by choosig a point on root locus nearer to real axis and on LHP, or theoritically by Magnitude Criterion, the value comes out be >= 11(approx.)

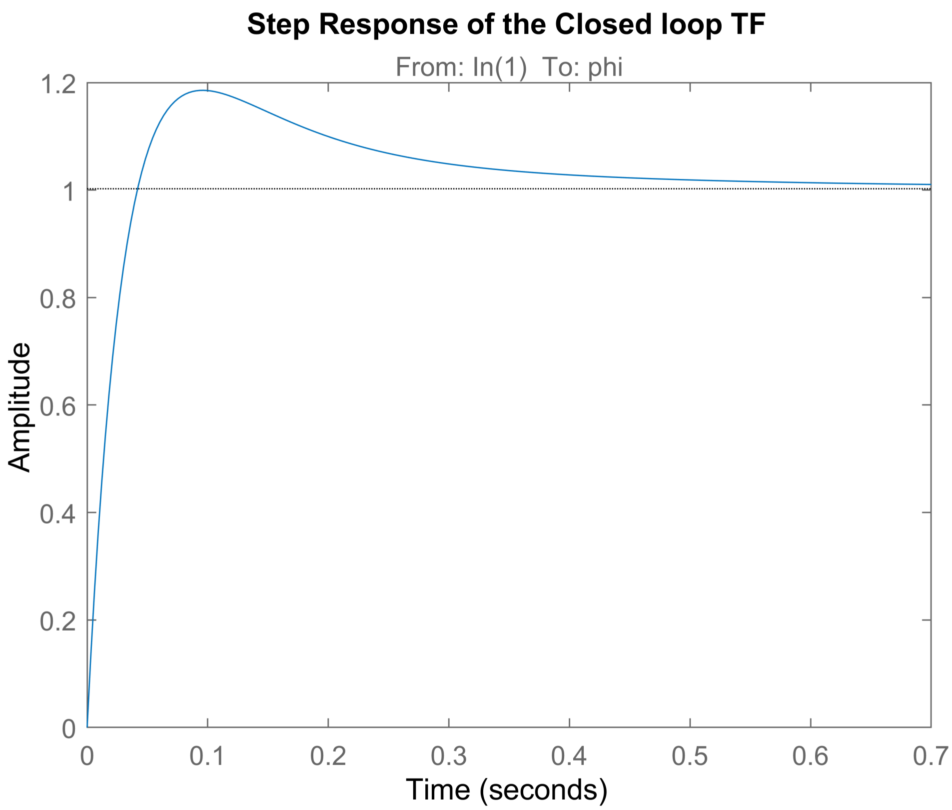
```
K = 11;
Tf = feedback(P_pend*K*C,1)
```

```
Tf =

  From input to output "phi":
       46.1 (s+8.9) (s+3)
  -----------------------------
  (s+30.53) (s+12.43) (s+3.234)

Continuous-time zero/pole/gain model.
```

```
stepplot(Tf)
title('Step Response of the Closed loop TF')
```




```
function [theta] = angle(x1,y1,x2,y2)
theta = atan2(y1 - y2, x1 - x2)*180/pi;
end
```