Frequency Domain Analysis

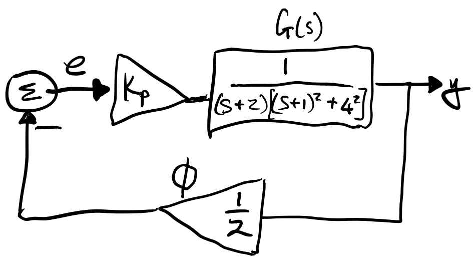
Kreitzer, M

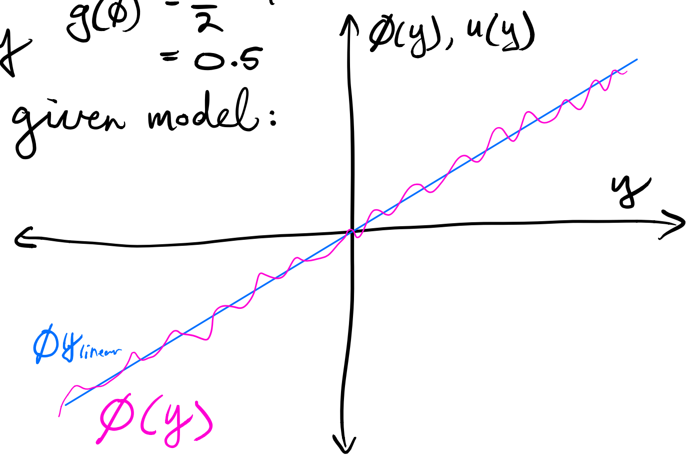
$$\phi(y) = \frac{3}{2}y - \sin(y)$$

$$\frac{\partial \phi}{\partial y} = \frac{3}{2} - \cos(y)$$

1) Linearized: $\phi y = \frac{3}{2}y$ $\quad g(\phi) = \frac{3}{2} - 1 = 0.5$

Which is stable for the given model:





$$g(sys) = \frac{k_p G_s}{1 + k_p G_s\left(\frac{1}{2}\right)}$$

$$= \frac{2k_p}{2s^3 + 8s^2 + 42s + (68 + k_p)}$$

① all signs are the same ✓✓

② Routh Array

s^3	2	42
s^2	8	$68+K_p$
s^1	$25-\frac{k_p}{4}$	0
s^0	$68+K_p$	0

$$\frac{8 \cdot 42 - 2(68+k_p)}{8}$$

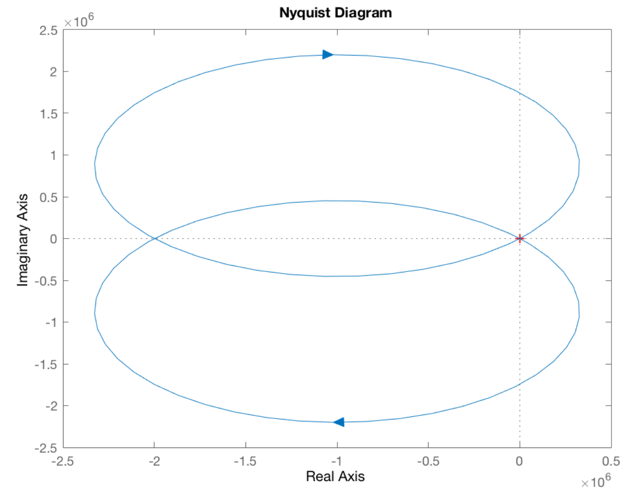
$$\frac{100 - k_p}{4}$$

$$25 - \frac{k_p}{4}$$

So for the system to be stable, $k_p < 100$

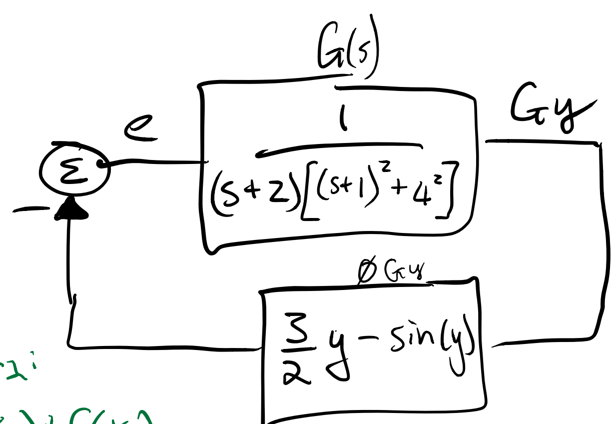
Using the following Matlab routine, I confirmed the analytical result:

```
format long;
kp_prev = 0;
for kp = linspace(99.5,101, 100000)
   G = kp * tf(1,[1 4 21 34]);
   H = tf(2*kp, [2 8 42 68+kp]);
   if ~isstable(H)
    H = tf(2*kp_prev, [2 8 42 68+kp_prev]);
    nyquist(H)
    Kp = kp_prev;
    pole(H)
    kp_prev
    break;

   end
   kp_prev = kp;
end

ans =
 -0.000000101352368 + 4.582575606488243i
 -0.000000101352368 - 4.582575606488243i
 -3.999999797295260 +
0.000000000000000i
kp_prev =
  99.999984999850000
```



Kp = 99.99 Linearized System

2) Circle Creterion



To find K_1, K_2:

$$y = f'(x_0)(x-x_0)+f(x_0)$$

$$y = \left(\tfrac{3}{2}-\cos(x_0)\right)(x-x_0)+\tfrac{3}{2}x_0-\sin(x_0)$$

$$0 = \left(\cancel{\tfrac{3}{2}}-\cos(x_0)\right)(0-x_0)+\cancel{\tfrac{3}{2}x_0}-\sin(x_0)$$

$$0 = -\cos(x_0)(-x_0)-\sin(x_0)$$

$$\sin(x_0) = \cos(x_0)\,x_0$$

$$x_0 = \tan(x_0)$$

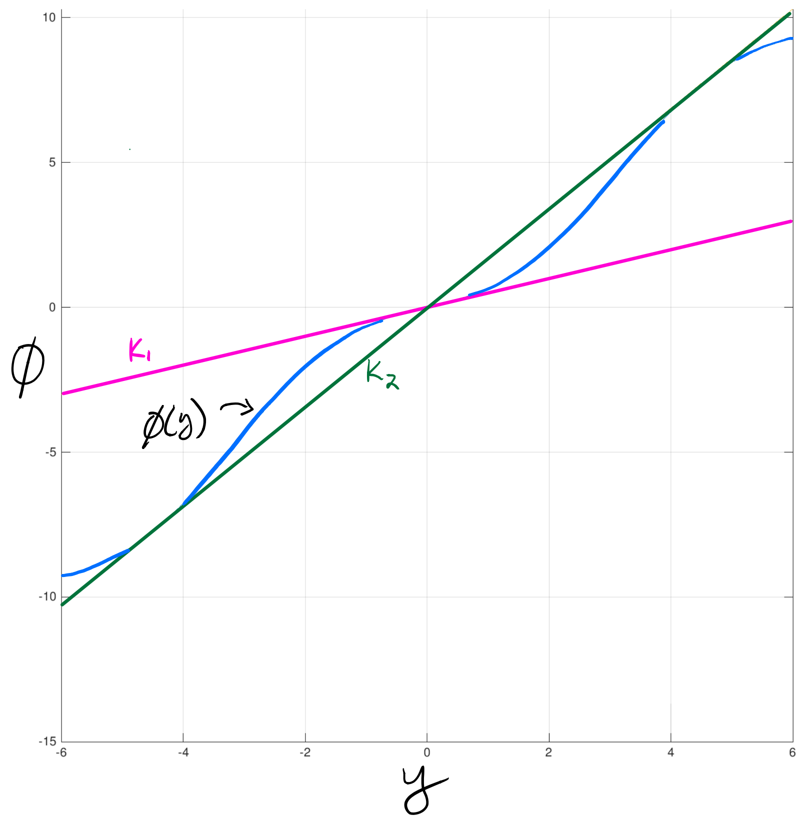
$$x_0 \approx 4.493409459$$

$$K_1 = \frac{3}{2}-\cos(0) = \frac{-1}{2}$$

$$K_2 = \frac{3}{2}-\cos(x_0) = 1.7172$$

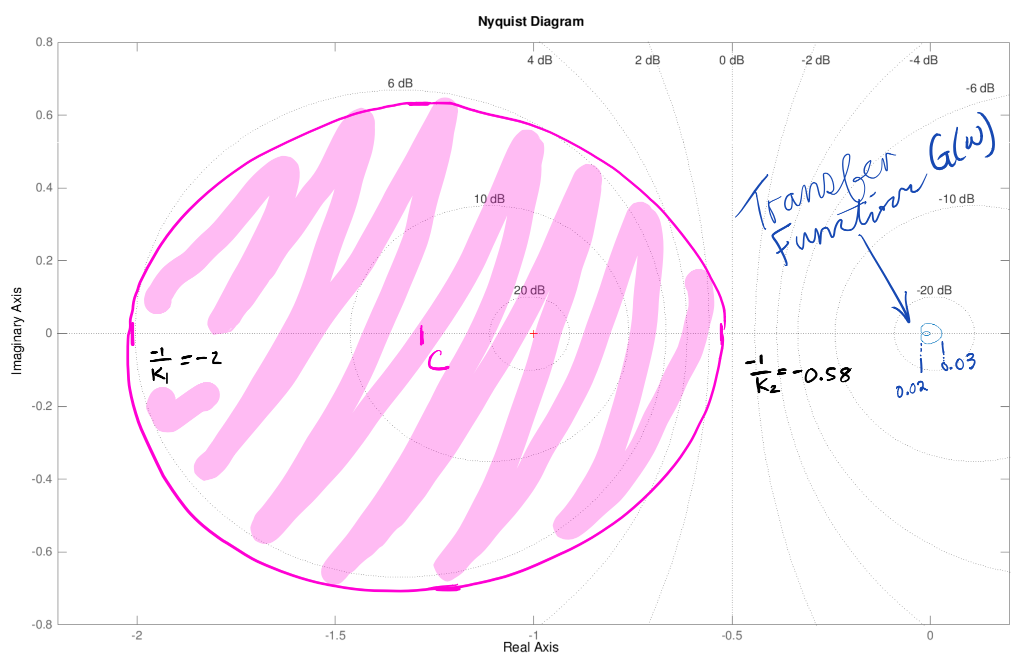
Code used to find K_2:

```
% Slopes that bound the function.
m = 3/2;
roots = fsolve(@(x) x-tan(x), [0,4]);
[k1, k2] = m - cos(roots)

y = linspace(-6, 6, 1000);
figure;
plot(y, 3*y/2-sin(y));
grid on;
hold on;
plot(y, k1*y);
plot(y, k2*y);

k1 = 0.5
k2 = 1.717233626818194
```



Code for Nyquist Diagram with circle drawn

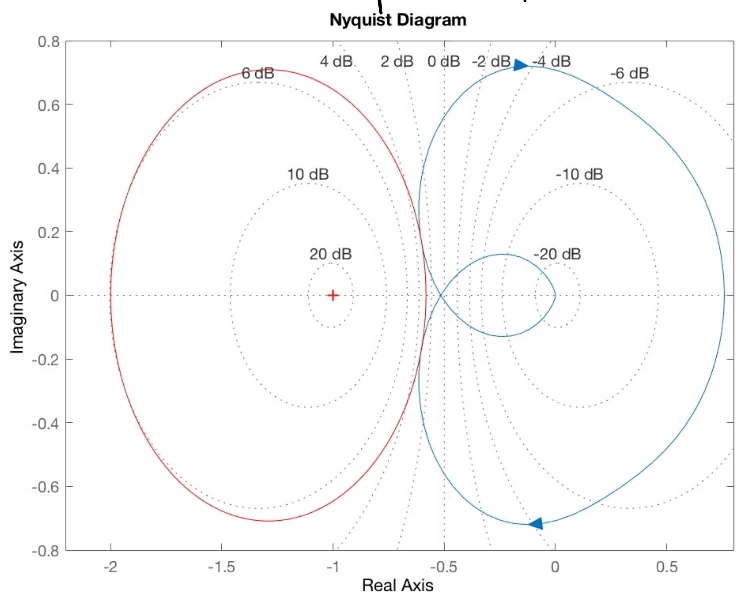
```
format long;
kp = 1;
G = kp * tf(1,[1 4 21 34]);
H = tf(2*kp, [2 8 42 68+kp]);
figure;
nyquist(G)
grid on;
axis([-2.2 0.2 -0.8 0.8])
```



3) Find $K_{p\,max}$ for NonLinear System
Using an iterative process in matlab, I found $K_p \approx 25.795$

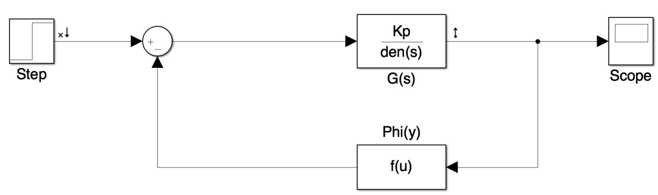
```
%% Circle Plot
figure;
grid on;
hold on;
kp = 25.795;
G = kp * tf(1,[1 4 21 34]);
nyquist(G);
axis([-2.2 0.8 -0.8 0.8]);
theta = linspace(0, 2*pi,
144);
x = radius*cos(theta);
y = radius*sin(theta);
plot(x+c, y, 'red');
grid on;
```

$K_p = 25.795$

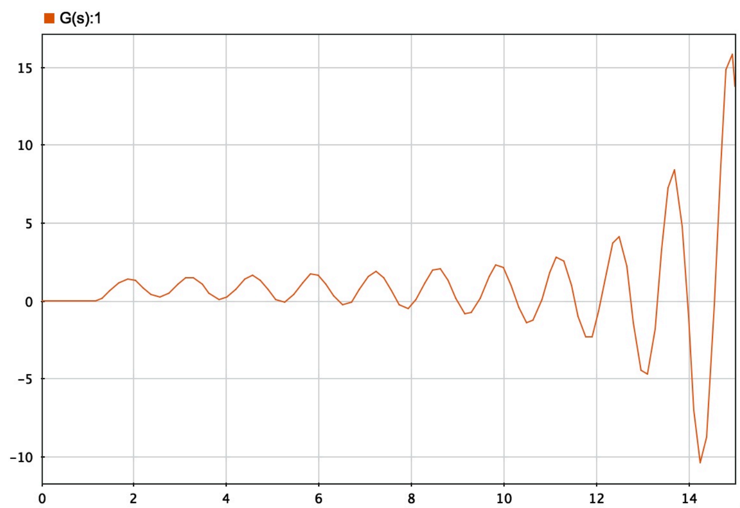


My observation in Simulink shows an oscillating system when given a unit step function. However, it eventually seems to settle.

4) When simulating the linearized model in Simulink, I used step and sinusoidal input signals



Using a unit-step input function, it could be observed for both models that if K_p was sufficiently large, the system would start oscillating and subsequently become unstable with the output exponentially growing. With sinusoidal input functions, the output $G(s)$ would act as a low pass filter. When the Kp parameter was larger than 65 or 100 for the nonlinear and linear respectively, the output would grow exponentially, as observed below:



The conclusions I have drawn from the above exercises is that Kp is affected by the nonlinearities of the system. System stability was less tolerant to a higher Kp value in the non-linear system versus the linear system. Using the oscilloscope sink in Simulink, I was also able to observe slight distortions in the output signal of the non-linear block, although it may have just been simulation error with a small step size.