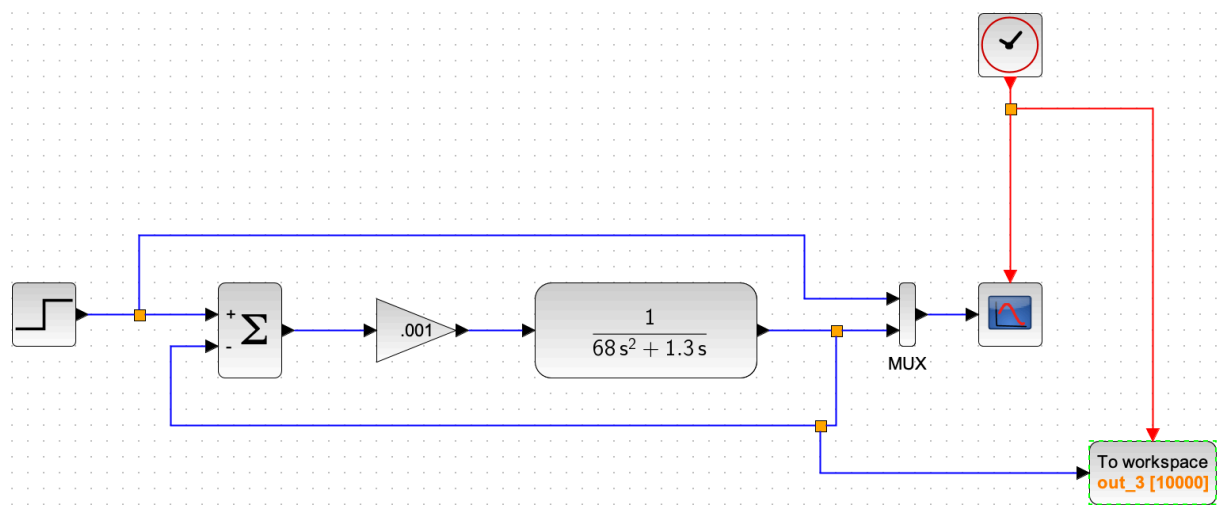
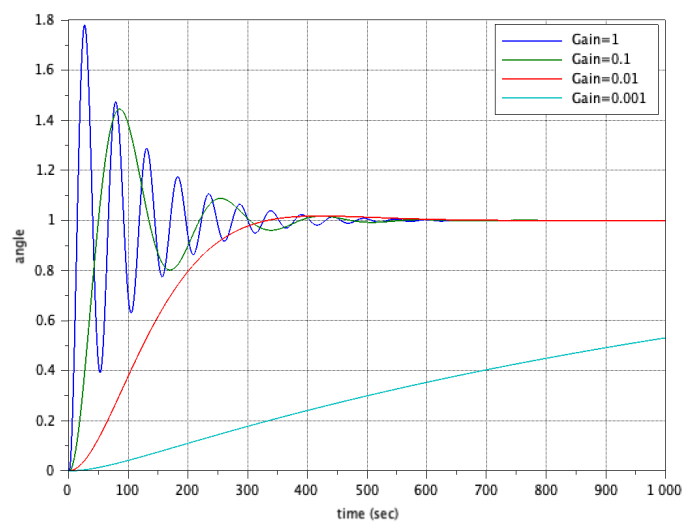


## Laporan Sementara

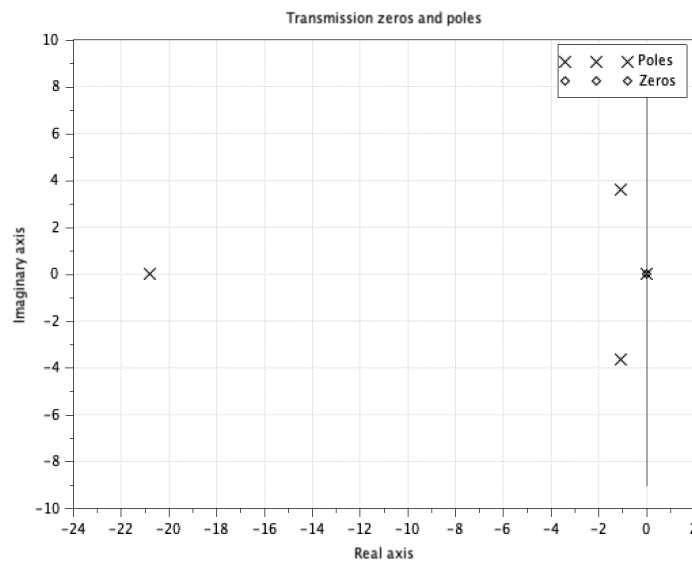
### 1. Block Diagram



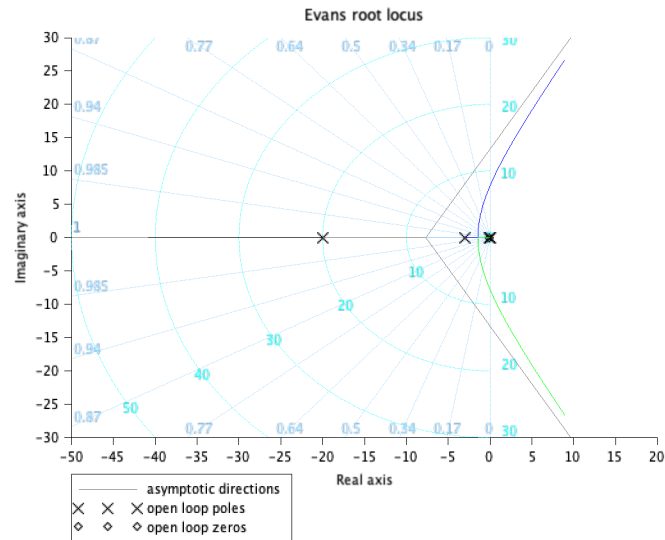
### 2. Step Response



### 3. Pole-zero Plot



#### 4. Root Locus



#### 5. Source Code

```
clc

scf(0) // figure(n)
plot(out.time, out.values, out_1.time, out_1.values, out_2.time,
out_2.values, out_3.time, out_3.values, 'LineWidth', 2);
xgrid()
xlabel('time (sec)');
ylabel('angle');
legend('Gain=1', 'Gain=0.1', 'Gain=0.01', 'Gain=0.001');

xs2png(0, 'step_response.png');

s = poly(0, 's');
C = syslin('c', 1/((68*s^2+1.3*s)*(s+3))); // Compensator
P = syslin('c', 20000 * (s+0.013)/(s+20)); // Plant

Tyr = P * C / (1 + P*C)
Tyr

scf(1)
plzr(Tyr)

xs2png(1, 'pole_zero+_plot.png');

scf(2)
L = C * P;
evans(L);
sgrid();

xs2png(2, 'rootlocus.png');

// Poles
pole = roots(Tyr.den)

// Zeros
zero = roots(Tyr.num)

[kmax, s] = kpure(L)
```

## 6. Console Log

```
--> Tyr
Tyr =

          3.8234318 +294.11765s
-----
3.8234317 +295.26469s +60.439698s2 +23.019117s3 +s4

--> pole
pole =

-20.795042 + 0.i
-1.1055459 + 3.5970851i
-1.1055459 - 3.5970851i
-0.0129835 + 0.i

--> zero
zero =

-0.0129997

--> L
L =

          260 +20000s
-----
78s +4109.9s2 +1565.3s3 +68s4

--> kmax
kmax =

4.7030123

--> s
s =

0. + 7.7550439i
```

## 7. Penjelasan

Berdasarkan plot *step response* terlihat bahwa semakin kecil *gain* yang diberikan maka akan semakin cepat juga sistem tersebut mencapai nilai *steady state*-nya.

Jika kita meninjau plot *pole-zero* dan nilai yang tercetak pada *console* maka akan terlihat bahwa seluruh *pole*-nya berada di sebelah kiri sumbu imajiner, sehingga dapat disimpulkan bahwa sistem tersebut merupakan sistem yang stabil.

Berdasarkan nilai yang tercetak pada *console* nilai maksimum sebelum sistem menjadi tidak stabil adalah  $k_{max} = 4.7030123$  dan  $s = 0 + 7.7550439i$ .