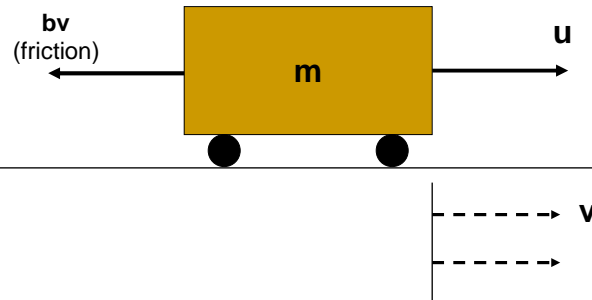

Sistem Kontrol Kapal

Pemodelan (1)

- Inersia dari roda-roda diabaikan
 - Diasumsikan friksi berlawanan dengan arah gerak kapal
 - Masalah dapat disederhanakan sebagai sistem massa (m) dan damper/peredam (bv)
-

Pemodelan (2)



Persamaan Sistem

- Dengan menggunakan hukum Newton

$$mv + bv = u$$

$$y = v$$

dimana u adalah gaya mesin

- Bila ditentukan

$$M = 1.000 \text{ kg}$$

$$B = 50 \text{ N.sec/m}$$

$$U = 500 \text{ N}$$

Desain Sistem

Kriteria desain yang diinginkan :

- Ketika mesin memberi gaya 500 Newton, kapal akan mencapai kecepatan 10 m/s
- Waktu naik < 5 detik
- Overshoot < 10%
- Kesalahan keadaan tunak < 2 %

Fungsi Transfer

- Transformasi Laplace :

$$msV(s) + bV(s) = U(s)$$

$$Y(s) = V(s)$$

- Sejak kecepatan merupakan keluaran. Substitusi $V(s)$ sebagai $Y(s)$:

$$msY(s) + bY(s) = U(s)$$

- Fungsi Transfer sistem menjadi :

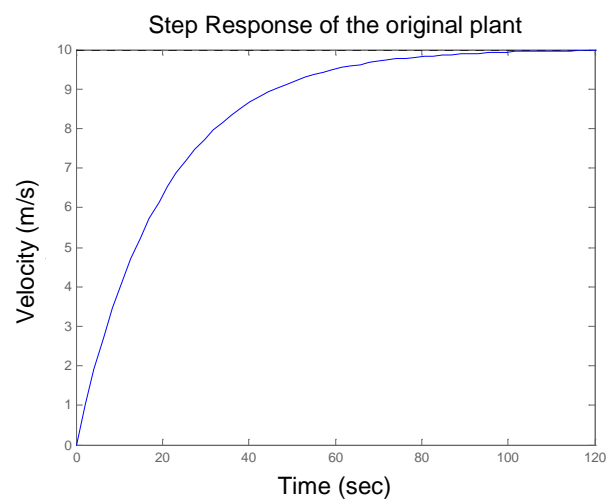
$$\frac{Y(s)}{U(s)} = \frac{1}{ms + b}$$

Matlab File Original Plant

- Buat m-file sebagai berikut :

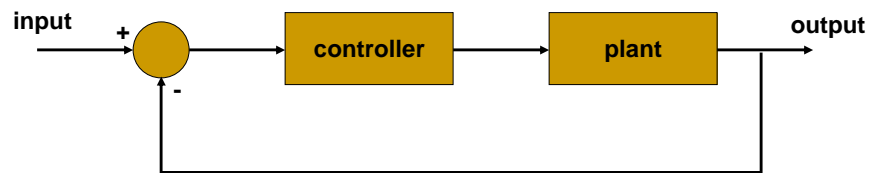
```
m=1000;  
b=50;  
u=500;  
num=[1];  
den=[m b];  
step(u*num, den)
```

Respon Loop Terbuka



Fungsi Transfer Loop Tertutup

- Menambahkan unity feedback controller



Proportional Control

- Menambahkan Proportional Control untuk mengurangi waktu naik
- Fungsi transfer loop tertutup menjadi :

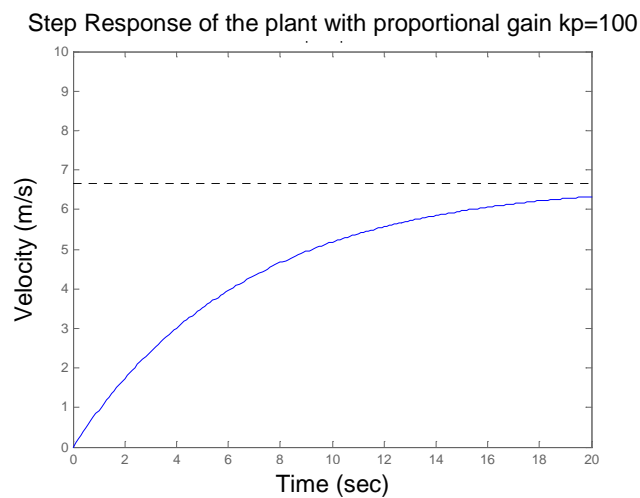
$$\frac{Y(s)}{U(s)} = \frac{K_p}{ms + (b + K_p)}$$

Matlab File using P Control

- Menentukan Kp sebesar 100 dan lihat respon yang terjadi !

```
Kp=100;  
m=1000;  
b=50;  
u=10;  
num=[Kp];  
den=[m b+Kp];  
t=0:0.1:20;  
step(u*num, den,t)  
axis([0 20 0 10])
```

Respon using P Control (1)



Respon using P Control (2)

Seperti terlihat dalam plot :

- Kesalahan keadaan tunak belum sesuai desain
- Waktu naik belum sesuai desain

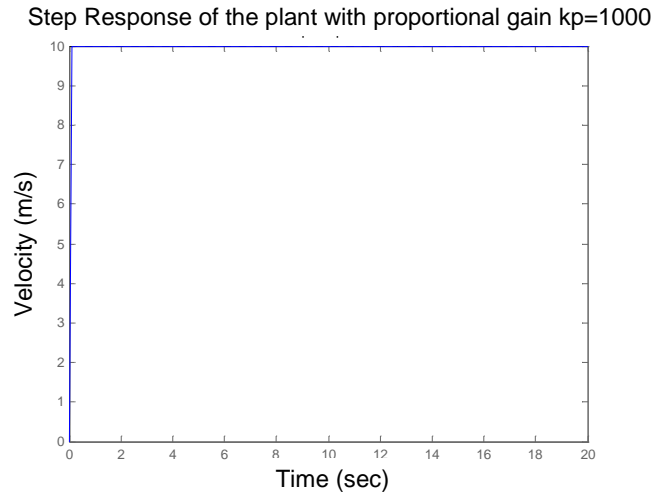
Mengubah nilai Kp menjadi 1000, dan lihat hasilnya !

Respon using P Control (3)

- Gunakan fungsi *cloop* untuk mencari respon loop tertutup langsung dari fungsi transfer loop terbukanya.

```
Kp=1000;  
m=1000;  
b=50;  
u=10;  
num=[Kp];  
den=[m b+Kp];  
[numc, denc]=cloop(Kp*num, den, -1);  
t=0:0.1:20;  
step(u*numc, denc,t)  
axis([0 20 0 10])
```

Respon using P Control (4)



Respon using P Control (5)

Seperti terlihat dalam plot :

- Kesalahan keadaan tunak turun mendekati nol
- Waktu naik kurang dari 0,5 detik

Tapi ini *impossible* membuat kecepatan kapal dari 0 ke 10 m/s kurang dari 0,5 detik

Solusinya memilih nilai K_p yang akan memberikan waktu naik yang wajar dan masuk akal dan menambah kontrol Integral untuk mengurangi kesalahan keadaan tunak

PI Control

- Menambahkan Integral Control untuk mengurangi kesalahan keadaan tunak
- Fungsi transfer loop tertutup menjadi :

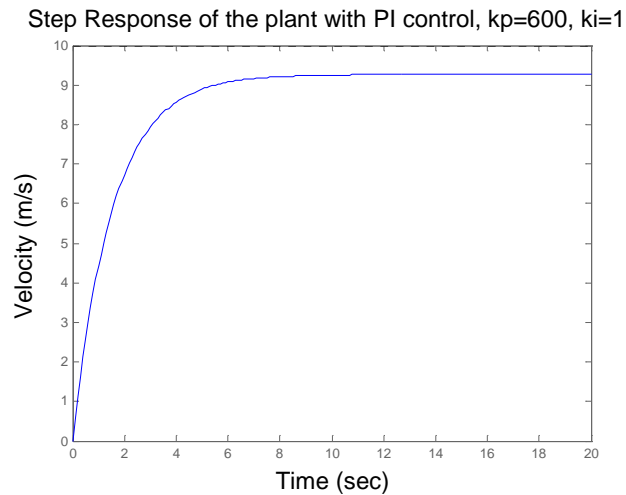
$$\frac{Y(s)}{U(s)} = \frac{Kp s + Ki}{ms^2 + (b + Kp)s + Ki}$$

Matlab File using PI Control

- Menentukan Kp sebesar 600 dan Ki = 1 dan lihat respon yang terjadi !

```
Kp=600;
Ki=1;
m=1000;
b=50;
u=10;
num=[1];
den=[m b];
num1=[Kp Ki];
den1=[1 0];
num2=conv(num, num1);
den2=conv(den, den1);
[numc, denc]=cloop(num2, den2, -1);
t=0:0.1:20;
step(u*numc, denc,t)
axis([0 20 0 10])
```

Respon using PI Control (1)



Respon using PI Control (2)

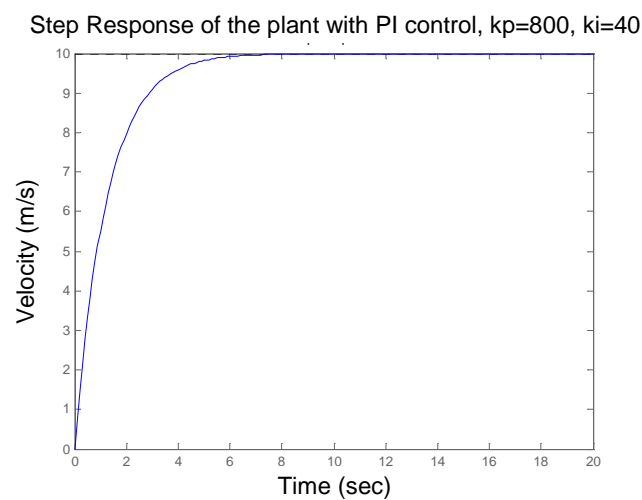
- Ubah K_p dan K_i untuk mendapatkan respon sesuai dengan desain awal.
- Ubah K_i dimulai dari nilai yang kecil, karena pengubahan nilai K_i yang besar akan membuat respon tidak stabil

Respon using PI Control (3)

- Ubah Kp menjadi 800 dan Ki = 40 dan lihat respon yang terjadi !

```
Kp=800;  
Ki=40;  
m=1000;  
b=50;  
u=10;  
num=[1];  
den=[m b];  
num1=[Kp Ki];  
den1=[1 0];  
num2=conv(num, num1);  
den2=conv(den, den1);  
[numc, denc]=cloop(num2, den2, -1);  
t=0:0.1:20;  
step(u*numc, denc,t)  
axis([0 20 0 10])
```

Respon using PI Control (4)



Respon using PI Control (5)

Seperti terlihat pada plot, respon sistem sudah sesuai dengan kriteria desain kita, dimana :

- Waktu naik < 5 detik
- Overshoot < 10%
- Kesalahan keadaan tunak < 2 %

Dalam kasus ini, tidak diperlukan lagi adanya kontrol derivatif untuk mendapatkan keluaran yang diinginkan.

Karakteristik PID Controller (1)

Proportional Control :

- Mengurangi waktu naik
- Tidak menghapus kesalahan keadaan tunak

Integral Control :

- Menghapus kesalahan keadaan tunak, tetapi respon transient memburuk

Derivatif Control :

- Meningkatkan stabilitas sistem
- Mengurangi overshoot
- Menaikkan respon transfer

Karakteristik PID Controller (2)

Respon Loop Tertutup	Waktu Naik	Overshoot	Waktu turun	Kesalahan Keadaan Tunak
Kp	Menurun	Meningkat	Perubahan kecil	Menurun
Ki	Menurun	Meningkat	Meningkat	Hilang
Kd	Perubahan kecil	Menurun	Menurun	Perubahan kecil