

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang pengujian alat yang telah dirancang dari percobaan yang dilakukan sebelumnya, untuk menunjang pengujian secara keseluruhan sistem. Pengujian dilakukan untuk mengetahui karakteristik dan letak kesalahan serta mempermudah analisis pada sistem. Pengujian dibagi menjadi beberapa bagian, yaitu:

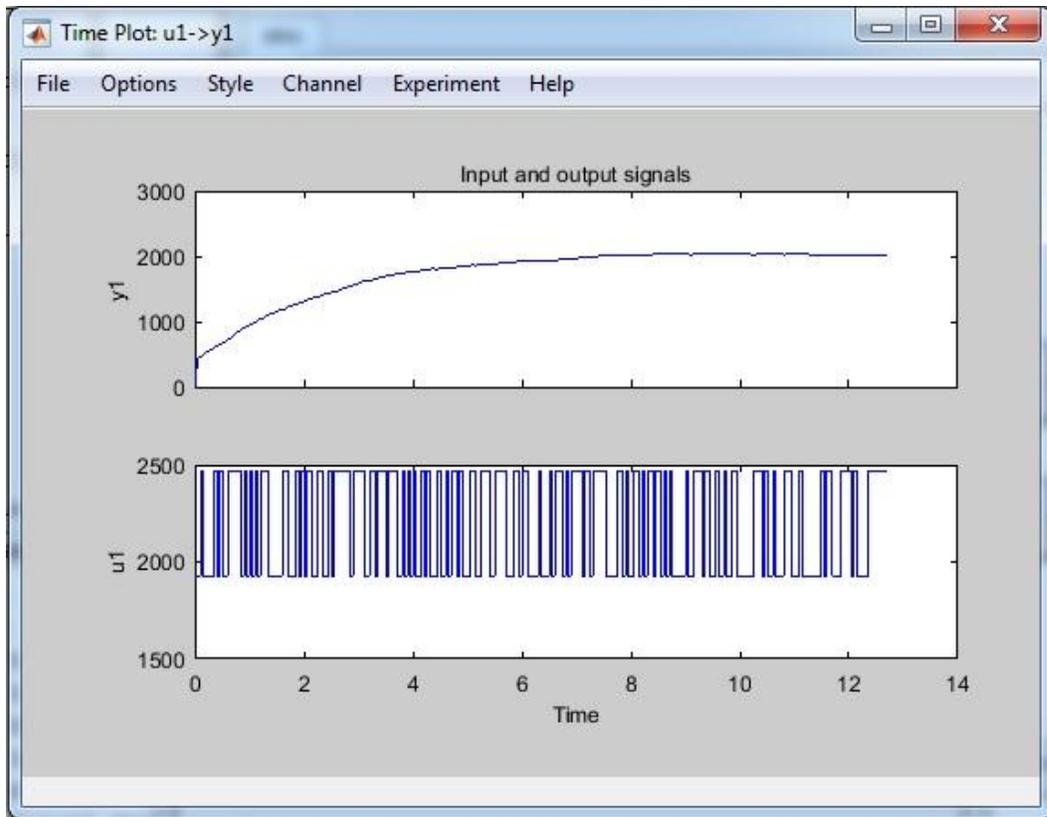
1. Penentuan fungsi alih motor DC RS-555
2. Penentuan parameter kontrol menggunakan MATLAB R2016a.
3. Pengujian sistem secara keseluruhan.

4.1 Penentuan Fungsi Alih Motor DC RS-555

Untuk mengendalikan motor DC RS-555 digunakan Arduino Mega 2560 sebagai pengolah dan memberikan data berupa *Pulse width Modulation* (PWM) agar motor bergerak. Motor DC yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan *rotary encoder* FC-03. Untuk mendapatkan karakteristik motor DC pada perancangan ini diberikan masukan unit step.

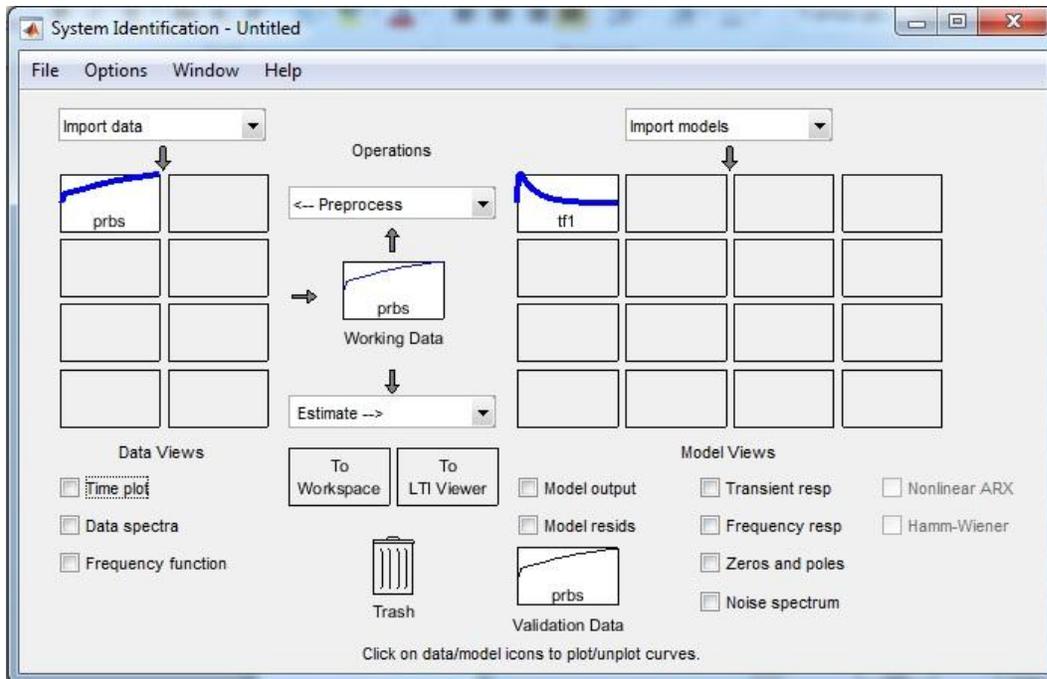
Untuk mendapatkan fungsi alih dari motor dilakukan pemodelan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS). Langkah yang dilakukan untuk membangkitkan sinyal PRBS adalah sebagai berikut :

1. Mencari nilai yang linier dari hasil kecepatan motor terhadap *duty cycle* PWM.
2. Memasukkan nilai batas atas dan bawah berdasarkan nilai yang linier untuk membangkitkan sinyal PRBS.
3. Nilai batas atas dan batas bawah yang digunakan dalam pencarian fungsi alih yang digunakan pada penelitian ini adalah *duty cycle* 80% dan *duty cycle* 60%.
4. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai masukan motor DC.
5. Setelah didapatkan data sinyal PRBS dan data kecepatan motor DC RS-555 (Gambar 4.1), selanjutnya adalah melakukan identifikasi dengan menggunakan *software* MATLAB.

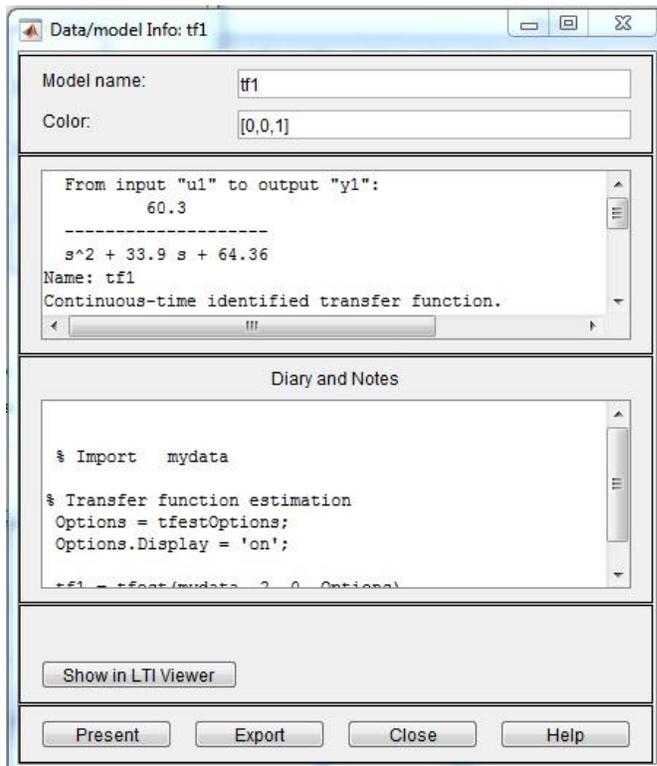


Gambar 4.1 Respon sinyal PRBS dan kecepatan motor DC RS-555

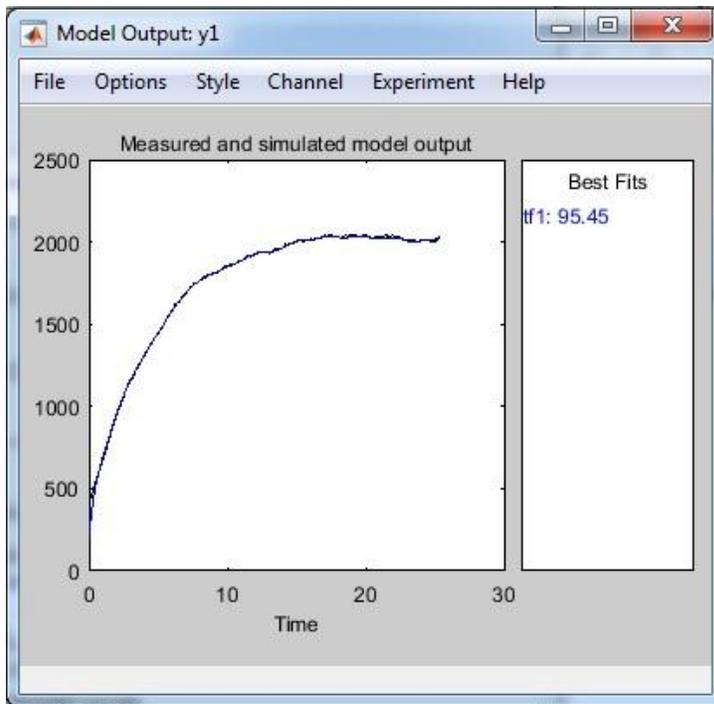
6. Dengan menggunakan sintaks *ident* pada *command window* pada MATLAB, data sinyal PRBS dan data kecepatan motor yang telah disimpan kemudian di *import* pada blok *System Identification Toolbox* (lihat Gambar 4.2). Setelah melakukan beberapa estimasi model berdasarkan data yang telah di import didapatkan fungsi alih dari motor (lihat Gambar 4.3). dengan best fit sebesar 95,45 (lihat Gambar 4.4).



Gambar 4.2 System Identification Toolbox



Gambar 4.3 fungsi alih motor DC RS-555



Gambar 4.4 Hasil estimasi model

7. Dari hasil identifikasi, fungsi alih motor yang didapat adalah :

$$\frac{Y(s)}{U(s)} = \frac{60,3}{s^2 + 33,9s + 64,36}$$

$$F(s) = \frac{60,3}{s^2 + 33,9s + 64,36}$$

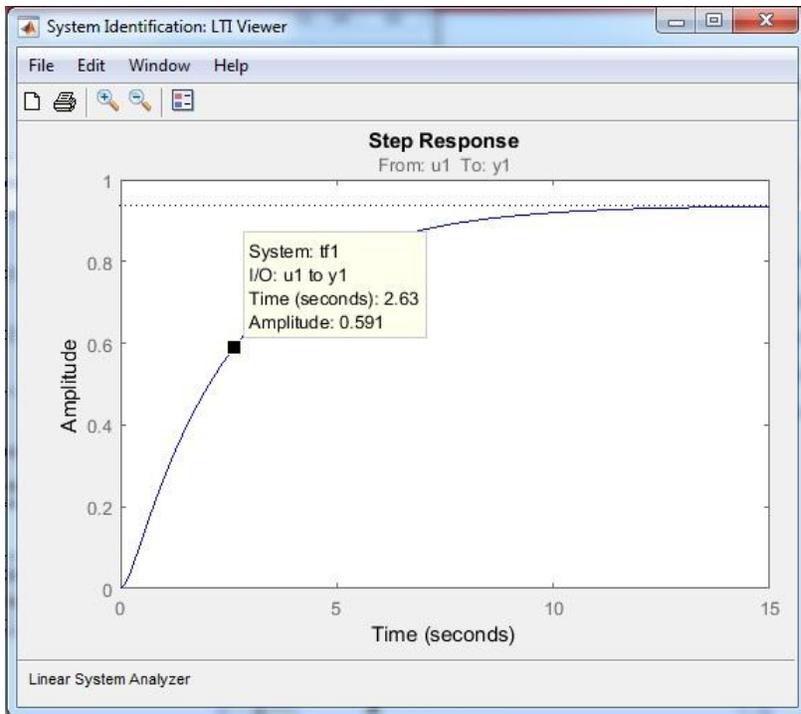
$$= \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$$2\xi\omega_n = 33,9$$

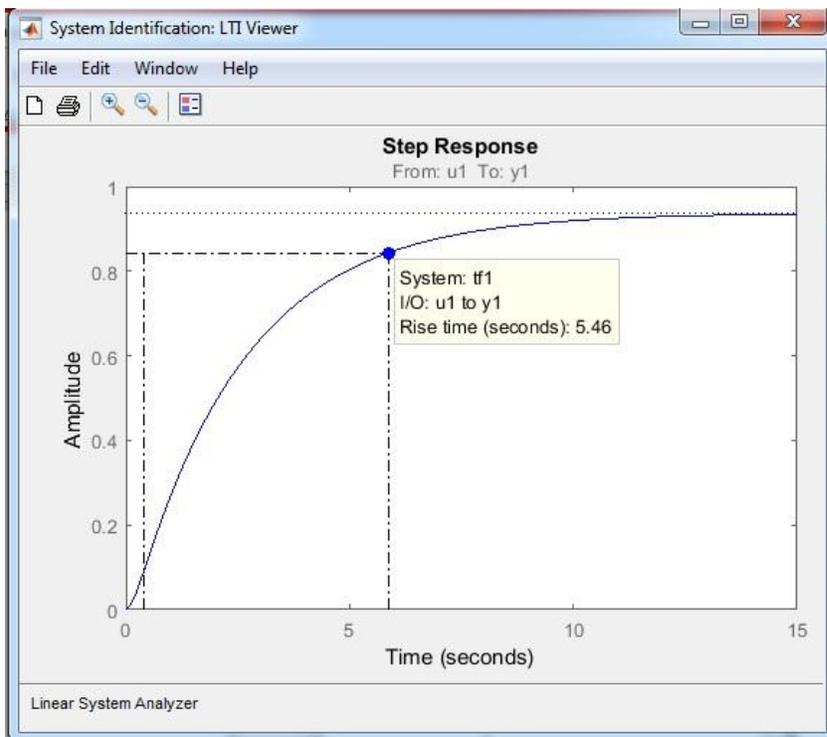
$$\omega_n = \sqrt{64,36} = 8,022$$

$$\xi = \frac{33,9}{2 \times 8,022} = 2,11$$

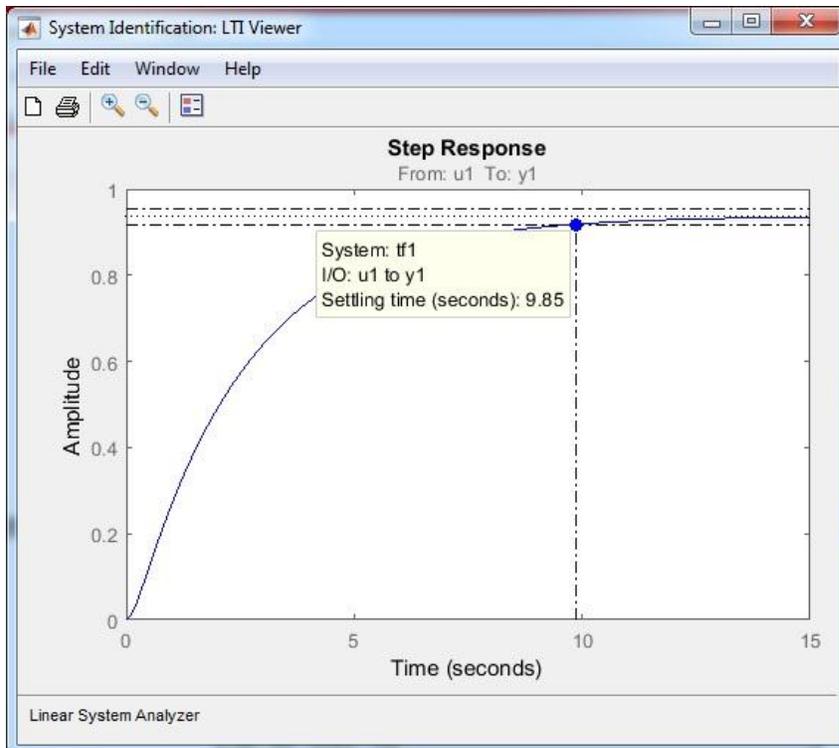
Dengan memberikan masukan *unit step* pada program MATLAB didapatkan nilai *time constant* fungsi alih merupakan waktu yang dibutuhkan respon untuk mencapai 63,2% dari nilai *steady state* yaitu 2.63 seconds (Gambar 4.5). Didapatkan pula nilai *rise time* adalah 5.46 seconds (Gambar 4.6) dan *settling time* adalah 9.85 seconds (Gambar 4.7).



Gambar 4.5 Nilai *time constant* output motor DC RS-555 dengan input *unit step*



Gambar 4.6 Nilai *rise time* output motor DC RS-555 dengan input *unit step*



Gambar 4.7 Nilai *settling time* output motor DC RS-555 dengan input *unit step*

4.2 Penentuan Parameter Kontroler PID dengan Metode *Root Locus*

Metode *root locus*/letak kedudukan akar digunakan untuk meneliti perilaku sistem berubah pada lingkup tertentu. Rancangan dimaksudkan agar letak *pole* dan *zero* dari fungsi alih loop tertutup terletak pada daerah yang ditentukan. Agar sistem stabil, *pole* dan *zero* harus terletak pada bidang s sebelah kiri sumbu imajiner. Untuk memenuhi tujuan performansi loop yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang dipilih ialah kontroler Proporsional Integral Differensial (PID). Langkah yang dilakukan untuk mendapatkan nilai parameter K_p , K_i , dan K_d adalah sebagai berikut :

1. Mencari fungsi alih motor DC tanpa kontroler terlebih dahulu dengan membangkitkan sinyal *pseudo random binary sequence* (PRBS).
2. Setelah didapatkan fungsi alih, maka diperiksa kestabilannya dengan *root locus* yang ada di MATLAB. Fungsi alih yang stabil adalah fungsi alih yang nilai akar s nya sudah berada di sumbu kiri imajiner.
3. Menentukan letak pole s_1 sesuai spesifikasi desain pada penelitian ini.

4. Mencari nilai parameter K_p , K_i , dan K_d di MATLAB dengan mensubstitusi nilai s_1 dan nilai fungsi alih sistem dalam Persamaan 2-18 dan memvariasikan nilai K_i .

Setelah didapatkan fungsi alih sistem yaitu $F(s) = \frac{60,3}{s^2 + 33,9s + 64,36}$. Selanjutnya menentukan

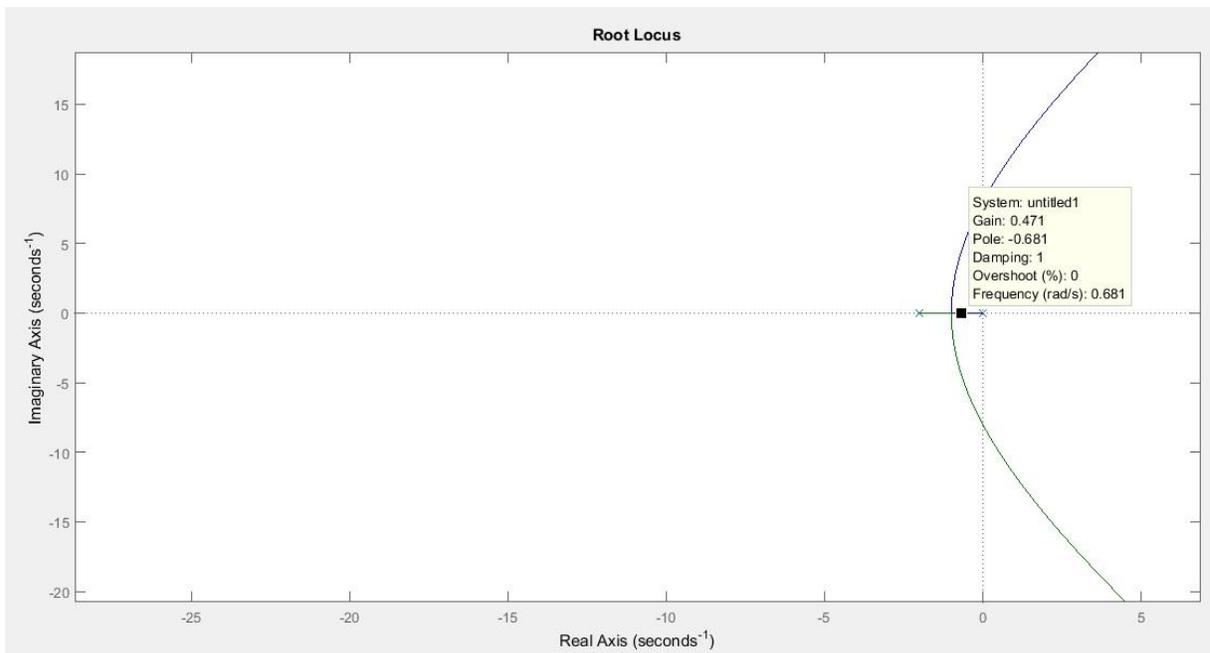
letak *pole* tertutup. Penentuan letak *poles* pada *root locus* harus sesuai dengan spesifikasi desain kontrol yang telah ditentukan. Pada penelitian ini akan dirancang agar sistem mempunyai respon transien yang lebih cepat dengan rasio redaman sama dengan 1, tidak terjadi *overshoot*, dan *settling time* kurang dari 6 detik. *Settling time* adalah ukuran waktu yang menyatakan respon telah masuk 5% atau 2% atau 0,5% dari keadaan *steady state*. Dari spesifikasi tersebut diperoleh,

$$\zeta = 1$$

$$ts = \frac{4}{\zeta \omega_n} \rightarrow \omega_n = \frac{4}{1 \cdot 6} = 0,67 \text{ rad/detik} \square \square 0,68 \text{ rad/detik}$$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} = 0,68 \sqrt{1 - 1^2} = 0,68 \text{ rad/detik}$$

Penentuan letak *poles* tersebut sudah sesuai dengan spesifikasi desain kontrol pada penelitian ini. Penentuan letak *poles root locus* dapat dilihat dalam Gambar 4.8.



Gambar 4.8 Letak *pole* pada diagram *root locus*

Dapat dilihat dalam Gambar 4.8, pada *gain* 1,15 *root locus* sistem akan berada pada *pole* yang diinginkan $s_1 = -0,681$, $\zeta = 1$, $\omega_n = 0,681$ rad/detik. Setelah menentukan letak *pole* yang diinginkan kemudian dengan mensubstitusi nilai s_1 dan nilai fungsi alih sistem dalam Persamaan 2-18 dan memvariasikan nilai K_i akan didapatkan parameter PID dalam Tabel 4.1. Pencarian parameter K_p , K_i , dan K_d dalam persamaan 2-18 dengan menggunakan MATLAB R2016a ditunjukkan pada *listing* program berikut :

```
%nilai pole yang ditentukan dari Gambar root locus

s1=-0.681
KI=[1 2 3 5 8]

plant_num=[0 0 60.3];
plant_den=[1 33.9 64.36];

s1mag = abs(s1)
beta = angle(s1)
plant_a1 = polyval(plant_num,s1)/polyval(plant_den,s1);
plants1mag = abs(plant_a1)
psi = angle(plant_a1)
t=0:1:20:300;

for k =1:5

    KP=-sin(beta+psi)/(plants1mag*sin(beta))    2*KI(k)*cos(beta)/s1mag
    nilai_KI= KI(k)
    KD = sin(psi)/(s1mag*plants1mag*sin(beta))+KI(k)/s1mag^2

    Gcnum = [KD KP KI(k)];
    Gcden = [0 1 0];

    Tnum = conv(plant_num,Gcnum);
    Tden = conv(plant_den,Gcden)+conv(plant_num,Gcnum);

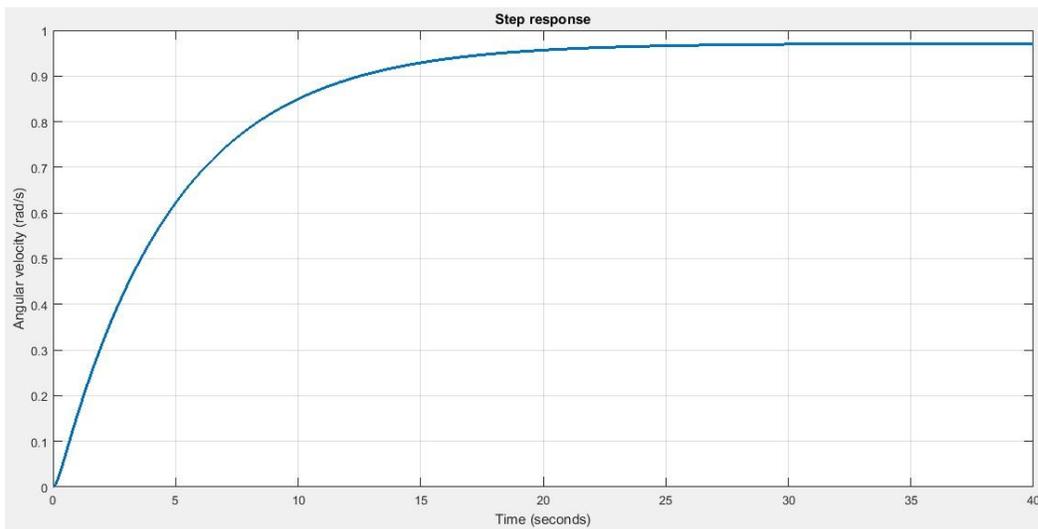
    r = roots(Tden)
    step (Tnum,Tden,t)
    %step (plant_num,plant_den,t)
    hold on
end
hold off
figure, rlocus(Tnum,Tden)
```

Hasil pencarian parameter K_p , K_i , dan K_d dari perhitungan pada program diatas ditunjukkan dalam Tabel 4.1

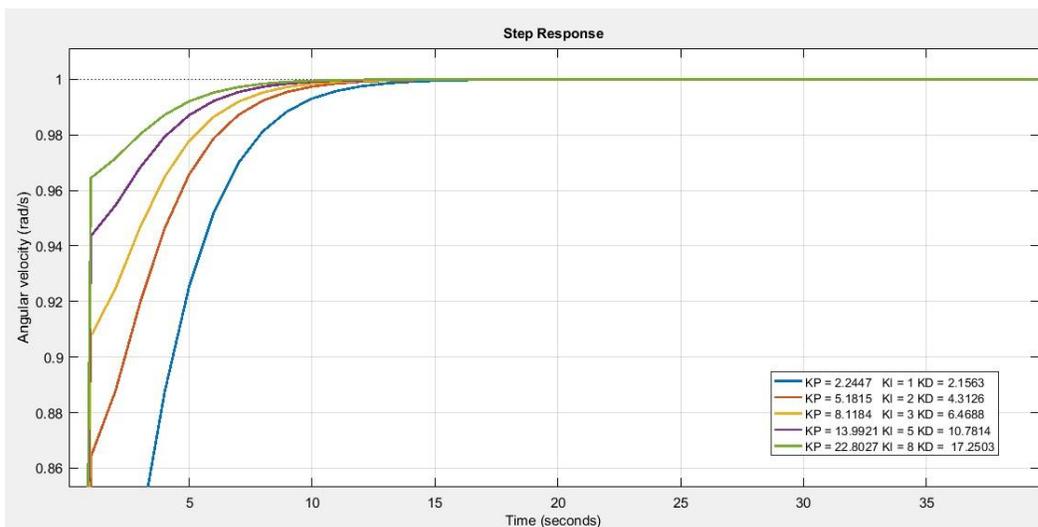
Tabel 4.1
Parameter PID dengan $s_1 = -0,681$

No.	Kp	Ki	Kd
1	2,2447	1	2,1563
2	5,1815	2	4,3126
3	8,1184	3	6,4688
4	13,9921	5	10,7814
5	22,8027	8	17,2503

Setelah didapatkan nilai Kp, Ki dan Kd hasil perhitungan, kemudian dilakukan pengujian terhadap sistem, dan parameter yang sesuai dengan sistem.



Gambar 4.9 Respon sistem tanpa kontroler PID

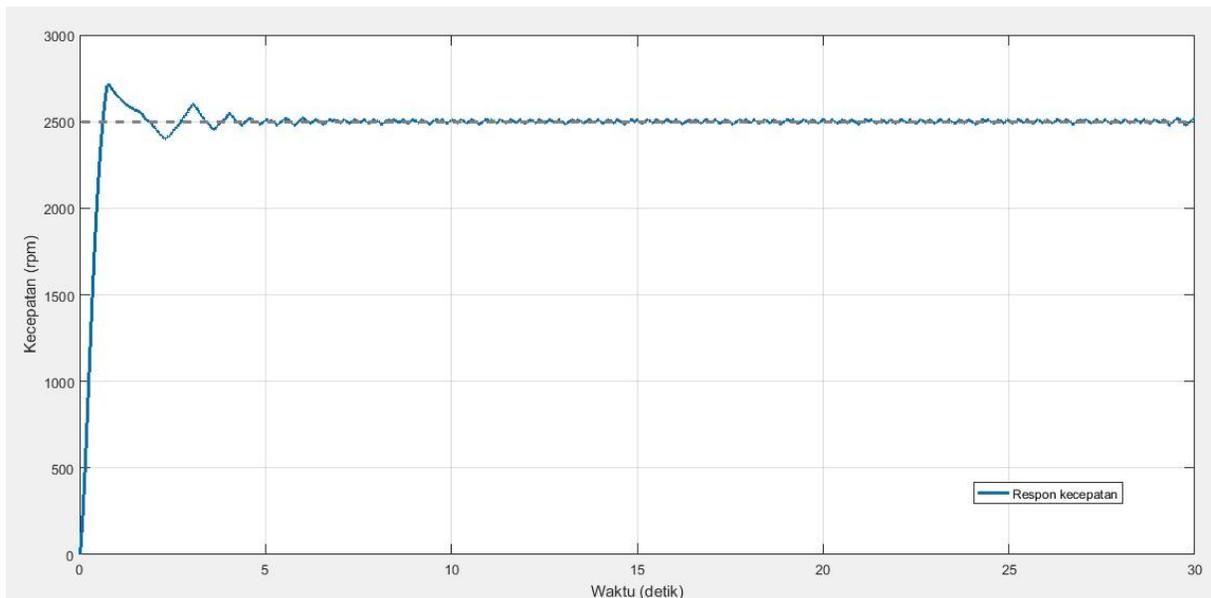


Gambar 4.10 Respon sistem dengan kontroler PID

Dari Gambar 4.9 dapat diketahui bahwa respon sistem tanpa menggunakan PID lebih lama menuju *steady state* dan tidak mencapai *setpoint* yang diinginkan. Dengan digunakannya parameter PID hasil tuning didapatkan respon yang lebih cepat dari pada respon tanpa menggunakan PID, serta dapat mencapai *setpoint* yang diinginkan seperti tertera dalam Gambar 4.10. Dari 5 jenis parameter PID yang didapat, dipilih nilai PID yang memiliki respon terbaik yaitu: $K_p = 22,8027$, $K_i = 8$, $K_d = 17,2503$

4.3 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan ini dilakukan untuk mengetahui seberapa besar nilai kontroler yang dibutuhkan agar sistem bekerja sesuai dengan *setpoint* yang diinginkan serta mengetahui hasil respon yang diimplementasikan pada alat. Pengujian sistem ini menggunakan *setpoint dust density* di udara sebesar $0,63 \text{ mg/m}^3$ dimana kecepatan putaran yang dihasilkan oleh karakteristik motor DC 12V sebesar 2500 rpm. Sehingga didapatkan respon sistem menggunakan metode *root locus* dalam Gambar 4.11 dan Gambar 4.12.



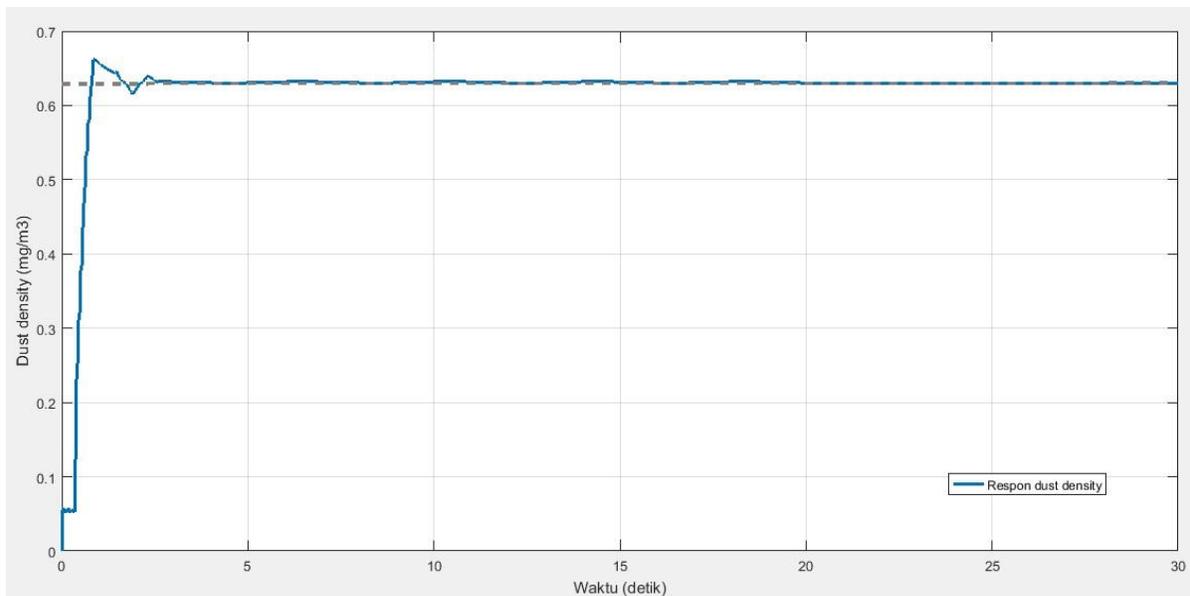
Gambar 4.11 Respon kecepatan motor DC

Dari respon sistem yang ditunjukkan dalam Gambar 4.11. Respon kecepatan motor DC dengan *setpoint* 2500 rpm memiliki *settling time* 4,5 detik. Walaupun sistem terdapat osilasi namun masih dalam toleransi 2%. Namun masih terjadi *overshoot* disaat sebelum *steady*, maka *overshoot* adalah:

$$\begin{aligned} \% M_p &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|2716 - 2500|}{2500} \times 100 \% \\ &= 8,64\% \end{aligned}$$

respon memiliki % *error* sebagai berikut:

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|2520 - 2500|}{2500} \times 100 \% \\ &= 0,8 \% \end{aligned}$$



Gambar 4.12 Respon *dust density*

Dari respon sistem yang ditunjukkan dalam Gambar 4.12 dapat diketahui semakin kecil kepadatan/massa jenis semen maka semen akan semakin baik. Dari respon dengan *setpoint dust density* 0,63 mg/m³ dengan karakteristik kecepatan motor sebesar 2500 rpm memiliki nilai *settling time* sebesar 3,4 detik. Namun masih terjadi *overshoot* disaat sebelum *steady*, maka *overshoot* adalah :

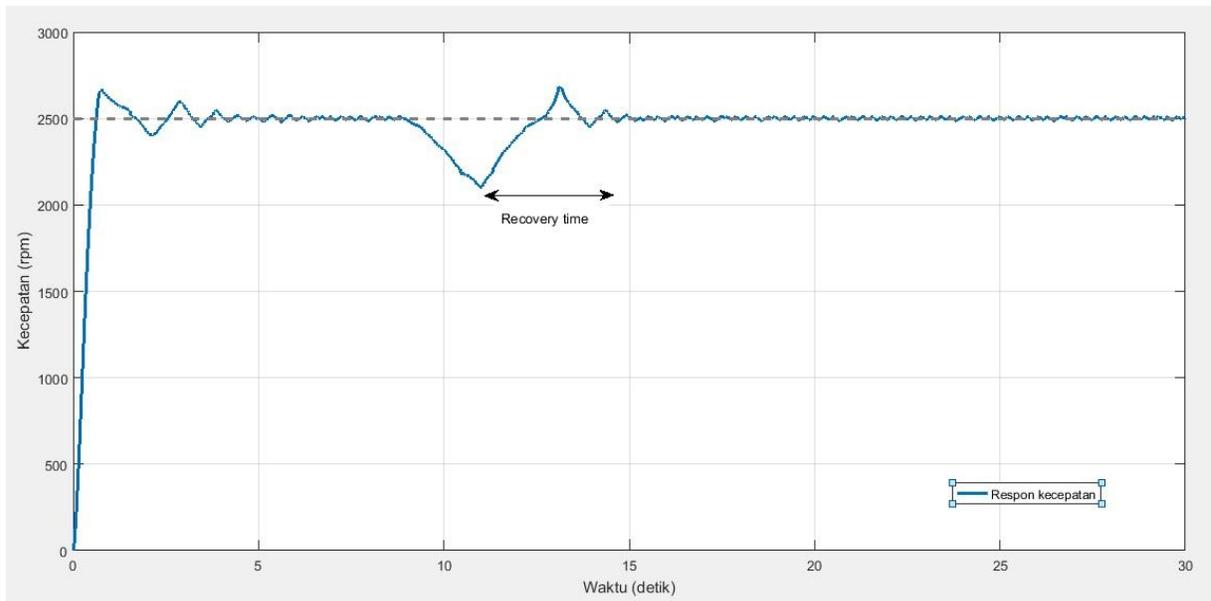
$$\begin{aligned} \% M_p &= \frac{|Density_{puncak} - Density_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|0.66 - 0,63|}{0,63} \times 100 \% \\ &= 4,76\% \end{aligned}$$

respon memiliki % *error* sebagai berikut:

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ Dust\ Density\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|0,632 - 0,63|}{0,63} \times 100\% \\ &= 0,31\% \end{aligned}$$

4.4 Pengujian Sistem dengan Diberikan Gangguan

Pengujian sistem dengan diberikan gangguan untuk mengetahui respon sistem pada *plant*. Gangguan yang diberikan pada *metal trap* pada saat berjalan adalah membuka tutup katub pada bagian atas *airslide* agar nantinya dapat dilihat apakah *plant* dapat mempertahankan posisi seperti awal dan mempunyai performa yang bagus ketika diberi gangguan yang besar ataupun gangguan yang kecil. Respon sistem dengan diberikan gangguan ditunjukkan dalam Gambar 4.13 dan Gambar 4.14.



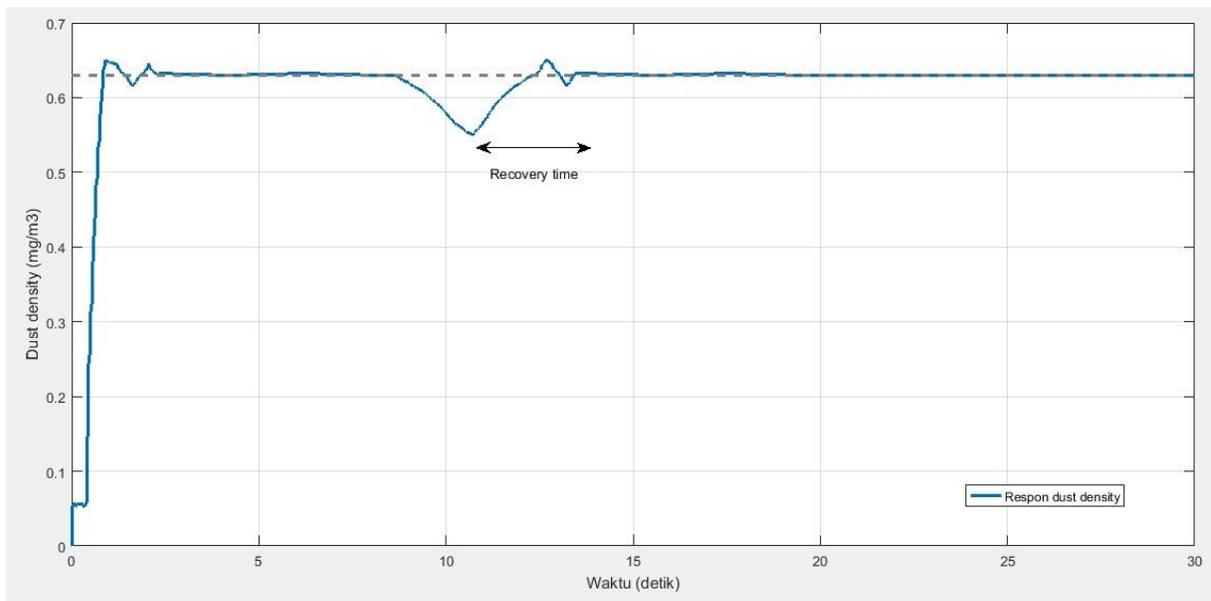
Gambar 4.13 Respon kecepatan motor DC saat diberikan gangguan

Dapat kita lihat dalam Gambar 4.13 respon kecepatan motor DC dengan *setpoint* 2500 rpm yang diberikan gangguan memiliki *settling time* 4,15 detik dan *recovery time* selama 3,7 detik. Sistem masih terjadi osilasi seperti saat sebelum diberi gangguan namun masih dalam toleransi 2%. Namun masih terjadi *overshoot* disaat sebelum *steady*, maka *overshoot* adalah:

$$\begin{aligned} \% M_p &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|2668 - 2500|}{2500} \times 100 \% \\ &= 6,72\% \end{aligned}$$

respon memiliki % *error* sebagai berikut:

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|2520 - 2500|}{2500} \times 100 \% \\ &= 0,8 \% \end{aligned}$$



Gambar 4.14 Respon *dust density* saat diberikan gangguan

Dapat kita lihat dalam Gambar 4.14, respon dengan *setpoint dust density* 0,63 mg/m³ dengan karakteristik kecepatan motor sebesar 2500 rpm sebelum diberikan gangguan memiliki settling time 3,2 detik. Ketika katub ditutup sebagai gangguan yang diberikan, respon mengalami perlambatan. Saat katub dibuka kembali, respon mengalami percepatan atau *recovery time* respon kembali keadaan *steady state* selama 3 detik. Namun masih terjadi *overshoot* disaat sebelum *steady*, maka *overshoot* adalah :

$$\begin{aligned} \% M_p &= \frac{|Density_{puncak} - Density_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|0,65 - 0,63|}{0,63} \times 100 \% \end{aligned}$$

$$= 3,17\%$$

respon memiliki % *error* sebagai berikut:

$$\% \text{ Ess} = \frac{|\text{Average Dust Density steady- setpoint}|}{\text{setpoint}} \times 100\%$$

$$= \frac{|0,632-0,63|}{0,63} \times 100 \%$$

$$= 0,31\%$$