

**DESAIN KONTROLER STRUKTUR *OUTPUT FEEDBACK*
CONTROL DENGAN *POLE PLACEMENT* PADA PENGONTROLAN
TEGANGAN *OUTPUT* GENERATOR DC**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



ZAKIYAH AMALIA

NIM. 125060300111065

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2016

UNIVERSITAS BRAWIJAYA



PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 11 April 2016

Mahasiswa,

ZAKIYAH AMALIA

NIM. 125060300111065

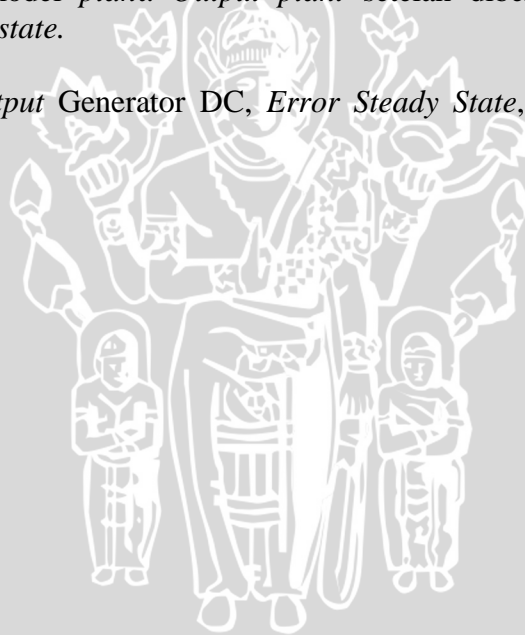


RINGKASAN

Zakiyah Amalia, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, April 2016, Desain Kontroler Struktur *Output Feedback Control* dengan *Pole Placement* pada Pengontrolan Tegangan *Output* Generator DC, Dosen Pembimbing: Erni Yudaningtyas dan Mochammad Rusli.

Plant motor generator DC 734 11 dengan beban resistif mempunyai *error steady state* yang cukup besar. Pengontrolan tegangan dilakukan untuk menjaga agar tegangan output generator DC tetap konstan dan tidak mempunyai *error steady state* apabila diberi perubahan beban berupa lampu yang bersifat resistif. *Pole placement* adalah suatu metode desain kontrol dimana menentukan letak *pole* sistem *close loop* pada bidang kompleks dengan mengatur *gain kontroler*. Penetapan kutub sistem dapat digunakan untuk mengoptimasi performansi sistem *close loop* dengan meminimalisasi energi dengan *output* mencapai kondisi *steady state* secepat mungkin. Salah satu desain kontroler adalah *Output Feedback Control* (OFC). OFC disertai dengan *observer/estimator* yang digunakan untuk mengestimasi *state-state* pada pengontrolan tegangan *output* generator DC 734 11 berdasarkan model *plant*. *Output plant* setelah diberi kontroler dapat meminimalkan *error steady state*.

Kata kunci: Tegangan *Output* Generator DC, *Error Steady State*, *Output Feedback Control*.



SUMMARY

Zakiyah Amalia, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, April 2016, Controller Design Structure Output Feedback Control with Pole Placement in Control of Output Voltage DC Generator, Academic Supervisor: Erni Yudaningtyas and Mochammad Rusli.*

734 11 DC motor generator plant with a resistive load has a quite large steady-state error. Voltage control is done to keep the output voltage of the DC generator remains constant and does not have a steady state error when given load change a lamp that is resistive. Pole placement method is a controller design method in which you determine the places of the closed loop system poles on the complex plane by setting a controller gain. Pole placement can be used to optimize system performance close loop by minimizing energy with output reaching steady state conditions as soon as possible. One of controller design is Output Feedback Control (OFC). OFC accompanied by observer/ estimator is used to estimate states of controlling the generator output voltage DC 734 11 based on the plant model. Plant Output after being given controller can minimize error steady state.

Keywords: *Output Voltage of Generator DC, Error Steady State, Output Feedback Control.*



PENGANTAR

Bismillahirrohmanirrohim. Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Desain Kontroler Struktur *Output Feedback Control* dengan *Pole Placement* pada Pengontrolan Tegangan *Output Generator DC*” dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

- Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayah-Nya.
- Keluarga tercinta, kedua orang tua Achmad Hariyadi dan Erni Yudaningtyas yang selalu memberikan kasih sayang dan doanya yang tiada akhir. Kakak tercinta Afif dan adik tercinta Amira yang selalu memberikan dukungan.
- Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya
- Bapak Ali Mustofa, ST., MT. selaku Ketua Program Studi S1 Jurusan Teknik Elektro Universitas Brawijaya
- Ibu Dr. Ir. Erni Yudaningtyas, MT. selaku Ka. Lab Sistem Kontrol sekaligus sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Bapak Ir. Mochammad Rusli, Dipl. Ing. sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Ferdian Ade Jiwandono, terima kasih atas waktu, pengertian, semangat, bantuan dan kesabarannya yang telah diberikan.
- Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya atas segala alat serta sarana dan prasarana yang dimanfaatkan penulis dalam melakukan penelitian.

- Pranata Laboratorium, Pak Dedi dan Keluarga besar asisten Laboratorium Dasar Elektrik dan Pengukuran, Ferdian, Gadis, Faizal, Fajar, Hesa, Bobby, Milky, Najar, Arif, Citra, Fitrah, Ina, Nola, Dini, Topan, Bima, Rahmat, Taka, Okto, terima kasih telah memberikan banyak bantuan, dukungan dan canda tawa.
- Teman-teman Suro, Nora, Indri, Regina, Faizal, Hesa, Hanif, Gabriel, Tyo, Odi.
- Teman-teman GM, Alifa, Isfana, Vyo, Agusta, Safitri, Mella, Yulia, Retno.
- Pranata Laboratorium Sistem Kontrol, Mbak Eka, teman-teman asisten Laboratorium Sistem Kontrol, Mas Rozi, Suro, Andri, Yudha, Faris, Dennis, Indrut, Hilmy, Avif, Diana yang telah memberikan banyak bantuan dan pengetahuan.
- Keluarga besar Sistem Kontrol angkatan 2012, teman-teman angkatan 2012 “Voltage” dan “Votagirls” atas do'a, semangat, serta dukungan yang diberikan pada penulis.
- Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, April 2016

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
PERNYATAAN ORISINALITAS SKRIPSI.....	i
RINGKASAN.....	ii
SUMMARY.....	iii
PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Mesin <i>Direct Current</i> (DC).....	5
2.1.1 Motor <i>Direct Current</i> (DC).....	5
2.1.2 Generator <i>Direct Current</i> (DC).....	7
2.1.3 <i>Plant</i> Motor Generator DC (734 11).....	9
2.2 <i>Pulse Width Modulation</i> (PWM).....	9
2.3 <i>High Current</i> 30 A DC Motor <i>Driver</i> dengan <i>half bridge infineon</i> BTS 7960 chip.....	10
2.4 Mikrokontroler Arduino Uno R3.....	11
2.5 Sensor Tegangan.....	12
2.6 Beban Resistif.....	12
2.7 Lampu Pijar (bohlam).....	13

2.8 Output Sistem Orde Satu	14
2.8.1 Output Unit Step Pada Sistem Orde Satu.....	14
2.9 Output Sistem Orde Dua.....	15
2.9.1 Keadaan Kurang Teredam / Underdamped ($0 < \xi < 1$)	17
2.9.2 Teredam Kritis / Critically Damped ($\xi = 1$)	18
2.9.3 Terlalu Teredam / Overdamped ($\xi > 1$)	18
2.10 Tanggapan Peralihan.....	19
2.11 Kontroler.....	20
2.11.1 Kontroler Proporsional (P).....	22
2.11.2 Kontroler Integral (I)	22
2.11.3 Kontroler Proporsional Integral (PI).....	23
2.11.4 Kontroler Proporsional Integral Diferensial (PID).....	24
2.12 Konsep Ruang Keadaan (<i>State Space</i>).....	25
2.13 Konsep Sistem Kontrol Optimal.....	26
2.14 Linear Quadratic Regulator (LQR).....	26
2.15 State Observer.....	28
2.16 <i>Output Feedback Control</i>	29
2.17 Desain Pra Filter	30
BAB III METODE PENELITIAN.....	33
3.1 Perancangan Blok Diagram Sistem	33
3.2 Spesifikasi Desain.....	34
3.3 Karakterisasi Motor DC.....	34
3.4 Karakterisasi <i>Driver Motor (High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip)</i>	36
3.5 Karakterisasi Sensor Tegangan.....	38
3.6 Penentuan Fungsi Alih <i>Plant</i> Motor Generator DC 734 11.....	40
3.7 Validasi Fungsi Alih Motor Generator 734 11	43

3.8 Diagram Blok Perancangan Sistem	44
3.9 Pembuatan Perangkat Keras	45
3.10 Perancangan Algoritma.....	47
3.11 Desain Kontrol Optimal LQR Struktur <i>Output Feedback Control</i>	48
3.12 <i>Flowchart</i> Program	51
BAB IV HASIL DAN PEMBAHASAN.....	53
4.1 Simulasi Output Fungsi Alih dan Diagram Blok Bentuk State Space pada Simulink	53
4.2 Simulasi penentuan matriks R, L, M dari perubahan pole kontroler dan perubahan pole observer.....	54
4.3 Simulasi Perubahan Pole Observer terhadap perubahan nilai matriks R, L, M....	55
4.3.1 Perubahan Nilai op1	55
4.3.2 Perubahan Nilai op2	58
4.4 Simulasi Perubahan Pole Kontroler terhadap perubahan nilai matriks R, L, M... 61	
4.4.1 Perubahan Nilai rp1.....	61
4.4.2 Perubahan Nilai rp2.....	64
4.5 Simulasi dengan Gangguan.....	67
4.6 Simulasi Karakteristik Plant Sebelum dan Sesudah diberi Kontroler	68
4.7 Implementasi pada Alat.....	69
4.7.1 Implementasi dengan mengubah-ubah setpoint.....	69
4.7.2 Implementasi dengan gangguan	72
BAB V KESIMPULAN DAN SARAN	74
5.1 Kesimpulan	74
5.2 Saran	74
DAFTAR PUSTAKA	75
LAMPIRAN	76

DAFTAR TABEL

Tabel 2. 1 Fungsi Pin *High Current 30 A DC Motor Driver* 11

Tabel 3. 1 Data pengujian kecepatan motor DC terhadap tegangan.....35

Tabel 3. 2 Data pengujian driver motor High Current 30 A DC Motor Driver.....37

Tabel 3. 3 Data pengujian sensor tegangan 39

Tabel 4. 1 Nilai matriks R, L, M dan eigen value terhadap perubahan nilai observer pole op1.....56

Tabel 4. 2 Nilai matriks R, L, M dan eigen value terhadap perubahan nilai observer pole op2..... 59

Tabel 4. 3 Nilai matriks R, L, M dan eigen value terhadap perubahan nilai kontroler pole rp1 62

Tabel 4. 4 Nilai matriks R, L, M dan eigen value terhadap perubahan nilai kontroler pole rp2 65

Tabel 4. 5 Nilai matriks R, L, M untuk simulasi dengan gangguan..... 67



DAFTAR GAMBAR

Gambar 2. 1 Rangkaian ekivalen motor DC magnet permanen	6
Gambar 2. 2 Diagram blok pengontrolan kecepatan motor DC magnet permanen.....	7
Gambar 2. 3 Motor DC magnet permanen D-6759	7
Gambar 2. 4 Plant motor generator DC (734 11)	9
Gambar 2. 5 Sinyal PWM.....	10
Gambar 2. 6 <i>High Current 30 A DC Motor Driver</i>	10
Gambar 2. 7 Skema Arduino Uno	12
Gambar 2. 8 Sensor tegangan	12
Gambar 2. 9 Simbol resistor	13
Gambar 2. 10 Lampu pijar.....	14
Gambar 2. 11 Sistem orde satu	14
Gambar 2. 12 Output unit step sistem orde satu	15
Gambar 2. 13 Sistem orde dua.....	16
Gambar 2. 14 Output unit step sistem orde dua.....	20
Gambar 2. 15 Diagram blok sistem dengan kontroler otomatis	21
Gambar 2. 16 Diagram blok kontroler proposional (P).....	22
Gambar 2. 17 Diagram blok kontroler integral (I).....	23
Gambar 2. 18 Diagram blok kontroler proporsional integral (PI).....	23
Gambar 2. 19 Diagram blok kontroler proporsional integral diferensial (PID)	24
Gambar 2. 20 Representasi sistem dalam bentuk persamaan state	25
Gambar 2. 21 <i>Full order state observer</i>	28
Gambar 2. 22 Struktur <i>output feedback control</i>	30
Gambar 2.23 Desain pra filter.....	31
Gambar 3. 1 Blok diagram sistem loop tertutup.....	33
Gambar 3. 2 Grafik perubahan kecepatan motor DC terhadap tegangan	36
Gambar 3. 3 Grafik perubahan tegangan terhadap duty cycle.....	38
Gambar 3. 4 Grafik tegangan output generator	40
Gambar 3. 5 Output sinyal PRBS	41
Gambar 3. 6 <i>System identification toolbox</i>	41
Gambar 3. 7 Hasil estimasi model	42

Gambar 3. 8 Nilai <i>time constant output plant</i> motor generator DC 734 11 dengan input <i>unit step</i>	42
Gambar 3. 9 Nilai <i>rise time output plant</i> motor generator DC 734 11 dengan input <i>unit step</i>	43
Gambar 3. 10 Nilai <i>settling time output plant</i> motor generator DC 734 11 dengan input <i>unit step</i>	43
Gambar 3. 11 Validasi fungsi alih dengan <i>output</i> motor DC D-6759	44
Gambar 3. 12 Diagram blok perancangan sistem	44
Gambar 3. 13 Skema perangkat keras	45
Gambar 3. 14 <i>Power supply unit (PSU)</i>	45
Gambar 3. 15 <i>Arduino Uno R3</i>	46
Gambar 3. 16 Sensor tegangan	46
Gambar 3. 17 <i>High current 30 A DC motor driver</i>	46
Gambar 3. 18 <i>Motor generator DC 734 11</i>	47
Gambar 3. 19 Skema pembuatan perangkat keras	47
Gambar 3. 20 <i>Flowchart</i> program	51
Gambar 4. 1 Simulasi <i>output</i> fungsi alih dan diagram blok pada simulink.....	53
Gambar 4. 2 Output fungsi alih dan diagram blok pada simulink.....	54
Gambar 4. 3 Simulasi penggunaan matriks R, L, M.....	55
Gambar 4. 4 Grafik U dengan nilai op1 berbeda-beda op2 tetap, $op=[op1;op2]$	57
Gambar 4. 5 <i>Output</i> sistem dengan nilai op1 berbeda-beda op2 tetap, $op=[op1;op2]$...	57
Gambar 4. 6 Grafik U dengan nilai op2 berbeda-beda op1 tetap, $op=[op1;op2]$	60
Gambar 4. 7 <i>Output</i> sistem dengan nilai op2 berbeda-beda op1 tetap, $op=[op1;op2]$...	60
Gambar 4. 8 Grafik U dengan nilai rp1 berbeda-beda rp2 tetap, $rp=[rp1;rp2]$	63
Gambar 4. 9 <i>Output</i> sistem dengan nilai rp1 berbeda-beda rp2 tetap, $rp=[rp1;rp2]$	63
Gambar 4. 10 Grafik U dengan nilai rp2 berbeda-beda rp1 tetap, $rp=[rp1;rp2]$	66
Gambar 4. 11 <i>Output</i> sistem dengan nilai rp2 berbeda-beda rp1 tetap, $rp=[rp1;rp2]$	66
Gambar 4. 12 Simulasi dengan gangguan	67
Gambar 4. 13 Tegangan <i>output</i> sistem dengan gangguan	68
Gambar 4. 14 Karakteristik tegangan <i>output plant</i> sebelum dan sesudah diberi kontroler	69
Gambar 4. 15 Tegangan <i>output</i> sistem dengan setpoint 1 Volt	70
Gambar 4. 16 Tegangan <i>output</i> sistem dengan setpoint 2,5 Volt	70
Gambar 4. 17 Tegangan <i>output</i> sistem dengan setpoint 3 Volt	71

Gambar 4. 18 Tegangan *output* sistem dengan setpoint 5 Volt 71

Gambar 4. 19 Tegangan *output* sistem dengan setpoint 7 Volt 72

Gambar 4. 20 Tegangan *output* sistem dengan setpoint 9 Volt 72

Gambar 4. 21 Tegangan *output* sistem dengan gangguan dengan setpoint 2,5 Volt 73



BAB I

PENDAHULUAN

1.1 Latar Belakang

Generator *Direct Current* (DC) adalah sebuah perangkat mesin listrik yang mengubah energi mekanik menjadi energi elektrik. Hukum Faraday menyebutkan jika terjadi perubahan garis gaya magnet pada sebuah kumparan kawat, maka akan timbul gaya gerak listrik (ggl) pada kawat tersebut. Jika kumparan kawat dihubungkan dengan rangkaian listrik tertutup, maka akan timbul pula arus listrik yang mengalir pada rangkaian.

Plant motor generator DC 734 11 merupakan panel pelatihan untuk teknologi kontrol otomatis yang cocok untuk bereksperimen dengan peralatan listrik. *Plant* motor generator DC 734 11 dapat dihubungkan ke beban listrik. Beban listrik ini bisa berupa lampu bohlam. Apabila terjadi perubahan beban, tegangan *output* generator DC akan berubah. Oleh karena itu, diperlukan metode kontrol yang bisa membuat error steady state seminimal mungkin dan bisa menjaga agar tegangan *output* generator DC bernilai hampir konstan walaupun terjadi perubahan beban agar performansi sistem tidak berubah.

Salah satu tujuan desain sistem kontrol adalah agar sinyal aktual (*output*) sistem mengikuti atau mencapai sinyal tujuan (*setpoint/input*) secepat mungkin. Salah satu parameter untuk mencapai *setpoint* adalah adanya sinyal *error* (selisih antara *setpoint/input* dengan sinyal *output*). Selain itu, magnitude dari perjalanan sinyal *error* adalah sekecil mungkin, dengan kata lain mempunyai redaman sebesar mungkin.

Pole placement adalah suatu metode desain kontrol dimana menentukan letak dari kutub sistem *close loop* pada bidang kompleks dengan mengatur *gain kontroler*. Penetapan kutub sistem dapat digunakan untuk mengoptimasi performansi sistem *close loop* dengan meminimalisasi energi dengan *output* mencapai kondisi *steady state* secepat mungkin. Salah satu desain kontroler adalah *Output Feedback Control* (OFC). OFC disertai *observer/estimator* untuk mengestimasi *state-state* yang tidak dapat diukur berdasarkan model *plant* dan *output* yang dapat diukur. Dengan OFC, *state-state* yang ada dapat diestimasi hanya dengan menggunakan *input* dan *output* sistem.

Pada pengontrolan tegangan *output* generator DC membutuhkan informasi mengenai *output* tegangan generator DC. Sensor tegangan mendeteksi perubahan

tegangan *output* generator DC. Output sensor yang berupa tegangan dibandingkan dengan tegangan *setpoint* generator DC. Apabila terdapat selisih (*error*), maka kecepatan motor akan berubah sesuai dengan tegangan *error* yang terjadi. Kecepatan putaran motor akan membuat tegangan *output* generator DC dapat sesuai dengan *setpoint* atau sama dengan tegangan sebelum diberi perubahan beban.

Dalam skripsi ini akan dibahas mengenai bagaimana cara mendesain dan menggunakan *OFC* pada *plant* motor generator DC. Hal tersebut bertujuan agar tegangan *output* generator bernilai hampir konstan walaupun terdapat perubahan beban. Dengan kata lain, *error steady state* sekecil mungkin walau terdapat perubahan beban pada *plant* motor generator DC. Selain itu, juga bertujuan untuk mendapatkan *settling time* secepat mungkin dengan energi seminimum mungkin.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut di atas, maka rumusan masalahnya dapat disusun sebagai berikut:

1. Bagaimana menentukan fungsi alih *plant* motor generator DC?
2. Bagaimana merancang algoritma kontroler pada pengontrolan tegangan *output* pada *plant* motor generator DC dengan struktur *Output Feedback Control* (*OFC*)?
3. Bagaimana *output plant* motor generator DC setelah menggunakan *OFC* pada simulasi?
4. Bagaimana *output plant* motor generator DC menggunakan *OFC* pada implementasi dan saat sistem diberi gangguan?

1.3 Batasan Masalah

Dalam skripsi ini terdapat beberapa batasan masalah, yaitu:

1. *Plant* yang digunakan adalah motor generator DC (734 11) di Laboratorium Sistem Kontrol FTUB.
2. *Plant* motor generator DC (734 11) terdiri dan 2 buah mesin DC magnet permanen 20 Volt yang dikopel, dimana 1 mesin adalah motor sebagai penggerak generator dan yang lainnya digunakan sebagai generator DC.
3. Mesin DC mempunyai spesifikasi catu daya maksimal 20 V, arus 0,47 A, daya 3,8 W dan range kecepatan (0 – 3000) rpm.
4. Kontroler yang digunakan adalah Mikrokontroler Arduino Uno R3.

5. Beban generator DC adalah beban resistif berupa lampu bohlam 2,5 Volt.
6. Pembahasan ditekankan pada cara mendesain dan penggunaan struktur *Output Feedback Control* (OFC).
7. Pola pembebanan menggunakan pola deterministik meliputi prosedur pengaktifan satu persatu beban dengan jeda waktu tertentu pada setiap pengaktifannya.

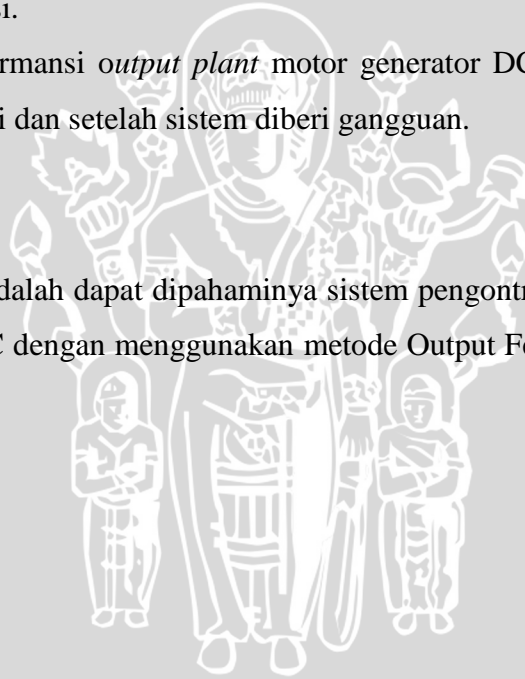
1.4 Tujuan

Tujuan skripsi ini adalah:

1. Menentukan fungsi alih *plant* motor generator DC.
2. Merancang algoritma kontroler pada pengontrolan tegangan *output plant* motor generator DC dengan struktur *Output Feedback Control* (OFC).
3. Mengetahui performansi *output plant* motor generator DC setelah menggunakan OFC pada simulasi.
4. Mengetahui performansi *output plant* motor generator DC menggunakan OFC pada implementasi dan setelah sistem diberi gangguan.

1.5 Manfaat

Manfaat skripsi ini adalah dapat dipahaminya sistem pengontrolan tegangan *output plant* motor generator DC dengan menggunakan metode *Output Feedback Control* pada sistem kontrol optimal.





BAB II

TINJAUAN PUSTAKA

Bab ini menguraikan teori-teori pendukung skripsi, yang terdiri atas :

2.1 Mesin *Direct Current* (DC)

Mesin *Direct Current* (DC) merupakan perangkat untuk mengubah energi mekanik menjadi energi listrik maupun untuk mengubah energi listrik menjadi energi mekanik. Elemen mesin DC terdiri atas elemen yang berdiri tetap/statis yang disebut stator, elemen yang berputar/ dinamis yang disebut rotor, dan bagian-bagian lainnya. Stator terdiri atas gandar, kutub utama dengan bagian inti kutub dan sepatu kutub, kutub komutasi/ kutub bantu, dan kumparan penguatan. Rotor terdiri atas inti jangkar, kumparan jangkar, kawat baja pengikat, dan komutator. Bagian-bagian lainnya terdiri atas celah-celah udara dan sikat-sikat.

2.1.1 Motor *Direct Current* (DC)

Mesin *Direct Current* (DC) yang digunakan untuk mengubah energi listrik menjadi energi mekanik adalah motor DC. Motor DC merupakan motor listrik yang sangat sering digunakan sebagai elemen kontrol akhir dalam sistem kontrol posisi dan kecepatan. Prinsip kerja motor DC sesuai dengan hukum Lorentz, apabila arus dialirkan melalui kumparan jangkar dari mesin DC dan kumparan medannya diberi penguatan, maka akan timbul Gaya Lorentz pada tiap sisi kumparan jangkar tersebut (Soemarwanto, 2010).

Besar gaya Lorentz yang ditimbulkan motor DC adalah

$$F = B \cdot I \cdot l \text{ dyne} \quad (2-1)$$

dengan:

B = kerapatan medan magnetik (Gauss).

I = arus listrik (Ampere)

l = panjang konduktor (cm)

Akibat timbulnya gaya Lorentz, maka timbul kopel yang besarnya adalah:

$$\text{Kopel: } M = F \times r \text{ dyne-cm} \quad (2-2)$$

dimana:

r = jari-jari jangkar (cm)

$$M = B \cdot I \cdot l \cdot r \text{ dyne-cm} \quad (2-3)$$

Apabila jumlah: $M = z \cdot B \cdot I \cdot l \cdot r$ dyne-cm

Karena $B = \frac{\phi}{A}$

dimana:

ϕ = Fluksi tiap kutub (Maxwell)

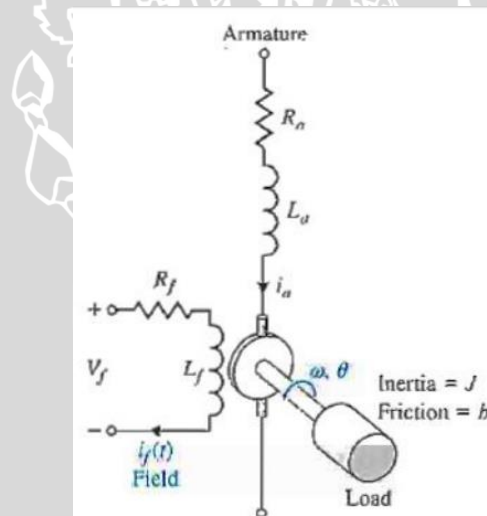
A = Luas penampang (cm^2)

z = Jumlah konduktor di jangkar

maka: $M = z \cdot \frac{\phi}{A} \cdot I \cdot l \cdot r$ dyne-cm (2-4)

Motor *Direct Current* (DC) merupakan aktuatur yang banyak digunakan dalam teknologi kontrol. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan dalam satu lilitan disebut *armature* (jangkar). *Armature* adalah sebutan untuk komponen yang berputar di antara medan magnet.

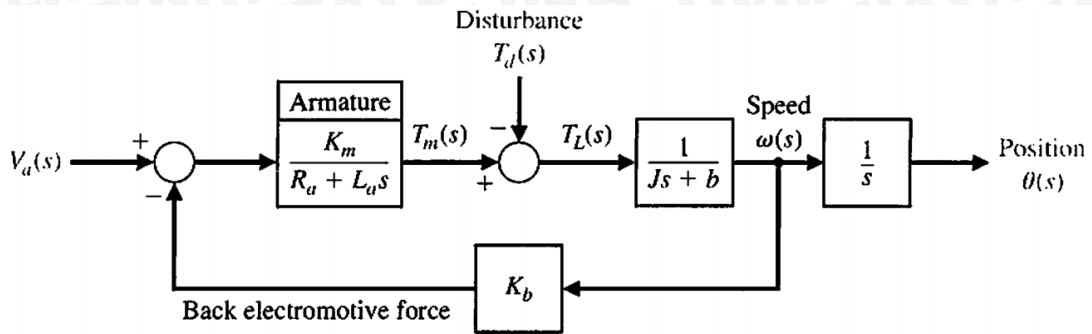
Rangkaian ekivalen motor DC magnet permanen dapat dilihat dalam Gambar 2.1.



Gambar 2. 1 Rangkaian ekivalen motor DC magnet permanen

Sumber: Dorf, R. C. & Bishop, R. H. (2011)

Pada motor DC magnet permanen, arus kumparan medan I_f dibuat konstan. Diagram blok pengontrolan kecepatan motor DC magnet permanen dapat dilihat dalam Gambar 2.2.



Gambar 2. 2 Diagram blok pengontrolan kecepatan motor DC magnet permanen
 Sumber: Dorf, R. C. & Bishop, R. H. (2011)

Apabila *disturbance* (gangguan) $T_d(s)=0$, maka fungsi alih motor DC adalah

$$\frac{\omega(s)}{V_a(s)} = \frac{K_m}{(R_a + L_a s)(J s + b) + (K_b K_m)}$$

dengan:

- K_m = konstanta motor
- K_b = konstanta *back electromotive force*
- R_a = resistansi *armature*
- L_a = induktansi *armature*
- J = inersia
- b = gesekan

Salah satu jenis motor DC adalah motor DC magnet permanen D-6759 seperti terlihat dalam Gambar 2.3.



Gambar 2. 3 Motor DC magnet permanen D-6759

2.1.2 Generator *Direct Current* (DC)

Mesin DC yang digerakkan secara mekanik oleh penggerak mula misalnya turbin uap, turbin hidrolik, atau mesin diesel dan menghasilkan energi listrik untuk lampu listrik atau mesin-mesin, maka mesin DC disebut generator DC yang menghasilkan arus DC.

Prinsip kerja dari suatu generator DC adalah berdasarkan hukum Faraday mengenai induksi elektromagnetik. Hukum Faraday menyebutkan “Apabila suatu



konduktor digerakkan dan memotong garis-garis gaya magnetik (fluksi), maka pada konduktor tersebut akan timbul suatu gaya gerak listrik (g.g.l.)” (Soemarwanto, 2010). Jika kumparan kawat dihubungkan dengan rangkaian listrik tertutup, maka akan timbul pula arus listrik yang mengalir pada rangkaian.

Berdasarkan Hukum Induksi Faraday/ Hukum Faraday besarnya g.g.l. yang diinduksikan satu belitan adalah:

$$e = -\frac{d\phi}{dt} \cdot 10^{-8} \text{ Volt } (\phi \text{ dalam Maxwell}). \quad (2-5)$$

Untuk N belitan maka:

$$e = -N \frac{d\phi}{dt} \times 10^{-8} \text{ Volt}. \quad (2-6)$$

Apabila kecepatan putaran = n putaran/ menit, maka untuk 1 putaran = 1/n menit = 60/n detik, konduktor akan memotong P x ϕ Maxwell.

$$\text{Sehingga: } e_{rata-rata} = \frac{p \cdot \phi}{60/n} \times 10^{-8} \text{ Volt} = \frac{p \cdot n \cdot \phi}{60} \times 10^{-8}$$

Karena ada $\frac{z}{a}$ konduktor pada setiap cabang jangkar paralel, maka:

$$E = \frac{z \cdot p \cdot n \cdot \phi}{a \cdot 60} \times 10^{-8} \text{ Volt}$$

$$\text{Jadi: } E = \frac{p \cdot n}{a \cdot 60} \cdot z \cdot \phi \times 10^{-8} \text{ Volt} \quad (2-7)$$

dengan:

- P = Jumlah kutub.
- a = Jumlah cabang jangkar paralel.
- n = Jumlah putaran (putaran/menit).
- z = Jumlah konduktor di jangkar.
- ϕ = Fluksi tiap kutub (Maxwell).

Di dalam generator arus searah besar p, a, dan z adalah tertentu/ tetap/ konstan.

$$\text{Maka dari rumus } E = \frac{p \cdot n}{a \cdot 60} \cdot z \cdot \phi \times 10^{-8} \text{ Volt}$$

Apabila $\frac{p \cdot 1}{a \cdot 60} \cdot z \times 10^{-8} C$ (konstan), maka:

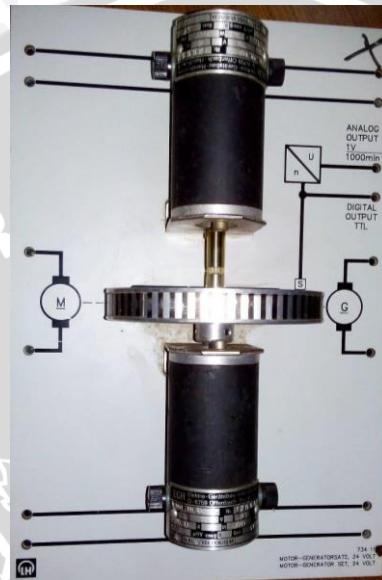
$$E = C \cdot n \cdot \phi \text{ Volt} \quad (2-8)$$

Dengan:

- E : Tegangan induksi
- C : Konstanta
- n : Jumlah putaran (putaran/menit).
- ϕ : Fluksi tiap kutub (Maxwell).

2.1.3 Plant Motor Generator DC (734 11)

Plant motor generator DC (734 11) di Laboratorium Sistem Kontrol TEUB (Gambar 2.4) terdiri atas generator DC yang dikopel dengan motor DC. Putaran motor DC diubah oleh generator DC menjadi tegangan listrik. Motor DC tersebut mempunyai spesifikasi yaitu catu daya maksimal 20 V, arus 0,47 A, daya 3,8 W dan range kecepatan (0 – 3000) rpm.



Gambar 2. 4 Plant motor generator DC (734 11)

2.2 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) adalah salah satu jenis modulasi. Modulasi PWM dilakukan dengan cara mengubah lebar pulsa dari suatu pulsa data. Total 1 periode (T) pulsa dalam PWM adalah tetap, dan data PWM pada umumnya menggunakan perbandingan pulsa positif terhadap total pulsa. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh *output* yang diinginkan. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada plant.

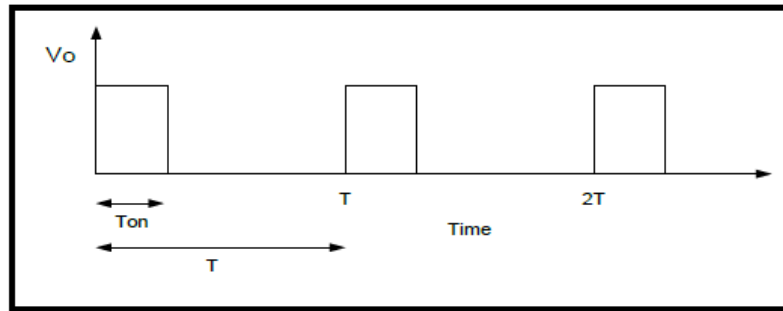
$$\text{Duty cycle} = T_{\text{on}}/T \times 100\% \quad (2-9)$$

dengan:

T_{on} = periode logika tinggi

T = periode keseluruhan

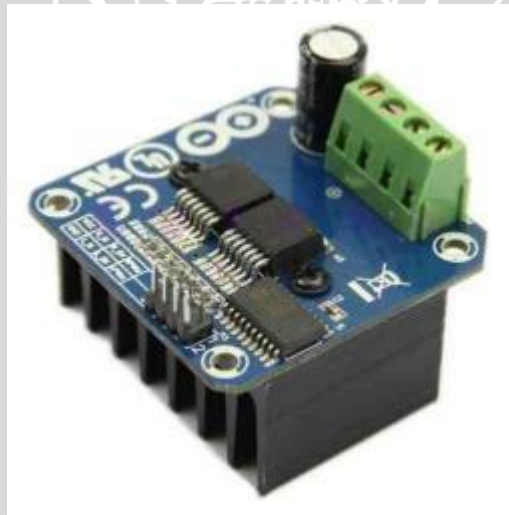
Sinyal PWM secara umum ditunjukkan dalam Gambar 2.5, dimana V_o adalah amplitudo.



Gambar 2. 5 Sinyal PWM
Sumber: Ardyani, F. (2013)

2.3 High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip

High Current 30 A DC Motor Driver (Gambar 2.6) merupakan *driver* H-Bridge yang didesain untuk menghasilkan *drive* 2 arah gerakan *clockwise* (cw) dan *counterclockwise* (ccw) dengan arus kontinu sampai dengan 30 A pada tegangan 5 Volt sampai 27 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengontrol. Modul ini mampu men-*drive* (menggerakkan) motor DC yang menjadi penggerak generator. Rangkaian ini dihubungkan dengan Arduino Uno R3.



Gambar 2. 6 High Current 30 A DC Motor Driver
Sumber: datasheet High Current 30 A DC Motor Driver

Fungsi masing-masing pin pada *Interface Header* dapat dilihat dalam Tabel 2.1.

Tabel 2. 1 Fungsi Pin *High Current 30 A DC Motor Driver*

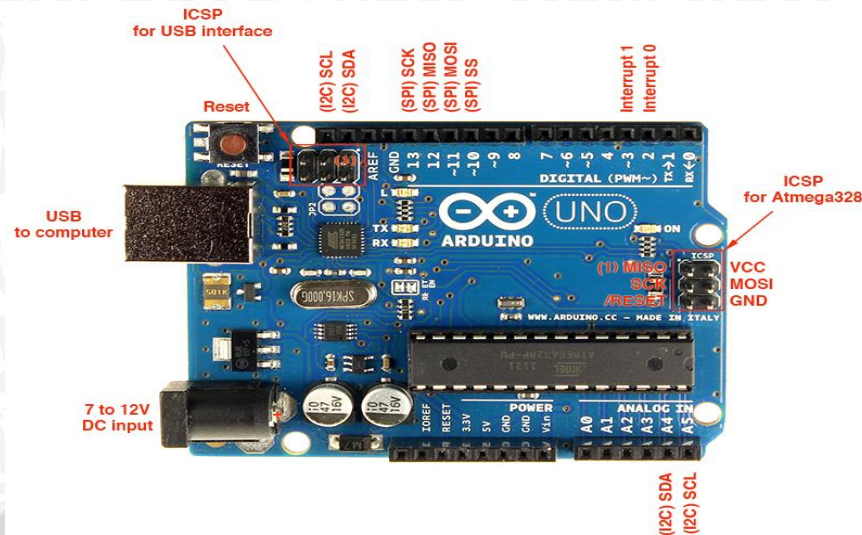
No	Pin	Fungsi
1	RPWM	Forward level atau input sinyal PWM, aktif <i>HIGH</i>
2	LPWM	Reverse level atau input sinyal PWM, aktif <i>LOW</i>
3	R_EN	Forward drive enable input, HIGH enable, LOW close
4	L_EN	Reverse drive enable input, HIGH enable, LOW close
5	R_IS	Forward drive-side current alarm output
6	L_IS	Reverse drive-side current alarm output
7	VCC	5V power input dari board arduino
8	GND	Ground

Sumber: *datasheet High Current 30 A DC Motor Driver*

2.4 Mikrokontroler Arduino Uno R3

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai *input* dan *output* serta kontrol dengan program yang bisa ditulis dan dihapus. Cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis.

Arduino UNO (Gambar 2.7) adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



Gambar 2. 7 Skema Arduino Uno

Sumber: electroschematics.com

2.5 Sensor Tegangan

Sensor tegangan (Gambar 2.8) digunakan untuk mengukur tegangan dengan range tertentu. Catu daya yang digunakan adalah 5V dan bisa mendeteksi tegangan DC dengan range 0-25 V. Sensor ini memiliki resolusi analog sebesar 0.00489V. Dengan input konektor terminal katoda dihubungkan ke V_{cc} dan anoda ke Ground (GND).

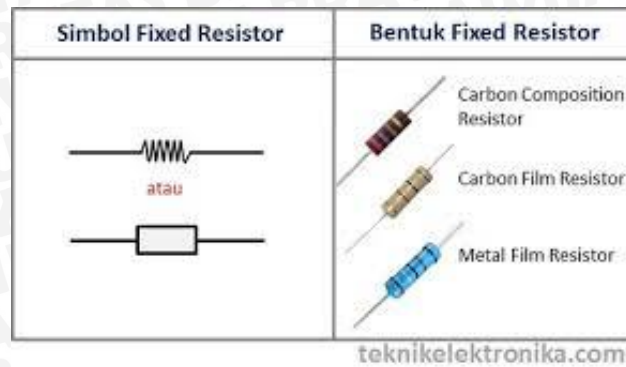


Gambar 2. 8 Sensor tegangan

Sumber: www.emartee.com

2.6 Beban Resistif

Elemen ini menerima energi dengan cara menyerap sehingga menimbulkan panas. Sering juga disebut dengan tahanan, hambatan, penghantar, atau resistansi di mana resistor mempunyai fungsi sebagai penghambat arus, pembagi arus, dan pembagi tegangan. Nilai resistor tergantung dari hambatan jenis bahan resistor (yang tergantung dari bahan pembuatnya), panjang resistor, dan luas penampang resistor itu sendiri (Ramdhani, M., 2008). Simbol resistor dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Simbol resistor
Sumber: teknikelektronika.com

Secara matematis nilai resistor (R) adalah:

$$R = \rho \frac{l}{A} \quad (2-10)$$

dengan:

- R = resistor (ohm)
- ρ = hambatan jenis
- l = panjang dari resistor
- A = luas penampang

Jika suatu resistor dilewati oleh sebuah arus (I), maka pada kedua ujung resistor tersebut akan timbul beda potensial atau tegangan, yang sesuai dengan Hukum Ohm, yaitu:

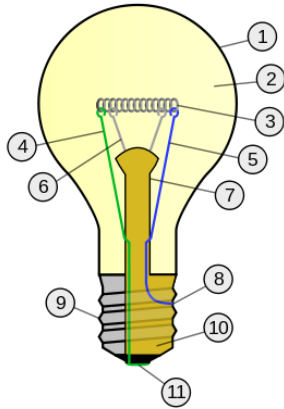
$$V_R = I R \quad (2-11)$$

dengan:

- V_R = tegangan di resistor (Volt)
- I = arus (Ampere)
- R = resistor (Ohm)

2.7 Lampu Pijar (bohlam)

Lampu pijar (bohlam) adalah sumber cahaya buatan yang dihasilkan melalui penyaluran arus listrik melalui filamen yang kemudian memanaskan dan menghasilkan cahaya. Komponen utama lampu pijar adalah bola lampu yang terbuat dari kaca, filamen yang terbuat dari wolfram, dasar lampu yang terdiri atas filamen, bola lampu, gas pengisi, dan kaki lampu (Gambar 2.10). Pada dasarnya filamen pada lampu pijar merupakan beban resistif.

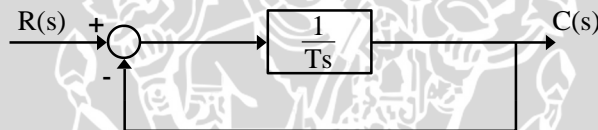


1. Bola lampu
2. Gas bertekanan rendah (argon, neon, nitrogen)
3. Filamen wolfram
4. Kawat penghubung ke kaki tengah
5. Kawat penghubung ke ulir
6. Kawat penyangga
7. Kaca penyangga
8. Kontak listrik di ulir
9. Sekrup ulir
10. Isolator
11. Kontak listrik di kaki tengah

Gambar 2. 10 Lampu Pijar
Sumber: Wikipedia, 2016

2.8 Output Sistem Orde Satu

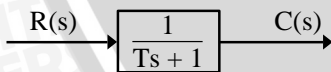
Diagram blok sistem orde satu dapat dilihat dalam Gambar 2.11 yang mempunyai fungsi alih sebagai berikut



Gambar 2. 11 Sistem orde satu

$$\frac{C(s)}{R(s)} = \frac{1/Ts}{1 + 1/Ts} \quad (2-12)$$

$$= \frac{1}{Ts + 1}$$



2.8.1 Output Unit Step Pada Sistem Orde Satu

Jika sistem orde satu diberi masukan unit step $r(t) = 1$, dari Transformasi Laplace fungsi ramp adalah $R(s) = 1/s$. Keluaran sistem adalah sebagai berikut

$$C(s) = \frac{1}{Ts + 1} R(s)$$

$$C(s) = \frac{1}{Ts+1} \frac{1}{s}$$

$$= \frac{1}{s(Ts+1)} \quad (2-13)$$

$$c(t) = \mathcal{L}^{-1}[C(s)]$$

$$c(t) = 1 - e^{-t/T} \quad (t \geq 0) \quad (2-14)$$

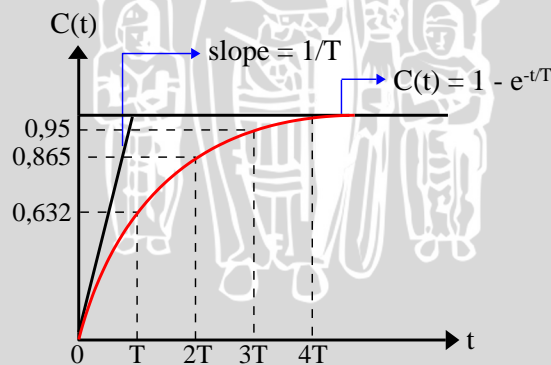
Persamaan keluaran tersebut menyatakan bahwa keluaran $c(t)$ mula-mula nol kemudian akhirnya menjadi satu (dapat dilihat dalam Gambar 2.12). Salah satu karakteristik penting output eksponensial $c(t)$ tersebut adalah bahwa pada $t = T$, maka $c(t) = 1 - e^{-1} \cong 0.632 = \frac{2}{3}$

$$T = \text{time constant / konstanta waktu sistem}$$

$$= \frac{2}{3} \text{ harga akhir}$$

Konstanta waktu T yang lebih kecil akan mempercepat output sistem. Karakteristik penting lainnya pada kurva output eksponensial adalah kemiringan garis singgung / gradien pada $t = 0$ adalah $1/T$, karena

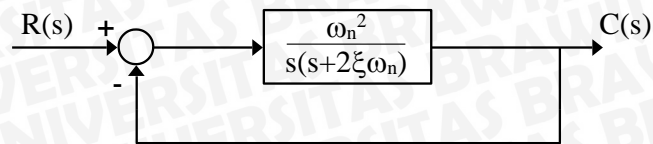
$$\frac{dc}{dt} = \frac{1}{T} e^{-t/T} \Big|_{t=0} = \frac{1}{T}$$



Gambar 2. 12 Output unit step sistem orde satu

2.9 Output Sistem Orde Dua

Diagram blok sistem orde dua (lihat Gambar 2.13), dengan fungsi alihnya adalah sebagai berikut:



Gambar 2. 13 Sistem orde dua

$$\begin{aligned} \frac{C(s)}{R(s)} &= \frac{\omega_n^2}{s(s+2\xi\omega_n)} \\ &= \frac{\omega_n^2}{1 + \frac{\omega_n^2}{s(s+2\xi\omega_n)}} \\ &= \frac{\omega_n^2}{s(s+2\xi\omega_n) + \omega_n^2} \\ &= \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \end{aligned} \quad (2-15)$$

Akar-akar penyebut fungsi alih atau persamaan karakteristik adalah

$$\begin{aligned} s_{1,2} &= \frac{-2\xi\omega_n \pm \sqrt{(2\xi\omega_n)^2 - 4\omega_n^2}}{2} \\ s_{1,2} &= -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1} \\ &= -\xi\omega_n \pm j\omega_n\sqrt{1 - \xi^2} \\ s_{1,2} &= -\xi\omega_n \pm j\omega_d \end{aligned} \quad (2-16)$$

dimana

- ξ = rasio peredaman sistem (*damping ratio*)
- ω_n = frekuensi natural/alamiah tak teredam
- ω_d = frekuensi natural/alamiah teredam

Kelakuan dinamik sistem orde dua dapat digambarkan dalam suku dua parameter ξ dan ω_n . Jika ($0 < \xi < 1$), maka pole loop tertutup merupakan konjugat kompleks dan berada pada bidang s sebelah kiri. Dalam hal ini, sistem dikatakan dalam peredaman dan tanggapan peralihan berosilasi. Jika ($\xi = 1$), maka sistem dikatakan teredam kritis. Sistem terlalu teredam berhubungan dengan ($\xi > 1$). Tanggapan peralihan sistem teredam kritis dan sistem terlalu teredam tidak berosilasi. Jika $\xi = 0$, tanggapan peralihan tidak muncul.

Pada sistem orde dua seperti terlihat dalam Gambar 2.13, berdasarkan output sistem dengan masukan unit step akan terdapat tiga keadaan yang berbeda yaitu keadaan teredam ($0 < \xi < 1$), teredam kritis ($\xi = 1$), dan sistem terlalu teredam ($\xi > 1$).

2.9.1 Keadaan Kurang Teredam / Underdamped ($0 < \xi < 1$)

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \quad (2-17)$$

Jika sistem diberi input berupa unit step atau $R(s) = \frac{1}{s}$, maka:

$$C(s) = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \quad (2-18)$$

Dari Tabel Transformasi Laplace didapatkan

$$c(t) = 1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin(\omega_n \sqrt{1-\xi^2} t + \phi)$$

$$\phi = \arctan \frac{\sqrt{1-\xi^2}}{\xi}$$

Jika $\omega_d = \omega_n \sqrt{1-\xi^2}$; maka

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_d t + \arctan \frac{\sqrt{1-\xi^2}}{\xi}\right) \quad t \geq 0 \quad (2-19)$$

Output sistem tersebut juga bisa diperoleh dengan menggunakan Transformasi Laplace balik jika $C(s)$ ditulis dalam bentuk berikut:

$$C(s) = \frac{1}{s} - \frac{s + 2\xi\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2-20)$$

$$= \frac{1}{s} - \frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2} - \frac{\xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}$$

$$\mathcal{L}^{-1}\left[\frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}\right] = e^{-\xi\omega_n t} \cos \omega_d t$$

$$\mathcal{L}^{-1}\left[\frac{\xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}\right] = e^{-\xi\omega_n t} \sin \omega_d t \quad (2-20)$$

oleh karena itu, transformasi laplace balik dari persamaan

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

diperoleh sebagai

$$\mathcal{L}^{-1}[C(s)] = c(t)$$

$$c(t) = 1 - e^{-\xi\omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right)$$

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin \left(\omega_d t + \arctan \frac{\sqrt{1-\xi^2}}{\xi} \right) \quad t \geq 0 \quad (2-21)$$

Sinyal kesalahan / error adalah $e(t) = r(t) - c(t)$, dimana

$$r(t) = 1$$

dan

$$c(t) = 1 - e^{-\xi\omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right)$$

sehingga

$$e(t) = e^{-\xi\omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right) \quad t \geq 0$$

$$\text{jika } \xi = 0 \Rightarrow c(t) = 1 - \cos \omega_n t$$

2.9.2 Teredam Kritis / Critically Damped ($\xi=1$)

Dalam hal ini apabila dua pole $C(s)/R(s)$ hampir sama, maka sistem dapat didekati dengan bentuk teredam kritis. Jika input berupa unit step dimana $R(s) = 1/s$ maka:

$$C(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad (2-22)$$

$$c(t) = 1 - e^{-\omega_n t} (1 + \omega_n t) \quad t \geq 0$$

2.9.3 Terlalu Teredam / Overdamped ($\xi > 1$)

Dalam hal ini pole $C(s)/R(s)$ adalah bilangan nyata / real negatif yang tidak sama.

Jika input berupa unit step dimana $R(s) = 1/s$ dan $C(s)$ dapat ditulis dengan :

$$C(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad (2-23)$$

$$C(s) = \frac{\omega_n^2}{(s + \xi\omega_n + \omega_n\sqrt{\xi^2 - 1})(s + \xi\omega_n - \omega_n\sqrt{\xi^2 - 1})}$$

$$c(t) = 1 + \frac{\omega_n}{2\sqrt{\xi^2 - 1}} \left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2} \right) \quad t \geq 0$$

$$\text{dengan } s_1 = (\xi + \sqrt{\xi^2 - 1})\omega_n$$

$$s_2 = (\xi - \sqrt{\xi^2 - 1})\omega_n \quad (2-24)$$

Tanggapan $c(t)$ terdiri dari dua suku eksponensial menurun.

2.10 Tanggapan Peralihan

Sistem dengan tenaga tidak dapat memberikan tanggapan seketika dan akan menunjukkan tanggapan peralihan walaupun diberi masukan ataupun gangguan. Karakteristik unjuk kerja sistem kontrol yang diinginkan dicirikan oleh suku tanggapan peralihan terhadap masukan unit step karena hal itu mudah dilakukan dan cukup drastis. Jika tanggapan terhadap masukan unit step diketahui, secara matematis dapat dihitung tanggapan untuk masukan yang lain.

Tanggapan peralihan sistem kontrol selalu menunjukkan osilasi teredam sebelum mencapai keadaan mantapnya, hal ini juga menunjukkan bahwa sistem tersebut mempunyai rasio peredaman ($0 < \xi < 1$) yang juga berarti bahwa sistem tersebut merupakan sistem yang kurang teredam / underdamped.

Tanggapan peralihan sistem kontrol terhadap masukan unit step umumnya dikelompokkan sebagai berikut (lihat Gambar 2.14):

- 1) Delay Time / Waktu Tunda, t_d

Waktu yang dibutuhkan oleh outputs untuk mencapai setengah harga akhir pada saat lonjakan pertama

- 2) Rise Time / Waktu Naik, t_r

Waktu yang dibutuhkan oleh outputs agar bertambah dari 10% menjadi 90% dari nilai akhir

- 3) Peak Time / Waktu Puncak, t_p

Waktu yang dibutuhkan oleh outputs untuk mencapai puncak pertama lonjakan (maksimum)

- 4) Maximum Overshoot / Lonjakan Maksimum, M_p

Merupakan nilai puncak kurva outputs diukur dari satu

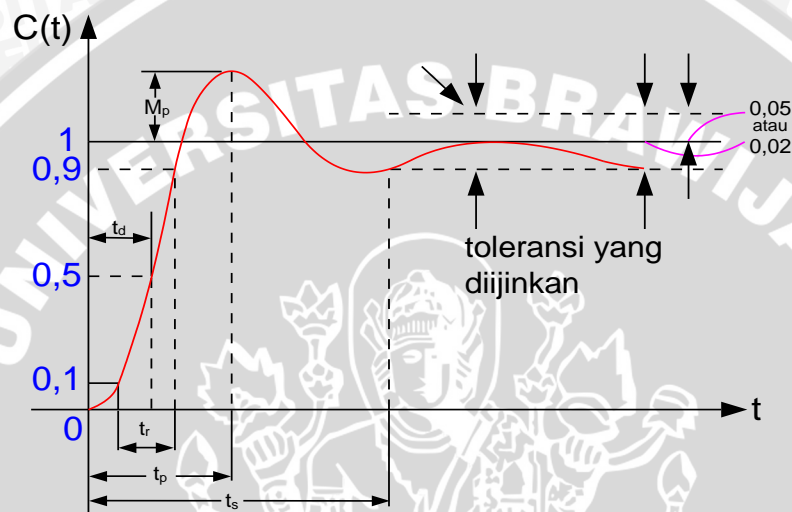
$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

dengan $c(t_p)$ = nilai outputs pada saat lonjakan maksimum.

$c(\infty)$ = nilai outputs pada saat keadaan mantap.

5) Settling Time / Waktu Turun, t_s

Waktu yang dibutuhkan oleh outputs untuk mencapai harga tertentu dan tetap dalam range nilai akhir (biasanya 5% atau 2%)



Gambar 2. 14 Output Unit Step Sistem Orde Dua

2.11 Kontroler

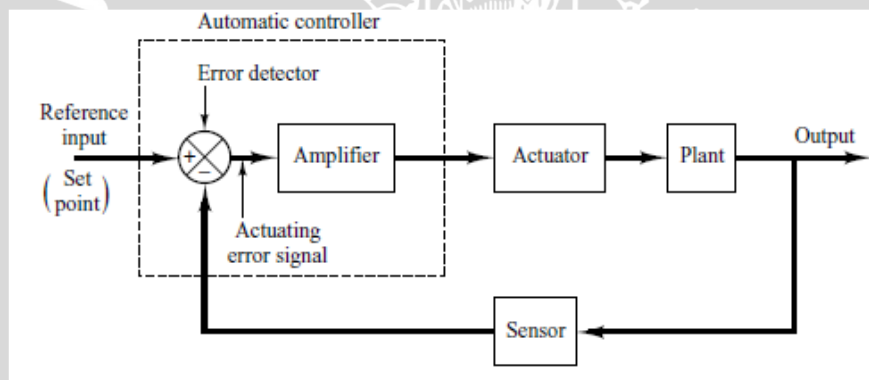
Kontroler seringkali juga disebut dengan istilah kompensator atau pengontrol. Kontroler adalah suatu sistem dinamis yang sengaja ditambahkan untuk mendapatkan karakteristik sistem keseluruhan yang diinginkan (Ogata K., 2010). Fungsi kontroler pada umumnya adalah sebagai berikut:

- 1) Membandingkan nilai *input* dan *output* sistem secara keseluruhan (*plant*).
- 2) Menentukan penyimpangan (*error*).
- 3) Menghasilkan sinyal kontrol (mengurangi penyimpangan (*error*) menjadi nilai nol/ nilai yang kecil).

Adapun tujuan kontrol secara khusus adalah sebagai berikut:

- 1) Meminimumkan error steady state.
- 2) Meminimumkan *settling time*.
- 3) Mencapai spesifikasi transien yang lain, misalnya meminimumkan *maximum overshoot*.

Sistem loop tertutup (Gambar 2.15) menggunakan sinyal *output* yang diumpanbalikkan terhadap *automatic controller* (kontroler otomatis) (Ogata, K., 2010), yang akan membuat perubahan terhadap sistem agar *output* sistem seperti yang diinginkan atau sesuai *set point*. Sensor/transduser digunakan sebagai elemen yang langsung mengadakan kontak dengan objek yang diukur. Transduser berfungsi untuk mengubah besaran fisis yang diukur menjadi besaran fisis lainnya, seperti mengubah besaran tekanan, temperatur, aliran, posisi menjadi besaran listrik. *Actuating error signal* merupakan sinyal kesalahan (error) yang merupakan selisih antara sinyal set point dan sinyal output. *Actuator* (aktuator) berfungsi untuk mengontrol aliran energi ke sistem yang dikontrol. Sebagai contoh adalah motor listrik, katub pengontrol, pompa dan sebagainya. *Amplifier* merupakan unit yang dibutuhkan karena daya dari *error detector* tidak cukup kuat untuk menggerakkan elemen *output*. Karena fungsi pengendalian adalah untuk mengendalikan *output* agar kesalahan (*error*) mendekati nol, maka diperlukan penguat daya (*power amplifier*).



Gambar 2. 15 Diagram Blok Sistem dengan Kontroler Otomatis
Sumber: Ogata, K. (2010)

Cara bagaimana kontroler otomatis menghasilkan sinyal kontrol disebut dengan aksi kontrol. Aksi kontrol dasar yang sering digunakan dalam kontroler adalah

1. Kontroler proporsional (P)
2. Kontroler integral (I)
3. Kontroler proporsional integral (PI)
4. Kontroler proporsional integral diferensial (PID)

2.11.1 Kontroler Proporsional (P)

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat output. Hubungan antara output kontroler $u(t)$ dan sinyal *error* $e(t)$ ditunjukkan dalam persamaan berikut:

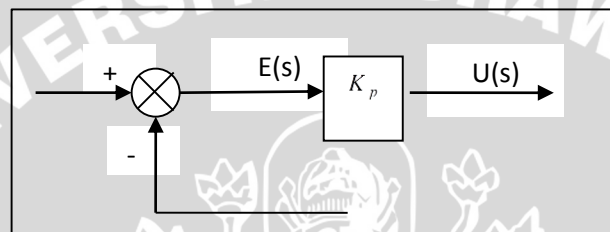
$$u(t) = K_p e(t) \quad (2-25)$$

atau, dalam fungsi alih

$$\frac{U(s)}{E(s)} = K_p$$

dimana K_p adalah penguatan.

Diagram blok kontroler proporsional (P) ditunjukkan dalam Gambar 2.16.



Gambar 2. 16 Diagram blok kontroler proporsional (P)
Sumber: Ogata, K. (2010)

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K., 2010).

2.11.2 Kontroler Integral (I)

Kontroler integral (I) memiliki kemampuan untuk mengurangi *offset* yang diakibatkan oleh kontroler proporsional. Output kontroler $u(t)$ diubah dengan laju yang sebanding dengan *error* $e(t)$. Persamaan kontroler integral (I) ditunjukkan dalam persamaan berikut (Ogata K., 2010).

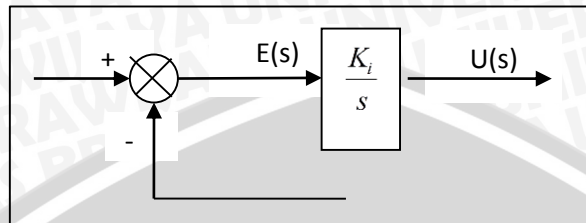
$$\frac{du(t)}{dt} = K_i e(t)$$

$$u(t) = K_i \int_0^t e(t) dt$$

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

(2-26)

yang merupakan fungsi alih kontroler integral (I), dengan K_i adalah konstanta integral yang dapat diubah nilainya. Jika $e(t)$ bernilai nol, maka nilai $m(t)$ tetap konstan. Aksi kontrol integral biasa disebut dengan kontrol reset. Gambar 2.17 menunjukkan diagram blok kontroler integral (I).



Gambar 2. 17 Diagram blok kontroler integral (I)
Sumber: Ogata, K. (2010)

2.11.3 Kontroler Proporsional Integral (PI)

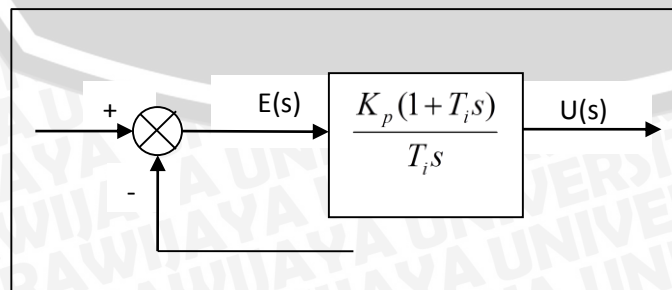
Kontroler proporsional integral (PI) memiliki kemampuan untuk mempercepat output dan mengurangi *offset*. Persamaan kontroler proporsional integral (PI) adalah

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2-27)$$

Adapun fungsinya adalah

$$\begin{aligned} \frac{U(s)}{E(s)} &= K_p \left(1 + \frac{1}{T_i s} \right) \\ &= \frac{K_p (1 + T_i s)}{T_i s} \end{aligned}$$

dengan K_p penguatan proporsional dan T_i disebut waktu integral, yang keduanya dapat ditentukan. Waktu integral mengatur aksi kontrol internal sedangkan perubahan nilai K_p berakibat pada bagian aksi kontrol proporsional maupun integral. Gambar 2.18 menunjukkan diagram blok kontroler proporsional integral (PI).



Gambar 2. 18 Diagram blok kontroler proporsional integral (PI)
Sumber: Ogata, K. (2010)

2.11.4 Kontroler Proporsional Integral Diferensial (PID)

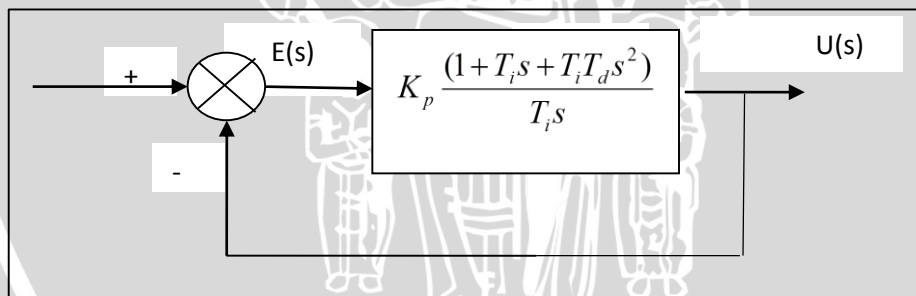
Gabungan aksi kontrol proporsional, integral, dan differensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga aksi kontrol tersebut. Masing-masing kontroler P, I, maupun D berfungsi untuk mempercepat output sistem, menghilangkan *offset*, dan mendapatkan energi ekstra ketika terjadi perubahan beban (mempercepat *settling time*). Aksi kontroler proporsional integral diferensial (PID) didefinisikan dengan persamaan

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2-28)$$

Sedangkan fungsi alihnya adalah

$$\begin{aligned} \frac{U(s)}{E(s)} &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\ &= K_p \frac{(1 + T_i s + T_i T_d s^2)}{T_i s} \end{aligned}$$

dimana K_p adalah penguatan proporsional, T_i adalah waktu integral dan T_d adalah waktu diferensial. Gambar 2.19 menunjukkan diagram blok kontroler proporsional integral diferensial (PID).



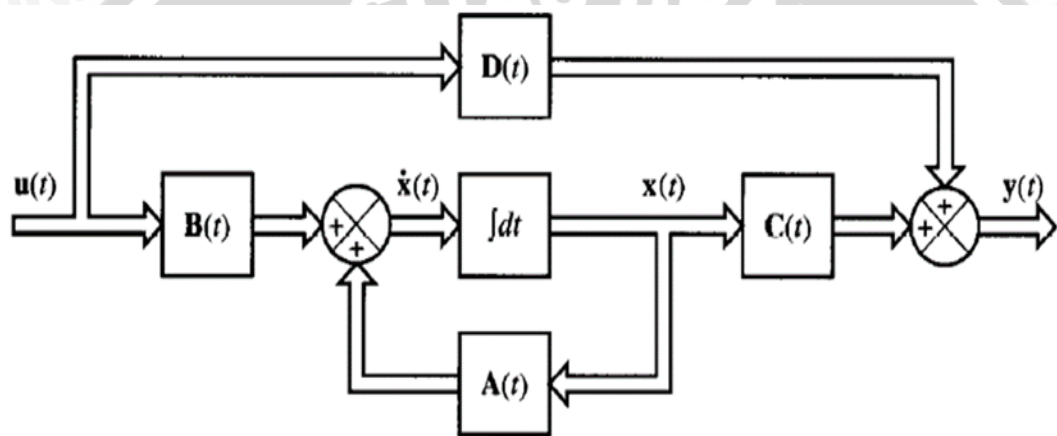
Gambar 2. 19 Diagram blok kontroler proporsional integral diferensial (PID)

Sumber: Ogata, K. (2010)

Kombinasi ini mempunyai keuntungan dibandingkan masing-masing kontroler, biasanya dengan kontroler ini didapatkan overshoot yang rendah, cepat mencapai *steady state* (keadaan mantap) dan *error steady state* (kesalahan keadaan mantap) yang kecil bahkan nol. Pada perancangan sistem kontrol, seringkali kita harus melakukan *trial and error*. Contohnya pada desain PID, kita harus melakukan *trial and error* kepada 3 variabel yang berbeda (P, I, dan D). Desain PID bisa juga dilakukan dengan menggunakan metode Ziegler-Nichols.

2.12 Konsep Ruang Keadaan (*State Space*)

Proses pemodelan sistem akan selalu dilakukan dalam mendesain sistem kontrol. Model dapat berupa diagram blok, diagram aliran sinyal, maupun dalam bentuk persamaan matematis. Model matematis sebuah sistem didefinisikan sebagai kumpulan informasi dalam bentuk persamaan matematis yang mewakili sistem fisik yang ditinjau. Terdapat suatu metode untuk memodelkan sistem dengan multi *input* atau *multi output*, yang dikenal dengan metode ruang keadaan (*state space*). Model ruang keadaan merupakan sekumpulan persamaan diferensial orde satu yang berhubungan dan ditulis dalam notasi matriks vektor. Bentuk umum ruang keadaan (*state space*) sebuah sistem dapat dilihat dari representasi sistem dalam bentuk persamaan *state* dalam Gambar 2.20.



Gambar 2. 20 Representasi sistem dalam bentuk persamaan state

Sumber: Ogata, K. (2010)

Persamaan *state*:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2-29)$$

Persamaan *output*:

$$y(t) = Cx(t) + Du(t) \quad (2-30)$$

dengan:

$x(t)$ = vektor state (n-vektor)

$u(t)$ = vektor kontrol (m-vektor)

$y(t)$ = vektor *output* (p-vektor)

A = matriks keadaan orde $n \times n$

B = matriks *input* orde $n \times m$

C = matriks *output* orde $p \times n$

D = matriks transmisi langsung orde $p \times n$

2.13 Konsep Sistem Kontrol Optimal

Sistem kontrol optimal adalah sistem kontrol yang mempunyai unjuk kerja terbaik (*best performance*) terhadap suatu acuan tertentu. Sistem kontrol mempunyai tolok ukur, artinya sistem tersebut sesuai terhadap suatu referensi (Houpis D'Azzo, 1995). Supaya suatu sistem dapat dikontrol, maka perlu dibuat model matematis yang menghubungkan *input* dan *output*. Pada sistem kontrol optimal, model yang banyak digunakan adalah persamaan keadaan (*state space*).

Cara untuk mengukur kualitas output transien pada sistem kontrol yang diberi *input step* yaitu dengan *control area*. Performansi sistem terbaik ditandai harga *output* $y(t)$ mendekati *set point (input)* $u(t)$ sehingga luasan *error* $e(t) = u(t) - y(t)$ mendekati nol. Selain itu, waktu yang dibutuhkan oleh *output* untuk mencapai kondisi *steady state* secepat mungkin.

Cara paling sederhana untuk mencapai kondisi “*steady state*” secepat mungkin adalah dengan memperbesar parameter kontroler proporsional (P). Sedangkan untuk mengurangi *error* adalah dengan memperbesar parameter kontroler integral (I). Namun kedua cara tersebut akan dapat mengakibatkan ketidakstabilan sistem dan mengurangi nilai redaman (ξ). Tujuan tersebut tidak dapat dicapai secara bersama-sama. Oleh karena itu diperlukan konsep kompromi atau pengambilan “jalan tengah” dari kedua tujuan itu. “Jalan Tengah” itu dikenal sebagai optimasi. Syarat proses optimasi adalah penetapan fungsi tujuan. Sinyal yang mungkin dapat digunakan untuk penggambaran itu adalah sinyal *error* (Rusli, M., 2015).

2.14 Pole Placement

Pole placement adalah suatu metode desain kontrol dimana menentukan letak dari *pole* sistem loop tertutup pada bidang kompleks dengan mengatur *gain kontroler*. Pada pendekatan secara konvensional, kita mengasumsikan bahwa pengaruh respon *pole* loop tertutup tak dominan dapat diabaikan. Pada pendekatan dengan penempatan *pole (pole placement)*, pendekatan penempatan *pole* yang diberikan menetapkan semua loop tertutup. Terdapat persyaratan pada bagian sistem untuk *pole-pole* loop tertutup yang akan ditempatkan pada lokasi yang dipilih secara sembarang yaitu bahwa sistem harus berupa keadaan lengkap yang dapat dikontrol.

Keterkontrolan keadaan secara sempurna dapat dilihat dari sistem waktu diskrit.

Sistem diskrit yang dinyatakan oleh “ $x((k + 1)T) = Gx(kT) + Hu(kT)$ “

Dimana :

$x(kT)$ = vektor keadaan (vektor n dimensi)

$u(kT)$ = sinyal kontrol

G = matriks nonsingular n x n

H = matriks n x 1

T = perioda cacah

Kondisi keterkontrolan sempurna diturunkan berdasarkan kenyataan bahwa jika suatu sistem terkontrol sempurna, maka ada sinyal kontrol kontinu sepotong-sepotong yang akan memindahkan setiap keadaan awal ke titik asal dalam sejumlah perioda cacah yang terhingga.

Observability atau keteramatan sistem linier dapat dinyatakan oleh persamaan berikut:

$$\dot{x}(t) = Ax(t)$$

$$y = Cx(t)$$

dimana :

x = vektor keadaan (vektor n-dimensi)

y = vektor keluaran (vektor m-dimensi)

A = matriks n x n

C = matriks m x n

Keteramatan sempurna dari sistem waktu diskrit. Tinjau sistem yang dinyatakan oleh :

$$x((k+1)T) = Gx(kT)$$

$$y(kT) = Cx(kT)$$

dimana:

$x(kT)$ = vektor keadaan (vektor n-dimensi)

$y(kT)$ = vektor keluaran (vektor m-dimensi)

G = matriks n x n

C = matriks m x n

T = periode cacah

Persamaan ruang keadaan waktu berbentuk :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2-31)$$

Dan sinyal kontrol diberikan oleh :

$$u = -kx \quad (2-32)$$

Dan substitusi persamaan (2-32) ke persamaan (2-31) sehingga didapatkan

$$\dot{x}(t) = (A - BK)x(t) \quad (2-33)$$

Solusi dari persamaan (2-33)

$$x(t) = e^{(A-BK)t}x(0)$$

Kestabilan dari sistem (2-33) ditentukan oleh nilai eigen dari $(A-BK)$, artinya jika matriks K dapat dipilih secara tepat, maka bagian riil dari nilai eigen matriks $(A-BK)$

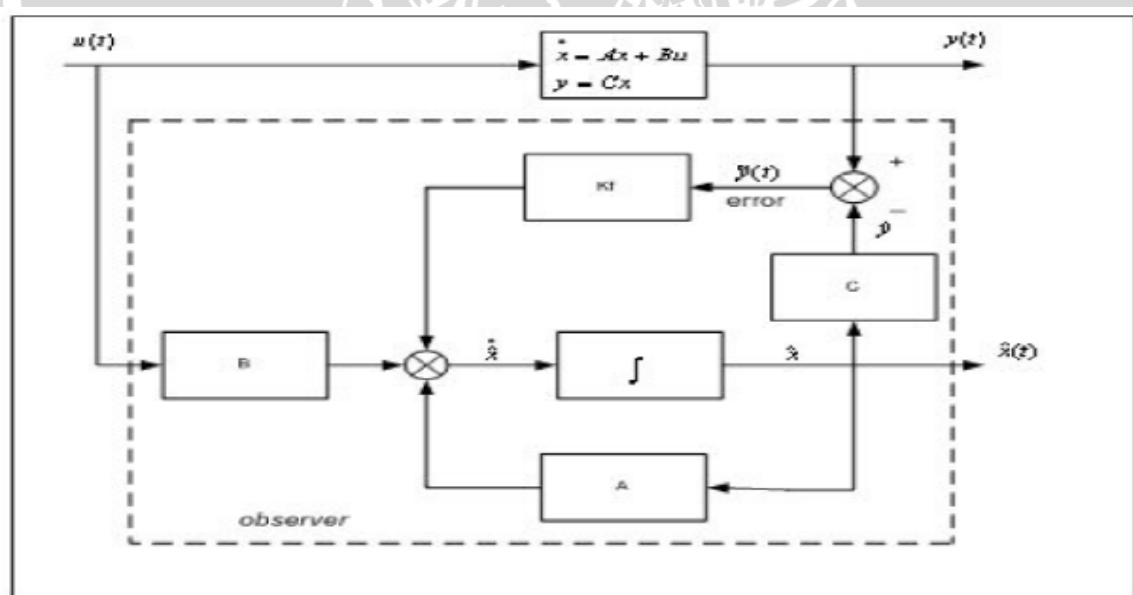
terletak di sebelah kiri sumbu imajiner bidang s , hal itu berarti, untuk semua $x(0) \neq 0$, $x(t)=0$ untuk $t \rightarrow \infty$. Syarat perlu dan cukup untuk penempatan pole dinyatakan dalam teorema berikut. :

Teorema [Ogata Katsuhiko, 1997]: “Jika diberikan suatu sistem, maka syarat perlu dan cukup untuk penempatan sembarang pole yang diinginkan adalah bahwa sistem tersebut terkontrol secara lengkap”.

2.15 State Observer

Pada sistem dengan banyak state, adalah tidak mungkin untuk mengetahui seluruh informasi mengenai state. Tidak semua state dapat diindera oleh sensor, dan dapat diumpanbalikkan. Sistem dengan *state feedback* membutuhkan sensor ideal yang mempunyai *bandwidth* tak terhingga. Kenyataannya tidak ada sensor yang ideal dan juga sensor memiliki *bandwidth* yang terbatas.

Konsep pengamat (*observer*) sebenarnya adalah bila tidak memiliki seluruh state, adalah mungkin untuk mengestimasi state-state tersebut dengan menggunakan *input* dan *output* sistem. Konsepnya adalah seperti dalam Gambar 2.21.



Gambar 2. 21 Full order state observer

Sumber: Gopal, M. (1987)

Untuk sistem,

$$\dot{x}(t) = Ax(t) + Bu(t); x(0) = x^0 \quad (2-34)$$

$$y = Cx \quad (2-35)$$

Dengan estimasi state $x(t)$ dinotasikan oleh

$$\hat{\dot{x}}(t) = A\hat{x} + Bu + K_f(y - C\hat{x}) \quad (2-36)$$

$$\hat{y} = \hat{C}x \quad (2-36)$$

Nilai $\tilde{y} = y - \hat{y}$ diusahakan bernilai sekecil mungkin, yang menandakan bahwa estimator bekerja dengan baik. Untuk membuat $\hat{x}(t)$ sedekat mungkin dengan $x(t)$, didefinisikan *state estimator error*:

$$\tilde{x} = x - \hat{x}$$

$$\tilde{\dot{x}} = \dot{x} - \hat{\dot{x}}$$

$$\tilde{\dot{x}} = Ax + Bu - A\hat{x} - Bu - K_f C(x - \hat{x})$$

$$\tilde{\dot{x}} = A(x - \hat{x}) - K_f C(x - \hat{x})$$

$$\tilde{\dot{x}} = (A - K_f C) \tilde{x} \quad (2-37)$$

Jika dapat dicari nilai matriks *observer gain* K_f sedemikian rupa sehingga $(A - K_f C)$ stabil, maka nilai *estimator error* akan menuju nol untuk sembarang nilai awal. Hal ini dapat dilakukan hanya jika pasangan matriks (A, C) *observable* (Lewis and Syrmos, 1995:450).

Dengan:

$$K_f^T = R^{-1}CP$$

Atau

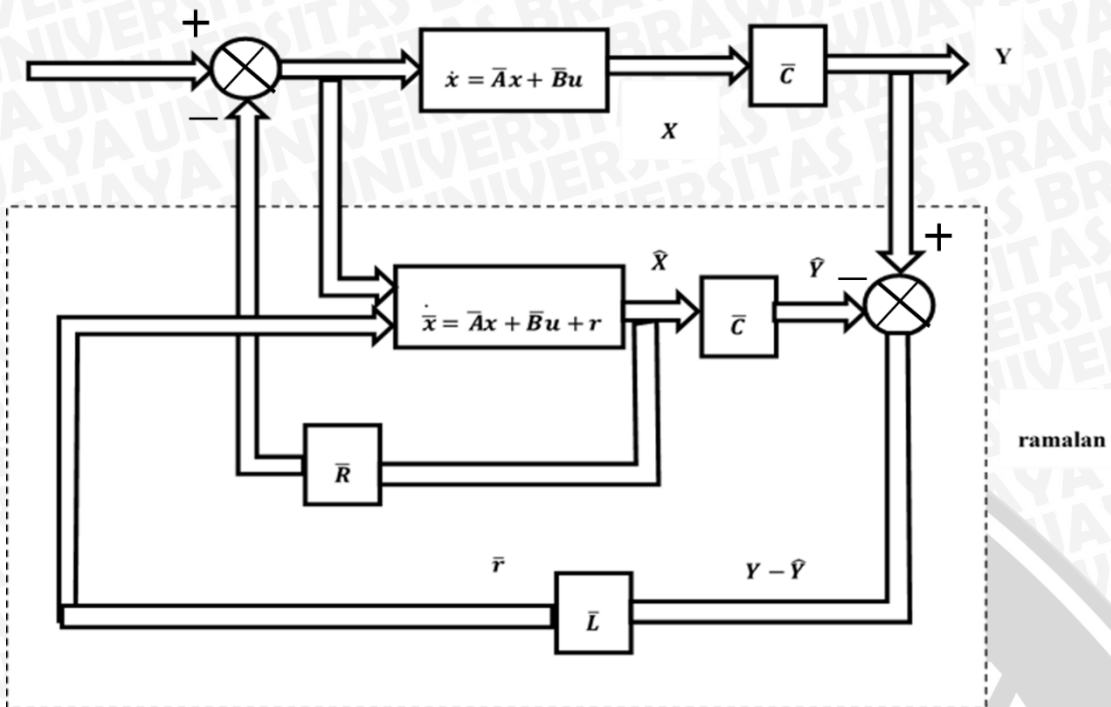
$$K_f^T = PC^T R^{-1}$$

Dimana P adalah solusi dari persamaan Riccati:

$$AP + PA^T + Q - PC^T R^{-1} CP = 0 \quad (2-38)$$

2.16 Output Feedback Control

Salah satu desain kontroler adalah *Output Feedback Control* (OFC). Dengan menggunakan *OFC*, dapat diestimasi *state-state* yang ada hanya dengan menggunakan *input* dan *output* sistem. Salah satu metode yang digunakan pada *OFC* adalah dengan menggunakan estimasi Leunberger. Estimasi tersebut dapat dilihat dalam Gambar 2.22.



Gambar 2. 22 Struktur Output Feedback Control
Sumber: Kwakernaak (2011)

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + r$$

$$\bar{r} = \bar{L}(Y - \hat{Y})$$

$$\hat{Y} = \bar{C}\hat{x}$$

$$Y = \bar{C}x$$

distubtitusi:

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + \bar{L}(Y - \hat{Y})$$

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + \bar{L}(\bar{C}x - \bar{C}\hat{x})$$

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + \bar{L}(x - \hat{x})$$

$(x - \hat{x}) = \tilde{x}$ dinamakan selisih error

$$\dot{\tilde{x}} = (\bar{A} - \bar{L}\bar{C})\tilde{x}$$

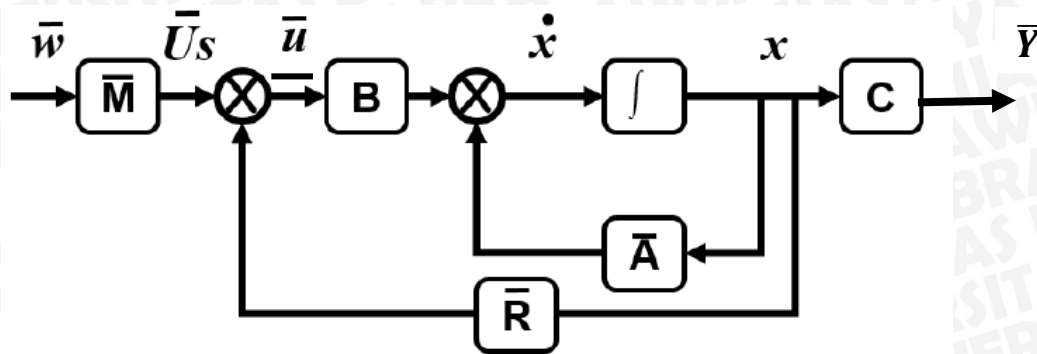
untuk mencari matriks L menggunakan rumus:

$$L = (-S^{-1}CP)^T \quad (2-39)$$

2.17 Desain Pra Filter

Pra filter digunakan untuk menunjang performa *output feedback control* (OFC).

Desain pra filter ditunjukkan dalam Gambar 2.23.



Gambar 2.23 Desain pra filter

Dimana:

$$\dot{x} = \bar{A}x + \bar{B}u$$

$$\bar{Y} = Cx$$

$$\bar{u} = -\bar{R}x + \bar{U}s$$

$$\bar{U}s = \bar{M}w$$

disubstitusi:

$$\dot{x} = \bar{A}x + \bar{B}(-\bar{R}x + \bar{U}s)$$

$$\dot{x} = (\bar{A} - \bar{B}\bar{R})x + \bar{B}\bar{U}s$$

ditransformasi Laplace:

$$s\bar{X}(s) = (\bar{A} - \bar{B}\bar{R})\bar{X}(s) + \bar{B}\bar{U}s(s)$$

$$\bar{X}(s) = [sI - (\bar{A} - \bar{B}\bar{R})]^{-1}\bar{B}\bar{U}s(s)$$

$$\bar{Y}(s) = C\bar{X}(s)$$

$$\bar{Y}(s) = C[sI - (\bar{A} - \bar{B}\bar{R})]^{-1}\bar{B}\bar{U}s(s)$$

$$\bar{M} = [-C(\bar{A} - \bar{B}\bar{R})^{-1}\bar{B}]^{-1} \quad (2-40)$$



BAB III

METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Dalam menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

1. Perancangan blok diagram sistem
2. Spesifikasi desain
3. Karakterisasi setiap blok

Karakterisasi setiap blok dilakukan untuk mempermudah analisis sistem.

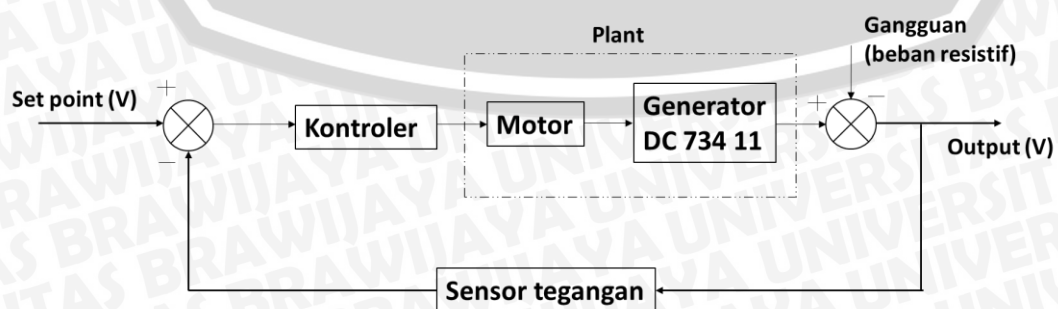
Karakterisasi dibagi menjadi beberapa bagian, yaitu:

1. Karakterisasi motor DC D-6759.
2. Karakterisasi driver motor (*High Current 30 A DC Motor Driver*)
3. Karakterisasi sensor tegangan.
4. Karakterisasi *plant* (motor generator DC)
4. Pembuatan perangkat keras
5. Perancangan algoritma.

Perancangan algoritma meliputi algoritma kontrol optimal dan *flowchart* program.

3.1 Perancangan Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan diagram blok sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan yang didesain dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Blok Diagram Sistem Loop Tertutup

Plant = Motor generator DC 734 11

Gangguan = Beban resistif berupa lampu bohlam 2,5 Volt pada output generator DC.

Input dan Output = Tegangan

3.2 Spesifikasi Desain

Desain yang diinginkan pada perancangan *Output Feedback Control* mempunyai spesifikasi yaitu:

1. *Error Steady State* <2%

Error Steady State <2%, karena sistem yang baik memiliki *output* dengan batas nilai akhir 2% dari *setpoint*.

2. $\xi = 1$

$\xi = 1$, agar *output* tidak berosilasi dan dapat mencapai *setpoint*.

3. *Settling time* 5 detik

Settling time 5 detik, karena diharapkan *Settling time* lebih cepat dari sebelum diberi kontroler yaitu 7 detik.

4. *Output* tanpa *overshoot*

Output yang memiliki *overshoot* dapat membuat beban menerima tegangan yang melebihi *setpoint* sehingga bisa menyebabkan filamen wolfram beban terputus.

3.3 Karakterisasi Motor DC

Karakterisasi motor DC dilakukan untuk mengetahui karakter atau gain motor DC. Hal tersebut diperoleh dengan mengamati kecepatan motor DC terhadap perubahan tegangan input motor DC.

Peralatan yang digunakan terdiri atas:

1. *Power Supply Unit* (PSU),
2. Motor DC D-6759,
3. Tachometer digital,
4. Perangkat komputer, dan
5. Kabel penghubung.

Langkah pengujian meliputi:

1. Menghubungkan tegangan *output* Power Supply Unit (PSU) dengan motor DC D-6759.
2. Mengatur tegangan output Power Supply Unit (PSU) dari 0 V sampai 20 V sebagai tegangan sumber motor DC D-6759.

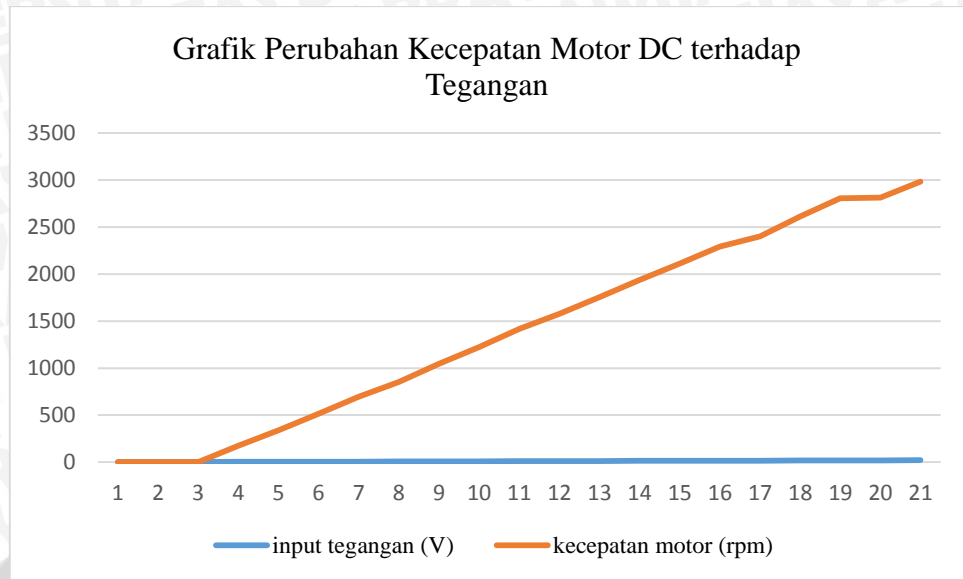
3. Menggunakan tachometer digital untuk mengetahui nilai putaran motor.
4. Mengamati dan mencatat hasil pengukuran putaran motor pada setiap kenaikan tegangan 1V.

Data hasil pengujian tegangan input terhadap kecepatan motor DC ditunjukkan dalam Tabel 3.1.

Tabel 3. 1 Data pengujian kecepatan motor DC terhadap tegangan

Input Tegangan (V)	Kecepatan Motor (rpm)
0	0
1	0
2	0
3	174,8
4	338,6
5	515,6
6	695,9
7	851,2
8	1045
9	1222
10	1419
11	1576
12	1756
13	1938
14	2111
15	2293
16	2403
17	2614
18	2808
19	2813
20	2985

Grafik hasil pengujian tegangan input terhadap kecepatan motor DC ditunjukkan dalam Gambar 3.2.



Gambar 3. 2 Grafik perubahan kecepatan motor DC terhadap tegangan

Dari grafik didapatkan *gain* motor DC adalah :

$$m = \frac{y_{13} - y_{10}}{x_{13} - x_{10}} = \frac{1938 - 1419}{13 - 10} = \frac{519}{3} = 173$$

3.4 Karakterisasi *Driver Motor (High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip)*

Karakteristik, kinerja dan output rangkaian *driver* motor High Current 30 A DC diketahui dengan membandingkan tegangan *output* efektif *driver* dengan masukan *duty cycle* sinyal PWM yang diberikan oleh Arduino Uno R3.

Peralatan yang digunakan terdiri atas:

1. Komputer atau PC,
2. *Power Supply Unit* (PSU),
3. *Driver* motor High Current 30 A DC Motor Driver,
4. Multimeter,
5. Arduino Uno R3 , dan
6. Kabel penghubung.

Langkah pengujian meliputi:

1. Menghubungkan *output* tegangan *Power Supply Unit* (PSU) pada *input* tegangan referensi *driver* motor High Current 30 A DC.
2. Menghubungkan *input* tegangan *driver* motor High Current 30 A DC dengan pin *output* PWM di Arduino Uno R3.

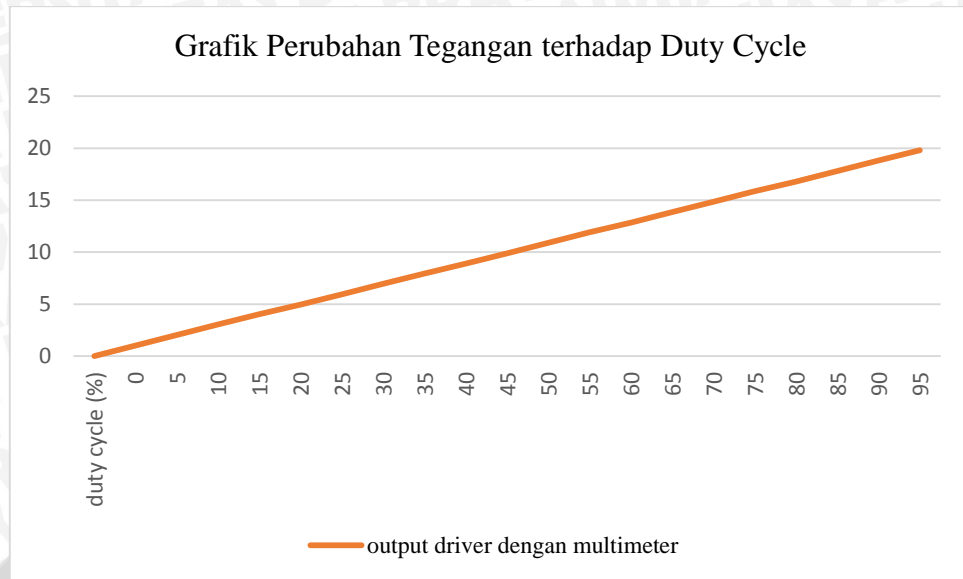
3. Menghubungkan *output* tegangan *driver* motor High Current 30 A DC dengan multimeter.
4. Mengatur *duty cycle* sinyal PWM pada Arduino Uno R3 dengan nilai 0%-100%.
5. Mengamati dan mencatat hasil pembacaan multimeter pada setiap kenaikan 5%.

Data hasil pengujian *driver* motor High Current 30 A DC ditunjukkan dalam Tabel 3.2.

Tabel 3. 1 Data pengujian driver motor High Current 30 A DC Motor Driver

Duty Cycle (%)	Output Driver dengan Multimeter
0	0
5	1.022
10	2.034
15	3.045
20	4.02
25	4.95
30	5.95
35	6.96
40	7.96
45	8.89
50	9.9
55	10.9
60	11.91
65	12.84
70	13.85
75	14.85
80	15.86
85	16.79
90	17.8
95	18.81
100	19.8

Grafik hasil pengujian *driver* motor High Current 30 A DC ditunjukkan dalam Gambar 3.3.



Gambar 3. 3 Grafik perubahan tegangan terhadap duty cycle

Dari grafik didapatkan *gain* driver motor adalah :

$$m = \frac{y_{60} - y_{50}}{x_{60} - x_{50}} = \frac{11,91 - 9,9}{60 - 50} = \frac{2,01}{10} = 0,201$$

3.5 Karakterisasi Sensor Tegangan

Karakteristik, kinerja dan output sensor tegangan diketahui dengan membandingkan tegangan *output* yang diberikan oleh Arduino Uno R3 dan *output* yang dibaca oleh multimeter.

Peralatan yang digunakan terdiri atas:

1. Komputer atau PC.
2. *Power Supply Unit* (PSU).
3. Sensor tegangan
4. Multimeter.
5. Arduino Uno R3
6. Kabel penghubung.

Langkah pengujian meliputi:

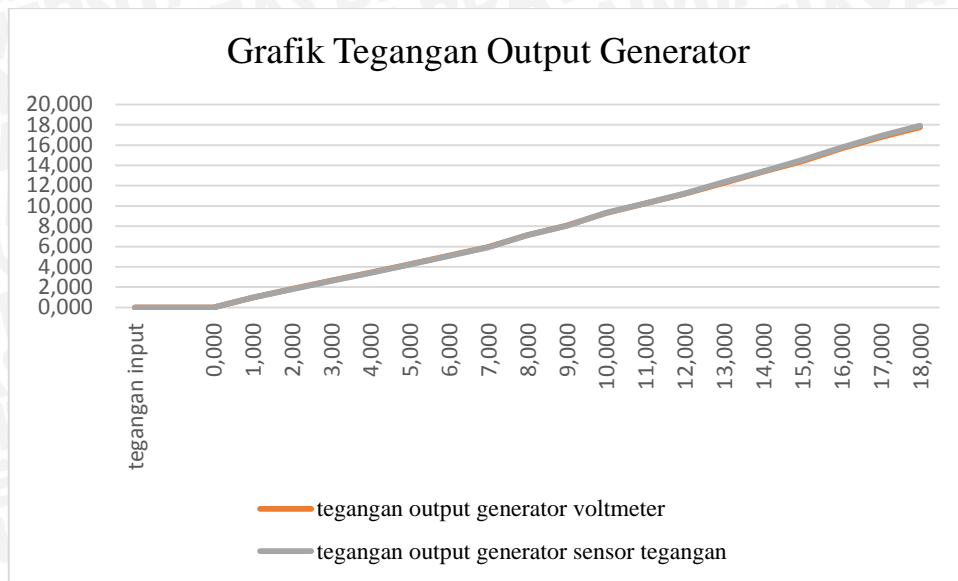
1. Menghubungkan *output* tegangan *Power Supply Unit* (PSU) pada *input* tegangan referensi *driver* motor High Current 30 A DC Motor Driver dan menyambungkan output tegangan generator dengan sensor tegangan.
2. Menghubungkan *output* tegangan motor generator dengan multimeter.
3. Mengatur tegangan keluaran *Power Supply Unit* (PSU).
4. Mengamati dan mencatat hasil pembacaan multimeter disetiap kenaikan 1V

Data hasil pengujian sensor tegangan ditunjukkan dalam Tabel 3.3.

Tabel 3. 2 Data pengujian sensor tegangan

Tegangan Input	Tegangan Output Generator	
	Voltmeter	Sensor Tegangan
0,000	0,000	0,000
1,000	0,000	0,000
2,000	0,000	0,000
3,000	0,977	0,960
4,000	1,823	1,800
5,000	2,644	2,610
6,000	3,421	3,400
7,000	4,240	4,220
8,000	5,090	5,080
9,000	5,970	5,940
10,000	7,120	7,120
11,000	8,060	8,040
12,000	9,300	9,320
13,000	10,260	10,260
14,000	11,200	11,220
15,000	12,270	12,350
16,000	13,360	13,400
17,000	14,420	14,520
18,000	15,680	15,750
19,000	16,770	16,910
20,000	17,750	17,900

Grafik hasil pengujian sensor tegangan ditunjukkan dalam Gambar 3.4.



Gambar 3. 4 Grafik tegangan output generator

Dari grafik didapatkan *gain* sensor tegangan adalah :

$$m = \frac{y_{15} - y_{10}}{x_{15} - x_{10}} = \frac{12,35 - 7,12}{15 - 10} = \frac{5,23}{5} = 1,046$$

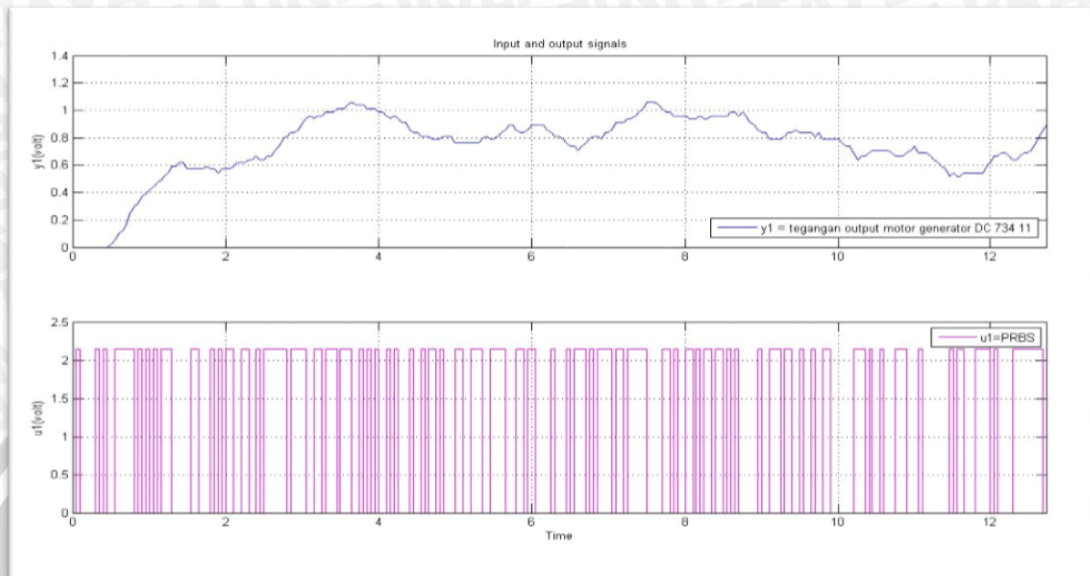
3.6 Penentuan Fungsi Alih *Plant* Motor Generator DC 734 11

Pengontrolan kecepatan motor generator DC 734 11 menggunakan Arduino Uno R3 sebagai pengolah dan memberikan data berupa *Pulse Width Modulation* (PWM) agar motor bergerak. Motor generator yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan sensor tegangan. Karakteristik motor generator DC 734 11 pada perancangan ini didapatkan dengan cara memberikan input unit step.

Fungsi alih *plant* motor generator didapatkan dari pemodelan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS). Langkah yang dilakukan untuk membangkitkan sinyal PRBS adalah sebagai berikut:

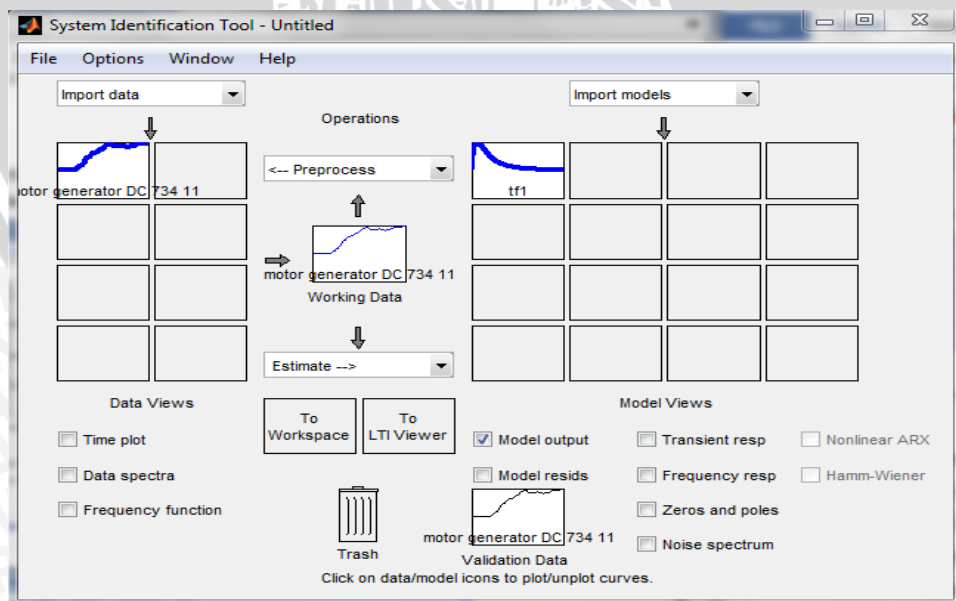
1. Mencari nilai yang linear dari hasil tegangan *output* terhadap duty cycle PWM.
2. Memasukkan nilai batas atas dan bawah berdasarkan nilai yang linier untuk membangkitkan sinyal PRBS.
3. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai *input* motor DC.

4. Setelah didapatkan data sinyal PRBS dan data tegangan *output* motor generator 734 11 (lihat Gambar 3.5), selanjutnya adalah melakukan identifikasi dengan menggunakan *software* Matlab.

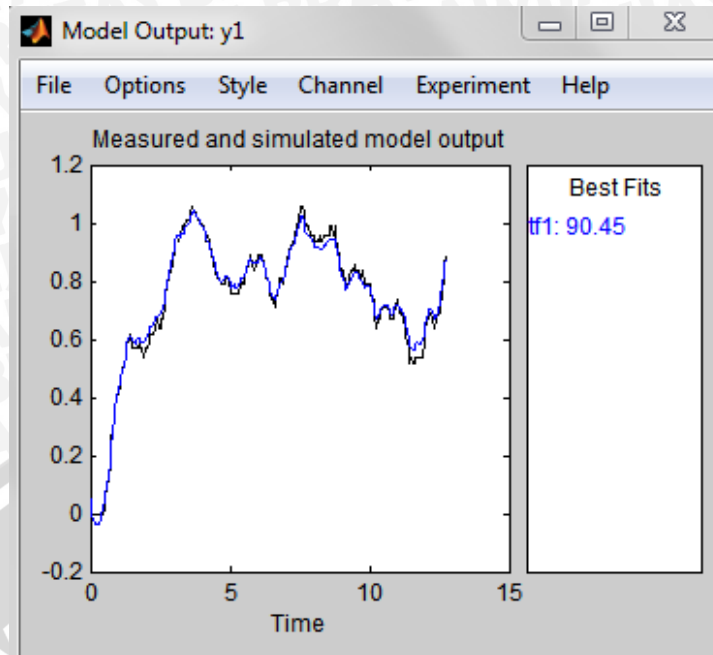


Gambar 3. 5 Output sinyal PRBS

5. 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 3.6). Setelah melakukan beberapa estimasi model berdasarkan data yang telah di *import* didapatkan fungsi alih *plant* motor generator dengan *best fit* sebesar 90.45 (lihat Gambar 3.7).



Gambar 3. 6 System identification toolbox

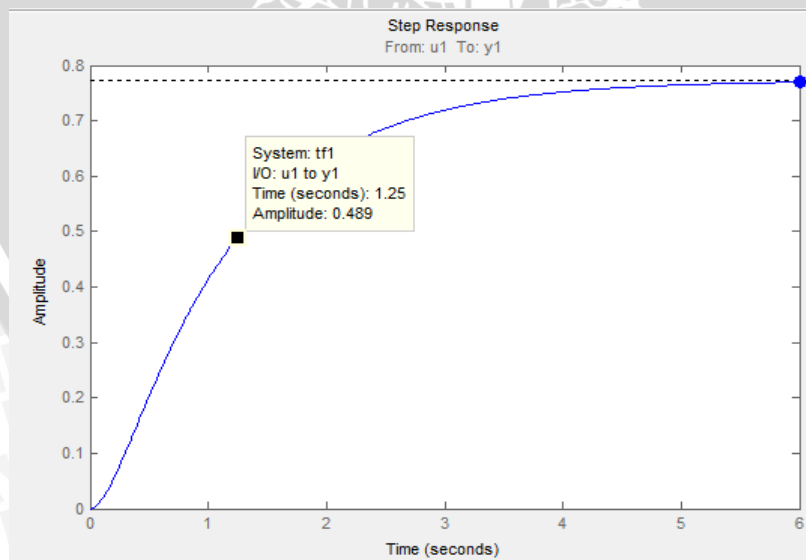


Gambar 3. 7 Hasil estimasi model

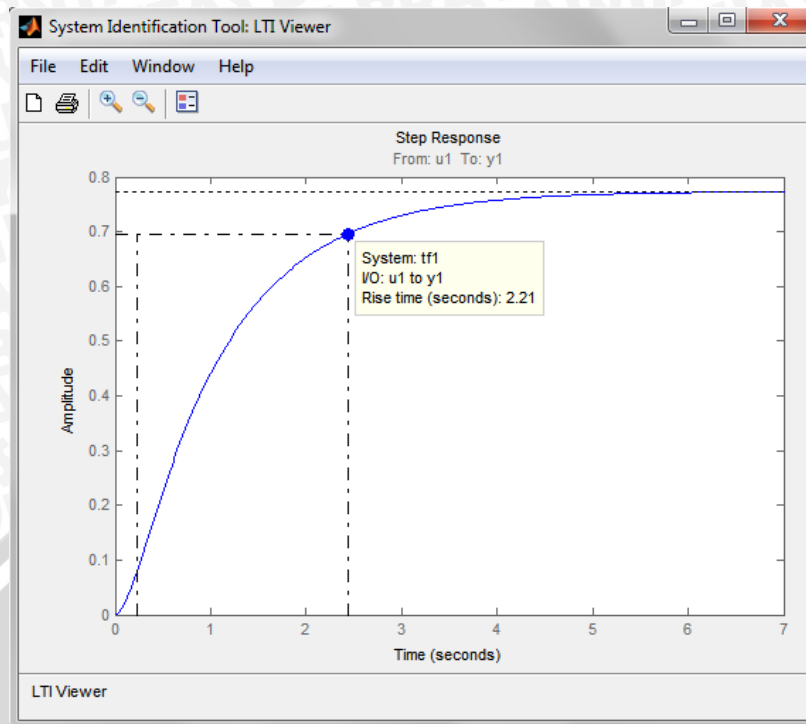
6. Dari hasil identifikasi, fungsi alih motor generator yang didapat adalah

$$\frac{Y(s)}{U(s)} = \frac{4,088}{s^2 + 6,503s + 5,291} \quad (3-1)$$

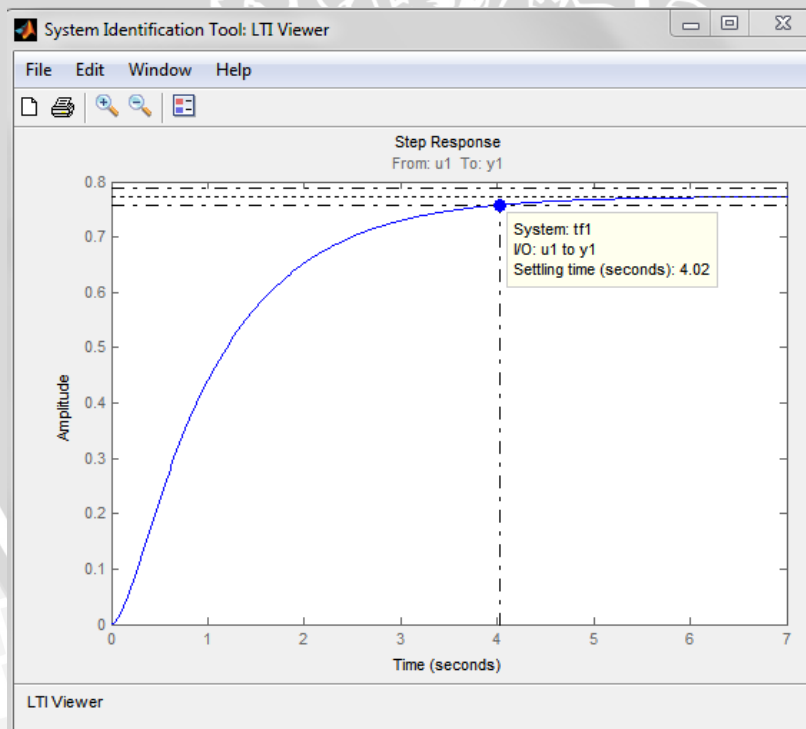
Dengan memberikan input unit step pada program Matlab didapatkan output dengan nilai *time constant* yang merupakan waktu yang dibutuhkan untuk mencapai 63,2% dari nilai *steady state*, yaitu 1,25 detik (Gambar 3.8). Didapatkan pula nilai *rise time* adalah 2,21 detik (Gambar 3.9) dan *settling time* adalah 4,02 detik (Gambar 3.10).



Gambar 3. 8 Nilai *time constant* output plant motor generator DC 734 11 dengan input *unit step*



Gambar 3. 9 Nilai *rise time* output plant motor generator DC 734 11 dengan input unit step

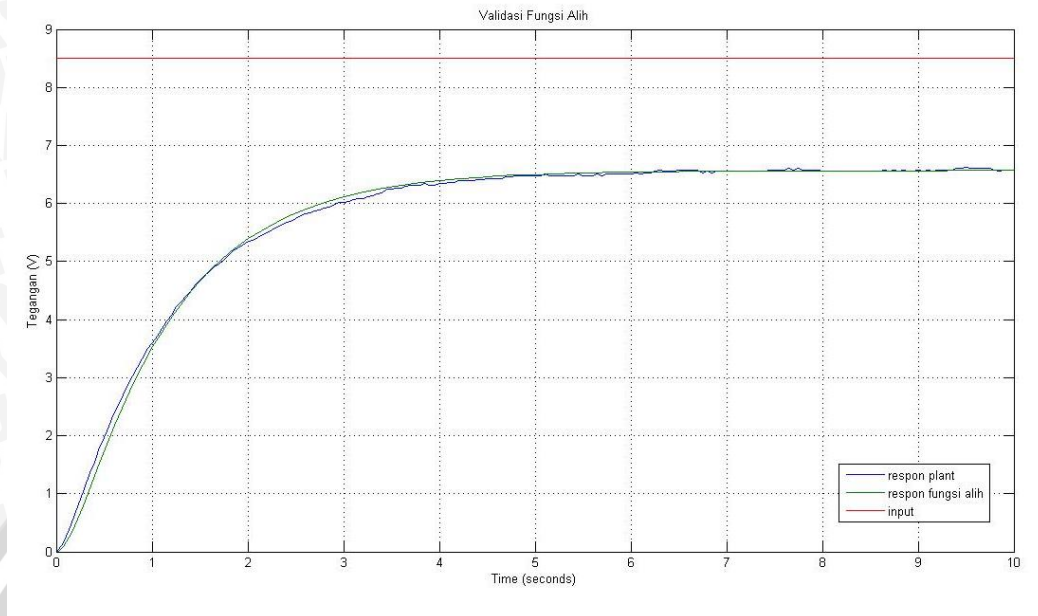


Gambar 3. 10 Nilai *settling time* output plant motor generator DC 734 11 dengan input unit step

3.7 Validasi Fungsi Alih Motor Generator 734 11

Dalam melakukan validasi fungsi alih motor generator dilakukan dengan cara membandingkan *output* fungsi alih dan tegangan *output* motor generator DC 734 11 yang

didapat dari pembacaan sensor tegangan. Berikut perbandingan kedua *output* yang didapat dengan menggunakan Matlab (lihat Gambar 3.11).

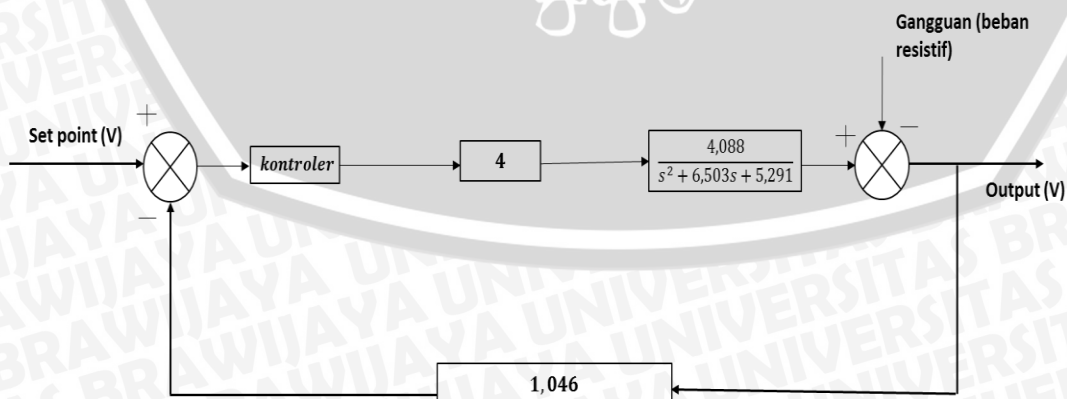


Gambar 3. 11 Validasi fungsi alih dengan *output* motor DC D-6759

Dari grafik dalam Gambar 3.11 dapat dilihat bahwa *output* fungsi alih yang telah didapat dari proses identifikasi hampir menyerupai tegangan *output* motor generator DC 734 11. Jadi fungsi alih yang telah didapatkan dianggap dapat mewakili pemodelan *plant* motor generator DC 734 11.

3.8 Diagram Blok Perancangan Sistem

Setelah didapatkan karakteristik dari masing-masing blok, maka didapatkan diagram blok perancangan sistem (Gambar 3.12).

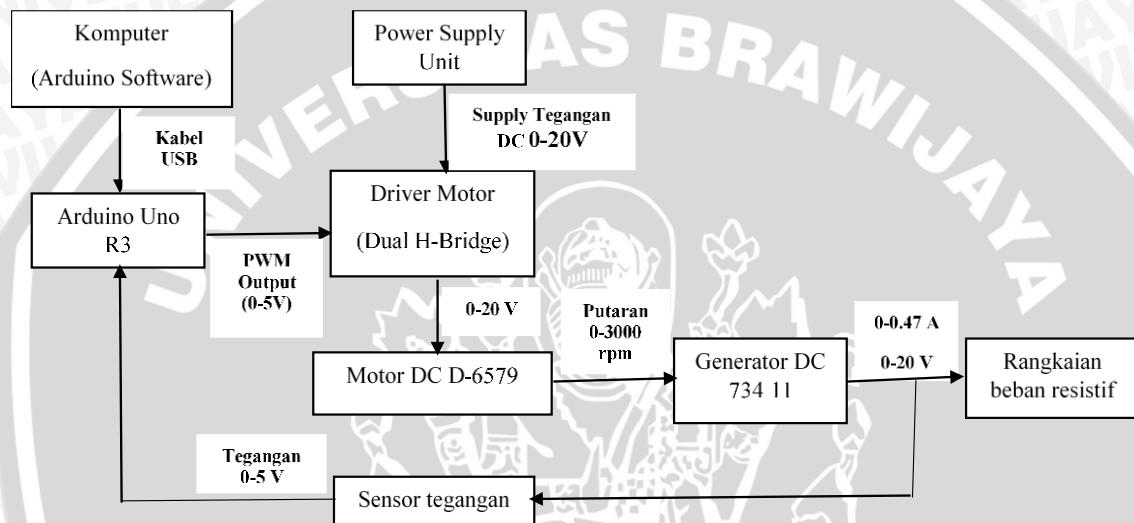


Gambar 3. 12 Diagram blok perancangan Sistem

3.9 Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya. Hal ini dimaksudkan agar pemodelan Generator DC (734 11) dengan pengontrol tegangan *output* menggunakan kontrol optimal LQR metode *Output Feedback Control* dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan.

1. Skema pembuatan perangkat keras (Gambar 3.13)



Gambar 3. 13 Skema Perangkat Keras

2. Penentuan modul elektronik yang digunakan meliputi :

- Power Supply Unit (lihat Gambar 3.14) sebagai catu daya driver motor memiliki range 0-24 Volt.



Gambar 3. 14 Power supply unit (PSU)

- Mikrokontroler Arduino Uno R3 (lihat Gambar 3.15) sebagai perangkat kontroler.



Gambar 3.15 *Arduino Uno R3*

- Sensor tegangan (lihat Gambar 3.16) sebagai *feedback*.



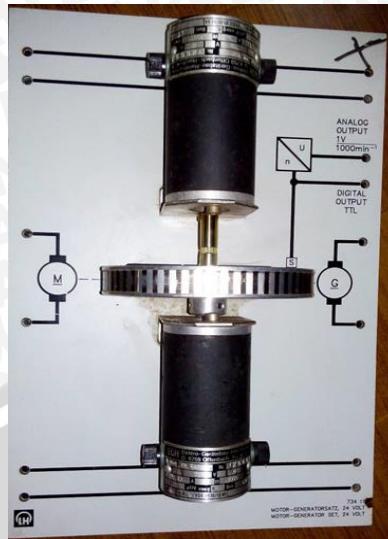
Gambar 3.16 *Sensor tegangan*

- Driver Motor (*High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip*) (lihat Gambar 3.17).



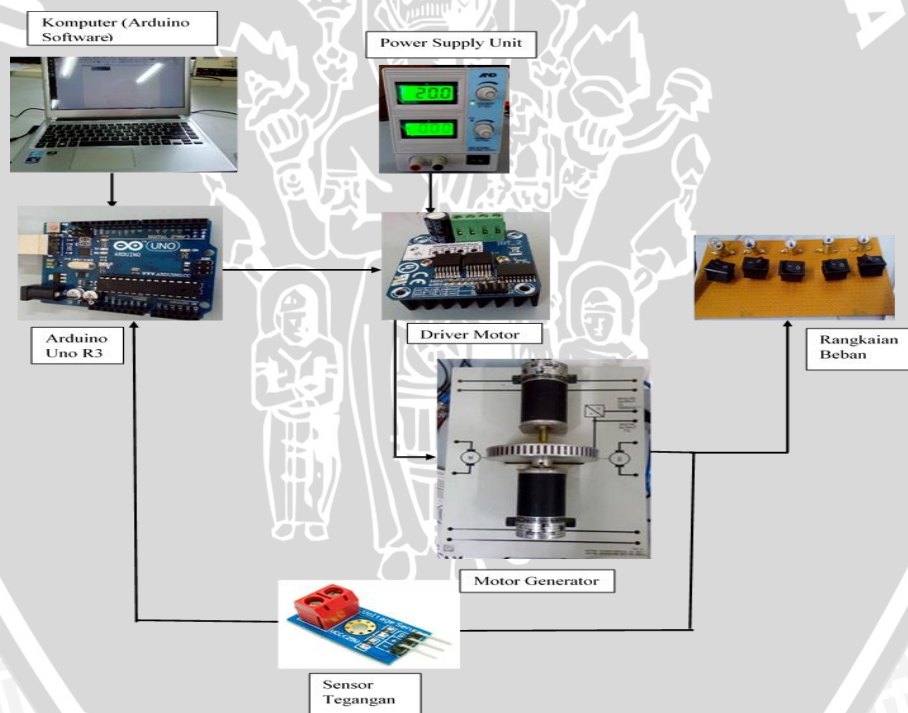
Gambar 3.17 *High current 30 A DC motor driver*

- Rangkaian beban resistif berupa lampu bohlam 2,5 Volt.
- Komputer atau PC yang sudah terinstall *software* arduino.
- Plant Motor Generator DC (734 11) (lihat Gambar 3.18).



Gambar 3. 18 Motor generator DC 734 11

Skema pembuatan alat dan pembuatan modul untuk pengontrolan tegangan output generator DC 734 11 dengan kontrol optimal ditunjukkan dalam Gambar 3.19.



Gambar 3. 19 Skema pembuatan perangkat keras

3.10 Perancangan Algoritma

Dalam perancangan perangkat lunak, hal pertama yang dilakukan adalah mengetahui karakteristik motor generator DC. Setelah itu, dilakukan penghitungan untuk menentukan algoritma kontroler dengan *Output Feedback Control*. Setelah didapatkan

algoritma, kemudian dibuat program untuk mikrokontroler dengan software program Arduino. Perancangan algoritma tersebut melalui beberapa tahap:

1. Mencari matriks R (matriks umpan balik).
2. Mencari matriks L (estimasi)
3. Mencari matriks M (pra filter).
4. Pembuatan simulasi pada software matlab.
5. Penyusunan flowchart yang algoritmanya diterapkan pada mikrokontroler.
6. Implementasi pada alat.

3.11 Desain Kontroler Struktur *Output Feedback Control*.

Dalam mendesain kontroler struktur *Output Feedback Control* diperlukan nilai matriks A, B, C, D, R, L, M. Dimana matriks A adalah matriks keadaan orde $n \times n$, B adalah matriks masukan orde $n \times m$, C adalah matriks keluaran orde $p \times n$, D adalah matriks transmisi langsung orde $p \times n$, R adalah matriks umpan balik, L adalah matriks estimasi, dan M adalah matriks pra filter. Langkah-langkah yang dilakukan untuk mencari matriks-matriks tersebut adalah sebagai berikut:

1. Fungsi alih *plant* motor generator yang didapatkan dari pemodelan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS) digunakan pertama-tama untuk mencari matriks A, B, C, dan D dengan menggunakan *state space*.

$$\frac{Y(s)}{U(s)} = \frac{4,088}{s^2 + 6,503s + 5,291}$$

$$Y(s)(s^2 + 6,503s + 5,291) = 4,088 U(s)$$

$$(s^2Y(s) + 6,503sY(s) + 5,291Y(s)) = 4,088 U(s)$$

$$\frac{d^2y}{dt^2} + 6,503 \frac{dy}{dt} + 5,291 y(t) = 4,088 u(t)$$

$$x_1 = y; x_2 = \dot{y}; \dot{x}_1 = x_2; \dot{x}_2 = \ddot{y} = -6,503x_2 - 5,291x_1 + 4,088 u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5,291 & -6,503 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 4,088 \end{bmatrix} u \quad (3-2)$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3-3)$$

Dari persamaan (3-2) dan (3-3) didapatkan matriks:

$$A = \begin{bmatrix} 0 & 1 \\ -5,291 & -6,503 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 4,088 \end{bmatrix}$$

$$C = [1 \quad 0]$$

$$D = [0]$$

2. Setelah diketahui matriks A, B, C, dan D dari plant selanjutnya adalah menentukan pole dari kontroler untuk mendapatkan nilai matriks umpan balik R. Matriks umpan balik R bisa didapatkan dengan menggunakan persamaan pada matlab yaitu:

$$r = \text{place}(a, b, rp);$$

dimana:

a= matriks A, matriks orde 2x2

b=matriks B, matriks orde 2x1

rp= pole kontroler yang nilainya bebas (rp=rp1;rp2), matriks orde 2x1

3. Setelah mendapatkan nilai matriks R, langkah selanjutnya adalah mencari matriks estimasi L. Matriks estimasi L bisa didapatkan dengan menggunakan persamaan pada matlab yaitu:

$$l = \text{place}(a', c', lp)';$$

dimana:

a= matriks A, matriks orde 2x2

c= matriks C, matriks orde 1x2

lp= pole observer yang nilainya bebas (lp=lp1;lp2), matriks orde 2x1

4. Setelah mendapatkan nilai matriks L, langkah selanjutnya adalah mencari matriks *pra filter* M menggunakan persamaan:

$$M = [-C(A - BR)^{-1}B]^{-1}$$

Dengan menggunakan matlab, kita bisa mencari nilai matriks M dengan menggunakan persamaan:

$$dcg = \text{dcgain}(anoob, bnoob, cnoob, dnoob);$$

$$m = 1 / dcg$$

dimana:

anoob=a-b*r;

bnoob=b;

$cnoob=c;$

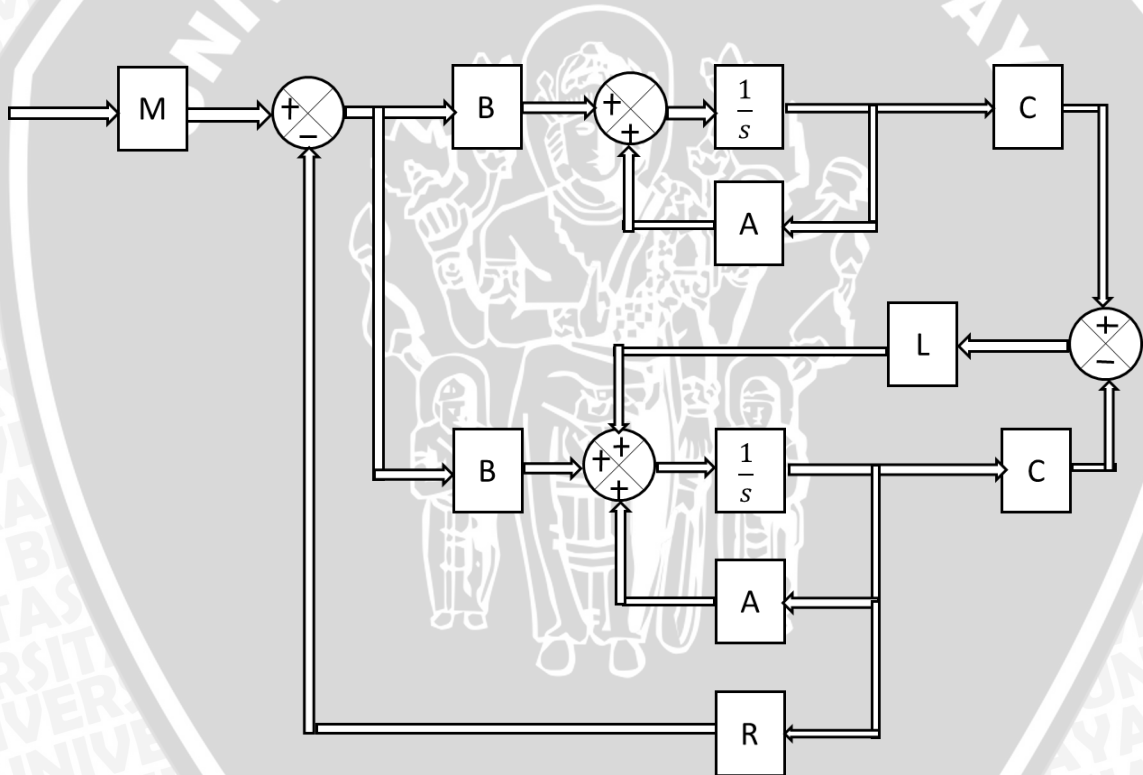
$dnoob=d;$

5. Setelah mendapatkan nilai matriks R , L , M kita perlu mengetahui kestabilan dari sistem dengan cara mencari nilai *eigen value* atau akar-akar dari persamaan karakteristik. Jika nilai *eigen value* bernilai negatif, maka sistem stabil. Kita bisa mencari nilai matriks *eigen value* pada Matlab menggunakan persamaan:

$$e = eig(A - L * C)$$

3.12 Diagram Blok Perancangan Desain

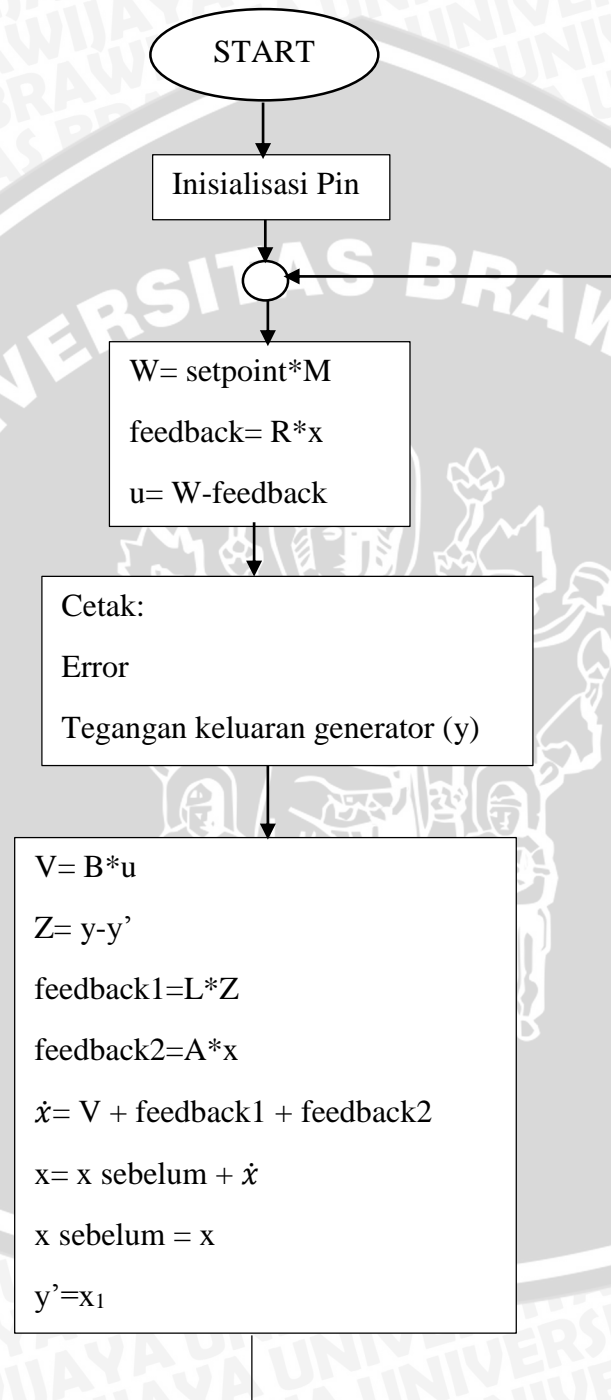
Setelah dilakukan desain, didapatkan blok perancangan desain (lihat Gambar 3.20).



Gambar 3. 20 Diagram Blok Perancangan Desain

3.13 Flowchart Program

Flowchart program merupakan gambaran alur proses program yang dilakukan oleh kontroler pada saat implementasi. *Flowchart* program dalam skripsi ini dapat dilihat dalam Gambar 3.21.



Gambar 3. 21 *Flowchart* program



BAB IV

HASIL DAN PEMBAHASAN

Hasil dan pembahasan dilakukan dengan melakukan simulasi dan menganalisis sistem. Simulasi dan analisis sistem dilakukan untuk mengetahui apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan skema pembuatan. Simulasi dilakukan untuk mendapatkan letak kesalahan dan mempermudah analisis pada sistem.

Simulasi dibagi menjadi beberapa bagian, yaitu:

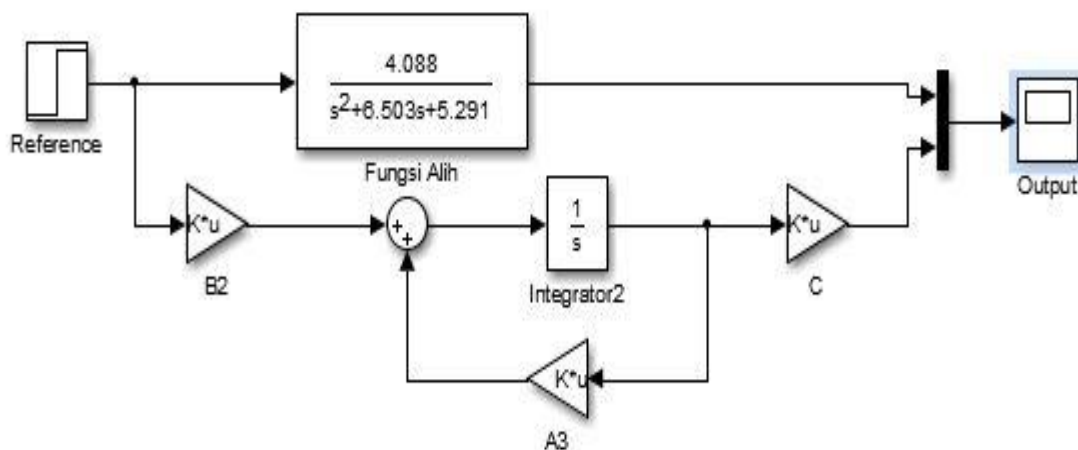
1. Simulasi output fungsi alih dan diagram blok pada simulink.
2. Simulasi penentuan matriks R, L, M dari perubahan pole kontroler dan perubahan pole observer.

Analisis sistem dibagi menjadi 2 bagian, yaitu:

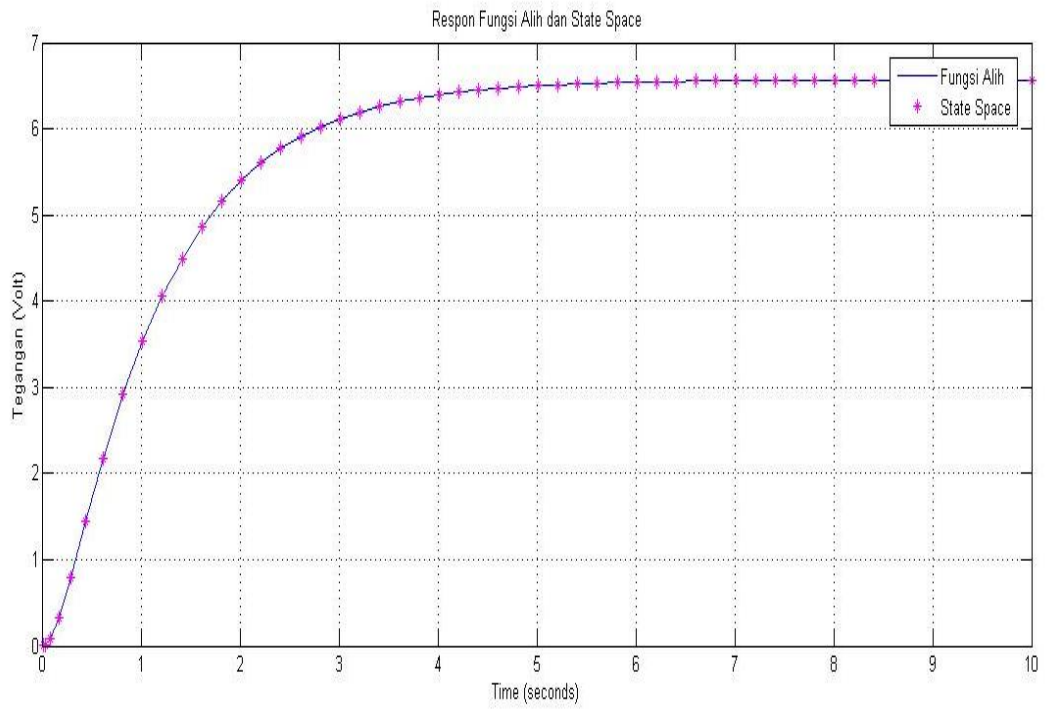
1. Analisis sistem berdasarkan simulasi.
2. Analisis sistem berdasarkan implementasi.

4.1 Simulasi *Output* Fungsi Alih dan Diagram Blok Bentuk *State Space* pada Simulink

Simulasi output fungsi alih dan diagram blok bentuk *state space* pada simulink (Gambar 4.1) diperlukan agar kita mengetahui apakah *output* diagram blok bentuk *state space* yang sudah dibuat memiliki *output* yang sama dengan fungsi alih.



Gambar 4. 1 Simulasi *output* fungsi alih dan diagram blok pada simulink



Gambar 4. 1 Output fungsi alih dan diagram blok pada simulink

Dari simulasi di atas bisa didapatkan bahwa *output* antara fungsi alih dengan diagram blok sama, sehingga simulasi lain bisa dilanjutkan menggunakan diagram blok tersebut.

4.2 Simulasi penentuan matriks R, L, M dari perubahan pole kontroler dan perubahan pole observer.

Simulasi penentuan matriks R, L, M menggunakan software matlab dengan sintaks:

```
%observer

%plant
a=[0 1;-5.291 -6.503];
b=[0;4.088];
c=[1 0];
d=0;

% penetapan pole-pole untuk observer dan controller (nilainya diubah)
lp=[-5.6759; -0.1941];
rp=[-5.5944; -0.7279];

k=place(a,b,rp); % Penentuan R pole untuk controller
l=place(a',c',lp)'; %Penentuan L pole untuk observer

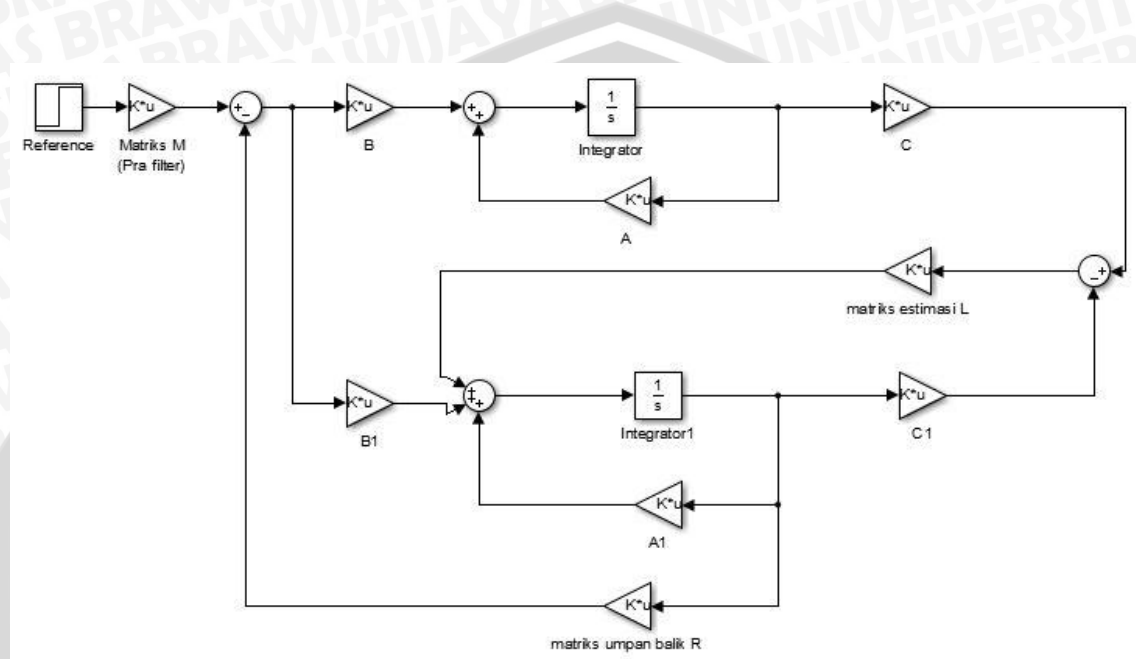
anoob=a-b*k;
bnoob=b;
cnoob=c;
dnoob=d;

deg=degain(anoob,bnoob,cnoob,dnoob);
```

$m=1/dcg$

`e=eig(a-l*c);%nilai eigenvalue`

setelah mendapatkan nilai R, L, M simulasi dilakukan dengan menggunakan simulink pada matlab dengan struktur *output feedback control*.



Gambar 4. 2 Simulasi penggunaan matriks R, L, M

4.3 Simulasi Perubahan Pole Observer terhadap perubahan nilai matriks R, L, M.

Sebelum melakukan implementasi pada alat, kita perlu mengetahui pengaruh perubahan pada pole observer (op) terhadap energi atau sinyal kontrol (U) dan terhadap *output* sistem. Kita misalkan $op = [op1; op2]$.

4.3.1 Perubahan Nilai $op1$

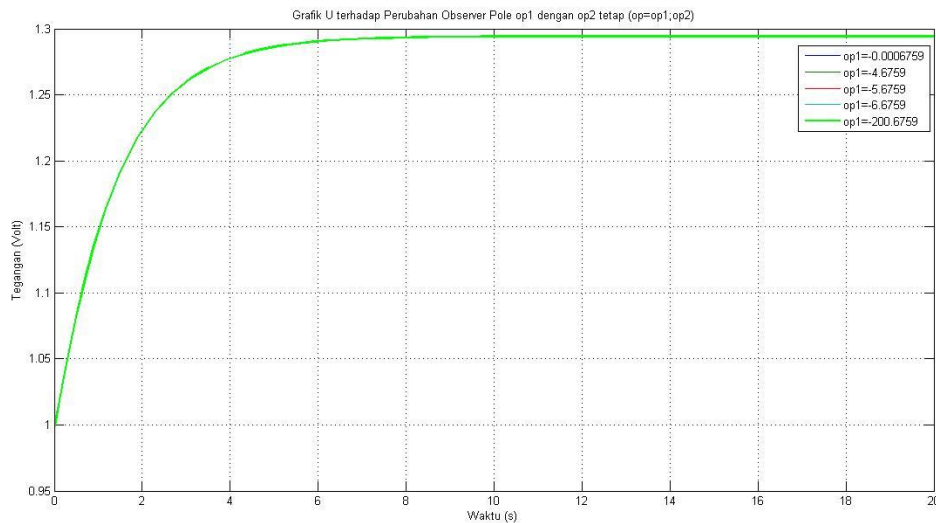
Simulasi perubahan nilai $op1$ menggunakan 5 data (Lihat Tabel 4.1). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai $op1$ terhadap nilai U (lihat Gambar 4.4) dan terhadap tegangan *output* sistem (lihat Gambar 4.5).



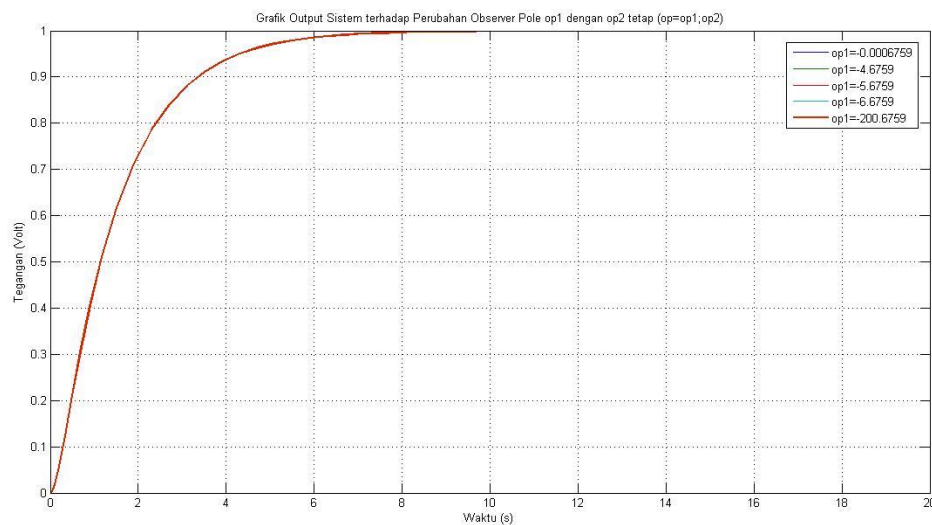
Tabel 4. 1 Nilai Matriks R, L, M dan Eigen Value terhadap Perubahan Nilai Observer Pole op1

No	L pole observer	R pole kontroler	R	L	M	Eigen Value
1	[-0,0006759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[-6,308224099999542; 35,731512514487180]	0,996126164	[-0,000675900000077; -0,194099999999683]
2	[-4,6759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[-1,632999999999997; 6,235991189999998]	0,996126164	[-0,194100000000003; -4,675900000000000]
3	[-5,6759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[-0,632999999999998; -0,072908810000004]	0,996126164	[-0,194100000000001; -5,675900000000000]
4	[-6,6759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[0,367000000000000; -6,381808810000003]	0,996126164	[-0,194099999999999; -6,675900000000000]
5	[-200,6759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[194,36700000000006; -1230,308408810006]	0,996126164	[-200,6759000000006; -0,194099999999989]

Dari data dalam Tabel 4.1 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan U dan *output* sistem terhadap perubahan *observer pole* $op1$.



Gambar 4. 3 Grafik U dengan nilai $op1$ berbeda-beda $op2$ tetap, $op=[op1;op2]$



Gambar 4. 4 Output sistem dengan nilai $op1$ berbeda-beda $op2$ tetap, $op=[op1;op2]$

Dari Gambar 4.4 dan Gambar 4.5 dapat dilihat bahwa perubahan nilai *observer pole* $op1$ tidak mempengaruhi nilai U. Begitu juga dengan *output* sistem, perubahan nilai *observer pole* $op1$ tidak mempengaruhi *output* sistem.

4.3.2 Perubahan Nilai op2

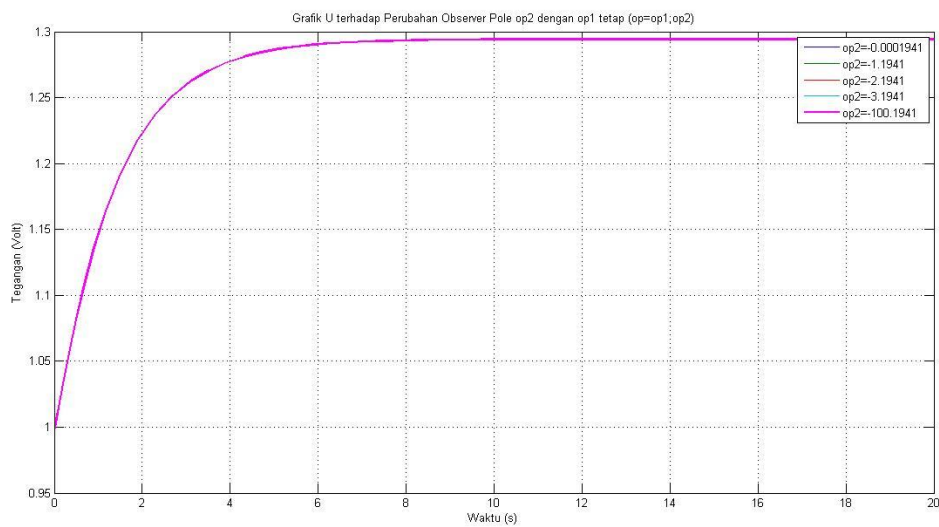
Simulasi perubahan nilai op2 menggunakan 5 data (lihat Tabel 4.2). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai op1 terhadap nilai U (lihat Gambar 4.6) dan terhadap *output* sistem (lihat Gambar 4.7).



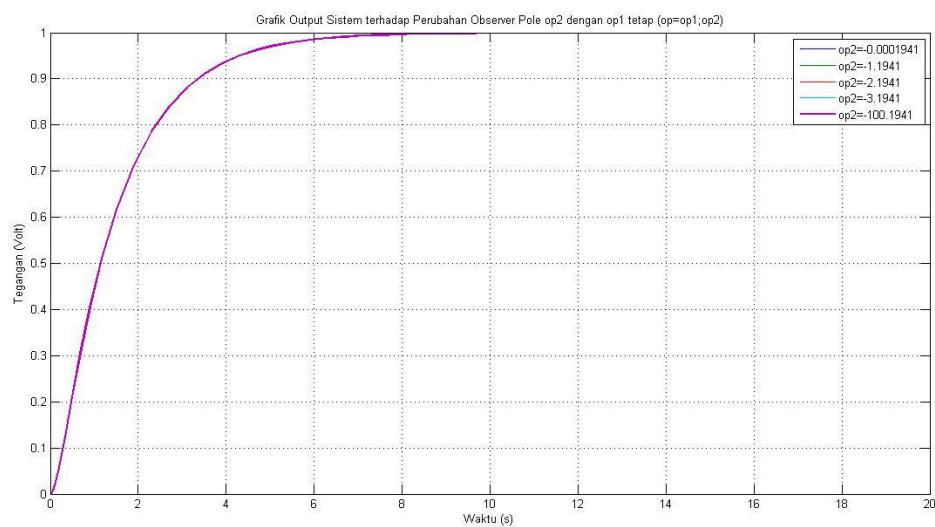
Tabel 4. 2 Nilai Matriks R, L, M dan Eigen Value terhadap Perubahan Nilai Observer Pole op2

No	L pole observer	R pole kontroler	R	L	M	Eigen Value
1	[-5,6759; -0,0001941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[-0,8269058999999999; 0,087470759889997]	0,996126164	[-0,000194100000000752; -5,6759000000000000]
2	[-5,6759; -1,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[0,3670000000000002; -0,9000088100000004]	0,996126164	[-1,1941000000000002; -5,6759000000000000]
3	[-5,6759; -2,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[1,3670000000000000; -1,7271088100000002]	0,996126164	[-2,1941000000000000; -5,6759000000000000]
4	[-5,6759; -3,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[2,3669999999999999; -2,5542088100000000]	0,996126164	[-3,1940999999999999; -5,6759000000000000]
5	[-5,6759; -100,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[99,366999999999980; -82,782908809999940]	0,996126164	[-100,19410000000000; -5,6759000000000000]

Dari data tabel 4.2 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan U dan *output* sistem terhadap perubahan *observer pole* op2.



Gambar 4. 5 Grafik U dengan nilai op2 berbeda-beda op1 tetap, op=[op1;op2]



Gambar 4. 6 Output sistem dengan nilai op2 berbeda-beda op1 tetap, op=[op1;op2]

Dari Gambar 4.6 dan Gambar 4.7 dapat dilihat bahwa perubahan nilai *observer pole* op2 tidak mempengaruhi nilai U. Begitu juga dengan *output* sistem, perubahan nilai *observer pole* op2 tidak mempengaruhi tegangan *output* sistem.

4.4 Simulasi Perubahan Pole Kontroler terhadap perubahan nilai matriks R, L, M.

Sebelum melakukan implementasi pada alat, kita perlu mengetahui pengaruh perubahan pada *pole observer* terhadap U dan terhadap *output* sistem. Kita misalkan $rp = [rp1; rp2]$.

4.4.1 Perubahan Nilai rp1

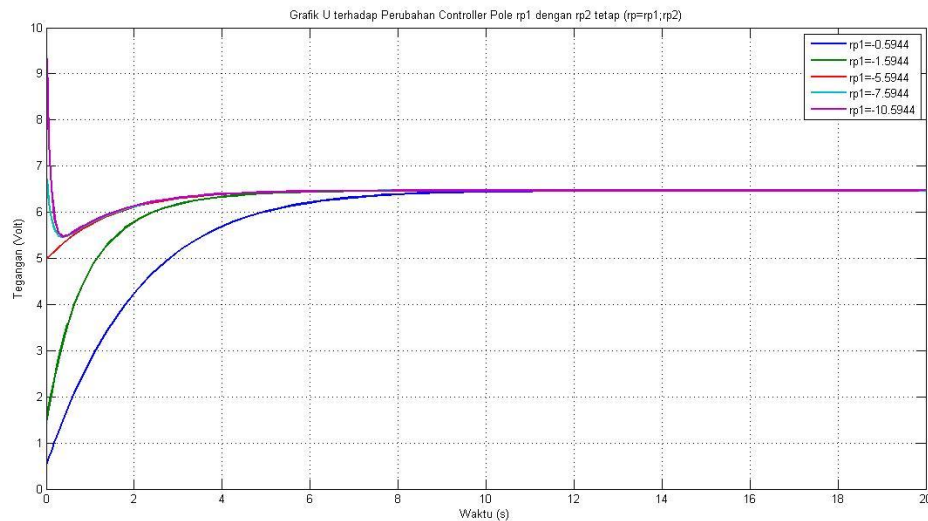
Simulasi perubahan nilai rp1 menggunakan 5 data (Lihat Tabel 4.3). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai rp1 terhadap nilai U (lihat Gambar 4.8) dan terhadap tegangan *output* sistem (lihat Gambar 4.9).



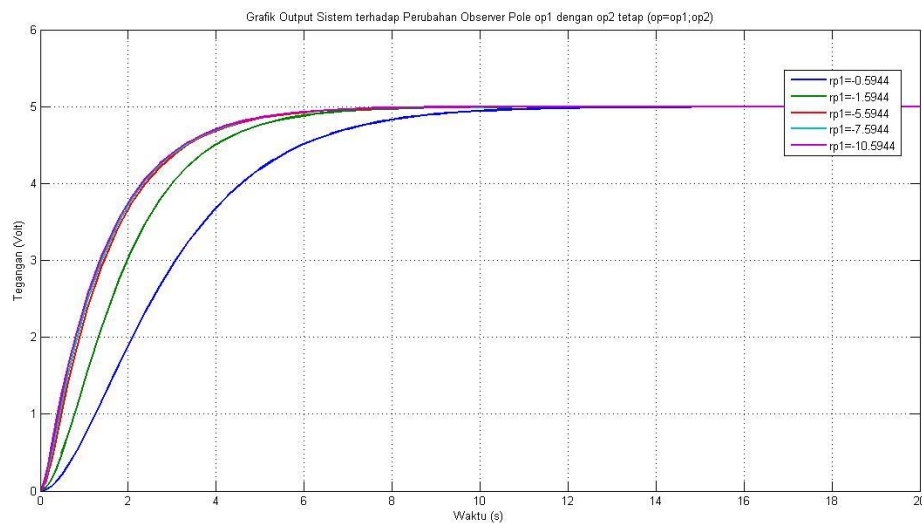
Tabel 4. 3 Nilai Matriks R, L, M dan Eigen Value terhadap Perubahan Nilai Kontroler Pole rp1

No	L pole observer	R pole kontroler	k	l	m	eigen value
1	[-5,6759; -0,1941]	[-0,5944; -0,7279]	[-1,188438414872798 -1,267294520547945]	[-0,632999999999998; -0,072908810000004]	0,105837515	[-0,194100000000001; -5,675900000000000]
2	[-5,6759; -0,1941]	[-1,5944; -0,7279]	[-1,010380684931507 -1,022676125244618]	[-0,632999999999998; -0,072908810000004]	0,283895245	[-0,194100000000001; -5,675900000000000]
3	[-5,6759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[-0,632999999999998; -0,072908810000004]	0,996126164	[-0,194100000000001; -5,675900000000000]
4	[-5,6759; -0,1941]	[-7,5944; -0,7279]	[0,057965694716243 0,445034246575342]	[-0,632999999999998; -0,072908810000004]	1,352241624266	[-0,194100000000001; -5,675900000000000]
5	[-5,6759; -0,1941]	[-10,5944; -0,7279]	[0,592138884540118 1,178889432485323]	[-0,632999999999998; -0,072908810000004]	1,88641481409	[-0,194100000000001; -5,675900000000000]

Dari data tabel 4.3 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan U dan *output* sistem terhadap perubahan kontroler pole $rp1$.



Gambar 4. 7 Grafik U dengan nilai $rp1$ berbeda-beda $rp2$ tetap, $rp=[rp1;rp2]$



Gambar 4. 8 *Output* Sistem dengan nilai $rp1$ berbeda-beda $rp2$ tetap, $rp=[rp1;rp2]$

Dari Gambar 4.8 dan Gambar 4.9 dapat dilihat bahwa semakin kecil nilai kontroler pole $rp1$ maka nilai U pada awal waktu semakin besar. Tetapi pada waktu selanjutnya, nilai U mempunyai nilai yang sama walaupun nilai $rp1$ berubah. Begitu juga dengan tegangan *output* sistem, semakin kecil nilai kontroler pole $rp1$ maka tegangan *output* sistem untuk mencapai *steady state* akan semakin cepat.

4.4.2 Perubahan Nilai rp_2

Simulasi perubahan nilai rp_2 menggunakan 5 data (lihat Tabel 4.4). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai rp_2 terhadap nilai U (lihat Gambar 4.10) dan terhadap *output* sistem (lihat Gambar 4.11).

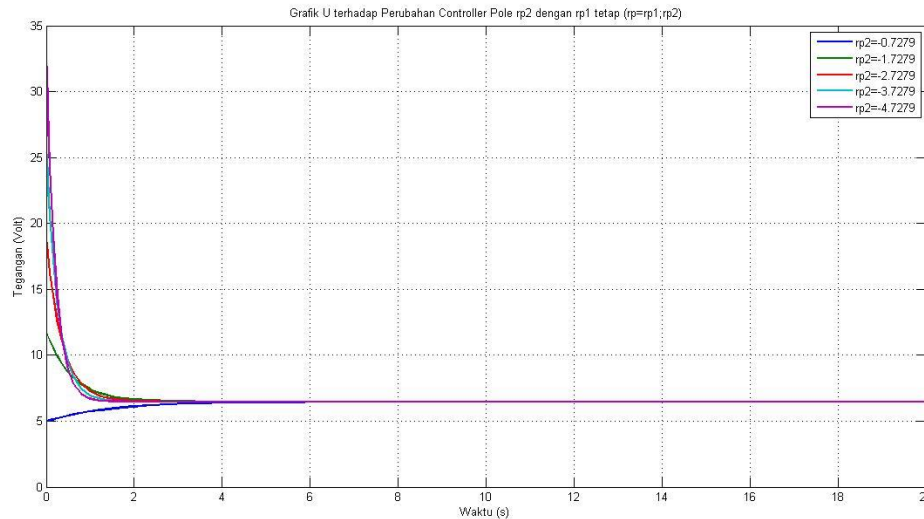


Tabel 4. 4 Nilai Matriks R, L, M dan Eigen Value terhadap Perubahan Nilai Kontroler Pole rp2

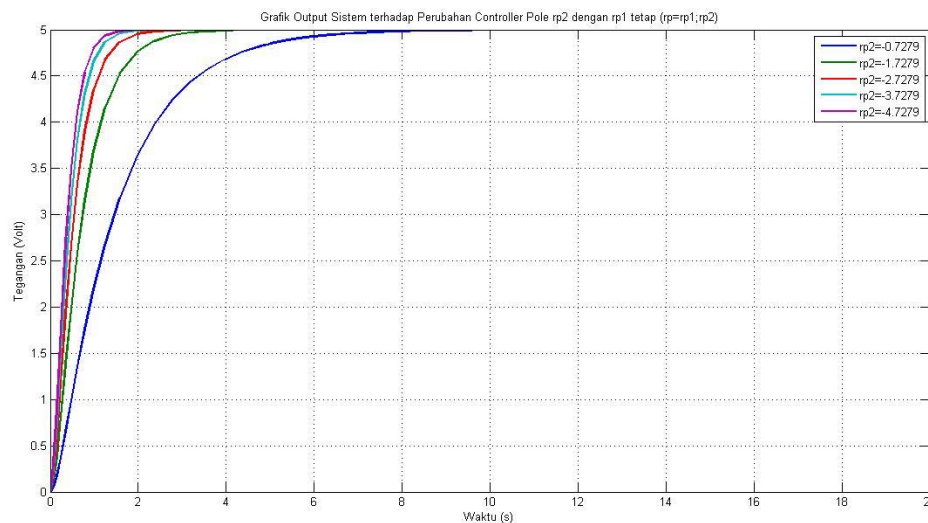
No	L pole observer	R pole kontroler	R	L	M	Eigen Value
1	[-5,6759; -0,1941]	[-5,5944; -0,7279]	[-0,298149765166340 -0,044202544031311]	[-0,632999999999998; -0,072908810000004]	0,996126164	[-0,194100000000001; -5,675900000000000]
2	[-5,6759; -0,1941]	[-5,5944; -1,7279]	[1,070343385518592 0,200415851272016]	[-0,632999999999998; -0,072908810000004]	2,36461931506849	[-0,194100000000001; -5,675900000000000]
3	[-5,6759; -0,1941]	[-5,5944; -2,7279]	[2,438836536203524 0,445034246575342]	[-0,632999999999998; -0,072908810000004]	3,73311246575342	[-0,194100000000001; -5,675900000000000]
4	[-5,6759; -0,1941]	[-5,5944; -3,7279]	[3,807329686888453 0,689652641878669]	[-0,632999999999998; -0,072908810000004]	5,10160561643835	[-0,194100000000001; -5,675900000000000]
5	[-5,6759; -0,1941]	[-5,5944; -4,7279]	[5,175822837573391 0,934271037181997]	[-0,632999999999998; -0,072908810000004]	6,47009876712329	[-0,194100000000001; -5,675900000000000]



Dari data dalam Tabel 4.4 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan U dan tegangan *output* sistem terhadap perubahan kontroler pole rp_2 .



Gambar 4. 9 Grafik U dengan nilai rp_2 berbeda-beda rp_1 tetap, $rp=[rp_1;rp_2]$

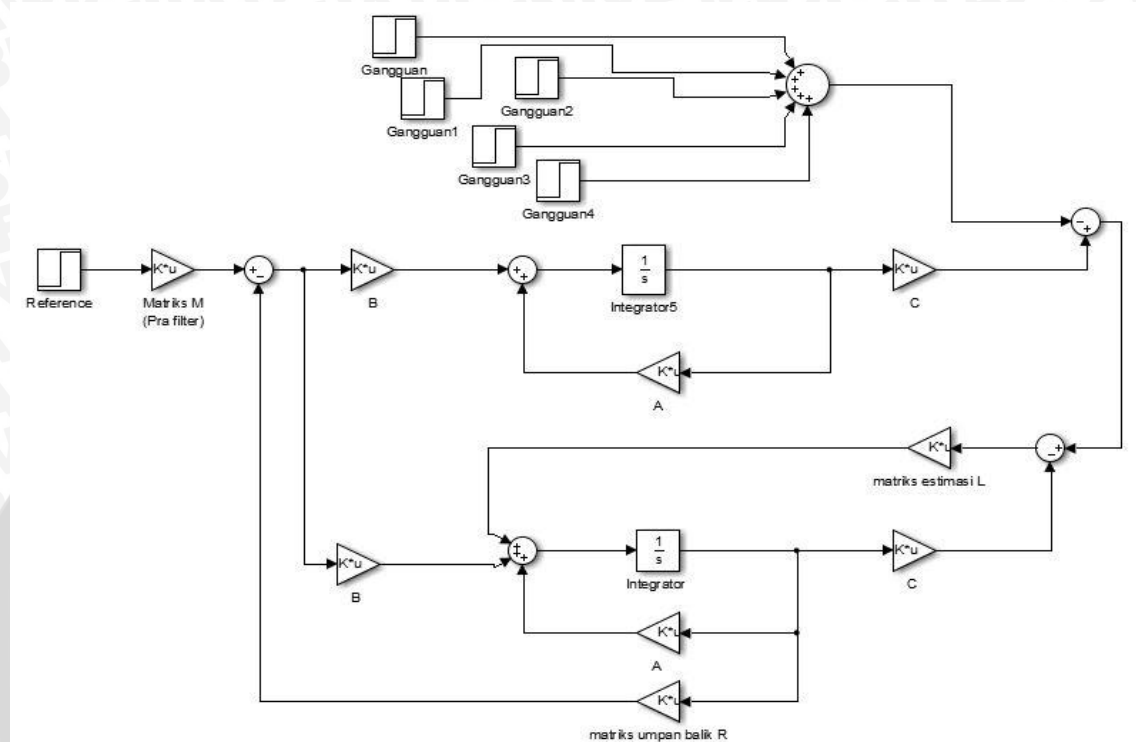


Gambar 4. 10 Output Sistem dengan nilai rp_2 berbeda-beda rp_1 tetap, $rp=[rp_1;rp_2]$

Dari Gambar 4.10 dan Gambar 4.11 dapat dilihat bahwa semakin kecil nilai kontroler pole rp_2 maka nilai U pada awal waktu semakin besar. Tetapi pada waktu selanjutnya, nilai U mempunyai nilai yang sama walaupun nilai rp_2 berubah. Begitu juga dengan tegangan *output* sistem, semakin kecil nilai kontroler pole rp_2 maka tegangan *output* sistem untuk mencapai *steady state* akan semakin cepat.

4.5 Simulasi dengan Gangguan

Setelah diketahui pengaruh perubahan kontroler pole dan *observer pole*, langkah selanjutnya adalah simulasi dengan memberikan gangguan pada tegangan *output* sistem. Simulasi dilakukan dengan menggunakan simulink pada matlab (Lihat Gambar 4.12).



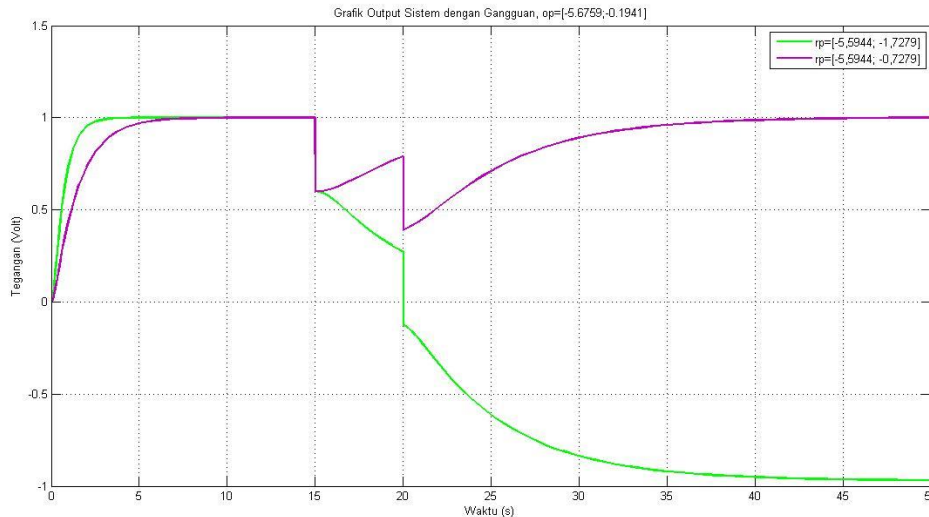
Gambar 4. 11 Simulasi dengan gangguan

Dari simulasi perubahan nilai kontroler pole dan *observer pole*, maka diputuskan simulasi dengan gangguan menggunakan 2 data yang berbeda. (lihat Tabel 4.5). Data ini dipilih dengan melihat dari grafik U dan tegangan *output* sistem pada perubahan kontroler pole. Nilai matriks R , L dan M yang dipilih adalah nilai yang mempunyai grafik dengan nilai U yang tidak terlalu besar, waktu yang diperlukan sistem untuk mencapai *steady state* cukup cepat.

Tabel 4. 5 Nilai Matriks R , L , M untuk Simulasi dengan Gangguan

No	L pole observer	R pole kontroler	R	L	M
1	[-5,6759;- 0,1941]	[-5,5944;- 1,7279]	[1,070343385518592 0,200415851272016]	[-0,632999999999998;- 0,072908810000004]	2,36461931506849
2	[-5,6759;- 0,1941]	[-5,5944;- 0,7279]	[-0,29814976516634 -0,04420254403131]	[-0,632999999999998;- 0,072908810000004]	0,996126164

Dari data dalam Tabel 4.5 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat tegangan *output* sistem apabila diberi gangguan (lihat Gambar 4.13).

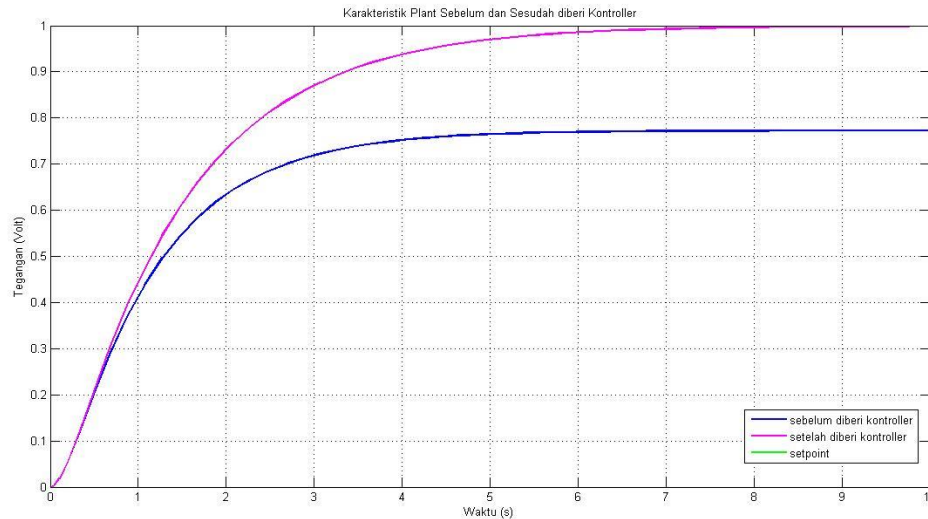


Gambar 4. 12 Tegangan *Output* sistem dengan gangguan

Dari simulasi yang dilakukan, dapat dilihat bahwa tegangan *output* sistem yang bisa kembali ke keadaan semula adalah data dengan nilai $rp=[-5,5944;-0,7279]$ meskipun waktu yang diperlukan output untuk mencapai steady state lebih lambat dari data dengan nilai $rp=[-5,5944;-1,7279]$. Oleh karena itu, data dengan nilai dari $op=[-5,6759;-0,1941]$, $rp=[-5,5944;-0,7279]$, $R=[-0,298149765166340 \quad -0,044202544031311]$, $L=[-0,6329999998;-0,072908810000004]$, dan $M= 0,996126164$ akan diimplementasikan pada alat.

4.6 Simulasi Karakteristik Plant Sebelum dan Sesudah diberi Kontroler

Simulasi karakteristik plant sebelum dan sesudah diberi kontroler ini berfungsi untuk mengetahui pengaruh kontroler terhadap tegangan *output* plant (lihat Gambar 4.14).



Gambar 4. 13 Karakteristik tegangan *output plant* sebelum dan sesudah diberi kontroler

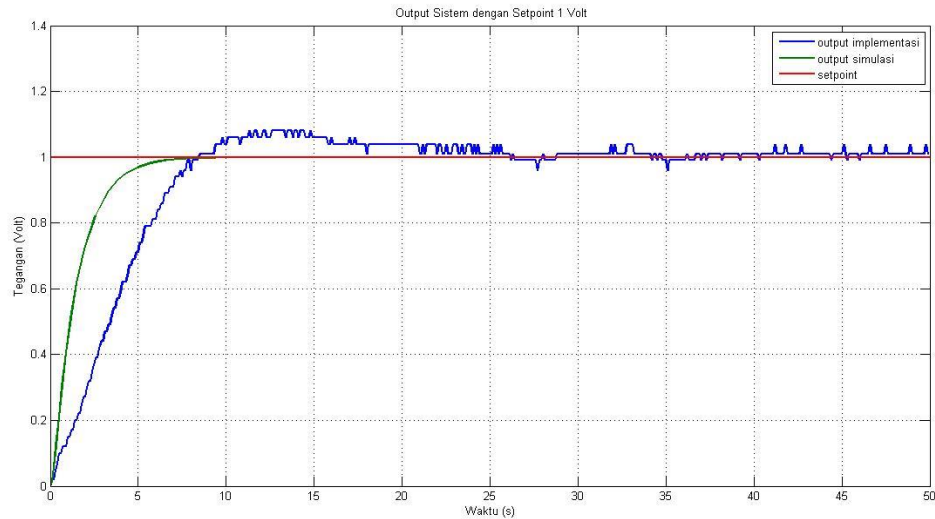
Dari simulasi dapat diketahui bahwa *output plant* saat diberi *input unit step* sebelum diberi kontroler mempunyai *settling time* 5 detik dengan *error steady state* 24% tanpa *overshoot*. Sedangkan *output plant* saat diberi *input unit step* setelah diberi kontroler mempunyai *settling time* 7 detik dengan *error steady state* <1% tanpa *overshoot*.

4.7 Implementasi pada Alat

4.7.1 Implementasi dengan mengubah-ubah setpoint

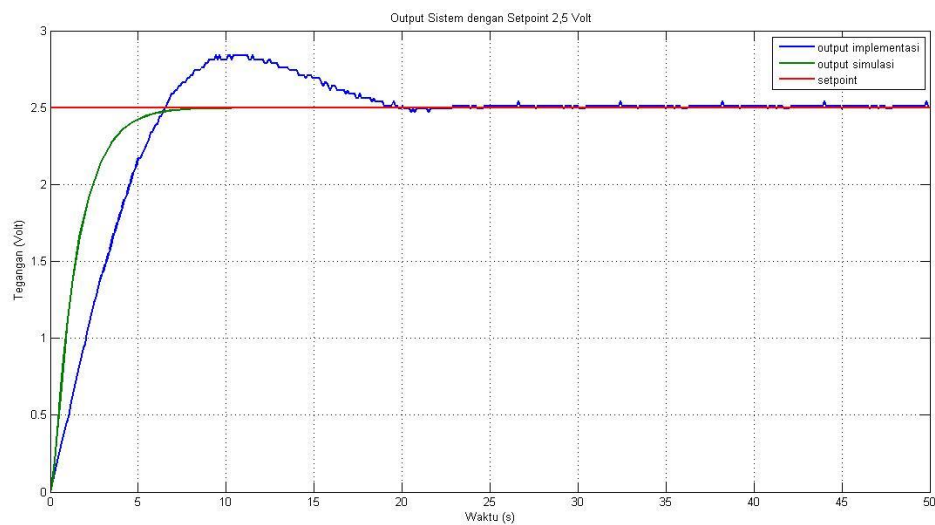
Data dengan nilai dari $op = [-5,6759; -0,1941]$, $rp = [-5,5944; -0,7279]$, $R = [-0,298149765166340 \ -0,044202544031311]$, $L = [-0,6329999998; -0,072908810000004]$, dan $M = 0,996126164$ akan diimplementasikan pada alat. Implementasi dengan mengubah-ubah setpoint ini dilakukan agar kita mengetahui bagaimana tegangan *output* sistem apabila nilai *setpoint* diubah-ubah.

Pada setpoint 1 Volt, diperoleh grafik tegangan *output* (lihat Gambar 4.15) dengan *settling time* 20 detik dengan *maximum overshoot* 8% dan *error steady state* sebesar 1%.



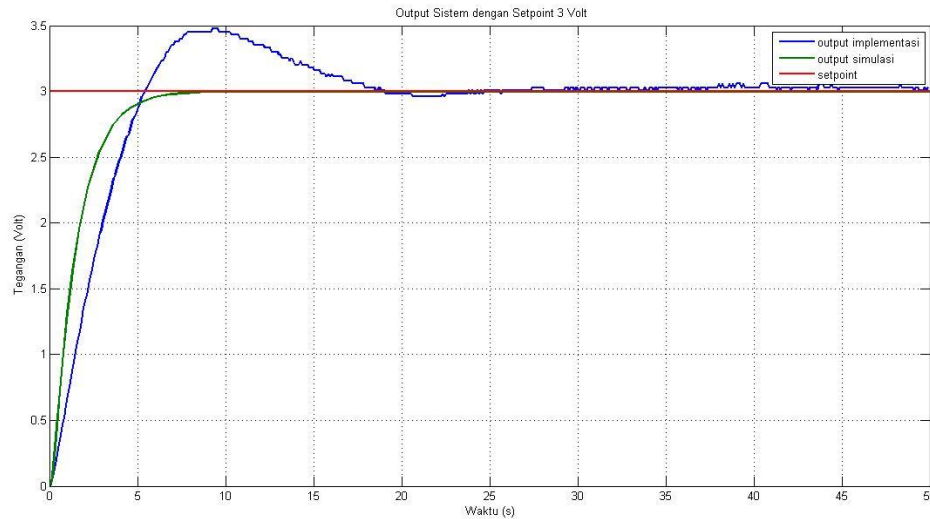
Gambar 4. 14 Tegangan *Output* sistem dengan setpoint 1 Volt

Pada *setpoint* 2,5 Volt, diperoleh grafik tegangan *output* (lihat Gambar 4.16) dengan *settling time* 20 detik dengan *maximum overshoot* 12% dan *error steady state* sebesar 0,4%.



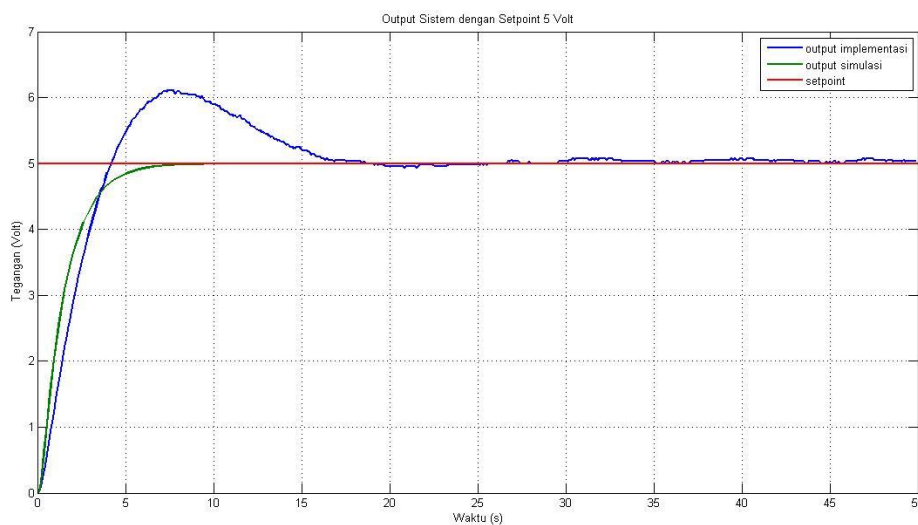
Gambar 4. 15 Tegangan *Output* sistem dengan setpoint 2,5 Volt

Pada *setpoint* 3 Volt, diperoleh grafik tegangan *output* (lihat Gambar 4.17) dengan *settling time* 18 detik dengan *maximum overshoot* 16,67% dan *error steady state* sebesar 1%.



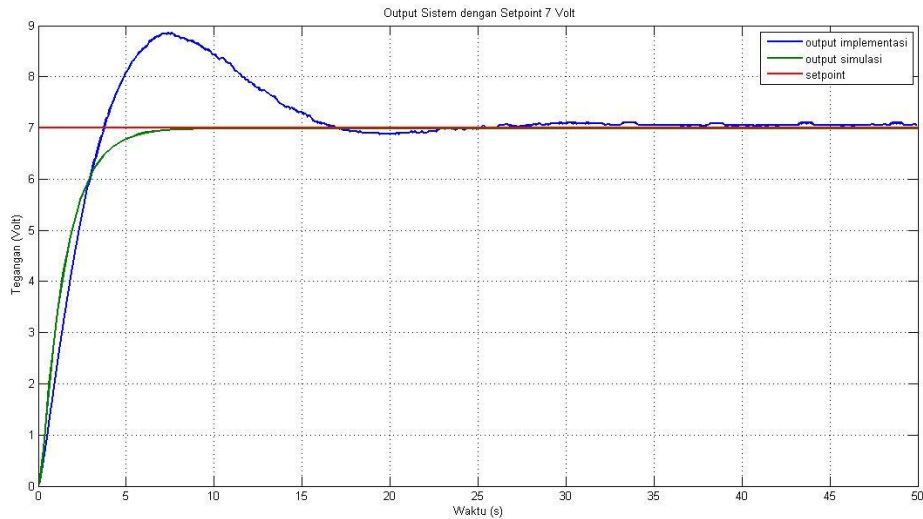
Gambar 4. 16 Tegangan *Output* sistem dengan setpoint 3 Volt

Pada *setpoint* 5 Volt, diperoleh grafik tegangan *output* (lihat Gambar 4.18) dengan *settling time* 16 detik dengan *maximum overshoot* 22% dan *error steady state* sebesar 1%.



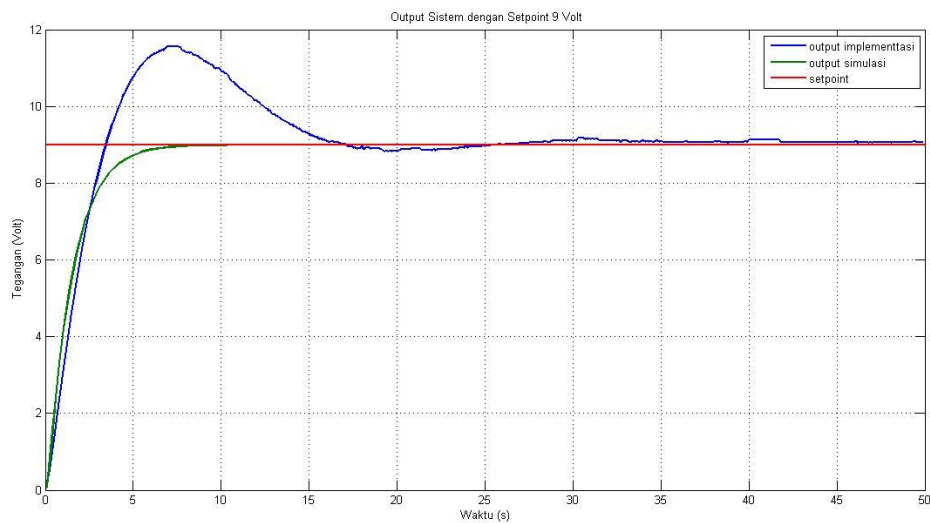
Gambar 4. 17 Tegangan *Output* sistem dengan setpoint 5 Volt

Pada *setpoint* 7 Volt, diperoleh grafik tegangan *output* (lihat Gambar 4.19) dengan *settling time* 16 detik dengan *maximum overshoot* 22% dan *error steady state* sebesar 1,4%.



Gambar 4. 18 Tegangan *Output* sistem dengan setpoint 7 Volt

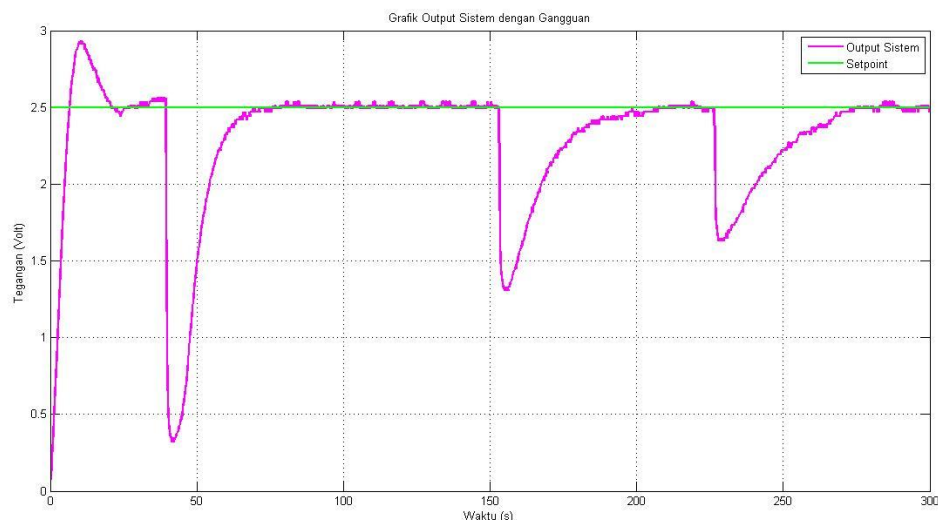
Pada *setpoint* 9 Volt, diperoleh grafik tegangan *output* (lihat Gambar 4.20) dengan *settling time* 16 detik dengan *maximum overshoot* 24% dan *error steady state* sebesar 1,1%.



Gambar 4. 19 Tegangan *Output* sistem dengan setpoint 9 Volt

4.7.2 Implementasi dengan gangguan

Implementasi dengan gangguan dilakukan dengan menggunakan setpoint 2,5 Volt, karena beban yang digunakan adalah lampu bohlam 2,5 Volt. Beban menggunakan 3 buah lampu yang dinyalakan satu-persatu dan kemudian menjadi 3 buah lampu bohlam 2,5 volt yang dinyalakan bersama-sama. Grafik tegangan *output* dengan gangguan dapat dilihat dalam Gambar 4.21.



Gambar 4. 20 Tegangan *output* sistem dengan gangguan dengan setpoint 2,5 Volt

Dari grafik tegangan *output* sistem dapat dilihat, bahwa waktu yang dibutuhkan sistem untuk mencapai *steady state* adalah 20 detik dengan *maximum overshoot* sebesar 12% dan nilai *error steady state* sebesar 0,4%.

Setelah dilakukan implementasi dengan gangguan, dapat dilihat bahwa waktu yang dibutuhkan oleh sistem untuk mencapai *steady state* setelah mendapatkan gangguan penyalan lampu yang pertama adalah 30 detik tanpa *overshoot* dengan nilai *error steady state* sebesar 0,4%. Saat dua lampu yang menyala, waktu yang dibutuhkan untuk mencapai *steady state* adalah 40 detik tanpa *overshoot* dengan nilai *error steady state* sebesar 0,4%. Dan saat tiga lampu menyala bersamaan, waktu yang dibutuhkan untuk mencapai *steady state* adalah 50 detik tanpa *overshoot* dengan nilai *error steady state* sebesar 0,4%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Berdasarkan hasil identifikasi *plant* menggunakan sinyal uji PRBS dan sintaks *ident* pada *software* matlab, fungsi alih *plant* motor generator DC 734 11 adalah
$$\frac{Y(s)}{U(s)} = \frac{4,088}{s^2+6,503s+5,291}$$
, mempunyai best fit sebesar 90,45% dan telah divalidasi.
2. Berdasarkan hasil simulasi didapatkan nilai kontroler *pole* dan *observer pole* yang optimal yaitu $op=[-5,6759;-0,1941]$, $rp=[-5,5944;-0,7279]$ dan menghasilkan nilai matriks $R=[-0,298149765166340 \ -0,044202544031311]$, $L=[-0,6329999998;-0,072908810000004]$, dan $M= 0,996126164$.
3. Simulasi *output* *plant* saat diberi *input unit step* setelah diberi kontroler struktur *Output Feedback Control* mempunyai *settling time* 7 detik tanpa *error steady state* dan tanpa *overshoot*. Hasil dari simulasi pada simulink tanpa gangguan, *setpoint* sebesar 2,5 Volt menghasilkan *output* *plant* dengan *settling time* 7 detik, dan tidak mempunyai *error steady state*.
4. Hasil dari implementasi tanpa gangguan menghasilkan *output* *plant* dengan *settling time* 20 detik dengan *maximum overshoot* 12% dan *error steady state* sebesar 0,4%. Implementasi dengan gangguan yaitu dengan menyalakan lampu pertama, tegangan *output plant* memiliki *settling time* 30 detik. Selanjutnya beban ditambah dengan menyalakan lampu kedua, *settling time* 40 detik. Kemudian beban ditambah dengan menyalakan lampu ketiga, *settling time* 50 detik. Ketiganya tanpa *overshoot* dengan nilai *error steady state* sebesar 0,4%.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah dengan menentukan parameter lain yang bisa digunakan pada sistem agar *output* saat implementasi tidak memiliki *overshoot*.

DAFTAR PUSTAKA

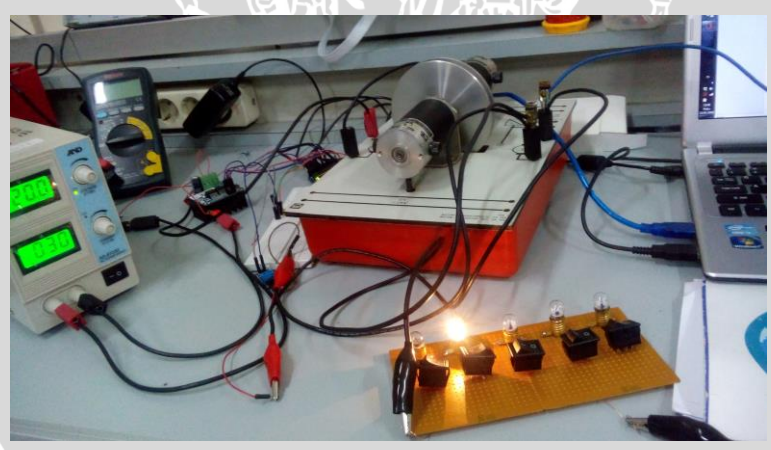
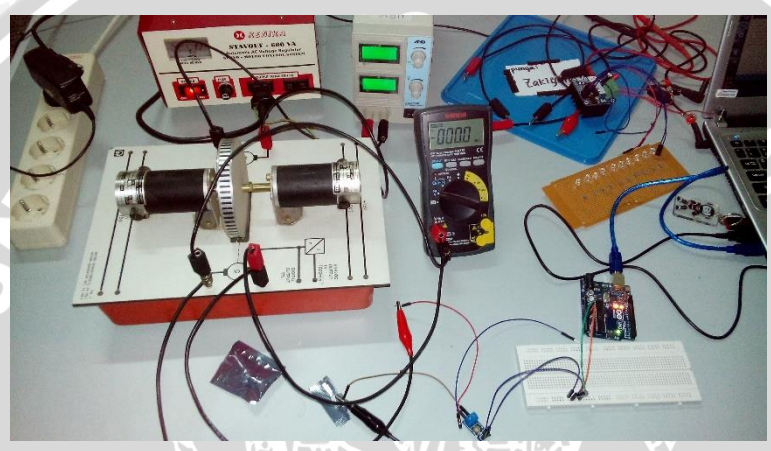
- Al-Farisi, R. 2011. *Implementasi Kontroler Proporsional Integral Differensial (PID) Untuk Pergerakan Robot Lengan ROB2R Pada Simulasi Pemindahan Objek*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Arismunandar, A. 1982. *Buku Pegangan Teknik Tenaga Listrik*. Jakarta: Penerbit PT. Pradnya Paramita.
- Ardyani, F. 2013. *Sistem Pengendalian Level Cairan Tinta Printer Epson C90 Sebagai Simulasi Pada Industri Percetakan Menggunakan Kontroler PID*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Bolton, W. 2004. *Sistem Instrumentasi Dan Sistem Kontrol*. Jakarta : Penerbit Erlangga.
- Houpis, D'Azzo. 1995. *Linear Control Systems Analysis and Design*. USA: McGraw-Hill, Inc.
- James, T. & Lesie, P. 1983. *Industrial Electronics*. Amerika : United of America.
- Kwakernaak, H., Sivan, R. 2011. *Linear Optimal Control Systems*. United States: Wiley Interscience.
- Landau, Ion. D. Zito, Gianluc. 2006. *Digital Control System*. German : Springer.
- Lewis, Syrmos. 1995. *Optimal Control*. USA: John Wiley & Sons, Inc.
- Leybold Didactic. Instruction Sheet Motor generator set, 24 V. http://gdedel95.free.fr/IUT/EEP/TPEEP_1ann%E9e/73411e.pdf. (diakses 5 Maret 2016)
- Ogata, K. 1997. *Teknik Kontrol Automatik*. terjemahan: Edi Laksono Ir. Jakarta : Penerbit Erlangga.
- Ogata, K. 2010. *Modern Control Engineering*, Pearson Education, Inc., publishing as Prentice Hall, One Lake Street, Upper Saddle River, New Jersey 07458. Fifth edition.
- Prayogo, R. 2014. *Aplikasi Kontrol Optimal LQG untuk Pengontrolan Water Level Steam Drum Boiler*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Ramhani, M. 2008. *Rangkaian Listrik*. Bandung: Penerbit Erlangga
- Rusli, M. 2015. *Dasar Teknik Kontrol*. Malang: Penerbit UB Press
- Soemarwanto. 2010. *Dasar Konversi Energi Elektrik*. Malang
- Weedy, B.M. 1988. *Sistem Tenaga Listrik*. Terjemahan: Daliati H. Gulo. Southampton: Penerbit Aksara Persada Indonesia
- Zuhal. 2004. *Dasar Tenaga Listrik*. Bandung : ITB Press

LAMPIRAN I

FOTO ALAT



FOTO ALAT



UNIVERSITAS BRAWIJAYA

LAMPIRAN II

LISTING PROGRAM



PROGRAM UTAMA

```

#define pwm 6
float MV;
float voltage,Ts;
float setpoint;
float
w,u,feedback,v1,v2,z,voltagebaru,feedback11,feedback12,xdot1,xdot2,feedback21,feed
back22,x1,x2,x1_last,x2_last;
float L1,L2,R1,R2,M;

void setup()
{
  pinMode(8, OUTPUT);
  pinMode(pwm,OUTPUT);
  pinMode(7, OUTPUT);
  digitalWrite(8, HIGH);
  digitalWrite(7, HIGH);
  Serial.begin(9600);

  setpoint = 2;
  Ts=0.1;
  L1=-0.5731;
  L2=-0.447;
  R1=-0.2981;
  R2=-0.0442;
  M=1.0038;

  noInterrupts();
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 64753; //49911;//45536; //((max preload timer-konstanta)// preload timer
max 65536-16MHz/256/2Hz

```

```

TCCR1B = 0b00000101; // prescaler 000 - 101 // 8,64,256,1024
TIMSK1 = 0x05; // enable timer overflow interrupt
interrupts(); // enable all interrupts
}

ISR(TIMER1_OVF_vect) {
    TCNT1 = 64753; //49911; //45536; // preload timer (preload timer-konstanta)

    w=setpoint*M;
    feedback=(R1*x1)+(R2*x2);

    u=w-feedback;
    MV=u*15.5747;
    sensor();
    kontroler();
}

void sensor()
{
    int sensorValue = analogRead(A0);
    // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
    voltage = sensorValue * (5.0 / 1023.0);
    // print out the value you read:
    Serial.print(z);
    Serial.print("\t");
    Serial.print(voltage);
    Serial.print("\n");
}

void kontroler()

```

```
{  
  v1=0*u;  
  v2=4.088*u;  
  z=voltage-voltagebaru;  
  
  feedback11=L1*z;  
  feedback12=L2*z;  
  
  feedback21=x2;  
  feedback22=(-5.291*x1)+(-6.503*x2);  
  
  xdot1=v1+feedback11+feedback21;  
  xdot2=v2+feedback12+feedback22;  
  
  x1=x1_last+(Ts*xdot1);  
  x2=x2_last+(Ts*xdot2);  
  
  x1_last=x1;  
  x2_last=x2;  
  
  voltagebaru=x1;  
}  
void loop()  
{  
  analogWrite(pwm,MV);  
}
```



UNIVERSITAS BRAWIJAYA

LAMPIRAN III

DATASHEET

