



TESIS-TF185471

**PERANCANGAN FAULT TOLERANT CONTROL
PADA PENGENDALIAN *SPEED SENSORLESS*
MOTOR DC**

**DIYAJENG LULUK KARLINA
NRP. 02311850010009**

**Dosen Pembimbing
Dr. Katherin Indriawati, S.T., M.T.**

**PROGRAM MAGISTER
BIDANG KEAHLIAN REKAYASA INSTRUMENTASI INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI DAN REKAYASA SISTEM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020**

Halaman ini sengaja dikosongkan



THESIS-TF185471

DESIGN FAULT TOLERANT CONTROL FOR SPEED SENSORLESS OF DC MOTOR

**DIYAJENG LULUK KARLINA
NRP. 02311850010009**

Supervisor
Dr. Katherin Indriawati, S.T., M.T.

**MASTER PROGRAM
INDUSTRIAL INSTRUMENTATION ENGINEERING
ENGINEERING PHYSICS DEPARTMENT
FACULTY OF INDUSTRIAL TECHNOLOGY AND SYSTEM ENGINEERING
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2020**

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh :

DIYAJENG LULUK KARLINA

NRP. 02311850010009

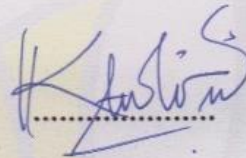
Tanggal Ujian : 6 Januari 2020

Periode Wisuda : Maret 2020

Disetujui oleh :

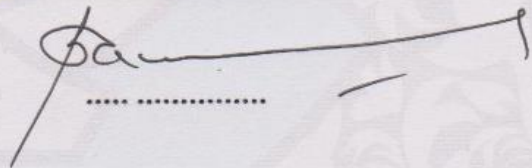
Pembimbing:

1. Dr. Katherin Indriawati, S.T., M.T.
NIP. 19760523 200012 2 001



Penguji:

1. Dr. Ir. Purwadi Agus Darwito, M.Sc.
NIP. 19620822 198803 1 001
2. Dr. Bambang L. W., S.T., M.T.
NIP. 19690507 199512 1 001



Kepala Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem



Dr. Suyanto, S.T., M.T.

NIP: 19711113 199512 1 002

Halaman ini sengaja dikosongkan

PERANCANGAN FAULT TOLERANT CONTROL PADA PENGENDALIAN *SPEED SENSORLESS* MOTOR DC

Nama Mahasiswa : Diyajeng Luluk Karlina
NRP : 02311850010009
Pembimbing : Dr. Katherin Indriawati, S.T., M.T

ABSTRAK

Dalam penelitian ini dikembangkan metode mengenai sistem kontrol tanpa sensor kecepatan (*speed sensorless*) dengan menggunakan metode *observer* pada sistem servo motor DC. Gangguan pada motor DC dapat terjadi akibat adanya pengaruh torsi beban. Oleh karena itu, diperlukan sistem kontrol yang mampu menoleransi gangguan berupa torsi beban. Sistem kontrol yang digunakan yaitu *fault tolerant control* (FTC). Pemodelan motor DC dilakukan menggunakan identifikasi sistem metode parametrik. Langkah pertama yang dilakukan adalah merancang kontroler, kontroler yang digunakan adalah *state feedback* dengan *integrator* menghasilkan nilai K_p sebesar 14,136 dan $-6,757$ sedangkan K_i sebesar 89,285. Langkah berikutnya adalah merancang *observer* sebagai *softsensor* yang berfungsi untuk mengestimasi kecepatan. Berdasarkan hasil penelitian bahwa *observer* yang dirancang memiliki nilai estimasi kecepatan yang sama dengan hasil aktual, sehingga *observer* sudah mampu bekerja sebagai *softsensor*. Dengan karakteristik respon *settling time* sebesar 105 milidetik dan tidak terjadi *error steady-state* maupun *maximum overshoot*. Ketika terjadi penambahan torsi beban, sistem motor DC mengalami *maximum undershoot* yang menunjukkan terjadinya kesalahan atau penyimpangan. Namun dengan menggunakan sistem FTC, kecepatan dari motor DC dapat kembali ke nilai *setpoint* yang telah ditentukan dengan waktu yang diperlukan sebesar 1225 milidetik.

Kata kunci: Fault Tolerant Control, Motor DC, Observer, Sensorless

Halaman ini sengaja dikosongkan

DESIGN PASSIVE FAULT TOLERANT CONTROL (FTC) FOR SPEED SENSORLESS OF DC MOTOR

Name : Diyajeng Luluk Karlina
NRP : 02311850010009
Supervisors : Dr. Katherin Indriawati, S.T., M.T

ABSTRACT

In this study developed a method regarding the control system without a sensorless speed sensor using the Observer method on the DC motor servo system. Interference with a DC motor can occur due to the influence of load torque. Therefore, a control system is required to tolerate interference in the form of load torque. The control system used is fault tolerant control (FTC). Modelling of DC Motors is carried out using parametric system identification methods. The first step is to design a controller, the controller used is the state feedback with the integrator generating a K_p value of 14.136 and -6.757 while the K_i amounted to 89.285. The next step is to design the observer to estimate the speed (sensorless). Based on the research results that the observer is designed to have the same speed estimation value as the actual result, so that the observer is able to work as a softsensor. With the slurries time response characteristic of 105 milliseconds and no steady-state and maximum overshoot error. When the addition of load torque occurs, DC motor systems undergo a maximum undershoot indicating errors or irregularities. However by using the FTC system, the speed of the DC motor can return to a specified setpoint value with a required time of 1225 milliseconds.

Keywords: Fault Tolerant Control, DC Motor, Observer, Sensorless

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat, hidayat, serta petunjuk-Nya sehingga penulis dapat menyelesaikan tesis pada jenjang magister yang berjudul “*Perancangan Fault Tolerant Control pada Pengendalian Speed Sensorless Motor DC*”. Penulis telah banyak mendapatkan bantuan dari berbagai pihak dalam menyelesaikan tugas akhir ini. Atas berbagai bantuan itu penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Suyanto, S.T., M.T selaku kepala departemen Teknik Fisika yang telah memberikan petunjuk, ilmu serta bimbingan selama menempuh pendidikan di jurusan Teknik Fisika
2. Ibu Dr. Katherin Indriawati, S.T., M.T selaku dosen pembimbing yang telah sabar memberikan motivasi, petunjuk, bimbingan, serta ilmu yang sangat bermanfaat dalam terselesainya tesis ini.
3. Bapak Dr. Ir. Syamsul Arifin, M.T selaku dosen wali yang telah membimbing penulis selama perkuliahan.
4. Kedua orang tua (Bapak Agus Riyanto dan Ibu Nurlia). Terimakasih atas segala cinta, kasih sayang, doa, perhatian, seta dukungan moril dan materiil yang telah diberikan.
5. Teman-teman Teknik Fisika angkatan 2014 yang memberikan dukungan dan semangat kepada penulis.
6. Teman-teman seperjuangan kuliah S2 dan pejuang Tesis yang telah berbagi ilmu dan pengalaman.
7. Teman – teman ruang kaca yang selalu memberikan saran dan dukungan bagi penulis.
8. Semua pihak yang telah mendukung dan memberikan doa yang tidak bisa penulis sebutkan satu per satu.

Penulis menyadari bahwa penulisan laporan tesis ini tidaklah sempurna. Oleh karena itu, sangat diharapkan kritik dan saran yang membangun dari semua pihak sehingga mencapai sesuatu yang lebih baik lagi. Penulis juga berharap semoga laporan ini dapat menambah wawasan yang bermanfaat bagi pembaca.

Surabaya, Januari 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian	4
TINJAUAN PUSTAKA	5
2.1 Sistem Motor DC	5
2.2 State Feedback dengan Integrator	9
2.3 <i>Speed sensorless</i>	10
2.4 Fault Tolerant Control	12
2.5 Kesalahan Pada Sistem	16
METODOLOGI PENELITIAN	19
3.1 Pengambilan Data Sistem Servo Motor DC	20
3.2 Pemodelan Sistem Servo Motor DC	21
3.3 Validasi	23
3.4 Perancangan Sistem Kontrol <i>State Feedback</i>	24
3.5 Perancangan <i>Observer</i>	26
3.6 Integrasi <i>Observer</i> dengan FTC	27
3.7 Uji Performansi FTC	28
HASIL DAN PEMBAHASAN	29
4.1 Validasi Data	29
4.2 Uji Traking Kontroler <i>State Feedback</i>	31

4.3 Uji Estimasi <i>Observer</i>	33
4.4 Uji Performansi	36
KESIMPULAN DAN SARAN	41
5.1 Kesimpulan	41
5.2 Saran	41
DAFTAR PUSTAKA	43
LAMPIRAN	47
BIODATA PENULIS	67

DAFTAR GAMBAR

Gambar 2.1 Motor DC (Hudaya, 2013).....	6
Gambar 2.2 Sistem Servo Modular MS150 DC (Manual Feedback)	6
Gambar 2.3 Rangkaian Servo Amplifier (Hughes dan Drury, 2013).....	8
Gambar 2.4 Diagram blok kontrol state feedback dengan integrator (Ogata, 1997)	9
Gambar 2.5 Analisa Respon Steady State (Ogata, K, 1997)	10
Gambar 2.6 Blok observer (Hussein dan Ghasm, 2017)	12
Gambar 2.7 Klasifikasi FTC (Oku, 2016)	12
Gambar 2.8 Skema Sistem AFTC (M. Dybkowski dan K.Klimkowski, 2017) ..	13
Gambar 2.9 Prinsip Kerja PFTC (M. Dybkowski dan K.Klimkowski, 2017).....	14
Gambar 3.1 Diagram Alir Metode Penelitian	19
Gambar 3.2 Diagram blok set up sistem servo modular MS150 DC	20
Gambar 3.3 Respon tegangan tachogenerator	22
Gambar 3.4 Respon arus.....	23
Gambar 3.5 Diagram blok pengendalian kecepatan motor DC dengan state feedback with integrator.....	24
Gambar 3.6 Struktur extended state observer	27
Gambar 3.7 Diagram blok Fault Tolerant Control pada Servo Motor DC.....	27
Gambar 4.1 Respon hasil validasi model kecepatan	29
Gambar 4.2 Respon hasil validasi model arus	30
Gambar 4.3 Uji tracking kontrol state feedback dengan integrator secara simulasi	32
Gambar 4.4 Uji tracking kontrol state feedback dengan integrator secara real ...	32
Gambar 4.5 Hasil estimasi kecepatan secara simulasi	33
Gambar 4.6 Hasil estimasi arus secara simulasi	34
Gambar 4.7 Hasil estimasi kecepatan secara real	34
Gambar 4.8 Hasil estimasi arus secara real	35
Gambar 4.9 Respon output dengan penambahan torsi beban 1 Nm secara simulasi	36
Gambar 4.10 Respon output dengan penambahan torsi beban 10 Nm secara simulasi	37
Gambar 4.11 Respon output dengan penambahan torsi beban 1 Nm secara real	38
Gambar 4.12 Respon output dengan penambahan torsi beban 10 Nm secara real	39

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 4.1 Perhitungan Validasi Model Kecepatan dengan Menggunakan MAPE	30
Tabel 4.2 Perhitungan Validasi Model Arus dengan Menggunakan MAPE.....	31
Tabel 4.3 Perbandingan Uji Tracking Kontroler	33
Tabel 4.4 Perbandingan Hasil Estimasi Kecepatan	35
Tabel 4.5 Perbandingan Hasil Estimasi Arus	36
Tabel 4.6 Karakteristik Respon Output dengan Torsi Beban 1 Nm Secara Simulasi	37
Tabel 4.7 Karakteristik Respon Output dengan Torsi Beban 10 Nm Secara Simulasi	37
Tabel 4.8 Karakteristik Respon Output dengan Torsi Beban 1 Nm Secara Real .	38
Tabel 4.9 Karakteristik Respon Output dengan Torsi Beban 10 Nm Secara Real	39

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia industri penggunaan jenis motor banyak digunakan untuk bidang transportasi berbasis listrik. Salah satu jenis motor yang digunakan untuk transportasi berbasis listrik yaitu motor DC. Motor DC lebih sering digunakan karena konsumsi dayanya rendah dan fleksibilitasnya tinggi (dalam artian kecepatan, laju dan arah putar) dapat diatur dengan mudah (Hilairret, 2007). Pada kendaraan listrik, motor DC bekerja sebagai sistem servo dengan kecepatan motor dipertahankan konstan sesuai kebutuhan. Dalam hal ini, informasi kecepatan sangat dibutuhkan sehingga dipasang sensor kecepatan. Namun sensor kecepatan memiliki keterbatasan resolusi dan perawatannya yang mahal. Oleh karena itu, mulai dikembangkan metode mengenai sistem kontrol tanpa sensor kecepatan (*speed sensorless*) dengan menggunakan informasi dari arus dan tegangan.

Selain itu, diharapkan dalam penerapan sistem servo motor DC tidak pernah mengalami gangguan. Namun, gangguan dapat terjadi secara mendadak yang akibatnya dapat mempengaruhi jalannya suatu sistem. Gangguan yang dapat terjadi pada motor DC yaitu adanya pengaruh torsi beban. Adanya pengaruh torsi beban dapat mengakibatkan kesalahan. Kesalahan merupakan bentuk penyimpangan yang tidak diinginkan dari kondisi standar yang biasa diterima. Sehingga diperlukan sistem kontrol yang mampu menoleransi kesalahan. Sistem kontrol ini dikenal sebagai *fault tolerant control* (FTC) dan merupakan bidang penelitian baru yang sedang berkembang pada bidang ilmu kontrol (J. Jiang dan X. Yu, 2012)

FTC merupakan sebuah sistem kontrol yang dapat menjaga kestabilan sistem keseluruhan meskipun terjadi kesalahan pada komponen sistem. FTC adalah sistem penting dalam aplikasi *safety* karena mampu mengatasi kesalahan minor (seperti kesalahan pada komponen aktuator dan sensor) yang akan menyebabkan kesalahan mayor bahkan kegagalan jika dibiarkan. FTC dapat diklasifikasikan 2 tipe yaitu *Passive Fault Tolerant Control* (PFTC) dan *Active Fault Tolerant Control* (AFTC) (J. Jiang dan X. Yu, 2012). PFTC didesain untuk menjadikan sistem kontrol bersifat robust terhadap kesalahan tanpa memerlukan proses identifikasi kesalahan yang

terjadi. Sedangkan AFTC bekerja dengan cara mengkompensasi kesalahan menggunakan hasil estimasi kesalahan (M. Dybkowski dan K.Klimkowski, 2017).

Beberapa penelitian terdahulu sudah dilakukan, yaitu pada penelitian Wang, dkk. (2009) dengan menggunakan metode berbasis model. Metode berbasis model menggunakan persamaan tegangan dan arus pada sistem motor DC untuk menghasilkan estimasi kecepatan. Namun, pada penelitian ini tidak mempertimbangkan adanya *disturbance* yang dapat terjadi pada sistem motor DC. Penelitian ini hanya membahas mengenai sistem *speed sensorless* motor DC.

Selanjutnya pada tahun 2012 dilakukan penelitian untuk mengestimasi kecepatan pada motor DC dengan menggunakan metode *observer*. *Observer* yang digunakan pada penelitian ini yaitu *kalman filter*. Metode ini mengubah parameter sistem motor DC dari persamaan fungsi transfer ke bentuk ruang keadaan (*state space*). Hasil penelitian ini menunjukkan bahwa estimasi kecepatan yang dilakukan dengan menggunakan algoritma *kalman filter* memiliki tingkat akurasi yang baik untuk mengestimasi kecepatan dan mampu bekerja ketika adanya *noise* (Aydogmus O dan Talu F, 2012). Namun, penelitian ini hanya bertujuan untuk mengestimasi kecepatan dan menghilangkan pengaruh *noise*, dan belum meninjau kesalahan yang dapat terjadi pada komponen sistem motor DC.

Pengembangan penelitian selanjutnya dilakukan pada tahun 2013 dengan menggunakan kontrol proporsional integral (PI) dan penggunaan *sliding mode observer* untuk mengestimasi kecepatan. Penelitian ini sudah mempertimbangkan adanya kesalahan pada sensor kecepatan dengan menggunakan *fault tolerant control* (FTC). Kesalahan yang terjadi pada sensor kecepatan merupakan kesalahan mayor sehingga penggunaan sensor kecepatan dihapuskan dan digantikan dengan *observer* (Ali Akbar, 2013). Jadi, FTC dalam penelitian ini berfungsi untuk menggantikan peran sensor akibat terjadi kesalahan mayor pada sensor kecepatan. Namun demikian, penelitian ini belum mempertimbangkan adanya kesalahan minor yang dapat terjadi pada sistem motor DC. Kesalahan minor dapat terjadi karena pengaruh torsi beban sehingga menyebabkan kesalahan.

Berdasarkan penelitian terdahulu yang sudah dipaparkan sebelumnya maka diusulkan penelitian ini. Penelitian ini akan merancang FTC pada *speed sensorless* sistem servo motor DC. Metode *speed sensorless* diperlukan untuk mengestimasi

kecepatan dengan menggunakan *observer* dari data pengukuran arus dan tegangan. Sedangkan metode FTC (*fault tolerant control*) digunakan untuk mentoleransi kesalahan yang terjadi akibat adanya perubahan parameter pada sistem servo motor DC. Metode FTC *speed sensorless* dipilih karena mampu mengatasi kelemahan dari penggunaan sensor kecepatan dan mentoleransi kesalahan yang terjadi pada motor DC, sehingga sistem dapat berjalan lebih optimal dan efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang maka rumusan masalah dari penelitian tesis adalah sebagai berikut,

- Bagaimana merancang *observer* untuk aplikasi *speed sensorless* pada sistem servo motor DC?
- Bagaimana merancang FTC pada sistem servo motor DC saat terjadi kesalahan?

1.3 Tujuan

Berdasarkan latar belakang dan perumusan masalah yang telah dipilih maka tujuan penelitian yang ingin dicapai dalam pengerjaan thesis ini adalah sebagai berikut,

- Merancang pengendalian kecepatan tanpa sensor pada sistem servo motor DC yang mampu mengestimasi kecepatan.
- Merancang sistem FTC yang mampu menjaga performansi sistem servo motor DC ketika terjadi kesalahan.

1.4 Batasan Masalah

Adapun terdapat beberapa batasan masalah yang dipilih dalam pengerjaan thesis ini antara lain yaitu,

- Motor DC yang digunakan adalah sistem servo modular MS150 DC dengan aktuator adalah servo amplifier dan sensor adalah tachometer.
- Kesalahan yang terjadi pada motor DC adalah kesalahan minor.
- Kesalahan berupa adanya pengaruh torsi beban pada sistem motor DC.

1.5 Manfaat Penelitian

Penelitian ini memiliki urgensi secara praktis karena motor DC banyak ditemui pada perangkat bantu kehidupan manusia. Pada penggunaan motor DC untuk *plant* beresiko tinggi (keamanan maupun biaya), strategi FTC sangat dibutuhkan untuk menghindari dan/atau mengurangi dampak kerugian.

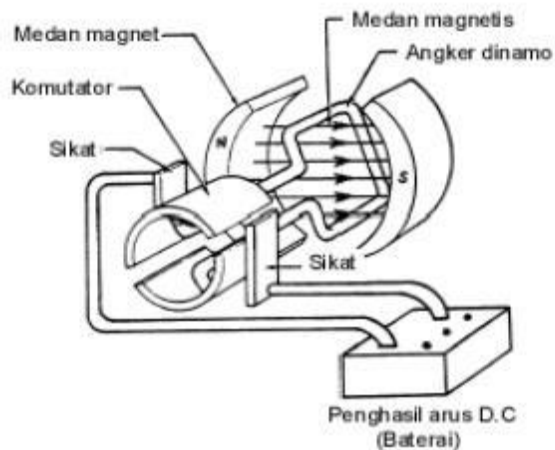
BAB II

TINJAUAN PUSTAKA

2.1 Sistem Motor DC

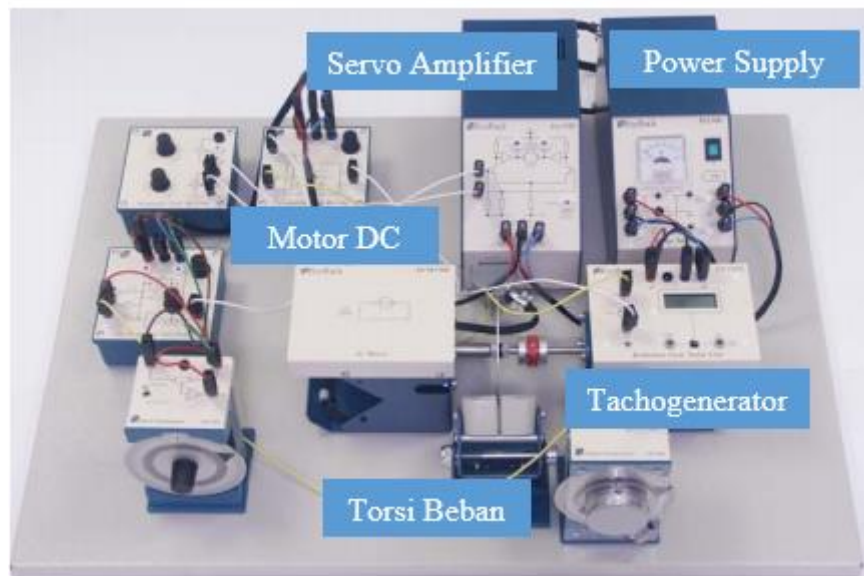
Motor DC merupakan perangkat elektromagnetis mengubah energi listrik menjadi energi mekanik. Motor DC sendiri merupakan salah satu jenis motor yang menggunakan energi listrik arus searah atau direct current untuk kemudian diubah menjadi gerakan rotasional. Penggunaan motor DC lebih mendominasi daripada penggunaan motor AC karena kecepatan motor DC mudah diatur. Aplikasi motor DC sebagai aktuator sangat luas mulai dari bidang industri sampai bidang transportasi. Bahkan motor DC digunakan sebagai penggerak utama pada berbagai kebutuhan peralatan rumah tangga seperti, kipas angin, mesin jahit, pompa, hingga mainan anak – anak.

Motor DC terdiri atas beberapa bagian yaitu kumparan jangkar (*armature*), magnet permanen, komutator (*commutator*), dan sikat (*brush*). Stator adalah bagian yang tidak bergerak (statis) dan rotor adalah bagian yang berputar. Prinsip kerja sistem motor DC yaitu magnet permanen sebagai stator tersusun dari dua magnet yang memiliki kutub berbeda yang saling berhadapan. Jika terjadi putaran pada kumparan jangkar dalam medan magnet maka akan timbul tegangan yang berubah – ubah arah pada setiap setengah putaran, sehingga dapat disebut tegangan bolak balik. Magnet permanen berfungsi sebagai penghasil medan magnet. Komutator merupakan bagian dari rotor yang terletak pada kedua ujung kumparan jangkar yang berfungsi untuk mengumpulkan arus induksi dari jangkar kemudian mengubahnya menjadi arus searah. Sedangkan sikat berfungsi untuk menyalurkan arus listrik dari sumber di luar motor ke dalam kumparan jangkar. Kumparan jangkar sebagai rotor motor digambarkan sebagai sebuah kawat yang memiliki bentuk persegi panjang dan berfungsi untuk merubah energi listrik menjadi energi mekanik dalam bentuk gerak putar (*Electric Motors and Variable Frequency Drives Handbook*).



Gambar 2.1 Motor DC (Hudaya, 2013)

Sistem servo modular MS150 DC adalah rangkaian blok elektronik yang digunakan sebagai pengendalian kecepatan posisi pada motor DC jenis magnet permanen. Sistem servo modular MS150 DC terdiri dari beberapa bagian, antara lain yaitu *power supply*, servo amplifier, unit motor DC, *reduction gear*, dan tachometer unit.



Gambar 2.2 Sistem Servo Modular MS150 DC (Manual Feedback)

Modular Servo System MS -150 merupakan suatu sistem modular buatan FEEDBACK yang akan digunakan dalam penelitian tesis ini. Sistem servo modular

MS150 DC adalah blok rangkaian yang digunakan untuk pengendalian kecepatan dan posisi dari motor DC jenis magnet permanen. Pada tesis ini yang diamati adalah pengendalian kecepatan motor DC dengan sensor adalah tachogenerator dan aktuator adalah servoamplifier. Sistem servo modular MS 150 DC pada Gambar 2.2 secara umum terdiri dari bagian yaitu sebagai berikut,

- **Power supply**
Modul feedback tipe PS150E yaitu sumber tegangan yang berfungsi untuk menyuplai rangkaian pengaturan motor DC. Modul ini menghasilkan tegangan keluaran sekitar 24 V. Terdapat ammeter yang digunakan sebagai penunjuk jumlah arus listrik yang mengalir ke motor. Selain itu terdapat tombol hijau sebagai tombol power.
- **Servoamplifier**
Modul servoamplifier yang digunakan adalah modular feedback servo amplifier SA150D yang terdiri dari rangkaian transistor. Rangkaian transistor tersebut dapat menggerakkan motor DC dengan dua arah putaran. Servo amplifier memiliki sistem proteksi yaitu overloading protection yang bertujuan sebagai sistem perlindungan motor dari arus yang berlebihan.
- **Tachogenerator**
Modular feedback tachogenerator GT150X berfungsi sebagai konversi besaran mekanik menjadi besaran listrik sehingga putaran motor dapat diukur. Prinsip kerja tachogenerator adalah bagian rotor pada motor dikoperl dan putaran motor digunakan untuk memutar rotor yang terdapat pada tachogenerator tersebut sehingga putaran rotor akan menghasilkan tegangan keluaran.
- **Motor DC**
Motor DC yang digunakan yaitu motor DC dengan tipe DCM150F yang terdiri dari motor DC magnet permanen, poros motor atau rotor yang diperpanjang, dan piringan magnet permanen.

2.1.1 Tachogenerator

Sensor yang digunakan pada sistem servo motor DC adalah tachogenerator. Tachogenerator merupakan salah satu sensor kecepatan yang biasanya digunakan

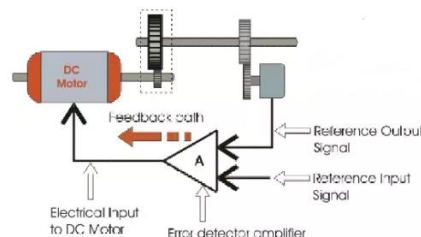
pada sistem pengendalian motor. Tachogenerator adalah generator listrik yang digunakan untuk mendeteksi kecepatan berputar peralatan listrik. Tachogenerator memberikan tegangan sebanding dengan kecepatan rotasi. Prinsip kerja tachogenerator yaitu mengubah energi mekanik menjadi energi listrik. Kemampuan deteksi kecepatan dari tachogenerator tergantung pada kekuatan magnet permanen. Jangkar pada tachogenerator berputar dengan kecepatan yang diukur pada bidang magnet permanen.

Prinsip kerja tachogenerator yaitu ketika konduktor bergerak di medan magnet, EMF terjadi dalam konduktor. Besar nilai EMF bergantung pada nilai fluks dan kecepatan putar dari shaft. Timbulnya EMF tergantung pada kekuatan fluks magnet, jumlah lilitan konduktor, dan sudut perpotongan fluks magnet dengan konduktor. Armature berputar diantara dua magnet permanen yang memiliki kutub berlawanan. Berputarnya armature mengakibatkan timbulnya GGL dalam koil. Arus yang terjadi pada kumparan jangkar adalah arus bolak – balik. Komutator mengubah arus bolak – balik menjadi arus searah dengan bantuan brush.

Penggunaan tachogenerator sebagai sensor kecepatan memiliki beberapa kelebihan dan kekurangan yang mungkin ditimbulkan. Kelebihan dari tachogenerator yaitu mampu menjaga inersia turun yang dapat diatasi dengan penggunaan sikat, sedangkan kelemahannya antara lain yaitu penggunaan sikat dalam upaya menjaga inersia dapat mengalami aus (Akram Ahmad, 2014).

2.1.2 Servo Amplifier

Servo amplifier merupakan aktuator yang digunakan pada sistem motor DC. Servo amplifier merupakan sebuah penguat yang mengolah sinyal error untuk mengoreksi perbedaan antara sinyal input dan sinyal feedback sehingga dapat menghasilkan output yang diharapkan (Hughes dan Drury, 2013)



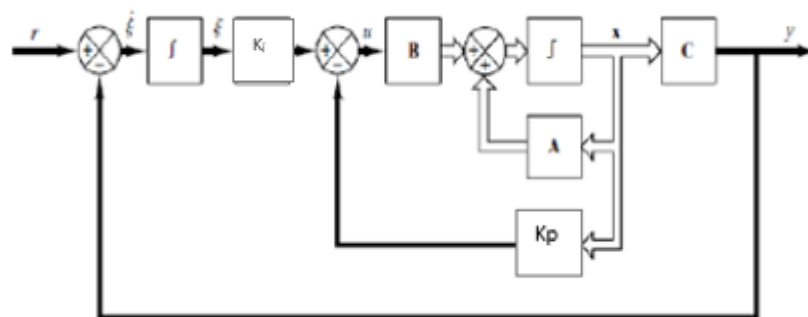
Gambar 2.3 Rangkaian Servo Amplifier (Hughes dan Drury, 2013)

Servo amplifier terdiri dari komparator yang mengolah sinyal error untuk mengoreksi perbedaan antara sinyal input (referensi) dengan sinyal umpan balik (feedback) sehingga menghasilkan output yang diharapkan. Servo amplifier merupakan sebuah unit pengatur motor yang tersusun dari transistor – transistor. Rangkaian pengukur arus motor dilengkapi dengan sistem proteksi untuk menghindari kelebihan beban (overloading).

2.2 State Feedback dengan Integrator

Sistem pengendalian adalah gabungan dari kerja suatu komponen yang berfungsi untuk mempertahankan variabel yang dikendalikan (*process variable*) pada suatu nilai tertentu (*set point*) sehingga sistem dapat berjalan stabil.

Sinyal keluaran sistem dibandingkan dengan sinyal referensi sehingga sistem kontrol tersebut disebut sistem kontrol *state feedback*. Ketika terjadi perbedaan atau penyimpangan antara sinyal keluaran dengan sinyal referensi maka sinyal tersebut dikoreksi sehingga dapat memperbaiki sinyal masukan sistem selanjutnya. Sistem pengendalian *speed sensorless* pada sistem motor DC menggunakan sistem kontrol *state feedback* dengan integrator. Penambahan integrator dilakukan guna menghilangkan *error steady state* (Ogata, 1997).



Gambar 2.4 Diagram blok kontrol *state feedback* dengan integrator (Ogata, 1997)

Berdasarkan Gambar 2.4, kontrol *state feedback* dengan integrator dapat ditulis dalam persamaan sebagai berikut,

$$u = Kp \cdot x + Ki \int e \cdot dt \quad (2.1)$$

Dimana,

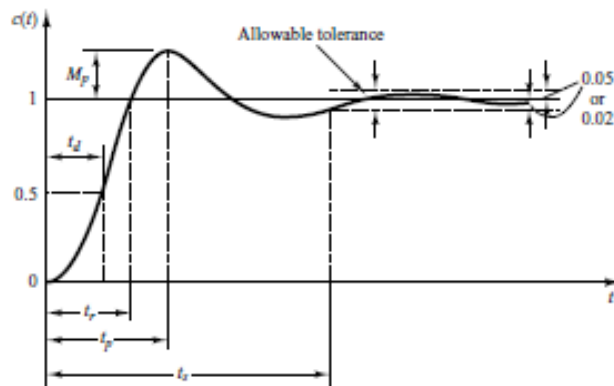
- u = input motor DC
- Kp = gain proportional
- Ki = gain integral
- x = state

Terdapat beberapa karakteristik respon dalam pengendali *state feedback* antara lain yaitu, *maximum overshoot*, *settling time*, *rise time*, *delay time*, dan *peak time*.

- *Maximum overshoot* adalah nilai puncak tertinggi dari suatu respon sistem pengukuran terhadap keadaan yang diinginkan.

$$\text{Maximum percent overshoot} = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2.2)$$

- *Settling time* adalah waktu yang dibutuhkan oleh respon sistem untuk mencapai dan tetap dalam kisaran keadaan yang diinginkan biasanya direntang $\pm 5\%$ atau $\pm 2\%$.
- *Peak time* adalah waktu puncak adalah waktu yang diperlukan untuk respons untuk mencapai puncak pertama overshoot.



Gambar 2.5 Analisa Respon Steady State (Ogata, K, 1997)

2.3 Speed sensorless

Pengukuran kecepatan tanpa sensor (*sensorless*) dengan metode berbasis model menggunakan data pengukuran arus dan tegangan (Radcliffe dan Kumar, 2015). Persamaan model matematis untuk pengukuran tegangan dengan resistor (R), induktansi (L) dan generator yang memberikan gaya gerak listrik balik (EMF) dituliskan sebagai berikut,

$$V_p = I_s R + L \frac{dI_s}{dt} + K\omega \quad (2.3)$$

Dimana V_p tegangan suplai, I_s arus suplai, (K) konstanta, (ω) kecepatan rotasi motor dan $K\omega = V_b$ adalah EMF. Jika *time constant* mekanik lebih besar daripada *time constant* elektrik maka induktansi dapat diabaikan sehingga persamaan dapat dituliskan sebagai berikut,

$$V_p = I_s R + K\omega \quad (2.4)$$

Sehingga persamaan yang digunakan untuk mengestimasi kecepatan dapat dituliskan sebagai berikut,

$$\text{Speed} \propto V_b$$

$$\text{Speed} \propto (V_p - I_s R)$$

Selain menggunakan metode berbasis model pengukuran tegangan balik, dapat juga digunakan metode *observer* untuk sistem pengukuran kecepatan tanpa sensor (*speed sensorless*).

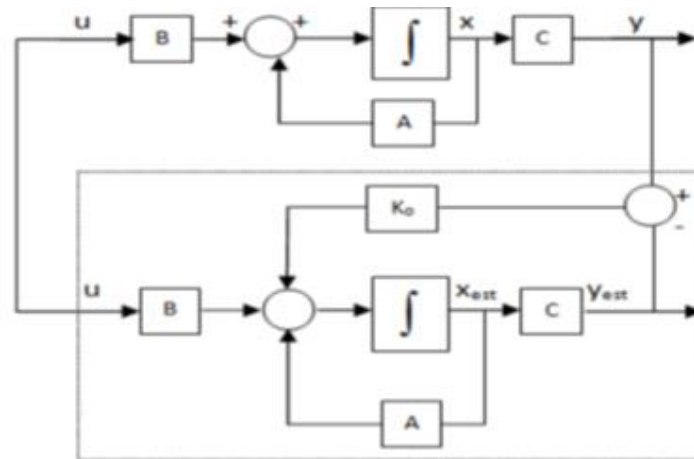
2.3.1 Observer

Observer dapat mengestimasi kecepatan dengan menggunakan persamaan model dari sistem. Adapun bentuk dari *observer* seperti yang ditunjukkan pada persamaan (2.7) dan (2.8).

$$\hat{\dot{x}} = A\hat{x} + Bu + K_o(y - \hat{y}) \quad (2.5)$$

$$\hat{y} = C\hat{x} + Du \quad (2.6)$$

Gambar 2.6 menunjukkan bahwa \hat{y} adalah hasil estimasi keluaran dari observer, dan y keluaran dari *plant*. Sedangkan K_o merupakan gain dari observer pada sistem tersebut. Nilai $(y - \hat{y})$ merupakan error dari sistem (perbedaan nilai aktual dan nilai hasil estimasi). Dan nilai error ini yang digunakan sebagai *feedback* untuk mengoreksi sistem agar model sistem mendekati dengan nilai yang diinginkan.

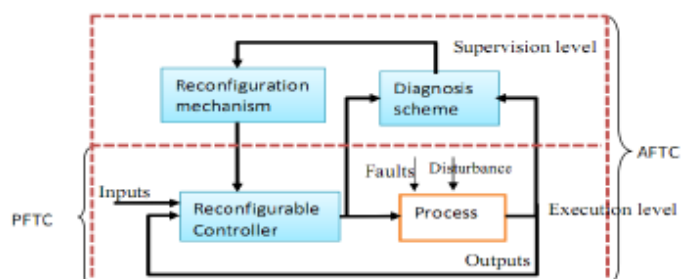


Gambar 2.6 Blok observer (Hussein dan Ghasm, 2017)

2.4 Fault Tolerant Control

Fault Tolerant Control (FTC) mendeteksi kegagalan komponen secara otomatis dengan mendesain ulang kontrol sedemikian rupa agar tetap bekerja secara optimal. Sistem FTC dirancang untuk mengakomodasi kesalahan sehingga ketika terjadi kesalahan minor pada komponen tidak menyebabkan sistem tersebut mengalami kegagalan. Sistem FTC dapat memperbaiki performa sistem secara keseluruhan karena dapat menyebabkan sistem terus beroperasi dalam durasi yang lama dengan persyaratan performa yang telah ditetapkan.

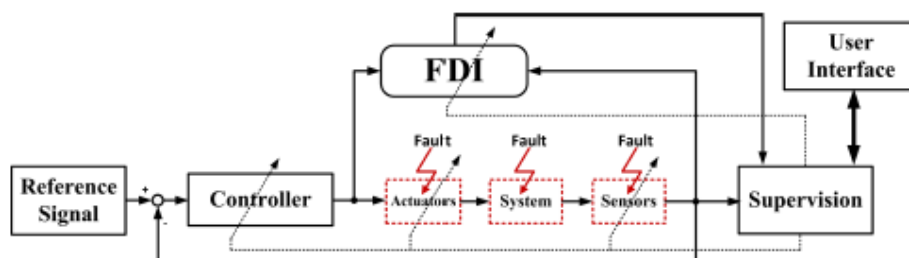
FTC dapat dibagi menjadi dua seperti yang diperlihatkan pada Gambar 2.7, yaitu *Active Fault Tolerant Control* (AFTC) dan *Passive Fault Tolerant Control* (PFTC). AFTC merupakan suatu sistem kontrol yang dirancang ulang dengan menggunakan informasi kesalahan secara on line dalam rangka mempertahankan performa serta kestabilan yang telah ditetapkan atau pada kondisi terparah dapat mencapai performa yang diijinkan.



Gambar 2.7 Klasifikasi FTC (Oku, 2016)

Sedangkan, PFTC merupakan suatu sistem kontrol yang bekerja dengan kontroler yang telah ditetapkan dan dirancang menggunakan teknik kontrol robust yang mengakibatkan sistem closed loop tidak sensitif terhadap kesalahan tertentu. Pendekatan dengan metode PFTC tidak membutuhkan informasi kesalahan secara online serta rekonfigurasi kontroler seperti pada pendekatan metode aktif. PFTC memiliki kelebihan yaitu tidak membutuhkan waktu *delay* – yang dibutuhkan oleh FDI – seperti AFTC. Selain itu unjuk kerja PFTC tidak lagi bergantung pada kesalahan deteksi yang mungkin terjadi. Namun penerapan metode pasif sangat terbatas karena hanya dapat menangani kesalahan yang menimbulkan dampak kecil saja (Zhang, Y, 2003).

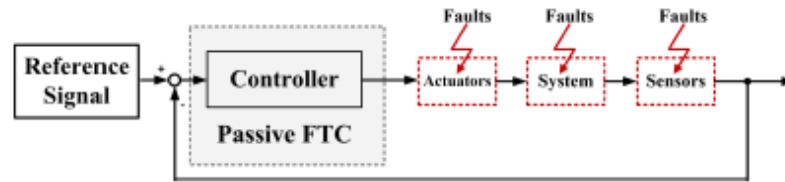
Pada Gambar 2.8, skema AFTC ditunjukkan. *Fault Detection and Isolation* (FDI) mendeteksi lokasi kegagalan serta tingkat kegagalan dari komponen tersebut. Berdasarkan input sistem, output dan informasi dari unit FDI, sistem pengawasan akan mengkonfigurasi ulang sinyal sensor dan / atau aktuator serta kontroler jika perlu untuk mengisolasi kesalahan.



Gambar 2.8 Skema Sistem AFTC (M. Dybkowski dan K.Klimkowski, 2017)

Sistem PFTC ditunjukkan pada Gambar 2.9. Parameter PFTC dibuat tetap dan didesain menggunakan algoritma kontrol robust untuk memastikan sistem kontrol mampu mengatasi kesalahan yang terjadi pada komponen sistem. Kelebihan sistem PFTC adalah besarnya kesalahan ditentukan sebelumnya sehingga sistem dapat didesain menjadi pengendali yang robust terhadap kesalahan serta tidak terjadi penundaan waktu karena tidak memerlukan tahapan Reconfigurable Controller (RC) dan Fault Detection and Isolation (FDI) seperti pada sistem AFTC. Namun kelemahannya adalah jika besaran kesalahan yang terjadi di luar nilai yang

ditentukan maka kinerja sistem akan memburuk (M. Dybkowski dan K.Klimkowski, 2017).



Gambar 2.9 Prinsip Kerja PFTC (M. Dybkowski dan K.Klimkowski, 2017)

2.4.1 *Extended State Observer*

Pada sistem FTC digunakan sebuah *observer extended state* untuk mengestimasi keadaan (*state*) dari variabel yang tidak terukur berdasarkan model matematis sistem, sehingga observer juga dapat digunakan untuk mengestimasi kesalahan yang terjadi pada suatu sistem jika kesalahan tersebut masuk dalam pemodelan. Suatu sistem umumnya dimodelkan dalam bentuk *state space*, yang dapat didefinisikan secara umum seperti persamaan sistem dibawah ini (Ogata, K, 1997):

$$\dot{x} = Ax + Bu \tag{2.7}$$

$$y = Cx + Du \tag{2.8}$$

dengan

A = matriks keadaan (*state*)

B = matriks kendali

C = matriks keluaran

D = matriks gangguan

x = vektor keadaan (*state*)

u = sinyal kontrol

y = vektor keluaran

Model matematis *observer* sama seperti model matematis pada sistem sehingga *observer* dapat mengestimasi keadaan sistem dalam bentuk persamaan ruang keadaan. Untuk mengkompensasi pengaruh torsi beban dan kesalahan yang terjadi pada motor DC pada waktu yang sama diperlukan modifikasi algoritma *observer* baru, algoritma tersebut dibentuk oleh (Indriawati, 2015). Efek dari penambahan

torsi beban direpresentasikan sebagai penambahan masukan vektor yang tidak diketahui bentuk vektor tersebut didefinisikan pada persamaan (2.13) dan (2.14), dimana matriks tersebut bertindak pada sistem dinamik dan pengukuran seperti yang direpresentasikan pada persamaan (2.9) dan persamaan (2.10) (Indriawati, 2015).

$$\dot{x} = Ax + Bu + F_a f_a \quad (2.9)$$

$$y = Cx + Du \quad (2.10)$$

Dengan:

F_a = matriks torsi beban

Tipe kesalahan sensor dimodelkan sebagai

$$s_f(t) = \alpha s(t) + \beta \quad (2.11)$$

$$f(t) = s_f(t) - s(t) \quad (2.12)$$

Dengan $s_f(t)$ adalah sinyal kesalahan dan $s(t)$ adalah sinyal actual, α dan β adalah konstanta.

Selanjutnya *state* baru, ditambahkan untuk memindahkan kesalahan sensor sebagai berikut,

$$\dot{z}(t) = A_z (y - z) \quad (2.13)$$

$$\dot{z}(t) = A_z (Cx + Du - z) \quad (2.14)$$

Untuk A_z merupakan matriks identitas (I) untuk $A_z \in I^q$, sedangkan matriks dari penambahan torsi didefinisikan sebagai $f = [f_a]$.

Dari persamaan (2.9) dapat didefinisikan matriks yang akan digunakan untuk mendapatkan matriks yang sudah diekspansi yang ditunjukkan pada persamaan berikut

$$\dot{x}_a = A_a x_a + B_a u + E_a f \quad (2.15)$$

$$y_a = C_a x_a \quad (2.16)$$

Sehingga persamaan (2.15) dan (2.16) dapat diubah ke dalam bentuk persamaan ruang keadaan menjadi,

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ AzC & -Az \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} F_a \\ 0 \end{bmatrix} \begin{bmatrix} f_a \\ f_s \end{bmatrix} \quad (2.17)$$

$$y = \dot{z} = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \quad (2.18)$$

Dengan

$$x_a = \begin{bmatrix} x \\ z \end{bmatrix}, A_a = \begin{bmatrix} A & 0 \\ AzC & -Az \end{bmatrix}, B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_a = \begin{bmatrix} 0 & I \end{bmatrix}, E_a = \begin{bmatrix} F_a \\ 0 \end{bmatrix}$$

Untuk memperoleh nilai dari *gain observer* maka diperlukan ekspansi matriks melalui persamaan ruang keadaan baru yang ditunjukkan pada persamaan (2.19),

$$\dot{\hat{x}}_a = A_a \hat{x}_a + B_a u + E_a \begin{bmatrix} \hat{f}_a \end{bmatrix} + L_x (y - \hat{y}) \quad (2.19)$$

$$\begin{bmatrix} \hat{f}_a \end{bmatrix} = L_f (y - \hat{y}) \quad (2.20)$$

$$\hat{y} = C_a \hat{x}_a \quad (2.21)$$

Apabila kedua persamaan (2.21) dan (2.22) dinyatakan dalam bentuk persamaan ruang keadaan sebagai berikut,

$$\begin{bmatrix} \dot{\hat{x}}_a \\ \dot{\hat{f}}_a \end{bmatrix} = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_a \\ \hat{f}_a \end{bmatrix} + \begin{bmatrix} B_a \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_f \end{bmatrix} \begin{bmatrix} C_a & 0 \end{bmatrix} (y - \hat{y}) \quad (2.22)$$

$$\hat{y} = \begin{bmatrix} C_a & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_a \\ \hat{f}_a \end{bmatrix} \quad (2.23)$$

Dengan $\hat{x}_a, \hat{f}_a, \hat{y}$ berturut turut adalah hasil estimasi *state*, kesalahan dan output. L_x adalah gain proporsional dari *observer* dan L_f adalah gain integral dari *observer*. Gain *observer* ini digunakan untuk menjamin kestabilan dinamik dari *error* estimasi.

2.5 Kesalahan Pada Sistem

Kesalahan dan kegagalan digunakan untuk menyatakan tingkat degradasi suatu sistem. Kesalahan yaitu sebagai bentuk penyimpangan yang tidak diinginkan dari kondisi standar yang biasa diterima. Kegagalan yaitu kerusakan secara permanen atas kemampuan sistem untuk melakukan fungsi yang diperlukan pada kondisi

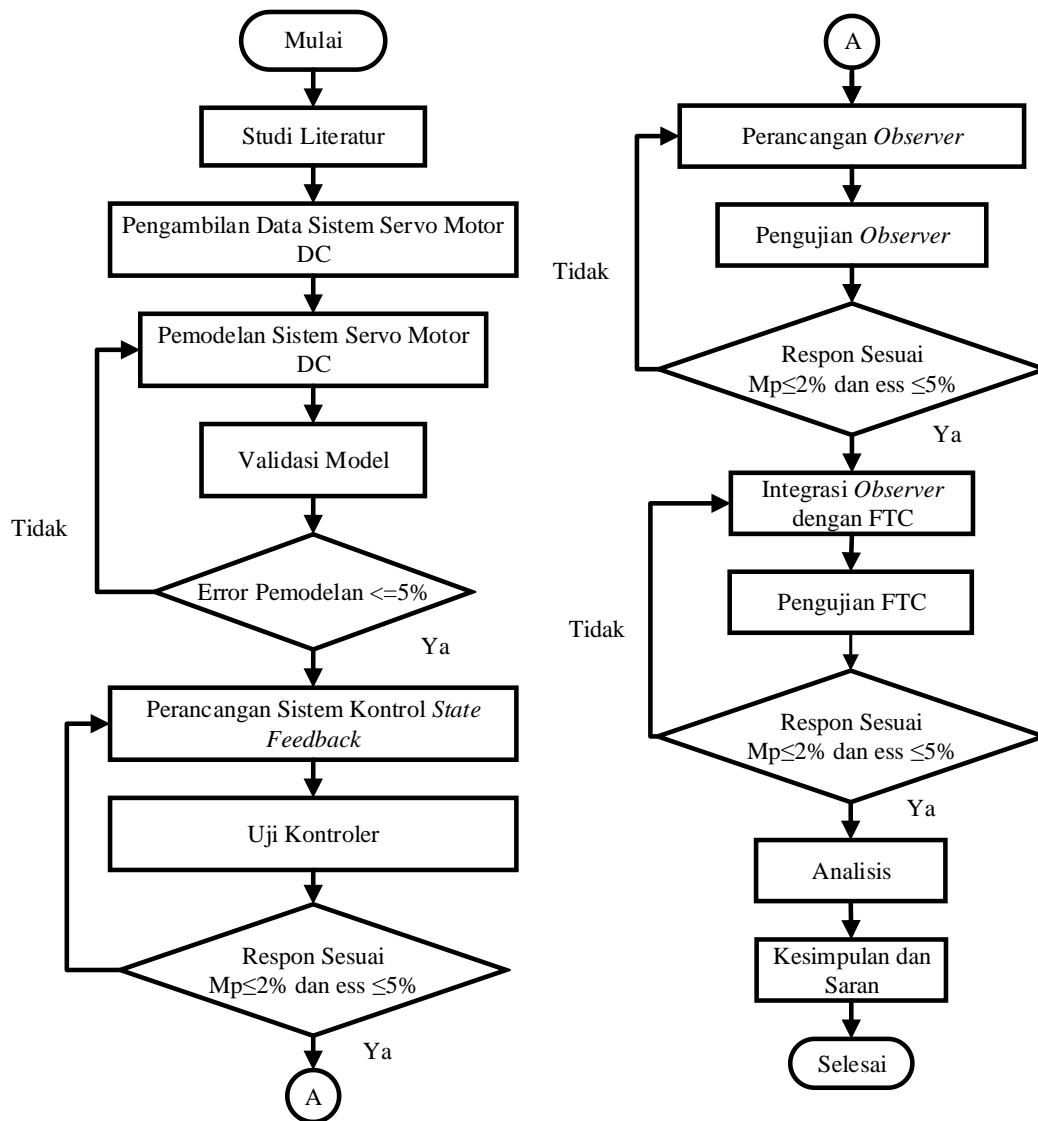
operasi tertentu. Sifat kerusakan yang ditimbulkan oleh kegagalan lebih berat daripada kesalahan.

Kesalahan dapat dibagi menjadi dua yaitu kesalahan minor dan kesalahan mayor. Kesalahan dibedakan berdasarkan tingkat keparahan. Tingkat keparahan yang dimaksud ditinjau dari dampak yang ditimbulkan pada performansi sistem. Kesalahan minor adalah kesalahan dengan tingkat keparahan yang kecil sehingga sistem masih bekerja. Kesalahan minor yang biasanya terjadi yaitu kesalahan pembacaan akibat perubahan parameter dan kalibrasi yang buruk. Sedangkan kesalahan mayor adalah kesalahan yang dapat mengakibatkan sistem tidak dapat mempertahankan performanya.

Halaman ini sengaja dikosongkan

BAB III METODOLOGI PENELITIAN

Bab ini berisi mengenai tahapan – tahapan yang dilakukan dalam penelitian. Prosedur penelitian terdiri dari beberapa langkah agar mampu mencapai tujuan dari penelitian. Secara garis besar prosedur penelitian digambarkan dalam diagram alir seperti dibawah ini,

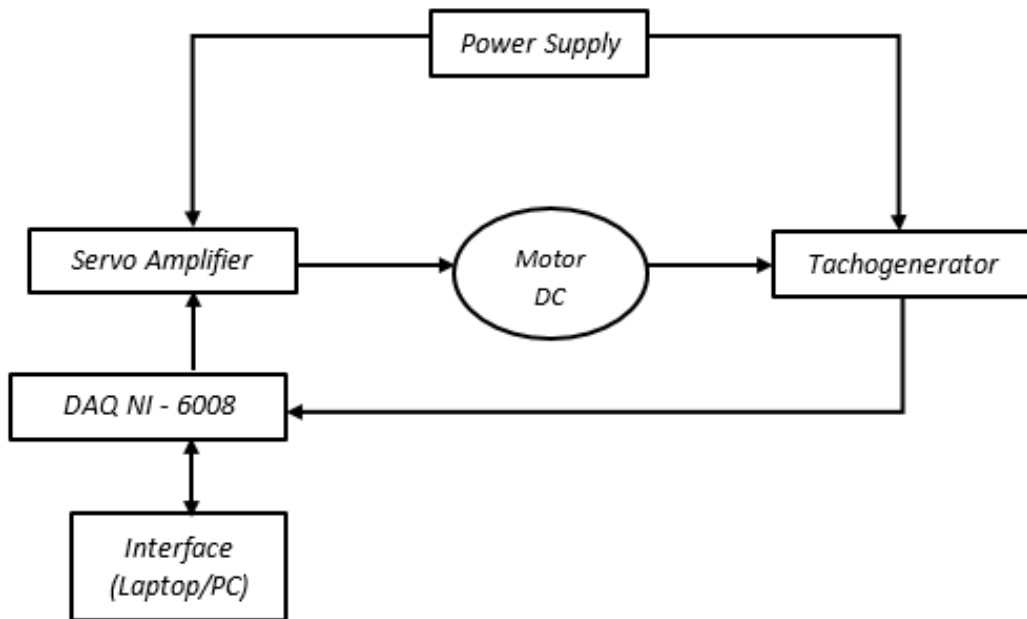


Gambar 3.1 Diagram Alir Metode Penelitian

Keterangan pada setiap proses akan dijelaskan pada beberapa tahapan seperti berikut,

3.1 Pengambilan Data Sistem Servo Motor DC

Pengambilan data sistem digunakan untuk memodelkan sistem motor DC. Data-data yang dibutuhkan untuk identifikasi adalah data kecepatan dalam bentuk tegangan yang dikonversi oleh *tachogenerator* dan data arus. Data kecepatan dan arus diperoleh dengan melakukan eksperimen secara langsung pada sistem motor DC di ruang laboratorium fisika rekayasa Teknik Fisika ITS. Sistem servo modular MS150 DC ditata sedemikian rupa seperti pada Gambar 3.2 agar data masukan dan keluaran dapat diambil untuk pemodelan sistem.



Gambar 3.2 Diagram blok set up sistem servo modular MS150 DC

Berdasarkan Gambar 3.2, sistem kendali berada pada komputer/laptop atau *user interface*. Sistem kendali berfungsi untuk mengatur sinyal masukan pada motor DC, karena pengendalian kecepatan motor DC dipengaruhi oleh besarnya tegangan masukan yang diberikan. Untuk dapat menjalin komunikasi antara komputer dengan plant maka diperlukan sebuah antarmuka yaitu DAQ NI-6008, selain itu DAQ NI-6008 juga diperlukan sebagai media pengkonversi sinyal *digital-to-analog*. DAQ NI-6008 memiliki karakteristik dapat menerima dan mengirimkan sinyal, sinyal akan dikirim menuju ke *servo amplifier*. Selanjutnya *servo amplifier* mengirimkan sinyal ke motor DC sehingga menyebabkan motor DC berputar. Kemudian motor DC berputar dengan rentang kecepatan rotasi sebesar 0-2000

RPM yang selanjutnya akan dikonversi menjadi sinyal tegangan dengan rentang 0–5 volt oleh *tachogenerator*.

3.2 Pemodelan Sistem Servo Motor DC

Identifikasi sistem dapat diperoleh dengan memodelkan sistem secara matematik. Untuk memperoleh model matematis dapat dilakukan melalui pendekatan analisis dan eksperimen (Wang, dkk., 2015). Pada pendekatan eksperimen sebuah model diperoleh dari data eksperimen yang telah dilakukan. Pemilihan model dilakukan dengan metode parametrik berdasarkan data masukan dan data keluaran dari plant yang telah diperoleh. Pemodelan dengan metode parametrik dipilih karena pada metode ini nilai parameter dari suatu sistem diperoleh secara langsung. Selain itu metode parametrik lebih banyak digunakan dalam aplikasi riil dan merupakan solusi yang tepat jika terkait dengan sistem riil (Rootsandy, 2011). Selanjutnya, dilakukan pengambilan data dengan set up sistem servo modular motor MS150 DC.

3.2.1 Model Kecepatan

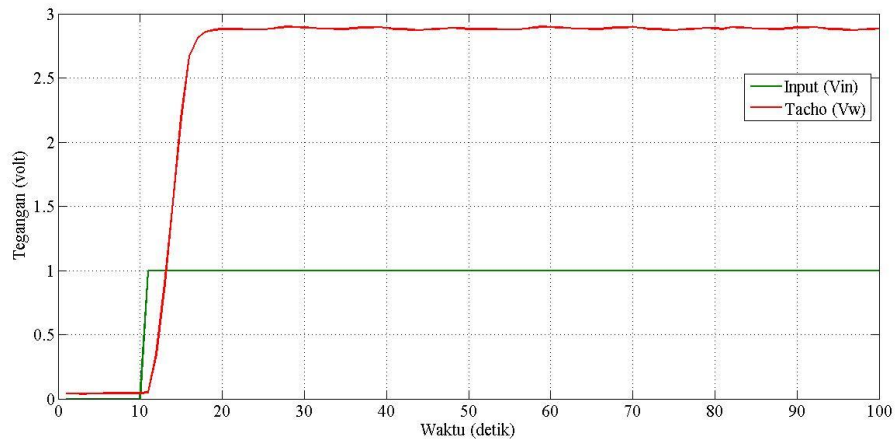
Kecepatan dari sistem servo dipengaruhi oleh satu input yaitu tegangan input dari *servo amplifier*. Model yang digunakan untuk mendekati dinamika kecepatan adalah fungsi transfer orde satu dengan parameter gain dan *time constant*. Seperti yang ditunjukkan pada persamaan (3.3) untuk fungsi transfer kecepatan.

Persamaan yang digunakan untuk mengetahui nilai kecepatan dengan menggunakan parameter gain dan tegangan input dapat dituliskan,

$$G_{\omega} = \frac{V_{\omega}}{V_{in}} \quad (3.1)$$

sehingga

$$V_{\omega} = V_{in} G_{\omega} \quad (3.2)$$



Gambar 3.3 Respon tegangan tachogenerator

Berdasarkan Gambar 3.3 didapatkan informasi nilai gain dan nilai *time constant* yang digunakan untuk memperoleh nilai fungsi transfer kecepatan dari sistem servo motor DC dengan memasukkan nilai tersebut pada persamaan (3.3),

$$G_{\omega} = \frac{K_{\omega}}{\tau_{\omega}s + 1} \quad (3.3)$$

Untuk mengetahui nilai konstanta kecepatan maka digunakan persamaan sebagai berikut,

$$K_{\omega} = \frac{\Delta V_{\omega}}{\Delta V_{in}} \quad (3.4)$$

Sedangkan untuk nilai *time constant* kecepatan didapatkan dari persamaan sebagai berikut,

$$\tau = \tau_1 - \tau_0 \quad (3.5)$$

3.2.2 Model Arus

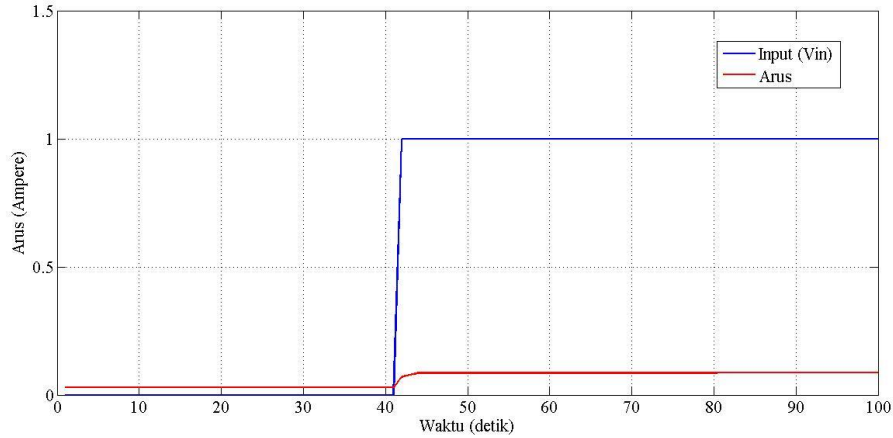
Aktuator yang digunakan pada sistem servo motor DC adalah servo amplifier. Data yang diambil merupakan data pembacaan arus dari servoamplifier.

Berdasarkan Gambar 3.4 didapatkan informasi nilai gain dan nilai *time constant* yang digunakan untuk memperoleh nilai fungsi transfer arus dari sistem servo motor DC dengan memasukkan nilai tersebut pada persamaan (3.6).

$$G_i = \frac{K_i}{\tau_i s + 1} \quad (3.6)$$

Sedangkan untuk nilai *time constant* arus didapatkan dari persamaan sebagai berikut,

$$\tau = \tau_1 - \tau_0 \quad (3.7)$$



Gambar 3.4 Respon arus

Persamaan (3.3) dan (3.6) merupakan persamaan fungsi alih dari sistem servo (dengan *output* kecepatan dan arus), dapat dituliskan dalam bentuk *state space* sebagai berikut

$$\begin{bmatrix} \dot{\omega} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_{\omega}} & 0 \\ 0 & -\frac{1}{\tau_i} \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} + \begin{bmatrix} \frac{K_{\omega}}{\tau_{\omega}} \\ \frac{K_i}{\tau_i} \end{bmatrix} U \quad (3.8)$$

Nilai K dan τ dapat diketahui berdasarkan respon tegangan dan arus pada servo motor DC. *Time Constant* (τ) adalah waktu yang diperlukan untuk *output* sistem mencapai 63% dari steady state. Berdasarkan pemodelan yang telah dilakukan maka diperoleh state space sistem sebagai berikut,

$$\begin{bmatrix} \dot{\omega} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} -\frac{1}{5} & 0 \\ 0 & -\frac{1}{0.01} \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} + \begin{bmatrix} \frac{2.8}{5} \\ \frac{0.086}{0.01} \end{bmatrix} V \quad (3.9)$$

3.3 Validasi

Validasi model dilakukan untuk memastikan model yang didapatkan sudah menggambarkan sistem dalam kondisi riil. Validasi dilakukan dengan cara

membandingkan respon antara model fungsi transfer dengan sistem riil. Untuk mendapatkan nilai error antara model dengan sistem riil menggunakan persamaan sebagai berikut:

$$MAPE = \left(\frac{100\%}{n} \right) \sum_{t=1}^n \frac{|X_t - F_t|}{X_t} \quad (3.10)$$

Dengan,

X_t = Data aktual (sistem riil)

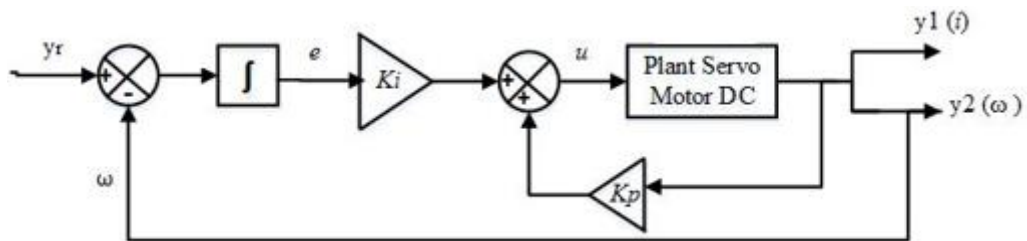
F_t = Data pemodelan

n = jumlah data

Persamaan MAPE (*Mean Absolute Percentage Error*) digunakan untuk memperoleh validasi nilai error seperti pada persamaan (3.10). MAPE adalah salah satu metode yang dapat digunakan untuk menghitung presentase penyimpangan antara data sistem riil dengan data model. Semakin kecil nilai error yang diperoleh maka semakin bagus hasil dari pemodelan. Selain itu jika error yang dihasilkan semakin kecil maka hasil model yang diperoleh sudah cukup merepresentasikan sistem dalam kondisi riil, sehingga model dapat digunakan.

3.4 Perancangan Sistem Kontrol *State Feedback*

Sistem pengendalian kecepatan pada motor DC menggunakan *state feedback*. Secara sistem *closed loop* dapat ditunjukkan dalam diagram blok pada Gambar 3.5 sebagai berikut.



Gambar 3.5 Diagram blok pengendalian kecepatan motor DC dengan *state feedback with integrator*

Pada tahap perancangan sistem kontrol *state feedback* menggunakan persamaan *state space* sebagai berikut,

$$\dot{x} = Ax + Bu \quad (3.11)$$

$$y = Cx + Du \quad (3.12)$$

Dengan $x = \begin{bmatrix} \omega \\ i \end{bmatrix}$ adalah kecepatan (ω) dan arus (i), sedangkan *output* yang diinginkan mengikuti *setpoint* adalah kecepatan sehingga,

$$y = C_r x = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} = \omega \quad (3.13)$$

Persamaan *error* sistem didapatkan sebagai berikut,

$$\dot{e}_c = y_{ref} - \omega \quad (3.14)$$

$$\dot{e}_c = \omega_{ref} - \omega \quad (3.15)$$

$$\dot{e}_c = \omega_{ref} - C_r \cdot x \quad (3.16)$$

$$\dot{e}_c = -\omega_{ref} + C_r \cdot x \quad (3.17)$$

Penggabungan persamaan (3.15) dan (3.17) dapat menghasilkan persamaan ruang keadaan baru sebagai berikut,

$$\begin{bmatrix} \dot{x} \\ \dot{e}_c \end{bmatrix} = \begin{bmatrix} A & 0 \\ C_r & 0 \end{bmatrix} \begin{bmatrix} x \\ e_c \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \omega_{ref} \quad (3.18)$$

$$\dot{x}_c = A_c x_c + B_c u + D_c \omega_{ref} \quad (3.19)$$

Dengan, $u = K_c x_c$ maka dari persamaan (3.19) dapat dituliskan sebagai berikut,

$$\dot{x}_c = A_c x_c + B_c (K_c x_c) + D_c \omega_{ref} \quad (3.20)$$

$$K_c = \begin{bmatrix} K_p & K_i \end{bmatrix} \quad (3.21)$$

Dimana, $K_c = \text{gain controller}$.

Penentuan gain kontroler dilakukan dengan menggunakan metode *pole placement*.

Untuk simulasi, nilai K_c diperoleh dengan memberikan nilai *pole* sebagai berikut,

$$\text{Pole ke-1} = -15 + 15i$$

$$\text{Pole ke-2} = -15 - 5i$$

$$\text{Pole ke-3} = -20$$

Maka diperoleh nilai *gain* pengendali *state feedback* sebagai berikut,

$$K_c = [14.136 \quad -6.757 \quad 89.285]$$

Sehingga nilai *gain* proportional dan integral adalah

$$K_p = [14.136 \quad -6.757]$$

$$K_i = [89.285]$$

3.5 Perancangan *Observer*

Observer merupakan metode yang digunakan untuk mendesain *speed sensorless*. *Observer* digunakan untuk mengestimasi kecepatan dan *disturbance* dengan cara mengembangkan persamaan ruang keadaan sistem. Persamaan dituliskan sebagai berikut,

$$\dot{x}(t) = Ax(t) + Bu(t) + F_a f_a(t) \quad (3.22)$$

$$y(t) = Cx(t) \quad (3.23)$$

Penambahan state diperlukan untuk memindahkan *output* ke persamaan keadaan baru sebagai berikut,

$$\dot{z}(t) = Az(y - z) \quad (3.24)$$

$$y(t) = C_s x(t) \quad (3.25)$$

$$\dot{z}(t) = Az(C_s x(t) - z) \quad (3.26)$$

Untuk Az merupakan matrik identitas (I) sedangkan C_s merupakan matriks yang menghasilkan informasi dari arus.

Persamaan (3.22) dan persamaan (3.26) dapat diubah menjadi persamaan *state space* sebagai berikut,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ AzC_s & -Az \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} F_a \\ 0 \end{bmatrix} f_a \quad (3.27)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \quad (3.28)$$

dengan

$$x_a = \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} A_a = \begin{bmatrix} A & 0 \\ AzC_s & -Az \end{bmatrix} B_a = \begin{bmatrix} B \\ 0 \end{bmatrix} E_a = \begin{bmatrix} F_a \\ 0 \end{bmatrix} f C_a = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Untuk memperoleh *gain observer* maka diperlukan persamaan ruang keadaan baru yang ditunjukkan pada persamaan (3.29)

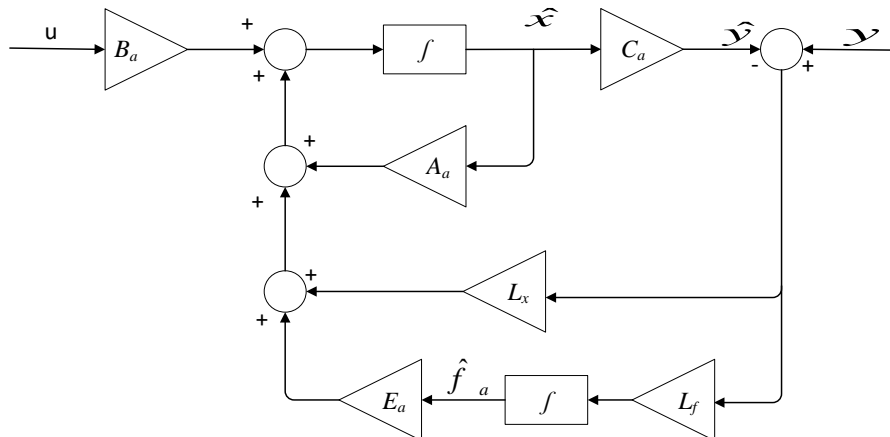
$$\dot{\hat{x}}_a = A_a \hat{x} + B_a u + E_a [\hat{f}_a] + L(y - \hat{y}) \quad (3.29)$$

$$\hat{y} = C_a \hat{x} \quad (3.30)$$

Dari persamaan (3.29) L merupakan hasil *gain observer*. *Gain observer* ini digunakan untuk menjamin kestabilan dinamik dari error estimasi. Sehingga diperoleh nilai *Gain observer* sebagai berikut,

$$L = \begin{bmatrix} -0.353 \\ 0.673 \\ 4.610 \\ 10 \end{bmatrix}$$

Pada persamaan (3.29) dan (3.30) jika digambarkan dalam bentuk struktur algoritma *observer* ditunjukkan seperti Gambar 3.6.

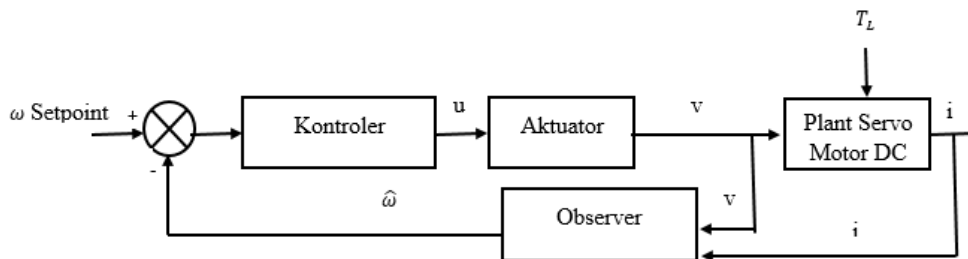


Gambar 3.6 Struktur extended state observer

3.6 Integrasi *Observer* dengan FTC

Untuk mengkompensasi kesalahan yang terjadi pada plant servo motor DC digunakan sistem FTC. FTC merupakan metode kompensasi yang berfungsi untuk menghapus sinyal kesalahan yang terjadi sehingga dapat menjamin *plant* yang mengalami kesalahan tetap memiliki performansi yang stabil dalam menghadapi perubahan input referensi dan gangguan.

Gambar 3.6 menjelaskan diagram blok dari perancangan FTC. Perancangan FTC ini menggunakan *observer*. Pada awalnya *observer* dirancang untuk mengestimasi kecepatan, setelah itu baru diperluas dengan memberikan penambahan kesalahan berupa adanya torsi beban.



Gambar 3.7 Diagram blok Fault Tolerant Control pada Servo Motor DC

3.7 Uji Performansi FTC

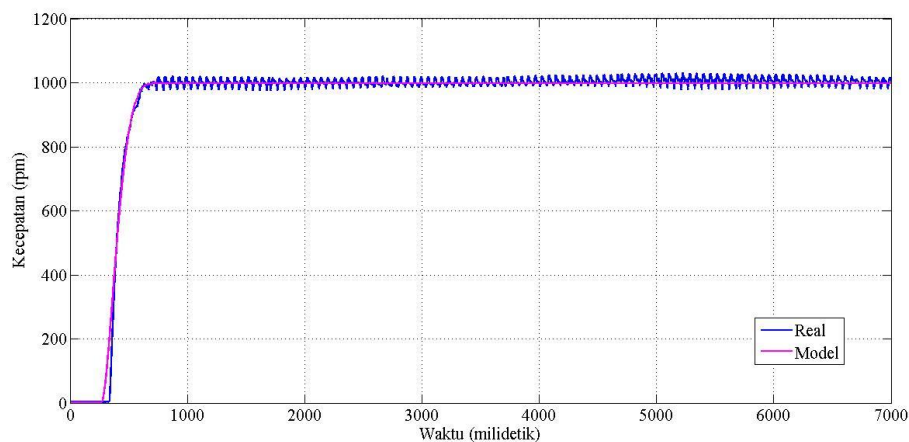
Pengujian performansi FTC dilakukan dengan memberikan pengaruh torsi beban sebagai *disturbance* pada sistem motor DC sebagai bentuk pentimpangan yang tidak diinginkan dari kondisi standar yang biasa diterima. *Fault Tolerant Control* dikatakan memiliki respon baik jika nilai *maximum overshoot* $\leq 2\%$ dan *error steady state* $\leq 5\%$. Uji performansi yang dilakukan pada sistem adalah uji *controller*, uji *observer*, dan uji kesalahan.

BAB IV HASIL DAN PEMBAHASAN

4.1 Validasi Data

Tahap perancangan yang telah dilakukan berupa pengambilan data pada sistem motor DC, selanjutnya dilakukan tahap identifikasi dengan menggunakan metode parametrik diperoleh persamaan fungsi alih dari sistem motor DC. Validasi dilakukan dengan membandingkan grafik keluaran model sistem dengan keluaran sistem secara langsung.

4.1.1 Validasi Kecepatan



Gambar 4.1 Respon hasil validasi model kecepatan

Gambar 4.1 menunjukkan dua grafik yaitu grafik berwarna biru yang menunjukkan keluaran kecepatan dari motor dc secara langsung dan grafik berwarna merah muda yang menunjukkan kecepatan dari model motor DC yang diperoleh. Dari hasil validasi kecepatan sistem motor DC diperoleh nilai *error* validasi sebesar 2,14%. Pada Gambar 4.1 diketahui terdapat *error* yang cukup besar dikarenakan pada kondisi *real* sistem motor DC dipengaruhi oleh *noise* yang cukup besar, namun *error* yang ditimbulkan masih dapat ditoleransi yaitu kurang dari 5% sehingga grafik respon kecepatan antara model yang diperoleh sudah mampu merepresentasikan kecepatan sistem secara *realtime*.

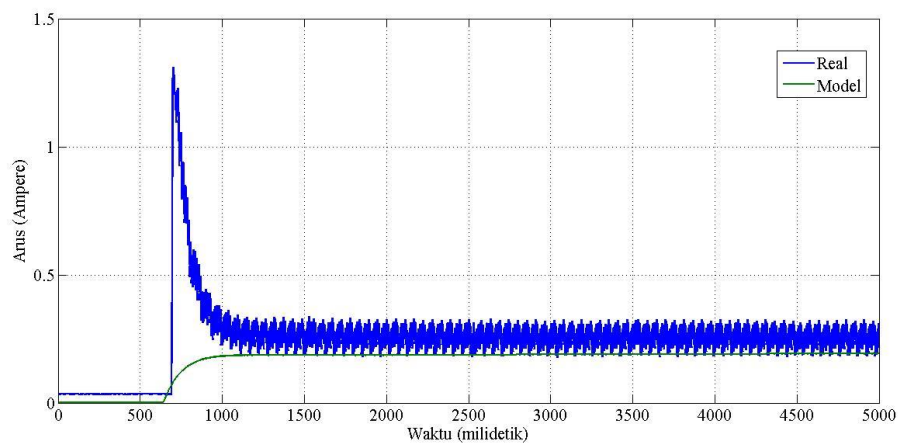
Pada Tabel 4.1 merupakan hasil perhitungan dari validasi kecepatan dengan menggunakan persamaan MAPE (*Mean Absolute Percentage Error*). Berdasarkan

persamaan (3.10) maka didapatkan hasil perhitungan MAPE ketika respon mencapai *steady* yaitu sebesar 0,021 atau 2,1%.

Tabel 4.1 Perhitungan Validasi Model Kecepatan dengan Menggunakan MAPE

Real (X_t)	Model (F_t)	$(X_t - F_t) / X_t$	MAPE
1020,175	998,275	0,021	2,1%
1020,174	998,275	0,021	2,1%
1020,173	998,275	0,021	2,1%
1020,163	998,275	0,021	2,1%
1020,156	998,275	0,021	2,1%
1020,148	998,275	0,021	2,1%
1020,137	998,275	0,021	2,1%
1020,126	998,275	0,021	2,1%
1020,12	998,275	0,021	2,1%
1020,109	998,275	0,021	2,1%

4.1.1 Validasi Arus



Gambar 4.2 Respon hasil validasi model arus

Gambar 4.2 menunjukkan dua grafik yaitu grafik berwarna biru yang menunjukkan keluaran arus dari motor dc secara langsung dan grafik berwarna hijau yang menunjukkan kecepatan dari model motor DC yang diperoleh. Dari hasil validasi kecepatan sistem motor DC diperoleh nilai *error* validasi sebesar 4,86%. Pada Gambar 4.2 diketahui terdapat *error* yang cukup besar dikarenakan pada kondisi *real* sistem motor DC dipengaruhi oleh *noise* yang cukup besar. Walaupun nilai *error* validasi arus cukup besar, namun model arus motor DC sudah cukup

merepresentasikan arus motor DC secara langsung karena memiliki nilai yang tidak jauh berbeda dari kondisi arus secara real.

Pada Tabel 4.2 merupakan hasil perhitungan dari validasi arus dengan menggunakan persamaan MAPE (*Mean Absolute Percentage Error*). Berdasarkan persamaan (3.10) maka didapatkan hasil perhitungan MAPE untuk model arus ketika respon mencapai *steady* yaitu sebesar 0,048 atau 4,8%.

Tabel 4.2 Perhitungan Validasi Model Arus dengan Menggunakan MAPE

Real (Xt)	Model (Ft)	(Xt-Ft)/Xt	MAPE
0,3052	0,2912	0,045	4,5%
0,3053	0,2912	0,046	4,6%
0,3054	0,2912	0,046	4,6%
0,3055	0,2912	0,046	4,6%
0,3056	0,2912	0,047	4,7%
0,3057	0,2912	0,047	4,7%
0,3058	0,2912	0,047	4,7%
0,3059	0,2912	0,048	4,8%
0,306	0,2912	0,048	4,8%
0,3061	0,2912	0,048	4,8%

4.2 Uji Traking Kontroler *State Feedback*

Perancangan sistem kontrol *state feedback* digunakan untuk memperoleh nilai gain kontrol yang akan digunakan pada sistem. Performansi kontroler *state feedback* dapat diketahui dengan dilakukan pengujian uji *tracking*. Dengan menggunakan *pole placement* maka diperoleh nilai *gain* kontroler sebagai berikut,

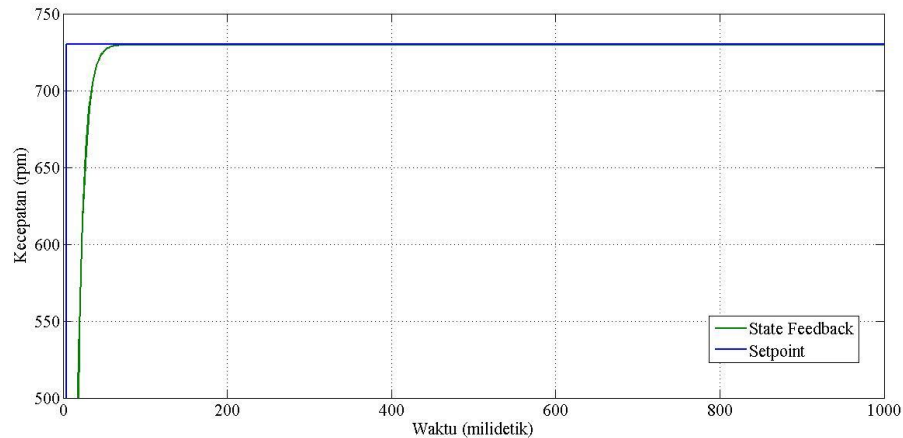
$$Kp = [14.1361 \quad -6.7577]$$

$$Ki = [89.2857]$$

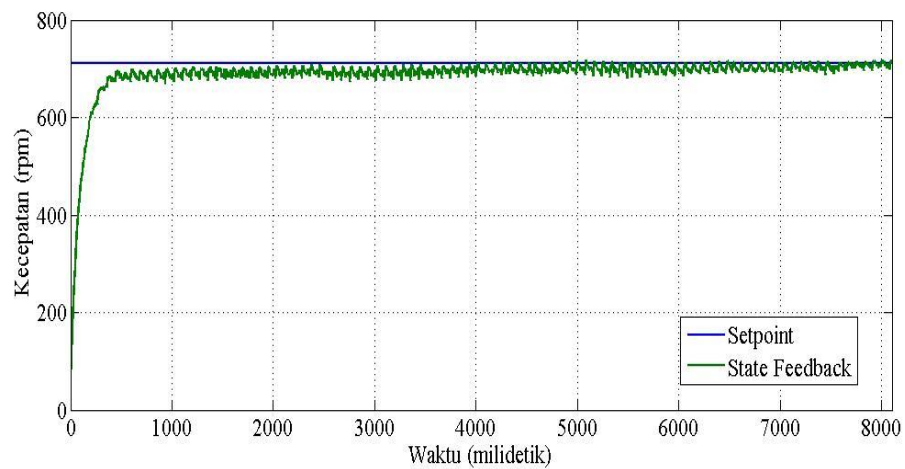
Grafik respon sistem kontrol *state feedback* dengan *integrator* pada motor DC secara simulasi ditunjukkan pada Gambar 4.3 menunjukkan respon sistem perancangan sistem kontrol *state feedback* dengan integrator. Grafik yang berwarna biru menunjukkan nilai *set point* dan grafik berwarna hijau menunjukkan hasil respon kontroler.

Berdasarkan grafik tersebut terlihat bahwa respon dari kontroler dapat *tracking* sesuai dengan nilai *set point*. Perubahan nilai *set point* yang terjadi sebesar 0 rpm

sampai 730 rpm. Hal tersebut dapat membuktikan bahwa kontroler yang dirancang dapat bekerja dengan baik karena respon mampu mengikuti nilai set point. Karakteristik respon dari uji *tracking* menunjukkan *maximum overshoot* dan *error steady state* sebesar 0 % sedangkan nilai *settling time* sebesar 115 milidetik.



Gambar 4.3 Uji *tracking* kontrol *state feedback* dengan integrator secara simulasi



Gambar 4.4 Uji *tracking* kontrol *state feedback* dengan integrator secara *real*

Gambar 4.4 menunjukkan grafik respon sistem kontrol *state feedback* dengan *integrator* secara *real* untuk pengendalian kecepatan sistem sistem servo motor DC. Gambar 4.4 menunjukkan 2 grafik yaitu grafik berwarna biru yang menunjukkan nilai *set point* dan grafik berwarna hijau yang menunjukkan sistem dengan *state feedback*, nilai *set point* yang diberikan sebesar 2 volt pada waktu ke 0 milidetik, dari kedua grafik tersebut dapat diketahui bahwa sistem kontrol *state feedback* dapat berkerja dengan baik karena mampu mencapai nilai *set point* dengan

karakteristik respon *settling time* sebesar 1081 milidetik, dan *error steady-state* sebesar 5%.

Tabel 4.3 Perbandingan Uji Tracking Kontroler

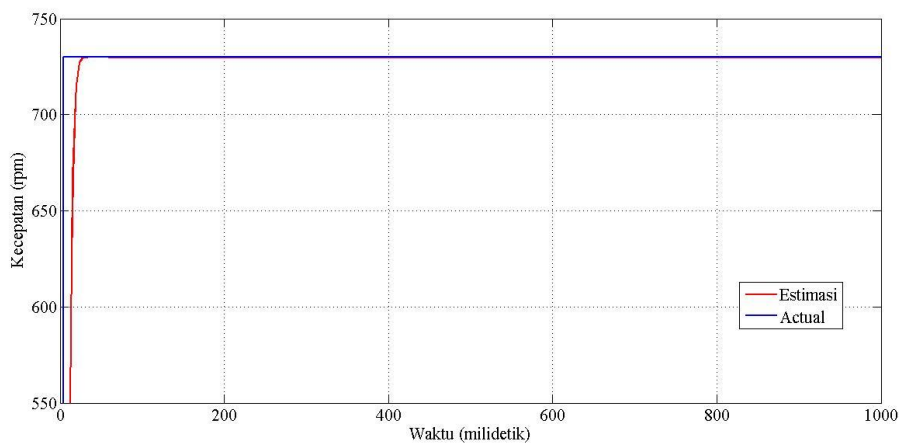
Parameter	Simulasi	Real
<i>Error steady state</i>	0%	5%
<i>Settling time</i>	115 milidetik	1081 milidetik

4.3 Uji Estimasi *Observer*

Pada tahapan uji hasil *observer* bertujuan untuk membuktikan bahwa *observer* yang dirancang mampu bekerja sebagai *softsensor* (pengganti sensor kecepatan) dengan menggunakan informasi arus.

4.3.1 Uji *Observer* Secara Simulasi

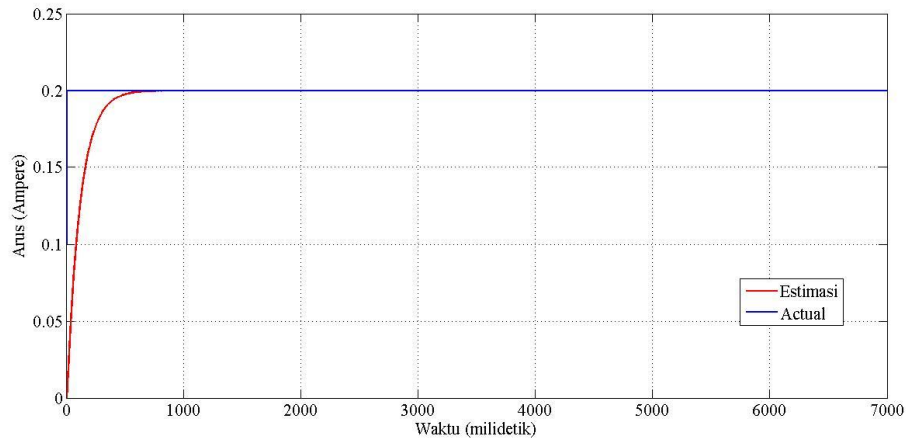
Gambar 4.5 menunjukkan hasil estimasi kecepatan secara simulasi, terdapat 2 grafik yaitu grafik berwarna biru yang menunjukkan aktual dan grafik berwarna merah yang menunjukkan estimasi. Dari kedua grafik tersebut dapat diketahui bahwa *observer* yang dirancang memiliki hasil yang sama dengan nilai aktual, sehingga *observer* sudah mampu bekerja sebagai *softsensor*. Dengan karakteristik respon *settling time* sebesar 105 milidetik dan tidak terjadi *error steady-state* maupun *maximum overshoot*.



Gambar 4.5 Hasil estimasi kecepatan secara simulasi

Gambar 4.6 menunjukkan hasil estimasi arus secara simulasi, terdapat 2 grafik yaitu grafik berwarna biru yang menunjukkan aktual dan grafik berwarna merah yang menunjukkan estimasi. Dari kedua grafik tersebut dapat diketahui

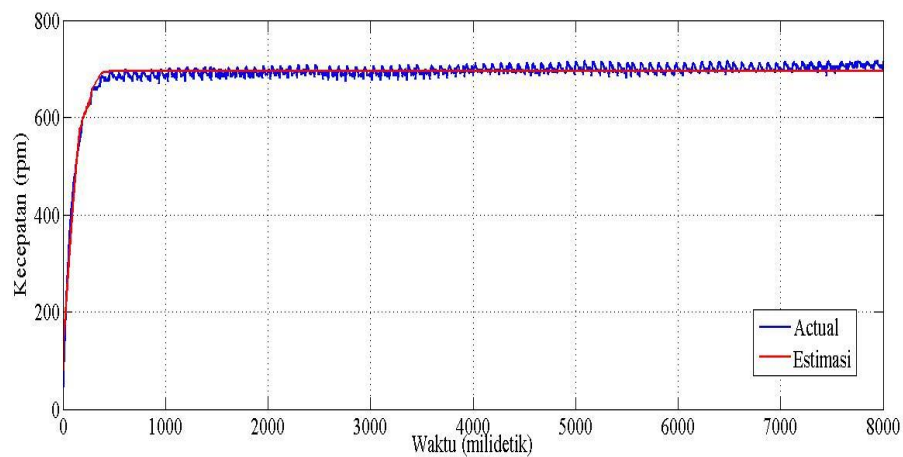
bahwa *observer* yang dirancang memiliki hasil yang sama dengan nilai aktual, sehingga hasil estimasi arus juga bekerja dengan baik. Dengan karakteristik respon *settling time* sebesar 920 milidetik dan tidak terjadi *error steady-state* maupun *maximum overshoot*.



Gambar 4.6 Hasil estimasi arus secara simulasi

4.3.2 Uji *Observer* Secara *Real*

Setelah dilakukan perancangan secara simulasi kemudian diterapkan secara *real* pada sistem servo motor DC.



Gambar 4.7 Hasil estimasi kecepatan secara *real*

Gambar 4.7 menunjukkan hasil estimasi kecepatan secara *real*, terdapat 2 grafik yaitu grafik berwarna biru yang menunjukkan aktual dan grafik berwarna merah yang menunjukkan estimasi. Dari kedua grafik tersebut dapat diketahui bahwa *observer* yang diterapkan pada kondisi *real* memiliki hasil yang sama dengan nilai *actual*, sehingga *observer* sudah mampu bekerja sebagai *softsensor*.

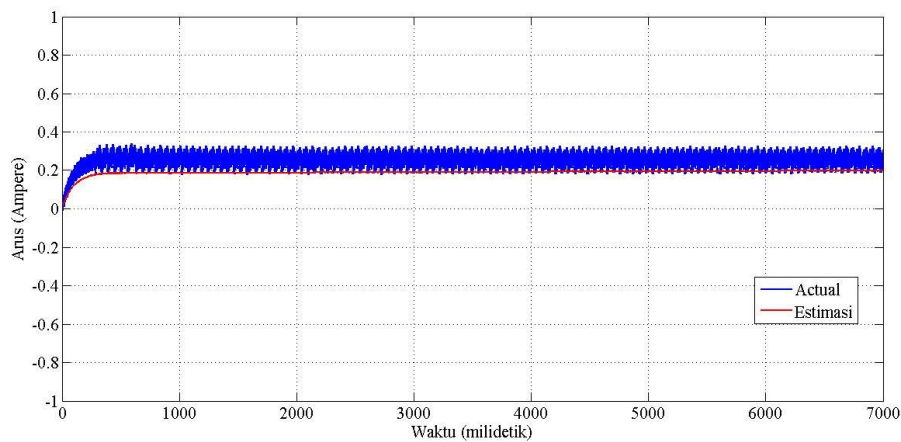
Dengan karakteristik respon *settling time* sebesar 398 milidetik dan *error steady-state* sebesar 0,15%.

Pada Tabel 4.4 menunjukkan perbandingan uji estimasi kecepatan secara simulasi dan *real*. Dari perbandingan menunjukkan perbedaan nilai *error steady state*, hal ini dikarenakan sistem *real* motor DC memiliki *noise* yang cukup besar.

Tabel 4.4 Perbandingan Hasil Estimasi Kecepatan

Parameter	Simulasi	<i>Real</i>
<i>Error steady state</i>	0%	0,15%
<i>Settling time</i>	105 milidetik	398 milidetik

Gambar 4.8 menunjukkan hasil estimasi arus secara *real*, terdapat 2 grafik yaitu grafik berwarna biru yang menunjukkan aktual dan grafik berwarna merah yang menunjukkan estimasi. Dari kedua grafik tersebut dapat diketahui bahwa *observer* yang dirancang memiliki hasil yang sama dengan nilai aktual, sehingga hasil estimasi arus juga bekerja dengan baik. Dengan karakteristik respon *settling time* sebesar 1028 milidetik dan *error steady-state* sebesar 3,13%.



Gambar 4.8 Hasil estimasi arus secara *real*

Pada Tabel 4.5 menunjukkan perbandingan uji estimasi arus secara simulasi dan *real*. Dari perbandingan menunjukkan perbedaan nilai *error steady state* yang cukup besar, hal ini dikarenakan sistem *real* motor DC memiliki *noise* yang cukup besar.

Tabel 4.5 Perbandingan Hasil Estimasi Arus

Parameter	Simulasi	Real
<i>Error steady state</i>	0%	3,13%
<i>Settling time</i>	920 milidetik	1028 milidetik

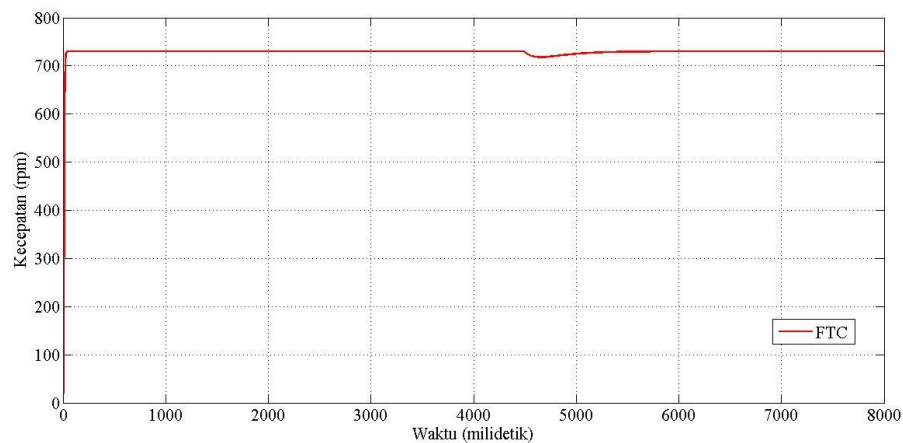
4.4 Uji Performansi

Uji performansi dilakukan untuk menguji algoritma *fault tolerant control* yang telah dirancang dapat memperbaiki kesalahan atau bentuk penyimpangan dari kondisi standar yang terjadi pada sistem servo motor DC.

4.4.1 Uji Performansi FTC Secara Simulasi

- **Uji FTC dengan torsi 1 Nm**

Uji secara simulasi dengan memberikan adanya penambahan beban pada sistem sebesar 1 Nm.



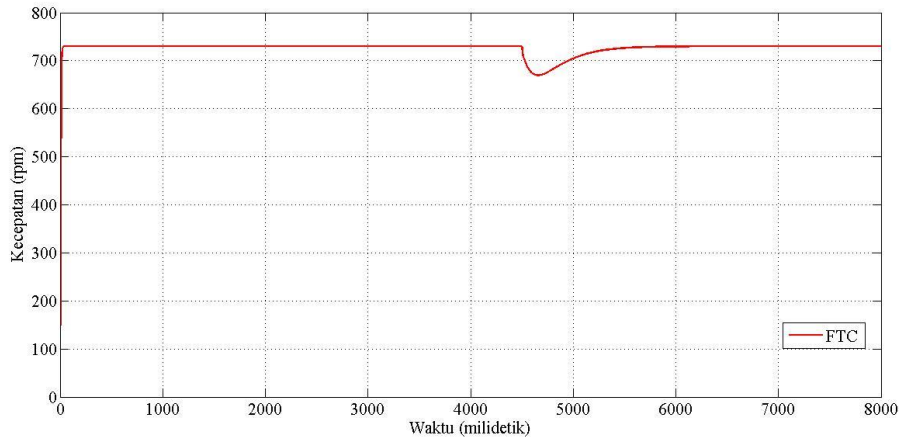
Gambar 4.9 Respon output dengan penambahan torsi beban 1 Nm secara simulasi

Gambar 4.9 menunjukkan respon kecepatan motor DC dengan menggunakan FTC yang ditunjukkan oleh grafik berwarna merah. Ketika diberikan pengaruh torsi beban pada waktu ke 4500 milidetik, respon kecepatan motor DC mengalami *maximum undershoot* yang menunjukkan terjadinya kesalahan atau penyimpangan. Namun sistem FTC dapat memperbaiki kesalahan sehingga dapat mencapai ke nilai *setpoint* atau kecepatan yang ditentukan sebesar 730 rpm pada detik ke 5122 detik. Karakteristik respon FTC ditunjukkan pada Tabel 4.6.

Tabel 4.6 Karakteristik Respon *Output* dengan Torsi Beban 1 Nm Secara Simulasi

Parameter	FTC
<i>Maximum Undershoot</i>	1,65%
<i>Settling time</i>	1225 milidetik

- Uji FTC dengan Torsi 10 Nm



Gambar 4.10 Respon output dengan penambahan torsi beban 10 Nm secara simulasi

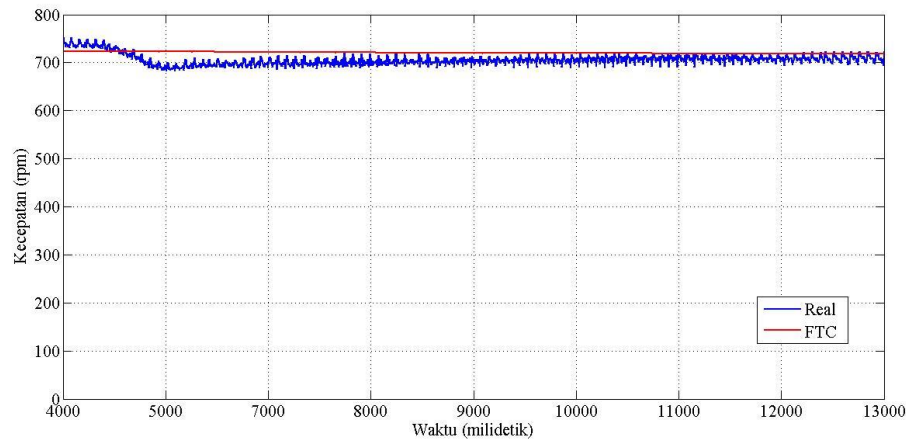
Uji FTC dengan memberikan penambahan beban pada sistem motor DC sebesar 10 Nm. Penambahan beban sebesar 10 Nm merupakan beban maksimal yang diberikan pada sistem motor DC. Gambar 4.10 menunjukkan respon kecepatan motor DC dengan menggunakan FTC yang ditunjukkan oleh grafik berwarna merah. Ketika diberikan pengaruh torsi beban pada waktu ke 4500 milidetik, respon kecepatan motor DC mengalami *maximum undershoot* yang menunjukkan terjadinya kesalahan atau penyimpangan. Namun sistem FTC dapat memperbaiki kesalahan sehingga dapat mencapai ke nilai *setpoint* atau kecepatan yang ditentukan sebesar 730 rpm pada detik ke 5835 detik. Karakteristik respon FTC ditunjukkan pada Tabel 4.7.

Tabel 4.7 Karakteristik Respon *Output* dengan Torsi Beban 10 Nm Secara Simulasi

Parameter	FTC
<i>Maximum Undershoot</i>	8,26%
<i>Settling time</i>	1335 milidetik

4.4.2 Uji Performansi FTC Secara *Real*

- Uji FTC dengan torsi 1 Nm



Gambar 4.11 Respon output dengan penambahan torsi beban 1 Nm secara *real*

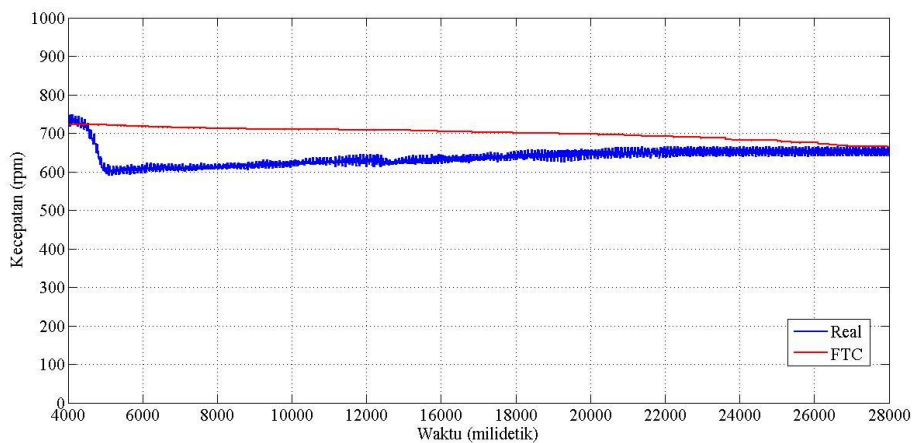
Uji secara *real* dengan memberikan adanya pengaruh beban pada motor DC sebesar 1 Nm. Gambar 4.11 menunjukkan respon kecepatan secara *real* pada motor DC ketika terjadi pengaruh torsi beban sebesar 1 Nm pada waktu ke 4500 milidetik. Kecepatan motor DC secara *real* ditunjukkan oleh warna biru, sistem real awalnya memiliki kecepatan sebesar 730 rpm ketika diberikan pengaruh torsi beban menurun menjadi 684,7 rpm. Grafik FTC ditunjukkan dengan warna merah. Hasil FTC tidak langsung memiliki nilai yang sama dengan nilai *real*, tetapi menurun secara perlahan dan membutuhkan waktu untuk menyesuaikan dengan respon motor DC secara *real*. Karakteristik respon ditunjukkan pada Tabel 4.8.

Tabel 4.8 Karakteristik Respon *Output* dengan Torsi Beban 1 Nm Secara *Real*

Parameter	FTC
<i>Settling time</i>	6477 milidetik
<i>Error steady state</i>	0,58%

- Uji FTC dengan Torsi 10 Nm

Uji FTC secara *real* dengan memberikan penambahan beban pada sistem motor DC sebesar 10 Nm. Penambahan beban sebesar 10 Nm merupakan beban maksimal yang diberikan pada sistem motor DC.



Gambar 4.12 Respon output dengan penambahan torsi beban 10 Nm secara *real*

Gambar 4.12 menunjukkan respon kecepatan secara *real* pada motor DC ketika terjadi pengaruh torsi beban sebesar 10 Nm pada waktu ke 5000 milidetik. Kecepatan motor DC secara *real* ditunjukkan oleh warna biru, sistem real awalnya memiliki kecepatan sebesar 730 rpm ketika diberikan pengaruh torsi beban menurun menjadi 599,7 rpm. Grafik FTC ditunjukkan dengan warna merah. Hasil FTC tidak langsung memiliki nilai yang sama dengan nilai *real*, tetapi menurun secara perlahan. Waktu yang diperlukan untuk hasil estimasi FTC sesuai dengan *real* cukup lama yaitu sebesar 14576 milidetik, hal ini dikarenakan besarnya torsi beban yang diberikan pada sistem motor DC. Karakteristik respon ditunjukkan pada Tabel 4.9.

Tabel 4.9 Karakteristik Respon Output dengan Torsi Beban 10 Nm Secara *Real*

Parameter	FTC
<i>Settling time</i>	14576 milidetik
<i>Error steady state</i>	0,43%

Hasil dari FTC yang diterapkan pada sistem motor DC secara simulasi maupun *real* menunjukkan bahwa FTC yang dirancang dapat bekerja dengan baik, hal ini dibuktikan sesuai Gambar 4.11 dan Gambar 4.12 hasil FTC dapat mengikuti hasil *real* dari sistem motor DC ketika adanya penambahan beban. Penambahan beban yang diberikan sebesar 1 Nm dan 10 Nm sistem FTC mampu bekerja, namun waktu yang diperlukan untuk sesuai dengan kondisi *real* cukup lama jika diberikan penambahan beban yang maksimal. Selain itu, munculnya *error steady state*

dikarenakan adanya *noise* pada sistem sehingga dapat mempengaruhi hasil *output* kecepatan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan, didapatkan kesimpulan dari penelitian mengenai perancangan *Fault Tolerant Control* pada pengendalian *speed sensorless* motor DC sebagai berikut:

- Sistem pengendalian *speed sensorless* motor DC dirancang dengan menggunakan kontrol *state feedback* dengan *integrator* yang mampu *tracking* dengan *error steady state* sebesar 5% dengan nilai K_p sebesar 14,136 dan -6,757 sedangkan K_i sebesar 89,285.
- Perancangan *sensorless* menggunakan metode *observer* dengan informasi dari arus. Hasil estimasi *observer* yang dirancang memiliki hasil yang sama dengan hasil aktual. Hasil simulasi menunjukkan tidak terjadinya *error steady state* dan *settling time* sebesar 105 milidetik, sedangkan hasil *real* menunjukkan *error steady state* sebesar 0,15% dan *settling time* sebesar 398 milidetik.
- Perancangan FTC secara simulasi maupun *real plant* dapat menjaga performansi ketika terjadi gangguan pada motor DC, berupa adanya penambahan beban sebesar 1 Nm dan 10 Nm. Dengan penambahan beban yang maksimal sebesar 10 Nm sistem motor DC tetap dapat bekerja dengan baik. Hal ini ditunjukkan dengan sistem FTC dapat sesuai dengan kecepatan yang ditentukan pada 14576 milidetik, dan *error steady state* sebesar 0,43%.

5.2 Saran

Dari penelitian tesis ini terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya adalah ditambahkan sensor arus yang memiliki *filter noise* sehingga dapat menghasilkan data pengukuran yang lebih baik.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Akram Ahmad. (2014). Speed control of a DC motor using Controllers. Automation, Control and Intelligent Systems,.
- Ali Akbar. (2013). Speed Sensorless and Sensor Fault Tolerant Optimal PI Regulator for Networked DC Motor System with Unknown Time-Delay and Packet Dropout. *IEEE*.
- Aydogmus O dan Talu F. (2012). Comparison of Extended Kalman and Particle Filter Based Sensorless Speed Control. *IEEE Transactions on Instrumentation and Measurement*, 61.
- C. Nie. (2012). Observer-Based Robust Fault Estimation For Fault-Tolerant Control.
- Ernesto, dkk. (2012). A New Method for Sensorless Estimation of the Speed and Position in Brushed DC Motors Using Support Vector Machines. *IEEE Transactions On Industrial Electronics*, 59.
- Hilairret. (2007). Speed Sensorless Control of a DC-motor via adaptive filters. *IET Electro Power Appl.*
- Hudaya, C. (2013). Motor DC. *Jakarta: Universitas Indonesia*.
- Hughes dan Drury. (2013). Electric Motors and Drives. Electric Motors and Drives.
- Hussein dan Ghasm. (2017). Implementation of Proportional Integral Observer Techniques for Load Frequency Control of Power System. *Science Direct*.
- Indriawati, K. e. (2015). Robust Observer-Based Fault Tolerant Tracking Control for Linear Systems with Simultaneous Actuator and Sensor Faults: Application to a DC Motor System. *Praise Worthy Praise*, 8.
- J. Jiang dan X. Yu. (2012). Fault-tolerant control systems: A comparative study between active and passive approaches. *Annu. Rev. Control*, 36, 60-72.

- K. Wang and shen. (2009). Study on rotor structure with different magnet assembly in high-speed sensorless brushless DC motor. *IET Electric Power Applications*.
- Koksal, M. (2007). Position Control of a Permanent Magnet DC motor by Model Reference Adaptive Control. *IEEE*, 112-117.
- Kresna, V. (2017). Perancangan Sistem Active Fault Tolerant Control Pada Pengendalian Kecepatan Sistem Servo Modular MS150 DC Dengan Kesalahan Pada Aktuator dan Sensor. *Surabaya: Institut Teknologi Sepuluh Nopember*.
- M. Dybkowski dan K.Klimkowski. (2017). Fault Tolerant Control Structure for Induction Motor Drive System. *Autom-J.Control. Meas.Electron.Comput.Commun*, 57, 638-647.
- M. Hilairet and F. Auger. (2007). Speed sensorless control of a DC-motor via adaptive filters. *IET. Electr. Power Appl*, 601-610.
- Ogata, K. (1997). *Modern Control Engineering 3rd ed.* London: Prentice Hall International.
- Oku. (2016). World ' s largest Science , Technology & Medicine Open Access book publisher c. Agricultural and Biological Sciences Grain Legumes.
- Praesomboon. (2009). Sensorless Speed Control of DC Servo Motor Using. *IEEE*.
- Radclife. (2015). Sensorless speed measurement for brushed DC motors. *IET Power Electronic*.
- Radclife dan Kumar. (2015). Sensorless speed measurement for brushed DC motors. *IET Power Electronics*, 8, 2223-2228.
- Rootsandy. (2011). Pengidentifikasian parameter fungsi alih sistem pada plant simulasi orde tiga dan empat dengan mBelakang, Lataretode algoritma genetik,. 1-11.
- Sebe N. (2012). Fault-Tolerant Servo System against Actuator Failures.

Wang, dkk. (2015). Rapid design of DC motor speed control system based on MATLAB. 168-171.

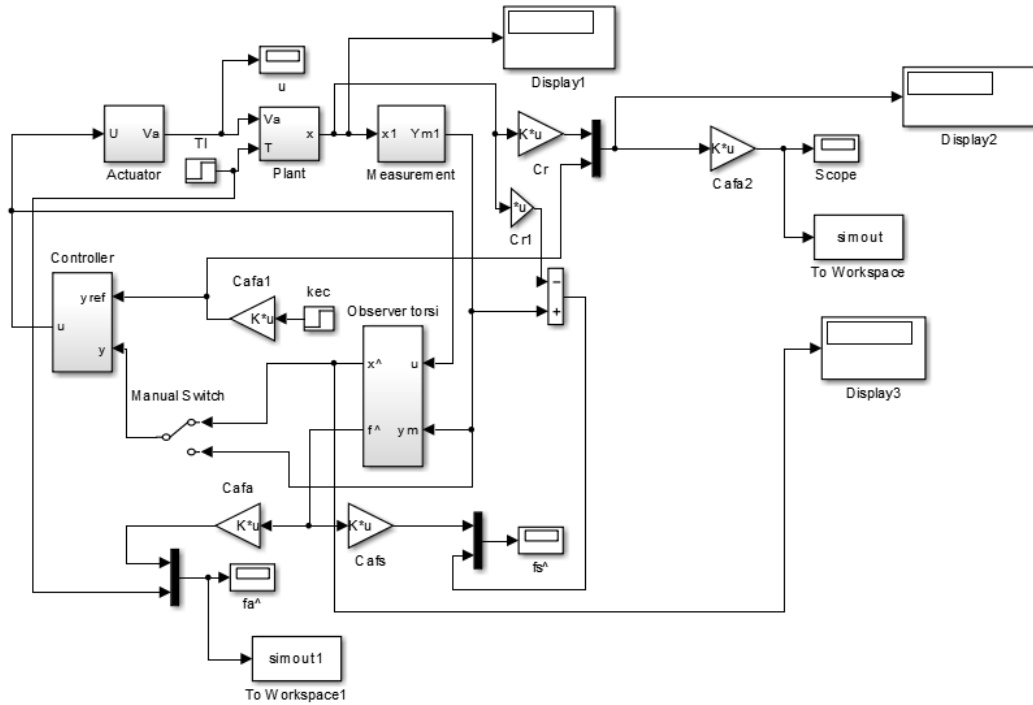
Zhang, Y & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Science Direct*, 229-252.

Zhang, Y. (2003). Bibliographical Review on Reconfigurable Fault - Tolerant Control Systems,. 265-276.

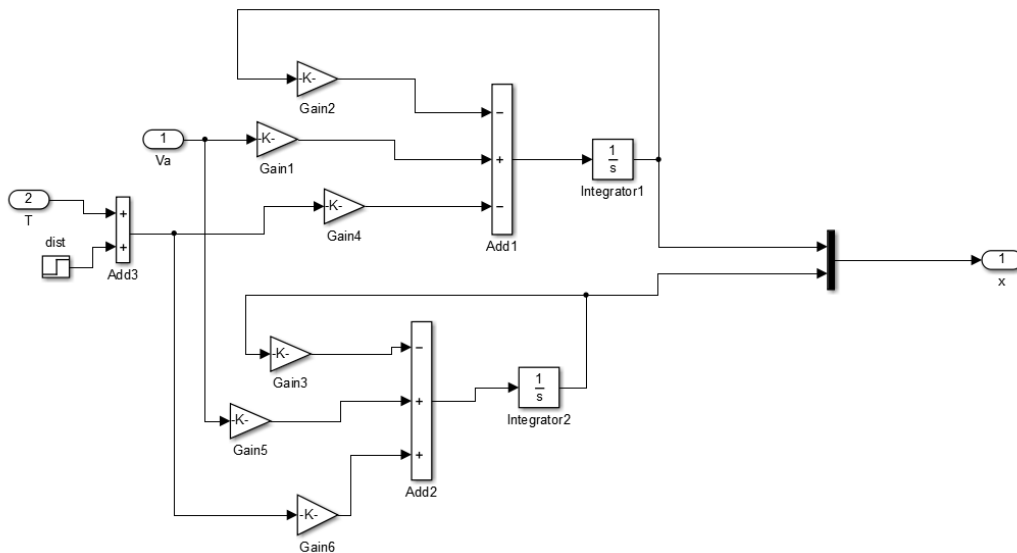
Halaman ini sengaja dikosongkan

LAMPIRAN

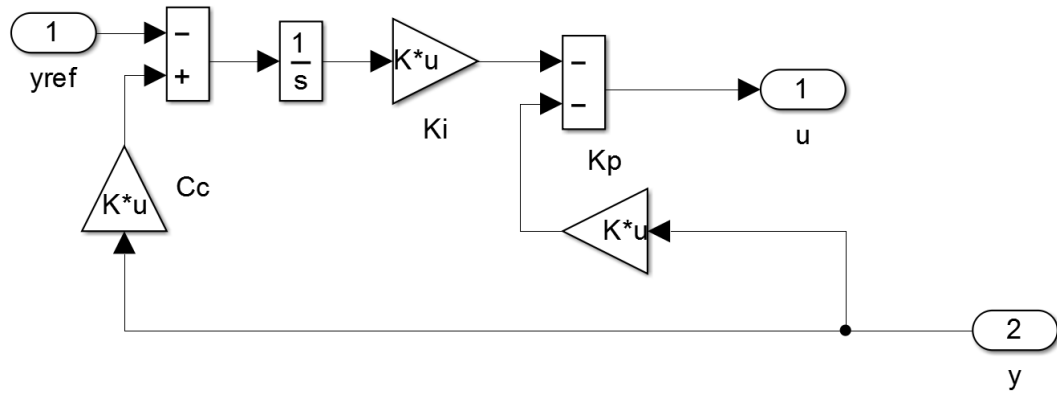
SIMULINK MATLAB KESELURUHAN PERANCANGAN SISTEM



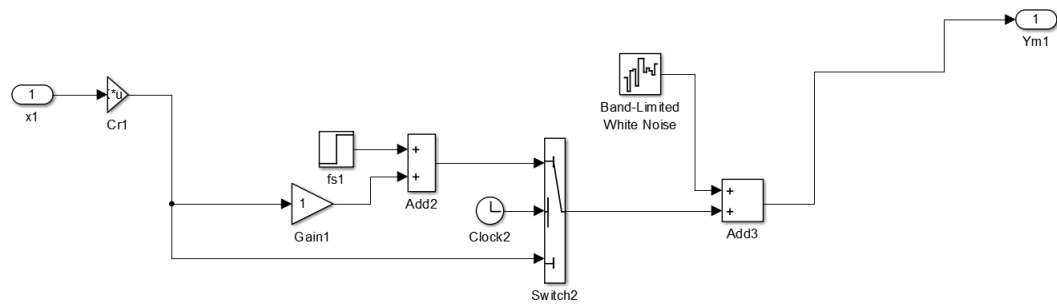
SIMULINK MATLAB PLANT MOTOR DC



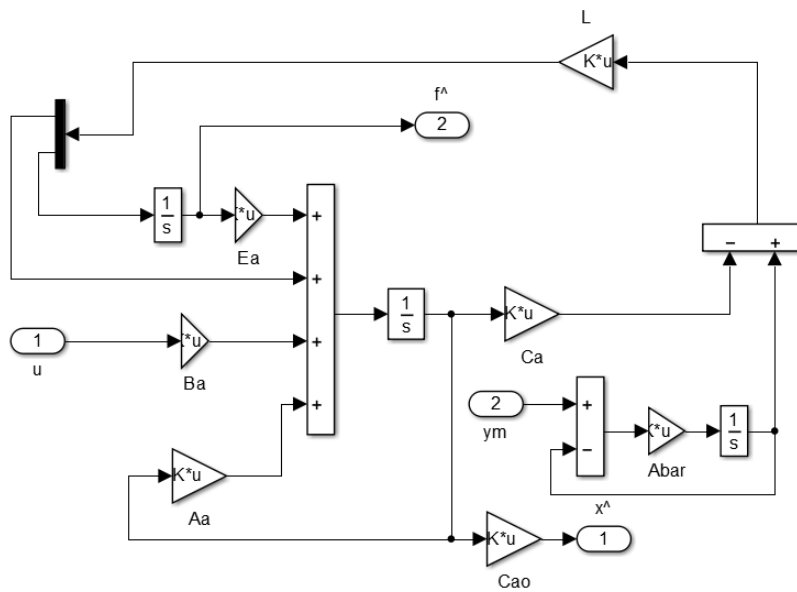
SIMULINK MATLAB KONTROLER



SIMULINK MATLAB SENSOR ARUS



SIMULINK MATLAB OBSERVER



SOURCE CODE PROGRAM MATLAB SISTEM MOTOR DC

```
A = [-0.2 0
      0 -100];
B = [0.56
      8.6];
C = [1 0;0 1];
Cr = [1 0];
D = 0;
nx=size(A,1);
nu=size(B,2);
nr=1; % number of reference inputs
% design PI controller
Ac = [A,zeros(nx,nr);C(1:nr,:),zeros(nr)];
Bc = [B;0];
p1 = -15 + 5i;
p2 = -15 - 5i;
p3 = -20;
Kc = place(Ac,Bc,[p1 p2 p3]);
Kp = Kc(1:2);
Ki = Kc(3);
Dd = [-1 0;0 -0.2];
Dw = 1;
F = 1;
E = [-0.022; 6.3];
nw = size(Dw,2);
nd = size(Dd,2);
nfs = size(F,2);
nfa = size(E,2);
Cs = [0 1];
ny=size(Cs,1);
Abar = 10*eye(ny);
Aa = [A zeros(nx,ny); Abar*Cs -Abar*eye(ny)];
Ba = [B; zeros(ny,nu)];
Ea = [E ; zeros(ny,nfa)];
%Fa = [zeros(nx,nw); Dw];
%Fa = [Dd;zeros(ny,nd)];
Fa = [Dd zeros(nx,nw);zeros(ny,nd) Abar*Dw];
Ca = [zeros(ny,nx) eye(ny)];
Cao = [Cs zeros(ny)];
Cafa = [ones(1,nfa)];
%Bo = [zeros(n+p,fa+fs); eye(fa+fs)];
Bo = [Fa zeros(nx+ny,nfa); zeros(nfa,nd+nw) eye(nfa)];
%% FTC against actuator and/or sensor failure
Aas=[Aa Ea; zeros(nfa,nx+ny) zeros(nfa)];
Cas=[Ca zeros(ny,nfa)];
R4=eye(ny)*1;
Q4=eye(nx+ny+1)*1e2;
L4=lqr(Aas',Cas',Q4,R4);
L=L4';
Lqr=L4';
```

Data Validasi MAPE Kecepatan

<i>Real</i>	Model	MAPE
841,69	839,865	0,002168
843,88	842,055	0,002163
846,07	844,61	0,001726
847,895	846,8	0,001291
850,085	848,99	0,001288
851,91	851,18	0,000857
853,37	853,37	0
854,465	855,195	-0,00085
854,83	857,385	-0,00299
854,465	859,575	-0,00598
853,37	861,765	-0,00984
852,64	863,59	-0,01284
853,005	865,78	-0,01498
854,1	867,605	-0,01581
856,29	869,795	-0,01577
859,575	871,62	-0,01401
864,685	873,445	-0,01013
869,43	875,635	-0,00714
873,81	877,46	-0,00418
878,555	879,285	-0,00083
881,84	881,11	0,000828
884,03	882,935	0,001239
885,855	884,76	0,001236
886,95	886,585	0,000412
887,315	888,41	-0,00123
886,585	890,235	-0,00412
886,585	892,06	-0,00618
887,315	893,52	-0,00699
889,14	895,345	-0,00698
890,965	897,17	-0,00696
892,79	898,63	-0,00654
893,885	900,455	-0,00735
894,615	901,915	-0,00816
895,345	903,74	-0,00938
898,265	905,2	-0,00772
902,28	906,66	-0,00485
905,93	908,485	-0,00282
909,215	909,945	-0,0008
911,04	911,405	-0,0004

911,04	912,865	-0,002
910,31	914,325	-0,00441
909,945	915,785	-0,00642
909,945	917,245	-0,00802
909,215	918,705	-0,01044
909,58	920,165	-0,01164
910,675	921,625	-0,01202
910,675	922,72	-0,01323
910,675	924,18	-0,01483
911,405	925,64	-0,01562
912,135	926,735	-0,01601
912,865	928,195	-0,01679
914,325	929,655	-0,01677
915,42	930,75	-0,01675
916,515	932,21	-0,01712
917,61	933,305	-0,0171
919,07	934,4	-0,01668
921,26	935,86	-0,01585
922,355	936,955	-0,01583
922,355	938,05	-0,01702
923,085	939,145	-0,0174
923,085	940,24	-0,01858
921,99	941,335	-0,02098
921,99	942,43	-0,02217
921,99	943,525	-0,02336
921,99	944,62	-0,02454
923,085	945,715	-0,02452
924,18	946,81	-0,02449
924,91	947,905	-0,02486
925,275	949	-0,02564
926,005	950,095	-0,02601
927,465	950,825	-0,02519
928,56	951,92	-0,02516
928,56	953,015	-0,02634
927,83	953,745	-0,02793
926,735	954,84	-0,03033
924,91	955,57	-0,03315
924,545	956,665	-0,03474
925,64	957,395	-0,03431
927,1	958,49	-0,03386
929,29	959,22	-0,03221
932,575	959,95	-0,02935

936,225	961,045	-0,02651
938,78	961,775	-0,02449
942,065	962,505	-0,0217
945,35	963,235	-0,01892
947,54	963,965	-0,01733
949,365	965,06	-0,01653
951,19	965,79	-0,01535
951,555	966,52	-0,01573
950,46	967,25	-0,01767
949	967,98	-0,02
947,905	968,71	-0,02195
946,81	969,44	-0,0239
947,175	969,805	-0,02389
948,635	970,535	-0,02309
949,73	971,265	-0,02267
954,475	971,995	-0,01836
961,41	972,725	-0,01177
967,98	973,09	-0,00528
974,185	973,82	0,000375
977,835	974,55	0,003359
977,835	974,915	0,002986
978,565	975,645	0,002984
980,025	976,375	0,003724
981,12	976,74	0,004464
981,485	977,47	0,004091
981,12	977,835	0,003348
978,93	978,565	0,000373
976,375	978,93	-0,00262
974,55	979,295	-0,00487
973,82	980,025	-0,00637
973,82	980,39	-0,00675
976,01	980,755	-0,00486
978,2	981,485	-0,00336
980,39	981,85	-0,00149
982,58	982,215	0,000371
985,135	982,58	0,002594
986,595	983,31	0,00333
988,42	983,675	0,004801
990,245	984,04	0,006266
990,245	984,405	0,005898
989,515	984,77	0,004795
988,785	985,135	0,003691

988,055	985,5	0,002586
986,96	985,865	0,001109
987,325	986,23	0,001109
988,055	986,595	0,001478
988,055	986,96	0,001108
988,785	987,325	0,001477
989,88	987,69	0,002212
990,975	988,055	0,002947
992,07	988,42	0,003679
994,26	988,785	0,005507
996,45	988,785	0,007692
997,545	989,15	0,008416
997,91	989,515	0,008413
997,545	989,88	0,007684
994,99	989,88	0,005136
992,8	990,245	0,002574
990,61	990,61	0
988,42	990,61	-0,00222
987,325	990,975	-0,0037
987,325	991,34	-0,00407
987,325	991,34	-0,00407
987,325	991,705	-0,00444
987,325	992,07	-0,00481
986,595	992,07	-0,00555
985,865	992,435	-0,00666
985,865	992,435	-0,00666
987,69	992,8	-0,00517
989,15	992,8	-0,00369
989,88	993,165	-0,00332
989,88	993,165	-0,00332
988,055	993,53	-0,00554
985,135	993,53	-0,00852
983,675	993,53	-0,01002
983,31	993,895	-0,01076
983,675	993,895	-0,01039
984,405	993,895	-0,00964
985,865	994,26	-0,00852
986,595	994,26	-0,00777
987,69	994,26	-0,00665
989,15	994,625	-0,00554
990,245	994,625	-0,00442
990,61	994,625	-0,00405

990,245	994,99	-0,00479
988,42	994,99	-0,00665
985,135	994,99	-0,01
982,2115	994,99	-0,01301
980,755	994,99	-0,01451
979,295	995,355	-0,0164
980,025	995,355	-0,01564
982,58	995,355	-0,013
985,135	995,355	-0,01037
987,325	995,355	-0,00813
989,88	995,355	-0,00553
992,07	995,355	-0,00331
993,53	995,72	-0,0022
995,355	995,72	-0,00037
996,815	995,72	0,001098
996,815	995,72	0,001098
996,085	995,72	0,000366
995,355	995,72	-0,00037
993,895	995,72	-0,00184
992,435	995,72	-0,00331
992,07	995,72	-0,00368
992,07	995,72	-0,00368
992,435	995,72	-0,00331
993,895	995,72	-0,00184
996,085	995,72	0,000366
997,545	995,72	0,001829
999,37	995,72	0,003652
1001,195	995,72	0,005468
1001,56	995,72	0,005831
1001,56	995,72	0,005831
1001,56	995,355	0,006195
1000,1	995,355	0,004745
997,545	995,355	0,002195
995,72	995,355	0,000367
993,165	995,355	-0,00221
990,975	995,355	-0,00442
990,975	995,355	-0,00442
991,705	995,355	-0,00368
992,07	994,99	-0,00294
992,8	994,99	-0,00221
1002,29	1000,465	0,001821
1000,465	1000,1	0,000365

1000,83	1000,1	0,000729
1001,925	1000,1	0,001821
1003,02	1000,1	0,002911
1002,655	1000,1	0,002548
1000,83	1000,1	0,000729
998,64	1000,1	-0,00146
998,275	999,735	-0,00146
999,37	999,735	-0,00037
1000,83	999,735	0,001094
1002,655	999,735	0,002912
1004,115	999,735	0,004362
1003,75	999,735	0,004
1002,655	999,735	0,002912
1001,925	999,735	0,002186
1001,56	999,37	0,002187
1000,83	999,37	0,001459
1000,465	999,37	0,001094
999,37	999,37	0
997,545	999,37	-0,00183
996,45	999,37	-0,00293
996,085	999,37	-0,0033
994,99	999,37	-0,0044
993,895	999,37	-0,00551
992,435	999,005	-0,00662
989,515	999,005	-0,00959
986,96	999,005	-0,0122
985,135	999,005	-0,01408
982,945	999,005	-0,01634
980,025	999,005	-0,01937
978,565	999,005	-0,02089
977,105	999,005	-0,02241
976,01	999,005	-0,02356
976,375	999,005	-0,02318
977,105	998,64	-0,02204
977,835	998,64	-0,02128
979,66	998,64	-0,01937
982,58	998,64	-0,01634
986,96	998,64	-0,01183
992,435	998,64	-0,00625
999,735	998,64	0,001095
1007,035	998,64	0,008336
1012,145	998,64	0,013343

1015,43	998,64	0,016535
1016,89	998,64	0,017947
1016,525	998,64	0,017594
1016,89	998,275	0,018306
1018,35	998,275	0,019713
1019,445	998,275	0,020766
1020,175	998,275	0,021467
1020,175	998,275	0,021467
1018,715	998,275	0,020064
1016,525	998,275	0,017953
1014,7	998,275	0,016187
1013,24	998,275	0,014769
1011,05	998,275	0,012635
1009,225	998,275	0,01085
1006,305	998,275	0,00798
1002,655	998,275	0,004368
1001,195	998,275	0,002917
1001,195	998,275	0,002917
1002,655	997,91	0,004732
1003,75	997,91	0,005818
1004,115	997,91	0,00618
1001,925	997,91	0,004007
999,37	997,91	0,001461
997,91	997,91	0
997,91	997,91	0
999,37	997,91	0,001461
1001,56	997,91	0,003644
1003,02	997,91	0,005095
1003,02	997,91	0,005095
1003,02	997,91	0,005095
1002,655	997,91	0,004732
1002,29	997,91	0,00437
1001,925	997,91	0,004007
1001,925	997,91	0,004007
1000,1	997,91	0,00219
997,545	997,91	-0,00037
996,085	997,91	-0,00183
994,99	997,91	-0,00293
993,895	997,545	-0,00367
993,165	997,545	-0,00441
992,07	997,545	-0,00552
989,88	997,545	-0,00774

987,69	997,545	-0,00998
985,865	997,545	-0,01185
983,675	997,545	-0,0141
981,485	997,545	-0,01636
979,295	997,545	-0,01864
977,105	997,545	-0,02092
975,645	997,545	-0,02245
975,645	997,545	-0,02245
976,01	997,545	-0,02206
976,74	997,545	-0,0213

Data Validasi MAPE Arus

<i>Real</i>	Model	MAPE
0,4448	0,1733	0,0905
0,415	0,1735	0,0805
0,3736	0,1736	0,066667
0,338	0,1738	0,054733
0,3284	0,1739	0,0515
0,3596	0,1741	0,061833
0,4024	0,1742	0,076067
0,4332	0,1744	0,086267
0,4271	0,1745	0,0842
0,3918	0,1746	0,0724
0,3475	0,1748	0,057567
0,3116	0,1749	0,045567
0,304	0,1751	0,042967
0,3337	0,1752	0,052833
0,3778	0,1753	0,0675
0,4158	0,1755	0,0801
0,4139	0,1756	0,079433
0,3865	0,1757	0,070267
0,3577	0,1758	0,060633
0,341	0,176	0,055
0,3549	0,1761	0,0596
0,3974	0,1762	0,073733
0,4303	0,1763	0,084667
0,4277	0,1764	0,083767
0,3921	0,1766	0,071833
0,3501	0,1767	0,0578
0,3191	0,1768	0,047433
0,3151	0,1769	0,046067

0,3442	0,177	0,055733
0,3825	0,1771	0,068467
0,4063	0,1772	0,076367
0,3913	0,1774	0,0713
0,3488	0,1775	0,0571
0,3017	0,1776	0,041367
0,2651	0,1777	0,029133
0,2611	0,1778	0,027767
0,2947	0,1779	0,038933
0,3376	0,178	0,0532
0,3546	0,1781	0,058833
0,339	0,1782	0,0536
0,3052	0,1783	0,0423
0,2661	0,1784	0,029233
0,2484	0,1785	0,0233
0,2704	0,1786	0,0306
0,3098	0,1787	0,0437
0,3405	0,1788	0,0539
0,3406	0,1789	0,0539
0,3142	0,179	0,045067
0,2808	0,179	0,033933
0,2637	0,1791	0,0282
0,2808	0,1792	0,033867
0,3228	0,1793	0,047833
0,3623	0,1794	0,060967
0,3661	0,1795	0,0622
0,3375	0,1796	0,052633
0,3011	0,1797	0,040467
0,277	0,1797	0,032433
0,2826	0,1798	0,034267
0,3211	0,1799	0,047067
0,3662	0,18	0,062067
0,3787	0,1801	0,0662
0,353	0,1801	0,057633
0,3152	0,1802	0,045
0,2828	0,1803	0,034167
0,2756	0,1804	0,031733
0,3079	0,1804	0,0425
0,3509	0,1805	0,0568
0,379	0,1806	0,066133
0,3688	0,1807	0,0627
0,3314	0,1807	0,050233

0,291	0,1808	0,036733
0,2646	0,1809	0,0279
0,2693	0,1809	0,029467
0,3084	0,181	0,042467
0,3514	0,1811	0,056767
0,3644	0,1811	0,0611
0,3456	0,1812	0,0548
0,3187	0,1813	0,0458
0,3002	0,1813	0,039633
0,3063	0,1814	0,041633
0,3423	0,1815	0,0536
0,3785	0,1815	0,065667
0,3795	0,1816	0,065967
0,3462	0,1816	0,054867
0,3066	0,1817	0,041633
0,2787	0,1818	0,0323
0,2767	0,1818	0,031633
0,3078	0,1819	0,041967
0,3459	0,1819	0,054667
0,3638	0,182	0,0606
0,343	0,182	0,053667
0,299	0,1821	0,038967
0,2546	0,1821	0,024167
0,227	0,1822	0,014933
0,2347	0,1823	0,017467
0,2751	0,1823	0,030933
0,3136	0,1824	0,043733
0,3178	0,1824	0,045133
0,2921	0,1825	0,036533
0,2548	0,1825	0,0241
0,2246	0,1826	0,014
0,2262	0,1826	0,014533
0,2628	0,1827	0,0267
0,3007	0,1827	0,039333
0,3102	0,1827	0,0425
0,2892	0,1828	0,035467
0,2579	0,1828	0,025033
0,2369	0,1829	0,018
0,246	0,1829	0,021033
0,2864	0,183	0,034467
0,3273	0,183	0,0481
0,334	0,1831	0,0503

0,3077	0,1831	0,041533
0,2727	0,1831	0,029867
0,2489	0,1832	0,0219
0,2543	0,1832	0,0237
0,2931	0,1833	0,0366
0,3388	0,1833	0,051833
0,3483	0,1833	0,055
0,3208	0,1834	0,0458
0,2841	0,1834	0,033567
0,2542	0,1835	0,023567
0,2519	0,1835	0,0228
0,2894	0,1835	0,0353
0,3315	0,1836	0,0493
0,3521	0,1836	0,056167
0,3352	0,1836	0,050533
0,2962	0,1837	0,0375
0,2599	0,1837	0,0254
0,2434	0,1837	0,0199
0,2597	0,1838	0,0253
0,3037	0,1838	0,039967
0,3387	0,1838	0,051633
0,3347	0,1839	0,050267
0,3091	0,1839	0,041733
0,2855	0,1839	0,033867
0,2769	0,184	0,030967
0,299	0,184	0,038333
0,341	0,184	0,052333
0,3626	0,1841	0,0595
0,3438	0,1841	0,053233
0,3051	0,1841	0,040333
0,2727	0,1841	0,029533
0,2591	0,1842	0,024967
0,2755	0,1842	0,030433
0,3149	0,1842	0,043567
0,3419	0,1842	0,052567
0,329	0,1843	0,048233
0,288	0,1843	0,034567
0,2433	0,1843	0,019667
0,2114	0,1844	0,009
0,2129	0,1844	0,0095
0,251	0,1844	0,0222
0,2941	0,1844	0,036567

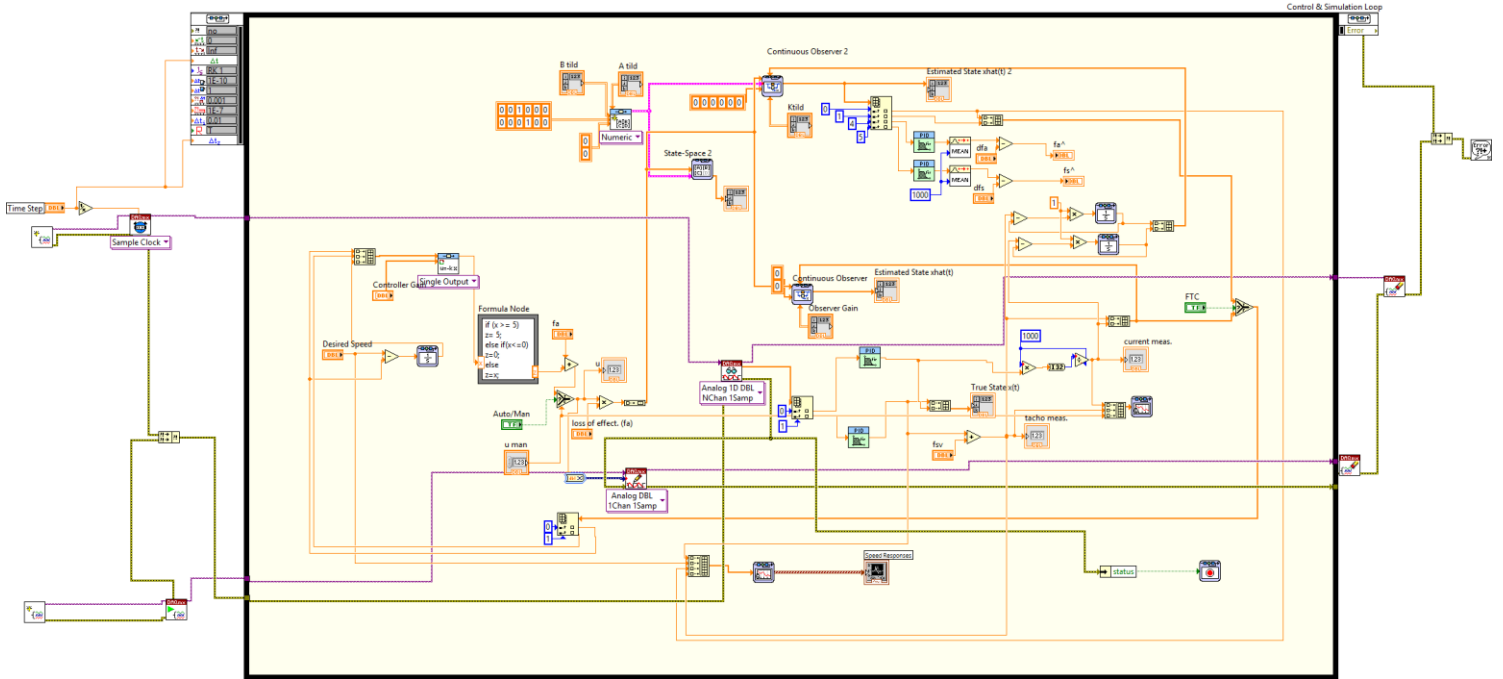
0,3026	0,1845	0,039367
0,2787	0,1845	0,0314
0,2428	0,1845	0,019433
0,2114	0,1845	0,008967
0,21	0,1845	0,0085
0,2475	0,1846	0,020967
0,2867	0,1846	0,034033
0,2963	0,1846	0,037233
0,2752	0,1846	0,0302
0,2435	0,1847	0,0196
0,2223	0,1847	0,012533
0,2319	0,1847	0,015733
0,2727	0,1847	0,029333
0,3125	0,1847	0,0426
0,3173	0,1848	0,044167
0,2897	0,1848	0,034967
0,2555	0,1848	0,023567
0,2336	0,1848	0,016267
0,2421	0,1848	0,0191
0,2826	0,1849	0,032567
0,3272	0,1849	0,047433
0,3328	0,1849	0,0493
0,3024	0,1849	0,039167
0,2668	0,1849	0,0273
0,2407	0,1849	0,0186
0,2428	0,185	0,019267
0,2832	0,185	0,032733
0,3231	0,185	0,046033
0,3256	0,185	0,046867
0,2959	0,185	0,036967
0,2569	0,185	0,023967
0,2254	0,185	0,013467
0,2235	0,1851	0,0128
0,2591	0,1851	0,024667
0,3046	0,1851	0,039833
0,3275	0,1851	0,047467
0,3152	0,1851	0,043367
0,2893	0,1851	0,034733
0,2701	0,1852	0,0283
0,2718	0,1852	0,028867
0,3035	0,1852	0,039433
0,3412	0,1852	0,052

0,3427	0,1852	0,0525
0,3101	0,1852	0,041633
0,272	0,1852	0,028933
0,2463	0,1852	0,020367
0,2472	0,1853	0,020633
0,2806	0,1853	0,031767
0,3176	0,1853	0,0441
0,3266	0,1853	0,0471
0,2999	0,1853	0,0382
0,2566	0,1853	0,023767
0,2187	0,1853	0,011133
0,2043	0,1853	0,006333
0,2252	0,1853	0,0133
0,2682	0,1854	0,0276
0,2882	0,1854	0,034267
0,2744	0,1854	0,029667
0,2423	0,1854	0,018967
0,2071	0,1854	0,007233
0,1939	0,1854	0,002833
0,2195	0,1854	0,011367
0,2611	0,1854	0,025233
0,2841	0,1854	0,0329
0,2751	0,1854	0,0299
0,2464	0,1854	0,020333
0,219	0,1855	0,011167
0,2151	0,1855	0,009867
0,2465	0,1855	0,020333
0,2895	0,1855	0,034667
0,3068	0,1855	0,040433
0,2886	0,1855	0,034367
0,2551	0,1855	0,0232
0,2282	0,1855	0,014233
0,2277	0,1855	0,014067
0,2605	0,1855	0,025
0,3064	0,1855	0,0403
0,3209	0,1855	0,045133
0,2973	0,1855	0,037267
0,2621	0,1856	0,0255
0,2325	0,1856	0,015633
0,2284	0,1856	0,014267
0,2635	0,1856	0,025967
0,3063	0,1856	0,040233

0,3273	0,1856	0,047233
0,3107	0,1856	0,0417
0,2724	0,1856	0,028933
0,2372	0,1856	0,0172
0,2229	0,1856	0,012433
0,2421	0,1856	0,018833
0,287	0,1856	0,0338
0,3178	0,1856	0,044067
0,311	0,1856	0,0418
0,2863	0,1856	0,033567
0,2651	0,1856	0,0265
0,2611	0,1856	0,025167
0,2876	0,1856	0,034
0,3272	0,1857	0,047167
0,3357	0,1857	0,05
0,3079	0,1857	0,040733
0,2699	0,1857	0,028067
0,2425	0,1857	0,018933
0,2395	0,1857	0,017933
0,2683	0,1857	0,027533
0,3066	0,1857	0,0403
0,3221	0,1857	0,045467
0,2997	0,1857	0,038
0,2573	0,1857	0,023867
0,2183	0,1857	0,010867
0,1991	0,1857	0,004467
0,2145	0,1857	0,0096
0,2571	0,1857	0,0238
0,2831	0,1857	0,032467
0,2745	0,1857	0,0296
0,2445	0,1857	0,0196
0,209	0,1857	0,007767
0,1917	0,1857	0,002
0,2117	0,1857	0,008667
0,2534	0,1857	0,022567
0,278	0,1857	0,030767
0,2714	0,1857	0,028567
0,2437	0,1857	0,019333
0,2161	0,1857	0,010133
0,2109	0,1857	0,0084
0,2408	0,1857	0,018367
0,2831	0,1857	0,032467

0,3005	0,1857	0,038267
0,2829	0,1857	0,0324
0,2502	0,1858	0,021467
0,2237	0,1858	0,012633
0,2232	0,1858	0,012467
0,2561	0,1858	0,023433
0,3019	0,1858	0,0387
0,3164	0,1858	0,043533
0,2932	0,1858	0,0358
0,2583	0,1858	0,024167
0,229	0,1858	0,0144
0,2255	0,1858	0,013233
0,2609	0,1858	0,025033
0,3034	0,1858	0,0392
0,323	0,1858	0,045733
0,3054	0,1858	0,039867
0,2672	0,1858	0,027133
0,2325	0,1858	0,015567
0,2196	0,1858	0,011267
0,2403	0,1858	0,018167
0,285	0,1858	0,033067
0,3141	0,1858	0,042767
0,3064	0,1858	0,0402
0,2817	0,1858	0,031967
0,2614	0,1858	0,0252
0,2591	0,1858	0,024433
0,2869	0,1858	0,0337

LABVIEW



Halaman ini sengaja dikosongkan

BIODATA PENULIS



Nama penulis Diyajeng Luluk Karlina dilahirkan di Tembagapura, tanggal 03 November 1995. Penulis telah menyelesaikan pendidikan di SDN 01 Klegen Madiun pada tahun 2008, pendidikan di SMPN 03 Madiun pada tahun 2011, pendidikan di SMAN 2 Madiun pada tahun 2014, pendidikan S1 di Teknik Fisika ITS pada tahun 2018. Penulis di terima sebagai mahasiswa S2 Teknik Fisika ITS dengan bidang keahlian rekayasa instrumentasi industri pada tahun 2018. Penulis fokus pada bidang minat rekayasa instrumentasi dan kontrol serta mengambil topik penelitian metode fault tolerant control yang diaplikasikan pada plant motor DC. Penulis dapat dihubungi melalui email diyajengluluk@gmail.com.