



TUGAS AKHIR - TF 181801

PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN POSISI SISTEM SERVO MODULAR MS150 DC

YULI MAULIZA
NRP. 02311640000111

Dosen Pembimbing:
Dr. Bambang L. Widjiantoro, S.T., M.T.

Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya
2020



TUGAS AKHIR - TF 181801

PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN POSISI SISTEM SERVO MODULAR MS150 DC

**YULI MAULIZA
NRP. 02311640000111**

**Dosen Pembimbing:
Dr. Bambang L. Widjiantoro, S.T., M.T.**

**Departemen Teknik Fisika
Fakultas Teknologi Industri Dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya
2020**

Halaman ini sengaja dikosongkan



FINAL PROJECT- TF 181801

***PASSIVE FAULT TOLERANT CONTROL DESIGN
FOR POSITION CONTROL OF MODULAR SERVO
SYSTEM MS150 DC***

**YULI MAULIZA
NRP. 02311640000111**

**Supervisors:
Dr. Bambang L. Widjiantoro, S.T., M.T.**

***Department Of Engineering Physics
Faculty of Industrial Technology and System Engineering
Institut Teknologi Sepuluh Nopember
Surabaya
2020***

Halaman ini sengaja dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini.

Nama : Yuli Mauliza
NRP : 02311640000111
Departemen/ Prodi : Teknik Fisika/ S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "*PERANCANGAN PASSIVE FAULT TOLERANT CONTROL (PFTC) PADA PENGENDALIAN POSISI SISTEM SERVO MODULAR MS150 DC*" adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya-benarnya.

Surabaya, 24 Juli 2020

Yang membuat pernyataan,



Yuli Mauliza

NRP. 02311640000111

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN
TUGAS AKHIR
PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC)
PADA PENGENDALIAN POSISI SISTEM SERVO MODULAR MS150
DC

Oleh:

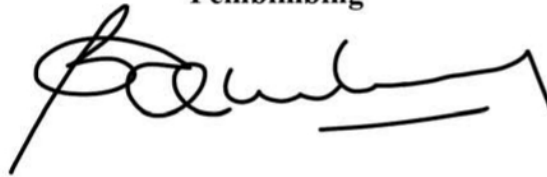
Yuli Mauliza

NRP. 02311640000111

Surabaya, 24 Juli 2020

Menyetujui,

Pembimbing



Dr. Bambang L. Widjiantoro, S.T., M.T

NIP. 19650309 1990021 001

Mengetahui,

Kepala Departemen
Teknik Fisika FTIRS-ITS



Dr. Suryanto, S.T., M.T

NIP. 19711113 199512 1 002

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN POSISI SISTEM SERVO MODULAR MS150 DC

TUGAS AKHIR

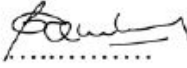

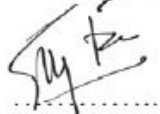


Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Progam Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Institut Teknologi Sepuluh Nopember

Oleh:

Yuli Mauliza

NRP. 02311640000111

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Bambang L. Widjiantoro, S.T., M.T  (Pembimbing)
2. Dr. Katherin Indriawati, S.T., M.T  (Ketua Penguji)
3. Dr. Suyanto, S.T., M.T  (Penguji I)
4. Detak Yan Pratama, S.T., MSc  (Penguji II)
5. Moh. Kamalul Wafi, S.T., M.Sc.DIC  (Penguji III)

SURABAYA

2020

Halaman ini sengaja dikosongkan

PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN POSISI SISTEM SERVO MODULAR MS150 DC

Nama : Yuli Mauliza
NRP : 02311640000111
Departemen : Teknik Fisika FTIRS –ITS
Dosen Pembimbing : Dr. Bambang L. Widjiantoro, S.T.,
M.T

ABSTRAK

Pengendalian posisi motor DC dengan sistem *servo modular* MS150 DC sulit mendapatkan performansi yang tinggi ketika terjadi kesalahan pada sensor dan atau aktuator. Oleh karena itu dilakukan perancangan sistem *Passive Fault Tolerant Control* (PFTC) pada pengendalian posisi sistem servo modular MS150 DC dengan kesalahan pada aktuator dan sensor sehingga dapat mengestimasi dan mengakomodasi kesalahan. Langkah pertama yang dilakukan adalah memodelkan servo modular MS150 DC. Langkah kedua adalah merancang sistem pengendali *cascade*. Langkah ketiga adalah merancang *observer* untuk mengestimasi nilai *state* yaitu posisi dan kecepatan. Langkah keempat adalah pengintegrasian antara *observer* dan sistem kontrol (PFTC) dan langkah terakhir pengujian performansi sistem PFTC. Hasil pengujian menunjukkan bahwa sistem PFTC mampu menjaga performansi sistem. Kesalahan bias maksimum pada sensor yang mampu diatasi sebesar 100%. Respon sistem dengan PFTC tidak menghasilkan *maximum overshoot* maupun *error steady state* sedangkan sistem tanpa PFTC menghasilkan *error steady state* sebesar 100%. Kesalahan bias maksimum pada aktuator yang mampu diatasi sebesar 80%. Sistem dengan PFTC menghasilkan *maximum overshoot* sebesar 25,2%. Sistem tanpa PFTC mengalami *overshoot* dan *error steady state* masing-masing sebesar 25,81% dan 100%.

Kata Kunci: Aktuator, Pengendali *cascade*, Sensor, Sistem PFTC, Sistem Servo Modular MS150 DC

Halaman ini sengaja dikosongkan

PASSIVE FAULT TOLERANT CONTROL DESIGN FOR POSITION CONTROL OF MODULAR SERVO SYSTEM MS150 DC

Name : Yuli Mauliza
NRP : 02311640000111
Department : Engineering Physics FTIRS –ITS
Supervisors : Dr. Bambang L. Widjiantoro, S.T.,
M.T

ABSTRACT

Control of the position of the DC motor with the MS150 DC modular servo system is difficult to get high performance when an error occurs in the sensor and or actuator. Therefore a Passive Fault Tolerant Control system was designed in the MS150 DC modular servo system position control with errors on the actuator and sensor so that it can estimate and accommodate errors. The first step is to model the MS150 DC modular servo. The second step is to design a cascade control system. The third step is to design an observer to estimate the state value, namely position, and speed. The fourth step is the integration between the observer and the control system (PFTC) and the final step is the PFTC system performance testing. Maximum bias error on sensors that can be overcome by 100%. System response with PFTC does not produce maximum overshoot or steady-state error while systems without PFTC produce a steady-state error of 100%. The maximum bias error on the actuator can be overcome by 80%. Systems with PFTC produce a maximum overshoot of 25.2%. Systems without PFTC experience overshoot and steady-state errors respectively at 25.81% and 100%.

Keywords: *Actuator, Cascade Controller, Sensor, PFTC System, MS150 DC Modular Servo System*

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 tugas akhir pada jenjang sarjana yang berjudul “Perancangan *Passive Fault Tolerant Control (PFTC)* Pada Pengendalian Posisi Sistem Servo Modular MS150 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. Kedua orang tua (Bapak Afifuddin dan Ibu Nurfadhilah) tercinta beserta saudara-saudara penulis yang mendukung dan memberikan segala cinta, kasih sayang, doa, perhatian, serta dukungan moril dan materiil.
2. Bapak Dr. Suyanto, S.T., M.T selaku Kepala Departemen Teknik Fisika – FTIRS – ITS.
3. Ibu Dr. Ing Doty Dewi Risanti, S.T., M.T selaku dosen wali..
4. Bapak Dr. Bambang L. Widjiantoro, S.T., M.T selaku dosen pembimbing tugas akhir yang telah memberikan bimbingan, arahan dan ilmu pengetahuan demi kelancaran mengerjakan Tugas Akhir ini.
5. Ibu Dr. Katherin Indriawati, S.T., M.T selaku kepala bidang minat Sistem Tertanan dan Siber-fisik
6. Seluruh dosen dan karyawan Departemen Teknik Fisika – FTIRS – ITS yang telah memberi dukungan dan ilmunya kepada penulis .
7. Kedua orang tua dan keluarga yang telah memberikan dukungan, doa dan perhatiannya yang luar biasa.
8. Sahabat-sahabat saya yang selalu memberikan semangat serta support.
9. Teman-teman mahasiswa ITS dan khususnya teman-teman di Departemen Teknik Fisika Angkatan 2016 terima kasih atas bantuan dan dukungannya.
10. Dan semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu sehingga dalam penyelesaian Tugas Akhir ini banyak kemudahan yang diperoleh

Penulis menyadari bahwa penulisan laporan tugas akhir ini tidaklah sempurna. Oleh karena itu, laporan ini dapat memberikan kontribusi yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika ITS khususnya, dan civitas akademika ITS pada umumnya dan semoga laporan tugas akhir ini dapat dipergunakan dengan sebaik-baiknya.

Surabaya, 24 Juli 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
COVER PAGE.....	iii
PERNYATAAN BEBAS PLAGIASI	v
LEMBAR PENGESAHAN	vii
LEMBAR PENGESAHAN	ix
ABSTRAK	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Laporan	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Motor <i>Direct Current</i> (DC) Magnet Permanen.....	5
2.2 Servo Modular MS150 DC.....	6
2.3 Model matematik Motor DC	8
2.4 Sistem Pengendalian <i>Cascade</i>	10
2.5 Pengendali PID	11

2.6	Kesalahan pada Sistem Kontrol	13
2.7	Sistem <i>Passive Fault Tolerant Control</i> (PFTC)	14
2.8	<i>Disturbance Observer</i>	16
BAB III METODOLOGI PENELITIAN		21
3.1	Studi literatur	21
3.2	Pemodelan Servo Modular MS150 DC	22
3.3	Perancangan Sistem Kendali <i>Cascade P-PI</i>	23
3.4	Uji Performansi pengendali <i>cascades</i>	24
3.5	Perancangan <i>Observer</i> Untuk Kesalahan Aktuator	24
3.6	Perancangan <i>Estimator</i> Untuk Kesalahan Sensor	26
3.7	Perancangan <i>Passive Fault Tolerant Control</i>	26
3.8	Uji Performansi PFTC	27
BAB IV HASIL DAN PEMBAHASAN		29
4.1	Uji <i>Tracking</i> Kontroler <i>Cascade P-PI</i>	29
4.2	Uji Hasil Estimasi <i>Observer</i>	30
4.3	Uji Performansi PFTC	31
4.4	Perbandingan Performansi Sistem PFTC dan AFTC.....	38
BAB V KESIMPULAN DAN SARAN		41
5.1	Kesimpulan	41
5.2	Saran	42
DAFTAR PUSTAKA.....		43
LAMPIRAN		45
BIODATA PENULIS		47

DAFTAR GAMBAR

Gambar 2.1 Motor DC (Hudaya, 2013).....	6
Gambar 2.2 Sistem motor servo MS150 DC (Pengaturan, 2015)	6
Gambar 2.3 Model motor DC (Namazov & Basturk, 2010)	8
Gambar 2.4 Skema pengendalian <i>cascade</i> (Visioli, 2006).....	11
Gambar 2.5 Diagram blok pengendali PI.....	12
Gambar 2.6 Klasifikasi <i>Fault Tolerant Control Systems</i> (FTCS) (Mahmoud, Jiang, & Zhang, 2003).....	15
Gambar 2.7 Prinsip Kerja PFTC (Jiang & Yu, 2011)	16
Gambar 2.8 Diagram blok <i>observer</i> (Ogata K. , 1997)	17
Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir	21
Gambar 3.2 Diagram blok sistem pengendalian <i>cascade</i>	24
Gambar 3.3 Diagram blok sistem PFTC pengendalian posisi.....	26
Gambar 4.1 Respon sistem perancangan sistem kontrol <i>cascade</i>	29
Gambar 4.2 Hasil perbandingan kesalahan aktual dengan estimasi kesalahan aktuator tanpa kesalahan	30
Gambar 4.3 Hasil perbandingan kesalahan aktual dengan estimasi kesalahan aktuator dengan kesalahan	30
Gambar 4.4 Respon sistem PFTC dengan kesalahan bias 18% pada aktuator.....	31
Gambar 4.5 Respon sistem PFTC dengan kesalahan bias 18% pada sensor.....	31
Gambar 4.6 Respon sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor.....	32
Gambar 4.7 Respon sistem PFTC dengan kesalahan bias 50% pada aktuator dan sensor.....	34
Gambar 4.8 Respon sistem PFTC dengan kesalahan bias 80% pada aktuator dan 80% pada sensor.....	35
Gambar 4.9 Respon sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor.....	36
Gambar 4.10 Hasil respon uji kesalahan bias pada aktuator sebesar 16% dan pada sensor sebesar 8% dengan AFTC (Hermawan, 2018).....	38

Gambar 4.11 Hasil respon uji kesalahan bias pada aktuator sebesar 16% dan pada sensor sebesar 8% dengan PFTC..... 38

DAFTAR TABEL

Tabel 4.1 Karakteristik respon Sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor	33
Tabel 4.2 Karakteristik respon Sistem PFTC dengan kesalahan bias 50%	34
Tabel 4.3 Karakteristik respon Sistem PFTC dengan kesalahan bias 80% pada aktuator dan 80% pada sensor	35
Tabel 4.4 Performansi sistem dengan kesalahan bias 100% pada aktuator dan 100% pada sensor	36
Tabel 4.5 Performansi sistem dengan kesalahan bias 16% pada aktuator dan 8% pada sensor	39

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Motor DC merupakan sebuah perangkat elektromagnetis yang berfungsi untuk mengubah energi listrik menjadi energi mekanik. Saat ini motor DC digunakan dalam berbagai aplikasi yang memerlukan kontrol kecepatan dan kontrol posisi motor seperti pertahanan, industri, dan robotika. Hal ini dikarenakan kesederhanaan, kemudahan aplikasi, keandalan dan biaya yang rendah dari penggunaan motor DC. Dua masalah utama yang dihadapi dalam kontrol motor DC adalah sifat parameter motor yang bervariasi waktu dalam kondisi pengoperasian dan adanya noise dalam loop sistem. Untuk mengatasi permasalahan ini biasanya digunakan metode konvensional berupa kontrol PID. Kontrol PID digunakan untuk aplikasi kontrol motor DC karena strukturnya yang sederhana dan algoritma kontrol yang dapat dipahami secara intuisi (Namazov & Basturk, 2010).

Pada pengendalian posisi motor DC satu loop, saat gain ditingkatkan respon kecepatan motor menjadi lebih besar. Namun hal tersebut berdampak dengan meningkatnya *overshoot* sehingga sistem menjadi tidak stabil (Indriawati & Sebe, 2020). Untuk mengatasi hal tersebut, maka digunakan sistem pengendalian tertutup dengan kontrol *cascade* untuk mengendalikan kecepatan motor. Penggunaan kontrol *cascade* dapat secara luas ditemukan pada proses industri dalam rangka meningkatkan pengurangan gangguan yang terjadi pada *plant* serta akan diperoleh pengendalian yang halus, akurat, dan cepat (Vilanova, 2008).

Desain kontrol umpan balik konvensional untuk pengendalian posisi motor DC sulit mendapatkan kinerja yang tinggi dalam penerapan *tracking* apabila terjadi kesalahan pada aktuator, sensor dan komponen lainnya dari sistem. Hal ini menyebabkan performansi sistem *tracking* menurun dan sistem menjadi tidak stabil. Oleh karena itu perlu dirancang metode kontrol yang mampu memastikan kinerja sistem ketika terjadi kesalahan. Sistem kontrol ini disebut dengan *fault tolerant control systems* (FTCS) (Noura, Theilliol, Ponsart, & Chamseddine,

2009). *Fault Tolerant Control Systems* (FTCS) dapat diklasifikasikan ke dalam dua kategori yaitu *Active Fault Tolerant Control* (AFTC) dan *Passive Fault Tolerant Control* (PFTC) (Mahmoud, Jiang, & Zhang, 2003). AFTC menanggapi kesalahan yang terjadi pada sistem dengan mendesain ulang aksi pengendali berdasarkan skema *Fault Detection and Isolation* (FDD). Sedangkan PFTC didesain untuk menjadikan sistem kontrol bersifat *robust* terhadap kesalahan tanpa memerlukan proses identifikasi kesalahan yang terjadi (Klimkowski & Dybkowski, 2017).

Dalam penelitian (Pradhana, 2017) telah dilakukan perancangan sistem AFTC pada pengendalian posisi servo modular MS150 DC menggunakan sistem kendali satu *loop* dengan kesalahan pada sensor dan aktuator. Pengembangan penelitian selanjutnya dilakukan pada tahun 2018 menggunakan *sistem* kendali *cascade* dengan kesalahan pada sensor dan aktuator (Hermawan, 2018). Metode yang dipilih adalah AFTC dan diterapkan pada pengendalian posisi motor DC. Hasil penelitian menunjukkan kesalahan berhasil dideteksi dan dikompensasi.

Berdasarkan penelitian yang sudah dilakukan sebelumnya, maka akan dilakukan perancangan sistem *Passive Fault Tolerant Control* (PFTC) untuk pengendalian posisi dengan kontrol *cascade* pada sistem servo modular MS150 DC dengan kesalahan pada aktuator dan sensor. Metode PFTC dipilih karena metode ini tidak melalui mekanisme FDI sehingga dapat meminimalisir adanya *delay* saat proses estimasi kesalahan. Dengan adanya kontrol *cascade*, diharapkan *overshoot* dan osilasi akan berkurang. Selain itu, PFTC diterapkan untuk menolerir kesalahan aktuator dan sensor yang terjadi pada sistem motor DC agar sistem dapat bekerja dengan stabil.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dalam penelitian tugas akhir ini adalah sebagai berikut:

- a. Apakah algoritma *Passive Fault Tolerant Control* (PFTC) dapat diterapkan untuk struktur kontrol *cascade* pada pengendalian posisi sistem servo modular MS150 DC dengan kesalahan sensor dan aktuator?

- b. Apakah sistem PFTC mampu menjaga performansi sistem servo modular MS150 DC saat terjadi kesalahan pada sensor dan aktuator?

1.3 Tujuan

Adapun tujuan yang ingin dicapai pada penelitian tugas akhir ini yaitu:

- a. Merancang sistem pengendalian posisi pada motor DC menggunakan *Passive Fault Tolerant Control* yang mampu menangani kesalahan pada sensor.
- b. Menganalisis sistem PFTC agar mampu menjaga performansi sistem servo modular MS150 DC saat terjadi kesalahan sensor.

1.4 Batasan Masalah

Berdasarkan identifikasi masalah maka batasan masalah pada penelitian tugas akhir ini adalah sebagai berikut:

- a. Variabel yang diukur adalah kecepatan dan posisi motor dalam bentuk tegangan dengan kesalahan sensor hanya terjadi pada pengukuran posisi.
- b. Kesalahan aktuator dan sensor yang diamati adalah kesalahan minor.
- c. Sistem yang digunakan tanpa penambahan variasi beban (*torque load*).

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut :

- a. BAB I PENDAHULUAN
Pada bab I ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika laporan.
- b. BAB II TINJAUAN PUSTAKA
Pada bab II ini dibahas mengenai teori-teori yang berkaitan dengan penelitian yang akan dilakukan, seperti *Observer*, *Passive Fault Tolerant Control* (PFTC), kontrol *cascade*.
- c. BAB III METODOLOGI PENELITIAN
Pada bab III ini berisi mengenai rancangan dari penelitian yang dilakukan, metode dan langkah-langkah dalam penelitian.
- d. BAB IV ANALISA DATA DAN PEMBAHASAN

Pada bab IV ini berisi tentang analisis hasil perancangan sistem *passive fault tolerant control*.

e. **BAB V KESIMPULAN DAN SARAN**

Pada bab V ini diberikan kesimpulan tentang tugas akhir yang telah dilakukan berdasarkan hasil yang diperoleh, serta diberikan saran sebagai penunjang maupun pengembangan tugas akhir selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

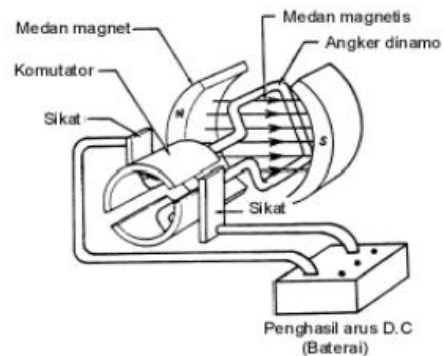
2.1 Motor *Direct Current* (DC) Magnet Permanen

Motor DC adalah sebuah mesin listrik yang prinsip kerjanya mengubah energi listrik menjadi energi mekanik. Energi mekanik yang dihasilkan merupakan energi gerak putar pada poros motor. Konversi energi tersebut dilakukan dalam waktu yang cepat dan merupakan implementasi hukum lorenzt, yaitu hukum yang berlaku untuk aliran listrik pada kawat penghantar yang berada didalam medan magnet (Sugiono, 2015).

Motor DC magnet permanen seperti pada Gambar 2.1 tersusun dari beberapa bagian seperti magnet permanen, kumparan jangkar atau angker dinamo (*armature*), komutator (*commutator*), dan sikat (*brush*). Terdapat dua prinsip dasar yang mendasari kerja motor DC. Pertama, magnet permanen sebagai stator motor tersusun dari dua magnet dengan kutub yang berbeda (kutub utara dan kutub selatan), yang saling berhadapan dan berfungsi menghasilkan fluks yang nilainya konstan dari kutub utara ke kutub selatan. Komutator yang termasuk bagian dari rotor motor terletak pada kedua ujung kumparan jangkar yang berfungsi mengumpulkan arus induksi dari jangkar dan mengkonversinya menjadi arus searah. Sikat berfungsi menyalurkan arus listrik dari sumber diluar motor ke dalam kumparan jangkar dan kumparan jangkar (*armature*) sebagai rotor motor digambarkan dalam bentuk sebuah kawat yang memiliki bentuk persegi panjang dan berfungsi untuk merubah energi listrik menjadi energi mekanik dalam bentuk gerak putar.

Kedua, sumber tegangan DC pada Gambar 2.1 diilustrasikan dengan baterai yang memiliki kutub positif dan kutub negatif, kemudian baterai dari kutub positif akan mengalirkan arus menuju sikat (*brush*) yang selanjutnya menuju komutator, kemudian arus menuju kumparan jangkar (*armature*), dan kembali lagi ke komutator, lalu sikat karbon dan terakhir menuju ke kutub negatif baterai. Dibagian stator motor, kumparan medan stator menghasilkan fluks dari kutub utara ke kutub selatan. Dengan adanya arus di dalam kumparan jangkar yang

terletak didalam medan magnet yang mengandung fluks, maka akan menghasilkan suatu gaya, yang biasa disebut gaya Lorentz (F), selanjutnya gaya tersebut akan menggerakkan kumparan jangkar untuk berputar. Komponen komutator yang selalu ikut berputar dengan rotor dan sikat karbon yang selalu diam, menjadi komponen yang akan menjaga arah arus listrik untuk selalu tetap yakni mengalir dari sisi kiri kawat angker ke kanan. Arah arus listrik yang selalu tetap di setiap setengah putaran rotor inilah yang akan membuat rotor motor listrik selalu berputar selama masih ada arus listrik yang mengalir ke kumparan jangkar (Setyaningrum, 2012).



Gambar 2.1 Motor DC (Hudaya, 2013)

2.2 Servo Modular MS150 DC

Sistem servo modular MS150 DC merupakan blok rangkaian elektronik yang digunakan untuk pengendalian kecepatan dan posisi dari motor DC jenis magnet permanen. Sistem servo modular MS150 DC terdiri dari suatu pemakaian sumber tegangan (*power supply*), *servo amplifier*, unit motor DC, *reduction gear* *tacho unit* (Pengaturan, 2015).



Gambar 2.2 Sistem motor servo MS150 DC (Pengaturan, 2015)

Sistem servo modular MS150 DC secara umum terdiri dari beberapa bagian yaitu (Setyaningrum, 2012):

a. Motor DC

Motor DC yang digunakan merupakan motor DC tipe DCM150F yang terdiri dari tiga bagian yaitu motor DC magnet permanen dengan poros yang diperpanjang, *tachogenerator* dengan output diatas unit, dan poros berkecepatan rendah.

b. *Servo amplifier*

Modular *feedback servo amplifier* SA150D terdiri dari rangkaian transistor yang dapat menggerakkan motor DC dengan dua arah putaran. Untuk mencegah adanya pembebanan berlebih pada motor DC maka terdapat meter arus yang membatasi arus maksimum yang mengalir dalam rangkaian sebesar 2 Ampere sehingga tidak terjadi *overloading*.

c. *Attenuator*

Attenuator AU150B berfungsi sebagai pemberi hambatan yang terdiri dari dua potensiometer dengan kapasitas masing-masing sebesar 10 K Ω yang dapat digunakan secara bersamaan.

d. *Output Potentiometer*

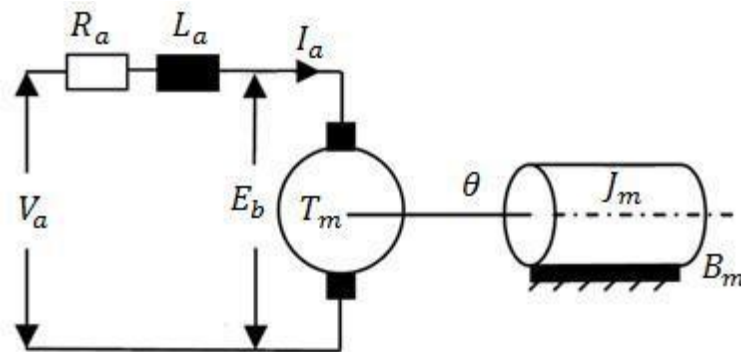
Potensiometer OP150K berfungsi sebagai sensor posisi. Potensiometer jenis ini merupakan potensiometer jenis rotasi yang dapat diputar 360 derajat. Potensiometer ini disuplai oleh *power supply* sebesar ± 15 Volt. Pada sudut nol derajat, keluaran tegangan yang dihasilkan oleh *output potentiometer* adalah sebesar 0 Volt.

e. *Tachogenerator*

Tachogenerator GT150X berfungsi untuk mengkonversi besaran mekanik menjadi besaran listrik sehingga putaran motor dapat diukur. Putaran motor dapat dinyatakan baik dalam bentuk tegangan atau rpm. Prinsip kerja dari *tachogenerator*, yaitu bagian rotor pada motor dikopel dan putaran motor tersebut digunakan untuk memutar rotor yang terdapat di *tachogenerator*, dengan prinsip kerja generator maka adanya putaran rotor akan menghasilkan tegangan keluaran.

2.3 Model matematik Motor DC

Salah satu dasar yang paling penting dalam merancang suatu sistem kendali adalah kemampuan dalam memodelkan sistem yang akan dikendalikan secara matematika. Model matematika dari suatu sistem dapat berupa persamaan diferensial, fungsi alih atau persamaan ruang keadaan (Ogata K. , 1997).



Gambar 2.3 Model motor DC (Namazov & Basturk, 2010)

Berdasarkan Gambar 2.3, tegangan sumber (V_a) untuk sistem motor DC adalah masukan yang dipasang pada armatur (R_a , L_a , θ). Tegangan tersebut oleh motor diubah menjadi gerak putar, dalam hal ini merupakan keluaran sistem dan sebagai kecepatan rotasi dari rotor motor. Sehingga pemodelan motor DC dapat didefinisikan dengan beberapa persamaan berikut (Namazov & Basturk, 2010).

$$v_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + e_b(t) \quad (2.1)$$

$$e_b(t) = K_b \cdot \omega(t) \quad (2.2)$$

$$T_m(t) = K_T \cdot i_a(t) \quad (2.3)$$

$$T_m(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \quad (2.4)$$

Keterangan:

e_a = tegangan yang dikenakan pada jangkar (V)

e_b = ggl - gaya gerak listrik balik (V)

I_a = arus kumparan jangkar (A)

R_a = tahanan kumparan jangkar (Ω)

L_a = induktansi kumparan jangkar (H)

v_a = tegangan masukan servo (V)

θ = posisi motor (rad)

ω = kecepatan angular (rad/s)

E_b = back emf (V)

B_m = koefisien gesekan viskos (Nms/rad)

K_T = konstanta torsi motor (Nm/A)

K_b = konstanta back emf (Vs/rad)

J_m = momen inersia ekuivalen dari motor (kg.m²)

Dengan mengkombinasikan Persamaan (2.1), (2.2), (2.3) dan (2.4) maka didapatkan persamaan berikut

$$v_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + K_b \cdot \omega(t) \quad (2.5)$$

$$K_T \cdot i_a(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \quad (2.6)$$

Persamaan (2.5) dan (2.6) dilakukan transformasi laplace sehingga menjadi

$$V_a(s) = R_a \cdot I_a(s) + L_a \cdot s \cdot I_a(s) + K_b \cdot W(s) \quad (2.7)$$

$$K_T \cdot I_a(s) = J_m W(s) + B_m W(s) \quad (2.8)$$

Jika arus yang diperoleh dari Persamaan (2.8) disubstitusikan ke Persamaan (2.7) maka diperoleh

$$V_a(s) = W(s) \cdot \frac{1}{K_T} [L_a \cdot J_m s^2 + (R_a \cdot J_m + L_a \cdot B_m) \cdot s + (R_a \cdot B_m + K_b \cdot K_T)] \quad (2.9)$$

Hubungan antara putaran rotor dan tegangan armatur direpresentasikan oleh fungsi ahli

$$\frac{W(s)}{V_a(s)} = \frac{K_T}{L_a \cdot J_m s^2 + (R_a \cdot J_m + L_a \cdot B_m) \cdot s + (R_a \cdot B_m + K_b \cdot K_T)} \quad (2.10)$$

Hubungan antara posisi dan kecepatan yaitu

$$\theta(s) = \frac{1}{s} W(s) \quad (2.11)$$

Sehingga diperoleh fungsi alih antara posisi dan tegangan armatur sebagai berikut

$$\frac{\theta(s)}{V_a(s)} = \frac{K_T}{L_a \cdot J_m s^3 + (R_a \cdot J_m + L_a \cdot B_m) \cdot s^2 + (R_a \cdot B_m + K_b \cdot K_T) \cdot s} \quad (2.12)$$

L_a sering kali diabaikan karena nilainya sangat kecil dibandingkan dengan R_a , sehingga

$$\frac{W(s)}{V_a(s)} = \frac{K_T}{(R_a \cdot J_m) \cdot s + (R_a \cdot B_m + K_b \cdot K_T)} \quad (2.13)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{K_T}{(R_a \cdot J_m) \cdot s^2 + (R_a \cdot B_m + K_b \cdot K_T) \cdot s} \quad (2.14)$$

Dalam bentuk persamaan ruang keadaan, Persamaan (2.11) dan (2.14) dapat dituliskan dengan memilih posisi motor dan kecepatan rotasi motor sebagai variabel *state* dan tegangan sebagai masukan serta posisi dan kecepatan motor sebagai keluaran.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B_m}{J_m} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} V \quad (2.15)$$

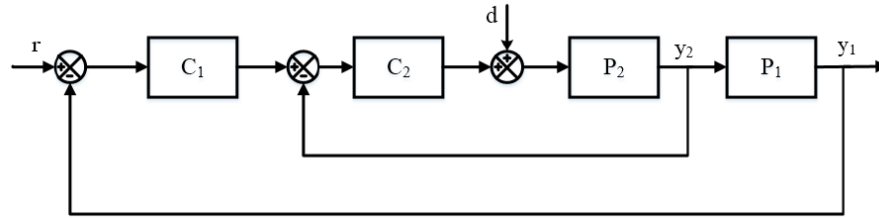
$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} \quad (2.16)$$

2.4 Sistem Pengendalian *Cascade*

Kontrol *cascade* adalah salah satu struktur kontrol kompleks yang paling sering diterapkan untuk meningkatkan kinerja penolakan gangguan dari sistem yang dikendalikan. Pengendali *cascade* dapat mengurangi efek gangguan dan meningkatkan respon dinamik pada sistem *close loop*. Dalam skema kontrol *cascade* proses memiliki satu input dan dua (atau lebih) output (Bavina, Jamliya, & Vashishtha, 2014).

Skema sistem pengendalian *cascade* ditunjukkan pada Gambar 2.4. Pengendalian *cascade* memiliki dua rantai pengendalian. *Loop* dalam adalah sebagai *loop* sekunder, sedangkan *loop* luar adalah sebagai *loop* primer. Fungsi transfer proses dinotasikan dengan $P(s) = P_2(s)P_1(s)$, y_1 adalah output primer, y_2 adalah output sekunder, C_2 adalah kontroler sekunder (*slave*) dan C_1 adalah kontroler primer (*master*). Sinyal output dari kontroler primer berfungsi sebagai *setpoint* untuk kontroler sekunder. Secara intuitif, jika $P_1(s)$ mewakili dinamika proses yang lambat dan $P_2(s)$ mewakili dinamika cepat, efektivitas sistem pengendalian *cascade* disebabkan oleh kenyataan bahwa gangguan yang

mempengaruhi loop sekunder (cepat) secara efektif dikompensasi sebelum mempengaruhi output proses utama y_1 (Visioli, 2006).



Gambar 2.4 Skema pengendalian *cascade* (Visioli, 2006)

Fungsi transfer dari *load disturbance* d terhadap variabel proses y_1 yaitu

$$T(s) = \frac{P_1(s)P_2(s)}{1 + C_2P_2(s) + C_1(s)C_2(s)P_1(s)P_2(s)} \quad (2.17)$$

Persamaan karakteristik sistem sebagai berikut:

$$1 + C_2P_2(s) + C_1(s)C_2(s)P_1(s)P_2(s) = 0 \quad (2.18)$$

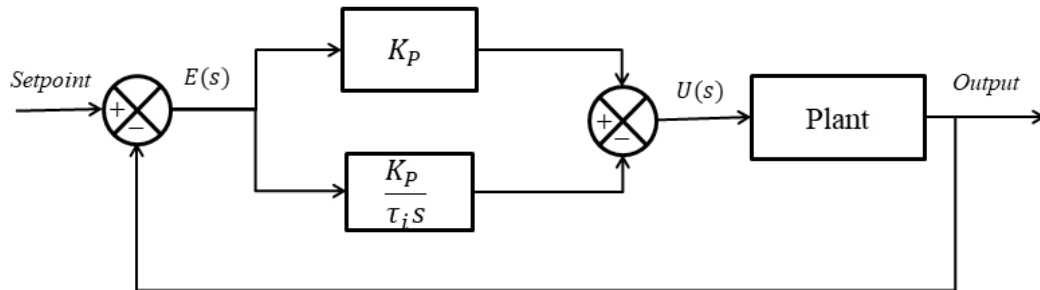
Desain keseluruhan sistem kontrol *cascade* biasanya dilakukan dengan terlebih dahulu menyetel pengendali sekunder, berdasarkan fungsi transfer proses sekunder (*loop* primer ditempatkan dalam mode manual). Kemudian, pengendali primer disetel berdasarkan fungsi transfer *loop* tertutup dari *loop* sekunder secara seri dengan fungsi transfer proses primer. Jika gangguan dapat diredam oleh pengendali sekunder maka gangguan dapat mudah ditangani secara efisien dan tanggapan sistem menjadi lebih baik.

2.5 Pengendali PID

Pengendali PID merupakan pengendali yang paling banyak diadaptasi di industri karena rasio biaya dan manfaat yang diberikannya. Pengendali ini mudah untuk digunakan dan mampu memberikan performansi yang memuaskan dalam berbagai pengendalian proses (Visioli, 2006). Karakteristik pengendali PID sangat dipengaruhi oleh aksi masing-masing parameter *proportional*, *integral* dan *derivative*.

Pengendali *proportional* memiliki keluaran yang sebanding dengan sinyal *error*. Jadi, keluaran pengendali *proportional* merupakan perkalian antara *gain proportional* dengan sinyal *error*. Pengendali PI merupakan kombinasi antara dua

jenis pengendali, yaitu P (*Proportional*) dan I (*Integral*). Diagram blok untuk pengendali PI adalah sebagai berikut.



Gambar 2.5 Diagram blok pengendali PI

Pengendali P berfungsi untuk mempercepat *rise time* agar respon sistem lebih cepat untuk mencapai *setpoint*, akan tetapi pengendali ini mempunyai kekurangan yaitu meningkatkan *offset*.

Persamaan pengendali P adalah sebagai berikut:

$$U(s) = K_p \cdot E(s) \quad (2.19)$$

dengan

$U(s)$:sinyal kontrol

$E(s)$:sinyal *error*

K_p :*gain* proporsional

Kelemahan dari pengendali P dapat dihilangkan dengan pengendali I sehingga dapat menghilangkan *offset* dan mengurangi *maximum overshoot* yang terlalu besar.

Persamaan pengendali I adalah sebagai berikut:

$$U(s) = \frac{K_p}{\tau_i s} E(s) \quad (2.20)$$

$$U(s) = \frac{K_i}{s} E(s) \quad (2.21)$$

Dimana

$U(s)$:sinyal kontrol

$E(s)$:sinyal *error*

K_i :*gain* integral

Pengendali PI menghasilkan respon lebih cepat dari pengendali I tapi mampu menghilangkan offset yang ditinggalkan pengendali P.

Persamaan pengendali PI adalah sebagai berikut:

$$U(E) = K_p \left\{ 1 + \frac{1}{\tau_i s} \right\} E(s) \quad (2.22)$$

Beberapa parameter dalam algoritma PID yang harus diketahui dalam sistem adalah (Ogata K. , 2001):

- a. *Maximum overshoot* adalah nilai maksimum respon sistem pengukuran terhadap keadaan yang diinginkan. Jika nilai *steady state* dari respon berbeda dengan nilai yang diinginkan, umumnya menggunakan *maximum percent overshoot* dengan persamaan:

$$M_p = \frac{y_p - y_{ss}}{y_{ss}} \times 100\% \quad (2.23)$$

Dengan y_p adalah nilai maksimum respon sistem dan y_{ss} adalah nilai *steady state* sistem.

- b. *Settling time* adalah waktu yang dibutuhkan respon sistem untuk mencapai dan berada di rentang $\pm 5\%$ atau $\pm 2\%$ dari nilai akhir.
- c. *Error steady state* adalah nilai *error* yang dihasilkan dari keadaan *steady* ketika belum mencapai *set point* yang diinginkan.

2.6 Kesalahan pada Sistem Kontrol

Kesalahan adalah penyimpangan yang tidak diinginkan atas paling tidak satu karakteristik atau parameter sistem dari kondisi normal yang cenderung menurunkan kinerja seluruh sistem. Pada sebuah sistem kontrol, kesalahan dapat terdiri dari beberapa jenis seperti kerusakan pada aktuator dan sensor, variasi parameter proses yang tidak normal, dan kegagalan pada peralatan karena perubahan struktural (Zhang, Swain, & Nguang, 2016). Kesalahan aktuator berkaitan dengan *loss of effectiveness* dan *bias*. Kesalahan sensor dapat disebabkan kalibrasi yang buruk atau *bias* dan perubahan karakteristik dinamik yang menyebabkan *error* pada keluaran sensor.

Kesalahan sensor dan aktuator diklasifikasikan berdasarkan tingkat keparahannya, yaitu kesalahan minor dan kesalahan mayor. Dalam hal ini, tingkat

kerusakan diukur dari dampak yang ditimbulkan terhadap performa sistem. Kesalahan minor adalah kesalahan dengan tingkat keparahan yang kecil sehingga sistem masih dapat bekerja dengan performa yang diizinkan. Pada sensor, kesalahan pembacaan akibat perubahan sensitivitas dan adanya *bias* merupakan bentuk kesalahan minor. Sedangkan pada aktuator, kesalahan minor dapat muncul dalam bentuk *loss of effectiveness* dan *bias* (Noura, Theilliol, Ponsart, & Chamseddine, 2009). Kesalahan mayor adalah kesalahan yang menyebabkan sistem tidak dapat mempertahankan performa yang ditetapkan.

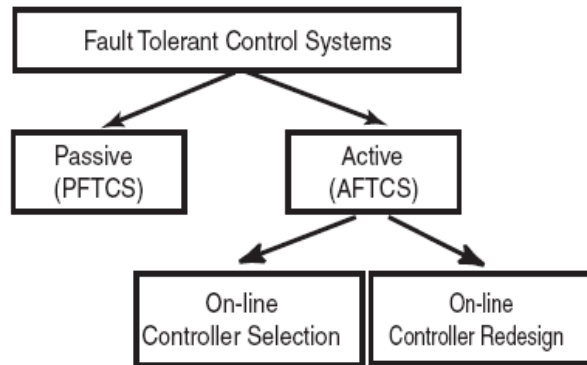
Kesalahan pada sensor menyebabkan sensor tidak dapat memberikan hasil pengukuran yang akurat yang diperlukan untuk input kontroler. Hal ini menyebabkan kontroler menghasilkan sinyal kontrol yang tidak lagi merepresentasikan situasi riil sistem. Sedangkan dampak kesalahan aktuator pada kasus *tracking*, yaitu kontroler membaca kesalahan aktuator sebagai bentuk ketidakpastian yang sepadan sehingga dalam batas tingkat kesalahan tertentu, kesalahan ini dapat ditoleransi secara pasif oleh kontroler nominal. Selain itu, dampak kesalahan aktuator adalah menimbulkan *error tracking* kondisi mantap (Shaker & Patton, 2012).

2.7 Sistem *Passive Fault Tolerant Control* (PFTC)

Fault Tolerant Control Systems (FTCS) adalah sistem kontrol yang dapat mengakomodasi kesalahan komponen sistem dan mampu mempertahankan stabilitas dan tingkat kinerja yang dapat diterima. Tidak hanya ketika sistem yang bebas kesalahan tetapi juga ketika ada kerusakan komponen. FTCS mencegah kesalahan dalam subsistem agar tidak berkembang menjadi kegagalan di tingkat sistem. FTCS berfungsi untuk meningkatkan keandalan sistem. FTCS memiliki tujuan yang berbeda untuk aplikasi yang berbeda. Sebuah FTCS dikatakan meningkatkan keandalan jika memungkinkan penyelesaian tugas yang normal bahkan setelah kesalahan komponen, FTCS dapat meningkatkan rawatan dengan menambah waktu antara tindakan pemeliharaan dan memungkinkan penggunaan prosedur perbaikan yang lebih sederhana (Mahmoud, Jiang, & Zhang, 2003).

FTCS dapat diklasifikasikan menjadi dua yaitu *Passive Fault Tolerant Control Systems* (PFTCS) dan *Active Fault Tolerant Control Systems* (AFTCS).

Pengklasifikasian ini berdasarkan pada kemampuan untuk menentukan kesalahan yang dialami sistem pada fase desain perilaku perubahan yang disebabkan oleh kesalahan dan jenis redundansi yang digunakan dalam sistem (Mahmoud, Jiang, & Zhang, 2003). Gambar 2.6 menunjukkan klasifikasi pendekatan FTCS.

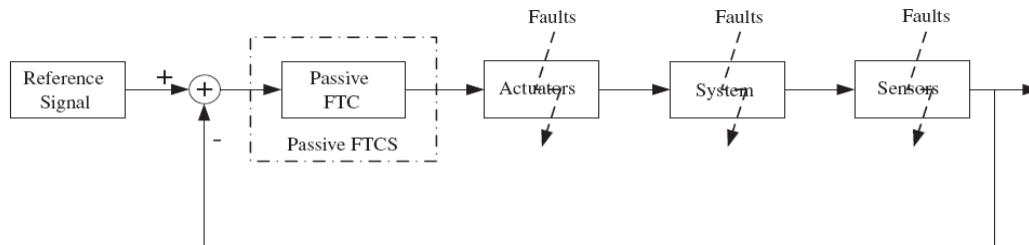


Gambar 2.6 Klasifikasi *Fault Tolerant Control Systems* (FTCS) (Mahmoud, Jiang, & Zhang, 2003)

PFTC dirancang agar dapat mempertahankan kinerja sistem kontrol ketika adanya kegagalan, sehingga kontrol yang dirancang bersifat *robust*. Sedangkan AFTC dirancang agar dapat melakukan rekonfigurasi kontrol ketika terdapat komponen yang gagal. FTCS memiliki tujuan untuk merancang sebuah sistem kontrol dimana struktur tersebut mampu untuk mencapai kestabilan sistem ketika komponen sistem tersebut berkerja secara normal maupun ketika terjadi kesalahan (Indriawati K. , 2016). Ketika *observer* telah mendeteksi kesalahan yang terjadi pada sistem, maka kesalahan tersebut akan diperbaiki agar performansi sistem tetap sesuai dengan nilai yang diinginkan. *Reconfigurable control* akan memperbaiki kesalahan sistem berdasarkan nilai estimasi kesalahan yang diperoleh (Katherin Indriawati, Trihastuti Agustinah, Achmad Jazidie, 2013).

Sistem PFTC ditunjukkan pada Gambar 2.7. 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 (Klimkowski & Dybkowski, 2017).



Gambar 2.7 Prinsip Kerja PFTC (Jiang & Yu, 2011)

2.8 Disturbance Observer

Observer merupakan algoritma yang digunakan untuk mengestimasi variabel keadaan (*state*) berdasarkan model dinamik sistem. Jika kesalahan (*disturbance*) direpresentasikan sebagai variabel *state* pada model sistem, maka kesalahan tersebut dapat diestimasi. Hal ini merupakan konsep dasar *disturbance observer* (Indriawati & Sebe, 2020). Oleh karena itu, langkah awal dalam merancang *disturbance observer* adalah memformulasikan model persamaan dinamik sistem sebagai berikut (Ogata K., 1997):

$$\dot{x} = Ax + Bu \quad (2.24)$$

$$y = Cx \quad (2.25)$$

dengan:

A = Matriks keadaan (*state*)

B = Matriks kendali

C = Matriks keluaran

D = Matriks gangguan

u = Masukan kendali

\dot{x} = Turunan vektor keadaan

y = Vektor keluaran

x = Vektor keadaan (*state*)

Model matematis dari *observer* sama seperti model matematis pada sistem dengan ditambahkan K_e (*observer gain matrix*) menjadi persamaan berikut

$$\dot{\hat{x}} = A\hat{x} + Bu + K_e(y - \hat{y}) \quad (2.26)$$

$$\hat{y} = C\hat{x} \quad (2.27)$$

dengan :

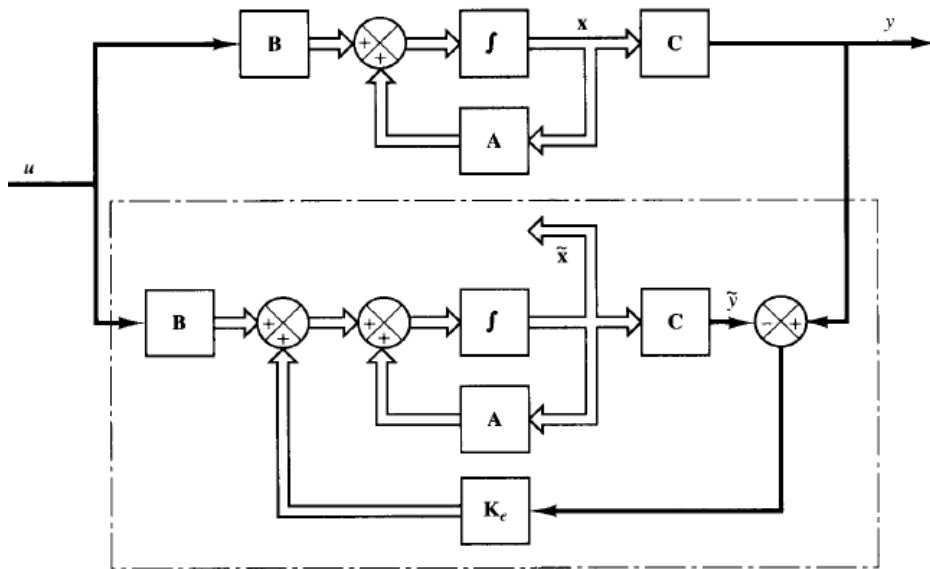
$\dot{\hat{x}}$ = Estimasi *state* \dot{x}

K_e = *Gain Observer*

\hat{x} = Estimasi *state* x

\hat{y} = Estimasi y

Diagram blok *observer* ditunjukkan pada Gambar 2.8. *Gain observer* K_e merupakan matriks konstan untuk mengoreksi perbedaan antara keluaran y aktual dan y estimasi sehingga dapat digunakan untuk mengestimasi *state*. Menggunakan prinsip dualisme, *gain observer* dapat ditentukan seperti halnya *gain controller* menggunakan metode *Linier quadratic regulator* (LQR).



Gambar 2.8 Diagram blok *observer* (Ogata K. , 1997)

Ketika terjadi kesalahan pada aktuator f_a . Persamaan ruang keadaan sistem menjadi sebagai berikut,

$$\dot{x}(t) = Ax(t) + Bu(t) + F_a \cdot f_a(t) \quad (2.28)$$

$$y(t) = Cx(t) \quad (2.29)$$

Dengan, $x(t) \in R^n$, $u(t) \in R^m$, $y(t) \in R^p$, $f_a(t) \in R^n$ berturut-turut adalah keadaan (*state*), masukan, keluaran dan kesalahan aktuator. Sedangkan A, B, C, F_a adalah matriks keadaan, masukan, keluaran dan kesalahan aktuator.

Jika diasumsikan bahwa kesalahan aktuator adalah konstan, maka

$$\dot{f}_a = f_a \quad (2.30)$$

Ketika terjadi kesalahan pada sensor posisi, maka persamaan output sistem menjadi

$$y = Cx + F_s \cdot f_s \quad (2.31)$$

Jika diasumsikan bahwa kesalahan sensor adalah konstan, maka

$$\dot{f}_s = f_s \quad (2.32)$$

Model *extended state space* yang terdiri dari vector kesalahan dengan mengkombinasikan Persamaan (2.26), (2.28) dan (2.30) menjadi seperti berikut:

$$\begin{bmatrix} \dot{x} \\ \dot{f}_a \\ \dot{f}_s \end{bmatrix} = \begin{bmatrix} A & F_a & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ f_a \\ f_s \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u \quad (2.33)$$

$$y = [C \quad 0 \quad F_s] \begin{bmatrix} x \\ f_a \\ f_s \end{bmatrix} \quad (2.34)$$

Selanjutnya Persamaan (2.31) dan (2.32) diubah dalam bentuk persamaan *observer* berikut:

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{f}}_a \\ \dot{\hat{f}}_s \end{bmatrix} = \begin{bmatrix} A & F_a & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{f}_a \\ \hat{f}_s \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} K_e \\ L_{ea} \\ L_{es} \end{bmatrix} (y - \hat{y}) \quad (2.35)$$

$$\hat{y} = [C \quad 0 \quad F_s] \begin{bmatrix} \hat{x} \\ \hat{f}_a \\ \hat{f}_s \end{bmatrix} \quad (2.36)$$

Kemudian ditambahkan *state* yang mengandung gain untuk mengestimasi kesalahan aktuator dan sensor,

$$\dot{\hat{x}} = A\hat{x} + Bu + F_a\hat{f}_a + K_e(y - \hat{y}) \quad (2.37)$$

$$\dot{\hat{f}}_a = L_{ea}(y - \hat{y}) + \hat{f}_a \quad (2.38)$$

$$\dot{\hat{f}}_s = L_{es}(y - \hat{y}) + \hat{f}_s \quad (2.39)$$

dengan L_{ea} dan L_{es} masing-masing merupakan *gain observer* yang digunakan untuk mengestimasi kesalahan sensor dan aktuator. Sedangkan *gain* K_e untuk mengestimasi *state*.

Persamaan *disturbance observer* dapat dituliskan kembali dalam bentuk sebagai berikut:

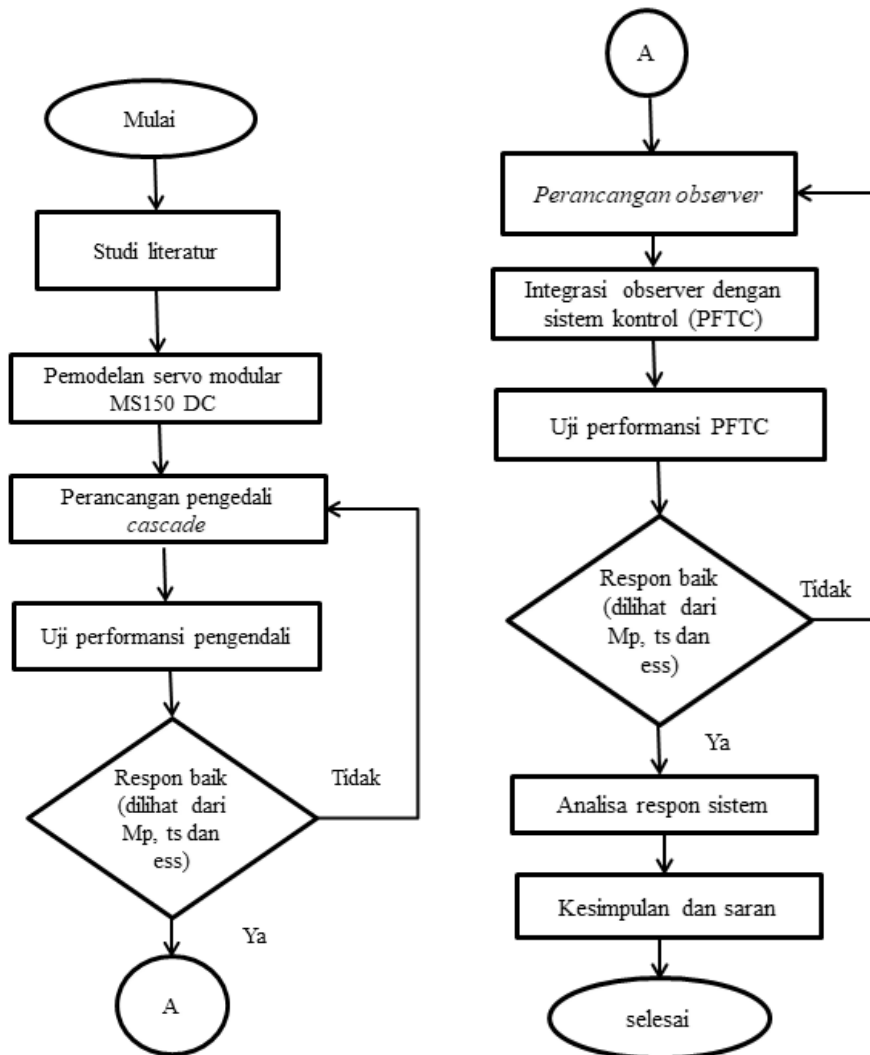
$$\hat{\tilde{x}}(k+1) = \tilde{A}.\hat{\tilde{x}}(k) + \tilde{B}.u(k) + \tilde{K}_e(\tilde{y}(k) - \hat{\tilde{y}}(k)) \quad (2.40)$$

$$\hat{\tilde{y}}(k) = \tilde{C}.\hat{\tilde{x}}(k) \quad (2.41)$$

Halaman ini sengaja dikosongkan

BAB III METODOLOGI PENELITIAN

Tahapan yang dilakukan dalam penelitian tugas akhir ini dapat ditampilkan dalam sebuah diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir

3.1 Studi literatur

Studi literatur dilakukan untuk mendapatkan pemahaman awal hingga detail mengenai topik tugas akhir. Dalam hal ini dilakukan studi literatur yang berhubungan erat dengan pengendalian posisi pada motor DC dan beberapa referensi dari penelitian sebelumnya. Pendalaman materi dilakukan untuk materi

motor DC, sistem servo modular MS150 DC, identifikasi sistem, sistem kendali PID, sistem pengendalian posisi motor DC dengan menggunakan pengendali *cascade*, *disturbance observer*, serta perancangan *Passive Fault Tolerant Control System* (PFTC).

3.2 Pemodelan Servo Modular MS150 DC

Pada tugas akhir ini pemodelan servo modular menggunakan model dari penelitian (Indriawati & Sebe, 2020). Pemodelan dilakukan dengan pemilihan model berupa persamaan fungsi transfer. Pemilihan fungsi transfer model mendekati grafik pengukuran dari *real plant* validasi *error* maksimum sebesar 5%.

Dengan menggunakan metode identifikasi sistem berupa respon step, didapatkan persamaan fungsi transfer hubungan antara kecepatan ω dan input tegangan u sebagai berikut:

$$\omega(z) = \frac{0,91}{z - 0,5948} u(z) \quad (3.1)$$

Setelah diperoleh nilai fungsi transfer untuk kecepatan motor DC maka nilai fungsi transfer diubah ke dalam bentuk persamaan ruang keadaan (*state space*) seperti berikut ini:

$$x(k + 1) = Ax(k) + Bu(k) \quad (3.2)$$

$$y(k) = Cx(k) + Du(k) \quad (3.3)$$

a. Kecepatan

$$\omega(z) = \frac{0,91}{z - 0,5948} u(z) \quad (3.4)$$

$$\omega(z)(z - 0,5948) = 0,91 u(z) \quad (3.5)$$

$$z\omega(z) - 0,5948\omega(z) = 0,91 u(z) \quad (3.6)$$

Kemudian Persamaan (3.6) diinvers Z menjadi persamaan

$$\omega(k + 1) = 0,5948\omega(k) + 0,91u(k) \quad (3.7)$$

b. Posisi

$$\frac{\theta(z)}{\omega(z)} = \frac{K_x \cdot T_s}{z - 1} \quad (3.8)$$

$$\theta(z)(z - 1) = K_x \cdot T_s \cdot \omega(z) \quad (3.9)$$

Kemudian Persamaan (3.9) diinvers Z menjadi persamaan

$$\theta(k + 1) - \theta(k) = K_x \cdot T_s \cdot \omega(k) \quad (3.10)$$

$$\theta(k + 1) = \theta(k) + K_x \cdot T_s \cdot \omega(k) \quad (3.11)$$

Diketahui,

$$K_x = 0,6$$

$$T_s = 0,1 \text{ detik}$$

Maka didapatkan,

$$\theta(k + 1) = \theta(k) + 0,06\omega(k) \quad (3.12)$$

Dari Persamaan (3.7) dan (3.12) diperoleh persamaan ruang keadaan untuk sistem posisi *servo modular* MS150 DC sebagai berikut.

$$\begin{bmatrix} \theta(k + 1) \\ \omega(k + 1) \end{bmatrix} = \begin{bmatrix} 1 & 0,06 \\ 0 & 0,5948 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0,91 \end{bmatrix} u(k) \quad (3.13)$$

$$y(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \omega(k) \end{bmatrix} \quad (3.14)$$

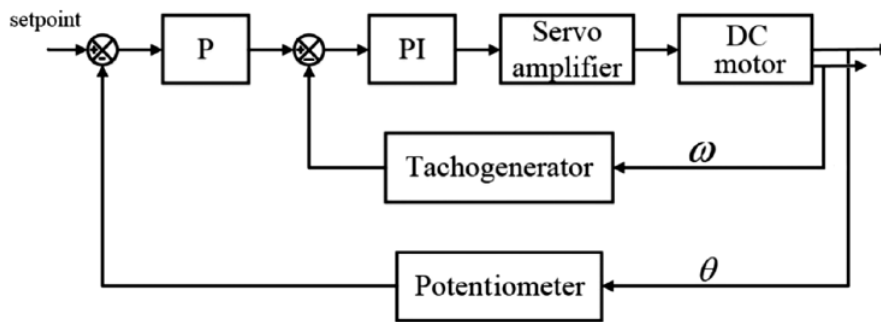
Kemudian Persamaan (3.13) dan (3.14) diubah dalam bentuk domain kontinyu dengan menggunakan *software* matlab sehingga menjadi

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 0,600 \\ 0 & -4,052 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 9,1 \end{bmatrix} u \quad (3.15)$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} \quad (3.16)$$

3.3 Perancangan Sistem Kendali *Cascade* P-PI

Perancangan sistem kendali bertujuan untuk mendapatkan *gain* kendali sehingga respon sistem dapat mencapai *setpoint*. Pada penelitian ini, variable yang dikendalikan merupakan hasil estimasi *observer* terdiri dari kecepatan dan posisi seperti ditunjukkan pada Gambar 3.2. Pada *inner loop* menggunakan algoritma kontrol berupa *proportional-integral* (PI), sedangkan pada *outer loop* menggunakan algoritma kontrol berupa *proportional* (P). Gangguan internal dapat difilter oleh *inner loop*. Setelah *inner loop* telah menghasilkan performansi yang sesuai dengan yang diinginkan, kemudian dilakukan *tuning* pada *outer loop*. Nilai K_p dan K_i diperoleh secara eksperimen dengan metode *trial and error* melalui *simulink* pada *software* matlab.



Gambar 3.2 Diagram blok sistem pengendalian *cascade*

3.4 Uji Performansi pengendali *cascades*

Uji performansi pengendali dilakukan pada saat sistem belum diberi kesalahan. Uji dilakukan dengan memberikan *setpoint* berupa sinyal step kemudian diamati responnya. Apabila respon sistem mencapai nilai *setpoint* dan memiliki performansi yang sesuai berdasarkan nilai *maximum overshoot*, *settling time* dan juga *error steady state*, maka pengendali *cascade* yang dirancang telah bekerja dengan baik.

3.5 Perancangan *Observer* Untuk Kesalahan Aktuator

Observer digunakan untuk mengestimasi kesalahan yang ada, dengan cara mengembangkan persamaan ruang keadaan sistem. Perancangan *observer* untuk mengestimasi kesalahan aktuator dilakukan dengan cara mencari nilai matriks sistem terlebih dahulu. Nilai matriks tersebut didapatkan dengan cara mengubah persamaan ruang keadaan (*state space*) sistem seperti Persamaan (3.17). Sinyal kesalahan atau gangguan direpresentasikan sebagai ketidakpastian. *State* kesalahan aktuator disimbolkan dengan f_a . Persamaan ruang keadaan yang mengandung kesalahan aktuator dapat dituliskan sebagai berikut,

$$\dot{x}(t) = Ax(t) + Bu(t) + Fa \cdot f_a(t) \quad (3.17)$$

$$y(t) = Cx(t) \quad (3.18)$$

Dengan, $x(t) \in R^n$, $u(t) \in R^m$, $y(t) \in R^p$, $f_a(t) \in R^n$ berturut-turut adalah keadaan (*state*), masukan, keluaran dan kesalahan aktuator. Sedangkan A, B, C, F_a adalah matriks keadaan, masukan, keluaran dan kesalahan aktuator.

Selanjutnya persamaan keadaan dan persamaan keluaran yang mengandung kesalahan aktuator digunakan untuk merancang *observer* sebagai berikut:

$$\dot{\hat{x}} = A\hat{x} + Bu + F_a \cdot f_a + K_e(y - \hat{y}) \quad (3.19)$$

$$\hat{y} = C\hat{x} \quad (3.20)$$

Dengan K_e adalah *gain observer*. Kemudian didefinisikan variabel baru \hat{f}_a untuk mengestimasi kesalahan aktuator dengan menggunakan persamaan berikut:

$$\dot{\hat{f}}_a = L_e + \hat{f}_a \quad (3.21)$$

dengan L_e adalah *gain observer* untuk kesalahan aktuator sehingga persamaan (3.18) dan (3.20) dapat digabungkan menjadi

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{f}}_a \end{bmatrix} = \begin{bmatrix} A_a & F_a \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{f}_a \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} K_e \\ L_e \end{bmatrix} e \quad (3.22)$$

$$\hat{y} = [C \quad 0] \begin{bmatrix} \hat{x} \\ \hat{f}_a \end{bmatrix} \quad (3.23)$$

Dengan $e = (y - \hat{y})$. Persamaan (3.22) dan (3.23) dapat diubah menjadi persamaan keadaan dan persamaan keluaran *observer* sebagai berikut:

$$\dot{\hat{\tilde{x}}} = \tilde{A} \cdot \hat{\tilde{x}} + \tilde{B} \cdot u + \tilde{K}(\tilde{y} - \hat{y}) \quad (3.24)$$

$$\hat{y} = \tilde{C} \cdot \hat{\tilde{x}} \quad (3.25)$$

\tilde{K} adalah *gain observer*. *Gain observer* ini digunakan untuk menjamin kestabilan dinamik dari error estimasi. Pada penelitian ini *observer* yang digunakan hanya menggunakan hasil pengukuran kecepatan, sehingga saat ada kesalahan sensor posisi tidak akan mempengaruhi kinerja sistem. Sehingga dengan menggunakan metode Lqr

$$\tilde{A} = \begin{bmatrix} -4,052 & 9,1 \\ 0 & 0 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} 9,1 \\ 0 \end{bmatrix}$$

$$\tilde{C} = [1 \ 0]$$

$$R = [1]$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

diperoleh nilai *gain observer* sebesar

$$\tilde{K} = \begin{bmatrix} 2,6277 \\ 1,0000 \end{bmatrix}$$

3.6 Perancangan *Estimator* Untuk Kesalahan Sensor

Dalam tugas akhir ini, kesalahan sensor dalam pengendalian posisi *servo modular* MS150 DC diestimasi di luar *observer*. Hal ini disebabkan karena hasil pengukuran arus motor DC tidak tepat sehingga tidak dapat digunakan untuk mengestimasi kesalahan sensor. Oleh karena itu, untuk mengestimasi kesalahan sensor digunakan hasil pengurangan antara sistem *real* dengan model persamaan posisi sistem yang diturunkan dari hasil pengukuran kecepatan motor DC. Estimasi kesalahan sensor posisi diperoleh dengan persamaan berikut:

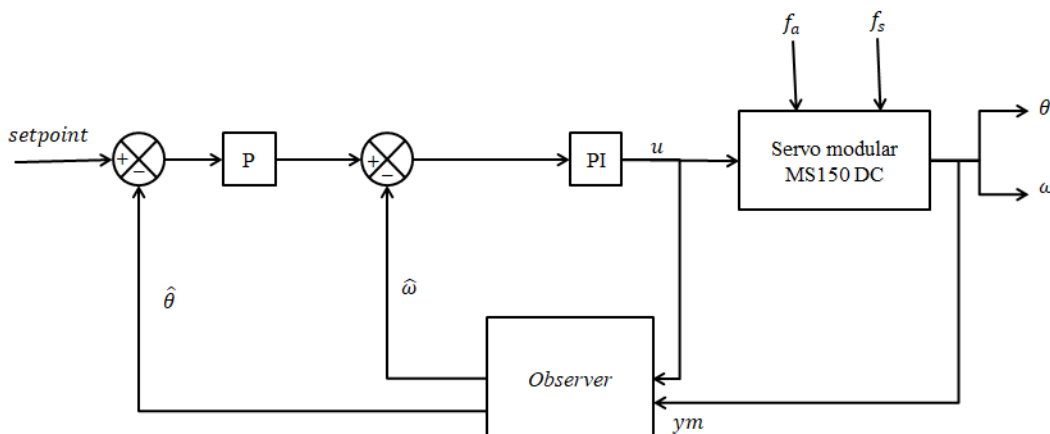
$$\hat{f}_s(k) = \hat{\theta} - \theta \quad (3.26)$$

dengan

$\hat{\theta}$ =posisi estimasi setelah diberi kesalahan

θ = posisi awal

3.7 Perancangan *Passive Fault Tolerant Control*



Gambar 3.3 Diagram blok sistem PFTC pengendalian posisi

Setelah perancangan *observer* maka dilakukan integrasi atau penggabungan *observer* dengan sistem kontrol (PFTC). Kesalahan pada aktuator disimbolkan sebagai f_a dan kesalahan sensor disimbolkan sebagai f_s . Kesalahan pada aktuator berhubungan dengan perubahan sebuah input kontrol yang akan diterapkan pada sistem. Sedangkan kesalahan sensor berhubungan dengan perubahan sebuah output pengukuran pada sistem.

Sistem PFTC bekerja dengan menggunakan hasil estimasi keadaan dari *observer*, menggantikan hasil pengukuran. Hasil estimasi *observer* terdiri dari

kecepatan dan posisi. Kemudian kontroler menghasilkan sinyal kontrol berdasarkan hasil estimasi tersebut agar respon sistem sesuai dengan yang diinginkan.

3.8 Uji Performansi PFTC

Uji performansi yang dilakukan pada sistem adalah uji kontroler, uji *observer*, dan uji kesalahan aktuator sensor. Pengujian algoritma PFTC dilakukan dengan memberikan kesalahan pada sensor dan aktuator. Kesalahan yang diberikan berupa kesalahan bias. Pada penelitian ini terdapat beberapa kasus yang diamati yaitu kesalahan aktuator, kesalahan sensor serta kesalahan sensor dan aktuator, dengan kesalahan sensor terjadi lebih dahulu. Setelah itu, respon sistem yang mengandung kesalahan dianalisa nilai *maximum overshoot* (M_p), *settling time* (t_s), dan *error steady state* (ess). Kemudian dibandingkan respon sistem antara sistem dengan PFTC dan sistem tanpa PFTC.

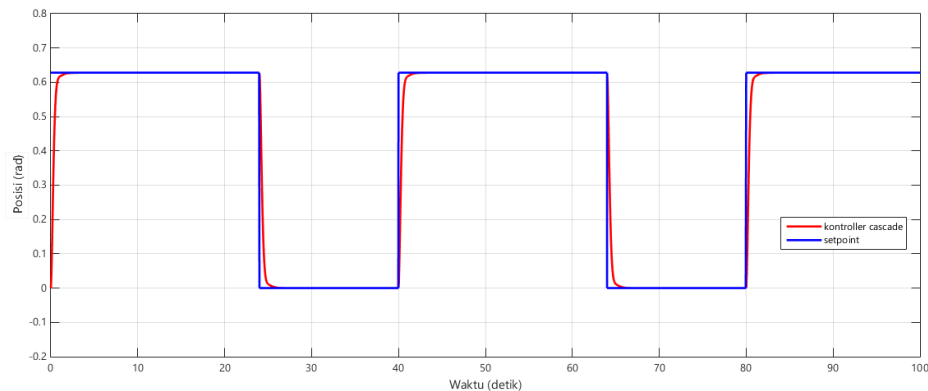
Halaman ini sengaja dikosongkan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Uji *Tracking* Kontroler *Cascade* P-PI

Perancangan sistem kontrol *cascade* digunakan untuk memperoleh nilai gain yang selanjutnya akan digunakan pada sistem. Performansi kontroler *cascade* dapat diketahui dengan dilakukan pengujian uji *tracking*. Dengan menggunakan metode *trial and error* melalui *simulink* pada *software* matlab R2014b maka diperoleh nilai *gain* kontroler *inner loop* dan *outer loop*. *Gain inner loop* masing-masing K_p dan K_i sebesar 1,8 dan 4,5, sedangkan *gain outer loop* yaitu K_p sebesar 3,8. Grafik respon sistem kontrol *cascade* pada pengendalian posisi motor DC secara simulasi ditunjukkan pada Gambar 4.1.

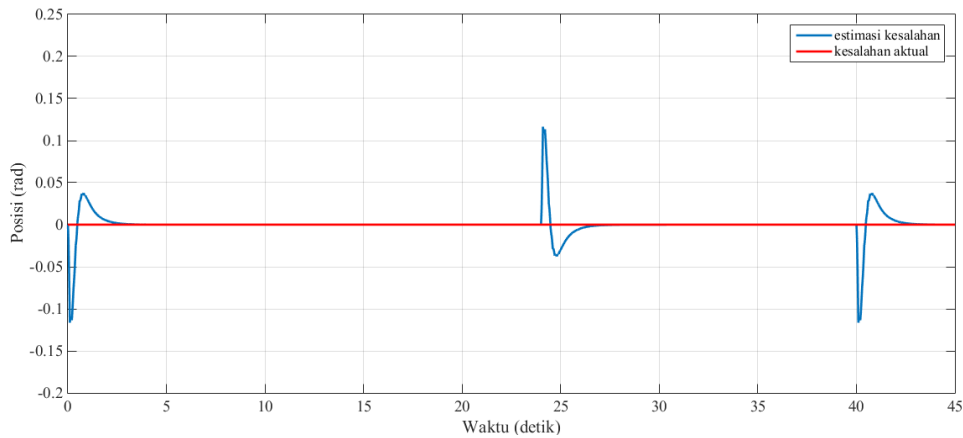


Gambar 4.1 Respon sistem perancangan sistem kontrol *cascade*

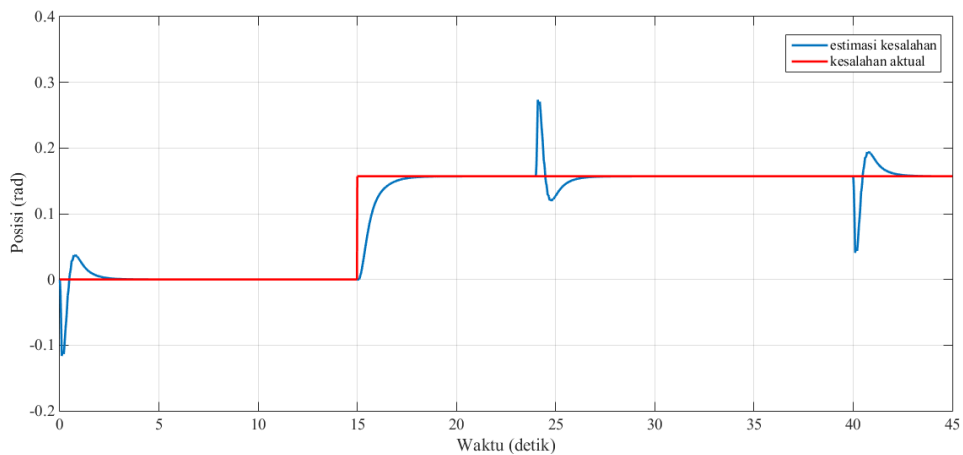
Gambar 4.1 menunjukkan respon sistem pengendali *cascade*. Grafik yang berwarna biru menunjukkan nilai *setpoint* dan grafik berwarna merah menunjukkan hasil respon kontroler. Nilai *setpoint* sebesar 0,628 radian diberikan pada waktu ke 0 detik, sehingga dari kedua grafik tersebut dapat diketahui bahwa sistem dengan kontroler P-PI yang dirancang dapat bekerja dengan baik karena mampu mencapai nilai *setpoint*. Respon kontroler yang dihasilkan memiliki karakteristik nilai *settling time* sebesar 1,89 detik dan tidak terdapat *overshoot* maupun *error steady state*.

4.2 Uji Hasil Estimasi *Observer*

Tahapan pengujian hasil estimasi *observer* bertujuan untuk menganalisa *observer* yang telah dirancang dalam mengestimasi kesalahan yang terjadi pada sistem. Pengujian dilakukan pada plant saat tidak terjadi kesalahan dan pada saat terjadi kesalahan.



Gambar 4.2 Hasil perbandingan kesalahan aktual dengan estimasi kesalahan aktuator tanpa kesalahan



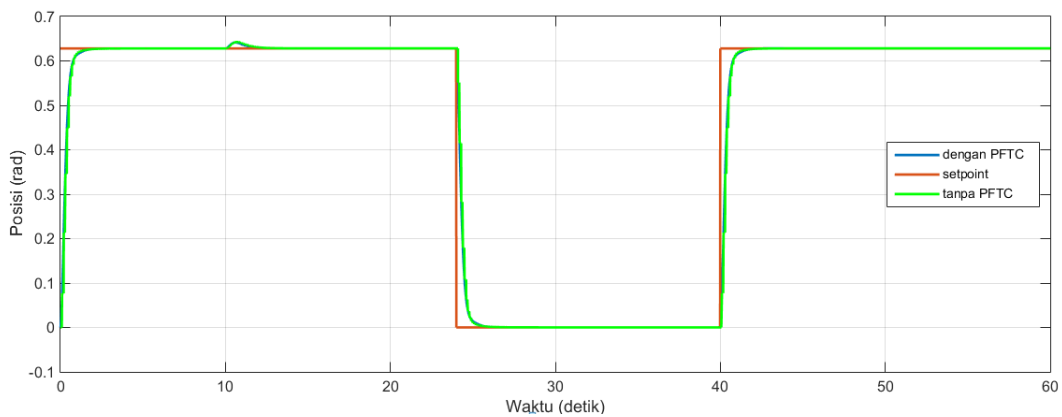
Gambar 4.3 Hasil perbandingan kesalahan aktual dengan estimasi kesalahan aktuator dengan kesalahan

Gambar 4.2 dan 4.3 menunjukkan hasil estimasi kesalahan dengan *observer* ketika tidak terjadi kesalahan dan ketika terjadi kesalahan. Garis merah menunjukkan kesalahan aktual dari aktuator dan garis biru menunjukkan estimasi kesalahan *observer*. Pada sistem tanpa kesalahan maupun dengan pemberian kesalahan diketahui bahwa hasil estimasi kesalahan aktuator mengikuti kesalahan

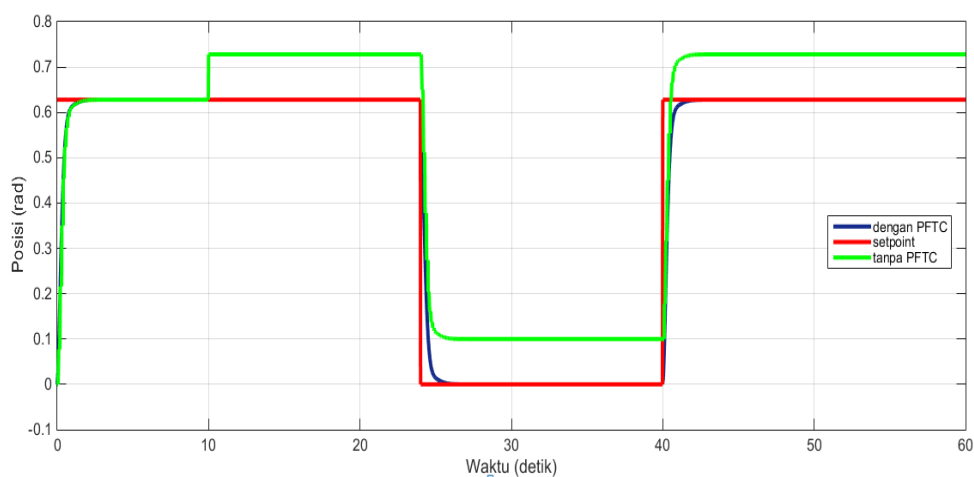
aktualnya. Sehingga dapat disimpulkan bahwa *observer* sudah bekerja dengan benar untuk mengestimasi besarnya kesalahan.

4.3 Uji Performansi PFTC

Uji performansi dilakukan untuk mengetahui algoritma *passive fault tolerant control* (PFTC) yang telah dibuat dapat bekerja dengan baik ketika diberikan kesalahan. Pengujian dilakukan dengan memberikan kesalahan pada aktuator dan sensor berupa kesalahan bias. Kesalahan bias merupakan penyimpangan nilai sinyal kendali dari nilai sinyal kendali seharusnya. Pada penelitian ini terdapat beberapa kasus yang diamati yaitu kesalahan aktuator, kesalahan sensor serta kesalahan sensor dan aktuator, dengan kesalahan sensor terjadi lebih dahulu.



Gambar 4.4 Respon sistem PFTC dengan kesalahan bias 18% pada aktuator

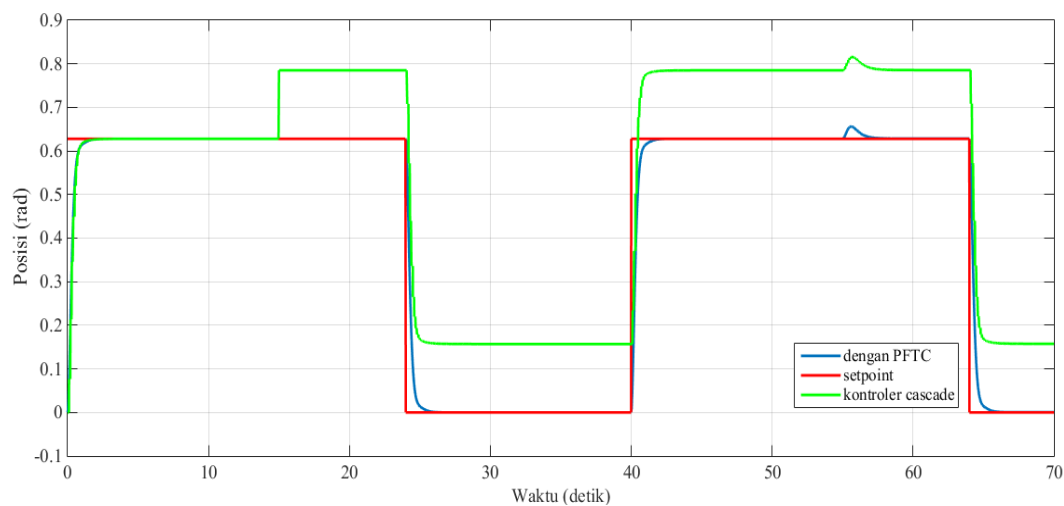


Gambar 4.5 Respon sistem PFTC dengan kesalahan bias 18% pada sensor

Gambar 4.4 adalah grafik yang merepresentasikan respon sistem PFTC ketika terjadi pada aktuator saja. Sedangkan Gambar 4.5 adalah grafik yang memrepresentasikan respon sistem PFTC ketika terjadi kesalahan pada sensor saja. Kesalahan bias yang diberikan pada kedua kasus tersebut sebesar 18% di detik ke-10. Berdasarkan kedua grafik tersebut dapat dilihat bahwa dengan kasus satu kesalahan serta diberikan nilai kesalahan yang sama tetapi memiliki respon yang berbeda. Ketika terjadi kesalahan bias pada aktuator, sistem dengan PFTC mengalami *overshoot*. sistem PFTC dapat memperbaiki kesalahan tersebut sehingga respon sistem dapat mencapai nilai *setpoint*. Pada saat terjadi kesalahan pada sensor, sistem PFTC bekerja dengan sangat baik. Hal ini terlihat dengan tidak adanya *overshoot* maupun *error steady state* serta respon dapat mencapai nilai *setpoint*.

4.3.1. Uji Kesalahan Bias pada Aktuator Sebesar 18% dan pada Sensor Sebesar 25%

Respon sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor ditunjukkan Gambar 4.1 berikut ini.



Gambar 4.6 Respon sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor

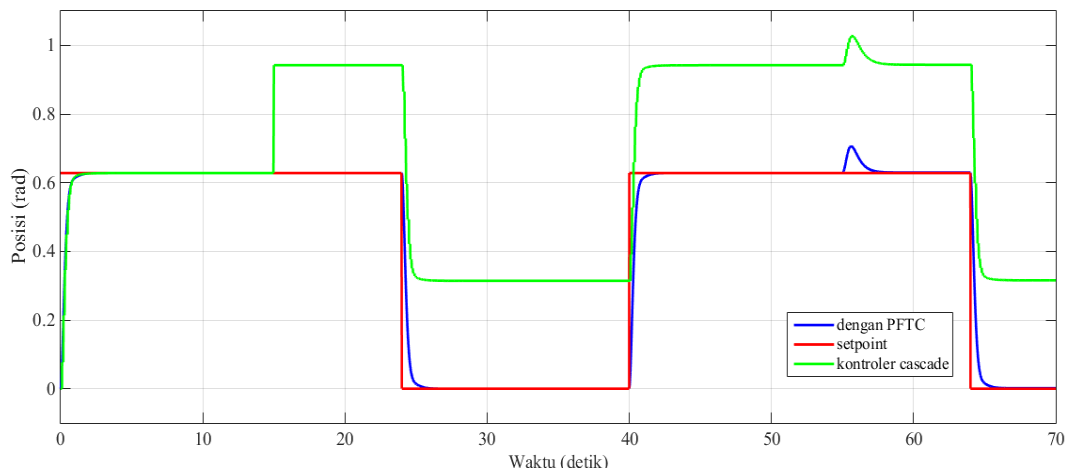
Tabel 4.1 Karakteristik respon Sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor

Karakteristik Sistem	Kesalahan Sensor		Kesalahan Aktuator	
	Dengan PFTC	Tanpa PFTC	Dengan PFTC	Tanpa PFTC
<i>Overshoot</i>	0%	0%	3,47%	3,93%
<i>Error Steady State</i>	0%	25%	0%	25%
<i>Settling Time</i>	1,89 detik	–	2,2 detik	–

Gambar 4.6 merupakan hasil dari simulasi PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor. Gambar menunjukkan bahwa dengan pemberian kesalahan bias pada sensor sebesar 0,157 di detik ke-15 dan kesalahan bias pada aktuator sebesar 0,113 di detik ke-55 maka dapat disimpulkan bahwa algoritma PFTC mampu mengatasi kesalahan tersebut sehingga respon sistem dapat kembali ke *setpoint*. Dari Tabel 4.1 dapat diketahui nilai performansi hasil simulasi PFTC. Nilai *maximum overshoot* untuk kesalahan sensor dengan PFTC sebesar 0% dan sistem tanpa PFTC menghasilkan *maximum overshoot* 0% sedangkan *maximum overshoot* untuk kesalahan aktuator dengan PFTC sebesar 3,47% dan tanpa PFTC sebesar 3,93%. *Error steady state* pada kesalahan sensor dengan PFTC sebesar 0% dan tanpa PFTC sebesar 25% sedangkan *error steady state* pada kesalahan aktuator dengan PFTC sebesar 0% dan tanpa PFTC sebesar 25%. Untuk *settling time* pada kesalahan sensor dengan PFTC sebesar 1,89 detik sedangkan *settling time* pada kesalahan aktuator dengan PFTC sebesar 2,2 detik.

4.3.2. Uji Kesalahan Bias pada Aktuator Sebesar 50% dan pada Sensor Sebesar 50%

Respon sistem PFTC dengan kesalahan bias 50% pada aktuator dan sensor ditunjukkan Gambar 4.2 berikut ini.



Gambar 4.7 Respon sistem PFTC dengan kesalahan bias 50% pada aktuator dan sensor

Tabel 4.2 Karakteristik respon Sistem PFTC dengan kesalahan bias 50%

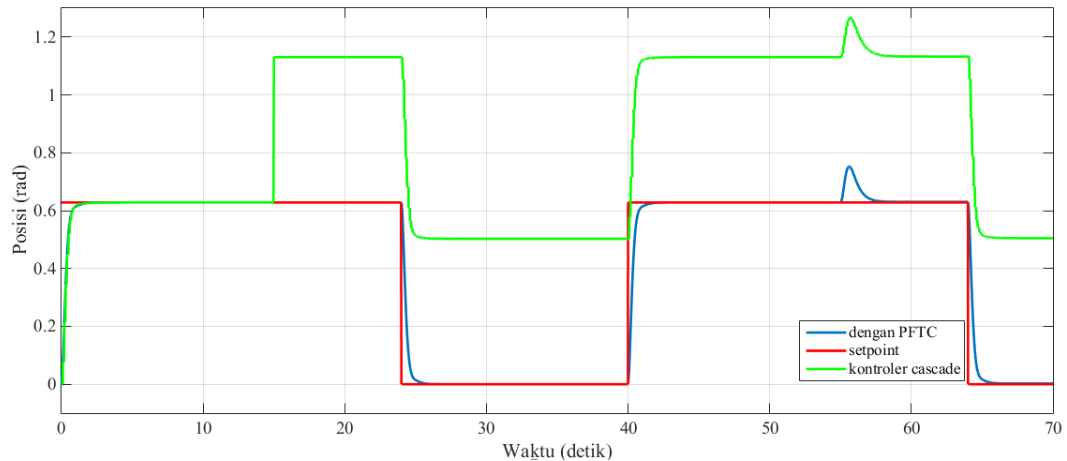
Karakteristik Sistem	Kesalahan Sensor		Kesalahan Aktuator	
	Dengan PFTC	Tanpa PFTC	Dengan PFTC	Tanpa PFTC
<i>Overshoot</i>	0%	0%	12,46%	13,57%
<i>Error Steady State</i>	0%	50%	0%	50%
<i>Settling Time</i>	1,89 detik	—	3,31 detik	—

Gambar 4.7 merupakan hasil dari simulasi PFTC dengan kesalahan bias 50% dari sinyal kendali. Gambar menunjukkan bahwa dengan pemberian kesalahan bias pada sensor sebesar 0,314 di detik ke-15 dan kesalahan bias pada aktuator sebesar 0,314 di detik ke-55 maka dapat disimpulkan bahwa algoritma PFTC mampu mengatasi kesalahan tersebut sehingga respon sistem dapat kembali ke *setpoint*. Dari Tabel 4.2 dapat diketahui nilai performansi hasil simulasi PFTC dengan kesalahan bias 50% dari sinyal kendali. Nilai *maximum overshoot* untuk kesalahan sensor dengan PFTC sebesar 0% dan sistem tanpa PFTC menghasilkan *maximum overshoot* 0% sedangkan *maximum overshoot* untuk kesalahan aktuator dengan PFTC sebesar 12,46% dan tanpa PFTC sebesar 13,57%. *Error steady state* pada kesalahan sensor dengan PFTC sebesar 0% dan tanpa PFTC sebesar 50% sedangkan *error steady state* pada kesalahan aktuator dengan PFTC sebesar 0% dan tanpa PFTC sebesar 50%. Untuk *settling time* pada kesalahan sensor dengan

PFTC sebesar 1,89 detik sedangkan *settling time* pada kesalahan aktuator dengan PFTC sebesar 3,31 detik.

4.3.3. Uji Kesalahan Bias pada Aktuator Sebesar 80% dan pada Sensor Sebesar 80%

Respon sistem PFTC dengan kesalahan bias 80% pada aktuator dan 80% pada sensor ditunjukkan Gambar 4.3 berikut ini.



Gambar 4.8 Respon sistem PFTC dengan kesalahan bias 80% pada aktuator dan 80% pada sensor

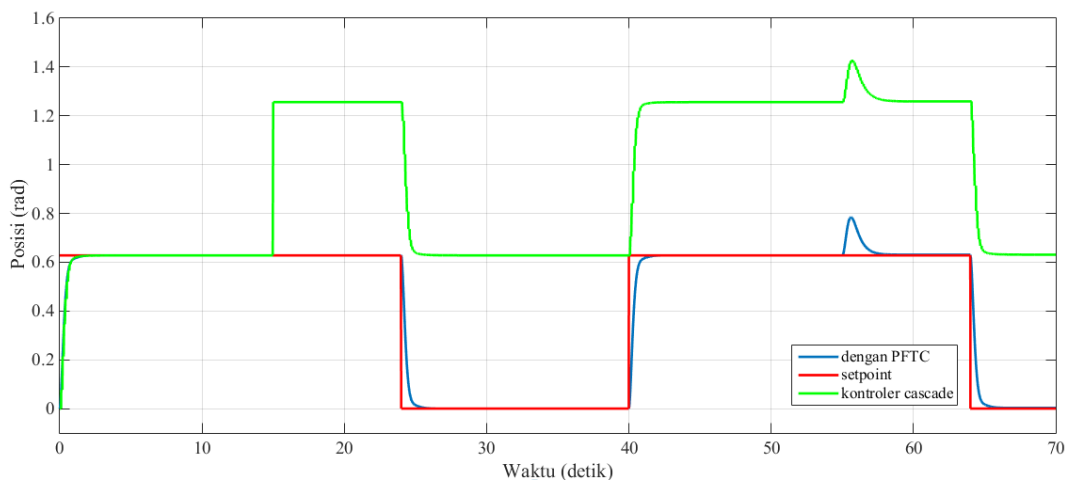
Tabel 4.3 Karakteristik respon Sistem PFTC dengan kesalahan bias 80% pada aktuator dan 80% pada sensor

Karakteristik Sistem	Kesalahan Sensor		Kesalahan Aktuator	
	Dengan PFTC	Tanpa PFTC	Dengan PFTC	Tanpa PFTC
<i>Overshoot</i>	0%	0%	19,90%	20,1%
<i>Error Steady State</i>	0%	80%	0%	80%
<i>Settling Time</i>	1,89 detik	—	4,2 detik	—

Gambar 4.8 merupakan hasil dari simulasi PFTC dengan kesalahan bias 80% pada aktuator dan 80% pada sensor. Gambar menunjukkan bahwa dengan pemberian kesalahan bias pada sensor sebesar 0,5024 di detik ke-15 dan kesalahan bias pada aktuator sebesar 0,5024 di detik ke-55 maka dapat disimpulkan bahwa algoritma PFTC mampu mengatasi kesalahan tersebut sehingga respon sistem dapat kembali ke *setpoint*. Dari Tabel 4.3 dapat diketahui

nilai performansi hasil simulasi PFTC. Nilai *maximum overshoot* untuk kesalahan sensor dengan PFTC sebesar 0% dan sistem tanpa PFTC menghasilkan *maximum overshoot* 0% sedangkan *maximum overshoot* untuk kesalahan aktuator dengan PFTC sebesar 19,90% dan tanpa PFTC sebesar 20,1%. *Error steady state* pada kesalahan sensor dengan PFTC sebesar 0% dan tanpa PFTC sebesar 80% sedangkan *error steady state* pada kesalahan aktuator dengan PFTC sebesar 0% dan tanpa PFTC sebesar 80%. Untuk *settling time* pada kesalahan sensor dengan PFTC sebesar 1,89 detik dan pada kesalahan aktuator dengan PFTC sebesar 4,2 detik.

4.3.4. Uji Kesalahan Bias pada Aktuator Sebesar 100% dan pada Sensor Sebesar 100%



Gambar 4.9 Respon sistem PFTC dengan kesalahan bias 18% pada aktuator dan 25% pada sensor

Tabel 4.4 Performansi sistem dengan kesalahan bias 100% pada aktuator dan 100% pada sensor

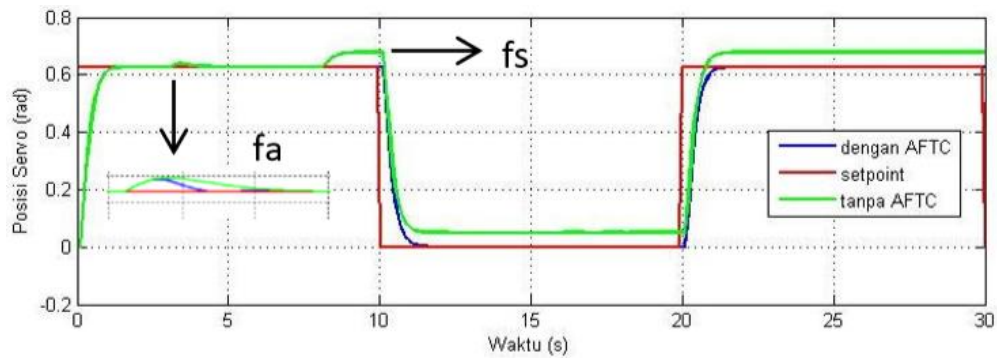
Karakteristik Sistem	Kesalahan Sensor		Kesalahan Aktuator	
	Dengan PFTC	Tanpa PFTC	Dengan PFTC	Tanpa PFTC
<i>Overshoot</i>	0%	0%	25,2%	25,81%
<i>Error Steady State</i>	0%	100%	0%	100%
<i>Settling Time</i>	1,89 detik	—	7,41 detik	—

Gambar 4.9 merupakan hasil dari simulasi PFTC dengan kesalahan bias 100% dari sinyal kendali. Gambar menunjukkan bahwa dengan pemberian kesalahan bias pada sensor sebesar 0,628 di detik ke-15 dan kesalahan bias pada aktuator sebesar 0,628 di detik ke-55 maka dapat disimpulkan bahwa algoritma PFTC mampu mengatasi kesalahan tersebut sehingga respon sistem dapat kembali ke *setpoint*. Dari Tabel 4.4 dapat diketahui nilai performansi hasil simulasi PFTC dengan kesalahan bias 100% dari sinyal kendali. Nilai *maximum overshoot* untuk kesalahan sensor dengan PFTC sebesar 0% dan sistem tanpa PFTC menghasilkan *maximum overshoot* 0% sedangkan *maximum overshoot* untuk kesalahan aktuator dengan PFTC sebesar 25,2% dan tanpa PFTC sebesar 25,81%. *Error steady state* pada kesalahan sensor dengan PFTC sebesar 0% dan tanpa PFTC sebesar 100% sedangkan *error steady state* pada kesalahan aktuator dengan PFTC sebesar 0% dan tanpa PFTC sebesar 100%. Untuk *settling time* pada kesalahan sensor dengan PFTC sebesar 1,89 detik sedangkan *settling time* pada kesalahan aktuator dengan PFTC sebesar 7,41 detik.

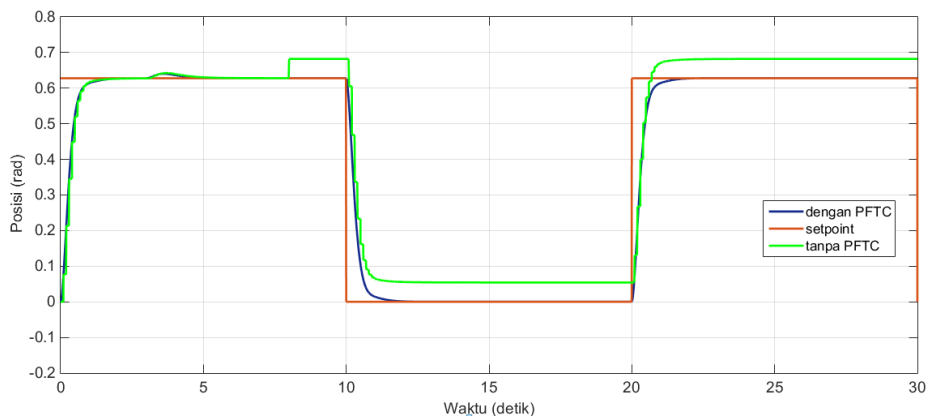
Berdasarkan hasil simulasi dapat diketahui bahwa ketika terjadi kesalahan aktuator dan sensor diperlukan algoritma PFTC untuk mengatasi kesalahan tersebut sehingga respon sistem dapat mencapai *setpoint* sedangkan tanpa menggunakan algoritma PFTC maka respon tidak dapat mencapai *setpoint*. Oleh karena itu, dapat diketahui bahwa kesalahan aktuator dan sensor tidak dapat diperbaiki dengan pengendali konvensional. Nilai *maximum overshoot* dari sistem yang menggunakan algoritma PFTC lebih kecil daripada tanpa algoritma PFTC untuk kesalahan sensor dan kesalahan aktuator. *Error steady state* dari sistem yang menggunakan algoritma PFTC juga lebih kecil daripada sistem tanpa PFTC. Selain itu, dapat diketahui bahwa nilai *settling time* sistem dengan menggunakan algoritma PFTC lebih cepat daripada sistem tanpa menggunakan algoritma PFTC. Hal tersebut terjadi dikarenakan sistem tanpa algoritma PFTC tidak bisa mencapai nilai *setpoint*. Berdasarkan hasil pengujian sistem PFTC dengan kesalahan bias 100% untuk pengendalian posisi servo modular MS150 DC dapat diketahui bahwa sistem PFTC memiliki batas toleransi kesalahan bias maksimum sebesar 80% untuk kesalahan aktuator karena *maximum overshoot* dari sistem dengan PFTC mencapai lebih dari 25%.

4.4 Perbandingan Performansi Sistem PFTC dan AFTC

Pada penelitian (Hermawan, 2018) telah dilakukan perancangan sistem pengendalian posisi servo modular dengan menggunakan algoritma AFTC. Berikut ini perbandingan performansi antara sistem PFTC yang diusulkan pada penelitian ini dan sistem dengan AFTC.



Gambar 4.10 Hasil respon uji kesalahan bias pada aktuator sebesar 16% dan pada sensor sebesar 8% dengan AFTC (Hermawan, 2018)



Gambar 4.11 Hasil respon uji kesalahan bias pada aktuator sebesar 16% dan pada sensor sebesar 8% dengan PFTC

Perbandingan performansi sistem dengan kesalahan bias 16% pada aktuator dan 8% pada sensor menggunakan AFTC dan menggunakan PFTC ditunjukkan Tabel 4.4 berikut.

Tabel 4.5 Performansi sistem dengan kesalahan bias 16% pada aktuator dan 8% pada sensor

Karakteristik Sistem	Kesalahan Sensor		Kesalahan Aktuator	
	Dengan PFTC	Dengan AFTC	Dengan PFTC	Dengan AFTC
<i>Overshoot</i>	0%	0%	3,54%	1,27%
<i>Error Steady State</i>	0%	0%	0%	0%
<i>Settling Time</i>	1,8 s	2,2 s	2,38 s	4,8 s

Gambar 4.10 menunjukkan sistem dengan AFTC sedangkan Gambar 4.11 menunjukkan sistem PFTC dengan nilai setpoint sebesar 0,628 radian. Perbandingan performansi antara sistem dengan PFTC dan sistem dengan AFTC ditunjukkan oleh Tabel 4.5. Ketika kesalahan bias 16% pada aktuator terjadi pada detik ke-3, sistem dengan AFTC memiliki nilai *overshoot* sebesar 1,27% dan tidak terdapat *error steady state*. Sedangkan sistem dengan PFTC memiliki nilai *overshoot* lebih besar yaitu sebesar 3,54% dan juga tidak terdapat *error steady state*. Selain itu *settling time* sistem dengan PFTC lebih baik dari pada sistem dengan AFTC. Hal tersebut menunjukkan bahwa sistem dengan AFTC memiliki performansi lebih baik daripada sistem dengan PFTC dalam mengkompensasi kesalahan yang ada pada aktuator.

Ketika kesalahan bias 8% pada sensor terjadi pada detik ke-8, sistem dengan AFTC maupun PFTC tidak memiliki *overshoot* dan juga tidak terdapat *error steady state*. Namun, *settling time* sistem dengan PFTC lebih baik daripada sistem dengan AFTC. Respon sistem dengan PFTC lebih cepat mencapai kondisi *steady*. Hal tersebut menunjukkan bahwa sistem dengan PFTC memiliki performansi lebih baik daripada sistem dengan AFTC dalam mengkompensasi kesalahan yang ada pada sensor.

Halaman ini sengaja dikosongkan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan, didapatkan kesimpulan dari studi mengenai perancangan sistem *Passive Fault Tolerant Control* pada pengendalian posisi sistem servo modular MS150 DC dengan kesalahan pada sensor dan aktuator adalah sebagai berikut :

- a. Telah dilakukan perancangan algoritma PFTC menggunakan kontroler *cascade* yang mampu *tracking set point* dengan nilai *gain inner loop* masing-masing K_p dan K_i sebesar 1,8 dan 4,5, serta *gain outer loop* yaitu K_p sebesar 3,8. Berdasarkan hasil respon sistem dengan nilai *gain* kendali tersebut maka diperoleh nilai *settling time* sebesar 1,89 detik dan tidak terdapat *overshoot* maupun *error steady state*.
- b. Sistem dengan algoritma PFTC dapat mengestimasi kesalahan bias pada sensor sebesar 25%, 50%, 80% dan 100%. Kesalahan bias maksimum yang mampu diatasi sebesar 100%. Sedangkan kesalahan bias pada aktuator yang dapat diestimasi oleh sistem PFTC sebesar 18%, 50%, 80% dan 100%. Namun, kesalahan bias maksimum pada aktuator yang mampu diatasi sebesar 80%.
- c. Respon yang diperoleh saat algoritma PFTC diterapkan pada sistem servo modular MS150 DC menunjukkan bahwa sistem yang dirancang dapat bekerja karena mampu mengestimasi kesalahan bias pada komponen aktuator dan sensor sehingga respon sistem dapat kembali ke nilai *setpoint*. Nilai *maximum overshoot* dari sistem yang menggunakan algoritma PFTC lebih kecil daripada tanpa algoritma PFTC untuk kesalahan sensor dan kesalahan aktuator. *Error steady state* dari sistem yang menggunakan algoritma PFTC juga lebih kecil daripada sistem tanpa PFTC. Selain itu, dapat diketahui bahwa nilai *settling time* sistem dengan menggunakan algoritma PFTC lebih cepat daripada sistem tanpa menggunakan algoritma PFTC. Hal tersebut terjadi dikarenakan sistem tanpa algoritma PFTC tidak bisa mencapai nilai *setpoint* ketika terjadi

kesalahan pada komponen aktuator dan sensor. Namun, sistem PFTC memiliki batas toleransi kesalahan bias maksimum sebesar 80% untuk kesalahan aktuator karena *maximum overshoot* dari sistem dengan PFTC mencapai lebih dari 25%.

- d. Respon sistem dengan AFTC memiliki performansi lebih baik daripada sistem dengan PFTC dalam mengkompensasi kesalahan yang ada pada aktuator dikarenakan sistem dengan AFTC memiliki nilai *overshoot* lebih kecil daripada sistem dengan PFTC. Sedangkan dalam mengkompensasi kesalahan sensor, sistem dengan PFTC lebih baik daripada sistem dengan AFTC. Hal ini dikarenakan respon sistem dengan PFTC lebih cepat mencapai kondisi *steady*.

5.2 Saran

Dari tugas akhir ini terdapat saran yang dapat digunakan untuk penelitian selanjutnya, yakni pengembangan algoritma PFTC dapat dilakukan secara *real time* pada servo modular MS150 DC dengan adanya variasi beban.

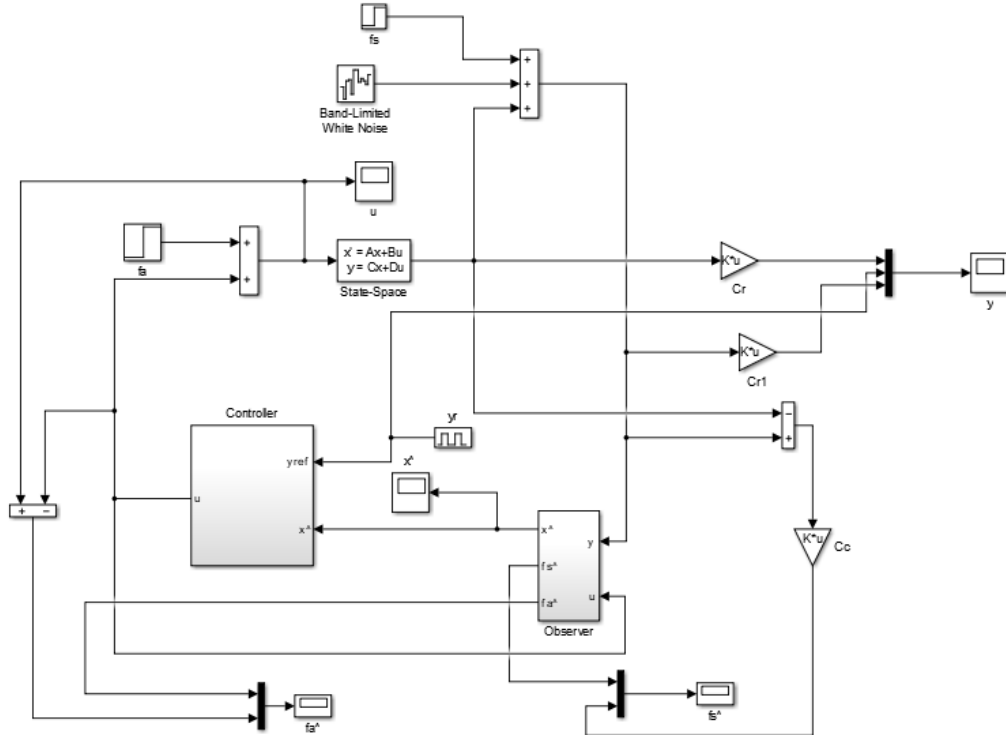
DAFTAR PUSTAKA

- Bavina, R., Jamliya, N., & Vashishtha, K. (2014). Cascade Control Of DC Motor With Advance Controller. *International Journal of Electrical, Electronics and Data Communication*, 131-134.
- Hermawan, T. I. (2018). *Perancangan Sistem Active Fault Tolerant Control (AFTC) untuk Pengendalian Posisi dengan Kontrol Cascade pada Sistem Servo Modular MS150 DC*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Hudaya, C. (2013). *Motor DC*. Jakarta: Universitas Indonesia.
- Indriawati, K. (2016). Robust Fuzzy Observer-based Fault Tolerant Tracking Control for Nonlinear System with Simultaneous Actuator and Sensor Fault: Application to a DC Series Motor Speed Drive. *Praize Worthly Priza*, 375-385.
- Indriawati, K., & Sebe, N. (2020). Fault Tolerant Method on Position Cascade Control of DC Servo System. *Jurnal Teknik ITS*, 144-151.
- Jiang, J., & Yu, X. (2011). Fault-Tolerant Control Systems: A Comparative Study Between Active and Passive Approachs. *Annual Reviews in Control*, 60-72.
- Katherin Indriawati, Trihastuti Agustinah, Achmad Jazidie. (2013). Reconfigurable Fault-Tolerant Control of Linear System with Actuator and Sensor. *IEEE*, 22-27.
- Klimkowski, K., & Dybkowski, M. (2017). A Fault Tolerant Control Structure for an Induction Motor Drive System. *Automatika*, 638–647.
- Mahmoud, M. M., Jiang, J., & Zhang, Y. (2003). *Active Fault Tolerant Control Systems*. Verlag Berlin Heidelberg NewYork: Springer.
- Namazov, M., & Basturk, O. (2010). DC Motor Position Control Using Fuzzy Proportional-Derivative Controllers With Different Defuzzification Methods. *An Official Journal of Turkish Fuzzy Systems Association*, 35-64.

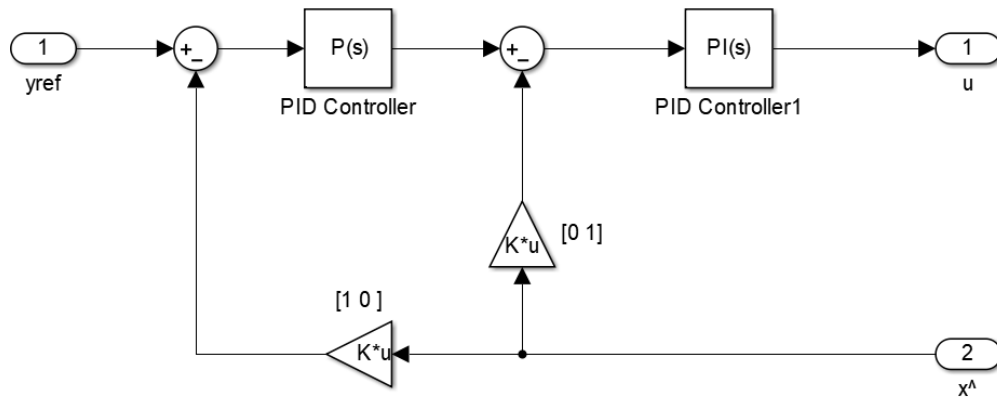
- Noura, H., Theilliol, D., Ponsart, J. C., & Chamseddine, A. (2009). *Fault-tolerant Control Systems: Design and Practical Applications*. London: Springer-Verlag.
- Ogata, K. (1997). *Modern Control Engineering 3rd ed*. London: Prentice Hall International.
- Ogata, K. (2001). *Modern Control Engineering 5th Edition*. Boston: Prentice Hall.
- Pengaturan, A. L. (2015). *Sistem Pengaturan Kecepatan Motor DC*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Pradhana, I. T. (2017). *Perancangan Sistem Active Fault Tolerant Control (Aftc) pada Pengendalian Posisi Sistem Servo Modular MS150 DC dengan Kesalahan pada Aktuator dan Sensor*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Setyaningrum, D. (2012). Desain dan Implementasi Model Reference Adaptive Control untuk Pengaturan Tracking Optimal Posisi Motor DC. *JURNAL TEKNIK POMITS Vol. 1, No. 1*, 1-6.
- Shaker, M., & Patton, R. J. (2012). A Multiple Model Approach to Fault Tolerant Tracking Control for Nonlinear Systems. *20th Mediterranean Conference on Control and Automation (MED)*. Barcelona.
- Sugiono, D. (2015). *Model Matematik Motor DC*. Malang: PPPP TK VEDC.
- Vilanova, R. (2008). PID Tuning for Cascade Control System Design.
- Visioli, A. (2006). *Practical PID Control*. Brescia: Springer.
- Zhang, J., Swain, A. K., & Nguang, S. K. (2016). *Robust Observer Based Fault Diagnosis for Nonlinier Systems Using MATLAB*. Springer.

LAMPIRAN

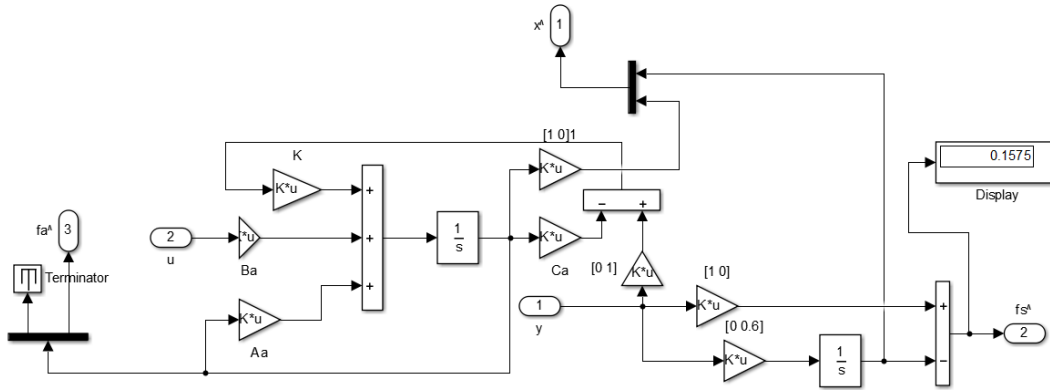
A. Simulasi Penerapan PFTC pada program Simulink MATLAB R2014B



B. Sistem pengendalian Cascade



C. Observer



D. Kode Matlab

```
clear all
clc
Ad= [1 0.06; 0 0.5948];
Bd= [0 ; 0.91];
Cd=[1 0;0 1];
Dd=[0;0];
Cr=[1 0];
Ts= 0.1;
%domain diskrit to kontinyu
A = (Ad - eye(2))/Ts;
B = Bd/Ts;
C = Cd;
D = Dd;
n = length(A);
p = size(C,1);
m = size(B,2);
nr=1;

%perancangan observer
Atild = [-4.052 9.1; 0 0];
Btild = [9.1; 0];
Ctild = [1 0];
R4=eye(1);
Q4=[10 0;0 1];
L4=lqr(Atild',Ctild',Q4,R4);

Lqr=L4';
```

BIODATA PENULIS



Nama penulis Yuli Mauliza. Penulis dilahirkan di Aceh, tanggal 25 Juli 1998 dari ayah adalah Afifuddin dan ibu adalah Nurfadhilah. Saat ini penulis tinggal di Peudawa, Aceh Timur, Provinsi Aceh. Penulis telah menyelesaikan pendidikan tingkat dasar di SDN Asan Ramphak pada tahun 2010, pendidikan menengah pertama di SMPN 1 Peudawa pada tahun 2013, pendidikan menengah atas di SMAN Unggul Aceh Timur pada tahun 2016, dan sedang menempuh pendidikan S1 Teknik Fisika FTIRS di Institut Teknologi Sepuluh Nopember Surabaya hingga sekarang. Pada bulan Juni 2020 penulis telah menyelesaikan Tugas Akhir dengan judul Perancangan *Passive Fault Tolerant Control* (PFTC) pada Pengendalian Posisi Sistem Motor DC. Bagi pembaca yang memiliki saran, kritik, atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini maka dapat menghubungi penulis melalui email: maulizayuli1998@gmail.com.