



TESIS - TE142599

# DESAIN KOMPENSASI KESALAHAN AKTUATOR MENGUNAKAN MRC BERBASIS PID CTC PADA MANIPULATOR 2 DOF

CHURNIA SARI  
22142020201

DOSEN PEMBIMBING  
Prof. Dr. Ir. Achmad Jazidie, M.Eng.  
Dr. Trihastuti Agustinah, ST., MT.

PROGRAM MAGISTER  
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017





TESIS - TE142599

**DESAIN KOMPENSASI KESALAHAN AKTUATOR  
MENGUNAKAN MRC BERBASIS PID CTC PADA  
MANIPULATOR 2 DOF**

CHURNIA SARI  
2214 202 201

DOSEN PEMBIMBING  
Prof. Dr. Ir. Achmad Jazidie, M.Eng.  
Dr. Trihastuti Agustinah, ST., MT.

PROGRAM MAGISTER  
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017



## LEMBAR PENGESAHAN

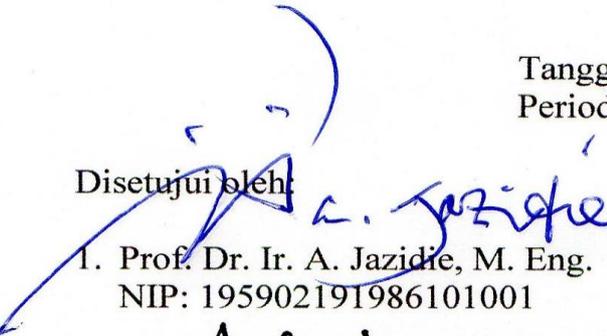
Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Teknik (M.T)  
di  
Institut Teknologi Sepuluh Nopember

oleh:

Churnia Sari  
NRP. 2214202201

Tanggal Ujian : 8 Juni 2017  
Periode Wisuda : September 2017

Disetujui oleh

- 
1. Prof. Dr. Ir. A. Jazidie, M. Eng. (Pembimbing I)  
NIP: 195902191986101001
  2. Dr. Trihastuti Agustinah, ST., MT. (Pembimbing II)  
NIP: 196808121994032001
  3. Prof. Dr. Ir. Mohammad Nuh, DEA (Penguji)  
NIP: 195906171984031002
  4. Ir. Rusdhianto Effendi A.K., MT. (Penguji)  
NIP: 195704241985021001

Dekan Fakultas Teknologi Elektro



Dr. Tri Anief Sardjono, S.T., M.T.  
NIP. 197002121995121001

*Halaman ini sengaja dikosongkan*

## **PERNYATAAN KEASLIAN TESIS**

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **“DESAIN KOMPENSASI KESALAHAN AKTUATOR MENGGUNAKAN MRC BERBASIS PID CTC PADA MANIPULATOR 2 DOF”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Mei 2016



Churnia Sari

NRP. 2214202201

*Halaman ini sengaja dikosongkan*

# DESAIN KOMPENSASI KESALAHAN AKTUATOR MENGUNAKAN MRC BERBASIS PID CTC PADA MANIPULATOR 2 DOF

Nama mahasiswa : Churnia Sari  
NRP : 2214202201  
Pembimbing : 1. Prof. Dr. Ir. A. Jazidie, M. Eng  
2. Dr. Trihastuti Agustinah, ST., MT.

## ABSTRAK

Penurunan performa kerja pada suatu sistem kontrol dapat diakibatkan oleh beberapa hal. Salah satu penyebabnya adalah terjadinya *fault* pada aktuator. Kesalahan aktuator pada Robot Manipulator 2 DOF (Degree of Freedom) akan membuat sinyal respon tidak mampu mengikuti referensinya, sehingga mengakibatkan terjadinya deviasi sinyal dan memperbesar *overshoot*. Permasalahan tersebut dapat diselesaikan dengan menggunakan metode Estimasi dan Kompensasi. Metode ini dapat dilakukan dengan memanfaatkan sinyal residual yang merupakan perbandingan hasil sinyal respon sistem yang sebenarnya dengan hasil sinyal respon dari *Model Reference Control* (MRC).

Dalam penelitian ini dirancang sebuah algoritma *Active Fault Tolerant Control* (AFTC), dimana kesalahan aktuator yang terjadi akan diestimasi dan dikompensasi dengan menggunakan MRC yang dirancang berdasarkan Hukum *Computed Torque Control* (CTC) dan Kontroler *Proportional Integral Derivative* (PID).

Dari hasil simulasi, terbukti bahwa sistem yang dirancang dapat melakukan kompensasi kesalahan aktuator yang bersifat *additive* maupun *multiplicative*. *Error tracking* dapat diperkecil dari 0.0356 rad menjadi 0.0077 rad, serta dapat memberikan respon yang cepat dengan nilai *rise time* kurang dari 1s. Rancangan sistem juga mampu memperkecil *overshoot* dari 0.11 menjadi 0.047 untuk Join 1 dan 0.041 untuk Join 2.

Kata kunci: Kesalahan Aktuator, Kompensasi, MRC, AFTC

*Halaman ini sengaja dikosongkan*

# DESIGN OF ACTUATOR FAULT COMPENSATION WITH MRC IN 2 DOF MANIPULATOR BASED ON PID CTC

By : Churnia Sari  
Student Identity Number : 2214202201  
Supervisor(s) : 1. Prof. Dr. Ir. A. Jazidie, M. Eng  
2. Dr. Trihastuti Agustinah, ST., MT.

## ABSTRACT

Performance degradation in a control system can be caused by several things. One of the causes of them is the occurrence of actuator fault. Actuator fault on Robot Manipulator 2 DOF (Degree of Freedom) makes response signal cannot track its reference, then it causes signal deviation and enlarge overshoot. These problems can be solved using estimation and compensation method can be a solution to this problem. These methods can be done by residual signal which is a comparison of the response signal between the actual system and Model Reference Control (MRC).

This paper explains designed of algorithm Active Fault Tolerant Control (AFTC) based MRC. When actuator fault happened, it was estimated and compensated with MRC, that could be designed base on Computed Torque Control (CTC) law and Proportional *Integral* Derivative (PID) Controller.

The simulation result shows that system has been designed can compensate actuator additive and multiplicative errors. It can minimize tracking error from 0.0356 rad to 0.0077 rad and outcome time response less than 1 s. The designed also reduce overshoot from 0.11 to 0.0047 for Join 1 and 0.041 for Join 2.

Key words: Actuator Fault, Compensation, MRC, AFTC

*Halaman ini sengaja dikosongkan*

## **KATA PENGANTAR**

Alhamdulillah, pada akhirnya tesis ini telah selesai dibuat. Pada dasarnya, tesis ini tidak terlalu rumit untuk dikerjakan dan akan menjadi jauh lebih mudah karena dikerjakan dengan ikhlas. Dengan diselesaikannya tesis ini, maka salah satu syarat kelulusan Program Magister Bidang Keahlian Teknik Sistem Pengaturan Fakultas Teknik Elektro telah terpenuhi, semoga syarat-syarat yang lainnya juga dapat dengan mudah diselesaikan.

Penulis juga mengucapkan banyak terimakasih terhadap para pembimbing yang selalu mencerahkan dan dengan penuh keikhlasan senantiasa mengarahkan penulis untuk dapat segera menyelesaikan tesis ini. Ucapan terimakasih juga penulis berikan secara khusus kepada suami, putri Ayasofia yang selalu menemani dan juga orang tua yang tak pernah berhenti memberi semangat, serta kepada satu pembimbing muda dari jurusan tetangga yang cukup bijak memberikan arahan, juga tak ketinggalan teman-teman Pasca Sarjana SP yang tak bosan menjawab segala pertanyaan. Untuk semua pihak yang telah membantu, semoga senantiasa diberikan kelancaran bagi setiap urusanya.

Penulis juga berharap, semoga tesis ini dapat bermanfaat dan dapat memperkaya kajian terkait bidang penelitian AFTC yang saat ini sedang dan telah banyak berkembang.

Surabaya, 4 Mei 2017

Penulis

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	iii
PERNYATAAN KEASLIAN TESIS .....	v
ABSTRAK .....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan .....	4
1.4 Batasan Masalah .....	4
1.5 Kontribusi .....	4
1.6 Metodologi Penelitian .....	4
BAB 2 KAJIAN PUSTAKA .....	7
2.1 Kajian Penelitian Terkait .....	7
2.1.1 <i>Robot Manipulator Control Theory and Practice</i> [2] .....	7
2.1.2 <i>Integrated Fault Estimation and Fault Tolerant Control: A Joint Design</i> [4] .....	10
2.1.3 <i>Fault Estimation and Model Reference Control (MRC) - Based Active Fault Tolerant Control</i> [5] .....	13
2.2 Teori Dasar .....	16
2.2.1 Pemodelan Dinamika Robot Manipulator 2 DOF .....	16
2.2.2 Kontrol CTC .....	19
2.2.3 Kontrol PD <i>Outer loop</i> .....	21
2.2.4 Pendefinisian Kesalahan Pada Robot Manipulator .....	21
2.2.5 Strategi FTC .....	22
BAB 3 METODOLOGI PENELITIAN .....	25
3.1 Rancangan Umum Sistem .....	25
3.2 Diagram Alir Kerja Sistem .....	26

3.3	Pendefinisian Kesalahan.....	26
3.4	Perancangan MRC.....	28
3.5	Perancangan Estimator dan Kompensator.....	30
BAB 4 HASIL DAN PEMBAHASAN .....		33
4.1	Pengujian Kontrol Nominal.....	33
4.1.1	Pengujian Kontrol Nominal Tanpa Terjadi <i>Fault</i> .....	33
4.1.2	Pengujian Kontrol Nominal Ketika Terjadi <i>Fault</i> .....	34
4.2	Pengujian MRC dan Residual .....	37
4.2.1	Pengujian MRC .....	37
4.2.2	Pengujian Residual .....	38
4.3	Pengujian Kompensasi Kesalahan Aktuator .....	39
4.3.1	Kompensasi Kesalahan <i>Additive</i> .....	39
4.3.2	Kompensasi Kesalahan <i>Multiplicative</i> .....	43
4.4	Hasil Estimasi Kesalahan Aktuator.....	45
4.4.1	Estimasi Kesalahan <i>Additive</i> .....	45
4.4.2	Estimasi Kesalahan Aktuator dengan <i>Noise</i> pada Estimator.....	46
4.4.3	Estimasi Kesalahan <i>Multiplicative</i> .....	48
4.5	Pengujian RMSE ( <i>Root Mean Square Error</i> ) .....	48
BAB 5 KESIMPULAN .....		51
DAFTAR PUSTAKA.....		53
LAMPIRAN .....		55

## DAFTAR GAMBAR

Gambar 2.1 Blok Diagram Kontroler PD CTC.....	8
Gambar 2.2 Tracking Posisi Robot Manipulator dengan Kontroler PD CTC .....	9
Gambar 2.3 Tracking Posisi Robot Manipulator dengan Kontroler PD CTC .....	9
Gambar 2.4 Tracking Posisi Robot Manipulator dengan Kontroler PID pada <i>Outer loop</i> .....	10
Gambar 2.5 Tracking Posisi Robot Manipulator dengan Kontroler PID pada <i>Outer loop (fault case)</i> .....	10
Gambar 2.6 Skema dari <i>Join Design</i> [4] .....	12
Gambar 2.7 <i>Tracking</i> Posisi dan Estimasi Kesalahan Aktuator .....	12
Gambar 2.8 <i>Tracking</i> Posisi dan Estimasi Kesalahan dengan <i>Noise</i> pada <i>Estimator</i> .....	13
Gambar 2.9 AFTC berbasis <i>Fault estimator</i> dan MRC [5] .....	14
Gambar 2.10 Fault pada Link 1,2 dan estimasinya .....	15
Gambar 2.11 <i>Tracking</i> Posisi Pada Link 1,2 .....	16
Gambar 2.12 Robot Manipulator (Planar 2 DOF) [6].....	17
Gambar 2.13 Kontrol CTC [2] .....	20
Gambar 2.14 Struktur AFTC dan PFTC [10].....	23
Gambar 2.15 Bagian-Bagian dari AFTC .....	23
Gambar 2.16 Klasifikasi Teknik FDI [14] .....	24
Gambar 2.17 Blok Diagram Teknik FDI berdasarkan Estimasi Kesalahan [14]..	24
Gambar 3.1 Blok Diagram Rancangan AFTC berbasis MRC .....	25
Gambar 3.2 <i>Flowchart</i> AFTC berbasis MRC .....	26
Gambar 3.3 MRC berbasis PID CTC.....	29
Gambar 4.1 <i>Tracking</i> Posisi Robot ( <i>fault free case</i> ).....	34
Gambar 4.2 <i>Error tracking</i> Robot ( <i>fault free case</i> ) .....	34
Gambar 4.3 <i>Tracking</i> Posisi dengan <i>Additive Fault</i> .....	35
Gambar 4.4 <i>Tracking</i> Posisi dengan beberapa Jenis <i>Fault</i> .....	35
Gambar 4.5 <i>Tracking</i> Posisi dengan <i>Multiplicative Fault</i> .....	36
Gambar 4.6 Perbandingan <i>error tracking</i> antara sistem dengan <i>Fault Additive</i> (a) dan Sistem dengan <i>Fault Multiplicative</i> (b).....	37
Gambar 4.7 <i>Tracking</i> Posisi dari MRC.....	38
Gambar 4.8 Sinyal Residual, Turunan Pertama, Turunan Kedua dan <i>Integral</i> dari Sinyal Residual .....	39
Gambar 4.9 <i>Tracking</i> Posisi Robot untuk <i>Fault</i> ke- 1 .....	40
Gambar 4.10 <i>Tracking</i> Posisi Robot untuk <i>Fault</i> ke-2 .....	40
Gambar 4.11 <i>Tracking</i> Posisi Robot untuk <i>Fault</i> ke-2 .....	41
Gambar 4.12 <i>Tracking</i> Posisi Robot ( $u=PID$ ).....	41
Gambar 4.13 <i>Tracking</i> Posisi dengan <i>Fault</i> Aktuator dan <i>Disturbance</i> .....	42
Gambar 4.14 <i>Tracking</i> Posisi dengan <i>Fault</i> Aktuator dan <i>Noise</i> pada <i>Output</i> .....	43

Gambar 4.15 <i>Tracking</i> Posisi dengan <i>Fault</i> Multiplikatif (a) 30% <i>Actuator Fault</i> (b) 50% <i>Actuator Fault</i> (c) 70% <i>Actuator Fault</i> .....	44
Gambar 4.16 Estimasi Kesalahan Aktuator pada Lengan 1 .....	46
Gambar 4.17 Estimasi Kesalahan Aktuator pada Lengan 2 .....	46
Gambar 4.18 Estimasi Kesalahan Aktuator dengan <i>Noise</i> pada <i>Estimator</i> (L2)...	47
Gambar 4.19 Estimasi Kesalahan Aktuator dengan <i>Noise</i> pada <i>Estimator</i> (L1)...	47

## DAFTAR TABEL

Tabel 4-1 Amplitudo <i>overshoot</i> untuk Beberapa Kesalahan Multiplikatif.....	44
Tabel 4-2 Hasil Estimasi <i>Fault</i> Aktuator (Multiplikatif).....	48
Tabel 4-3 Tabel RSME Hasil Pengujian.....	49

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Robotika adalah bidang yang menantang di bidang rekayasa maupun sains. Robotika telah memberikan kontribusi yang penting pada bidang industri, di mana telah banyak industri yang menggunakan robot untuk tugas-tugas seperti perakitan (*assembling*), pengelasan, pengecatan, pemindahan material, dan lain-lain. Pada penelitian ini, bidang robotika yang di khususkan untuk kepentingan penelitian adalah manipulator robot dengan 2 derajat kebebasan. Telah diketahui bahwa, robot-robot seperti *biped robot*, *walking robot*, *mobile manipulator* atau dari keluarga robot lain yang terinspirasi dari makhluk hidup adalah menggunakan kajian robot planar 2 DOF sebagai pembangun sistem yang paling kecil. Pergerakan antar dua lengan dengan dua sendi menjadi dasar bagi struktur persendian robot-robot ini. Oleh karena itu, penelitian dalam bidang manipulator dua sendi ini menjadi sangat penting sebagai langkah untuk melakukan penelitian ke kelas robot yang lebih kompleks. Meskipun terlihat sangat sederhana dengan dua sendi, namun jika dikaji secara saintifik akan terasa bahwa robot planar 2 DOF ini sudah sangat kompleks [1].

Salah satu kegunaan robot dalam bidang industri adalah menggantikan fungsi lengan manusia. Hal yang diharapkan, dengan adanya substitusi dari tenaga manusia menjadi tenaga mesin adalah penyelesaian tugas menjadi lebih efektif dan efisien, karena bisa meminimalisir adanya *human error* dan dapat dikerjakan dalam waktu yang tidak terbatas, sehingga baik secara kualitas dan kuantitas semua hasil pekerjaan dapat ditingkatkan. Oleh karena itu, performa kerja dari robot harus selalu dalam kondisi bagus. Penelitian yang telah dilakukan oleh [2] terbukti bahwa robot manipulator 2 DOF dapat dikontrol dengan baik menggunakan kontroler PD CTC ataupun PID CTC, namun ternyata penelitian ini tidak mempertimbangkan adanya kemungkinan terjadinya *fault* pada aktuator robot.

Proses monitoring sangat diperlukan untuk meminimalisir efek dari terjadinya *fault* tersebut. Proses monitoring ini, diperlukan agar kesalahan dapat segera diketahui dan ditentukan langkah perbaikannya. Proses monitoring pertama

kali banyak diteliti pada tahun 1990an yang dikenal dengan *Fault Detection and Diagnostic*. Berbagai macam teknik dikembangkan sampai saat ini untuk mendapatkan Teknik FDD yang *robust*. Dalam perkembangan Teknik FDD, sebagian peneliti fokus pada *Fault Estimation* (FE) termasuk pada penelitian ini. Teknik FE pada penelitian ini, akan dibangkitkan menggunakan teknik desain MRC (*Model Reference Control*). Jika kesalahan yang terjadi dapat diketahui besar, dan waktunya, maka hasil dari monitoring tersebut dapat digunakan untuk keperluan kompensasi kesalahan aktuator pada robot manipulator 2 DoF.

Pada umumnya FE dan *Fault Compensator* bekerja secara terpisah, mula mula FE bekerja terlebih dahulu, kemudian kompensator akan bekerja setelah FE memberikan hasil estimasi kesalahannya. Oleh karena itu, perlu metode khusus untuk melakukan integrasi agar FTC dapat dibentuk [3]. Namun, dari penelitian [4] dijelaskan bahwa FTC dapat dibentuk dengan cara yang berbeda dengan menggunakan *Joint Design* dimana FE dan kompensator bekerja secara simultan, sehingga tidak akan ditemukan kerancuan untuk menentukan bagaimana dinamika FE dan aktif FTC dapat terbentuk menjadi satu. Selain itu dalam penelitian ini, juga dapat menghindari kemungkinan interaksi dari keduanya, atau jika kompensator ternyata tidak dapat bekerja secara sempurna. Dalam penelitian ini, juga dapat dihindari kekhawatiran adanya satu atau beberapa subsistem yang akan bekerja lebih lambat, karena semua bekerja secara simultan. Permasalahan tersebut yang mendasari penelitian [4] dirancang dan juga mendasari penelitian [4] memiliki kontribusi dalam bidang FTC, yaitu dirancangnya sebuah metode yang berbeda dari FTC pada umumnya dan disebut dengan *Integrated Fault Tolerant Control*. Sayangnya, penelitian ini belum diterapkan dalam plan apapun, sehingga matriks *state space* dan vektor *fault*-nya disesuaikan dengan syarat yang dibutuhkan oleh *estimator*. Pada penelitian ini kesalahan yang diuji juga sangat terbatas, meskipun peneliti juga meyakinkan bahwa sistem ini akan bertahan dari gangguan *error* estimasi. Gangguan *error* estimasi dapat dimodelkan dengan cara memberikan sinyal dengan distribusi normal berupa model Gaussian, yang terjadi bersamaan dengan terjadinya *fault*. Gangguan tersebut juga berasal dari sumber yang sama dengan sumber terjadinya *fault*. Hal tersebut dirancang untuk membuktikan bahwa FE dapat bekerja dengan kesalahan yang bersifat stokastik.

Selanjutnya, pada penelitian [5] dibangun sebuah algoritma AFTC (*Active Fault Tolerant Control*) berbasis *Model Reference Control* yang secara simulasi telah diterapkan pada plan Robot Manipulator 2 DOF. Dengan teknik ini, kesalahan yang terjadi akan diperbaiki, sehingga performa robot tetap bisa dijaga stabil dan optimal. Kesalahan yang diuji-pun diperlebar oleh peneliti tidak hanya kesalahan konstan tapi juga terdapat kesalahan yang tidak konstan. Kesalahan akan di deteksi secara *online*, kemudian diestimasi dan dikompensasi sekaligus. Sehingga efek dari terjadinya *fault actuator* dapat diminimalisir dan sistem masih mampu bekerja secara optimal. Pada [5] dijelaskan bahwa kontribusi dari penelitian ini adalah desain dari MRC yang digunakan dalam algoritma AFTC. MRC pada penelitian ini hanya bergantung dari nilai referensi saja. Perihal tersebut juga dibuktikan secara matematis dalam pembentukan *state*-nya. Kekurangan dari sistem ini adalah respon sistem menjadi cukup lambat dan tidak mampu memproses *tracking* referensi posisi dan kecepatan selain sinyal konstan. Jika pemodelan MRC dapat disubstitusi dengan persamaan model MRC yang lain, ada kemungkinan sistem bekerja lebih cepat, karena *estimator* pada sistem ini sangat bergantung terhadap MRCnya.

Model MRC lain untuk dinamika manipulator 2 DOF mungkin dapat kita peroleh dari penelitian [6]. Pada penelitian ini deteksi dan estimasi kesalahan yang dilakukan oleh sebuah *PD Descriptor Observer* bekerja sangat cepat. Jika dengan kontroler yang sama, PID CTC dapat digunakan untuk pembentukan MRC, maka diharapkan FTC berbasis MRC dapat bekerja lebih cepat dan lebih baik dari penelitian [5].

## 1.2 Rumusan Masalah

Permasalahan yang akan dibahas pada penelitian ini adalah, bagaimana merancang dan menerapkan suatu algoritma AFTC yang dapat meminimalisir efek dari kesalahan (berupa *overshoot* ataupun deviasi sinyal) dengan *rise time* yang cepat. AFTC akan dibangun berdasarkan MRC untuk mengestimasi sekaligus mengkompensasi kesalahan aktuator.

### 1.3 Tujuan

Tujuan dari penelitian ini adalah merancang *fault estimator* dan *compensator* pada sistem dengan kondisi *fault*, memperbaiki performa *closed loop* sistem agar dapat meminimalisir efek dari *fault* aktuator, serta menghasilkan kontrol *tracking* yang mampu mengikuti trayektori referensi dengan *error* minimal.

### 1.4 Batasan Masalah

Kesalahan pada penelitian ini terjadi pada aktuator. Kesalahan yang terjadi tidak ditentukan secara spesifik sifat dan juga pemodelanya.

### 1.5 Kontribusi

Penelitian ini diharapkan memberikan kontribusi berupa: menghasilkan rancangan “*Active Fault Tolerant Control*” pada Robot manipulator 2 DOF menggunakan *Model Reference Control* yang dibentuk berdasarkan PID CTC, serta menghasilkan rancangan kompensator yang dapat menangani kesalahan yang bersifat *additive* maupun *multiplicative* berdasarkan rancangan tersebut.

### 1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini antara lain :

#### 1. Studi Literatur

Studi literatur dilakukan dengan cara mengumpulkan dan mempelajari penelitian – penelitian yang relevan terhadap topik yang akan diteliti. Melalui pembelajaran dan perbandingan beberapa kajian penelitian, diperoleh topik yang terfokus untuk diteliti. Selanjutnya, studi literatur dilakukan dengan mempelajari pembahasan yang sesuai, diantaranya dinamika robot manipulator 2 DOF, kontroler PD, Hukum CTC, MRC dan mekanisme AFTC.

#### 2. Pemodelan Sistem

Pada tahap ini akan dilakukan pemodelan fisik robot manipulator 2 DOF agar dapat disimulasikan dan digunakan dalam rencana kerja sistem yang akan dirancang. Pemodelan yang akan digunakan pada penelitian ini tidak

ditentukan sendiri oleh penulis, melainkan diambil dari referensi yang sudah ada sebelumnya.

### 3. Perancangan Sistem

Berdasarkan hasil pemodelan *plant*, didesain sistem dengan kontroler nominal, kemudian didesain MRC yang dilengkapi dengan Desain *estimator* dan kompensator

### 4. Pengujian dan Analisis Sistem

Pada tahap pengujian dan analisis, kompensasi kesalahan aktuator akan diuji keandalannya dalam beberapa pengujian yang akan dijelaskan pada Bab IV. Dari pengujian ini akan diketahui bahwa hasil rancangan kompensator dapat bekerja dengan baik atau tidak dalam memperbaiki performa kerja Robot Manipulator yang mengalami penurunan akibat dari terjadinya *fault* pada aktuatornya.

### 5. Kesimpulan

Kesimpulan diperoleh sesuai dengan hasil pengujian dan analisis yang dilakukan.

### 6. Penulisan Laporan Tesis

Penulisan laporan tesis dilakukan sebagai dokumentasi dari hasil penelitian yang dilakukan.

*Halaman ini sengaja dikosongkan*

## **BAB 2**

### **KAJIAN PUSTAKA**

Pada penelitian ini, bidang robotika yang di khususkan untuk kepentingan penelitian ini adalah manipulator robot dengan 2 derajat kebebasan. Tidak bisa dipungkiri bahwa seluruh sistem mekanik ataupun elektronik dapat mengalami penurunan performa dalam menjalankan tugasnya. Penurunan performa ini, dapat diakibatkan oleh berbagai hal, seperti terjadinya *fault* pada sistem, adanya gangguan ataupun juga karena banyaknya *noise* pada sistem tersebut. Agar bisa menjalankan tugasnya dengan baik, sudah banyak dilakukan penelitian dalam bidang manipulator. Beberapa penelitian yang dilakukan akan dibahas dalam subbab di bawah ini.

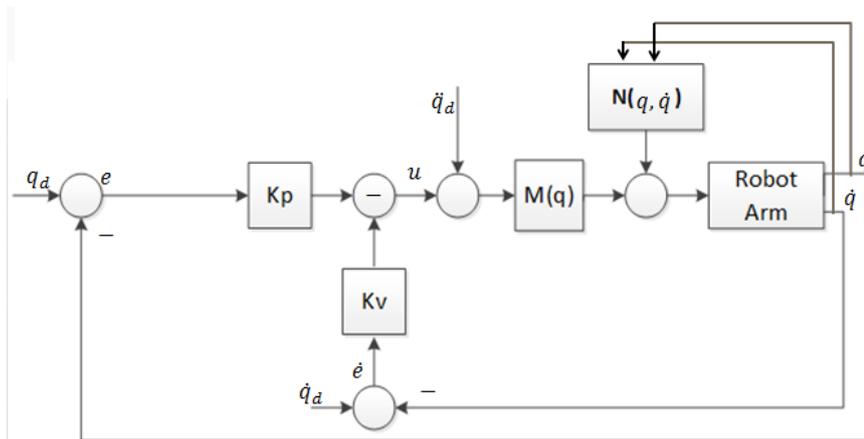
#### **2.1 Kajian Penelitian Terkait**

Pada bagian ini akan dijelaskan beberapa kajian yang digunakan sebagai ide awal, penunjang maupun acuan dari penelitian ini.

##### **2.1.1 *Robot Manipulator Control Theory and Practice* [2]**

Penelitian tentang manipulator 2 DoF dilakukan oleh Frank Lewis pada Bab IV buku ini, yaitu tentang pengaturan *tracking* dinamika robot dengan pendekatan PFTC (*Pasif Fault Tolerant Control*) menggunakan kontroler PD CTC dan PID CTC. CTC merupakan sebuah kontroler yang menerapkan linearisasi pada *feedback* kontrolnya.

Pada awalnya Lewis menerapkan kontroler PD CTC untuk melakukan *tracking* pada Robot Manipulator 2 DOF. Gambar 2.1 menjelaskan blok diagram yang digunakan Lewis untuk mengatur *Tracking Robot Manipulator 2 DOF*. Pada Gambar 2.1 untuk  $q_d$ ,  $\dot{q}_d$ ,  $\ddot{q}_d$  masing-masing adalah referensi posisi, kecepatan dan percepatan. Sedangkan  $q$  dan  $\dot{q}$  adalah posisi dan kecepatan sebenarnya.



Gambar 2.1 Blok Diagram Kontroler PD CTC

Jika diketahui dinamika manipulator robot adalah sebagai berikut:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (2.1)$$

$$\text{dan } N(q, \dot{q}) = V(q, \dot{q}) + G(q) \quad (2.2)$$

maka dinamika manipulator dapat ditulis sebagai berikut:

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (2.3)$$

Pada Gambar 2.1 dapat dijelaskan fungsi *input* kontrol yang digunakan adalah:

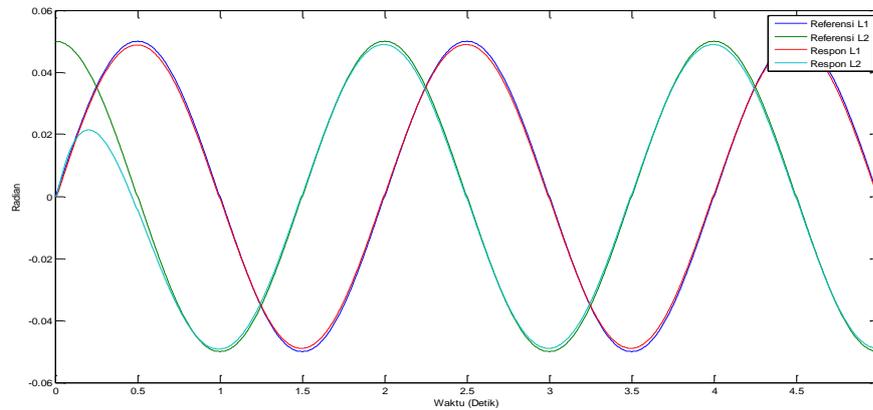
$$u = -K_v \dot{e} - K_p e \quad (2.4)$$

dengan torsi linearisasi umpan balik ditambah dengan kontroler PD adalah:

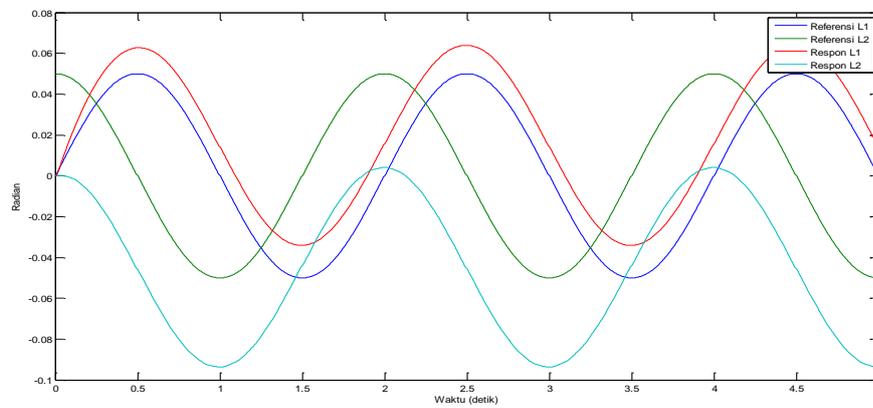
$$\tau = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \quad (2.5)$$

Hasil dari penerapan kontroler PD CTC, respon posisi manipulator mampu mengikuti sinyal referensi dengan sangat baik seperti pada Gambar 2.2. Namun ternyata kontroler ini tidak mempertimbangkan terjadinya *fault*. Sementara pada kenyataannya dalam hampir semua mesin mekanik maupun elektronik, *fault* sangat memungkinkan terjadi. Sehingga ketika terjadi *fault*, respon posisi akan bergeser cukup jauh dari referensinya, seperti yang terlihat pada Gambar 2.3.

Pertimbangan dalam pengkajian penanganan *fault* tersebut, mendasari penulis untuk membentuk sistem yang lebih robust, yaitu menambahkan *Gain Integral*. Namun, tetap menggunakan pendekatan PFTC. Penambahan *Gain Integral* pada kontroler diharapkan membuat sistem lebih *robust* terhadap kesalahan yang mungkin terjadi.

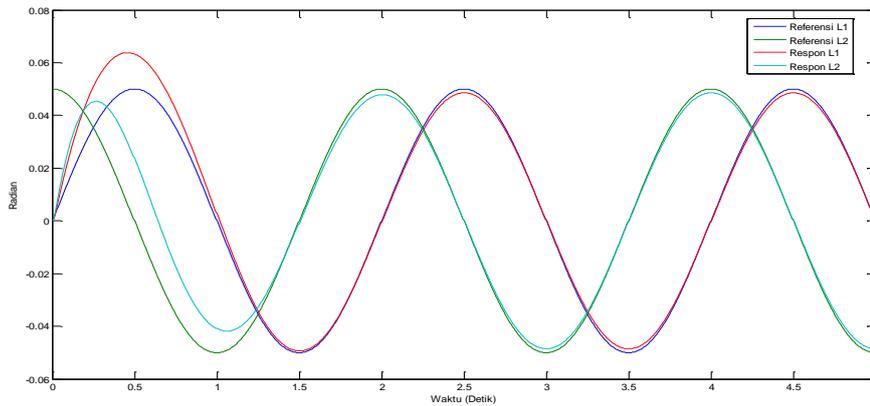


Gambar 2.2 *Tracking* Posisi Robot Manipulator dengan Kontroler PD CTC

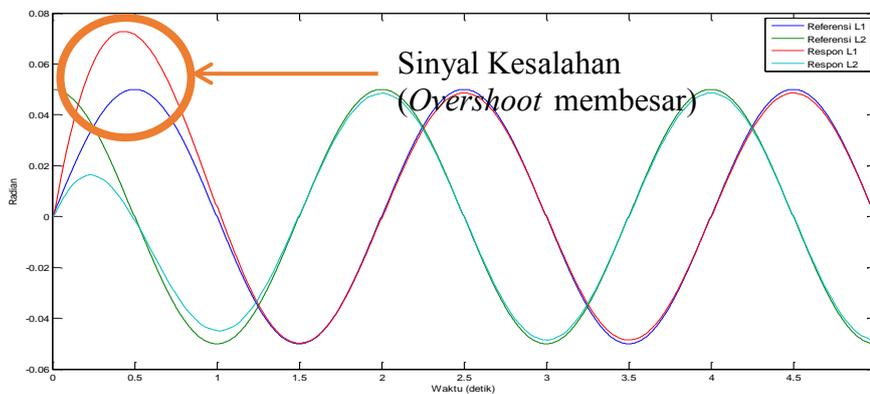


Gambar 2.3 *Tracking* Posisi Robot Manipulator dengan Kontroler PD CTC  
(*fault case*)

Hasil dari penerapan kontroler ini sistem lebih *robust* terhadap sinyal kesalahan. Meskipun sistem mampu melakukan *tracking* terhadap referensinya, namun akibat dari penambahan *Gain Integral*, muncul sinyal *overshoot*, dan sinyal *overshoot* ini akan semakin membesar ketika terjadi *fault* seperti pada Gambar 2.4 dan 2.5. Kedua metode ini cukup menguntungkan untuk dipakai dalam melakukan kontrol pada Robot Manipulator. Namun disamping kelebihan tersebut, masing-masing punya kekurangan seperti yang telah dijelaskan.



Gambar 2.4 *Tracking* Posisi Robot Manipulator dengan Kontroler PID pada *Outer loop*



Gambar 2.5 *Tracking* Posisi Robot Manipulator dengan Kontroler PID pada *Outer loop (fault case)*

### 2.1.2 *Integrated Fault Estimation and Fault Tolerant Control: A Joint Design* [4]

Pada penelitian ini, permasalahan yang muncul pada penjabaran sebelumnya akan coba diselesaikan, yaitu untuk menyelesaikan permasalahan hadirnya kesalahan aktuator pada suatu sistem. Namun pada dasarnya meski mampu menjawab permasalahan yang hadir, penelitian ini tidak diterapkan pada sebuah plan tertentu sehingga belum dibuktikan secara simulasi apakah dapat berjalan sesuai harapan jika digunakan untuk plan tertentu. Yang menarik dari penelitian ini adalah, menghasilkan sebuah rancangan *Integrated Fault Tolerant Control*. Maksud dari *Integrated Fault Tolerant Control* adalah FE dan kompensator bekerja secara simultan, sehingga waktu yang dibutuhkan lebih cepat

dan bekerja secara selaras (sekaligus). Hal ini dibuat untuk menghindari adanya interaksi yang saling melemahkan antar keduanya, dan juga menghindari adanya kompensator yang tidak bekerja secara sempurna. Tidak seperti FTC pada umumnya dimana FE akan bekerja lebih dulu, baru setelahnya kompensasi bisa dilakukan, dalam penelitian ini kompensator dan *estimator* bekerja sekaligus. Untuk bisa merancang FTC seperti dalam penelitian ini, dibutuhkan sebuah syarat khusus dimana *rank* matriks  $E$  dan  $B$  pada Persamaan 2.6 harus *uncollinear* ( $\text{rank} [B \ E] \neq \text{rank} B$ ), meski penulis juga memberikan keterangan jika ternyata ditemui sebuah kasus dua matriks tersebut *collinear* maka bukan tidak mungkin *Integrated Fault Tolerant* bisa diterapkan, kemungkinan besar masih sangat bisa, tapi akan lebih baik hasilnya jika digunakan algoritma lain yaitu *Sliding Mode Observer*.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + Ef(t) \\ y(t) &= Cx(t)\end{aligned}\tag{2.6}$$

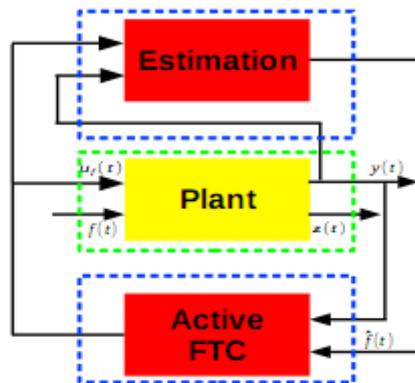
Dalam kondisi ketika terjadi kesalahan, sistem akan dirubah kedalam bentuk *Augmented* dan digunakan dalam Persamaan Luenberger *Observer* demi keperluan estimasi. Persamaan yang dirancang untuk keperluan tersebut adalah seperti berikut:

$$\begin{aligned}\dot{\eta}(t) &= \hat{A}\hat{\eta}(t) + \hat{B}u(t) + L(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= \hat{C}\hat{\eta}\end{aligned}\tag{2.7}$$

dimana  $\hat{\eta}$  dan  $\hat{y}$  merupakan *state* dan *ouput observer*. *State* ( $\eta$ ) pada *observer* ini merupakan bentuk *augmented*, yaitu  $\eta(t) = \begin{bmatrix} x(t) \\ f(t) \end{bmatrix}$ . Kemudian, untuk bisa merancang *Integrated Fault Tolerant Control*, maka harus dipenuhi kaidah berikut ini:  $\text{rank} [B \ E] \neq \text{rank} B$ , dimana untuk memenuhi kaidah tersebut kontroler nominal harus masuk kedalam rancangan aktif FTC (kompensator). Persamaan 2.8 merupakan bentuk perluasan dari Persamaan 2.7, dimana dalam Persamaan 2.8 terdapat performasi evaluasi sinyal, sekaligus terdapat *feedback* kontroler pada *feedback state*-nya. Sehingga persamaan aktif FTC yang dibentuk secara keseluruhan adalah:

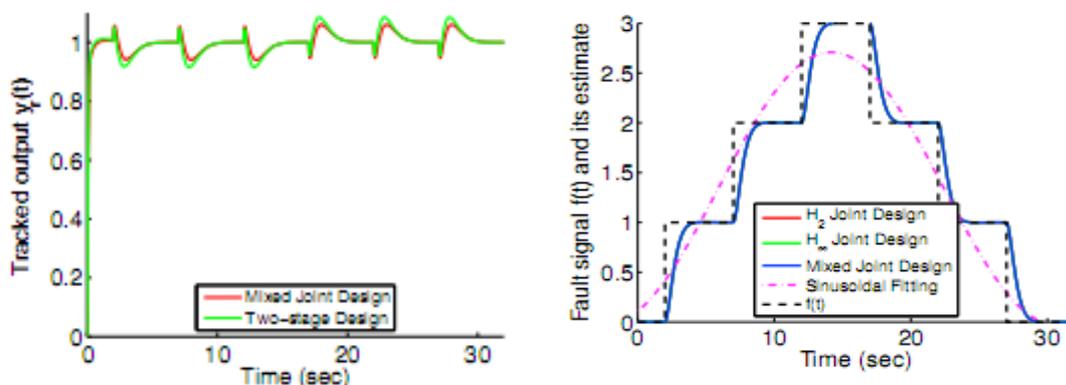
$$\begin{aligned}\dot{\zeta}(t) &= (\bar{A} + \bar{B}K)\zeta(t) + \bar{E}r(t) \\ z(t) &= \bar{C}\zeta(t)\end{aligned}\tag{2.8}$$

dimana  $\zeta = \begin{bmatrix} \eta \\ e_I \end{bmatrix}$ , jika  $e_I$  adalah performansi evaluasi sinyal, maka rancangan *Fault Tolerant* tersebut mampu memperhitungkan potensi terjadinya kesalahan dan melakukan pelacakan secara langsung. Sehingga kesalahan yang terjadi dapat diketahui kemudian dilakukan kompensasi.



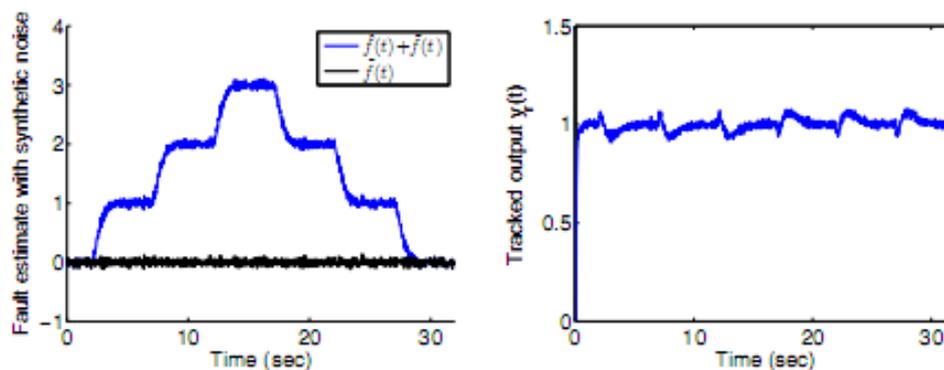
Gambar 2.6 Skema dari *Joint Design* [4]

Gambar 2.6 menunjukkan bahwa struktur dari *Joint Design* ini sebenarnya tidak jauh berbeda dengan desain AFTC lain, hanya saja ada modifikasi pada kompensatornya agar dapat bekerja secara bersamaan dengan *estimator*-nya. Diharapkan sistem dapat bekerja lebih cepat dan tidak saling mempengaruhi antara *estimator* dengan kompensatornya. Dalam penelitian ini performa *tracking* yang dihasilkan (sinyal yang berwarna merah), serta estimasi dari sinyal kesalahan (sinyal yang berwarna biru) dapat dilihat pada Gambar 2.7. Sistem ini memerlukan waktu 2 detik untuk dapat melakukan estimasi



Gambar 2.7 *Tracking* Posisi dan Estimasi Kesalahan Aktuator

Kelemahan dari sistem ini adalah, *fault* yang diteliti sangat terbatas, yaitu untuk jenis *fault* yang memiliki *long time varying*. meskipun peneliti juga meyakini bahwa sistem ini akan bertahan dari gangguan *error* estimasi dengan memberikan sinyal random dengan distribusi normal ( $\mu = 0$  dan  $\sigma = 0.04$ ) yang terjadi bersamaan dengan terjadinya *fault*. Pada penelitian ini estimator dan kompensator hanya dapat bekerja pada *fault* yang memiliki *long time varying*. Gangguan tersebut juga berasal dari sumber yang sama dengan sumber terjadinya *fault*. Hal tersebut dilakukan penulis untuk membuktikan bahwa sistem dapat bekerja dengan sinyal kesalahan yang bersifat stokastik. Estimasi kesalahan stokastik yang terjadi dan *tracking* dari sistem akibat adanya kesalahan tersebut dapat dilihat pada Gambar 2.8.

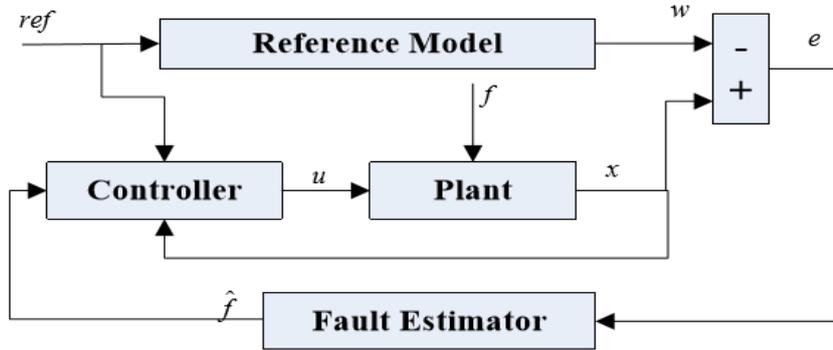


Gambar 2.8 *Tracking* Posisi dan Estimasi Kesalahan dengan *Noise* pada *Estimator*

### 2.1.3 *Fault Estimation and Model Reference Control (MRC) - Based Active Fault Tolerant Control* [5]

Pertimbangan dalam penanganan *fault* aktuator, juga mendasari Chenglei dalam melakukan sebuah penelitian. Selain mampu menangani lebih dari satu jenis *fault*, Chenglei juga mampu menangani adanya *overshoot* pada *tracking* robot. Penelitian ini juga telah disimulasikan dalam plan Robot Manipulator 2 DoF. Namun disini, Chenglei melakukan dari pendekatan yang berbeda yaitu dengan metode AFTC (*Active Fault Tolerant Control*). Dalam penelitian ini, Chenglei mengestimasi kesalahan yang terjadi secara online berdasarkan *Model Reference Control*, kemudian mengkompensasi kesalahan tersebut dengan hasil estimasinya yang sudah dilakukan, seperti yang terdapat pada Gambar 2.9. MRC pada penelitian

ini disusun sedemikian rupa hingga mampu memberikan sinyal respon menyerupai sistem yang sebenarnya tanpa terjadi *fault*. Dengan mengikuti kaidah yang telah dilakukan oleh Erzberger dan Chen pada tahun 1968, maka *perfect model matching condition* dapat diperoleh.



Gambar 2.9 AFTC berbasis *Fault estimator* dan MRC [5]

Pada Gambar 2.9 untuk  $ref, e, u, x, w, f_a$  dan  $\hat{f}_a$  masing masing secara urut adalah sinyal referensi yang memenuhi *state*  $\dot{w}$ , vektor *state error*, sinyal *input*, *state* dari plan, *state* dari MRC, *fault* dan estimasi *fault*.

Pemodelan robot pada penelitian ini, dibawa ke dalam bentuk LPV (*Linear Time Varying*) terdapat pada Persamaan 2.9.

$$\dot{x} = A(x)x + B(x)u + Ff_a \quad (2.9)$$

dengan  $Ff_a$  adalah *fault* yang terjadi pada sistem.

Pada kondisi ketika tidak terjadi *fault* sinyal kontrol pada sistem tersebut dapat dinyatakan seperti berikut:

$$u = B(x)^+(-K_1(x)x + K_2ref) \quad (2.10)$$

dengan  $(-K_1(x)x = (A(x) - A_m)$  dan  $K_2 = B_m$ ,  $A(x)$  adalah matriks A pada pemodelan plan, sedangkan  $A_m$  adalah matriks A pada pemodelan MRC, dan  $B_m$  adalah matriks B pada pemodelan MRC. Pemodelan MRC dapat dinyatakan seperti pada Persamaan 2.11 berikut ini,

$$\dot{w} = A_m w + B_m ref \quad (2.11)$$

$$\text{diketahui vektor } \textit{state error} \text{ adalah } e = x - w \quad (2.12)$$

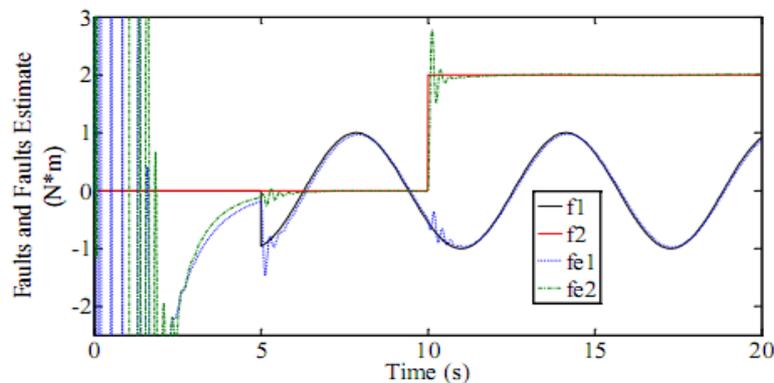
$$\text{dan dinamika } \textit{error} \text{ adalah } \dot{e} = A_m e + (A(x) - A_m)x + B(x)u - B_m ref \quad (2.13)$$

$$\text{jika Persamaan (2.11) dimasukkan pada Persamaan (2.13) maka dinamika } \textit{error} \text{ sistem menjadi: } \dot{e} = A_m e \quad (2.14)$$

Pada kondisi ketika terjadi *fault*, diasumsikan bahwa dinamika *fault estimator* adalah sebagai berikut:  $\dot{\hat{f}}_a = K(\dot{e} + e)$  (2.15)

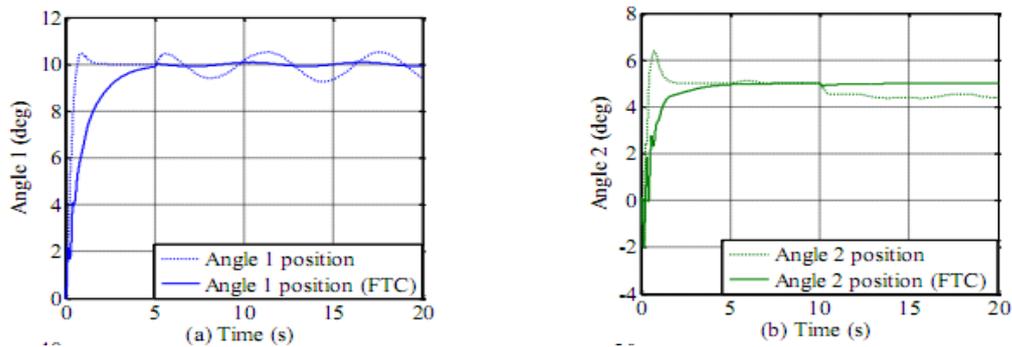
dimana  $\hat{f}_a$  adalah estimasi *fault* sedangkan  $K$  adalah *Gain estimator* yang berseuaian. Sehingga AFCT (sinyal kontrol dan kompensasi kesalahan) seperti pada Gambar 2.9 dapat dituliskan pada Persamaan (2.16). Dimana kinerja dari sistem akan sangat bagus, bergantung pada hasil dari estimasi sinyal kesalahan yang dilakukan.

$$u = B(x)^+(-K_1(x)x + K_2ref - F\hat{f}_a)$$
 (2.16)



Gambar 2.10 *Fault* pada Join 1,2 dan estimasinya

Hasil dari diterapkannya algoritma AFTC berdasarkan MRC pada manipulator 2 DOF dapat dilihat pada Gambar 2.12. Jika dibandingkan dengan PFTC, AFTC dengan metode ini menjadi jauh lebih lambat dalam mencapai *steady state*. Hal ini dikarenakan *fault estimator* juga bekerja cukup lambat dalam mengestimasi sinyal kesalahan yang terjadi. Pada Gambar 2.11 terlihat bahwa estimasi sinyal dapat terealisasi pada detik ke-5. Sehingga kompensasi kesalahanpun dapat dilakukan setelah sinyal hasil estimasinya terbentuk. Perlu diketahui bahwa kesalahan disini mengacu pada kesalahan aktuator pada manipulator 2 DOF.



Gambar 2.11 *Tracking* Posisi Pada Join 1,2

Kelebihan dari sistem ini adalah AFTC mampu bekerja dengan baik dan menjaga agar sistem mampu bekerja dengan optimal meski sedang terjadi *fault*. AFTC yang diterapkan juga mampu menghilangkan *overshoot* akibat dari *fault* yang terjadi, meskipun juga masih terdapat kekurangan seperti yang dijelaskan sebelumnya [5].

## 2.2 Teori Dasar

Beberapa teori dasar yang menunjang dalam penelitian ini akan dibahas pada subbab dibawah ini, salah stu diantaranya adalah Hukum CTC yang akan digunakan sebagai dasar dari seluruh konsep yang akan dirancang pada sisitem ini.

### 2.2.1 Pemodelan Dinamika Robot Manipulator 2 DOF

Simulasi Robot Manipulator 2 DOF secara luas digunakan seperti pada literatur [7,8]. Pemodelan dinamika robot ini, dapat diturunkan dengan persamaan gerak Lagrange. Dinamika manipulator pada Gambar 2.12 dapat diperoleh dengan mengasumsikan masa *link* dikonsentrasikan pada ujungnya. Variabel Join adalah  $q = [q_1 \ q_2]^T$  dan torsi yang bekerja pada robot  $\tau = [\tau_1 \ \tau_2]^T$ .

#### 2.2.1.1 Energi Potensial dan Energi Kinetik

Dinamika Robot Manipulator 2 DOF dapat ditentukan dengan menggunakan Persamaan Lagrange. Namun sebelum itu, terlebih dahulu harus didapatkan energi kinetik dan potensial yang bekerja pada lengan robot seperti pada Gambar 2.12. Untuk *link* 1 energi kinetik dan potensialnya adalah:

$$K_1 = \frac{1}{2} m_1 a_1^2 \dot{q}_1^2 \quad (2.17)$$

$$P_1 = m_1 g a_1 \sin q_1 \quad (2.18)$$

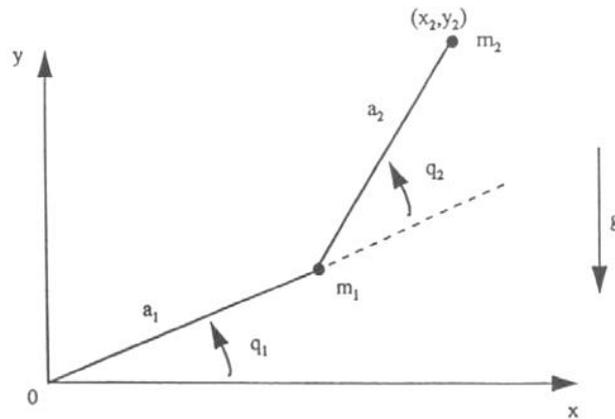
Sedangkan untuk *link* 2, terdapat posisi dan kecepatan:

$$x_2 = a_1 \cos q_1 + a_2 \cos(q_1 + q_2) \quad (2.19)$$

$$y_2 = a_1 \sin q_1 + a_2 \sin(q_1 + q_2) \quad (2.20)$$

$$\dot{x}_2 = -a_1 \dot{q}_1 \sin q_1 - a_2 (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2) \quad (2.21)$$

$$\dot{y}_2 = a_1 \dot{q}_1 \cos q_1 + a_2 (\dot{q}_1 + \dot{q}_2) \cos(q_1 + q_2) \quad (2.22)$$



Gambar 2.12 Robot Manipulator (Planar 2 DOF) [6]

Kuadrat dari kecepatan diperoleh:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = a_1^2 \dot{q}_1^2 + a_2^2 (\dot{q}_1 + \dot{q}_2)^2 + 2a_1 a_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \cos q_2 \quad (2.23)$$

sehingga energi kinetik untuk *link* 2:

$$K_2 = \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_2 a_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 a_2^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 a_1 a_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \cos q_2 \quad (2.24)$$

dan energi potensialnya:

$$P_2 = m_2 g y_2 = m_2 g [a_1 \sin q_1 + a_2 \sin(q_1 + q_2)] \quad (2.25)$$

### 2.2.1.2 Persamaan Lagrange - Euler

Persamaan Lagrange – Euler pada sistem dapat diterapkan jika terlebih dahulu didefinisikan persamaan seperti dibawah ini:

$$L = K - P = K_1 + K_2 - P_1 - P_2 \quad (2.26)$$

dimana  $L$  adalah selisih antara energi kinetik ( $K$ ) dengan energi potensial ( $P$ ). Sehingga jika dimasukkan Persamaan (2.17, 2.18, 2.24 dan 2.25) didapatkan persamaan sebagai berikut:

$$L = \frac{1}{2}m_1a_1^2\dot{q}_1^2 + \frac{1}{2}m_2a_1^2\dot{q}_1^2 + \frac{1}{2}m_2a_2^2(\dot{q}_1 + \dot{q}_2) + m_2a_1a_2(\dot{q}_1^2 + \dot{q}_1\dot{q}_2)\cos q_2 - (m_1 + m_2)ga_1\sin q_1 - m_2ga_2\sin(q_1 + q_2) \quad (2.27)$$

Berikutnya, Persamaan Lagrange – Euler untuk gerakan translasi adalah:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \text{ dengan } i = 1, 2, \dots, n \quad (2.28)$$

dengan:

$q_i$  = Posisi *end of effector* - i

$\dot{q}_i$  = Turunan pertama  $q_i$

$\tau_i$  = Torsi yang diaplikasikan pada robot, untuk menggerakkan join ke-i

$m_i$  = masa lengan ke-i

$a_i$  = panjang lengan ke-i

$g$  = gaya gravitasi

Penerapan Metode LE untuk menyelesaikan persamaan dinamik Robot Manipulator 2 DOF seperti pada Gambar 2.13 adalah sebagai berikut:

$$\frac{\partial L}{\partial \dot{q}_1} = (m_1 + m_2)a_1^2\dot{q}_1 + m_2a_2^2(\dot{q}_1 + \dot{q}_2) + m_2a_1a_2(2\dot{q}_1 + \dot{q}_2)\cos q_2 \quad (2.29)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} = (m_1 + m_2)a_1^2\ddot{q}_1 + m_2a_2^2(\ddot{q}_1 + \ddot{q}_2) + m_2a_1a_2(2\ddot{q}_1 + \ddot{q}_2)\cos q_2 - m_2a_1a_2(\dot{q}_1^2 + \dot{q}_1\dot{q}_2)\sin q_2 \quad (2.30)$$

$$\frac{\partial L}{\partial q_1} = -(m_1 + m_2)ga_1\cos q_1 - m_2ga_2\cos(q_1 + q_2) \quad (2.31)$$

$$\frac{\partial L}{\partial \dot{q}_2} = m_2a_2^2(\dot{q}_1 + \dot{q}_2) + m_2a_1a_2\dot{q}_1\cos q_2 \quad (2.32)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} = m_2a_2^2(\ddot{q}_1 + \ddot{q}_2) + m_2a_1a_2\ddot{q}_1\cos q_2 - m_2a_1a_2\dot{q}_1\dot{q}_2\sin q_2 \quad (2.33)$$

$$\frac{\partial L}{\partial q_2} = -m_2a_1a_2(\dot{q}_1^2 + \dot{q}_1\dot{q}_2)\sin q_2 - m_2ga_2\cos(q_1 + q_2) \quad (2.34)$$

Sehingga di dapatkan dinamika persaman dinamika Robot Manipulator 2 DOF dari Persamaan (2.29– 2.34), dalam persamaan diferensial non linear untuk torsi manipulator 2 DOF adalah:

$$\tau_1 = [(m_1 + m_2)a_1^2 + m_2a_2^2 + m_2a_1a_2\cos q_2]\ddot{q}_1 - [m_2a_2^2 + m_2a_1a_2\cos q_2]\ddot{q}_2 - m_2a_1a_2(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2)\sin q_2 + (m_1 + m_2)ga_1\cos q_1 + m_2ga_2\cos(q_1 + q_2) \quad (2.35)$$

$$\tau_2 = [m_2a_2^2 + m_2a_1a_2\cos q_2]\ddot{q}_1 + m_2a_2^2\ddot{q}_2 + m_2a_1a_2\dot{q}_1^2\sin q_2 + m_2ga_2\cos(q_1 + q_2) \quad (2.36)$$

### 2.2.1.3 Persamaan Dinamika Robot Manipulator 2 DOF

Persamaan umum dinamika robot manipulator dapat dinyatakan sebagai:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) \quad (2.37)$$

jika  $N(q, \dot{q}) = V(q, \dot{q}) + G(q)$

$$\text{maka} \quad \tau = M(q)\ddot{q} + N(q, \dot{q}) \quad (2.38)$$

dengan:

$\tau$  = vektor torsi aktuatuatur yang diterapkan pada sendi-i

$M(q)$  = vektor matriks transformasi dinamik ( $n \times n$ ),  $n$  adalah n-DOF

$V(q, \dot{q})$  = vektor matriks torsi ( $n \times 1$ ) efek gaya Coriolis dan gerak sentrifugal pada sendi-i

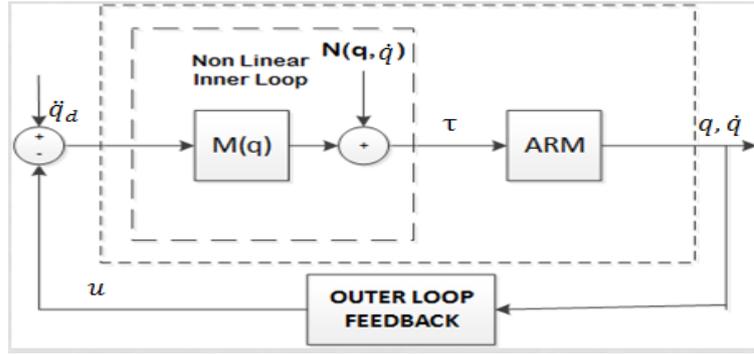
$G(q)$  = vektor matriks torsi ( $n \times 1$ ) efek gravitasi

Dengan demikian dari hasil uraian komponen dinamik robot, dari Persamaan (2.35 dan 2.36) dapat diperoleh persamaan sebagai berikut [2,6]:

$$\begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2\cos q_2 & m_2a_2^2 + m_2a_1a_2\cos q_2 \\ m_2a_2^2 + m_2a_1a_2\cos q_2 & m_2a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -m_2a_1a_2(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2)\sin q_2 \\ m_2a_1a_2\dot{q}_1^2\sin q_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2)ga_1\cos q_1 + m_2ga_2\cos(q_1 + q_2) \\ m_2ga_2\cos(q_1 + q_2) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (2.39)$$

### 2.2.2 Kontrol CTC

*Computed Torque Control* (CTC) adalah algoritma yang digunakan untuk mengatasi dinamika nonlinear robot manipulator. CTC sendiri merupakan sebuah kontroler nonlinear yang telah digunakan secara luas pada kontrol robot manipulator. Kontrol ini didasarkan pada linearisasi umpan balik dan perhitungan torsi yang diperlukan oleh lengan robot menggunakan hukum kontrol umpan balik nonlinear [2].



Gambar 2.13 Kontrol CTC [2]

Teknik ini memerlukan transformasi linear yang dapat menghilangkan unsur nonlinieritas manipulator. Untuk memastikan trayektori *tracking* dengan Variabel Join, *error tracking* didefinisikan sebagai berikut:

$$e(t) = q_d(t) - q(t) \quad (2.40)$$

untuk menunjukkan akibat dari torsi *input*  $\tau(t)$  pada *error tracking*, didefinisikan,

$$\dot{e} = \dot{q}_d - \dot{q} \quad (2.41)$$

$$\ddot{e} = \ddot{q}_d - \ddot{q}$$

Penyelesaian untuk  $\ddot{q}$  pada Persamaan (2.38) disubstitusi ke dalam Persamaan (2.43) menjadi:

$$\ddot{e} = \ddot{q}_d + M^{-1}(N - \tau) \quad (2.42)$$

Persamaan 2.43 tersebut dapat didefinisikan sebagai fungsi *input* kontrol:

$$u = \ddot{q}_d + M^{-1}(N - \tau) \quad (2.43)$$

Didefinisikan juga suatu *state*  $x(t) \in R^{2n}$  dengan

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (2.44)$$

dan dituliskan dinamika *error tracking* sebagai:

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f \quad (2.45)$$

Transformasi linierisasi umpan balik dari Persamaan (2.43) adalah:

$$\tau = M^{-1}(\ddot{q}_d - u) + N \quad (2.46)$$

Persamaan (2.46) disebut sebagai *computed torque control law* [2]. Blok diagram dari kontrol CTC dapat dilihat pada Gambar 2.14.

### 2.2.3 Kontrol PD Outer loop

*Outer loop* kontrol yang dipilih untuk kontrol tambahan  $u(t)$  adalah menggunakan umpan balik *Proporsional* dan *Derivative* (PD) [2]:

$$u = -K_v \dot{e} - K_p e \quad (2.47)$$

dimana  $u$ ,  $e$ ,  $\dot{e}$ ,  $K_p$ ,  $K_v$  masing-masing adalah kontrol nominal, *error tracking* robot, turunan pertama dari *error tracking* robot, *Gain Proportional* dan *Gain Derivative*.

Sehingga dapat diperoleh *input* lengan robot secara keseluruhan menjadi Persamaan (2.48):

$$\tau = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \quad (2.48)$$

### 2.2.4 Pendefinisian Kesalahan Pada Robot Manipulator

Pada penelitian ini, kesalahan terjadi pada *input* dari dinamika robot, sesuai dengan [8,9,10] kesalahan tersebut dapat dikategorikan sebagai kesalahan aktuator dengan definisi seperti di bawah ini:

ketika kondisi *fault*, torsi sebenarnya yang harus diaplikasikan pada robot adalah tidak diketahui,  $\tau$  disini adalah torsi nominal yang dihasilkan oleh kontroler robot. Sedangkan  $f_a$  adalah *input fault*, sehingga torsi aktual yang bekerja pada robot dapat dituliskan:  $\bar{\tau} = \tau + f_a$ . Pada kenyataanya *fault* yang terjadi dapat disebabkan karena kerusakan pada power *supply* atau aktuator yang diakibatkan oleh pengkabelan, atau juga bisa karena usia dari mekanik robot yang sudah cukup lama sehingga performa kerjanya sudah menurun. Dalam penelitian lain [8] jenis / tipe *fault* yang terjadi pada dinamika robot dapat dituliskan sebagai berikut:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau + \tau_f \quad (2.49)$$

dengan jenisnya sebagai berikut:

- Total *Fault*:  $\tau_{f,i} = \tau_i$
- Partial *Fault*:  $\tau_{f,i} = \varepsilon \tau_i$
- Saturation:  $\tau_{f,i} = \tau_i + \tau_{i,max}$
- Bias:  $\tau_{f,i} = b_i$

### 2.2.5 Strategi FTC

Pada dasarnya ada dua tipe strategi FTC, yaitu berupa *Active Fault Tolerant Control* (AFTC) dan *Passive Fault Tolerant Control* (PFTC). Dari beberapa penelitian strategi ini cukup handal dalam menjaga performa sistem agar terhindar dari sinyal kesalahan. Baik AFTC maupun PFTC masing masing punya batasan dalam kerjanya tergantung dari desain yang dirancang [11,12]. Gambar 2.14 ini menjelaskan tentang struktur umum FTC dan perbedaan antara AFTC dan PFTC.

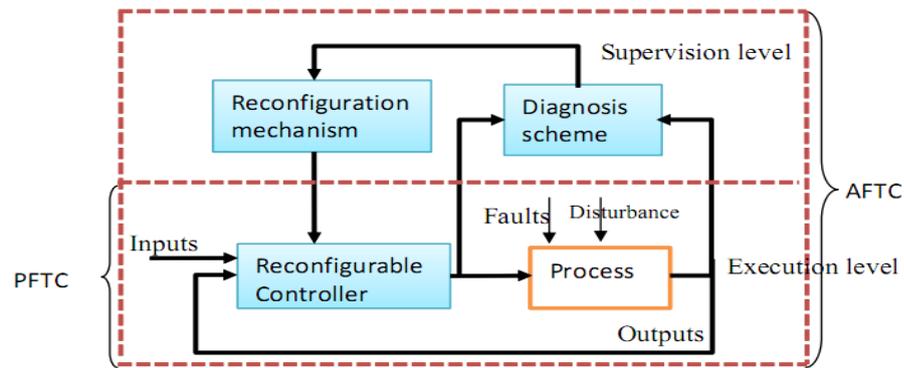
Dalam penelitian ini, PFTC tidak akan dibahas lebih lanjut karena pendekatan strategi yang akan digunakan adalah AFTC. Dalam menggunakan strategi AFTC ada 2 komponen utama yang nantinya akan memberi andil signifikan terhadap kinerja AFTC, yaitu [11]:

1. *Reconfigurable Control* dan Mekanisme Rekonfigurasi
2. Skema FDD (*Fault Detection and Diagnostic*) atau skema FDI (*Fault Detection and Isolation*)

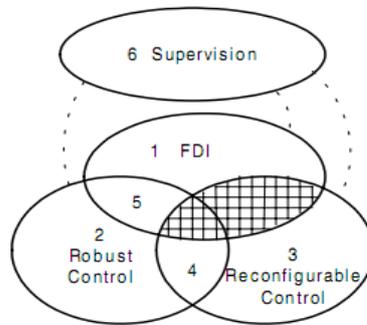
Agar dapat memberikan performa *close loop* sistem yang handal dibutuhkan analisa dan usaha yang besar untuk dapat mengelola kedua komponen tersebut menjadi satu sistem yang *robust*. Gambar 2.15 mengilustrasikan beberapa aktifitas AFTC yang harus disatukan dalam satu sistem. Karena tingkat kerumitan pada masing masing bagian dalam AFTC, banyak para peneliti hanya fokus pada salah satu bagian saja [13]. Namun sesuai dengan penelitian [3], telah dijelaskan bahwa beberapa penelitian AFTC akan berbeda dengan AFTC pada umumnya, karena sistem yang dibangun adalah *Integrated Fault Tolerant Control*. Dalam sistem aktif FTC seperti itu, tidak diperlukan adanya strategi untuk memadukan seluruh bagian dari FTC, karena sistem sudah akan terintegrasi jika syarat dan ketentuan yang dibutuhkan dipatuhi.

Oleh karena itu, dapat ditentukan bahwa bagian dari AFTC yang menjadi fokus pada penelitian adalah Mekanisme Rekonfigurasi. Dimana mekanisme rekonfigurasi pada aktif FTC dapat dilakukan dengan dua cara [12]:

1. Merancang ulang kontroler (*controller redesign*) dengan memodifikasi kontrol nominal
2. Mengkompensasi kesalahan dengan tanpa mengubah kontroler nominal



Gambar 2.14 Struktur AFTC dan PFTC [10]



Gambar 2.15 Bagian-Bagian dari AFTC

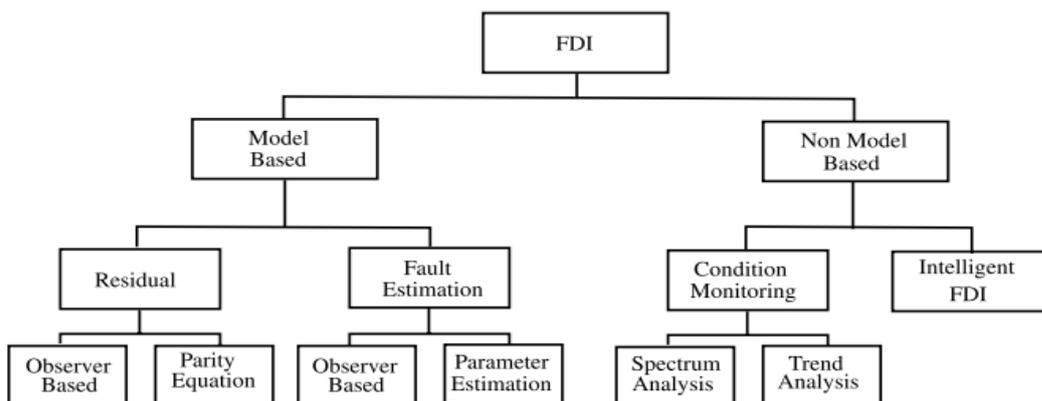
Metode kompensasi kesalahan merupakan bidang penelitian yang relatif baru dan banyak diminati. Dipilihnya metode ini untuk algoritma aktif FTC adalah karena metode ini tidak memerlukan beban komputasi yang besar karena skema identifikasi kesalahan dan mekanisme rekonfigurasi didesain secara terintegrasi nantinya [12].

Dalam perkembangannya, bagian dari AFTC yaitu FDI juga mendapat banyak perhatian untuk dilakukan penelitian. FDI (*Fault Detection and Isolation*) adalah suatu teknik yang digunakan untuk memberikan informasi apakah terjadi suatu kesalahan pada sistem. *Output* dari teknik ini diharapkan mampu memberikan informasi kesalahan secara akurat yang terjadi pada sistem, baik waktu, lokasi dan juga jenisnya, sehingga kesalahan yang terjadi dapat segera ditangani.

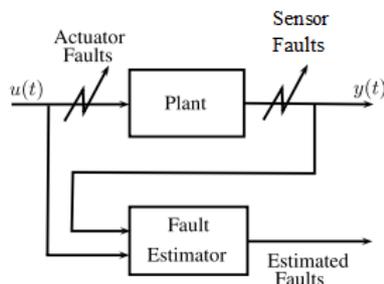
Ide dasar dari teknik FDI adalah membandingkan perilaku sistem yang diamati dengan perilaku sistem yang diharapkan. Perbedaan ini sering dikenal

dengan nama residual. Berdasarkan [14] terdapat beberapa klasifikasi Teknik FDI seperti pada Gambar 2.16

Saat ini telah berkembang konsep baru untuk membangun skema FDI tanpa menggunakan analisis residual, yaitu pendekatan estimasi kesalahan. Ide dasarnya adalah jika kesalahan dapat diestimasi secara akurat, maka seluruh informasi meliputi jenis, besar, lokasi dan waktu kesalahan dapat diperoleh. Dengan demikian, pendekatan estimasi kesalahan memberikan cara yang lebih langsung untuk memperoleh informasi kesalahan dari pada pendekatan berbasis residual. Gambar 2.17 adalah ilustrasi dari deteksi kesalahan berdasarkan estimasi kesalahan yang terjadi pada sistem [14]. Namun menurut Patton, pembagian metode dalam dinamika AFTC sendiri tidak bisa dipastikan karena banyanya penelitian yang sedang dilakukan dan terus berkembang sejak tahun 1997 [12].



Gambar 2.16 Klasifikasi Teknik FDI [14]



Gambar 2.17 Blok Diagram Teknik FDI berdasarkan Estimasi Kesalahan [14]

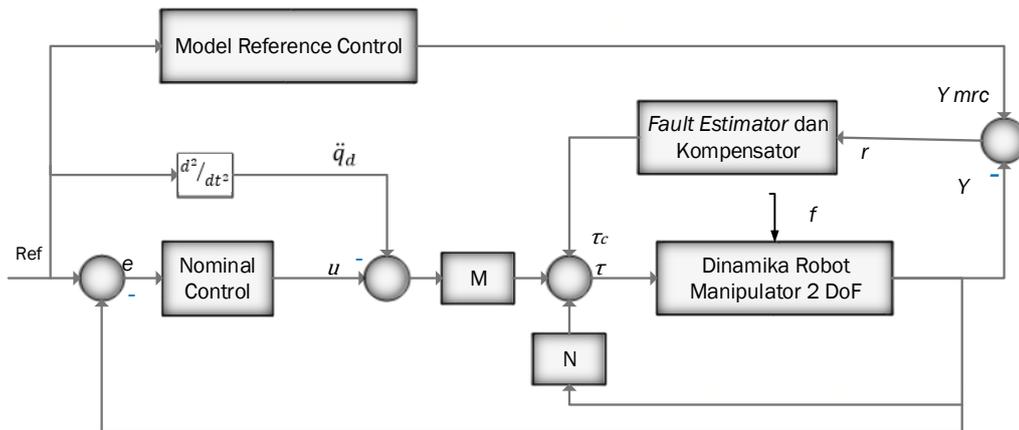
## BAB 3

### METODOLOGI PENELITIAN

Pada bagian ini diuraikan desain, metode, atau pendekatan yang digunakan dalam menjawab permasalahan penelitian untuk mencapai tujuan penelitian, serta tahapan penelitian secara rinci, singkat dan jelas.

#### 3.1 Rancangan Umum Sistem

Rancangan algoritma AFTC yang diajukan pada penelitian ini terdapat pada Gambar 3.1. *Model Reference Control* pada sistem ini, digunakan untuk mengestimasi sekaligus mengkompensasi sinyal kesalahan aktuator. Dengan adanya mekanisme ini, diharapkan dinamika robot masih tetap bekerja dengan baik meskipun terjadi *fault* pada aktuatornya.



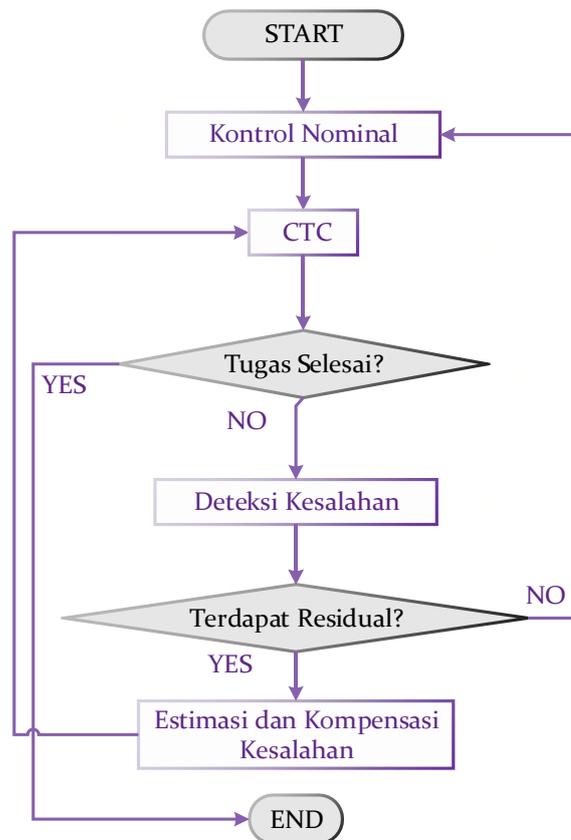
Gambar 3.1 Blok Diagram Rancangan AFTC berbasis MRC

Pada Gambar 3.1,  $f, \tau_c, e, \tau, u, \ddot{q}_d, Y, Y_{mrc}, r$  masing-masing adalah kesalahan aktuator, torsi kompensator, *error tracking* sistem, torsi nominal, sinyal kontrol nominal, referensi percepatan, *output* aktual dari Robot Manipulator, *output* dari MRC, sinyal residual. Pada blok diagram tersebut dapat diketahui bahwa adanya kesalahan pada plan akan diestimasi dan dikompensasi menggunakan *Model Reference Control* yang dirancang berdasarkan Hukum CTC dan kontroler PID. Dengan adanya kompensasi kesalahan yang terjadi pada aktuator, diharapkan sinyal

kesalahan yang terjadi pada sistem dapat diminimalisir serta dapat mempertahankan performa kerja sistem agar tidak mengalami penurunan.

### 3.2 Diagram Alir Kerja Sistem

Gambar 3.2 menjelaskan diagram alir dari sistem pada penelitian ini. Pada *flowchart* tersebut dapat diketahui urutan dan sistematika kerja dari sistem yang dirancang.



Gambar 3.2 *Flowchart* AFTC berbasis MRC

### 3.3 Pendefinisian Kesalahan

Seperti yang telah dijelaskan pada Kajian Pustaka 2.1 tentang karakteristik Robot Manipulator 2 DOF dengan kesalahan pada aktuator, dapat diketahui bahwa berdasarkan kajian terkait [2] maka Robot Manipulator 2 DOF tidak mampu melakukan *tracking* dengan baik ketika *fault* terjadi pada aktuatornya, seperti pada Gambar 2.3 dan Gambar 2.5. Sehingga berdasarkan pendefinisian kesalahan pada

Subbab 2.2.4, dapat diperoleh persamaan kesalahan aktuator pada Robot Manipulator 2 DOF [1] yang bersifat *additive* sebagai berikut:

1. Persamaan model plan dengan kesalahan *additive* pada aktuator akan mengakibatkan torsi *input* mengalami perubahan, karena efek dari *fault* yang terjadi, seperti pada Persamaan (3.1):

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau + \tau_f \quad (3.1)$$

dimana  $\tau_f$  adalah efek dari kesalahan aktuator yang mempengaruhi torsi *input* atau dalam penelitian ini akan disebut sebagai *fault torque*.

2. Jika diketahui dinamika *error tracking* pada Persamaan (2.43) adalah  $\ddot{e} = \ddot{q}_d - \ddot{q}$ , kemudian dari Persamaan (3.1) diperoleh:

$$\ddot{q} = M^{-1}(-N + \tau_f + \tau) \quad (3.2)$$

jika penyelesaian  $\ddot{q}$  disubstitusikan pada  $\ddot{e}$ , maka diperoleh:

$$\ddot{e} = \ddot{q}_d - M^{-1}(-N + \tau_f + \tau) \quad (3.3)$$

dari Persamaan (3.3) dapat diperoleh dua fungsi yaitu: fungsi *input* kontrol pada Persamaan (2.41) dan fungsi kesalahan aktuator yang bersifat *additive* dapat didefinisikan sebagai berikut:

$$f_a = M^{-1}\tau_f \quad (3.4)$$

dimana  $f_a$  adalah kesalahan aktuator. Pada jenis kesalahan ini, kesalahan stokastik dapat dianggap terjadi sesuai dengan penelitian [4], yaitu dengan cara memberikan Sinyal Gaussian yang bersifat random dengan distribusi normal ( $\mu = 0$  dan  $\sigma = 0.04$ ).

Sedangkan untuk kesalahan *multiplicative*, maka:

$$f_a = (100\% - \beta)\tau \quad (3.5)$$

Pada Persamaan (3.5) diasumsikan ketika  $\tau = 100\%$ , maka sistem bekerja dengan normal, sehingga jika Persamaan (2.46) disubstitusikan pada Persamaan (3.5), fungsi  $f_a$  pada sistem ini akan menjadi:

$$f_a = (100\% - \beta) \left\{ M^{-1}(\ddot{q}_d - u) + N \right\} \quad (3.6)$$

dimana  $\beta$  adalah range kesalahan yang terjadi pada aktuator, dimana *range* kesalahan aktuator yang akan diuji adalah  $0\% \leq \beta \leq 75\%$ .

### 3.4 Perancangan MRC

Kontribusi dari penelitian ini terdapat pada perancangan MRC yang akan dijelaskan pada subbab ini, dimana MRC yang dirancang disusun berdasarkan performa dinamika robot dengan Kontroler PID dan Hukum CTC. Berdasarkan [5,14] MRC dapat dituliskan dalam sebuah persamaan yaitu:

$$\dot{x} = Ax + Bv \quad (3.7)$$

dimana  $\dot{x}$  dan  $v$  masing-masing adalah vektor *state* dari MRC dan vektor *input/referensi* sistem. MRC dapat dibentuk jika Matriks  $A$  dan  $B$  adalah *Controllable* dan *Observable* [5]. Kontroler PID disini berfungsi untuk memenuhi persyaratan tersebut dan mendapatkan nilai matriks  $A$  dan  $B$  yang bersesuaian. Sedangkan Hukum CTC, akan menghasilkan sebuah persamaan *state*, dimana pada persamaan tersebut Matriks  $B$  akan bergantung hanya pada nilai referensi saja seperti pada Persamaan (3.7). Lebih jelasnya, Persamaan (3.7) dapat diperoleh dengan langkah-langkah sebagai berikut:

1. Berdasarkan hukum CTC dalam kondisi ideal tanpa terjadi kesalahan aktuator, maka persamaan dinamika *error tracking* dapat ditentukan sebagai berikut:

$$\ddot{e} = u \quad (3.8)$$

dan untuk mendesain MRC berdasarkan konsep tersebut, maka dapat dituliskan sebuah hubungan seperti ini:  $e_m = u_m$ , sehingga untuk mendapatkan matriks  $A$  dan  $B$  yang bersesuaian sekaligus memenuhi persyaratan yang sudah ditentukan, dapat didefinisikan nilai  $u_m$  sebagai berikut:

$$u_m = K_v \dot{q}_m + K_p q_m - K_v \dot{q}_d - K_p q_d - K_i \varepsilon_m \quad (3.9)$$

dimana  $K_v, K_p, K_i, \dot{q}_m, q_m, \varepsilon_m$  masing-masing adalah *Gain Derivative*, *Gain Proportional*, *Gain Integral*, kecepatan dari MRC, posisi dari MRC dan *Integral error* pada MRC. Nilai *gain* yang akan digunakan pada pemodelan ini sama dengan nilai *gain* pada Kontrol Nominal. Nilai *Gain*  $K_p$  dan  $K_v$  masing masing adalah 30 dan 300, sedangkan nilai *Gain*  $K_i$  pada pemodelan ini berupa Matriks Identitas

2. Kemudian, turunan kedua dari *error* pada MRC ( $\ddot{e}_m$ ) dapat dihasilkan dengan cara:  $\ddot{q}_d - \ddot{q}_m$ , jika  $e_m = u_m$  maka dapat dituliskan sebuah persamaan sebagai berikut:

$$\ddot{q}_d - \ddot{q}_m = K_v \dot{q}_m + K_p q_m - K_v \dot{q}_d - K_p q_d - K_i \varepsilon_m \quad (3.10)$$

Solusi dari  $\ddot{q}_m$  dari Persamaan (3.10) akan menghasilkan persamaan seperti di bawah ini:

$$\ddot{q}_m = -K_v \dot{q}_m - K_p q_m + K_v \dot{q}_d + K_p q_d + K_i \varepsilon_m + \ddot{q}_d \quad (3.11)$$

3. Kemudian untuk mendesain Matriks *A* dan *B* dapat didefinisikan *state* variabel sebagai berikut:

$$x_1 = \varepsilon_m,$$

$$x_2 = q_m$$

$$x_3 = \dot{q}_m$$

$$\dot{x}_1 = \dot{\varepsilon}_m = e_m = q_d - q = q_d - x_2 \quad (3.12)$$

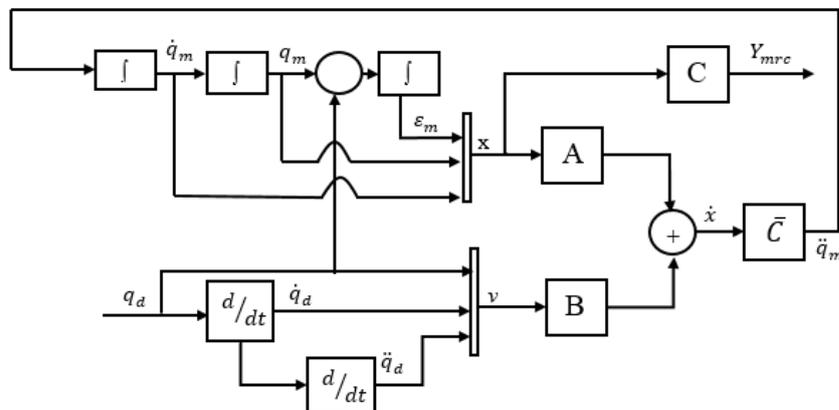
$$\dot{x}_2 = \dot{q}_m = x_3$$

$$\dot{x}_3 = \ddot{q}_m$$

maka Persamaan (3.12) dapat ditulis kedalam bentuk matriks-matriks penyusun Persamaan (3.7) sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & -I & 0 \\ 0 & 0 & I \\ K_i & -K_p & -K_v \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ K_p & K_v & I \end{bmatrix} \begin{bmatrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{bmatrix} \quad (3.13)$$

dimana  $q_d, \dot{q}_d, \ddot{q}_d$  adalah vektor *input/referensi* yang pada Persamaan (3.7) dituliskan dengan ( $v$ ).



Gambar 3.3 MRC berbasis PID CTC

Gambar 3.3 merupakan rancangan MRC yang dihasilkan, dengan nilai Matriks  $A$  dan  $B$  yang telah didapatkan pada Persamaan (3.13).  $Y_{\text{mrc}}$  adalah *output* dari MRC sedangkan *ref* adalah *input* referensi posisi, kecepatan, dan percepatan. *Gain* pada pemodelan MRC ini ditulis dalam bentuk matriks dengan cara:

$$K = \text{diag}\{K\} = \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \quad (3.14)$$

Nilai  $C$  dan  $\bar{C}$  pada Gambar 3.3 masing-masing adalah:

$$C = [0 \quad I \quad 0] \text{ dan } \bar{C} = [0 \quad 0 \quad I]$$

Matriks  $\bar{C}$  pada perancangan ini digunakan untuk mendapatkan nilai *state*  $\dot{q}_m$  dari pemodelan MRC sehingga *state feedback* posisi, kecepatan dan *integral error* pada pemodelan ini bisa diperoleh.

### 3.5 Perancangan *Estimator* dan *Kompensator*

*Estimator* dalam penelitian dirancang berdasarkan *invers* model dinamika *error* keseluruhan sistem *closed loop*-nya. *Error* yang akan digunakan oleh *estimator* untuk mengestimasi kesalahan merupakan hasil perbandingan antara sistem sebenarnya dengan MRC atau dapat disebut dengan residual. Berdasarkan Persamaan (3.3) dapat diketahui bahwa ketika terjadi kesalahan pada aktuator maka dapat didefinisikan sebuah persamaan sebagai berikut:

$$\ddot{e} = u + f_a \quad (3.15)$$

Karena residual merupakan hasil perbandingan antara posisi yang diharapkan dari MRC dengan posisi yang sebenarnya dari Robot Manipulator, maka dapat ditulis sebuah persamaan:

$$r = q_m - q \quad (3.16)$$

dimana  $r$  adalah residual,  $q_m$  adalah posisi dari MRC, dan  $q$  adalah posisi sebenarnya dari Robot Manipulator. Karena residual dapat dikatakan sebagai *error* dari keseluruhan sistem, maka Persamaan (3.15) dapat digunakan sebagai dasar penyusunan persamaan *estimator*. Berdasarkan perumusan tersebut maka dapat disusun sebuah Persamaan *State Estimator* sebagai berikut:

$$\dot{r} = A_e r + B_e f_a \quad (3.17)$$

dimana matriks  $A_e$  merupakan matriks *state estimator* yang bersesuaian dengan matrik  $A_m$  pada MRC, sehingga matriks tersebut dapat dibentuk dengan mengalikan antara sinyal residual dengan *Gain* PD dari Kontrol Nominal ditambah dengan adanya *Gain*  $K_i$ . Nilai *Gain*  $K_i$  yang diberikan pada pemodelan *state estimator* ini adalah *diag* {1000}.

Matriks  $B_e$  merupakan matriks yang berfungsi untuk menandakan hadirnya kesalahan pada sistem yang dirancang. Sehingga untuk mendapatkan matriks-matriks penyusun Persamaan *state estimator* tersebut dapat didefinisikan sebuah Persamaan seperti pada (3.18).

$$\ddot{r} = -K_p r - K_v \dot{r} - K_i r_i + f_a \quad (3.18)$$

dimana  $\dot{r}, r_i$  masing-masing adalah turunan pertama dari residual dan intergral dari residual. *Estimator* dapat dirancang dengan langkah-langkah di bawah ini:

1. Jika ditentukan *variable state* adalah sebagai berikut:

$$\begin{aligned} z_1 &= r_i \\ z_2 &= r \\ z_3 &= \dot{r} \\ \dot{z}_1 &= r = z_2 \\ \dot{z}_2 &= \dot{r} = z_3 \\ \dot{z}_3 &= \ddot{r} \end{aligned} \quad (3.19)$$

sehingga berdasarkan Persamaan (3.19) maka Persamaan (3.18) dapat diubah kedalam bentuk Persamaan *state* seperti di bawah ini:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_v \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} f_a \quad (3.20)$$

atau dapat ditulis dalam persamaan *state* sebagai berikut:

$$\dot{z} = A_e z + B_e f_a$$

2. Kemudian lakukan *invers model* pada Persamaan (3.20) seperti berikut:

$$\dot{z} - A_e z = B_e f_a \quad (3.21)$$

Substitusi Persamaan (3.4) ke dalam Persamaan (3.21) maka akan didapatkan:

$$(\dot{z} - A_e z)B_e^+ = M^{-1} \tau_f \quad (3.22)$$

Persamaan (3.22) akan dapat digunakan sebagai estimasi kesalahan aktuator. Dari persamaan tersebut, estimasi kesalahan aktuator dapat dituliskan sebagai berikut:

$$\hat{f}_a = (\dot{z} - A_e z)B_e^+ \quad (3.23)$$

Kompensasi kesalahan aktuator dapat dilakukan dengan cara menambahkan Torsi Kompensasi pada Torsi Nominal. Torsi Kompensasi dapat diperoleh dengan cara mencari penyelesaian  $\tau_f$  pada Persamaan (3.22).

Penyelesaian dari  $\tau_f$  pada Persamaan (3.22) selanjutnya akan disebut sebagai Torsi Kompensator atau dapat ditulis seperti pada Persamaan (3.24):

$$M((\dot{z} - A_e z)B_e^+) = \tau_c \quad (3.24)$$

Nilai dari Torsi Nominal dapat diperoleh dari Persamaan (2.46) dengan nilai kontroler nominal ( $u$ ) terdapat pada Persamaan (2.47).

## **BAB 4**

### **HASIL DAN PEMBAHASAN**

Pada bab ini, hasil dari perancangan pada Bab III akan diberikan, beserta dengan pembahasan yang diperlukan. Dengan adanya pengujian, akan diketahui apakah perancangan telah berhasil dilakukan atau masih memiliki kekurangan sekaligus analisa dari keduanya. Rancangan penelitian pada Bab III akan diaplikasikan pada plan berupa Robot Manipulator 2 DOF dan akan disimulasikan dengan Simulink pada Matlab. Robot Manipulator 2 DOF akan diuji menggunakan trayektori sudut  $(q_d, \dot{q}_d, \ddot{q}_d)$  dalam beberapa kasus. Kasus yang pertama adalah kinerja dari MRC yang telah dirancang, kemudian kinerja dari rancangan kompensator dan *estimator*.

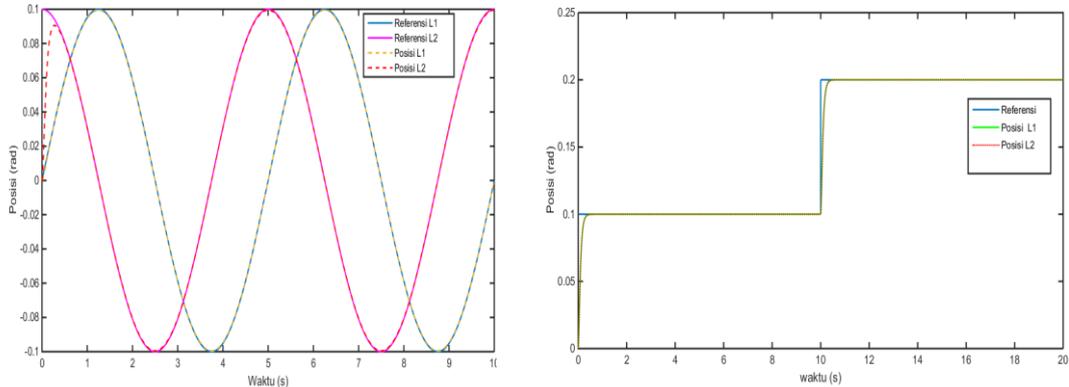
#### **4.1 Pengujian Kontrol Nominal**

Pengujian *tracking* untuk subbab ini, sebenarnya sudah dilakukan dan diberikan hasilnya pada Bab 2. Oleh karena itu, pada perancangan Bab III juga tidak diberikan penjabaran terkait rancangan untuk kontrol nominal. Pada bab ini hasil dari kinerja kontrol nominal hanya akan ditunjukkan kembali untuk mempermudah pembaca dalam membandingkan hasil sebelum dan sesudah diterapkan sistem hasil rancangan atau setelah algoritma AFTC diberikan.

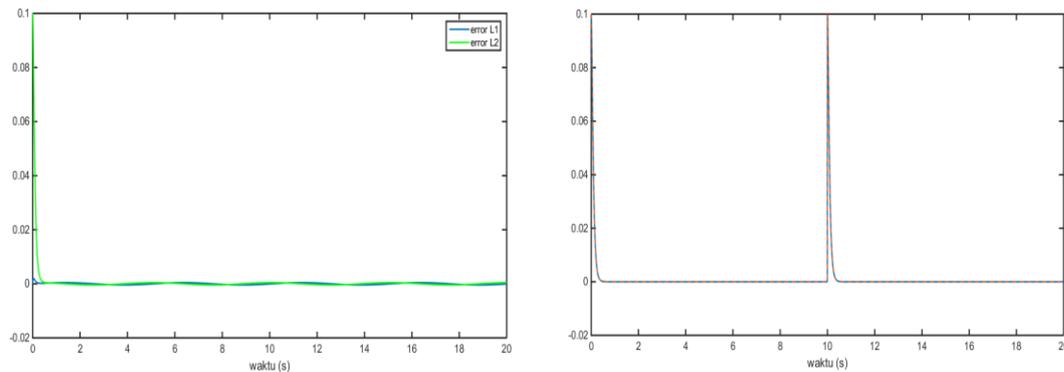
##### **4.1.1 Pengujian Kontrol Nominal Tanpa Terjadi *Fault***

Gambar 4.1 adalah pengujian *tracking* dengan kontroler nominal tanpa terjadi kesalahan pada aktuator. Referensi Trayektori yang diberikan adalah berupa sinyal sinus dan sinyal step. Dari gambar tersebut dapat diketahui bahwa kontroler nominal dapat bekerja dengan baik ketika tidak terjadi *fault* pada aktuatornya. Kinerja kontroler ketika terjadi *fault* akan dijelaskan pada subbab selanjutnya. Gambar 4.2 menunjukkan bahwa *error tracking* robot sangat kecil bahkan mendekati nol. Nilai yang tinggi pada awal waktu maupun pada detik ke-10 Gambar 4.2 hanyalah terjadi ketika sistem mengalami *rise time*, atau bergerak dari posisi awal

menuju posisi yang ditentukan. Sehingga dapat disimpulkan sistem bekerja dengan baik pada kondisi *fault free*.



Gambar 4.1 Tracking Posisi Robot (*fault free case*)



Gambar 4.2 Error tracking Robot (*fault free case*)

#### 4.1.2 Pengujian Kontrol Nominal Ketika Terjadi *Fault*

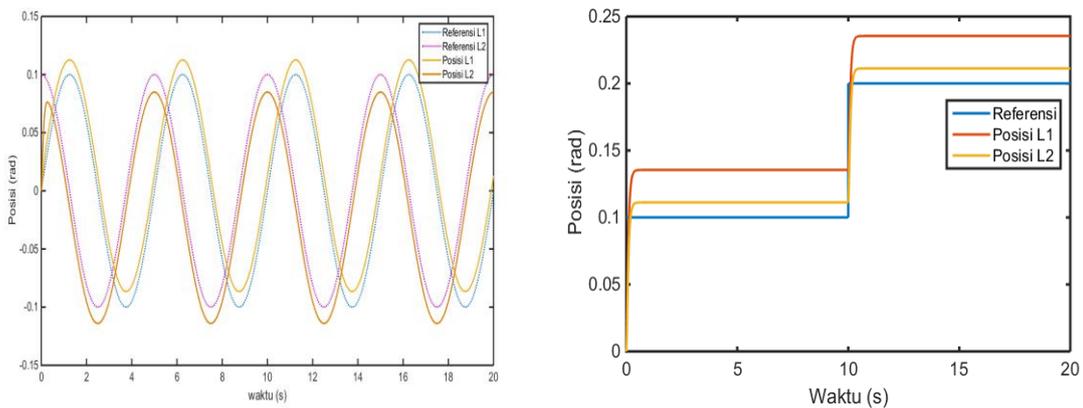
*Fault* yang terjadi pada penelitian ini bersifat *additive* dan multiplikatif. Kesalahan multiplikatif pada penelitian ini akan terjadi dalam bentuk persentase. Pada subbab ini tidak dibahas jenis-jenis kesalahan yang terjadi, karena subbab ini hanya difokuskan untuk mengetahui perbedaan pada sistem sebelum dan sesudah terjadi *fault*.

##### 4.1.2.1 Pengujian Kontrol Nominal dengan Kesalahan *Additive*

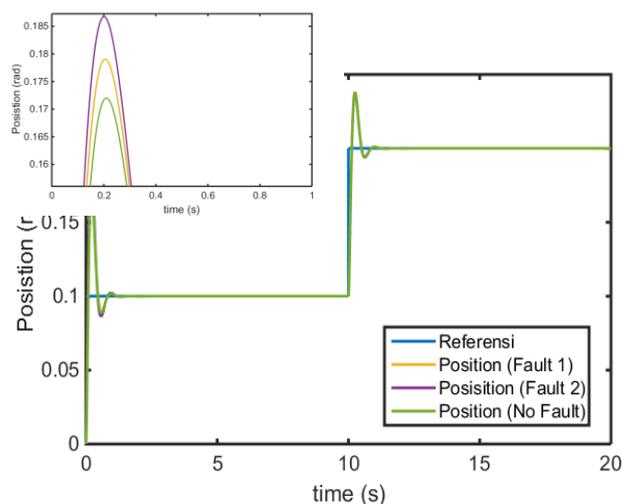
Seperti yang telah dituliskan pada Persamaan (3.1), kesalahan *additive* merupakan sinyal yang tidak diketahui yang mempengaruhi kinerja torsi *input*. Sehingga, torsi yang bekerja akan mengalami kenaikan atau penurunan dari nilai sebenarnya. Gambar 4.3 merupakan kinerja kontrol nominal ketika terjadi

kesalahan aktuator. *Fault* terjadi pada aktuator, berupa sinyal konstan, yang menyebabkan torsi *input* yang diterima oleh dinamika robot tidak maksimal, torsi hanya memberikan daya setengah dari kondisi normalnya, atau jika dalam kesalahan multiplikatif dapat dituliskan dengan nilai  $\beta$  adalah 50%.

Dari gambar tersebut dapat diketahui bahwa *tracking* robot mengalami pergeseran dari referensinya, sehingga sistem memiliki *error tracking* yang besar. Sistem ini pada dasarnya dapat diperbaiki jika persamaan  $u$  pada Persamaan (2.43), ditambah dengan *Gain Integral* yang dikalikan dengan *integral error*-nya. Dalam kondisi normal, posisi robot memiliki nilai *overshoot* lebih besar yaitu sekitar 0.097 rad dan dari Gambar 4.4 *overshoot* akan semakin membesar seiring dengan besarnya *magnitude* kesalahan aktuator yang terjadi, meskipun kontroler ini mampu menghasilkan *error tracking* yang mendekati nol.



Gambar 4.3 *Tracking* Posisi dengan *Additive Fault*

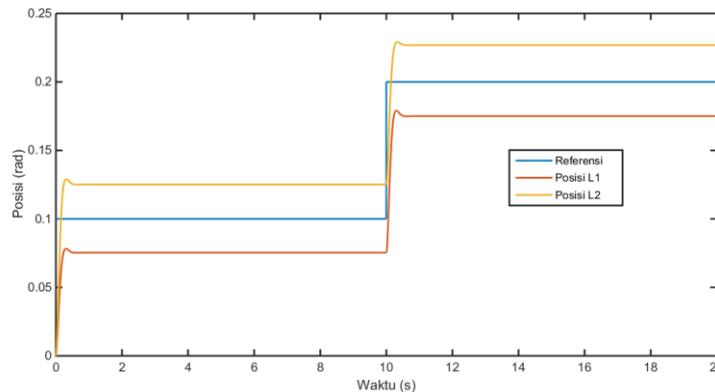


Gambar 4.4 *Tracking* Posisi dengan beberapa *Fault*

*Fault-1* yang terjadi pada robot ini adalah berupa sinyal konstan dengan *magnitude* 10 Nm pada Join-1 dan 5 Nm pada Join-2. *Fault-2*, juga dalam bentuk sinyal konstan dengan *magnitude* yang lebih besar yaitu 20 Nm untuk Join-1 dan 7 Nm untuk Join-2. Perlu diketahui bahwa *fault* pada aktuator robot terjadi sejak robot mulai bekerja atau ketika  $t = 0$ . Dari Gambar 4.4 dapat diketahui bahwa akibat dari terjadinya *fault* pada aktuator robot, *overshoot* akan semakin besar seiring dengan besarnya *magnitude fault* yang terjadi. Jadi bisa disimpulkan bahwa penambahan *Gain Integral*, belum cukup untuk menangani kesalahan aktuator yang terjadi pada Robot Manipulator.

#### 4.1.2.2 Pengujian Kontrol Nominal dengan Kesalahan *Multiplicative*

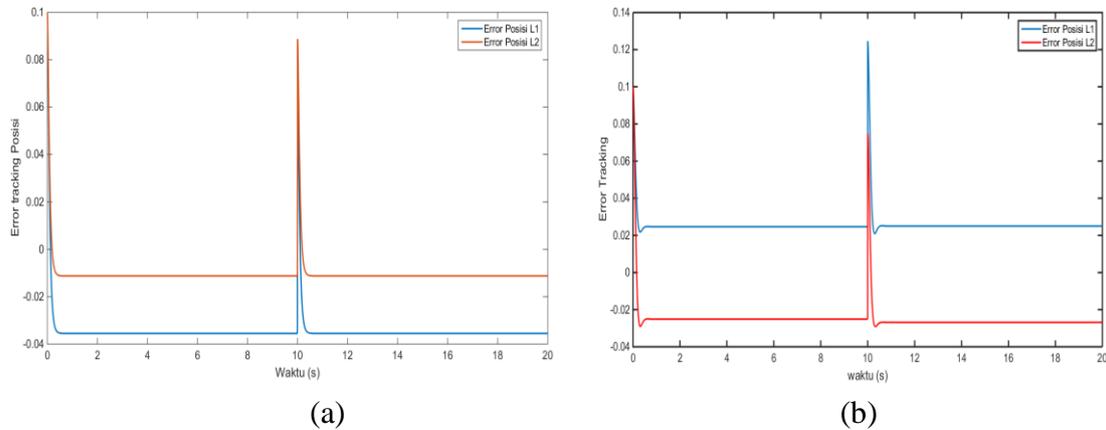
Gambar 4.5 adalah sistem yang mengalami degradasi sebesar 50% pada aktuatornya, sehingga sistem hanya mampu memberikan setengah dari torsi *input* yang seharusnya pada dinamika Robot.



Gambar 4.5 *Tracking* Posisi dengan *Multiplicative Fault*

Gambar 4.6 merupakan perbandingan antara *error tracking* posisi akibat kesalahan *additive* dan *error tracking* posisi akibat kesalahan multiplikatif. Dari gambar tersebut dapat diketahui bahwa sistem dengan kesalahan yang bersifat multiplikatif memiliki dampak yang sama pada kedua posisi lengan robot, sedangkan sistem dengan kesalahan *additive* memiliki dampak yang lebih besar pada Join-1 Robot jika terjadi selama  $t$  simulasi. Jika pada kesalahan multiplikatif *error tracking* yang terjadi adalah sekitar 0.025 untuk Join-1 dan Join-2, maka *error tracking* pada *Fault* yang bersifat *additive* adalah sekitar 0.035 untuk Join-1 dan 0.011 untuk Join-2. Dari Gambar 4.5 terlihat bahwa kesalahan multiplikatif

berdampak lebih cepat dari pada kesalahan *additive*, sehingga posisi Robot wajar jika mengalami *overshoot* sebelum akhirnya bergeser dari nilai referensinya.



Gambar 4.6 Perbandingan *error tracking* antara sistem dengan *Fault Additive* (a) dan Sistem dengan *Fault Multiplicative* (b)

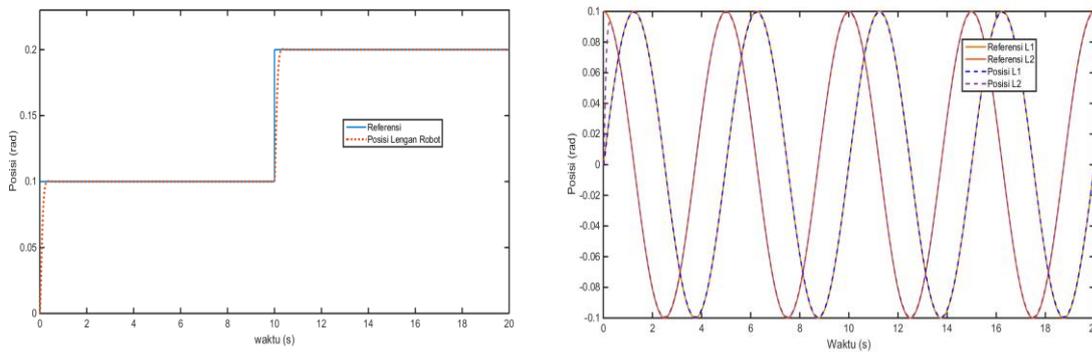
## 4.2 Pengujian MRC dan Residual

*Model Reference Control* atau pada umumnya dikenal sebagai MRC adalah sistem yang diharapkan mampu memberika performa *tracking* yang ideal. MRC pada penelitian ini, akan digunakan sabagai pembanding. Respon posisi dari MRC akan dibandingkan dengan respon posisi Robot yang sebenarnya. Jika sistem yang sebenarnya mengalami *fault*, kemudian dibandingkan dengan respon MRC, maka AFTC akan dapat melakukan estimasi dan kompensasi, jika hasil dari perbandingan tersebut terdapat nilai residual yang signifikan. Jadi dapat dikatakan sinyal residual adalah hasil perbandingan antara sistem yang sebenarnya dengan sistem yang diharapkan.

### 4.2.1 Pengujian MRC

Dari Gambar 4.6 dapat diketahui bahwa hasil rancangan MRC mampu melakukan *tracking* dengan baik. MRC hasil rancangan memerlukan waktu 1 detik untuk *rise time* dan *settling time*. Hal ini akan mempengaruhi kinerja *estimator* dan kompensator berdasarkan [4], jika MRC dapat bekerja dengan cepat, maka ada kemungkinan *estimator* dan kompensator juga akan bekerja dengan cepat.

Selain itu MRC hasil rancangan juga dapat melakukan *tracking* dengan referensi berupa sinyal sinus, sementara untuk penelitian [4] sistem hanya mampu melakukan *tracking* dengan referensi sinyal konstan atau step. Hal ini disebabkan karena nilai referensi pada [4] adalah sebuah persamaan yang membuat nilai referensinya *fix*. Hasil respon posisi dan kecepatan dari MRC pada [4] juga memerlukan waktu lebih lama untuk *rise time* dan *settling time* yaitu sekitar 2 detik.



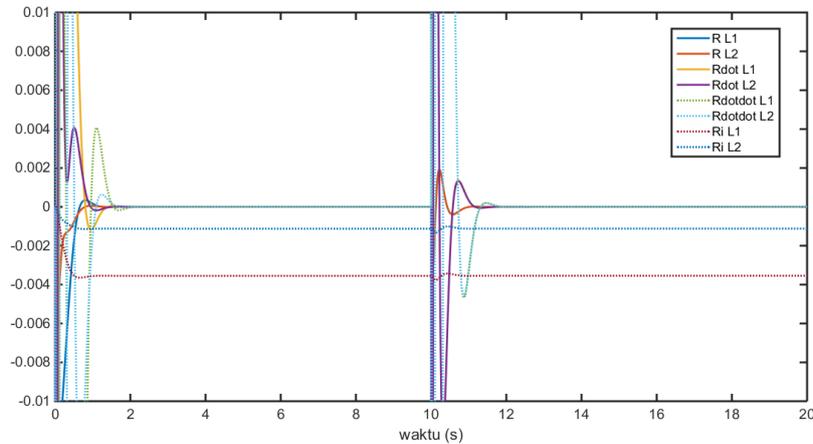
Gambar 4.7 *Tracking* Posisi dari MRC

#### 4.2.2 Pengujian Residual

Sinyal Residual pada penelitian ini mengambil peran cukup penting, seperti yang telah dijelaskan sebelumnya dan berdasarkan Persamaan (3.19) bahwa kerja *estimator* dan kompensator sangat bergantung dari sinyal residual ini. *State* dari *estimator* selain bergantung pada sinyal residual juga bergantung dari turunan dari residual dan juga *integral* dari residual tersebut. *Estimator* dan kompensator akan bekerja dengan sangat baik jika *state* dari *estimator* memiliki nilai selama *t* simulasi dan diketahui nilainya. Gambar 4.8 adalah sinyal residual yang dihasilkan oleh sistem ketika terjadi *fault*.

Perlu diperhatikan bahwa, jika nilai *u* pada Persamaan (2.43) ditambah dengan *Gain Integral*, sinyal residual memang masih bisa dihasilkan namun hanya sesaat ketika *rise time*. Hal ini disebabkan oleh, penambahan *Gain Integral* akan membuat sistem mampu mengatasi terjadinya kesalahan itu sendiri, sehingga meskipun sistem mengalami *fault*, *tracking* masih bisa dilakukan dengan baik. Namun seperti yang telah dijelaskan sebelumnya, penambahan *Gain Integral* juga masih belum mampu menjaga performa robot tetap optimal. Oleh karena itu,

kompensator yang dirancang, juga akan diterapkan pada kontroler tersebut untuk memperkecil efek dari *fault* yang terjadi.



Gambar 4.8 Sinyal Residual, Turunan Pertama, Turunan Kedua dan *Integral* dari Sinyal Residual

### 4.3 Pengujian Kompensasi Kesalahan Aktuator

Pengujian dalam subbab ini, merupakan bagian terpenting dalam penelitian ini. Pengujian akan dilakukan untuk kompensasi kesalahan yang bersifat *additive* maupun multiplikatif. Dalam penelitian ini robot akan mengalami pergeseran dari referensinya jika memiliki  $error\ tracking \geq 0.01$  rad, dan hanya diperbolehkan memiliki amplitudo  $overshoot \leq 0.05$ . Batasan ini diperoleh berdasarkan kinerja robot dalam kondisi ideal.

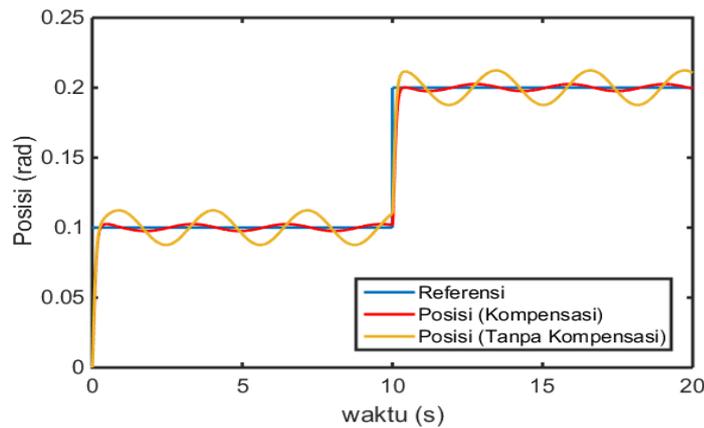
#### 4.3.1 Kompensasi Kesalahan *Additive*

Pengujian terhadap kompensasi kesalahan yang bersifat *additive* akan dilakukan dalam 2 jenis sinyal kesalahan, berupa sinyal konstan dan sinyal sinus. Jenis sinyal konstan yang diberikan akan setara dengan besarnya kesalahan yang terjadi pada kesalahan yang bersifat multiplikatif. Selain itu juga akan ditambahkan pengujian dengan sinyal kesalahan yang memiliki amplitudo yang berbeda pada  $t$  tertentu.

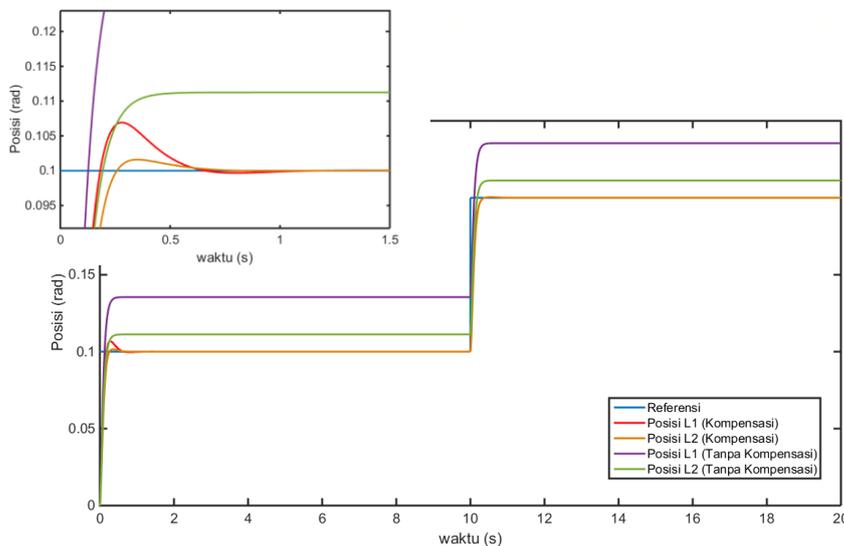
### 4.3.1.1 Kompensasi Kesalahan Aktuator

Kesalahan aktuator yang terjadi pada pengujian ini adalah sama untuk masing-masing joint pada robot, dengan *magnitude* yang diberikan adalah sebagai berikut:

1.  $f_a = 5 * \sin(2t)$  untuk  $0 \leq t \leq 20$
2.  $f_a = \text{Join} - 1 = 14.2 \text{ (Nm)}$ ,  $\text{Join} - 2 = 4.5 \text{ (Nm)}$  untuk  $0 \leq t \leq 20$
3.  $f_a = 0 \leq f_a \leq 10 \text{ (Nm)}$  untuk  $4 \leq t \leq 10$  dan  $14 \leq t \leq 20$



Gambar 4.9 *Tracking* Posisi Robot untuk *Fault* -1

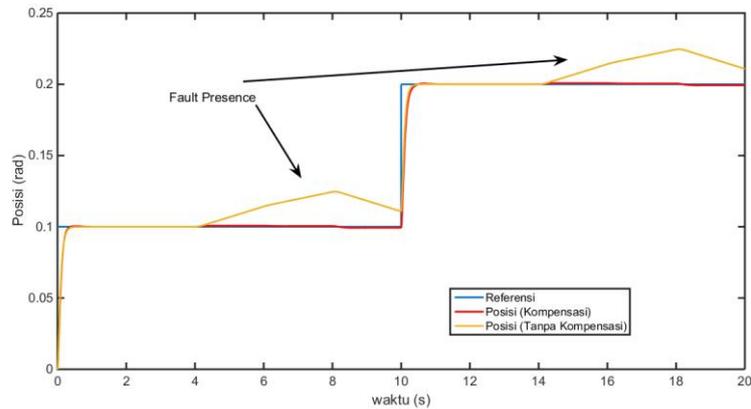


Gambar 4.10 *Tracking* Posisi Robot untuk *Fault* -2

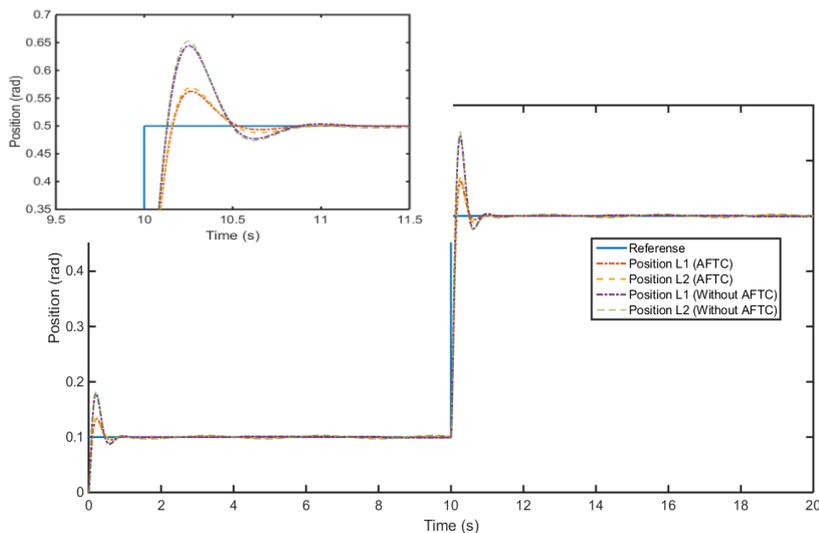
Gambar 4.9 adalah hasil kompensasi kesalahan aktuator dengan *fault* yang pertama berupa sinyal sinus. Sedangkan Gambar 4.10 dan 4.11 adalah hasil kompensasi kesalahan untuk *fault* yang kedua dan ketiga. Dari ketiga Gambar tersebut dapat diketahui bahwa desain kompensator yang telah dirancang dapat bekerja dengan baik dan menghasilkan minimal *error tracking* posisi, mencapai 0.0081 rad untuk

jenis *fault* pertama, 0.0077 rad untuk jenis *fault* kedua. Meskipun sinyal hasil kompensasi pada jenis *fault* ini mengalami *overshoot* pada awalnya, *overshoot* yang terjadi sangat kecil dan masih bisa ditolehir yaitu sekitar 0.0068 rad untuk Join-1 dan 0.0018 rad untuk Join-2. Sedangkan untuk jenis *fault* yang terakhir sistem mampu memberikan *error tracking* mencapai 0.0082 rad. Dari ketiga Gambar tersebut dapat disimpulkan bahwa rancangan kompensator dapat bekerja dengan baik untuk mengkompensasi sinyal kesalahan akibat terjadinya *fault* pada aktuator Robot Manipulator.

Gambar 4.12 membuktikan bahwa kompensator mampu meminimalisir *overshoot* sampai 0.047 rad untuk Join-1 dan 0.041 rad untuk Join-2, serta menghasilkan *error tracking* 0.008 rad. Nilai *overshoot* hasil kompensasi bahkan lebih kecil dari nilai *overshoot* ketika kondisi *fault free case*.



Gambar 4.11 Tracking Posisi Robot untuk *Fault* ke-2



Gambar 4.12 Tracking Posisi Robot ( $u=PID$ )

#### 4.3.1.2 Kompensasi Kesalahan Aktuator dan *Disturbance*

Pada pengujian kali ini akan ditambahkan gangguan pada torsi *input*. Sehingga selain mengalami *fault*, terdapat juga gangguan pada torsinya. Besarnya gangguan yang diberikan adalah sebagai berikut, pada Join-1 dan Join-2:

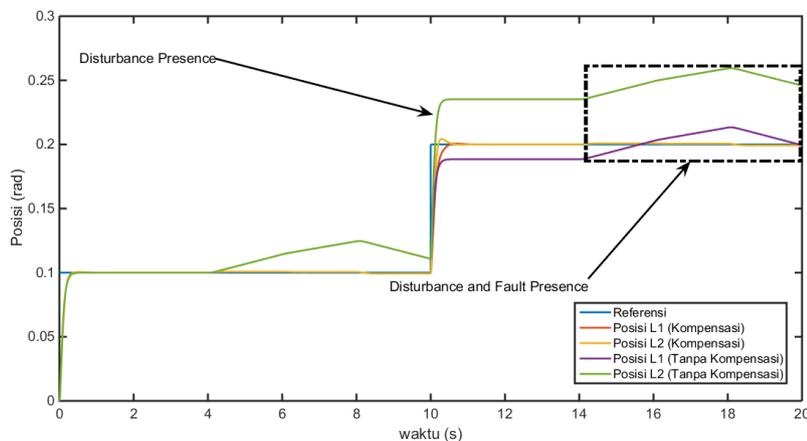
$$d = 5 \text{ (Nm)}, \text{ untuk } 10 \leq t \leq 20$$

dengan *fault* yang terjadi adalah:

$$0 \leq f_a \leq 10 \text{ (Nm)} \text{ untuk } 4 \leq t \leq 10 \text{ dan } 14 \leq t \leq 20$$

Gambar 4.13 menunjukkan bahwa sistem mampu mengkompensasi hadirnya gangguan sekaligus kesalahan yang terjadi pada actuator hingga mencapai *error tracking* sebesar 0.0084 rad untuk Join-1 dan 0.0079 untuk Join-2. Hal ini bisa terjadi karena rancangan kompensator akan selalu memberikan torsi tambahan pada torsi nominal.

Selama kompensator mendeteksi adanya sinyal residual maka selama itu pula kompensator akan bisa bekerja, meskipun sinyal residual hanya muncul sesaat ataupun akan muncul selama  $t$  yang ditentukan.



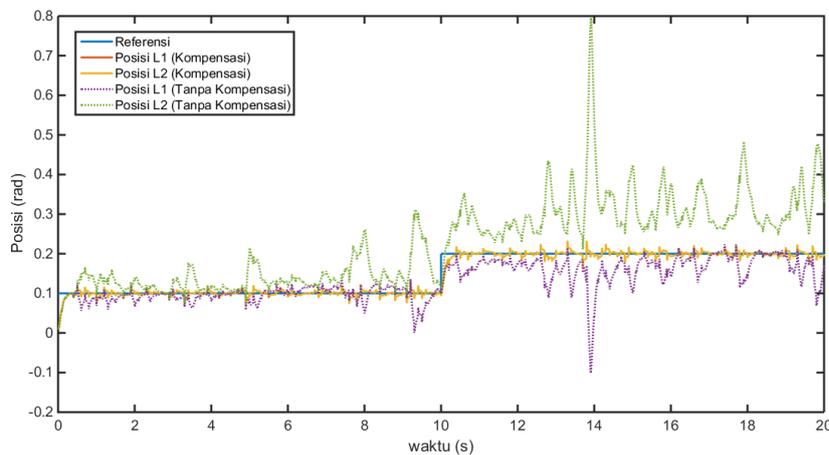
Gambar 4.13 *Tracking* Posisi dengan *Fault* Aktuator dan *Disturbance*

#### 4.3.1.3 Kompensasi Kesalahan Aktuator dan *Noise* pada *Output*

Pada pengujian ini, selain *fault* pada aktuator juga terdapat *noise* pada *output*-nya. *Noise* yang terjadi berupa Sinyal Gaussian yang bersifat random dengan distribusi normal ( $\mu = 0$  dan  $\sigma = 8 \times 10^{-5}$ ). Sedangkan *fault* yang terjadi adalah *fault-2* pada Persamaan (4.1). *Noise* dengan nilai *variance* yang diberikan, adalah *noise* maksimal yang mampu ditangani oleh kompensator.

Meskipun rancangan kompensator kurang maksimal dalam memperbaiki performa *tracking* robot, namun kompensator masih bisa mencapai selisih paling minimal dengan ketentuan *error tracking* yaitu 0.0098 rad, dalam kondisi ketika terjadi *fault* pada aktuatornya disertai dengan terjadinya *noise* pada aktuator.

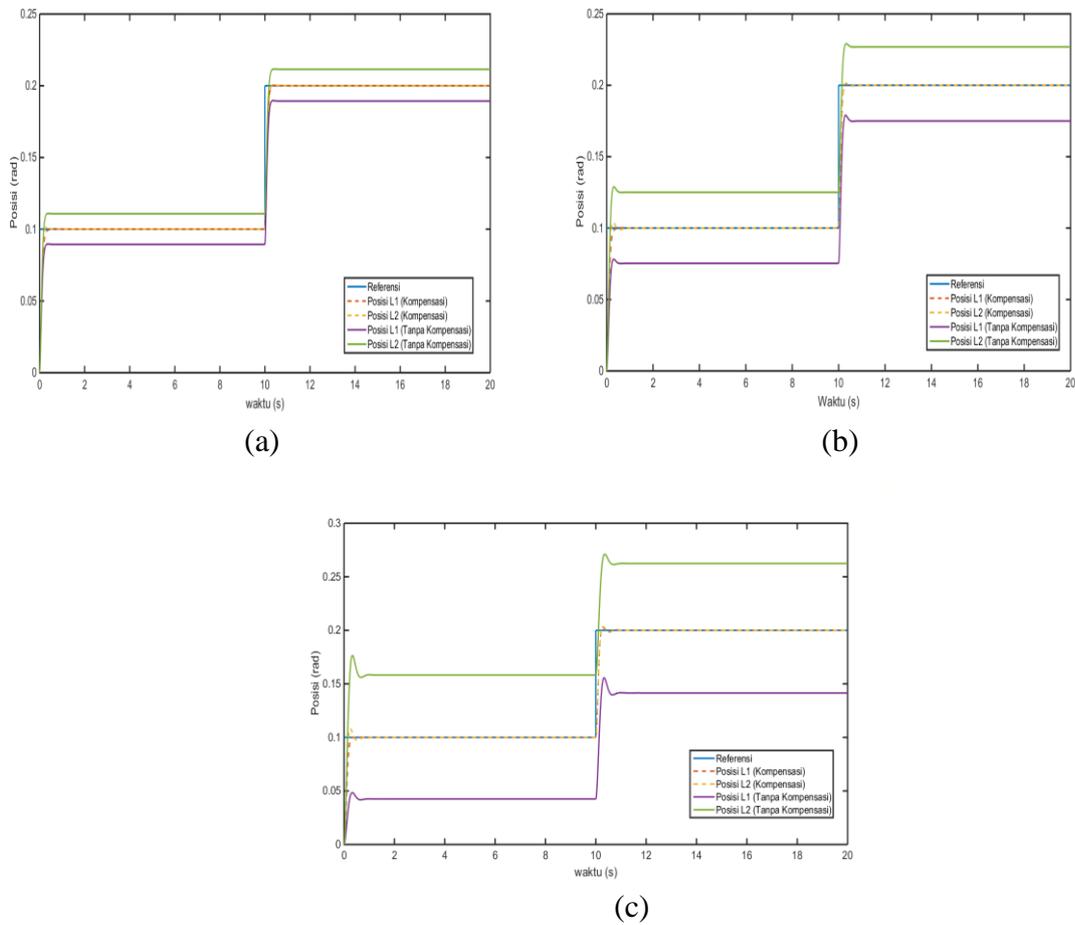
Dari beberapa pengujian yang ada dapat disimpulkan bahwa kompensator mampu bekerja dengan baik untuk mengatasi terjadinya *fault* pada aktuator, hal ini disebabkan karena kompensator mendapatkan *input* dari sinyal residual yang akan diolah menjadi torsi kompensator yang akan selalu ditambahkan dengan torsi nominalnya, sehingga jika torsi nominal mengalami penurunan, maka torsi kompensator akan selalu memberikan tambahan *input* torsi sebesar nilai residual yang diterimanya.



Gambar 4.14 *Tracking* Posisi dengan *Fault* Aktuator dan *Noise* pada *Output*

### 4.3.2 Kompensasi Kesalahan *Multiplicative*

Pengujian terhadap kompensasi kesalahan yang bersifat *multiplicative* akan diberikan dalam bentuk persentase. Beberapa hasil akan ditampilkan, dengan masing-masing persentase *fault* pada aktuatornya adalah sebesar: 30%, 50% dan 70%. Sistem dianggap masih mampu mempertahankan performa sistem jika masih memenuhi kriteria yang telah disebutkan sebelumnya yaitu memiliki *error tracking* tidak lebih dari 0.0082 rad dan memiliki *overshoot* tidak lebih dari 0.05 rad.



Gambar 4.15 Tracking Posisi dengan *Fault* Multiplikatif (a) 30% *Actuator Fault* (b) 50% *Actuator Fault* (c) 70% *Actuator Fault*

Dari data yang telah diperoleh dapat diketahui, amplitudo *overshoot* untuk masing masing pengujian adalah sebagai berikut:

Tabel 4-1 *Amplitudo overshoot* untuk Beberapa Kesalahan Multiplikatif

No	Presentase <i>Fault</i>	<i>overshoot</i>
1	30%	0.003
2	50%	0.005
3	70%	0.008
4	100% ( <i>Failure</i> )	0.02

Dari Tabel 4.1 dan Gambar 4.15 dapat diketahui bahwa kompensator hasil rancangan mampu memperbaiki performa kesalahan aktuator yang bersifat multiplikati, bahkan mampu memperbaiki performa Robot yang *failure*, dimana aktuator tidak dapat berfungsi atau tidak dapat menghasilkan torsi *input* sama

sekali. Berdasarkan Gambar 4.15 dapat ditarik kesimpulan bahwa semakin besar presentase *fault* maka pada respon hasil kompensasi akan terdapat *overshoot* yang semakin besar pula. Adanya *overshoot* diakibatkan karena besarnya pergeseran sinyal dari referensinya, sehingga *input* dari kompensator juga akan sangat besar, hingga menyebabkan torsi kompensator juga akan besar. Hal ini pula yang menyebabkan hasil kompensasi kesalahan aktuator dengan presentase 100% menyebabkan respon posisi mengalami osilasi selama kurang dari 2 detik dengan nilai *overshoot* 0.02 rad, sangat mendekati batas minimal *overshoot* yang diperbolehkan.

#### 4.4 Hasil Estimasi Kesalahan Aktuator

Pada penelitian ini, untuk kasus Robot Manipulator yang hanya mengalami *fault* pada aktuatornya saja, maka sistem selain dapat melakukan kompensasi juga dapat memberikan sinyal hasil estimasi pada kesalahan aktuatornya. Perlu diketahui bahwa estimasi yang dilakukan hanya ketika sistem mengalami *fault* pada aktuatornya saja. Ketika *fault* terjadi bersamaan dengan adanya gangguan ataupun *noise*, *estimator* tidak digunakan untuk mengestimasi keduanya. Karena persamaan yang dirancang hanya digunakan untuk mengestimasi terjadinya *fault* pada aktuator. Namun seperti yang sudah dikatakan sebelumnya kompensasi masih bisa dilakukan meskipun estimasi gangguan dan *noise* tidak dapat dilakukan, karena kompensator pada dasarnya bergantung dari sinyal residual yang terjadi akibat adanya selisih antara sistem sebenarnya dengan sistem yang diharapkan.

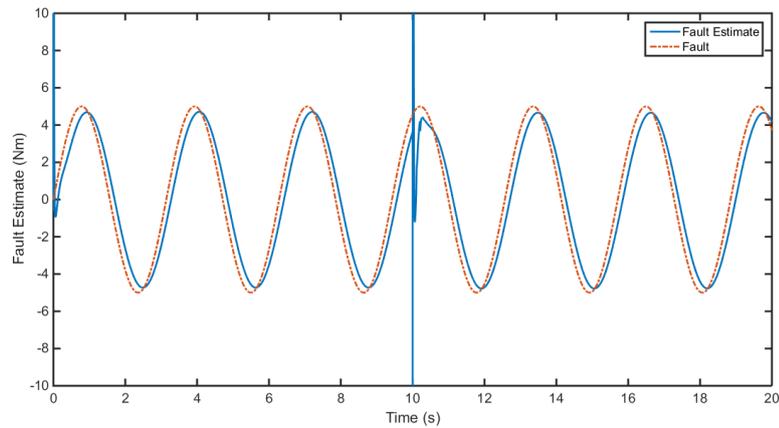
##### 4.4.1 Estimasi Kesalahan *Additive*

Gambar 4.16 merupakan estimasi kesalahan aktuator yang bersifat *additive* dengan jenis kesalahan pada Join-1 dan Join-2 masing-masing adalah:

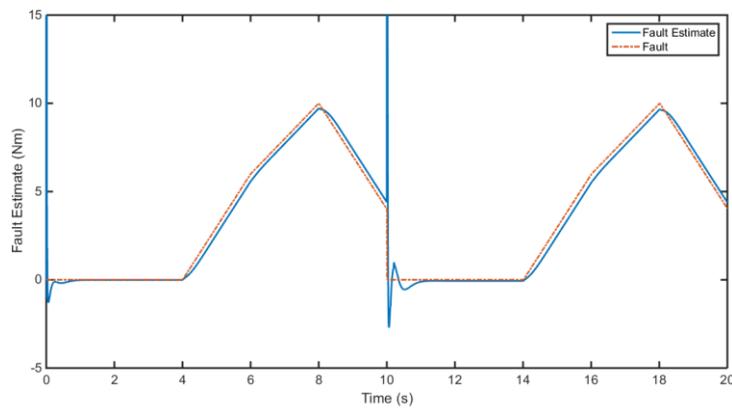
$$f_{a-J1} = 5 * \sin(2t)$$

$$f_{a-J2} = 0 \leq f_a \leq 10 \text{ (Nm)} \text{ untuk } 4 \leq t \leq 10 \text{ dan } 14 \leq t \leq 20$$

Dari kedua Gambar tersebut, dihasilkan estimasi kesalahan aktuator dengan *error* estimasi mencapai 0.2 Nm.



Gambar 4.16 Estimasi Kesalahan Aktuator pada Join-1



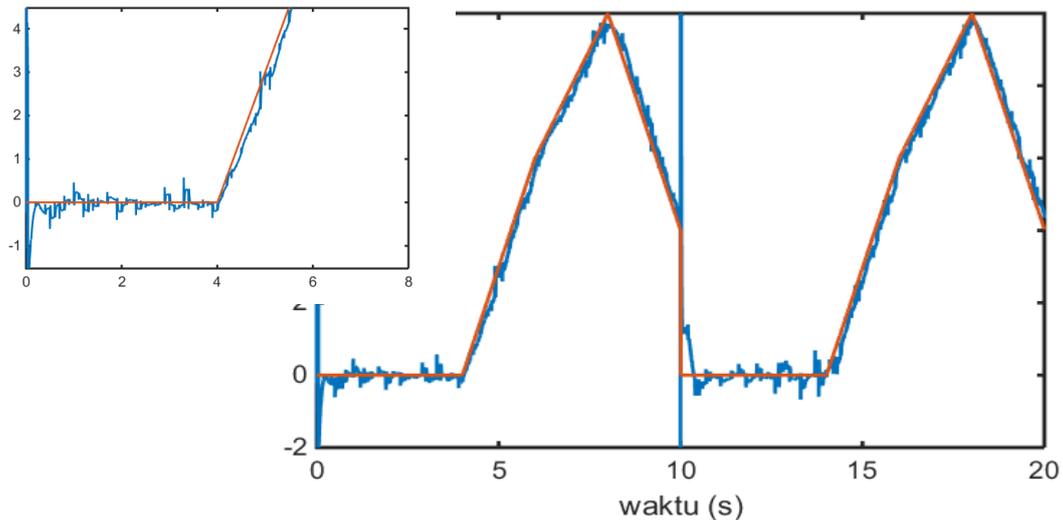
Gambar 4.17 Estimasi Kesalahan Aktuator pada Join-2

#### 4.4.2 Estimasi Kesalahan Aktuator dengan *Noise* pada *Estimator*

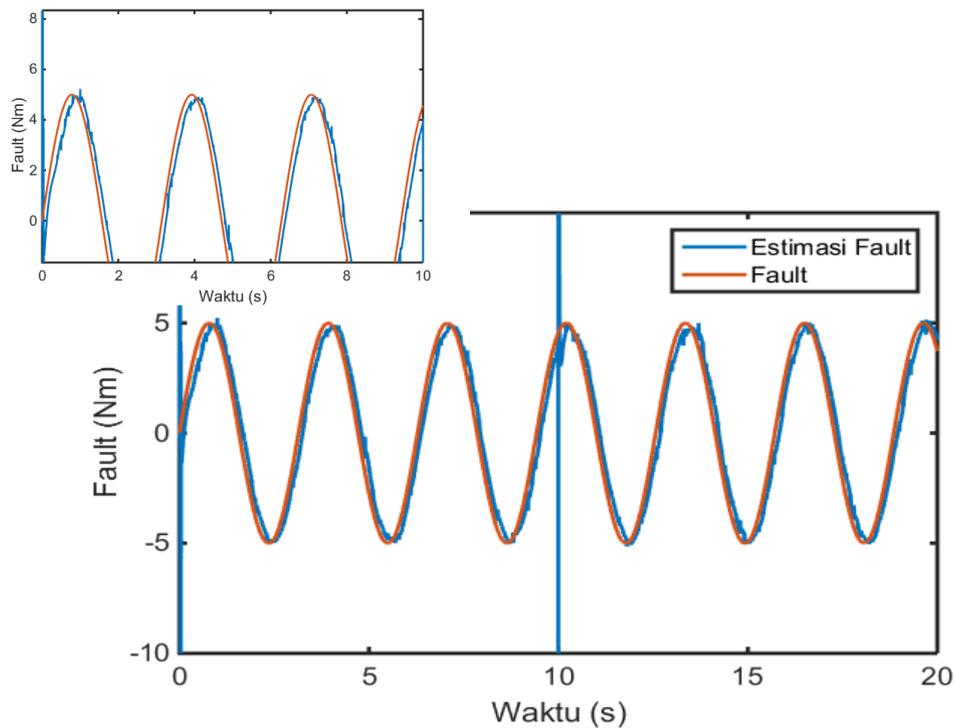
Sesuai dengan penelitian [3] untuk membuktikan kehandalan estimasi kesalahan aktuator maka akan diberikan noise pada estimator dengan *variance* 20% dari nilai nominal *error* estimasinya. Pemberian *noise* pada *estimator* ini menurut [3] juga merupakan pembuktian bahwa hasil rancangan dapat bekerja dengan nilai kesalahan aktuator yang bersifat stokastik. *Noise* yang terjadi berupa Sinyal Gaussian yang bersifat random dengan distribusi normal ( $\mu = 0$  dan  $\sigma = 0.04$ ).

Karena pada penelitian ini, *noise* ( $\omega$ ) terjadi pada *state estimator*, sehingga Persamaan (3.17) akan berubah menjadi:

$$\ddot{r} = A_e r + B_e (f_a + \omega) \quad (4.2)$$



Gambar 4.18 Estimasi Kesalahan Aktuator dengan *Noise* pada *Estimator* (J2)



Gambar 4.19 Estimasi Kesalahan Aktuator dengan *Noise* pada *Estimator* (J1)

Gambar 4.18 dan 4.19 adalah hasil estimasi dari *fault* aktuator dengan *noise* pada estimatornya masing-masing untuk Join-2 dan Join-1. Dari kedua pengujian tersebut didapatkan nilai *error* estimasinya adalah sebesar 0.3 Nm untuk

Join-1 dan 0.5 Nm untuk Join-2. *Noise* pada *estimator* mengakibatkan *error* estimasi membesar 0.1 Nm dan 0.3 Nm dari *error* sebelumnya untuk Join-1 dan Join-2.

#### 4.4.3 Estimasi Kesalahan *Multiplicative*

Pengujian ini merupakan pengujian terakhir yang akan dilakukan dalam penelitian ini. Dalam penelitian ini estimator akan memperkirakan seberapa besar *fault* yang terjadi pada aktuator robot. Dalam pengujian ini, estimator dikatakan berhasil jika estimator mampu memperkirakan seberapa besar nilai range  $\beta$  mempengaruhi nilai torsi nominal. Sehingga data *estimator* akan diberikan dalam bentuk Tabel 4.2. Torsi Nominal yang harus bekerja pada sistem adalah 28.06 Nm untuk Join-1 dan 8.95 Nm untuk Join-2.

Tabel 4-2 Hasil Estimasi *Fault* Aktuator (Multiplikatif)

No	$\beta$	Nilai <i>Fault</i> Seharusnya (Nm)		Hasil Estimasi <i>Fault</i> (Nm)	
		Join-1	Join-2	Join-1	Join-2
1	30%	8.41	2.68	8.59	2.75
2	50%	14.03	4.47	14.2	4.55
3	70%	19.64	6.26	19.82	6.34
<i>Error Estimasi</i>				0.076	0.076

#### 4.5 Pengujian RMSE (*Root Mean Square Error*)

Pengujian RMSE dilakukan untuk mengetahui seberapa jauh respon yang dihasilkan oleh sistem menyimpang dari nilai referensinya.

RMSE dapat dihitung dengan cara:

$$\sqrt{\frac{1}{n} \sum_i^n (q_{di} - q_i)^2} \quad (4.3)$$

dengan  $n, i, q_{di}, q_i$  masing-masing adalah waktu akhir dari simulasi, waktu awal dari simulasi, sinyal referensi ke- $i$  dan sinyal respon ke- $i$ .

Dari Tabel 4-3 dapat diketahui bahwa sistem dapat dikatakan bergeser dari nilai referensinya jika memiliki nilai RMSE  $\geq 0.01$  dan tidak diperbolehkan memiliki *overshoot* lebih dari 0.05 berdasarkan pengujian pada Gambar 4.12.

Tabel 4-3 Tabel RSME Hasil Pengujian

No	Jenis Pengujian	Over- shoot	RMSE		Keterangan
			Join-1	Join-2	
1	Pengujian Kontrol Nominal	0	0.0078	0.0078	
2	Pengujian Kontrol Nominal dengan <i>Fault Additive</i>	0	0.0356	0.0131	Terjadi Pergeseran
3	Pengujian Kontrol Nominal dengan <i>Fault Multiplikatif</i>	0.004	0.0268	0.0267	Terjadi Pergeseran
4	Pengujian Kontrol Nominal dengan <i>Fault (u=PID)</i>	0.118 & 0.104	0.0065	0.0089	<i>Overshoot</i> > 0.09
5	Pengujian MRC	0	0.0082	0.0082	
6	Kompensasi Kesalahan Aktuator				
a	$5 * \sin(2t)$ untuk $0 \leq t \leq 20$	0	0.0081	0.0081	
b	Join – 1 = 14.2 (Nm) Join – 2 = 4.5 (Nm) untuk $0 \leq t \leq 20$	0.007	0.0077	0.0079	
c	$0 \leq f_a \leq 10$ (Nm) untuk $4 \leq t \leq 10$ dan $14 \leq t \leq 20$	0	0.0082	0.0082	
d	jika $u = \text{PID}$	0.047& 0.041	0.0080	0.0079	
7	Kompensasi Kesalahan Aktuator dan <i>Disturbance</i>	0	0.0084	0.0078	
8	Kompensasi Kesalahan Aktuator dan <i>Noise</i> pada <i>Output</i>	0	0.0098	0.0098	
9	Kompensasi Kesalahan Multiplikatif				
a	<i>Fault</i> 30%	0.003	0.0083	0.0081	
b	<i>Fault</i> 50%	0.005	0.0085	0.0082	
c	<i>Fault</i> 70%	0.008	0.0088	0.0083	
d	<i>Fault</i> 100% ( <i>Failure</i> )	0.02	0.0095	0.0088	

Dari tabel tersebut juga dapat disimpulkan bahwa rancangan hasil kompensator dapat melakukan *tracking* dengan baik, dan meminimalisir *overshoot*. Kelebihan dari rancangan kompensator berdasarkan MRC ini adalah kompensator mampu melakukan pemulihan terhadap *fault* pada aktuator yang mengalami *failure*. Hal ini disebabkan karena kompensator bekerja berdasarkan pembacaan sinyal *error*, semakin besar sinyal *error* maka akan semakin besar torsi kompensator yang diberikan bahkan untuk menggantikan torsi yang tidak bekerja.

## **BAB 5**

### **KESIMPULAN**

Berdasarkan hasil pengujian dan Analisis pada bab sebelumnya, maka dapat disimpulkan bahwa:

1. Rancangan desain kompensator dapat melakukan kompensasi kesalahan aktuator yang bersifat *additive* maupun multiplikatif pada Robot Manipulator 2 DOF, dengan *error tracking* (RMSE) dari 0.0356 menjadi 0.0077 setelah dikompensasi, dan mampu meminimalisir *overshoot* dari 0.11 menjadi 0.047 untuk Join-1 dan dari 0.1 menjadi 0.041 untuk Join-2 setelah dikompensasi.
2. Pada penelitian ini, rancangan desain kompensator juga dapat digunakan untuk mengkompensasi adanya gangguan atau *noise* pada *output* yang terjadi bersamaan dengan terjadinya *fault* pada aktuator.
3. Hasil rancangan desain kompensator bergantung dari nilai residual, sehingga dalam penelitian ini, meskipun *estimator* tidak memberikan hasil yang tepat, kompensator masih bisa bekerja selama sistem masih mendeteksi adanya sinyal residual.

*Halaman ini sengaja dikosongkan*

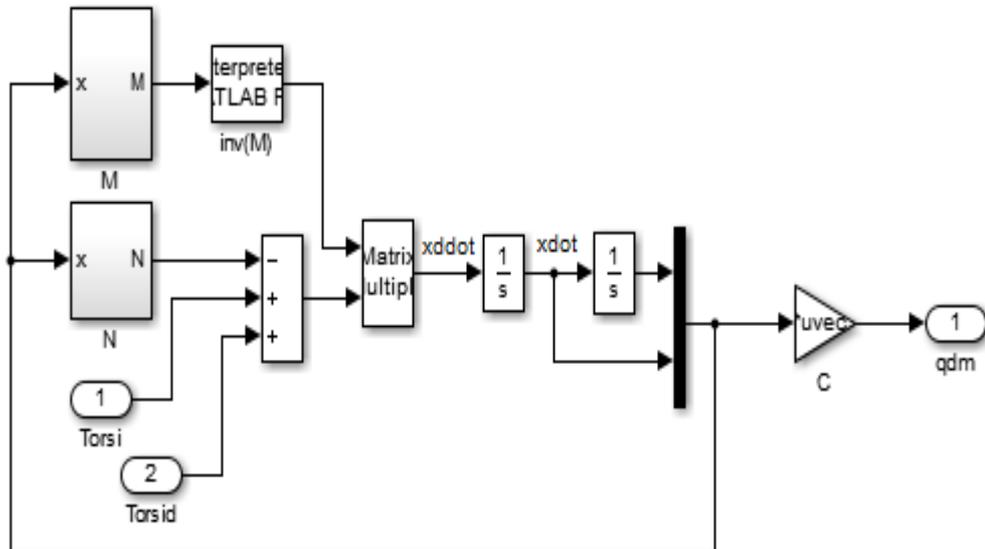
## DAFTAR PUSTAKA

- [1] E. Pitowarno, (2006), “Robotika: Desain, Kontrol, dan Kecerdasan Buatan”, Andi Yogyakarta.
- [2] L. L. Frank, M. D. Darren. T. A. Chaouki. (2004), “Robot Manipulator Control Theory and Practice”, MARCEL DEKKER, Inc: Basel, New York, USA.
- [3] M. Giuseppe and Francesco Pierri, “A Fault Tolerant Adaptive Control for Robot Manipulators” *International Conference on Robotics and Biomimetic*, Guangzhou, China, 2012.
- [4] D. Tan and Ron J. Patton, “Integrated Fault Estimation and Fault Tolerant Control: A Joint Design”, IFAC Paper Online, pages 517-522, 2015.
- [5] C. Nie dan R. J. Patton, “Fault Eestimation and MRC based Active FTC”. IFAC World Congress: Italy, August 28—September 2, 2011.
- [6] T. Andriani. (2015), “Desain dan Simulasi Deteksi dan Isolasi Kesalahan berbasis *PD Descriptor Observer* Menggunakan *CTC Law* pada Robot Manipulator PUMA 6 DOF”, Thesis, Institut Teknologi Sepuluh Nopember: Surabaya-Indonesia.
- [7] B. Xiao, S. Yin, “A Fast Reconstruction Approach for Actuator Fault in Robot Manipulator ”, *IEEE 14<sup>th</sup> International Workshop on Variable Structure Systems (VSS)*, 2016.
- [8] M. J. Hong, H. Y. Guang, “Simultaneous Fault Diagnosis for Robot Manipulator with Actuator and Sensor Faults”, *ELSEVIER Information Science*, vol. 366, pp. 12-3, 2016.
- [9] D. Brambilla, M. C. Luca. F. Antonella, P. Pierluigi, “Actuators and Sensors Fault Detection for Robot Manipulators Second Order Sliding Mode Observer”, *IEEE International Workshop on Variable Structure Systems*, 2005.
- [10] A. De Luca and M. Raffaella., “An Identification Scheme for Robot Actuator Faults”, *IEEE Intelligent Robots and Systems*, pp. 1127-1131, 2005.
- [11] Shi, Fengming. (2013), “Observer based Active FTC of Descriptor System”, Thesis for the Degree of Philosophy, University of Hull: UK.
- [12] K. Indriawati, (2016), “Strategi Sensor dan Aktuator *Fault Tolerant Tracking Control* Berbasis *Robust Extended State Observer*”, Disertasi, Institut Teknologi Sepuluh Nopember: Surabaya-Indonesia.
- [13] R. J. Patton, “Fault Tolerant Control: The 1997 Situation”, *IFAC Safeprocess*, 1997.
- [14] H. Alwi et al., “Fault Detection and Fault Tolerant Control Using Sliding Modes”, *SPRINGER Advances in Industrial Control*: London, 2011.

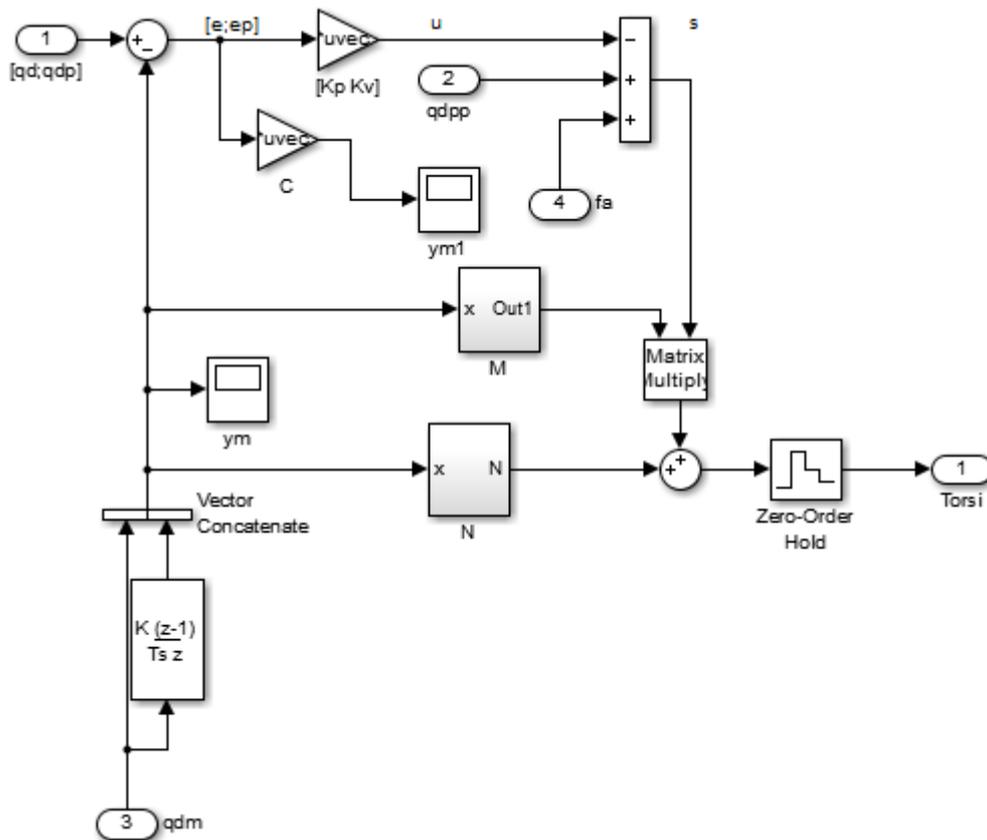
*Halaman ini sengaja dikosongkan*

## LAMPIRAN

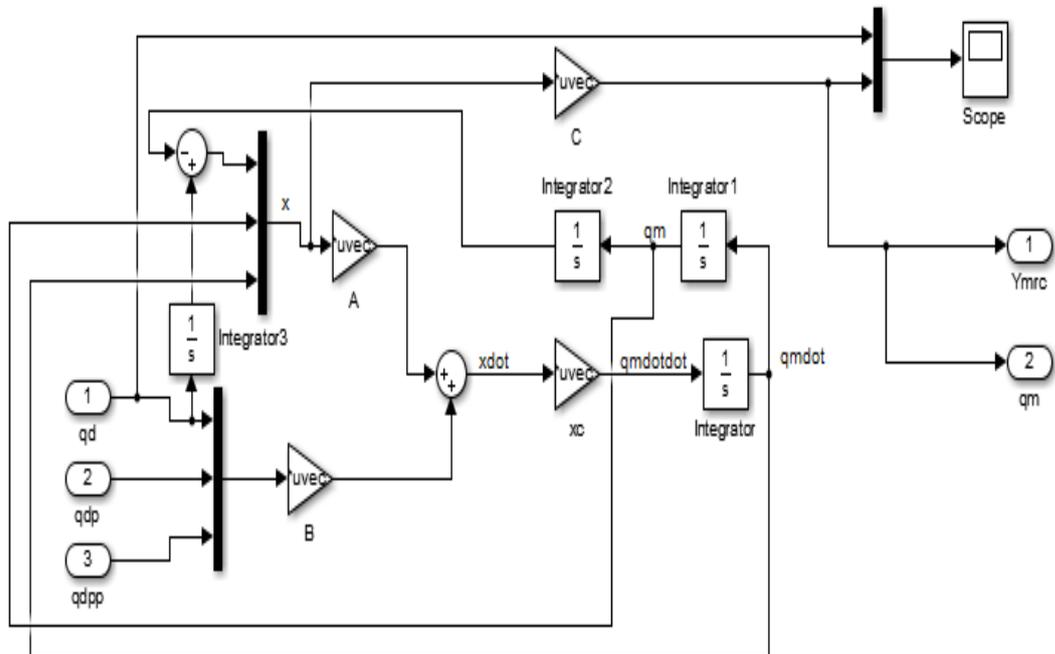
### Pemodelan Robot



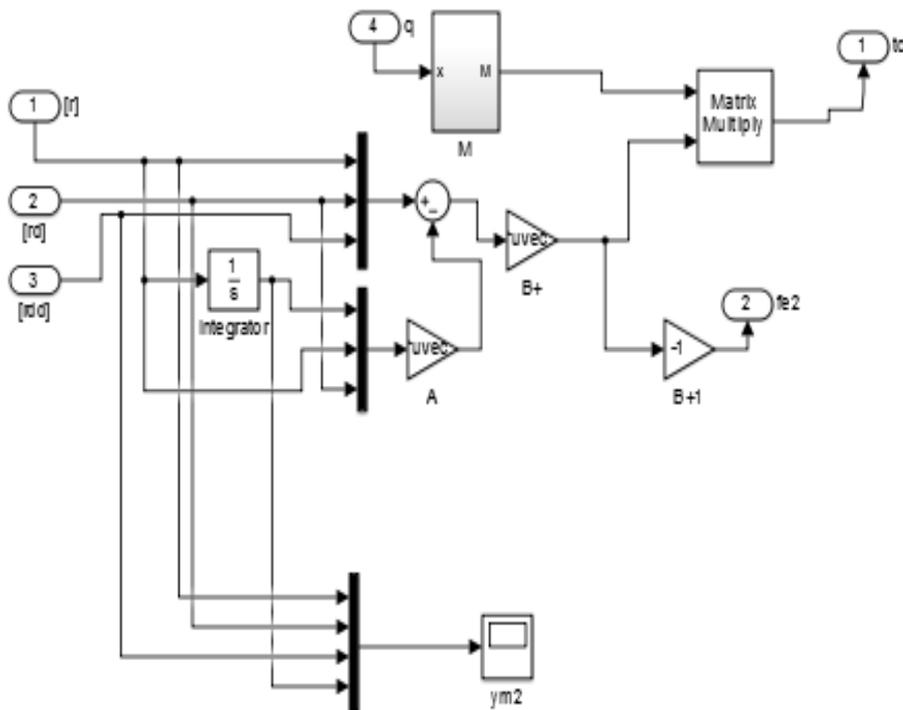
### Pemodelan Kontroler Nominal



## Pemodelan MRC



## Pemodelan Estimator dan Kompensator



## Nilai Skalar dan Matriks pada M-File

```
m1=1;
m2=1;
a1=1;
a2=1;
g = 9.8;
g1=0.1;
g2=0.1;
T = 5;
fact = 2*pi/T;
Kp = 300*eye(2);
Kv = 30*eye(2);
Ki = 1*eye(2);
Kir =4000*eye(2);
A = [zeros(2) -eye(2) zeros(2); zeros(2) zeros(2) eye(2);
Ki -Kp -Kv];
B = [eye(2) zeros(2) zeros(2); zeros(2) zeros(2) zeros(2);
Kp Kv eye(2)];
C = [zeros(2) eye(2) zeros(2)];
xc =[zeros(2) zeros(2) eye(2)];
F = [zeros(2) zeros(2) eye(2)];
```

*Halaman ini sengaja dikosongkan*

## BIODATA PENULIS



**Churnia Sari** dilahirkan di Ngawi pada Tanggal 8 Desember 1990. Putri kedua dari Bapak Maryono dan Ibu Pasetyanik, istri dari Bapak Ogi T. Maroghi sekaligus ibu dari Aya Sofia. Penulis memulai Pendidikan Perguruan Tinggi pertama kali di Politeknik Elektronika Negeri Surabaya untuk Program Diploma Jurusan Teknik Elektronika pada tahun 2009. Merasa kurang puas dengan ilmunya, akhirnya penulis memutuskan untuk menimba ilmu kembali pada Program Lintas Jalur di Jurusan Teknik Elektro-Institut Teknologi Sepuluh Nopember (ITS) Surabaya (pada saat itu belum menjadi Fakultas, tahun 2012). Bidang Studi yang

ditekuni oleh penulis adalah Teknik Sistem Pengaturan karena dianggap lebih bersahabat untuk dipelajari. Pada tahun 2014 penulis menyelesaikan Pendidikan Sarjananya, kemudian melanjutkan Pendidikan Masternya pada bidang keahlian yang sama di almamater yang sama pula. Penulis menyelesaikan studi Masternya pada tahun 2017 dari Fakultas Teknologi Elektro - ITS Surabaya.