



TESIS - TE142599

DESAIN KONTROL *PATH FOLLOWING*  
QUADCOPTER MENGGUNAKAN *OUTPUT*  
*FEEDBACK* DENGAN *COMMAND GENERATOR*  
*TRACKER MODEL FOLLOWING*

ANGGARA TRISNA NUGRAHA  
2215202008

DOSEN PEMBIMBING  
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 KONTROL *PATH FOLLOWING*  
QUADCOPTER DENGAN *COMMAND GENERATOR*  
*TRACKER MODEL FOLLOWING*

ANGGARA TRISNA NUGRAHA  
2215202008

DOSEN PEMBIMBING  
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

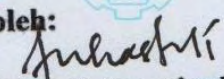


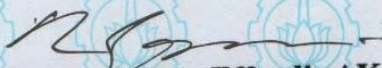
Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Teknik (M.T)  
di  
Institut Teknologi Sepuluh Nopember

oleh:

Anggara Trisna Nugraha  
NRP. 2215202008

Tanggal Ujian : 09 Juni 2017  
Periode Wisuda : September 2017

Disetujui oleh:

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

Dekan Fakultas Teknologi Elektro



  
Dr. Tri Arief 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 KONTROL *PATH FOLLOWING* QUADCOPTER MENGGUNAKAN *OUTPUT FEEDBACK* DENGAN *COMMAND GENERATOR TRACKER MODEL FOLLOWING*” 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, 26 Juli 2017

Anggara Trisna Nugraha

NRP. 2215202008

*--Halaman ini sengaja dikosongkan--*



# DESAIN KONTROL *PATH FOLLOWING* QUADCOPTER DENGAN *COMMAND GENERATOR TRACKER MODEL FOLLOWING*

Nama mahasiswa : Anggara Trisna Nugraha  
NRP : 2215202008  
Pembimbing : Dr. Trihastuti Agustinah, ST., MT.

## ABSTRAK

Quadcopter merupakan salah satu jenis Unmanned Aerial Vehicle (UAV) yang banyak digunakan sebagai objek penelitian saat ini. Dengan empat rotor sebagai penggerak utama, quadcopter mampu bergerak secara rotasional dan translasional. Kedua sistem gerak ini secara matematis tergolong nonlinier dan tidak stabil. Dalam penelitian ini, dirancang desain kontrol *path following quadcopter* menggunakan *output feedback* dengan *command-generator tracker model following* yang dapat melakukan *path following* terhadap *path* lingkaran walau terdapat gangguan eksternal. Sistem quadcopter dibagi berdasarkan sistem geraknya, yaitu sistem gerak rotasi dan sistem gerak translasi. Masing-masing sistem gerak dikendalikan dengan kontroler yang berbeda.

Sebagai *inner-loop* dalam sistem quadcopter, sistem gerak rotasi dituntut memiliki *settling time* lebih cepat dari sistem gerak translasi yang merupakan *outer loop*. Nilai penguat kontroler yang mampu menjamin kestabilan sistem dan memenuhi performa  $H_\infty$  diperoleh dengan bantuan Linear Matrix Inequality (LMI).

Permasalahan *path following* dalam sistem gerak translasi diatasi dengan menggunakan *command-generator tracker model following*. Sistem dikombinasikan dengan algoritma *line of sight* yang mana algoritma tersebut untuk menjaga arah hadap dari quadcopter serta mampu mengatasi adanya gangguan eksternal.

Hasil simulasi menunjukkan bahwa metode kontrol yang digunakan mampu membawa sudut *yaw* pada nilai yang diharapkan dengan nilai ISE sebesar 0.024 Rad dan pada saat terdapat gangguan sebesar 0.0903 Rad. Quadcopter dapat melakukan *path following* secara otomatis dengan nilai maksimum ISE sebesar 0.001 m. Selain itu, sistem kontrol juga mampu mengatasi gangguan eksternal berupa *wind gust* dengan penyimpangan posisi *X* maksimum mencapai 0.013168 m serta pada posisi *Y* sebesar 0.0003 m. Kontroler ini mampu mengatasi *delay* sebesar  $\pm 3.17$  detik serta mampu melakukan *path following* pada lintasan persegi dan spiral (*helix*).

Kata kunci: Quadcopter, LOS, Command-Generator Tracker Model *Following*, LQ regulator

*--Halaman ini sengaja dikosongkan--*

# **QUADCOPTER *PATH FOLLOWING CONTROL DESIGN* WITH COMMAND GENERATOR TRACKER MODEL *FOLLOWING***

By : Anggara Trisna Nugraha  
Student Identity Number : 2215202008  
Supervisor : Dr. Trihastuti Agustinah, ST., MT.

## **ABSTRACT**

Quadcopter is one type of UAV that is widely used as the object of current research. With four main drive rotors, quadcopter is able to move rotationally and translationally. Both motions are mathematically classified as non-linear and unstable. In this study, designed the path following quadcopter control using output feedback with command-generator tracker model following which can do path following to the circle path although there are external troubles. Quadcopter system divided by the movement system, which is rotation move system and translation move system. Each of them are controlled with different controller.

As an inner-loop in quadcopter system, rotation movement system is required to have settling time faster than translation movement system which using outer loop. The value of controller amplifier which can guarantee the stability of the system and comply the performances of  $H_\infty$  obtained with help of the matrix inequality (LMI).

The path following problems in translation movement system are solved with using command-generator tracker model following. The system is combinationed with line of sight algorithm, which is that algorithm used to keep the headings of quadcopter and also can conquer the external problems.

The result of the simulation shows that control method which used can make yaw angle in to the expected value according to the ISE value which is 0.024 Rad, and when there's a trouble which is 0.0903 Rad. Quadcopter can do path following automatically with the maximum of ISE value which 0.001 m. Other than that, the control system also can handle the external problems such as wind gust with the deviation of  $X$  maximum position reach out 0.013168 and in  $Y$  position reach out 0.0003 m. This controller can overcome the delay in amount of +/- 3.17 second and also can do path following in square and helix path.

Key words: Quadcopter, LOS, Command-Generator Tracker Model *Following*, LQ regulator

*--Halaman ini sengaja dikosongkan--*

## KATA PENGANTAR

Alhamdulillahirabbil'alamin,

Segala puji bagi Allah SWT, pemilik semesta beserta segala pengetahuannya. Tesis dengan judul

***Desain Kontrol Path following Quadcopter Menggunakan Output Feedback Dengan Command Generator Tracker Model Following***

ini digunakan sebagai persyaratan guna menyelesaikan pendidikan Magister pada Bidang Keahlian Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Ucapan terimakasih penulis sampaikan kepada ibunda Ismatul Mas'adah dan ayahanda Syaiful Anam serta adikku Andalusia Trisna Salsabila atas kepercayaan, dukungan, dan doa-doa indah yang selalu menyertai, kepada Ibu Dr. Trihastuti Agustinah ST., MT., atas bimbingan, didikan dan motivasi selama penulis menempuh studi hingga terselesaikannya Tesis ini, kepada tim riset quadcopter dan untuk teman-teman S2 (teman seperjuangan meraih mimpi)..

*"Kamu baru boleh mengeluh setelah kamu mengerahkan seluruh kemampuan mu. Daripada kamu kecewa dan menyerah, hanya karena dirimu tidak jenius, lebih baik kamu percaya kalau kemampuan mu tak hanya sebatas ini"*

Ungkapan tersebut menjadi dasar yang tersirat dalam penyusunan Tesis ini, semoga dapat memberi manfaat bagi pembaca dan terutama bagi penulis.

Surabaya, 26 Juli 2017

Penulis

*--Halaman ini sengaja dikosongkan--*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	iii
PERNYATAAN KEASLIAN TESIS .....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xxi
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Kontribusi .....	3
1.6 Metodologi Penelitian .....	3
1.7 Posisi dan Road Map Penelitian .....	4
1.7.1 Penelitian Sebelumnya berdasarkan kajian pustaka .....	4
1.7.2 Road Map Penelitian .....	9
1.7.3 Penelitian Sebelumnya yang Telah Dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan. 10	
BAB 2 KAJIAN PUSTAKA .....	17
2.1 Kajian Penelitian Terkait .....	17
2.1.1 A Globally Stabilizing <i>Path following Controller</i> for Rotorcraft with Wind <i>Disturbance Rejection</i> [1] .....	17
2.1.2 <i>Optimal Path Tracking Control</i> of a Quadcopter UAV [2] .....	18
2.1.3 Simulation of the Quadcopter <i>Controller</i> with LQR Integral Effect [3] .....	20
2.1.4 Takagi-Sugeno Model for Quadcopter Modelling and <i>Control</i> Using Non linear State <i>feedback Controller</i> [4] .....	21
2.1.5 Kontrol <i>Tracking</i> untuk Quadrotor Menggunakan Metode Optimal Output <i>Feedback</i> dengan Command Generator Tracker [5] .....	23

2.1.6	Design of $H^\infty$ Command and Control Loops for Unmanned Aerial Vehicles using Static Output-Feedback [6]	26
2.2	Dasar Teori	27
2.2.1	Quadcopter (Quanser Qball X4)	27
2.2.2	Pemodelan Quadcopter	29
2.2.3	Linearisasi	38
2.2.4	Analisa Kestabilan	40
2.3	Kontroler	42
2.3.1	Linear Matrix Inequality (LMI)	42
2.3.2	Kontrol Static Output Feedback dengan performa $H^\infty$	43
2.3.3	Command Generator Tracking dengan Model Following	44
2.3.4	Blok Koreksi	48
2.3.5	Path following	48
BAB 3 METODOLOGI PENELITIAN		53
3.1	Linearisasi Model Dinamika Quadcopter	53
3.1.1	Parameter Quadcopter	53
3.1.2	Pemodelan Linearisasi sudut Roll dan Pitch	54
3.2	Diagram blok quadcopter	57
3.3	Perancangan Kontroler	58
3.3.1	Prosedur Perancangan Sistem	58
3.4	Perancangan Kontroler Outer loop (Posisi)	60
3.4.1	Generate Path dan Guidance	60
3.4.2	Perancangan Kontroler posisi pada sumbu $X$ dan $Y$	62
3.5	Blok Koreksi	67
3.6	Perancangan Kontroler Inner loop (Rotasi)	68
3.7	Pemodelan Efek Hembusan Angin (Wind gust)	72
3.7.1	Pemodelan Disturbance Kecepatan Angin Konstan (Step)	72
3.7.2	Pemodelan Disturbance Kecepatan Angin Gust	73
BAB 4 HASIL DAN PEMBAHASAN		75
4.1	Hasil Perancangan Sistem Kontrol	75
4.2	Pengujian Sistem Kontrol Quadcopter	76
4.2.1	Pengujian Sistem Kontrol Quadcopter Tanpa Adanya Gangguan	76
4.2.2	Pengujian Sistem Kontrol Quadcopter dengan Efek Wind gust	81



4.3	Pengujian Sistem Kontrol Terhadap <i>Noise</i> Pengukuran.....	96
4.4	Pengujian Pelemahan Terhadap Gangguan ( <i>Disturbance</i> ).....	100
4.5	Simulasi dengan Sinyal Referensi yang Berbeda-beda .....	114
4.6	Perbandingan Kontroler <i>Output Feedback Command-Generator Tracker</i> (CGT) dengan Hasil Desain.....	116
4.6.1	Pengujian Lintasan Lingkaran .....	116
4.6.2	Pengujian Lintasan Persegi .....	122
BAB 5 KESIMPULAN.....		129
5.1	Kesimpulan.....	129
5.2	Saran .....	129
DAFTAR PUSTAKA .....		131
LAMPIRAN.....		133
BIODATA PENULIS .....		151

*--Halaman ini sengaja dikosongkan--*

## DAFTAR GAMBAR

Gambar 1.1. Peta jalan penelitian .....	9
Gambar 2.1 a. Respon sistem saat melakukan <i>path following</i> dan b. Respon posisi $z$ saat diberi <i>Disturbance</i> .....	18
Gambar 2.2 Diagram blok <i>Linear Quadratic Tracking</i> dengan waktu diskrit.....	19
Gambar 2.3a. Respon sistem <i>path tracking</i> ( $x, y, z$ ) dengan tanpa gangguan dan b. Respon sistem <i>path tracking</i> ( $x, y, z$ ) dengan adanya gangguan .....	19
Gambar 2.4 a. Struktur <i>Linear Quadratic Regulator</i> dengan integral dan b. Respon <i>attitude</i> dan <i>altitude</i> dari quadcopter.....	20
Gambar 2.5(a) Posisi 3D untuk $K_{stabilisasi}$ , (b) Posisi 3D untuk $K_{placement}$ .....	22
Gambar 2.6 Respon <i>attitude</i> quadcopter ( $\phi, \theta, \psi, x, y, z$ ) .....	23
Gambar 2.7 Diagram blok kontroler <i>outer loop</i> [19].....	24
Gambar 2.8a. Respon jika dilihat pada bidang $X$ dan $Y$ dimana lintasan berbentuk lingkaran dan b. Respon <i>tracking</i> lingkaran terhadap efek <i>wind gust</i> .....	25
Gambar 2.9 Respon <i>tracking</i> posisi pada sumbu $X$ dan $Y$ .....	25
Gambar 2.10 Respon <i>tracking</i> posisi pada sumbu $X$ dan $Y$ terhadap efek <i>wind gust</i> .....	26
Gambar 2.11a. Struktur kontroler <i>inner</i> dan <i>outer loop</i> dan b. Respon posisi terhadap <i>command</i> yang diberikan.....	27
Gambar 2.12 <i>Thrust</i> ( $U_1$ ), <i>B. Roll</i> ( $U_2$ ), <i>C. Pitch</i> ( $U_3$ ), dan <i>D. Yaw</i> ( $U_4$ ).....	28
Gambar 2.13 Gambar Semua <i>State</i> dari Quadcopter (e untuk <i>Frame</i> Bumi dan b untuk <i>Frame</i> Badan Quadcopter).....	30
Gambar 2.14a. <i>Guidance Geometri</i> [13] dan b. Referensi sudut <i>course</i> $\chi_d$ pada LOS <i>guidance</i> [17].....	49
Gambar 3.1 Diagram blok sistem kontrol untuk quadcopter .....	57
Gambar 3.2 Kontrol <i>Inner loop</i> .....	59
Gambar 3.3 Kontrol <i>outer loop</i> .....	60
Gambar 3.4 Diagram simulink kontroler untuk posisi $X$ .....	62
Gambar 3.5 Diagram simulink kontroler untuk posisi $Y$ .....	65
Gambar 3.6 Bode diagram respon frekuensi dari $\ T_{zw}(s)\ _{\infty}$ .....	71
Gambar 3.7 Perubahan kecepatan angin <i>step</i> .....	73
Gambar 3.8 Perubahan kecepatan angin ' <i>gust</i> '.....	73
Gambar 3.9 Diagram blok untuk simulasi dengan gangguan .....	74
Gambar 4.1 <i>Path</i> lingkaran .....	77
Gambar 4.2 Sinyal referensi posisi pada sumbu $X$ dan $Y$ .....	77
Gambar 4.3 Referensi lintasan dalam bidang $X, Y$ , dan $Z$ .....	77
Gambar 4.4 Respon referensi dan aktual <i>path following</i> pada sumbu $X, Y$ .....	78
Gambar 4.5 Respon sistem <i>path following</i> pada sumbu $X$ .....	78
Gambar 4.6 Respon Sistem <i>path following</i> pada sumbu $Y$ .....	79
Gambar 4.7 Respon sistem <i>path following</i> pada sumbu $Z$ .....	79

Gambar 4.8 Hasil <i>sniping tool</i> dari <i>virtual world quadcopter</i> .....	79
Gambar 4.9 Respon sistem sudut <i>roll</i> .....	80
Gambar 4.10 Respon sistem sudut <i>pitch</i> .....	80
Gambar 4.11 Respon sistem sudut <i>yaw</i> dengan algoritma LOS.....	80
Gambar 4.12 a. <i>Wind gust</i> secara keseluruhan, b. pemberian gangguan pada detik ke-15 hingga detik ke-40 dan c. <i>Wind gust</i> pada sistem.....	82
Gambar 4.13 <i>Path following</i> dengan gangguan dalam sumbu $X, Y$ .....	83
Gambar 4.14 <i>Path following</i> sumbu $X$ (m) pada saat terdapat angin konstan .....	83
Gambar 4.15 <i>Path following</i> sumbu $Y$ (m) pada saat terdapat angin konstan .....	84
Gambar 4.16 <i>Path following</i> sudut <i>roll</i> pada saat terdapat angin konstan .....	85
Gambar 4.17 <i>Path following</i> sudut <i>pitch</i> pada saat terdapat angin konstan .....	85
Gambar 4.18 <i>Path following</i> dengan gangguan dalam sumbu $X, Y$ .....	86
Gambar 4.19 <i>Path following</i> Sumbu $X$ (m) pada saat terdapat <i>gust</i> .....	86
Gambar 4.20 <i>Path following</i> Sumbu $Y$ (m) pada saat terdapat <i>gust</i> .....	87
Gambar 4.21 <i>Path following</i> sudut <i>pitch</i> pada saat terdapat <i>wind gust</i> .....	88
Gambar 4.22 <i>Path following</i> sudut <i>roll</i> pada saat terdapat <i>gust</i> .....	88
Gambar 4.23 <i>Path following</i> dengan gangguan dalam sumbu $X, Y$ .....	89
Gambar 4.24 <i>Path following</i> Sumbu $X$ (m) pada saat terdapat <i>wind gust</i> .....	89
Gambar 4.25 <i>Path following</i> Sumbu $Y$ (m) pada saat terdapat <i>wind gust</i> .....	90
Gambar 4.26 <i>Path following</i> Sudut <i>Roll</i> pada saat terdapat <i>wind gust</i> .....	91
Gambar 4.27 <i>Path following</i> Sudut <i>Pitch</i> pada saat terdapat <i>wind gust</i> .....	91
Gambar 4.28 <i>Path following</i> Sudut <i>Yaw</i> pada saat terdapat <i>wind gust</i> .....	91
Gambar 4.29 <i>Sniping Tool Virtual World</i> saat simulasi dengan efek <i>wind gust</i> 15-40(s).....	92
Gambar 4.30 <i>Path following</i> dengan gangguan dalam sumbu $X, Y$ .....	93
Gambar 4.31 <i>Path following</i> Sumbu $X$ (m) pada saat terdapat <i>wind gust</i> .....	93
Gambar 4.32 <i>Path following</i> Sumbu $Y$ (m) pada saat terdapat <i>wind gust</i> .....	94
Gambar 4.33 <i>Path following</i> Sudut <i>Roll</i> pada saat terdapat <i>wind gust</i> .....	94
Gambar 4.34 <i>Path following</i> Sudut <i>Pitch</i> pada saat terdapat <i>wind gust</i> .....	95
Gambar 4.35 <i>Path following</i> Sudut <i>Yaw</i> pada saat terdapat <i>wind gust</i> .....	95
Gambar 4.36 <i>Sniping Tool Virtual World</i> saat simulasi dengan efek <i>wind gust</i> 0-62.5(s).....	95
Gambar 4.37 Sinyal random Gaussian dengan $N(0,0.01)$ .....	96
Gambar 4.38 Posisi $X$ dan sudut <i>roll</i> saat diberi <i>noise</i> $N(0,0.01)$ .....	97
Gambar 4.39 Posisi $Y$ dan sudut <i>roll</i> saat diberi <i>noise</i> $N(0,0.01)$ .....	98
Gambar 4.40 Sudut <i>yaw</i> saat diberi <i>noise</i> $N(0,0.01)$ .....	98
Gambar 4.41 Respon <i>path following</i> lintasan lingkaran pada bidang $X, Y$ .....	99
Gambar 4.42 <i>Virtual World</i> dari quadcopter saat simulasi diberi efek <i>noise</i> .....	100
Gambar 4.43a. Respon posisi $X$ , b. Respon sudut <i>roll</i> , dan c. Respon kecepatan sumbu $X$ dengan berbagai nilai $\gamma$ saat <i>step</i> .....	101

Gambar 4.44 Respon posisi $Y$ , b. Respon sudut <i>Roll</i> , dan c. Respon kecepatan sumbu $Y$ dengan berbagai nilai $\gamma$ saat <i>step</i> .....	102
Gambar 4.45 Respon posisi $X$ , b. Respon sudut <i>roll</i> , dan c. Respon kecepatan sumbu $X$ dengan berbagai nilai $\gamma$ .....	105
Gambar 4.46 Respon posisi $Y$ , b. Respon sudut <i>Roll</i> , dan c. Respon kecepatan sumbu $Y$ dengan berbagai nilai $\gamma$ .....	106
Gambar 4.47 Respon posisi $X$ , b. Respon sudut <i>roll</i> , dan c. Respon kecepatan sumbu $X$ dengan berbagai nilai $\gamma$ saat <i>Step+Gust</i> 15-40(s) .....	109
Gambar 4.48 Respon posisi $Y$ , b. Respon sudut <i>Roll</i> , dan c. Respon kecepatan sumbu $Y$ dengan berbagai nilai $\gamma$ saat <i>Step+Gust</i> 15-40(s) .....	110
Gambar 4.49 Respon posisi $X$ , b. Respon sudut <i>roll</i> , dan c. Respon kecepatan sumbu $X$ dengan berbagai nilai $\gamma$ dengan <i>Disturbance Step+Gust</i> 0-62.5(s) .....	112
Gambar 4.50 Respon posisi $Y$ , b. Respon sudut <i>Roll</i> , dan c. Respon kecepatan sumbu $Y$ dengan berbagai nilai $\gamma$ dengan <i>Disturbance Step+Gust</i> 0-62.5(s) .....	113
Gambar 4.51 <i>Path following</i> Posisi dengan Lintasan Persegi.....	115
Gambar 4.52 Posisi X dan Y Pada Saat <i>Path following</i> Lintasan Berbentuk Persegi .....	115
Gambar 4.53 <i>Path following</i> Posisi dengan Lintasan Spiral ( <i>Helix</i> ) .....	116
Gambar 4.54 <i>Tracking</i> dan <i>path following</i> Lintasan Lingkaran tanpa gangguan	117
Gambar 4.55 Perbandingan dalam posisi sumbu X(m) .....	117
Gambar 4.56 Waktu <i>delay</i> yang dibutuhkan kontroler CGT-MF .....	117
Gambar 4.57 Waktu <i>delay</i> yang dibutuhkan kontroler CGT .....	118
Gambar 4.58 a. <i>Wind gust</i> secara keseluruhan, b. pemberian gangguan pada detik ke-15 hingga detik ke-40 dan c. <i>Wind gust</i> pada sistem .....	119
Gambar 4.59 Respon quadcopter pada sumbu X dan Y dengan gangguan .....	119
Gambar 4.60 Perbandingan respon posisi X dengan gangguan .....	120
Gambar 4.61 Perbandingan respon posisi Y dengan gangguan .....	120
Gambar 4.62 Perbandingan respon sudut <i>Roll</i> dengan gangguan.....	121
Gambar 4.63 Perbandingan respon sudut <i>Pitch</i> dengan gangguan .....	121
Gambar 4.64 <i>Tracking</i> dan <i>path following</i> Lintasan Persegi tanpa gangguan....	123
Gambar 4.65 a. <i>Wind gust</i> secara keseluruhan, b. pemberian gangguan pada detik ke-15 hingga detik ke-40 dan c. <i>Wind gust</i> pada sistem .....	124
Gambar 4.66 Respon quadcopter pada sumbu X dan Y dengan gangguan .....	125
Gambar 4.67 Perbandingan respon posisi X dengan gangguan .....	125
Gambar 4.68 Perbandingan respon posisi Y dengan gangguan .....	126
Gambar 4.69 Perbandingan respon sudut <i>Roll</i> dengan gangguan.....	126
Gambar 4.70 Perbandingan respon sudut <i>Pitch</i> dengan gangguan .....	126

*--Halaman ini sengaja dikosongkan--*

## DAFTAR TABEL

Tabel 1.1. Daftar penelitian sebelumnya berdasarkan kajian pustaka .....	4
Tabel 1.2 Daftar penelitian sebelumnya yang telah dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan.....	10
Tabel 2.1 Variabel pada Pergerakan Quadcopter [14].....	29
Tabel 3.1 Parameter Sistem Quanser Qball-X4 [14] .....	54
Tabel 3.2 Algoritma <i>path following straigh line</i> [13].....	61
Tabel 3.3 Nilai yang digunakan sebagai parameter kontrol posisi $X$ dan $Y$ .....	65
Tabel 3.4 Kontroler <i>Outer loop</i> (Posisi) .....	65
Tabel 3.5 Algoritma Kontrol Rotasi [10].....	70
Tabel 4.1 Nilai <i>Integral Square Error</i> (ISE) tiap perubahan <i>magnitude stept</i> .....	84
Tabel 4.2 Nilai <i>Integral Square Error</i> (ISE) tiap perubahan <i>magnitude gust</i> .....	87
Tabel 4.3 Nilai <i>Integral Square Error</i> (ISE) tiap perubahan <i>magnitude step+gust</i> 15-40(s) .....	92
Tabel 4.4 Nilai <i>Integral Square Error</i> (ISE) tiap perubahan <i>magnitude step+gust</i> 0-62.5(s) .....	96
Tabel 4.5 Nilai <i>Integral Square Error</i> (ISE) pada perubahan <i>Variance Noise</i> ... ..	100
Tabel 4.6 Penyimpangan posisi quadcopter pada sumbu $X$ dan $Y$ saat <i>step</i> .....	103
Tabel 4.7 Penyimpangan posisi quadcopter pada sudut <i>roll</i> dan <i>pitch</i> saat <i>step</i> .....	103
Tabel 4.8 Nilai gain $K$ pada <i>inner loop</i> dan norm- $\infty$ sistem untuk berbagai nilai $\gamma$ .....	104
Tabel 4.9 Penyimpangan posisi quadcopter pada sumbu $X$ dan $Y$ saat <i>gust</i> .....	107
Tabel 4.10 Penyimpangan posisi quadcopter pada sudut <i>roll</i> dan <i>pitch</i> saat <i>gust</i> .....	107
Tabel 4.11 Gain $K$ pada <i>inner loop</i> dan norm- $\infty$ sistem untuk berbagai nilai $\gamma$ .....	107
Tabel 4.12 Penyimpangan posisi quadcopter pada sumbu $X$ dan $Y$ saat <i>Step+Gust</i> 15-40(s) .....	110
Tabel 4.13 Penyimpangan posisi quadcopter pada sudut <i>roll</i> dan <i>pitch</i> saat <i>Step+Gust</i> 15-40(s) .....	110
Tabel 4.14 Gain $K$ pada <i>inner loop</i> dan norm- $\infty$ sistem untuk berbagai nilai $\gamma$ .....	111
Tabel 4.15 Penyimpangan posisi quadcopter pada sumbu $X$ dan $Y$ saat <i>Step+Gust</i> 0-62.5(s) .....	114
Tabel 4.16 Penyimpangan posisi quadcopter pada sudut <i>roll</i> dan <i>pitch</i> saat <i>Step+Gust</i> 0-62.5(s) .....	114
Tabel 4.17 Gain $K$ pada <i>inner loop</i> dan norm- $\infty$ sistem untuk berbagai nilai $\gamma$ .....	114
Tabel 4.18 Perbandingan rata-rata penyimpangan dan kecepatan kedua metode dalam lintasan lingkaran .....	122
Tabel 4.19 Perbandingan rata-rata penyimpangan dan kecepatan kedua metode dalam lintasan persegi .....	127

*--Halaman ini sengaja dikosongkan--*



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini teknologi *Unmanned Aerial Vehicle* (UAV) telah banyak digunakan dalam keperluan pengawasan, pencarian dan penyelamatan. Beberapa penelitian tentang quadcopter telah dilakukan, diantaranya adalah penelitian tentang kestabilan quadcopter, *take-off* dan *landing*, *maneuver*, kokoh terhadap gangguan dari luar, *tracking waypoint* serta *path following*. Hal tersebut yang membuat quadcopter menjadi daya tarik bagi peneliti dalam mengembangkan berbagai metode kontrol.

Berbagai macam metode yang digunakan untuk menyelesaikan permasalahan kontrol pada quadcopter sebagai contoh nonlinear kontrol[1], linear kontrol [2]-[3], dan *intelligent control*[4]. Metode *nonlinear backstepping* lebih bisa mendekati karakteristik sistem *nonlinear*[1]. Sehingga, langkah perhitungan yang digunakan lebih rumit. Metode *Linear Quadratic Regulator* (LQR) dengan menambah integrator / kompensator digunakan untuk menyelesaikan *tracking* sinyal referensi *step*[2]. Teknik ini mampu merepresentasikan dinamika nonlinear quadrotor, model Takagi Sugeno digunakan [4], dengan gain *state feedback* dalam aturan kontroler takagi sugeno diperoleh dengan metode optimal. LQR juga digunakan pada untuk persoalan *attitude*[3].

Kontrol ini juga sudah banyak diterapkan seperti pada [4], yang menggunakan *Linear Quadratic Regulator* (LQR) dan *Linear Quadratic Tracking* (LQT). Kelebihan dari kedua metode ini adalah penggunaan model linier yang sederhana namun tetap dapat merepresentasikan model nonlinier dari quadrotor disekitar titik *equilibrium*.

Pengembangan terus dilakukan untuk memperbaiki respon sistem pada quadcopter dalam permasalahan *tracking* dimana [5], mampu melakukan *tracking* dalam bentuk lingkaran, persegi maupun helix / spiral dengan menggunakan metode optimal yaitu optimal *output feedback* dengan menambahkan struktur dari *Command-Generator Tracker* (CGT) yang digunakan untuk mendapatkan kesesuaian *tracking* sinyal referensi dengan *error tracking* sekecil mungkin. Untuk menjaga kestabilan *attitude*, digunakan kontroler *output feedback* yang memiliki performa  $H_{\infty}$  sehingga quadcopter lebih handal dalam menangani adanya gangguan dari luar.

Aplikasi dari metode-metode kontrol tersebut menggunakan hukum kontrol *state feedback*. Oleh karena itu dalam implementasi sistem kontrol tersebut mensyaratkan semua *state* sistem quadrotor harus terukur. Apabila terdapat *state* yang tidak dapat diukur maka metode metode tersebut tidak dapat di

implementasikan pada *real plant*. Sistem kontrol berbasis *output feedback* dapat digunakan untuk mengatasi persoalan keterbatasan state yang terukur. *Static Output Feedback* (SOF) [6] berbasis performa  $H_\infty$  digunakan untuk *trajectory tracking* dengan struktur kontrol *inner* dan *outer loop*. Sistem kontrol ini robust terhadap gangguan.

Representasi *input* referensi dalam struktur *Command-Generator Tracker Model Following* (CGTMF) dibahas dalam [9]-[16]. Sinyal referensi yang digunakan dalam *trajectori* atau *path following* dapat berupa sinyal apa saja. Untuk sinyal referensi yang bukan unit *step* penambahan *integrator* pada struktur kontrol *tracking* tidak dapat menjamin *zero steady-state tracking error*. Oleh karena itu, desain kontrol *tracking* menggunakan *Command-Generator Tracker Model Following* (CGTMF) diajukan dalam [9]-[16]. Pengaturan arah hadap dari suatu *plant* terhadap suatu *path* atau sinyal referensi dilakukan dengan menggunakan algoritma *Line Of Sight* (LOS) [13]-[17].

Tujuan dari tesis ini adalah mendesain sistem kontrol menggunakan *output feedback* dengan *Command-Generator Tracker Model Following* (CGTMF) berbasis *Line Of Sight* (LOS) untuk melakukan *path following* dengan mengatur arah hadap. *Path* yang digunakan adalah lingkaran, persegi dan *helix*. Quadcopter dalam mengikuti *path* yang diberikan sudah dalam kondisi *hover*.

## 1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah bagaimana merancang stuktur kontrol untuk quadcopter yang dapat melakukan *path following* terhadap *path* lingkaran walaupun terdapat gangguan eksternal.

## 1.3 Tujuan

Tujuan dari tesis ini adalah mendesain sistem kontrol menggunakan *output feedback* dengan *Command-Generator Tracker Model Following* (CGTMF) berbasis *Line Of Sight* (LOS) untuk melakukan *path following* dengan mengatur arah hadap serta mampu mengatasi adanya gangguan eksternal.

## 1.4 Batasan Masalah

Pada usulan penelitian ini terdapat beberapa batasan masalah diantaranya yaitu model yang digunakan adalah quadcopter Quanser Q-Ball X-4, tidak membahas permasalahan kontrol pada saat *take off* dan *landing*, dan gangguan yang diberikan bernilai terbatas.

## 1.5 Kontribusi

Kontribusi dalam usulan penelitian ini adalah menghasilkan struktur kontrol *command-generator tracker model following* berbasis *Line Of Sight* yang stabil dengan performa  $H_\infty$  dipertimbangkan agar mampu mengatasi gangguan eksternal dan dapat melakukan *path following* pada *path* lingkaran.

## 1.6 Metodologi Penelitian

Pada penelitian ini dilakukan beberapa proses secara bertahap dan berurutan agar tercapai tujuan akhir dari penelitian:

### 1. Studi Literatur

Hal-hal yang akan dipelajari untuk mendukung pengerjaan tesis adalah konsep Quadcopter, kontrol *Linear Quadratic regulator* berbasis *output-feedback*, konsep *Command-Generator Tracker*, Performansi  $H_\infty$ , Linear Matrix Inequality (LMI), dan *path following* dengan algoritma *Line Of Sight* (LOS).

### 2. Permodelan Sistem

Pemodelan dilakukan dengan melakukan identifikasi fisik dengan menggunakan pendekatan Newton-Euler agar diketahui hubungan antar gaya yang bekerja pada sistem.

### 3. Perancangan kontroler

Tahap perancangan kontroler dilakukan untuk mendapatkan desain kontroler yang tepat untuk memenuhi performa yang diinginkan. Adapun kontroler yang akan dirancang adalah kontroler optimal *output feedback* yang memiliki performa  $H_\infty$  untuk kontrol stabilisasi dan optimal *output feedback* dengan struktur CGT model *following* dengan menambahkan algoritma LOS untuk kontrol *path following*.

### 4. Pengujian dan Analisis

Kemampuan sistem untuk dapat melakukan *path following* sesuai dengan referensi yang diinginkan akan dibuktikan pada simulasi. Hasil desain akan dianalisa apakah sistem sudah memenuhi kriteria kestabilan Lyapunov.

### 5. Kesimpulan

Pada tahap ini, 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.

## 1.7 Posisi dan Road Map Penelitian

Pada sub bab ini akan disampaikan sejauh mana perkembangan dan pencapaian penelitian lain yang sebidang dan posisi kontribusi dari penelitian ini.

### 1.7.1 Penelitian Sebelumnya berdasarkan kajian pustaka.

Penelitian pada bidang quadcopter khususnya pada trayektori dan *path following* sudah banyak dilakukan. Tabel 1.1 adalah beberapa studi literatur penelitian sebelumnya yang dikelompokkan sesuai dengan fokus penelitian.

Tabel 1.1. Daftar penelitian sebelumnya berdasarkan kajian pustaka.

Topik	Tahun					
	2007	2012	2014	2015	2016	2017
<i>Path following</i>				<b>David Cabecian, Rita Cunha, and Carlos Silvestre.</b>  Menguji kontroler secara global untuk mengarahkan quadcopter menuju <i>path</i> yang sudah ditentukan.		<b>Anggara Trisn Nugraha, Trihastuti agustinah.</b>  Menguji kontroler secara global untuk mengarahkan quadcopter menuju <i>path</i> lingkaran,

Topik	Tahun					
	2007	2012	2014	2015	2016	2017
Trajectory tracking	<p><b>J. Gadewadikar, F. L. Lewis, K. Subbarao and B. M. Chen.</b></p> <p><i>Trajectory tracking</i> yang dilakukan quadcopter pada posisi x, y, z</p>	<p><b>E. C. Vivas G, M. V. Pena G and C. I. Rodriguez F.</b></p> <p><i>Trajectory tracking</i> yang dilakukan quadcopter pada gerak altitude.</p> <p><b>F. Yacef, O. Bouhali, H. Khebbache and F. Boudjema</b></p> <p><i>Trajectory tracking</i> yang dilakukan quadcopter pada</p>		<p><b>E. C. Suicmez and A. T. Kutay.</b></p> <p><i>Trajectory tracking</i> yang dilakukan quadcopter pada posisi x, y, z</p>		
					<p><b>Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh.</b></p> <p><i>Trajectory tracking</i> yang dilakukan quadcopter pada orientasi dan translasi. Dengan jalur berbentuk lingkaran</p>	persegi dan spiral.

Topik	Tahun					
	2007	2012	2014	2015	2016	2017
Stabilisasi dengan metode NonLinear		orientasi dan translasi.		David Cabecian, Rita Cunha, and Carlos Silvestre.  Menggunakan <i>adaptive state feedback (backstepping)</i> .		
	J. Gadewadikar, F. L. Lewis, K. Subbarao and B. M. Chen.  Menggunakan <i>state feedback</i> .	E. C. Vivas G, M. V. Pena G and C. I. Rodriguez F.  Menggunakan <i>state feedback</i> .	E. C. Suicmez and A. T. Kutay.  Menggunakan <i>state feedback</i> .		Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh.  Menggunakan <i>static output feedback</i> .	Anggara Trisn nugraha, Trihastuti agustinah.  Menggunakan <i>static output feedback</i> .
Stabilisasi dengan metode Linear		F. Yacef, O. Bouhali, H. Khebbache and F. Boudjema.				

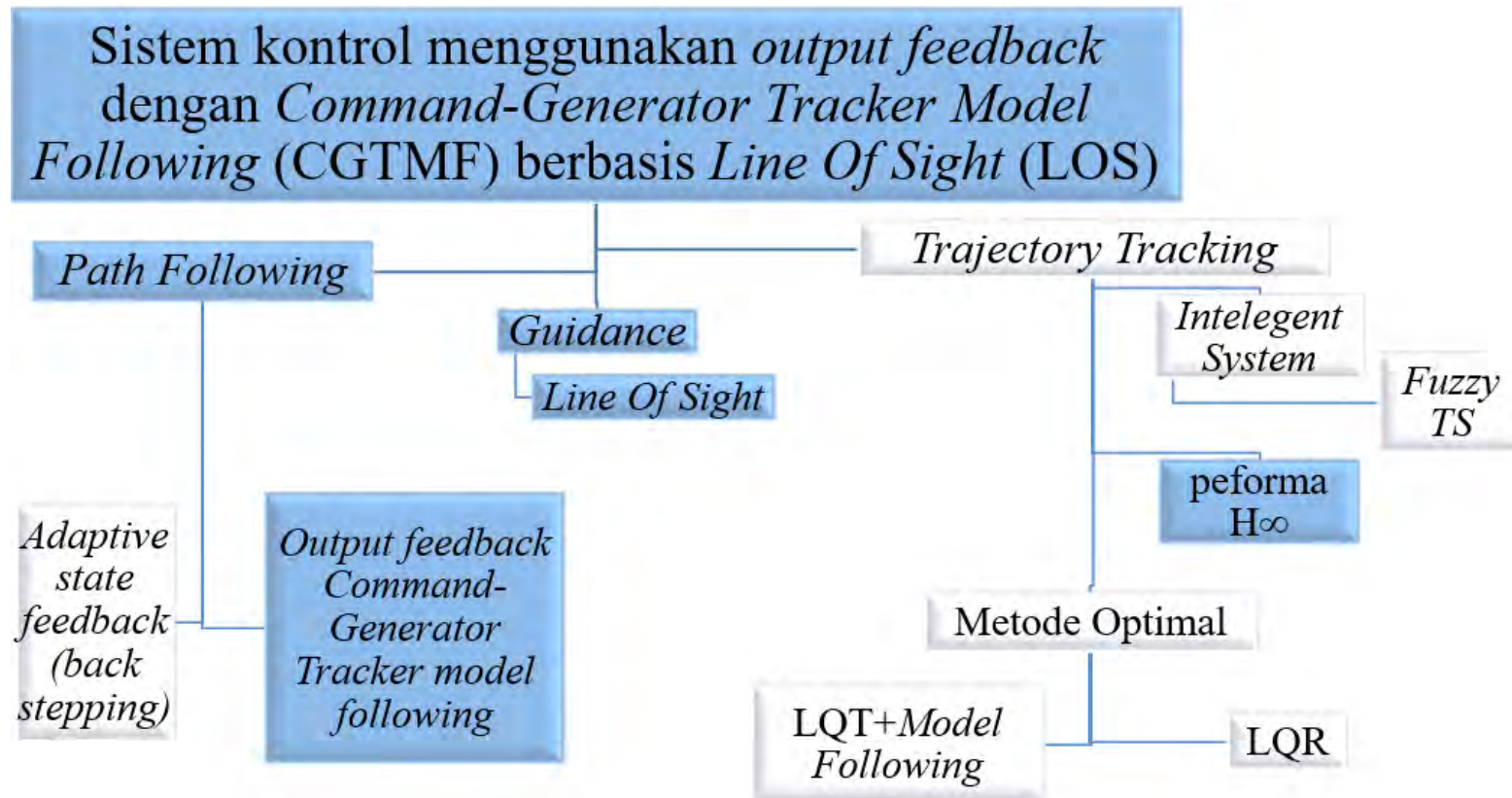
Topik	Tahun					
	2007	2012	2014	2015	2016	2017
Metode optimal		Menggunakan <i>state feedback</i> dengan LMI yang ditambahkan <i>pole placement</i>				
		<b>E. C. Vivas G, M. V. Pena G and C. I. Rodriguez F.</b>	<b>E. C. Suicmez and A. T. Kutay.</b>		<b>Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh.</b>	<b>Anggara Trisn Nugraha, Trihastuti agustinah.</b>
		Menggunakan kontrol <i>linear quadratic regulator</i> dengan <i>effect integrator</i> .	Menggunakan <i>linear quadratic tracking</i> yang ditambahkan <i>model following</i>		Menggunakan <i>optimal output feedback command-generator tracker</i>	Menggunakan <i>optimal output feedback command-generator tracker model following</i>
Metode <i>intelegent system</i>		<b>F. Yacef, O. Bouhali, H. Khebbache and F. Boudjema.</b>				

Topik	Tahun					
	2007	2012	2014	2015	2016	2017
Performa <i>Robust</i>		Menggunakan fuzzy takagi sugeno.				
	J. Gadewadikar, F. L. Lewis, K. Subbarao and B. M. Chen. Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> dan <i>outerloop</i> .				Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh. Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> .	Anggara Trisna nugraha, Trihastuti agustinah. Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> .
LOS ( <i>Line Of Sight</i> )						Anggara Trisna nugraha, Trihastuti agustinah. Penambahan algoritma <i>Line Of sight</i> (LOS) untuk menjaga arah hadap.



### 1.7.2 Road Map Penelitian

Berdasarkan telaah pustaka dari penelitian sebelumnya, penelitian tentang sistem kontrol menggunakan *output feedback* dengan *Command-Generator Tracker Model Following* (CGTMF) berbasis *Line Of Sight* (LOS) dapat dibagi menjadi beberapa topik penelitian yaitu (1) *Path following* dan (2) *Trayektori tracking*. Berdasarkan pembagian topik penelitian, maka dapat dianalisa capaian penelitian pendahulu pada masing-masing topik yang dapat dijelaskan pada diagram peta jalan penelitian gambar 1.1.



Gambar 1.1. Peta jalan penelitian

**1.7.3 Penelitian Sebelumnya yang Telah Dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan.**

Penelitian pada bidang quadcopter yang telah dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan. Tabel 1.2 adalah beberapa studi literatur penelitian sebelumnya yang dikelompokkan sesuai dengan fokus penelitian.

Tabel 1.2 Daftar penelitian sebelumnya yang telah dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan.

Topik	Tahun				
	2013	2014	2015	2016	2017
<i>Hover</i>	<b>Jumiyatun</b> Pengaturan kestabilan hover.				
	<b>Rudy Kurniawan</b> Menjaga posisi ketinggian saat hover				
<i>Landing</i>		<b>Nofria Hanafi</b> Pendaratan otomatis pada <i>platform</i> yang bergerak			
<i>VTOL</i>					<b>Chalidia Nurin Hamdani</b>

Topik	Tahun				
	2013	2014	2015	2016	2017
<i>Path following</i>					Perancangan autonomus VTOL pada quadcopter <b>Anggara Trisna nugraha, Trihastuti agustinah.</b> Menguji kontroler secara global untuk mengarahkan quadcopter menuju <i>path</i> lingkaran, persegi dan spiral.
<i>Waypoint Tracking</i>		<b>Mokhammad Khoizin</b> <i>Waypoint tracking</i> pada <i>micro</i> quadrotor	<b>Swadexi Istiqphara</b> <i>Waypoint tracking</i> pada quadcopter.		

Topik	Tahun				
	2013	2014	2015	2016	2017
<i>Trajectory tracking</i>			<b>Hanum Arrosida</b> Pengendalian gerak <i>cruise</i> pada quadrotor	<b>Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh.</b> <i>Trajectory tracking</i> yang dilakukan quadcopter pada orientasi dan translasi. Dengan jalur berbentuk lingkaran	<b>Mohammad Nur</b> Kontrol <i>Tracking</i> pada quadrotor.
<i>Fault Tolerant</i>				<b>Anisa Ulya Darajat</b> Kompensasi kesalahan aktuator	
Stabilisasi dengan metode NonLinear	<b>Jumiyatun</b> Menggunakan <i>backstepping</i> .	<b>Mokhammad Khoizin</b> Menggunakan <i>backstepping</i> .	<b>Swadexi Istiqphara</b> Menggunakan <i>Sliding Mode Control</i>	<b>Anisa Ulya Darajat</b> Menggunakan <i>backstepping</i> .	
	<b>Rudy Kurniawan</b>		<b>Hanum Arrosida</b>		

Topik	Tahun				
	2013	2014	2015	2016	2017
Stabilisasi dengan metode Linear	Menggunakan <i>Sliding Mode Control</i>		Menggunakan <i>Nonlinear PD</i> dan <i>Nonlinear decoupling PD</i>	Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh. Menggunakan <i>static output feedback</i> .	Anggara Trisna nugraha, Trihastuti agustinah. Menggunakan <i>static output feedback</i> .
					Chalidia Nurin Hamdani <i>State Feedback liniearization</i>

Topik	Tahun				
	2013	2014	2015	2016	2017
<b>Metode optimal</b>				<b>Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh.</b> Menggunakan <i>optimal output feedback command-generator tracker</i>	<b>Anggara Trisna nugraha, Trihastuti agustinah.</b> Menggunakan <i>optimal output feedback command-generator tracker model following</i>
<b>Metode intelegent system</b>		<b>Nofria Hanafi</b> <i>Neuro fuzzy</i>			<b>Mohammad Nur</b> Menggunakan <i>linear quadratic tracking</i>  <b>Chalidia Nurin Hamdani</b> Menggunakan <i>fuzzy Takagi Sugeno</i>
<b>Metode adaptif</b>	<b>Rudy Kurniawan</b> Menggunakan <i>self tunning PID.</i>	<b>Mokhammad Khoizin</b> Menggunakan <i>adaptive parameter</i>	<b>Swadexi Istiqphara</b> Menggunakan <i>self Tunning Sliding Mode Control</i>		

Topik	Tahun				
	2013	2014	2015	2016	2017
Performa <i>Robust</i>		<i>approximation (adaptive backstepping)</i>		Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh. Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> .	Anggara Trisna nugraha, Trihastuti agustinah. Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> .
				Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> .	Chalidia Nurin Hamdani Penambahan performa $H_\infty$ pada kontrol <i>innerloop</i> .
Filter			Swadexi Istiqphara Menggunakan <i>complementary filter</i>	Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh.	Mohammad Nur Menggunakan <i>extended kalman filter</i>

Topik	Tahun				
	2013	2014	2015	2016	2017
Estimator			<i>(high pass + low pass)</i>	Menggunakan <i>command-generator tracker</i>	<b>Anggara Trisna nugraha, Trihastuti agustinah.</b>  Menggunakan <i>command-generator tracker</i> .
			<b>Swadexi Istiqphara</b>  Menggunakan <i>Least Square Parameter Estimator</i>	<b>Anisa Ulya Darajat</b>  Menggunakan <i>Least Square Parameter Estimator</i>	
LOS ( <i>Line Of Sight</i> )					<b>Anggara Trisna nugraha, Trihastuti agustinah.</b>  Penambahan algoritma <i>Line Of sight (LOS)</i> untuk menjaga arah hadap.

Agar lebih jelas dalam pembahasan perkembangan penelitian yang sudah dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan, dapat dilihat pada lampiran C.



## BAB 2

### KAJIAN PUSTAKA

#### 2.1 Kajian Penelitian Terkait

Pada sub-bab ini menjelaskan penelitian terakhir yang telah dilakukan baik dari segi kinematika dan dinamika quadcopter, penggunaan berbagai macam metode untuk mendapatkan respon yang diinginkan serta penggunaan metode yang dapat mengatasi perubahan kondisi pada saat quadcopter melakukan *tracking* dan *path following*, sehingga diperoleh ide dasar penyusunan tesis.

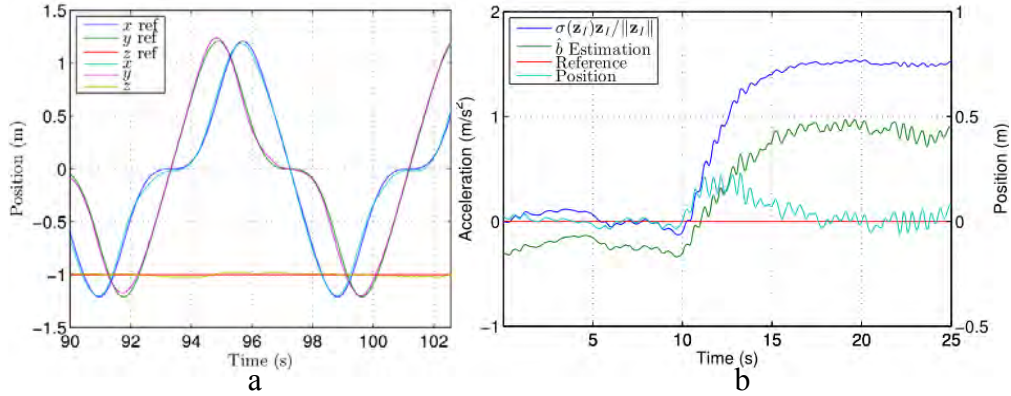
##### 2.1.1 A Globally Stabilizing Path following Controller for Rotorcraft with Wind Disturbance Rejection [1]

*Paper* ini membahas bagaimana mendesain dan menguji kontroler secara global yang digunakan untuk mengarahkan quadcopter pada jalur / *path* yang sudah terdefinisi. Selain itu, quadcopter juga diuji kekokohan terhadap adanya gangguan berupa angin yang bernilai konstan.

Untuk mengatasi permasalahan yang akan diuji, pada *paper* ini menggunakan struktur kontrol *nonlinear adaptive state feedback (backstepping)* pada saat quadcopter dalam kondisi *hover*. Penggunaan kontroler *backstepping* ini menjamin kesesuaian terhadap suatu jalur/*path* yang sudah terdefinisi sehingga menghasilkan nilai *error* sekecil mungkin ( $error = 0$ ) pada saat adanya gangguan angin yang bernilai konstan. Selain itu, menjamin kondisi aktuasi yang dapat membatasi fungsi dari *error* kecepatan dan posisi, tanpa memaksakan kondisi aktuasi secara berlebihan dan sesuai batas pada aksi kontrol dengan performa yang tinggi.

Dilihat dari Gambar 2.1a quadcopter melakukan *path following* pada lintasan berbentuk lingkaran yang ditampilkan dalam bentuk plot posisi  $x$ ,  $y$ , dan  $z$  dengan kondisi waktu dimana *path* terbentuk dari referensi posisi. Respon sistem yang dihasilkan dapat mengikuti referensi dengan baik pada saat tidak ada gangguan.

Gambar 2.1b menunjukkan kondisi perubahan *disturbance* setelah diestimasi. Efek *disturbance* diberikan pada detik ke 10 dimana quadcopter sudah dalam kondisi *hover*. Garis berwarna biru merupakan efek *disturbance* dimana pada saat kipas angin menyala *error* dari posisi  $z$  mencapai 15cm dan pada saat itu juga dengan cepat sistem dari kontrol tersebut mengestimasi adanya *disturbance* sehingga respon dari sistem kembali ke titik nol.



Gambar 2.1 a. Respon sistem saat melakukan *path following* dan b. Respon posisi  $z$  saat diberi *Disturbance*.

Kelebihan pada *paper* ini adalah quadcopter mampu melakukan *path following* terhadap referensi yang diberikan saat tidak ada gangguan dan kontroler mampu mengatasi serta mereduksi adanya *wind disturbance*. Namun, kekurangannya adalah terjadi *ripple* saat diberi gangguan.

### 2.1.2 Optimal Path Tracking Control of a Quadcopter UAV [2]

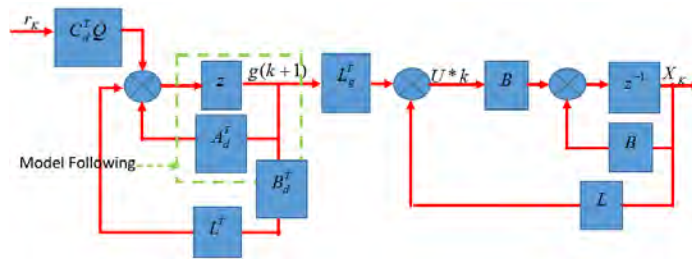
Kontrol *Linear Quadratic Tracking* (LQT) digunakan dalam *paper* ini untuk mengontrol *path tracking* quadcopter. Model yang digunakan adalah model *non linear* dari quadcopter di linierisasi hanya pada saat *hover*. Penyelesaian permasalahan kontrol optimal menggunakan hasil linierisasi dari model. Indeks performa didefinisikan dari pertimbangan untung rugi antara performansi *tracking* yang baik dengan konsumsi energi. Indeks performa yang digunakan adalah sebagai berikut:

$$J = \frac{1}{2} [C_d X(k_f) - r(k_f)]^T F [C_d X(k_f) - r(k_f)] + \frac{1}{2} \sum_{k=k_0}^{k_f-1} ([C_d X(k) - r(k)]^T Q [C_d X(k) - r(k)] + U^T(k) R U(k)) \quad (2.1)$$

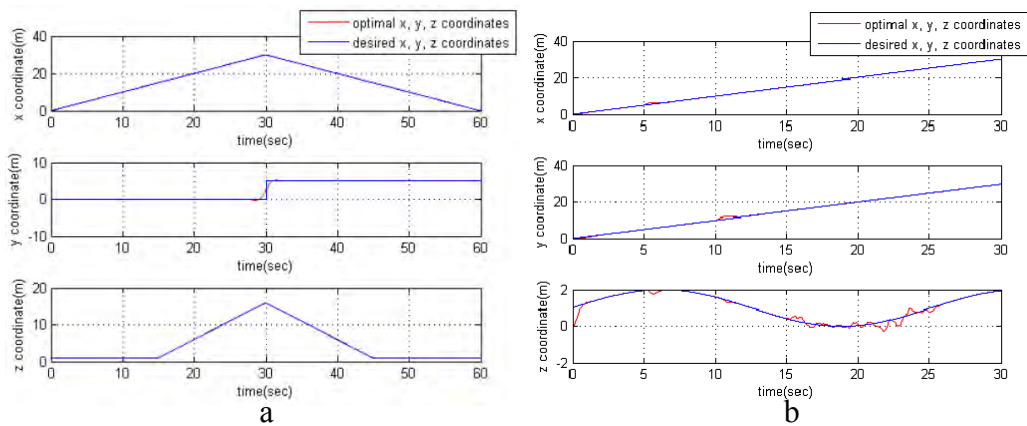
dengan menggunakan nilai kondisi batas  $X(k_0) = X_0$  adalah kondisi saat *hover*.

Kontroler *state feedback* dicari dengan memecahkan persamaan Riccati waktu diskrit. Sehingga untuk mendapatkan *state feedback gain* menggunakan algoritma kontrol sebagai berikut:

$$\begin{aligned} P(k) &= A_d^T P(k+1) [I + E P(k+1)]^{-1} A_d + V \\ V &= C_d^T Q C_d \\ L(k) &= [R + B_d^T P(k+1) B_d]^{-1} B_d^T P(k+1) A_d \\ Lg(k) &= [R + B_d^T P(k+1) B_d]^{-1} B_d^T \end{aligned} \quad (2.2)$$



Gambar 2.2 Diagram blok *Linear Quadratic Tracking* dengan waktu diskrit



Gambar 2.3a. Respon sistem *path tracking* ( $x, y, z$ ) dengan tanpa gangguan dan b. Respon sistem *path tracking* ( $x, y, z$ ) dengan adanya gangguan

Pada *paper* ini, penyelesaian kontrol LQT diselesaikan menggunakan persamaan Riccati secara *offline* untuk mendapatkan *feedback gain* dan *feedforward gain*. Sedangkan quadcopter adalah sistem yang *non linear*, penyelesaian dengan persamaan Riccati lebih cocok untuk sistem linier, karena solusi persamaan Riccati hanya untuk satu subsistem linier saja. Struktur Kontrol yang digunakan adalah *tracking* dengan tambahan model *following* agar quadcopter mampu mengikuti sinyal referensi serta mampu mereduksi adanya *overshoot* seperti pada Gambar 2.2.

Kelebihan dari *paper* ini adalah dengan menggunakan teknik *linearisasi* sederhana tetap mampu merepresentasikan model *nonlinear* dari quadcopter. Selain itu, kontroler mampu mereduksi *random Disturbance* sehingga dapat melakukan *tracking* dengan baik seperti yang terlihat pada Gambar 2.3a dan Gambar 2.3b.

Namun pada saat sistem diberi gangguan dari luar berupa *gaussian* dan *random*, respon yang terlihat pada Gambar 2.3b menunjukkan bahwa pada saat gangguan diberikan pada sumbu  $z$ , sistem tidak dapat menangani gangguan dengan baik pada saat *tracking* terbukti dengan adanya *ripple* pada respon sistem.

### 2.1.3 Simulation of the Quadcopter Controller with LQR Integral Effect [3]

Dinamika quadcopter yang memiliki kondisi *nonlinear* dan tidak stabil sehingga, dilinierisasi pada saat kondisi *hover*. Permasalahan kontrol *tracking* untuk *altitude* dan *attitude* yang diangkat sebagai suatu permasalahan pada paper ini. Paper ini menggunakan teknik kontrol linier, yaitu LQR untuk memecahkan permasalahan mengenai kontrol *attitude* pada quadcopter. *Error tracking* dihasilkan dari sinyal referensi berupa  $z_d, \phi_d, \theta_d, \psi_d$  yang akan dibandingkan dengan *output actual* yaitu  $z, \phi, \theta, \psi$ .

Desain kontrol dengan integrator yang ditambahkan dalam struktur kontrol ini, seperti yang terdapat pada Gambar 2.4a digunakan untuk meminimumkan *error tracking*. Indeks performa yang digunakan adalah sebagai berikut:

$$J = \int_0^{\infty} [x^T(x)R_{xx}\bar{x}(x) + u^T(t)R_{uu}u(t)]dt \quad (2.3)$$

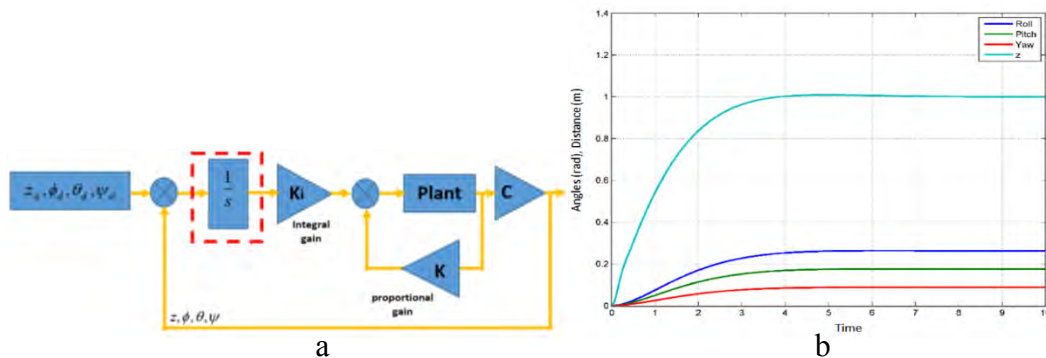
*State augmented* yang dituliskan dalam persamaan *state space* berupa  $\bar{x}$  adalah sebagai berikut:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_1(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_1(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} r(t) \quad (2.4)$$

dapat dilihat bahwa, kontroler yang terdiri dari dinamika kompensator dan dinamika *plant* yang diperluas. Maka sinyal kontrol dari *system augmented* adalah sebagai berikut:

$$u(t) = -\begin{bmatrix} K & K_1 \end{bmatrix} \begin{bmatrix} x(t) \\ x_1(t) \end{bmatrix} = -\bar{K}\bar{x}(t) \quad (2.5)$$

Penggunaan kontrol LQR yang ditambahkan dengan adanya integrator dapat menjadikan quadcopter mampu melakukan *tracking* terhadap sinyal referensi. Sehingga, hasil simulasi dapat dilihat pada Gambar 2.4b, dimana respon yang dihasilkan dari sinyal berupa *step* tanpa adanya *overshoot*.



Gambar 2.4 a. Struktur *Linear Quadratic Regulator* dengan integral dan b. Respon *attitude* dan *altitude* dari quadcopter

Namun, waktu untuk mencapai *steady state* cukup lama serta tidak dilakukan uji coba terhadap gangguan dan juga sinyal referensi yang diberikan hanya berupa sinyal *step*.

#### 2.1.4 Takagi-Sugeno Model for Quadcopter Modelling and Control Using Non linear State feedback Controller [4]

Pendekatan *multiple* model yang digunakan dalam paper ini, dimana merupakan pendekatan untuk pemodelan dengan interpolasi yang sebagian kecil dari dinamika sistem yang relatif sederhana dipadukan dengan model *fuzzy* T-S untuk pemodelan pada quadcopter. Pendekatan *multiple model*, memiliki tiga model lokal dalam daerah ruang kerja (*operating space*). Sistem *non linear* dibagi kedalam beberapa sub-sistem yang sederhana. Setiap sub-sistem sederhana tersebut adalah model lokal linier yang menggambarkan dinamika sistem *non linear* dalam region yang kecil dari ruang kerja. Gambaran mengenai pendekatan *multiple model* dapat dilihat pada Gambar 2.5.

Dengan pendekatan ini, untuk mendapatkan model *fuzzy* TS dari quadcopter hanya menggunakan tiga titik kerja dari kecepatan sudut *roll*, *pitch* dan *yaw*  $\xi(t) = [\phi, \theta, \psi]^T$ . Dengan beberapa model linier yang dipadukan dan fungsi aktivasi Gaussian, maka model dinamika quadcopter dapat ditulis sebagai berikut:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^r \mu_i(\xi(t))(A_i x(t) + B_i(t) + d_i) \\ y(t) = Cx(t) \end{cases} \quad (2.6)$$

dengan  $\mu_i(\xi(t)), i = 1, \dots, r$  ( $r = 3$ ) adalah fungsi aktivasi yang dinormalisasi, dan

$$\mu_i(\xi(t)) = \frac{\omega_i(\xi(t))}{\sum_{j=1}^3 \omega_j(\xi(t))}, \omega_i(\xi(t)) = \prod_{j=1}^3 \exp\left(-\frac{\xi_j(t) - c_{i,j}}{2\sigma_{i,j}^2}\right) \quad (2.7)$$

dimana  $c_{i,j}$  dan  $\sigma_{i,j}$  adalah parameter fungsi aktivasi Gaussian.

Setelah didapatkan model T-S quadcopter, selanjutnya dirancang kontroler *state feedback* menggunakan konsep PDC. Teori kontrol linier digunakan untuk merancang hukum kontrol, karena model T-S digambarkan oleh persamaan sistem linier. Persamaan hukum kontrol tersebut adalah sebagai berikut:

$$u(t) = -\sum_{j=1}^r \mu_j(\xi(t))u_j(t) = -\sum_{j=1}^r \mu_j(\xi(t))K_j x(t) \quad (2.8)$$

dimana  $K_i$  adalah penguat *state feedback*. Susbtitusi (2.8) kedalam (2.6), maka model *fuzzy* T-S lup tertutup dapat dipresentasikan sebagai berikut:

$$\dot{x}(t) = -\sum_{j=1}^r \mu_j(\xi(t)) \mu_j(\xi(t)) (A_i - B_i K_i) x(t) \quad (2.9)$$

Pada *paper* ini, kestabilan sistem *loop* tertutup dicari menggunakan teknik *pole placement*. Nilai eigen dari model *fuzzy* T-S ditempatkan dalam sub-region (*D*) pada sebelah kiri bidang kompleks untuk mencegah dinamika kontroler yang terlalu cepat dan mencapai perilaku transien yang diinginkan. Solusi penyelesaian dilakukan menggunakan teknik numerik LMI. Terdapat dua metode kontrol yang digunakan untuk menentukan nilai *state feedback*, yaitu model LMI tanpa *constraint*, *gain K* didesain sehingga menjadi persamaan sebagai berikut:

$$G_{ii}^T P + P G_{ii} < 0, \quad \forall i \in I \quad (2.10)$$

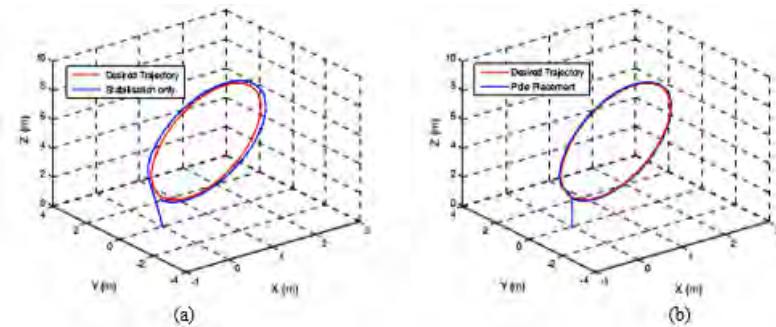
$$\left( \frac{G_{ij} + G_{ji}}{2} \right)^T P + P \left( \frac{G_{ij} + G_{ji}}{2} \right) < 0, \quad \forall (i, j) \in I_r^2, i < j$$

harus terpenuhi. Dimana  $G_{ii} = A_i - B_i K_i$ .

Hasil yang didapat menekankan pada perbedaan pengaruh penentuan kontroler *state feedback*. Simulasi dilakukan dalam dua kondisi, pertama *state feedback gain* akan stabil *asymptotic* pada  $\lim_{t \rightarrow \infty} x(t) = 0$ , dan kedua penempatan *pole* dalam region yang dibatasi (region *D*) pada sebelah kiri bidang kompleks untuk mencegah dinamika kontroler yang terlalu cepat dan mencapai perilaku transien yang diinginkan.

Model *closed loop* dari *fuzzy* T-S dikatakan stabil dalam region *D* untuk nilai *state feedback*  $K_i$  jika dan hanya jika terdapat matriks simetris positif definit  $X$  yang memenuhi pertidaksamaan berikut:

$$\begin{pmatrix} -\rho X & qX + XA_i^T + N_i^T B_i^T \\ qX + A_i + B_i N_i & -\rho X \end{pmatrix} < 0, i = j \quad (2.11)$$

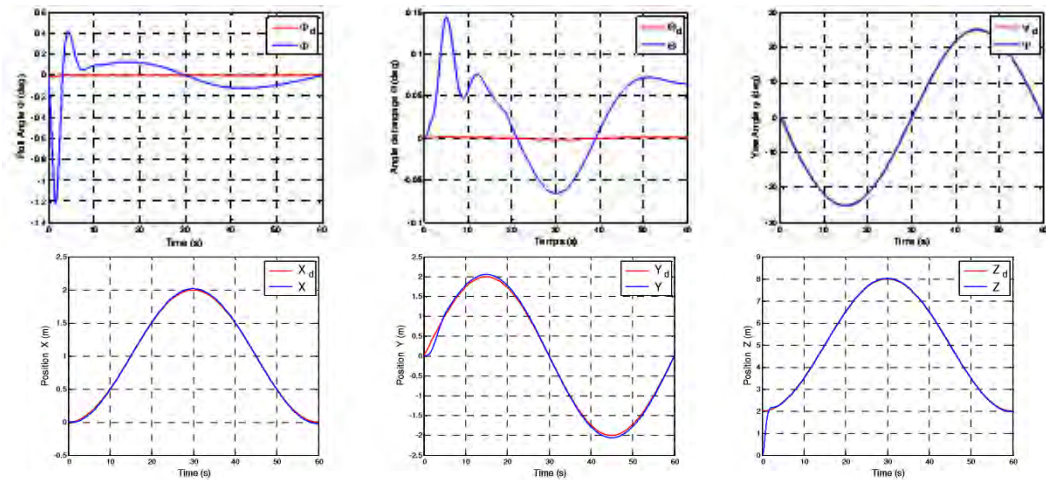


Gambar 2.5(a) Posisi 3D untuk  $K_{stabilisasi}$ , (b) Posisi 3D untuk  $K_{placement}$

Hasil simulasi menunjukkan bahwa kontroler LMI dengan teknik *pole placement* memberikan respon yang dapat melakukan *tracking* trayektori, sedangkan pada kontroler LMI tanpa teknik *pole placement* menghasilkan adanya

penyimpangan posisi seperti yang ditunjukkan pada Gambar 2.5. Kurva respon disajikan hanya membutuhkan waktu hingga detik ke-60.

Kekurangannya adalah respon sudut *roll* dan *pitch* tidak dapat mengikuti sinyal referensi (tidak dapat kembali ke 0 rad), seperti terlihat pada Gambar 2.6. Kelebihan dari *paper* ini adalah selain menggunakan konsep sektor *non linear*, pendekatan *multiple model* juga dapat diterapkan pada pemodelan *fuzzy T-S* untuk quadcopter sehingga aturan *fuzzy* menjadi lebih sederhana. Namun, dalam *paper* ini tidak dilakukan pengujian kekokohan sistem terhadap gangguan.



Gambar 2.6 Respon *attitude* quadcopter ( $\phi, \theta, \psi, x, y, z$ )

### 2.1.5 Kontrol *Tracking* untuk Quadrotor Menggunakan Metode Optimal Output Feedback dengan Command Generator Tracker [5]

Tesis ini menjelaskan tentang bagaimana quadcopter mampu melakukan *tracking* dalam bentuk lingkaran, persegi dan helix / spiral dengan menggunakan metode optimal yaitu optimal *output feedback* dengan menambahkan struktur *Command Generator Tracking* (CGT) [19].

*Command-Generator Tracker* (CGT) merupakan struktur desain kontrol yang memberikan kompensator sesuai dengan kebutuhan sistem untuk mendapatkan *error tracking* yang kecil untuk *input* referensi  $r(t)$  yang diinginkan. Pendekatan CGT ini akan menggabungkan model dinamika dari  $r(t)$  kedalam sistem kontrol. Oleh karena itu, dinamika sistem kontrol *tracking* akan terdiri dari gabungan dinamika *plant* dan kompensator.

*State augmented*

$$\frac{d}{dt} \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix} = \begin{bmatrix} \vdots & 0 \\ G & \vdots & -H \\ \cdots & \cdots & \cdots \\ 0 & \vdots & A \end{bmatrix} \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ -H \end{bmatrix} \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix} \quad (2.12)$$

Desain regulator LQ diterapkan pada sistem ini, karena jika *state* pada sistem ini menuju nol, maka *tracking error*  $e(t)$  akan hilang. Jika *state output* yang akan menjadi *feedback* bagi sistem, maka input kontrol sistem modifikasi menjadi

$$\mu = -[K_\varepsilon \quad K_y] \begin{bmatrix} \varepsilon \\ C\xi \end{bmatrix} \quad (2.13)$$

Dalam permasalahan tesis ini, konsep CGT diterapkan dalam sistem kontrol *tracking* untuk quadcopter dan nilai gain K dicari menggunakan metode regulator LQ dengan *output feedback*. Maka sistem *closed loop* dinyatakan oleh  $\dot{x} = (A - BKC)x + A_c x$ .

dengan indeks performa (PI) kuadratik untuk permasalahan ini adalah

$$J = \int_0^\infty (x^T Qx + u^T Ru) dt \quad (2.14)$$

dengan  $Q \geq 0, R > 0$  adalah matriks pembobot semidefinit positif. Diasumsikan sistem *closed-loop* adalah stabil *asymptotic* sehingga *state*  $x(t)$  akan bernilai nol hingga waktu yang tak hingga, maka  $J$  menjadi

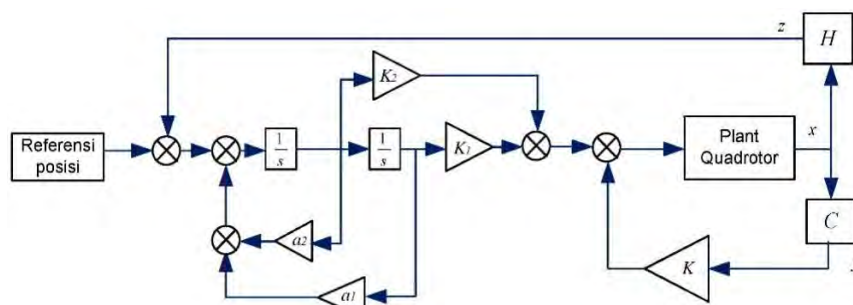
$$J = \frac{1}{2} (x^T(0) p x(0)) \quad (2.15)$$

Untuk kontrol pada *inner loop* menggunakan *Output feedback* dengan performa  $H_\infty$  yang mampu mengatasi gangguan dari luar. Struktur kontrol yang digunakan adalah *tracking* dengan tambahan *command generator tracking*

Respon *tracking* sistem terdapat pada Gambar 2.8a, menunjukkan respon jika dilihat pada bidang  $X$  dan  $Y$  dimana lintasan berbentuk lingkaran. Respon dapat mengikuti referensi yang diberikan dengan rata-rata penyimpangan  $\pm 0.09$  m.

Gambar 2.9 menunjukkan bahwa posisi quadcopter pada sumbu  $X$  dan  $Y$  dapat mengikuti referensi yang berbentuk sinyal sinusoidal. *Tracking* dimulai pada detik ke-2.5 (pada awal *tracking*), terjadi keterlambatan respon sekitar 0.5 detik yang menyebabkan penyimpangan respon sebesar 0.1 m. Pada detik ke-5.5 respon dapat kembali berada pada lintasan referensi.

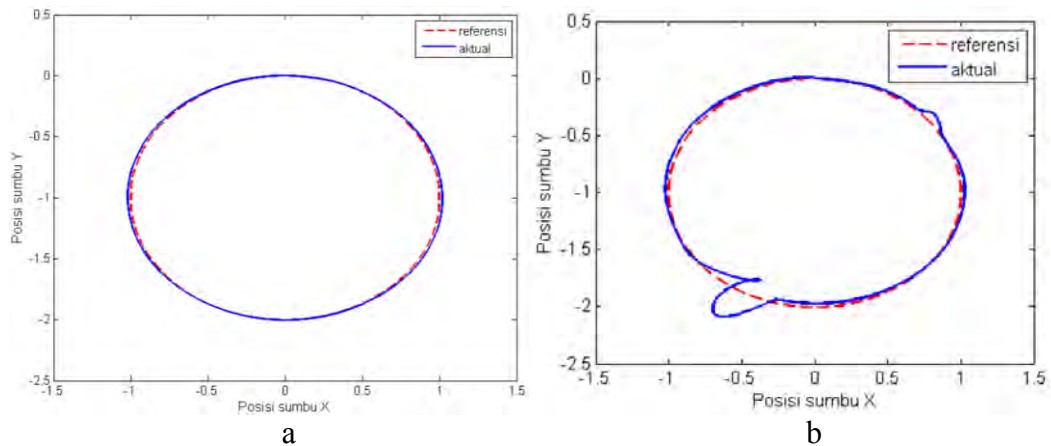
Gambar 2.8b menunjukkan respon *tracking* quadcopter pada sumbu  $X$  dan  $Y$ . Keberadaan *wind gust* membuat quadcopter menyimpang dari lintasan referensi,



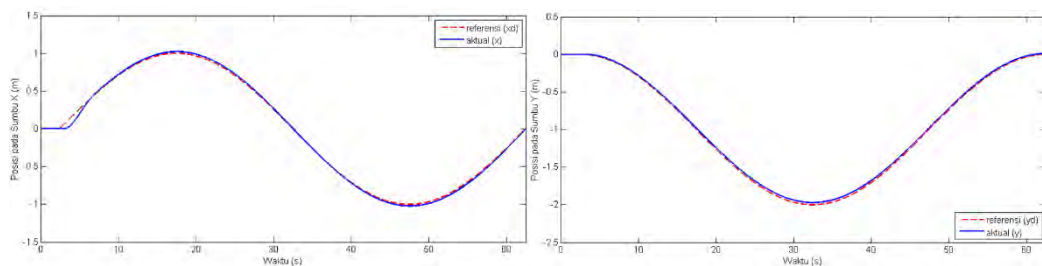
Gambar 2.7 Diagram blok kontroler *outer loop* [19].



pada saat yang bersamaan gerakan *roll* dan *pitch* dari quadcopter menyesuaikan agar pengaruh *wind gust* membuat quadcopter dapat kembali *tracking* sesuai dengan referensi yang diberikan.



Gambar 2.8a. Respon jika dilihat pada bidang *X* dan *Y* dimana lintasan berbentuk lingkaran dan b. Respon *tracking* lingkaran terhadap efek *wind gust*

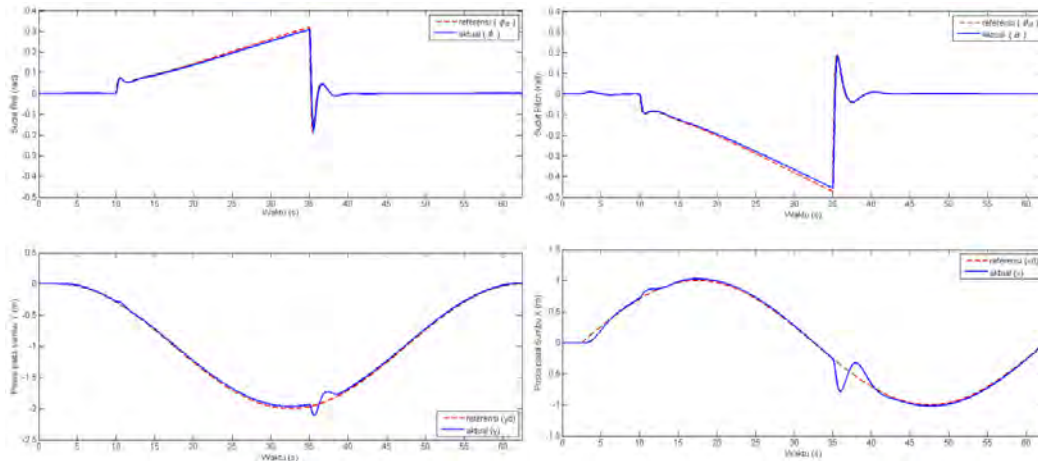


Gambar 2.9 Respon *tracking* posisi pada sumbu *X* dan *Y*

Seperti yang ditunjukkan pada Gambar 2.10, pengaruh *wind gust* terhadap sudut  $\phi$  terjadi dari detik ke-10 s/d detik ke-40, pergerakan dari  $-0.2$  rad s/d  $0.3$  rad atau sekitar  $-11^\circ$  s/d  $18^\circ$ . Pada respon posisi pada sumbu *X* dan *Y* dan, penyimpangan yang kecil terjadi pada detik ke-10, sedangkan pada detik ke-35 s/d detik ke-40.

Kelebihan dari tesis ini adalah pada saat respon tanpa adanya *wind gust* dapat mengikuti referensi yang terdapat pada Gambar 2.8a. Sedangkan Gambar 2.8b dengan adanya *wind gust* membuat adanya penyimpangan dari lintasan referensi, pada saat yang bersamaan gerakan *roll* dan *pitch* dari quadcopter dapat kembali *tracking* sesuai dengan referensi. Struktur *default* dari CGT (konsep *feedforward*) yang mana menggunakan orde3 tetapi pada tesis ini bisa diterapkan pada orde2.

Sedangkan kekurangannya pada Gambar 2.9 dimana respon terhadap posisi *x* mengalami keterlambatan yang mengakibatkan penyimpangan.



Gambar 2.10 Respon *tracking* posisi pada sumbu *X* dan *Y* terhadap efek *wind gust*

Pemberian gangguan berupa *wind gust* hanya diberikan pada periode waktu tertentu (detik ke 10-35) bukan dari awal quadcopter terbang.

### 2.1.6 Design of $H_\infty$ Command and Control Loops for Unmanned Aerial Vehicles using Static Output-Feedback [6]

Paper ini menjelaskan bagaimana mendesain sebuah kontrol posisi menggunakan kontrol SOF yang memiliki performa  $H_\infty$  pada *plant* yang berupa helicopter. Permasalahan kontrol *tracking* untuk helikopter yang dijadikan pokok permasalahan pada paper ini.

Kontrol *inner loop* berfungsi untuk menjaga *attitude* helikopter dengan mengontrol *state roll* dan *pitch*. Kontroler *outer loop* berfungsi untuk menjaga posisi helikopter. Struktur kontrol tersebut dapat dilihat pada Gambar 2.11a yang mana terdapat *inner loop* dan *outer loop*.

Algoritma kontrol optimal *output feedback* dengan performa  $H_\infty$  digunakan pada kedua kontroler (*inner* dan *outer loop*) namun, pada *outer loop* ditambahkan desain *loop shaping*. Hal tersebut bertujuan untuk mengatasi *trade off* antara performa dan kekakuan dalam *loop shaping*.

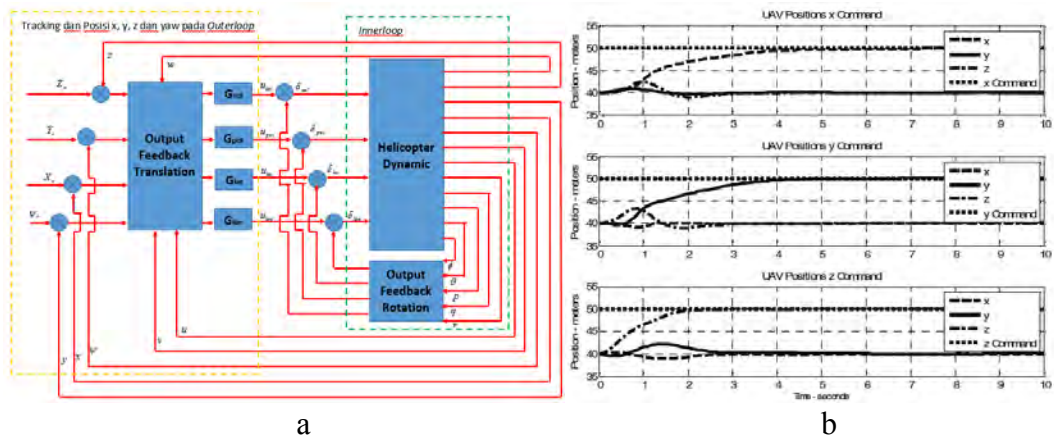
Tujuan yang didapat ditinjau dari persamaan *system* berikut:

$$\begin{aligned} \dot{x} &= Ax + Bu + Dd \\ y &= Cx \end{aligned} \tag{2.16}$$

dan performa *output* memenuhi

$$\|z(t)\|^2 = x^T Qx + u^T Ru \tag{2.17}$$

norm Euclidean dari *state* dan *input* sistem merupakan hasil dari performa *output*. Sistem  $L_2$  gain dikatakan dilemahkan oleh level pelemahan ( $\gamma$ ) jika



Gambar 2.11a. Struktur kontroler *inner* dan *outer loop* dan b. Respon posisi terhadap *command* yang diberikan

$$\frac{\int_0^{\infty} \|z(t)\|^2 dt}{\int_0^{\infty} \|d(t)\|^2 dt} = \frac{\int_0^{\infty} (x^T Q x + u^T R u) dt}{\int_0^{\infty} (d^T d) dt} \quad (2.18)$$

untuk setiap *input* gangguan  $d$ .

Untuk menghasilkan *gain* SOF sehingga  $A_0 = (A - BKC)$  adalah stabil *asymptotic* dengan  $L_2$  *gain* dilemahkan oleh ( $\gamma$ ), diasumsikan matriks ( $A, C$ ) adalah *detectable* dan terdapat matriks  $L$  dan  $P = P^T \geq 0$  sehingga *output feedback gain* adalah  $KC = R^{-1}(B^T P + L)$ .

Hasil simulasi pada paper ini menunjukkan bahwa sistem kontrol yang dirancang dapat untuk membuat quadcopter melakukan *tracking* dengan memenuhi kriteria performa  $H^\infty$ . Hal yang menarik dari paper ini yaitu penentuan performa *output* yang terdiri dari persamaan *state* dan *input* kontrol dari sistem.

## 2.2 Dasar Teori

Pada bab ini terdapat teori dasar yang menunjang dalam merumuskan dan menyelesaikan masalah yang dihadapi dalam pengerjaan tesis. Bagian awal terdapat teori tentang quadcopter, baik secara umum maupun konsep gerak dari quadcopter. Bagian selanjutnya membahas tentang teori optimal *output feedback* dengan  $H^\infty$ , LMI, teori dasar struktur *Command-Generator Tracker* model *Following, Path following*.

### 2.2.1 Quadcopter (Quanser Qball X4)

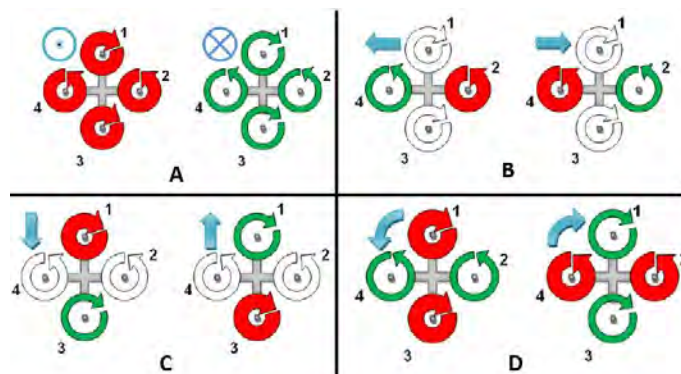
Quanser Qball X4 merupakan helikopter tanpa awak yang dikombinasikan dengan empat motor yang polanya diletakkan secara menyilang. Quadcopter menghasilkan gaya angkat sebesar nilai dari keempat motor. Dua motor saling

berlawanan membentuk satu pasang, dimana pasangan pertama (motor 1 dan motor 3) arah pergerakannya diatur searah dengan sumbu x dan berotasi searah dengan jarum jam sedangkan pasangan kedua (motor 2 dan motor 4) arah pergerakannya diatur searah dengan sumbu y dan berotasi berlawanan dengan jarum jam.

Quadcopter memiliki beberapa kelebihan dibandingkan dengan jenis wahana terbang/UAV yang lain diantaranya, dapat melakukan *Vertical Take-Off Landing* (VTOL), konstruksi yang lebih sederhana dibandingkan dengan helikopter konvensional, mampu terbang *hover*, tidak seperti UAV bertipe *fixed wing*, energi kinetik lebih kecil dibandingkan dengan helikopter konvensional sehingga lebih aman jika terjadi kecelakaan, memiliki kemampuan manuver yang lebih baik dibandingkan dengan UAV bertipe *fixed wing*. Selain itu quadcopter juga memiliki kekurangan diantaranya, memiliki sifat konsumsi energi yang relatif lebih besar jika dibandingkan dengan UAV tipe *fixed wing*, kontrol kestabilan dari quadcopter relatif lebih sulit dicapai karena sifat dari dinamika sistem yang *non linear*.

Gerakan dan kecepatan quadcopter ditentukan oleh kecepatan masing-masing motor. Selain mengendalikan kecepatan masing-masing motor, pada quadcopter perlu diperhatikan juga mengenai pengendalian dalam hal *attitude* quadcopter yaitu, gerakan akselerasi (*thrust*), gerakan mengguling (*roll*), gerakan mengangguk (*pitch*) dan gerakan memutar (*yaw*).

Gambar 2.12.A adalah gaya *thrust* ( $U_1$ ) dengan mempercepat atau memperlambat kecepatan motor dari seluruh *propeller* secara bersamaan akan menghasilkan percepatan vertical. Gambar 2.12.B adalah *Roll* ( $U_2$ ) dengan mempercepat atau memperlambat *propeller* pada sisi kiri dan secara bersamaan memperlambat atau mempercepat *propeller* di sisi kanan akan diperoleh rotasi dengan kecepatan sudut. Gambar 2.12.C merupakan *Pitch* ( $U_3$ ) yaitu mekanisme yang sama seperti gerak *roll*, namun dilakukan pada *propeller* di sisi depan dan belakang akan diperoleh percepatan sudut *pitch* dan *Yaw* ( $U_4$ ) pada Gambar 2.12.D adalah mempercepat atau memperlambat *propeller* yang berputar berlawanan arah jarum jam, dan memperlambat atau mempercepat *propeller* yang berputar searah jarum jam akan diperoleh gerak *yaw* dengan percepatan sudut.



Gambar 2.12 Thrust ( $U_1$ ), B. Roll ( $U_2$ ), C. Pitch ( $U_3$ ), dan D. Yaw ( $U_4$ )

### 2.2.2 Pemodelan Quadcopter

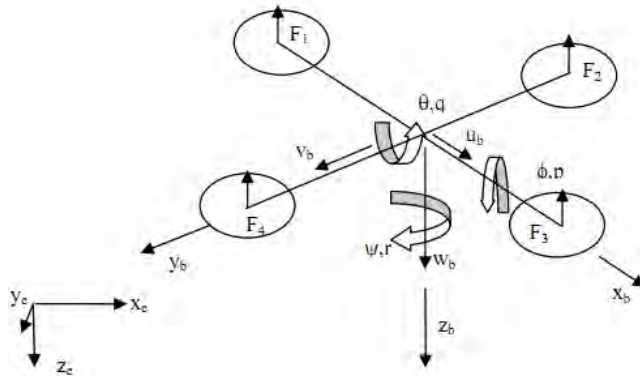
Pemodelan secara fisik terbilang kompleks apabila tanpa adanya asumsi yang digunakan untuk menyederhanakan persamaan pada quadcopter. Quadcopter memiliki 6 *degree of freedom* (DoF) dengan 12 keluaran, 6 keluaran dari 12 keluaran ini menentukan *attitude* dari quadcopter. Model kinematika dan dinamika quadcopter diturunkan berdasarkan Newton-Euler dengan beberapa asumsi penyederhanaan yang harus dipahami [14]:

1. Percepatan gravitasi konstan dan tegak lurus terhadap permukaan bumi.
2. Desain quadcopter yang dibuat, dianggap simetris.
3. Struktur body dan baling-baling dari quadcopter merupakan benda kaku (rigid) sehingga pemodelan dapat menggunakan pendekatan Newton Euler.
4. Struktur frame body dari *yaw* quadcopter bersifat simetris sepanjang sumbu  $x$  dan  $y$ .
5. Qball quadcopter merupakan quadcopter yang digunakan dalam ruangan sehingga kecepatan dianggap rendah, gesekan udara diabaikan, efek gyroscopic dan torsi aerodinamis dapat diabaikan.
6. gaya *thrust* dan drag adalah proporsional dengan kuadrat dari kecepatan propeller. Variabel-variabel tersebut ditunjukkan pada Tabel 2.1.

Untuk menentukan acuan-acuan dinamika pada quadcopter perlu 2 sistem koordinat yang dibutuhkan untuk mendefinisikan keadaan dari quadcopter. Pertama, sumbu  $X_b$ ,  $Y_b$ , dan  $Z_b$  yang terletak pada titik gravitasi (COG) quadcopter.

Tabel 2.1 Variabel pada Pergerakan Quadcopter [14]

Variabel	Keterangan
$X$	Posisi quadcopter terhadap sumbu $X_u$
$Y$	Posisi quadcopter terhadap sumbu $Y_u$
$Z$	Posisi quadcopter terhadap sumbu $Z_u$
$U$	Kecepatan quadcopter yang diukur pada sumbu $X_b$
$V$	Kecepatan quadcopter yang diukur pada sumbu $Y_b$
$W$	Kecepatan quadcopter yang diukur pada sumbu $Z_b$
$\phi$	Sudut <i>roll</i> terhadap sumbu $X_u$
$\theta$	Sudut <i>pitch</i> terhadap sumbu $Y_u$
$\psi$	Sudut <i>yaw</i> terhadap sumbu $Z_u$
$P$	Kecepatan sudut <i>roll</i> yang diukur pada sumbu $X_b$
$Q$	Kecepatan sudut <i>pitch</i> yang diukur pada sumbu $Y_b$
$R$	Kecepatan sudut <i>yaw</i> yang diukur pada sumbu $Z_b$



Gambar 2.13 Gambar Semua *State* dari Quadcopter (e untuk *Frame* Bumi dan b untuk *Frame* Badan Quadcopter)

Kedua, sistem koordinat yang tetap yaitu pusat bumi yang mengacu pada utara-timur-bawah ( $X_E, Y_E, Z_E$ ). Jadi ada 2 *frame*, yaitu badan *frame* (*frame* badan quadcopter) dan *earth frame* (*frame* bumi) dan sudut rotasi *roll*( $\phi$ ), *pitch*( $\theta$ ), dan *yaw*( $\psi$ ). Acuan ini ditunjukkan pada Gambar 2.13.

Jika didefinisikan sumbu  $X$  menuju utara dan depan quadcopter, sumbu  $Z$  menjauhi pusat bumi dan menuju atas quadcopter, maka berdasarkan kaidah tangan kanan *cross vector* dapat ditentukan bahwa sumbu  $Y$  menuju barat atau kiri quadcopter. Sehingga  ${}^B V$  sebagai kecepatan vektor,  ${}^B \omega$  kecepatan sudut pada *frame* badan dan  ${}^u X$  posisi quadcopter adalah:

$${}^B \bar{V} = u\bar{i} + v\bar{j} + w\bar{k} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.19)$$

$${}^B \bar{\omega} = p\bar{i} + q\bar{j} + r\bar{k} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.20)$$

$${}^u \bar{x} = x\bar{i} + y\bar{j} + z\bar{k} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.21)$$

Sehingga sudut *euler* pada quadcopter yang berotasi pada sumbu  $X, Y, Z$  (*frame* badan) dapat di repressetasikan

$$\Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (2.22)$$

Posisi linier quadcopter ditentukan dari koordinat vektor antara origin *frame body* serta origin *frame* bumi. Posisi linier dan posisi angular terdapat pada persamaan

$$v = (X, Y, Z)^T \quad (2.23)$$

$$\Theta = (\phi, \theta, \psi)^T \quad (2.24)$$

sedangkan enam derajat kebebasan quadcopter digambarkan oleh tiga buah translasi  $V = (u, v, w)^T$  dan tiga buah rotasi  $\Omega = (p, q, r)^T$ .

### A. Matriks transformasi

Sebuah matriks transformasi yang digunakan untuk mengtransformasi vector *state* dari *frame*-E menuju *frame*-B. Matrik dari sumbu  $X, Y, Z$  dapat ditemukan berdasarkan kaidah dimana rotasi terhadap  $X$ , rotasi terhadap  $Y$ , dan rotasi terhadap  $Z$  sehingga menggunakan persamaan:

$${}^B_E R = R_{x,\phi} * R_{y,\theta} * R_{z,\psi} \quad (2.25)$$

dimana  ${}^B_E R = R_{x,\phi} * R_{y,\theta} * R_{z,\psi}$  matrik rotasi mengelilingi tiap sumbu. sehingga mengikuti:

- **Matriks rotasi terhadap sumbu  $X$**

Nilai matriks rotasi terhadap sumbu  $X$  sebagai berikut:

$$X_E = X_B \cos 0^\circ + Y_B \cos 90^\circ + Z_B \cos 90^\circ \quad (2.26)$$

$$Y_E = X_B \cos 90^\circ + Y_B \cos \phi + Z_B \cos(90^\circ + \phi)$$

$$Z_E = X_B \cos 90^\circ + Y_B \cos(90^\circ - \phi) + Z_B \cos(\phi)$$

sehingga dalam bentuk matriks menjadi:

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.27)$$

Jika dibawa kedalam bentuk matrik rotasi menjadi:

$$R_{(x,\phi)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.28)$$

dengan sudut *yaw*  $\psi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$

- **Matriks rotasi terhadap sumbu  $Y$**

Nilai matriks rotasi terhadap sumbu  $Y$  sebagai berikut:

$$X_E = X_B \cos \theta + Y_B \cos 90^\circ + Z_B \cos(90^\circ - \theta) \quad (2.29)$$

$$Y_E = X_B \cos 90^\circ + Y_B \cos 0^\circ + Z_B \cos 90^\circ$$

$$Z_E = X_B \cos(90^\circ + \theta) + Y_B \cos 90^\circ + Z_B \cos(\theta)$$

sehingga dalam bentuk matriks menjadi:

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.30)$$

Jika dibawa kedalam bentuk matrik rotasi menjadi:

$$R_{(Y,\theta)} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.31)$$

dengan sudut *yaw*  $\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$

- **Matriks rotasi terhadap sumbu  $Z$**

Nilai matriks rotasi terhadap sumbu  $Z$  sebagai berikut:

$$X_E = X_B \cos \psi + Y_B \cos(90^\circ + \psi) + Z_B \cos 90^\circ \quad (2.32)$$

$$Y_E = X_B \cos(90^\circ + \psi) + Y_B \cos \psi + Z_B \cos 90^\circ$$

$$Z_E = X_B \cos 90^\circ + Y_B \cos 90^\circ + Z_B \cos 0$$

sehingga dalam bentuk matriks menjadi:

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.33)$$

Jika dibawa kedalam bentuk matrik rotasi menjadi:

$$R_{(z,\psi)} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.34)$$

dengan sudut *yaw*  $\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$

Jika menggunakan (2.27), (2.30) dan (2.33) maka bentuk transformasi matriks dapat diperoleh sebagai berikut:

$${}^B_E R = R_{(X,\phi)}, R_{(Y,\theta)}, R_{(Z,\psi)} \quad (2.35)$$

$${}^B_E R = \begin{bmatrix} c\theta c\psi & s\psi c\theta & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}$$

dengan  $s$  adalah notasi dari  $\sin$ ,  $c$  adalah notasi dari  $\cos$ . Sehingga jika transformasi matrik dari *frame* bodi menuju *frame* bumi maka persamaannya menjadi berikut:

$${}^E_B R = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta s\psi & s\phi s\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (2.36)$$



## B. Kinematika Quadcopter

Transformasi matriks yang sudah terdefinisi maka, dapat ditemukan kecepatan *vehicle* pada *frame* inersia yaitu koordinat badan sehingga persamaan dapat diberikan:

$${}^e \dot{x} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = {}^e R^B V = {}^e R \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (2.37)$$

$$= \begin{bmatrix} (c\theta c\psi)u + (-c\phi s\psi + s\phi s\theta s\psi)v + (s\phi s\psi + c\phi s\theta c\psi)w \\ (c\theta s\psi)u + (c\phi c\psi + s\phi s\theta s\psi)v + (-s\phi c\psi + c\phi s\theta s\psi)w \\ (s\theta)u + (s\phi c\theta)v + (c\phi c\theta)w \end{bmatrix}$$

dengan  $s$  adalah notasi dari sin,  $c$  adalah notasi dari cos.

Maka statenya:

$$\dot{\Lambda} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = Q^{-1}(\Lambda)^B \omega \quad (2.38)$$

dimana matriks  $Q(\Lambda)$  dan matriks invers  $Q^{-1}(\Lambda)$ :

$$Q(\Lambda) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & -c\phi c\theta \end{bmatrix} \quad (2.39)$$

$$Q^{-1}(\Lambda) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \quad (2.40)$$

dengan  $s$  adalah notasi dari sin,  $c$  adalah notasi dari cos dan  $t$  adalah notasi dari tan. Persamaan (2.20) disubstitusikan ke (2.38) sehingga persamaannya:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.41)$$

Sehingga (2.37) dan (2.41) adalah persamaan kinematik dari quadcopter. Model kinematika dan dinamika quadcopter diturunkan berdasarkan formula Newton-Euler dengan beberapa asumsi penyederhanaan. Percepatan gravitasi dianggap konstan dan tegak lurus terhadap permukaan bumi, posisi pusat massa tepat berada di tengah, efek momentum badan quadcopter pada gerakan translasi diabaikan ketika quadcopter bergerak perlahan. *Frame* yang digunakan pada quadcopter dianggap *rigid* dan simetris.

### C. Dinamika Quadcopter

Jenis pesawat seperti helikopter atau quadcopter dipengaruhi banyak efek fisik ketika melakukan penerbangan. Efek aerodinamis akibat rotasi baling-baling dan *blade flapping* dapat mempengaruhi sistem ini. Mengubah kecepatan dari putaran baling-baling akan menyebabkan *inertial counter torques*. Selain itu ketika helikopter atau quadcopter berputar atau bergerak maju, terjadi efek *gyroscopic* dan gesekan yang mempengaruhi seluruh sistem. Oleh karena itu, pemodelan dinamika dari sistem ini terbilang kompleks.

#### a. Model Dinamika Translasi Quadcopter

Suatu gaya yang dihasilkan oleh keempat motor penggerak yang dipasang baling-baling pada quadcopter. Sebelum menganalisa model dinamika dari quadcopter, akan dijelaskan mengenai gaya dan momen aerodinamis pada quadcopter yang bernilai kecil sehingga dapat diabaikan dalam proses pemodelan. Persamaan berikut menunjukkan gaya-gaya yang terjadi [14]:

$$U_1 = F_1 + F_2 + F_3 + F_4 \quad (2.42)$$

$$U_2 = F_2 - F_4 \quad (2.43)$$

$$U_3 = F_2 - F_4 \quad (2.44)$$

$$U_4 = F_1 + F_3 - F_2 - F_4 \quad (2.45)$$

dengan  $d$  adalah konstanta gaya drag,  $\ddot{x}$  adalah percepatan *vehicle* dan  $F_i$  adalah gaya angkat (*thrust*) yang dihasilkan oleh tiap motor.

$$\sum F_i = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \frac{d}{dt}(mV) = m \frac{d\vec{V}}{dt} = m\ddot{x} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (2.46)$$

jika dilipat gandakan keduanya pada matrik transformasi dari  ${}^B_R$  pada (2.35), maka (2.46) dengan  $K_T$  adalah konstanta gaya dorong,  $\omega$  adalah lebar *bandwidth* motor dan  $u_i$  adalah sinyal kontrol dari *innerloop* kontroler ke motor.

$$\sum F_i = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \frac{d}{dt}(mV) = m \frac{d\vec{V}}{dt} = m\ddot{x} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (2.47)$$

$$\sum F_i = m {}^B_R \frac{d}{dt}(\dot{x}) \quad (2.48)$$

$$\sum F_i = m {}^B_R \frac{d}{dt}({}^B_R V) = m {}^B_R ({}^B_R \dot{V} + {}^B_R V) = m ({}^B_R {}^E_B \dot{V} + {}^B_R {}^E_B V) \quad (2.49)$$

$$\sum F_i = m (\dot{V} + {}^B_R {}^E_B V) = m (\dot{V} + {}^B_R [{}^E_B V S \omega] V)$$

$$\sum F_i = (mV + S\omega V)$$

$$F_i = K_T \frac{\omega}{s + \omega} (u_i) \quad (2.50)$$

Dinamika gerak translasi diperoleh berdasarkan persamaan hukum Newton II, yaitu

$$\begin{aligned}\sum F &= m\ddot{v} \\ U_1 + F_g z_E &= m\ddot{v}\end{aligned}\quad (2.51)$$

dengan  $m$  adalah massa quadcopter, posisi pusat massa yang berada pada *frame* bumi dinyatakan dengan  $v = [x \ y \ z]^T$  total gaya angkat (*thrust*) yang dihasilkan oleh keempat motor dinyatakan dengan  $U_i$ , sedangkan gaya gravitasi dinyatakan oleh  $F_g = -mg$  dan  $Z_g = [0 \ 0 \ 1]^T$ .

Pemodelan sumbu translasi terletak pada koordinat bumi, untuk transformasi dari koordinat *frame body* ke koordinat *frame* bumi maka diperlukan matriks rotasi seperti pada (2.36). Persamaan (2.51) dapat ditulis menjadi karena gaya pada quadcopter hanya terjadi pada sumbu  $Z_B$  (pada kondisi *hover*),

$$\ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \frac{U_1}{m} \quad (2.52)$$

$$\ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \frac{U_1}{m} \quad (2.53)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{U_1}{m} \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi s\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.54)$$

dan total thrust dinotasikan oleh  $U_i$  seperti pada (2.42), sehingga persamaan gerak translasi quadcopter dinyatakan sebagai berikut:

$$U_1 {}^E_B R + F_g z_E = m\ddot{v} \quad (2.55)$$

$$\frac{U_1}{m} {}^E_B R - gz_E = \ddot{v} \quad (2.56)$$

$$\ddot{z} = -g + (\cos\phi \cos\theta) \frac{U_1}{m} \quad (2.57)$$

### b. Model Dinamika Rotasi Quadcopter

Persamaan gerak rotasi diperoleh dalam *frame body* menggunakan metode Euler dengan formula berikut:

$$\sum {}^E \tau = J^E \ddot{\Theta} \quad (2.58)$$

$$\sum {}^E \tau = \frac{d}{dt} ({}^E R J {}^B \Omega) = J ({}^E \dot{R} \Omega + {}^E R \dot{\Omega}) \quad (2.59)$$

karena  $\dot{R} = RS({}^B \Omega)$  dimana  $S({}^B \Omega)$  adalah matriks *skew-symmetric* yang dinyatakan oleh

$$S({}^B \Omega) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.60)$$

jika ruas kiri dikalikan dengan  ${}^B_E R$  maka (2.59) menjadi

$$\begin{aligned}\sum {}^B_E R^E \tau &= J({}^B_E \dot{R}^B \Omega + {}^B_E R^B \dot{\Omega}) \\ \sum {}^B_E R^E \tau &= J {}^B_E R S({}^B \Omega)^B \Omega + J {}^B_E R^B \dot{\Omega} \\ \sum {}^B \tau &= \Omega x J \Omega + J \dot{\Omega}\end{aligned}\quad (2.61)$$

Sehingga dapat ditulis sebagai berikut: (2.63)

$$\begin{aligned}J \dot{\Omega} &= -\Omega x J \Omega + \tau_f \\ \dot{\Omega} &= \frac{-\Omega x J \Omega + \tau_f}{J}\end{aligned}\quad (2.62)$$

dengan  $\tau_f$  merupakan resultan torsi yang dihasilkan oleh masing-masing motor.

### c. Torsi

Quadcopter memiliki batasan rotor yang rigid, sehingga dinamika setiap rotor disekitar sumbu rotasinya dapat diperlakukan sebagai sistem *decouple* dalam variabel kecepatan angular  $\Omega$  secara umum. Torsi yang digunakan oleh setiap motor dinotasikan oleh  $\tau_{f-i}$ . Dalam quadcopter, torsi motor berlawanan dengan gaya aerodinamik drag  $\tau_{drag} = d\Omega^2$ . Menggunakan Hukum II Newton,

$$I_M \dot{\Omega}_i = -\tau_{drag} + \tau_{f-i} \quad (2.63)$$

dimana  $I_M$  adalah momen kecepatan sudut dari motor ke- $i$  dan  $d > 0$  merupakan konstanta manuver stationer dalam penerbangan bebas. Dalam keadaan *steady state*, yaitu ketika  $\dot{\Omega} = 0$  torsi yaw adalah

$$\tau_{f-i} = d\Omega^2 \quad (2.64)$$

sehingga torsi dalam *frame body* secara umum dapat ditulis menjadi

$$\tau_f = \begin{bmatrix} \sum_{i=1}^4 \tau_{f-i} \\ l(F_2 - F_4) \\ l(F_1 - F_3) \end{bmatrix} = \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \quad (2.65)$$

dengan  $l$  adalah jarak antara motor dan pusat gravitasi dan  $\tau_\phi, \tau_\theta, \tau_\psi$  adalah momen yawing, pitching dan rolling, sehingga dapat ditulis menjadi persamaan berikut:

$$\begin{aligned}\tau_{f-x} &= l(F_2 - F_4) \\ \tau_{f-y} &= l(F_1 - F_3) \\ \tau_{f-z} &= d(F_1 + F_3 - F_2 - F_4)\end{aligned}\quad (2.66)$$

dengan  $l$  adalah jarak antara pusat massa quadcopter dengan sumbu rotasi balingbaling dan  $d$  adalah koefisien gaya drag.

Matriks  $J \in R^{3 \times 3}$  adalah matriks inersia pada *frame body* yang didefinisikan oleh

$$J = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \quad (2.67)$$

nilai  $J_{xy} = J_{xz} = J_{yx} = J_{yz} = J_{zx} = J_{zy} = 0$  karena quadcopter simetris, sehingga *invers* dari  $J$  adalah

$$J^{-1} = \begin{bmatrix} \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \quad (2.68)$$

Kecepatan sudut dinotasikan oleh  $\Omega$  yang didefinisikan sebagai

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.69)$$

Nilai  $-\Omega x J \Omega$  dinyatakan dalam (2.69) yang dapat dihitung dengan mensubstitusi (2.65) dan (2.66).

$$-\Omega x J \Omega = J \Omega x \Omega \quad (2.70)$$

$$= \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} x \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$= \dot{\phi} \dot{\theta} (J_{xx} - J_{yy}) k + \dot{\phi} \dot{\psi} (J_{zz} - J_{xx}) j + \dot{\theta} \dot{\psi} (J_{yy} - J_{zz}) i$$

atau

$$-\Omega x J \Omega = \begin{bmatrix} \dot{\phi} \dot{\theta} (J_{xx} - J_{yy}) \\ \dot{\phi} \dot{\psi} (J_{zz} - J_{xx}) \\ \dot{\theta} \dot{\psi} (J_{yy} - J_{zz}) \end{bmatrix}$$

Moment gaya  $\tau_j$  pada (2.64) dan  $-\Omega x J \Omega$  pada (2.70) disubstitusi kedalam (2.78) maka dinamika gerak rotasi quadcopter menjadi seperti berikut:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{\phi} \dot{\theta} (J_{xx} - J_{yy}) \\ \dot{\phi} \dot{\psi} (J_{zz} - J_{xx}) \\ \dot{\theta} \dot{\psi} (J_{yy} - J_{zz}) \end{bmatrix} + \begin{bmatrix} l(F_{\tau 2} - F_{\tau 4}) \\ l(F_{\tau 1} - F_{\tau 3}) \\ d(F_{\tau 1} + F_{\tau 3} - F_{\tau 2} - F_{\tau 4}) \end{bmatrix} \quad (2.71)$$

Jika (2.42)-(2.45) disubstitusi kedalam (2.71) dapat diperoleh persamaan lengkap dinamika gerak rotasi quadcopter adalah sebagai berikut:

$$\ddot{\phi} = \frac{J_{yy} - J_{zz}}{J_{xx}} qr + \frac{Jr}{J_{xx}} q\Omega + \frac{U_2}{J_{xx}} \quad (2.72)$$

$$\ddot{\theta} = \frac{J_{zz} - J_{xx}}{J_{yy}} pr - \frac{Jr}{J_{yy}} p\Omega + \frac{U_3}{J_{yy}} \quad (2.73)$$

$$\ddot{\psi} = \frac{J_{xx} - J_{yy}}{J_{zz}} pq + \frac{U_4}{I_{zz}} \quad (2.74)$$

dan persamaan dinamika dari quadcopter pada gerak translasi dan rotasi dituliskan pada Persamaan (2.52), (2.53), (2.57), dan (2.72)-(2.74) [14].

$$\ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \frac{U_1}{m} \quad (2.75)$$

$$\ddot{y} = (\cos\phi \sin\theta \cos\psi - \sin\phi \sin\psi) \frac{U_1}{m} \quad (2.76)$$

$$\ddot{z} = -g + (\cos\phi \cos\theta) \frac{U_1}{m} \quad (2.77)$$

$$\ddot{\phi} = \frac{J_{yy} - J_{zz}}{J_{xx}} qr + \frac{Jr}{J_{xx}} q\Omega + \frac{U_2}{J_{xx}} \quad (2.78)$$

$$\ddot{\theta} = \frac{J_{zz} - J_{xx}}{J_{yy}} pr - \frac{Jr}{J_{yy}} p\Omega + \frac{U_3}{J_{yy}} \quad (2.79)$$

$$\ddot{\psi} = \frac{J_{xx} - J_{yy}}{J_{zz}} pq + \frac{U_4}{I_{zz}} \quad (2.80)$$

dengan  $m$  adalah massa,  $g$  adalah gravitasi,  $l$  adalah jarak antara pusat massa quadcopter dengan sumbu rotasi *propeller*,  $d$  adalah koefisien gaya drag, dan  $I_x$ ,  $I_y$ ,  $I_z$  adalah inersia pada sumbu  $x$ ,  $y$ ,  $z$  berturut-turut.

### 2.2.3 Linearisasi

Sistem nonlinear memiliki titik ekuilibrium lebih dari satu, tidak seperti sistem yang linear. Titik ekuilibrium sendiri merupakan titik dimana perubahan *state* dari sistem bernilai nol. Sifat nonlinear dibagi menjadi dua, yaitu *inherent (natural)* dan *intentional (artificial)*. Sifat nonlinear *inherent* adalah sifat nonlinear yang muncul akibat aktivitas gerak dari sistem, seperti gaya gesek, histerisis, dan saturasi. Sedangkan sifat nonlinear *intentional* merupakan sifat nonlinear yang sengaja dimunculkan. Pada umumnya untuk menyelesaikan permasalahan sebuah sistem yang mengandung sifat nonlinear secara matematis sangat sulit dan memerlukan waktu yang cukup lama. Oleh karena itu linearisasi merupakan metode yang cukup efektif untuk menyelesaikan permasalahan nonlinearitas dengan melakukan pendekatan kedalam beberapa daerah kerja yang terbatas, sehingga analisa kestabilan pada sistem linear bisa diterapkan dan memudahkan dalam proses desain.

Sistem dapat direpresentasikan kedalam persamaan diferensial seperti pada (2.81).

$$\left. \begin{aligned} \dot{x}_1 &= f_1(x_1, \dots, x_n, u_1, \dots, u_m) \\ \dot{x}_2 &= f_2(x_1, \dots, x_n, u_1, \dots, u_m) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, \dots, x_n, u_1, \dots, u_m) \end{aligned} \right\} \quad (2.81)$$

$$\left. \begin{aligned} y_1 &= h_1(x_1, \dots, x_n, u_1, \dots, u_m) \\ y_2 &= h_2(x_1, \dots, x_n, u_1, \dots, u_m) \\ &\vdots \\ y_p &= h_p(x_1, \dots, x_n, u_1, \dots, u_m) \end{aligned} \right\} \quad (2.82)$$

Jika vektor kolom  $x = [x_1, \dots, x_n]^T$  merupakan vektor state sistem, vektor kolom  $u = [u_1, \dots, u_m]^T$  merupakan sinyal kontrol dan vektor kolom  $y = [y_1, \dots, y_p]^T$  merupakan keluaran sistem, maka (2.81) dan (2.82) dapat disederhakan menjadi (2.83) dan (2.84).

$$f(x, u) = \begin{bmatrix} f_1(x_1, \dots, x_n, u_1, \dots, u_m) \\ f_2(x_1, \dots, x_n, u_1, \dots, u_m) \\ \vdots \\ f_n(x_1, \dots, x_n, u_1, \dots, u_m) \end{bmatrix} \quad (2.83)$$

$$h(x, u) = \begin{bmatrix} h_1(x_1, \dots, x_n, u_1, \dots, u_m) \\ h_2(x_1, \dots, x_n, u_1, \dots, u_m) \\ \vdots \\ h_p(x_1, \dots, x_n, u_1, \dots, u_m) \end{bmatrix} \quad (2.84)$$

sehingga dapat ditulis:

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x, u) \end{aligned} \quad (2.85)$$

Jika (2.85) merupakan sistem linear, maka sistem tersebut dapat ditulis dalam representasi *state space*:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (2.86)$$

dengan  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $D \in \mathbb{R}^{p \times m}$ .

Suatu sistem nonlinear dinyatakan dalam bentuk sebagai berikut

$$\dot{x} = f(x) \quad (2.87)$$

dimana  $y = f(x)$  merupakan suatu kurva nonlinear. Jika  $f(x)$  direpresentasikan dalam pendekatan linear di sekitar titik  $x^*$  maka dapat ditulis

$$\delta y = f \delta x = \left. \frac{df(x)}{dx} \right|_{x=x^*} \delta x \quad (2.88)$$

dengan  $\delta y = f(x) - f(x^*)$  dan  $\delta x = x - x^*$ . Apabila sistem bekerja pada titik origin (0,0), maka;

$$\delta y = f(x) - f(x^*) \quad (2.89)$$

$$\delta y = f(x) - 0$$

$$\delta y = f(x)$$

$$\delta x = x - x^*$$

$$\delta x = x - 0$$

$$\delta x = x$$

$$y = \left. \frac{df(x)}{dx} \right|_{x=x^*} x$$

Jika  $f$  dan  $h$  pada (2.85) merupakan sistem nonlinear dari  $x$  dan  $u$ , kemudian dilakukan pendekatan linear disekitar titik  $x^* = [x_1^*, \dots, x_n^*]^T$ , dengan titik tersebut diperoleh pada saat  $u^* = [u_1^*, \dots, u_m^*]^T$ , maka hasil linearisasi lokal (2.87) berdasarkan (2.89) dapat dituliskan dengan (2.90-2.93) sebagai berikut:

$$A = \left. \frac{df(x, u)}{dx} \right|_{x^*, u^*} = \left[ \begin{array}{ccc} \frac{df_1(x, u)}{dx_1} & \dots & \frac{df_1(x, u)}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n(x, u)}{dx_1} & \dots & \frac{df_n(x, u)}{dx_n} \end{array} \right]_{x^*, u^*} \quad (2.90)$$

$$B = \left. \frac{df(x, u)}{du} \right|_{x^*, u^*} = \left[ \begin{array}{ccc} \frac{df_1(x, u)}{du_1} & \dots & \frac{df_1(x, u)}{du_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n(x, u)}{du_1} & \dots & \frac{df_n(x, u)}{du_n} \end{array} \right]_{x^*, u^*} \quad (2.91)$$

$$C = \left. \frac{dh(x, u)}{dx} \right|_{x^*, u^*} = \left[ \begin{array}{ccc} \frac{dh_1(x, u)}{dx_1} & \dots & \frac{dh_1(x, u)}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{dh_n(x, u)}{dx_1} & \dots & \frac{dh_n(x, u)}{dx_n} \end{array} \right]_{x^*, u^*} \quad (2.92)$$

$$D = \left. \frac{dh(x, u)}{du} \right|_{x^*, u^*} = \left[ \begin{array}{ccc} \frac{dh_1(x, u)}{du_1} & \dots & \frac{dh_1(x, u)}{du_n} \\ \vdots & \ddots & \vdots \\ \frac{dh_n(x, u)}{du_1} & \dots & \frac{dh_n(x, u)}{du_n} \end{array} \right]_{x^*, u^*} \quad (2.93)$$

## 2.2.4 Analisa Kestabilan

Dinamika sistem linier dapat di representasikan sebagai berikut:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.94)$$



$$y(t) = Cx(t) \quad (2.95)$$

dimana  $x(t) \in R^n$  adalah vektor *state* dari sistem,  $u(t) \in R^m$  adalah vektor *input*,  $A \in R^{n \times n}$  adalah matriks *state* dan  $B(t) \in R^{n \times m}$  adalah matriks *input*.

Hal yang perlu diperhatikan dalam sistem linier adalah kontrolabilitas dari sistem tersebut. Jika sistem *controllable* maka akan mudah untuk membawa sistem dari nilai inisial ke nilai akhir. Sistem (2.94) dan (2.95) dikatakan *controllable* jika dan hanya jika [7]:

$$\text{rank}([B \ AB \ A^2B \ \dots \ A^{n-1}B]) = n \quad (2.96)$$

Kestabilan dan karakteristik natural respon dari sistem linier dapat dipelajari dengan melihat *pole-pole* dari sistem. Pada persamaan *state space*, *pole* sistem sama dengan *eigenvalue* dari matriks A.

Alternatif lain untuk melakukan analisa kestabilan sistem dapat menggunakan teori kestabilan Lyapunov. Berdasarkan metode Lyapunov, sistem dapat dikatakan stabil *asymptotic* dalam pengertian Lyapunov jika memungkinkan untuk menemukan sebuah fungsi  $V(x) : R^n \rightarrow R$ . Pada sistem linier, fungsi kandidat Lyapunov dinyatakan dengan  $V(x) = x^T P x$  dimana  $P \in R^{n \times n}$  adalah matriks simetris. Kondisi untuk memenuhi batasan pada teorema di atas adalah [14]:

- Batasan Teorema 1 selalu terpenuhi untuk  $V(x)$  jika titik equilibrium sistem berada pada *origin* ( $x=0$ ), maka fungsi Lyapunov pada titik equilibrium adalah  $0^T P 0 = 0, \forall P$
- Batasan Teorema 2, sistem *asymptotic* global apabila terdapat matriks  $P$  positif definit, atau dapat ditulis:

$$P > 0, \quad (2.97)$$

- Pada teorema 3, persamaan  $\dot{V}(x)$  dinyatakan sebagai:

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} = (Ax + Bu)^T P x + x^T P (Ax + Bu) \quad (2.98)$$

dengan memilih  $u = 0$ , maka analisa kestabilan sistem *open-loop* menjadi:

$$\dot{V}(x) = x^T [A^T P + PA] x < 0 \quad (2.99)$$

Maka teorema 3 terpenuhi jika dan hanya jika memenuhi pertidaksamaan:

$$A^T P + PA < 0 \quad (2.100)$$

jika *input*  $u = -Kx$ , maka teorema 3 terpenuhi jika dan hanya jika sistem linier memenuhi pertidaksamaan berikut:

$$(A - BK)^T P + P(A - BK) < 0; \quad (2.101)$$

## 2.3 Kontroler

Kontroler merupakan salah satu komponen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*set point*) atau sinyal *error* mejadi sinyal kontrol.

### 2.3.1 Linear Matrix Inequality (LMI).

Kestabilan sistem dapat diketahui jika terdapat fungsi Lyapunov yang memenuhi (2.97)-(2.98). Jika fungsi Lyapunov didefinisikan dalam  $V(x) = x^T P x$ , maka sistem dikatakan stabil jika memenuhi (2.97) dan  $P > 0$ . Untuk sistem yang sederhana, matriks  $P$  dapat dicari melalui perhitungan analitik. Namun untuk *system* orde tinggi, perhitungan analitik tidak mudah untuk dilakukan. Perhitungan numerik menjadi pilihan terbaik dengan menggunakan metode *Linear Matrix Inequalities* (LMI). LMI memiliki bentuk seperti berikut: [12]:

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.102)$$

dengan  $F_i = F_i^T \in R^{n \times n}$ ,  $i = 0, 1, \dots, m$  adalah matriks yang diketahui  $x \in R^m$  adalah variabel. Terlihat dari (2.102), LMI merupakan fungsi kendala dari  $\mathcal{X}$ . LMI adalah sebuah metode yang digunakan untuk mencari nilai  $\mathcal{X}$  sedemikian hingga  $F(x) > 0$ . Jika terdapat lebih dari satu LMI, seperti  $F_1(x) < 0, \dots, F^{(p)}(x) < 0$  maka beberapa LMI tersebut dapat disederhanakan menjadi satu LMI saja, seperti pertidaksamaan berikut: ini:

$$\bar{F}(x) = \text{diag}\left[F^{(1)} \quad \dots \quad F^{(p)}(x)\right] < 0 \rightarrow \begin{cases} F^{(1)}(x) < 0 \\ \vdots \\ F^{(p)}(x) < 0 \end{cases} \quad (2.103)$$

Oleh karena itu, penyelesaian beberapa LMI dapat disederhanakan menjadi satu  $\bar{F}(x) < 0$ . Kendala pertidaksamaan pada contoh diatas merupakan kombinasi *linear* dari variabel  $\mathcal{X}$ . Jika kendala merupakan pertidaksamaan yang *nonlinear*, maka kendala tersebut dapat diubah menjadi LMI dengan menggunakan *Schur Complement*. Contoh jika terdapat LMI

$$L(x) = \begin{bmatrix} E(x) & F(x) \\ G(x) & H(x) \end{bmatrix} < 0 \quad (2.104)$$

dengan  $E(x) = E(x)^T$ ,  $H(x) = H(x)^T$ , dan  $G(x) = F(x)^T$  maka LMI (2.102) merupakan penyederhanaan dari fungsi kendala pertidaksamaan *nonlinear* pada (2.92).

$$L(x) = \begin{bmatrix} E(x) & F(x) \\ G(x) & H(x) \end{bmatrix} < 0 \quad (2.105)$$

### 2.3.2 Kontrol Static Output Feedback dengan performa $H_\infty$

*Static Output Feedback* (SOF) merupakan salah satu dari banyak permasalahan dalam teori sistem kontrol. Penggunaan *output feedback* memungkinkan fleksibilitas dan kesederhanaan dalam implementasi. Terutama, dalam aplikasi yang pengukuran *full-state* biasanya tidak dapat dilakukan. Keuntungan dari desain SOF yaitu hanya memerlukan sinyal yang terukur dari *plant* untuk dikontrol[6].

Pada sub-bab ini akan dijelaskan mengenai metode untuk menemukan *gain* kontroler SOF yang memiliki performa  $H_\infty$  untuk permasalahan regulator. *Gain* kontroler dihitung dengan teknik numerik iterative LMI. Pertimbangkan sistem *linear time-invariant*, dengan *input* kontrol  $u(t)$ , *output*  $y(t)$ , dan gangguan  $w(t)$  diketahui sebagai berikut:

$$\dot{x}(t) = Ax(t) + Bu(t) + Dw(t) \quad (2.106)$$

$$y(t) = Cx(t) \quad (2.107)$$

dengan performa *output*  $z(t)$ , sebagai berikut: [7]:

$$\|z(t)\|_2 = (x^T Qx + u^T Ru)dt \quad (2.108)$$

dimana  $Q > 0, R > 0$ .

Untuk permasalahan *output feedback*, solusi permasalahan kontrol LQ regulator pada (2.106) dan (2.107) adalah

$$u(t) = -Ky(t) = -KCx(t) \quad (2.109)$$

dengan definisi matriks  $(A, B)$  dikatakan *stabilizable* jika terdapat matriks  $K$  yang mana  $A - BK$  adalah stabil *asymptotic*. Matriks  $(A, C)$  dikatakan *detectable* jika terdapat matriks  $L$  yang mana  $A - LC$  adalah stabil.

Diasumsikan matriks  $(A, C)$  adalah *detectable* dan terdapat matriks  $L$  dan  $P = P^T \geq 0$  sehingga *output feedback gain* adalah

$$KC = R^{-1}(B^T P + L) \quad (2.110)$$

Kontroler ini didesain untuk meminimalkan pengaruh terburuk dalam performa *output* ketika terdapat gangguan yang diberikan pada sistem. Pengaruh terburuk gangguan terhadap performa *output* harus dilemahkan pada level pelemahan ( $\gamma$ ) tertentu. Pada [9] didefinisikan bahwa *norm- $\infty$*  dari fungsi alih  $T_{zw}(s)$  dapat dihitung dari nilai maksimal  $L_2$  *gain*  $z(t)$  terhadap  $w(t)$ . Sesuai definisi tersebut, maka *norm- $\infty$*  dapat dihitung seperti pada persamaan berikut:

$$\|T_{zw}\| = \sup_{\omega} |T_{zw}(j\omega)| = \sup_{\|w(t)\|_2 \neq 0} \frac{\|z(t)\|_2}{\|w(t)\|_2} \quad (2.111)$$

Kestabilan  $L_2$  sangat memegang peran penting dalam analisis dari sebuah sistem. Dalam banyak permasalahan kontrol, sistem direpresentasikan sebagai pemetaan *input-output*, dari gangguan *input*  $w$  ke performa *output*  $z$ , yang harus bernilai kecil. Dengan sinyal *input*  $L_2$ , sistem kontrol didesain untuk membuat *gain*  $L_2$  *input-output* stabil dan untuk meminimalkan  $L_2$  *gain* [9]. Oleh karena itu, sistem  $L_2$  *gain* dikatakan dilemahkan oleh  $\gamma$  jika

$$\frac{\int_0^{\infty} \|z(t)\|_2^2 dt}{\int_0^{\infty} \|w(t)\|_2^2 dt} = \frac{\int_0^{\infty} (x^T Q x + u^T R u) dt}{\int_0^{\infty} (w^T w) dt} \leq \gamma^2 \quad (2.112)$$

untuk setiap *input* gangguan  $W$  yang *non-zero*.

Persamaan (2.109) akan membuat indeks performa mencapai nilai *minimum*  $J = \frac{1}{2} x^T(0) P x(0)$  dimana  $P$  adalah solusi definit positif untuk persamaan Riccati. Karena *closed loop* dari sistem stabil, maka diasumsikan bentuk fungsi kuadrat Lyapunov  $V(x) = x(t)^T P x(t)$ , sehingga dapat diasumsikan  $J \leq V(x(0))$  sistem merupakan sistem linier, sehingga  $J \leq V(x(0))$  jika dan hanya jika [13]:

$$\frac{dV(x(t))}{dt} \leq \gamma^2 w(t)^T w(t) - z(t)^T z(t) \quad (2.113)$$

dimana  $w(t)$  adalah  $w(t) = \frac{1}{\gamma^2} D^T P x(t)$ .

### 2.3.3 Command Generator Tracking dengan Model Following

*Command-Generator Tracker* (CGT) merupakan struktur desain kontrol yang memberikan kompensator sesuai dengan kebutuhan sistem untuk mendapatkan *error tracking* yang kecil untuk *input* referensi  $r(t)$  yang diinginkan [10]. Pendekatan CGT ini akan menggabungkan model dinamika dari  $r(t)$  kedalam sistem kontrol. Oleh karena itu, dinamika sistem kontrol *tracking* akan terdiri dari gabungan dinamika *plant* dan kompensator.

Persamaan *plant* seperti pada (2.106) dan performa *output*  $z(t)$  adalah

$$z(t) = Hx(t) \quad (2.114)$$

sehingga dari inisial persamaan *plant* dengan performa *output* dibawa ke *input* referensi  $r(t)$  menghasilkan pendekatan sebagai berikut: [10]:

$$\dot{x}' = \begin{bmatrix} A & 0 \\ 0 & \underline{A} \end{bmatrix} x' + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ B \end{bmatrix} r \equiv A' x' + B' u + W' r \quad (2.115)$$

Inisial kondisi dari sinyal referensi  $r(t)$  memenuhi persamaan differensial berikut [10]:

$$r^{(d)} + a_1 r^{(d-1)} + \dots + a_d r = 0 \quad (2.116)$$

dengan  $d$  adalah derajat dan  $a_i$  adalah koefisien. Jika (2.116) ditulis dalam bentuk variabel *state* (kanonik observabel), untuk kasus  $d = 3$ , maka

$$\begin{aligned} \dot{\rho} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \rho \equiv G\rho \\ r &= [1 \ 0 \ 0] \rho \end{aligned} \quad (2.117)$$

Karakteristik polynomial dari command generator didefinisikan sebagai

$$\Delta(p) = P^d + a_1 p^{d-1} + \dots + a_d \quad (2.118)$$

dimana  $p$  menotasikan  $\frac{d}{dt}$  dalam domain waktu. Kemudian (2.116) dapat ditulis dalam domain  $p$  seperti berikut:

$$\Delta(p)r = 0 \quad (2.119)$$

### A. Modifikasi Sistem

Dalam teknik kontrol ini, permasalahan *tracking* akan diubah kedalam permasalahan regulator dimana *error* harus diatur menjadi nol. Untuk menghasilkan sistem *tracking* secara keseluruhan maka dilakukan modifikasi. Didefinisikan dinamika tersebut *multiplying augmented* (2.120) yang dioperasikan didalam  $\Delta(p)$  sehingga *system* yang termodifikasi seperti berikut [10]:

$$\dot{\xi} = A' \xi + B' \mu \quad (2.120)$$

dimana vektor *state*, kontrol *input* setelah dimodifikasi menjadi

$$\xi = \Delta(p)x' = (x')^{(d)} + a_1 (x')^{(d-1)} + \dots + a_d (x') \quad (2.121)$$

$$\mu = \Delta(p)u = (u)^{(d)} + a_1 (u)^{(d-1)} + \dots + a_d (u) \quad (2.122)$$

dan vektor *error* serta turunannya dinyatakan oleh

$$\varepsilon(t) = [e \ \dot{e} \ \dots \ e^{(d-1)}]^T \quad (2.123)$$

maka *error tracking* dapat ditulis menjadi

$$\Delta(p)e = [-H \ H] \xi = H' \xi \quad (2.124)$$

Persamaan (2.124) dapat ditulis dalam bentuk *observable* kanonik seperti pada (2.125)

$$\begin{aligned} \dot{\varepsilon} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \varepsilon + \begin{bmatrix} 0 \\ 0 \\ H' \end{bmatrix} \xi \equiv Fe + \begin{bmatrix} 0 \\ 0 \\ H' \end{bmatrix} \xi \\ e &= [1 \ 0 \ 0] \varepsilon \end{aligned} \quad (2.125)$$

kemudian dinamika pada (2.120) dan (2.125) dibuat kedalam sebuah *state augmented* adalah sebagai berikut [10]:

$$\frac{d}{dt} \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix} = \begin{bmatrix} G & \vdots & 0 \\ & \vdots & H' \\ \dots & \vdots & \dots \\ 0 & \vdots & A \end{bmatrix} \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ B' \end{bmatrix} \mu \quad (2.126)$$

Desain regulator LQ diterapkan pada sistem ini, karena jika *state* pada sistem ini menuju nol, maka *error tracking*  $e(t)$  akan hilang. Jika *state output* yang akan menjadi *feedback* bagi *system* pada (2.128) adalah

$$v = \begin{bmatrix} I & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix} \quad (2.127)$$

maka *input* kontrol sistem modifikasi menjadi

$$\mu' = \begin{bmatrix} K_p & K_m \end{bmatrix} \begin{bmatrix} C \xi_p \\ C \xi_m \end{bmatrix} \quad (2.128)$$

## B. LQ Regulator dengan *output feedback*

Dalam permasalahan tesis ini, konsep *command-generator tracker model following* (CGT-MF) diterapkan dalam sistem kontrol *path following* untuk quadcopterr dan nilai gain  $K$  dicari menggunakan metode regulator LQ dengan *output feedback*. Jika sistem closed loop dari (2.106)-(2.107) dinyatakan oleh  $\dot{x} = (A - BKC)x = A_c x$

dengan indeks performa (PI) kuadratik untuk permasalahan ini adalah

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.129)$$

dengan  $Q \geq 0, R > 0$  adalah matriks pembobot semidefinit positif. Diasumsikan sistem closed-loop adalah stabil asymptotic sehingga *state*  $x(t)$  akan bernilai nol hingga waktu yang tak hingga, maka  $J$  menjadi

$$J = \frac{1}{2} (x^T(0) P x(0)) \quad (2.130)$$

Agar dapat menemukan matriks positif-semidefinit  $P$ , maka permasalahan optimisasi dapat dikonversi kedalam persamaan yang memenuhi

$$-J = \frac{d}{dt}(x^T P x) \quad (2.131)$$

$$x^T (A_c^T P + P A_c) x = -x^T (Q + C^T K^T R K C) x$$

Kondisi ini berlaku untuk semua inisial kondisi oleh karena itu, untuk semua state trajectory  $x(t)$ , dapat ditulis menjadi

$$g = A_c^T P + P A_c + C^T K^T R K C + Q = 0 \quad (2.132)$$

jika  $K$  dan  $Q$  diketahui dan  $P$  diperoleh, maka (2.132) disebut persamaan Lyapunov.

Inisial kondisi  $x(0)$  saja yang diperlukan untuk menghitung biaya closedloop dibawah pengaruh input kontrol  $u = -Ky$ . Untuk menghitung gain  $K$  yang meminimumkan indeks performa dapat menggunakan identitas trace dari matriks A dan B yaitu,  $tr(AB) = tr(BA)$ . sehingga PI pada (2.129) dapat ditulis menjadi

$$J = \frac{1}{2} tr(PX) \quad (2.133)$$

dimana  $x = \{x(0)x^T(0)\}$

namun, ketergantungan  $J$  pada inisial kondisi  $x(0)$  tidak diharapkan karena membuat gain optimal tergantung pada nilai inisial kondisi. Permasalahan ini merupakan tipikal dari permasalahan desain output-feedback. Oleh karena itu, untuk mengatasi permasalahan ini, dilakukan dengan tidak meminimumkan PI tetapi meminimumkan nilai ekspektasi dari PI. Persamaan (2.133) dapat diganti menjadi

$$J = \frac{1}{2} E\{x^T(0)Px(0)\} = \frac{1}{2} tr(PX) \quad (2.134)$$

sejangan  $X = E\{x(0)x^T(0)\}$  adalah nilai inisial autokorelasi dari state.

Fungsi Hamiltonian untuk permasalahan LQR output feedback ini adalah

$$H = tr(PX) + tr(gS) \quad (2.135)$$

dengan  $S$  adalah matriks simetris  $n \times n$  dari perkalian Lagrange yang harus ditentukan. Kondisi perlu untuk solusi LQR dengan output feedback adalah [9]:

$$\frac{\partial H}{\partial S} = A_c^T P + P A_c + C^T K^T R K C + Q = 0 \quad (2.136)$$

$$\frac{\partial H}{\partial P} = A_c S + S A_c^T + X Q = 0$$

$$\frac{1}{2} \frac{\partial H}{\partial K} = R K C S C^T - B^T P S C^T = 0$$

Jika  $R$  adalah positif definit dan  $C S C^T$  adalah non-singulir, maka gain  $K$  didapat

sebagai berikut [9]

$$K = R^{-1}B^T PSC^T (CSC^T)^{-1} \quad (2.137)$$

### 2.3.4 Blok Koreksi

Pada diagram blok sistem kontrol quadcopter yang terdapat dalam Gambar 3.1, terdapat sebuah blok koreksi. Blok koreksi berfungsi untuk memberikan hubungan antara input kontrol virtual  $(U_x, U_y)$  dengan sudut referensi  $(\phi_d, \theta_d)$ . Dimana  $U_x$  dan  $U_y$  adalah output dari kontroler posisi pada sumbu x dan y berturut-turut, sedangkan  $\phi_d$  dan  $\theta_d$  adalah input referensi yang dibutuhkan oleh kontroler rotasi. Berdasarkan (2.51), (2.55) dan (2.56), pergerakan quadcopter dapat dinyatakan menjadi:

$$\dot{V} = -gz_E + \frac{F_T}{m} RZ_E \quad (2.138)$$

dimana  $F^T$  adalah total gaya angkat (thrust) yang dihasilkan oleh empat motor,  $Z_E = (0,0,1)^T$  adalah unit vektor pada frame bumi,  $m$  adalah massa dan  $R$  adalah matriks rotasi.

Definisikan vektor dari input kontrol virtual sebagai  $U_v = \ddot{v} = (U_x, U_y, U_z)^T$ . Pada (2.126) ganti  $\dot{V}$  dengan  $U_v$  sehingga menghasilkan:

$$U_v = -gz_E + \frac{F_T}{m} RZ_E \quad (2.139)$$

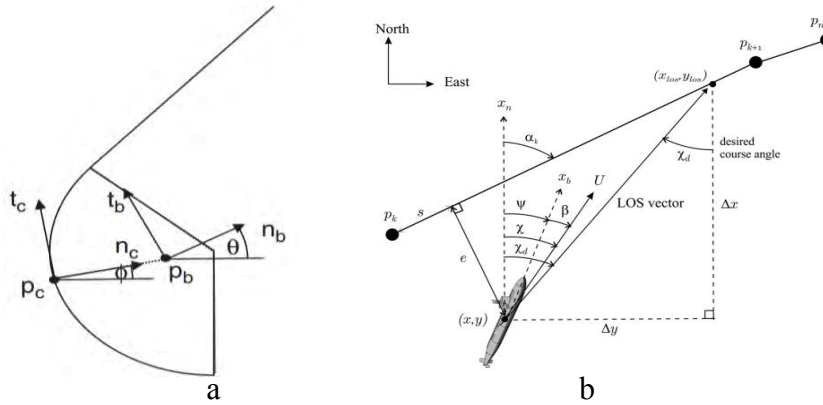
### 2.3.5 Path following

*Path following* adalah kondisi yang memungkinkan UAV dapat mengikuti kecepatan dan keakuratan pada *heading*. Pada kasus ini uav akan diterbangkan dekat dengan jalur yang sudah dibangun dan uav ditugaskan untuk melacak keberadaan jalur serta mampu mengikuti jalur [12].

- **Dubins Path pada Path following**

UAV mengikuti dubins *Path*, seperti yang ditunjukkan pada Gambar 2.14a. Gambar tersebut menunjukkan UAV pada axis  $t_b$  dan  $n_b$ , yang mewakili sumbu pada UAV, dengan  $\psi$  sebagai sudut *yaw* UAV. Dalam kurva Dubins ( $t_c$  dan  $n_c$ ) diperbaruhi sehingga, UAV terletak di sepanjang  $n_c$  pada sumbu normal. UAV akan bergerak pada ruang inersia dimana sumbu kurva Dubins akan bergerak sepanjang kurva untuk mempertahankan posisi UAV pada vektor  $n_c$  secara normal.





Gambar 2.14a. *Guidance Geometri* [13] dan b. Referensi sudut *course*  $\chi_d$  pada LOS *guidance* [17]

### A. Hukum LOS (*Line Of Sight*) pada Persamaan Steering

Menurut [17], gerak UAV pada bidang horizontal 2D memiliki kelajuan yang dapat didefinisikan dengan:

$$U(t) := \|v(t)\| = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} \geq 0 \quad (2.140)$$

dan sudut yang berhubungan dengan persamaan *steering* dapat dihitung dengan:

$$\chi(t) := a \tan 2(\dot{y}(t), \dot{x}(t)) \in S := [-\pi, \pi] \quad (2.141)$$

Persamaan (2.141) diatas merupakan sudut *course*, sudut ini yang digunakan dalam melakukan proses kontrol dari kasus *steering* yaitu dengan memberikan suatu referensi sudut *course* dari suatu kendaraan agar bergerak mendekati sama dengan *path* yang sudah didefinisikan sebelumnya, dengan syarat  $U(t) > 0$  (kelajuannya positif).

Suatu garis lurus dapat didefinisikan dengan 2 *waypoint*  $P_k^n = [x_k, y_k]^T \in R^2$  dan  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T \in R^2$ . Garis lurus yang terbentuk dari 2 *waypoint* tersebut sebenarnya adalah suatu kerangka referensi dari suatu *path* yang tetap, dimana jika sumbu  $x$  dari kerangka tersebut diputar dengan sudut yang bernilai positif, maka sudut tersebut adalah

$$a_k := a \tan 2[y_{k+1} - y_k, x_{k+1} - x_k] \in S \quad (2.142)$$

yang relatif terhadap sumbu  $x$  bumi. Sehingga koordinat dari suatu kendaraan yang bergerak dalam kerangka referensi *path* yang tetap tersebut dapat dihitung dengan

$$\varepsilon(t) = R_p(a_k)^T (P^n(t) - P_k^n) \quad (2.143)$$

dimana:

$$R_p(a_k) := \begin{bmatrix} c(a_k) & -s(a_k) \\ s(a_k) & c(a_k) \end{bmatrix} \in SO(2) \quad (2.144)$$

dan  $\varepsilon(t) = [s(t), e(t)]^T \in R^2$  dengan:

$s(t)$  = *along-track distance* (bersinggungan dengan *path*),

$e(t)$  = *cross-track error* (normal to *path*).

Tujuan dari *path following* adalah membuat suatu kendaraan agar konvergen menuju *path* yang sudah didefinisikan, yaitu dengan membuat *cross-track error*  $e(t) = 0$ . Perhitungan persamaan (2.141) merupakan *along-track distance* dan *cross-track error*, persamaan tersebut dapat dijabarkan secara eksplisit dalam bentuk *state*, yaitu

$$s(t) = [x(t) - x_k] \cos(a_k) + [y(t) - y_k] \sin(a_k) \quad (2.145)$$

$$e(t) = -[x(t) - x_k] \sin(a_k) + [y(t) - y_k] \cos(a_k) \quad (2.146)$$

Dalam melakukan proses *guidance*, suatu kendaraan mendapatkan referensi sudut *course* ( $\chi(t)$ ) dan *heading* ( $\psi(t)$ ). Sudut tersebut digunakan untuk mengarahkan kendaraan mendekati *path* yang diinginkan. Proses perhitungan referensi sudut tersebut dihitung pada sistem *guidance*, dimana metode perhitungannya menggunakan dua prinsip *guidance* menurut [17]:

- **Enclosure-based steering**
- **Lookahead-based steering**

dan pada tesis ini prinsip yang digunakan adalah *Enclosure-based steering*, sehingga tidak dibahas prinsip *lookahead-based steering*.

## B. Enclosure-Based Steering

Strategi berdasarkan *enclosure* digunakan untuk mengarahkan  $e(t)$  menuju nol, kemudian mengarahkan vektor kecepatan menuju titik perpotongan  $P_{los}^n = [x_{los}, y_{los}]^T$  agar sesuai dengan arah dari *path*, dimana *path* tersebut secara implisit didefinisikan oleh urutan di mana *waypoints* yang sudah didefinisikan. *Path* tersebut melibatkan secara langsung penetapan ( $\chi_d(t)$ ) seperti pada Gambar 2.14b.

$$\tan(\chi_d(t)) = \frac{\Delta y(t)}{\Delta x(t)} = \frac{y_{los} - y(t)}{x_{los} - x(t)} \quad (2.147)$$

Koordinat pusat suatu kendaraan  $\{b\}$  pada Gambar 2.14b didefinisikan dengan  $P^n = [x, y]^T$  dan digunakan sebuah lingkaran dengan radius  $R > 0$  yang ditarik dari titik pusat kendaraan  $\{b\}$ . Lingkaran tersebut akan memotong suatu *path* pada dua titik, dimana salah satu titiknya adalah  $(x_{los}, y_{los})$ . Perhitungan dua titik LOS  $P_{los}^n = [x_{los}, y_{los}]^T$  yang tidak diketahui, dapat dicari dengan menyelesaikan dua persamaan dibawah ini:

$$[x_{los} - x(t)]^2 + [y_{los} - y(t)]^2 = R^2 \quad (2.148)$$

$$\tan(a_k) = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \quad (2.149)$$

$$= \frac{y_{los} - y_k}{x_{los} - x_k} = \text{konstan}$$

dimana persamaan (2.146) merupakan persamaan *pythagoras*, sedangkan persamaan (2.147) menunjukkan sudut kemiringan dari *path* terhadap sumbu  $x$  bumi sebesar  $a_k$ . Kemiringan diantara dua *waypoints* tersebut adalah konstan dan berlaku juga besar kemiringannya untuk titik-titik diantaranya. Proses perhitungan *enclosure* secara analitik berdasarkan [17] dapat dijabarkan dengan:

Argumen 1: untuk  $|\Delta x| > 0$ , maka persamaan (2.147),

$$y_{los} = \frac{\Delta y}{\Delta x} (x_{los} - x_k) + y_k \quad (2.150)$$

dengan  $\Delta x := x_{k+1} - x_k$  dan  $\Delta y := x_{y+1} - x_y$  merupakan perbedaan posisi  $x$  dan  $y$  diantara dua *waypoints*.

*--Halaman ini sengaja dikosongkan--*

## BAB 3

### METODOLOGI PENELITIAN

Pada bab ini, perancangan sistem dibagi menjadi tiga tahap. Tahap pertama linearisasi dinamika quadcopter, sehingga bisa dilakukan analisa dan perancangan kontroler. Tahap kedua yaitu menyusun kontroler pada quadcopter dimana kontroler ini akan mengatur kestabilan gerak rotasi dan gerak translasi quadcopter. Tahap terakhir yaitu penyusunan simulasi sistem yang dibangun dengan menggunakan bantuan Matlab.

#### 3.1 Linearisasi Model Dinamika Quadcopter

Pada bagian ini akan dijelaskan mengenai linearisasi dinamika quadcopter, dimana konsep dinamika sudah diterangkan pada sub-bab dinamika quadcopter. Persamaan (2.75)-(2.77) tersebut kemudian dilinearisasi dimana quadcopter dalam kondisi melayang (*hover*). Sudut *yaw* bernilai 0 rad dan kecepatan sudut *roll*, *pitch*, dan *yaw* bernilai mendekati 0 rad/s. Gaya angkat (*thrust*) yang dibangkitkan oleh tiap propeller dimodelkan dalam bentuk orde satu. Nilai parameter diperoleh dari [14], yang ditulis dalam tabel 3.1. variable *state*  $v$ , yang akan digunakan sebagai representasi dinamika pada aktuator quadcopter seperti berikut:

$$v = \frac{\omega}{s + \omega} u \quad (3.1)$$

##### 3.1.1 Parameter Quadcopter

Parameter yang digunakan adalah parameter Qball-X4. Parameter ini digunakan sebagai acuan penyusunan kontroler dan simulasi. Nilai parameter Qball-X4 dapat dinyatakan pada Tabel 3.1. Nilai-nilai inersia  $J_{xx}$  merupakan inersia quadcopter terhadap sumbu  $x$ ,  $J_{yy}$  merupakan inersia quadcopter terhadap sumbu  $y$ , dan  $J_{zz}$  merupakan inersia quadcopter terhadap sumbu  $z$  serta  $J_r$  merupakan inersia motor.

Keseluruhan sistem kontrol terdiri dari 2 subsistem kontrol, yaitu sistem kontrol untuk gerak rotasi dan sistem kontrol untuk gerak translasi. Hal ini didasarkan pada model dinamik quadcopter yang telah diperoleh, dimana sudut-sudut dinamik (sudut *pitch*, *roll*, dan *yaw*) beserta turunannya tidak bergantung pada komponen translasi. Akan tetapi dinamika gerak translasi bergantung pada sudut-sudut dinamik.

Tabel 3.1 Parameter Sistem Quanser Qball-X4 [14]

Parameter	Simbol	Nilai
Massa	$m$	3,499 kg
Gravitasi	$g$	9,81 kg/m <sup>2</sup>
Moment Inersia pada sumbu X	$J_{xx}$	0.03 kg.m <sup>2</sup>
Moment Inersia pada sumbu Y	$J_{yy}$	0.03 kg.m <sup>2</sup>
Moment Inersia pada sumbu Z	$J_{zz}$	0.04 kg.m <sup>2</sup>
Jarak rotor dari pusat massa	$l$	0.2 m
Gaya drag	$d$	3,13x10 <sup>-5</sup>
Gaya trust	$b$	7,5x10 <sup>-7</sup>
<i>Bandwith</i> aktuator	$\omega$	15 rad/s
Konstanta gaya dorong	$K$	120 N

Secara umum, sistem navigasi memberikan titik-titik referensi atau posisi yang diharapkan ( $x_d, y_d, z_d$ ), kemudian sistem kontrol posisi atau sistem kontrol translasi akan memberikan referensi sudut-sudut dinamik yang diperlukan untuk bisa mencapai posisi yang diharapkan pada sistem kontrol rotasi.

Sistem kontrol rotasi akan memberikan sinyal kontrol pada aktuator untuk memberikan aksi kontrol yang sesuai pada sistem (quadcopter). Variabel-variabel *output* akan diukur oleh sensor untuk kemudian diumpan-balikkan pada kedua sistem kontrol.

### 3.1.2 Pemodelan Linearisasi sudut *Roll* dan *Pitch*

Asumsikan rotasi disekitar sumbu  $x$  dan  $y$  adalah *decoupled*, pergerakan dalam sumbu *roll/pitch*. Gaya angkat yang dihasilkan oleh setiap motor dapat dikalkulasi dan menggunakan *input* masing-masing yang berhubungan. Rotasi disekitar pusat gravitasi akan menghasilkan perbedaan dalam membangkitkan gaya angkat (*thrust*). Jika konstanta drag diabaikan dan quadcopter dianggap dalam kondisi melayang dimana percepatan quadcopter mendekati.

$$\phi \ll 0.1 \Rightarrow \sin(\phi) \cong 0, \cos(\phi) \cong 1 \quad (3.2)$$

$$\theta \ll 0.1 \Rightarrow \sin(\theta) \cong 0, \cos(\theta) \cong 1$$

Bentuk kinematika sederhana *roll, pitch* dan *yaw*

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -p \\ q \\ r \end{bmatrix} \quad (3.3)$$

Sudut *roll/pitch* yang dimisalkan  $\theta$ , dapat diformulasikan menggunakan dinamika sebagai berikut:

$$J\ddot{\theta} = \Delta F l \quad (3.4)$$

dengan  $J = J_{roll} = J_{pitch}$ ,  $l$  adalah jarak propeller terhadap pusat gravitasi, dan  $\Delta F = F_1 - F_2$  terdapat perbedaan antara gaya yang telah dibangkitkan oleh kedua motor. Perbedaan gaya yang dihasilkan oleh perbedaan *input* untuk kedua motor yaitu,  $\Delta u = u_1 - u_2$

Jika diambil pada kondisi derivative pada persamaan (2.24), sehingga:

$$\ddot{\phi} = \dot{p} = \frac{U_2 l}{J_{xx}} - \frac{qr}{J_{xx}} (J_{zz} - J_{yy}) \quad (3.5)$$

$$\ddot{\phi} = \dot{p} = \frac{l}{J} K_T \frac{\omega}{s + \omega} \Delta u_2 \quad (3.6)$$

$$\ddot{\phi} = \dot{p} = \frac{U_3 l}{J_{yy}} - \frac{pr}{J_{yy}} (J_{xx} - J_{zz})$$

$$\ddot{\phi} = \dot{p} = \frac{l}{J} K_T \frac{\omega}{s + \omega} \Delta u_1$$

dengan  $\Delta u_1 = u_1 - u_3$  dan  $\Delta u_2 = u_2 - u_4$

Sehingga dapat ditulis kembali menjadi

$$\ddot{\phi} = \dot{p} = \frac{U_2 l}{J_{xx}} \quad (3.7)$$

$$\ddot{\phi} = \dot{p} = \frac{U_3 l}{J_{yy}} \quad (3.8)$$

Jika (3.8)-(3.9) dan (3.1) dikombinasikan, maka bentuk persamaan *state space* untuk model linier dinamika *roll* dan *pitch* dapat dinyatakan:

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{lK_T}{J} \\ 1 & 0 & -\omega \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \Delta u_2 \quad (3.9)$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{lK_T}{J} \\ 1 & 0 & -\omega \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \Delta u_1 \quad (3.10)$$

#### A. Pemodelan Linearisasi sudut *Yaw*

Perbedaan torsi yang dihasilkan akibat perputaran dua propeller yang searah jarum jam dan dua propeller berlawanan arah jarum jam yang mengakibatkan gerak pada sumbu *yaw*. Pergerakan dalam sumbu *yaw* dapat dimodelkan mengikuti persamaan berikut:

$$J_{zz} \dot{\psi} = \Delta \tau \quad (3.11)$$

dengan  $\tau$  adalah torsi yang akan dibangkitkan oleh tiap-tiap motor pada propeller sehingga, hubungan torsi dengan input motor didefinisikan oleh:

$$\tau = K_T u \quad (3.12)$$

*Gain* yang bernilai positif ( $K_T$ ) dimana nilai parameter terdapat pada tabel 3.1. sehingga, hasil dari kombinasi (2.78) dan (3.12) didapatkan model dengan  $\Delta u = u_1 + u_2 + u_3 + u_4$

$$\ddot{\psi} = \dot{r} = \frac{U_4}{J_{zz}} - \frac{pq}{J_{zz}} (J_{yy} - J_{xx}) \quad (3.13)$$

$$\ddot{\psi} = \dot{r} = \frac{U_4}{J_{zz}} = \frac{K_T}{J_{zz}} \Delta u$$

sehingga dapat ditulis kembali menjadi

$$\ddot{\psi} = \dot{r} = \frac{U_4}{J_{zz}} \quad (3.14)$$

model linier yang didapat dari dinamika gerakan *yaw* dapat ditulis dalam bentuk *state sspace* yaitu

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_T}{J_{zz}} \end{bmatrix} \Delta u \quad (3.15)$$

## B. Pemodelan Linearisasi X dan Y

Gaya angkat total dan perubahan sudut *roll/pitch* yang terjadi sepanjang sumbu  $X$  dan  $Y$ . Jika diasumsikan sudut *yaw* adalah nol maka, dinamika gerak dalam sumbu  $X$  dan  $Y$  dari (2.75)-(2.76) dapat ditulis sebagai berikut:

$$\ddot{x} = \frac{4F}{m} (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \quad (3.16)$$

$$\ddot{y} = \frac{4F}{m} (-\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \quad (3.17)$$

Percepatan sudut *roll* dan *pitch* mendekati nol yang diasumsikan, maka dengan mensubstitusikan (3.1), model posisi sumbu  $X$  dan  $Y$  dapat ditulis menjadi:

$$\ddot{x} = \frac{4F}{m} v \sin \theta \approx \frac{4K_T}{m} v \theta \quad (3.18)$$

$$\ddot{y} = \frac{4F}{m} v (-\sin \phi) \approx -\frac{4K_T}{m} v \phi \quad (3.19)$$

Model linier dinamika posisi yang didapat pada sumbu  $X$  dan  $Y$  dalam bentuk *state space* adalah sebagai berikut:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{4K_T}{J} \theta \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} u \quad (3.20)$$

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{4K_T}{J} \phi \\ 0 & 0 & -\omega \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} u \quad (3.21)$$

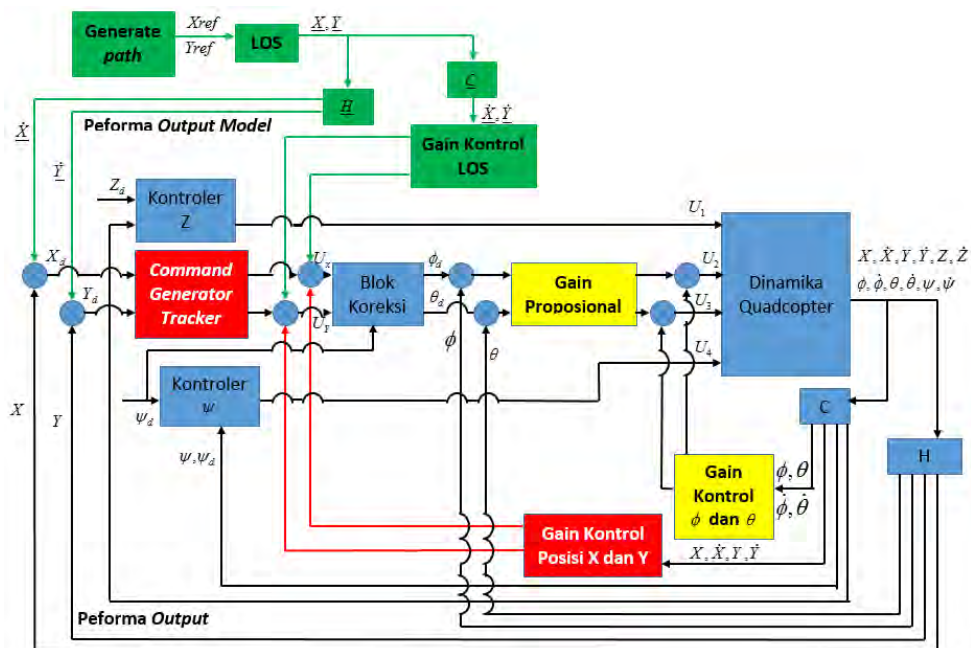


### 3.2 Diagram blok quadcopter

Gambar 3.1 merupakan diagram blok dari strategi kontrol Quadcopter yang akan dirancang. Struktur kontrol terdiri model *following*, *inner* dan *outer loop*. Model *following* (yang ditandai dengan blok berwarna hijau) merupakan referensi  $x$  dan  $y$  model dari *generate path* serta penggunaan algoritma LOS dengan output berupa sudut  $\psi_d$  yang akan masuk dalam blok kontrol  $\psi$ , *Inner loop* merupakan kontroler rotasi sudut  $(\phi, \theta, \psi)$  yang menambahkan performa  $H_\infty$  (ditandai dengan blok berwarna kuning), sedangkan *outer loop* merupakan kontroler *tracking* posisi  $(x, y, z)$  yang menggunakan struktur CGT (ditandai dengan blok berwarna merah). Sinyal referensi untuk sistem kontrol quadcopter ini terdiri atas referensi untuk posisi  $x_d, y_d, z_d$  dan sudut  $\psi_d$ .

Sinyal referensi model berupa posisi  $x$  dan  $y$  dimana output dari hasil model adalah  $\dot{x}$  dan  $\dot{y}$  yang kemudian dibandingkan dengan sinyal aktualnya sehingga terdapat *error tracking*. Sedangkan keluran dari algoritma LOS yang berupa sudut  $\psi_d$  untuk mengatur arah hadap dari quadcopter agar konvergen menuju *path* yang sudah didefinisikan, yaitu dengan membuat *cross-track error*.

*Error tracking* akan diproses pada blok *Command generator tracker* yang menerapkan konsep dari *feed forward* dimana konsep tersebut digunakan untuk meminimalkan adanya gangguan yaitu sebagai filter serta mampu mengatasi pergeseran *tracking* (*asymptotic error*).



Gambar 3.1 Diagram blok sistem kontrol untuk quadcopter

*Output* dari CGT akan dibandingkan dengan *output* dari blok *gain output feedback* untuk posisi  $x$  dan  $y$ . *Output* dari posisi  $x$  dan  $y$  adalah *input* virtual yang dinotasikan oleh  $U_x$  dan  $U_y$ . Blok koreksi memberikan hubungan antara *input* virtual dengan referensi sudut yang diinginkan  $\phi_d, \theta_d$ .

Pada kontroler rotasi, sinyal sudut referensi  $\phi_d, \theta_d, \psi_d$ , dibandingkan dengan sinyal aktualnya sehingga menghasilkan *error tracking* rotasi. *Error* ini akan diolah, kemudian *output* dari kontroler rotasi adalah  $U_2, U_3$  dan  $U_4$  yang merupakan *input* kontrol untuk quadcopter. Sedangkan *input* kontrol  $U_1$  dihasilkan dari *output* kontroler posisi  $z$ .

### 3.3 Perancangan Kontroler

Pada quadcopter terdapat dua macam permasalahan yang harus diselesaikan agar quadcopter mampu terbang dan benjalan sesuai lintasan *path* yang diinginkan. Masalah pertama yaitu kestabilan sudut  $\phi, \theta$  dan  $\psi$  akibat gerak *roll*, *pitch*, dan *yaw*. Masalah kedua yaitu *path following*  $x, y$  dan  $z$ , atau memaksa quadcopter agar mampu mengikuti dan bergerak pada lintasan yang diinginkan. Pada Tesis ini permasalahan kestabilan sudut  $\phi, \theta$  dan  $\psi$  akan diselesaikan dengan teknik LMI *pole placement*, sedangkan permasalahan rotasi (*inner loop*) diselesaikan dengan *output feedback* dengan performa  $H^\infty$ , untuk permasalahan posisi (*outer loop*) diselesaikan dengan LQ regulator dengan menambahkan *Command Generator Tracker* dengan model *following* serta penambahan algoritma geometri *Line Of Sight* (LOS) untuk mengatur arah hadap dari quadcopter yang dihitung dari nilai referensi yang telah diberikan oleh *generate path*.

#### 3.3.1 Prosedur Perancangan Sistem

Karakteristik yang diinginkan dari *outer loop* harus diketahui terlebih dahulu. Kemudian, setelah menentukan karakteristik *outer loop*, maka dirancanglah sebuah kontroler *inner loop* sehingga dinamika *inner loop* diharapkan mampu bekerja lebih cepat dari dinamika *outer loop*. Sehingga, didapatkan performa *path following* yang ingin dicapai dari sebuah rancangan metode kontroler pada quadcopter. Adapun langkah-langkah desain kontroler untuk quadcopter pada tesis ini adalah:

##### 1. Tentukan spesifikasi kontroler posisi (*outer loop*)

- Menentukan *path* yang akan terbentuk dari sinyal referensi model berupa posisi  $\underline{x}$  dan  $\underline{y}$ , menghitung *Line Of Sight* (LOS) dengan *output* berupa sudut  $\psi$  sebagai pencarian arah hadap dengan algoritma program untuk menghitung rumus *pythagoras* yaitu

$$[x_{los} - x(t)]^2 + [y_{los} - y(t)]^2 = R^2 .$$

- Update  $\psi_d$  dan  $u$ , sehingga quadcopter mampu menghitung posisi dan kecepatan yang terukur pada kondisi selanjutnya.
- Dalam kondisi melayang, quadcopter sudah melakukan *path following* posisi pada sumbu  $x$  dan  $y$ .
- Solusi penyelesaian menggunakan metode *output feedback* sehingga nilai Matriks  $(\sqrt{Q}, A)$  harus *detectable*.
- Penempatan *pole closed loop* lebih mendekati titik 0 dan kondisi sistem *closed loop* dari dinamika posisi  $x$  dan  $y$  harus memiliki *eigenvalue* bernilai negatif.
- Kontroler *outer loop* dikhususkan untuk permasalahan *path following* dengan sinyal *path* sebagai referensi berbentuk lingkaran.
- Model *following* yang ditambahkan pada struktur CGT menggunakan indeks performa.

$$J = \frac{1}{2} \int_0^{\infty} [(HAx + HBu - \underline{AH}x)^T Q (HAx + HBu - \underline{AH}x) u^T Ru] dt$$

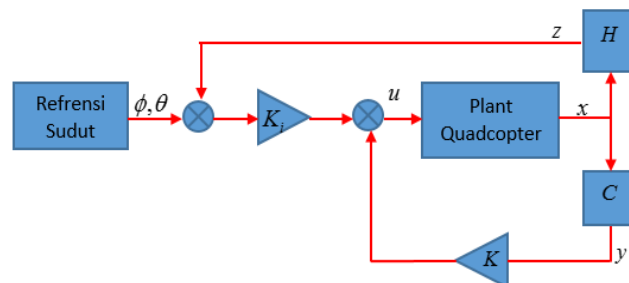
- $W$  adalah *cross-weighting between*  $u(t)$  dan  $x(t)$ . Sehingga kondisi model *following* dimana  $W \neq 0$ .

$$W = (HA - \underline{AH})^T QHB$$

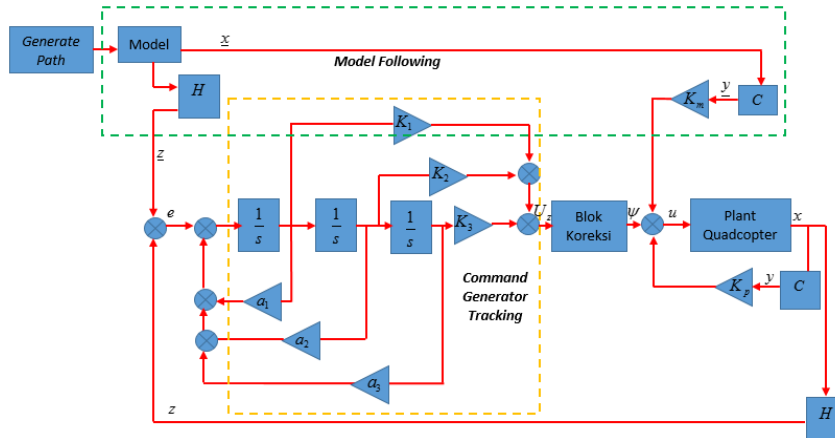
## 2. Menentukan spesifikasi kontroler rotasi (*inner loop*)

- Penempatan *pole* dinamika translasi, atau *eigenvalue* dari *inner loop* > (minimal 2 s/d 4)\**outer loop*. Sehingga penempatan *pole-pole* pada kondisi *closed loop* dari sistem rotasi harus berada minimal 2 kali lebih jauh dari.

Berikut adalah rancangan diagram blok untuk *outer loop* dan *inner loop*



Gambar 3.2 Kontrol *Inner loop*



Gambar 3.3 Kontrol *outer loop*

### 3.4 Perancangan Kontroler *Outer loop* (Posisi)

Sistem kontrol posisi dirancang dengan tujuan agar respon sistem dapat mengikuti sinyal referensi dari *path* yang diinginkan. Struktur kontrol yang digunakan pada kontroler posisi ini berbeda dengan struktur kontrol *tracking* yang telah dilakukan oleh [1]-[5], oleh karena itu struktur kontrol ini yang menjadi kontribusi dalam Tesis ini.

#### 3.4.1 Generate *Path* dan Guidance

Pada struktur kontrol posisi ini menggunakan *command-generator tracker model following* dimana pada struktur tersebut terdapat blok model yang merepresentasikan model dari referensi (*generate path*). Pada blok model tersebut menggunakan algoritma LOS (*Line Of Sight*) sehingga menghasilkan gerak sesuai dengan apa yang diinginkan dan mengatasi permasalahan pada saat quadcopter melakukan *path following* dengan jalur yang tegak lurus.

Blok *generate path* pada penelitian ini digunakan untuk memberikan referensi *path*. Referensi tersebut di *generate* menggunakan *waypoint*  $P_k^n = [x_k, y_k]^T$  dan  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T$  yang relatif terhadap bumi  $\{n\}$ . Dengan prinsip dari LOS vector yaitu *Enclosure based steering* maka, Mempertimbangkan lingkaran dengan radius  $R > 0$  yang memasukkan  $P^n = [x, y]^T$ . Jika radius lingkaran yang dipilih cukup besar, lingkaran tersebut akan memotong garis lurus pada dua titik. Prinsip *Enclosure* untuk mengatur *error*  $e(t)$  menuju nol untuk menyearahkan kecepatan terhadap titik perpotongan  $p_{los}^n = [x_{los}, y_{los}]^T$  yang sesuai dengan keinginan. Sehingga, solusi ini melibatkan penentuan  $\chi_d$  maka:

$$\tan(\chi_d(t)) = \frac{\Delta y(t)}{\Delta x(t)} = \frac{y_{los} - y(t)}{x_{los} - x(t)} \quad (3.22)$$

sudut yang diinginkan dapat dihitung dengan:

$$\chi_d(t) = a \tan 2(y_{los} - y(t), x_{los} - x(t)) \quad (3.23)$$

Pengaturan tersebut didasarkan pada besarnya nilai *error* antara arah hadap yang diinginkan dengan respon aktualnya ( $e = \psi_d - \psi$ ).  $\psi_d$  merupakan arah yang diinginkan berdasarkan titik tujuan  $(x_{los}, y_{los})$ . Untuk menghitung dua variable yang belum diketahui pada  $p_{los}^n = [x_{los}, y_{los}]^T$  maka, persamaan tersebut:

$$[y_{los} - y(t)]^2 + [x_{los} - x(t)]^2 = R^2 \quad (3.24)$$

$$\tan(\alpha_k) = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \quad (3.25)$$

$$\tan(\alpha_k) = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \text{konstan}$$

$R$  pada persamaan (3.24) merupakan jarak antara  $P_k^n = [x_k, y_k]^T$  dengan  $P^n(t) = [x(t), y(t)]^T$ . Jarak tersebut digunakan untuk menghitung titik belok quadcopter dan setelah melewati titik belok arah hadap baru ditentukan lagi berdasarkan  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T$  dan begitu seterusnya. Namun pada penelitian ini, titik belok quadcopter harus ditentukan berapa meter sebelum  $P_k^n = [x_k, y_k]^T$  dan  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T$ . Penentuan berapa meter ini yang disebut dengan  $p_{los}^n = [x_{los}, y_{los}]^T$  dan pada penelitian ini diberi jarak 6 meter sebelum  $P_k^n = [x_k, y_k]^T$  dan  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T$ . Jika diasumsikan bahwa kecepatan diabaikan maka sudut arah hadap pada (3.26):

$$\psi_d(t) = a \tan 2(y_{los} - y(t), x_{los} - x(t)) \quad (3.26)$$

Tabel 3.2 Algoritma *path following straigh line* [13]

---

**Algoritma Path following Straigh Line (Garis Lurus) Menggunakan LOS**

---

1. Inisialisasi *waypoint*  $P_k^n = [x_k, y_k]^T$  sampai  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T$  yang digunakan *generate path*.
  2. Untuk setiap iterasi hitung nilai *error* antara posisi  $P_k^n = [x_k, y_k]^T$  dengan posisi aktual Quadcopter  $p^n = [x(t), y(t)]^T$ .
  3. Tentukan titik *los*  $p_{los}^n = [x_{los}, y_{los}]^T$  melalui berapa meter yang diinginkan dari *waypoint*  $P_k^n = [x_k, y_k]^T$  dan  $P_{k+1}^n = [x_{k+1}, y_{k+1}]^T$ .
  4. Hitung nilai absolut *error* antara posisi *waypoint* dengan aktual Quadcopter  
 $|e_p| = \sqrt{(x_{los} - x(t))^2 + (y_{los} - y(t))^2}$ .
  5. Hitung nilai sudut *psi*  $\psi$  dengan  $\psi_d = a \tan 2(x_{los} - x(t), y_{los} - y(t))$
-

### 3.4.2 Perancangan Kontroler posisi pada sumbu $X$ dan $Y$

Struktur kontrol *Path following* posisi diperluas (*augmented*) dengan menambahkan *Command-Generator Tracker*(CGT) model *Following* bertujuan untuk mengatasi adanya pergeseran *path following* (*zero path following error*) antara sinyal referensi dan performa *output* dan meminimalisir adanya *Disturbance*.

#### A. Perancangan kontroler posisi pada sumbu $X$

Objektif kontrol *path following* posisi adalah membuat quadcopter dapat mengikuti lintasan / *path* yang sudah terdefinisi yang diinginkan dalam kondisi quadcopter sudah melayang (*hover*). Oleh karena itu, sinyal referensi yang diberikan untuk melakukan *path following* terhadap posisi  $X$  adalah sebuah sinyal sinusoidal dengan fasa sebesar  $-0.26$  rad dan frekuensi  $0.1047$  rad/s atau sama dengan  $0.0167$  Hz dengan. Persamaan sinyal referensinya adalah:

$$x_d = -\sin(0.05\pi t) \quad (3.27)$$

maka didapat fungsi transfernya dengan cara mengubah fungsi transfernya dengan transformasi Laplace

$$\frac{-0.05\pi}{s^2 + (0.05\pi)^2} = \frac{-0.157}{s^2 + 0.02} \quad (3.28)$$

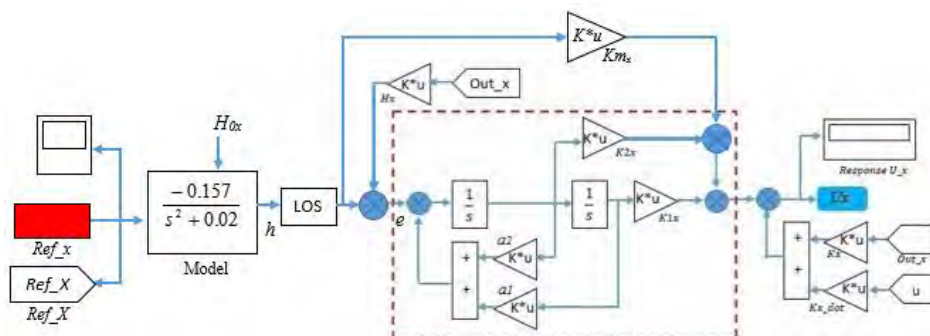
sehingga berdasarkan (2.116) dimana referensi  $X_d$  yang merupakan karakteristik *polynomial* adalah

$$\Delta(p)x_d = p^2 + 0.02 = 0 \quad (3.29)$$

dimana bentuk *state space* dari (3.29) seperti berikut:

$$\dot{x}_d = \begin{bmatrix} 0 & 1 \\ -0.02 & 0 \end{bmatrix} x_d = Gx_d \quad (3.30)$$

Persamaan state sistem quadcopter untuk posisi sumbu  $X$  sama dengan sistem untuk posisi  $Y$ , yaitu:



Gambar 3.4 Diagram simulink kontroler untuk posisi  $X$

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 27.2537 \\ 0 & 0 & -15 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 15 \end{bmatrix} U_1 \\ y &= [1 \ 0 \ 0]x \end{aligned} \quad (3.31)$$

dengan performa *output* adalah

$$z = Hx = [1 \ 0 \ 0]x \quad (3.32)$$

Dinamika sistem tersebut kemudian dimodifikasi dengan memperluas (*augmented*) dinamika pada posisi  $X$  dengan dinamika sinyal referensi yang sesuai dengan (2.126). Berikut adalah persamaan state sistem yang telah dimodifikasi,

$$\tilde{\dot{x}}_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -0.02 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 27.2537 \\ 0 & 0 & 0 & 0 & -15 \end{bmatrix} \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ x \\ \dot{x} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 15 \end{bmatrix} U_1 \quad (3.33)$$

dan persamaan *output* menjadi

$$\tilde{y}_x = \tilde{C}\tilde{x} = \begin{bmatrix} I & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \varepsilon \\ x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ x \\ \dot{x} \\ v \end{bmatrix} \quad (3.34)$$

Diagram Simulink untuk kontrol posisi pada sumbu  $X$  dapat dilihat pada Gambar 3.4, dimana yang menjadi bagian dari CGT ditandai oleh garis putus-putus. Model following sudah dalam bentuk fungsi transfer (3.28). Algoritma LOS digunakan untuk menghitung referensi  $X$  yang akan dibandingkan dengan aktual  $X$ . Untuk sinyal referensi yang bukan unit step penambahan integrator pada struktur kontrol *tracking* dan *path following* tidak dapat menjamin *zero steady-state tracking error*. Oleh karena itu, desain kontrol *path following* menggunakan *Command-Generator Tracker model following* terdapat dalam [16]

## B. Perancangan kontroler posisi pada sumbu $Y$

Pada tesis ini quadcopter difokuskan untuk dapat melakukan *path following* pada lintasan yang sudah terdefinisi berupa lingkaran (*Loiter*) yang diberikan sebagai sinyal referensi bagi quadcopter (dalam bidang 3 dimensi). Oleh karena itu, sinyal referensi yang diberikan untuk posisi  $Y$  adalah sebagai berikut:

$$y_d = -1 - \cos(0.05\pi) \quad (3.35)$$

jika diubah dalam bentuk Laplace menjadi

$$y_d = -\frac{1}{s} - \frac{s^2}{s^2 + (0.05\pi)^2} = \frac{-(s^2 + 0.02) - s^3}{s(s^2 + 0.02)} = \frac{-s^3 - s^2 + 0.02}{s^3 + 0.02} \quad (3.36)$$

persamaan karakteristik polynomial adalah

$$\Delta(p)y_d = p^3 + 0.02p = 0 \quad (3.37)$$

Jika berdasarkan persamaan (2.16)-(2.117) maka, persamaan (3.38) adalah sinyal referensi dengan orde  $d=3$  yang merupakan persamaan karakteristik polynomial, sehingga (3.46) dapat ditulis dalam bentuk matriks state space dengan dimensi  $3 \times 3$ . Tidak semua derivatif dari sinyal referensi perlu digunakan dalam struktur sistem, dikarenakan struktur CGT model *following* merupakan representasi dari sinyal referensi. Sehingga dalam perancangan ini dilakukan modifikasi, (3.38) dapat ditulis sebagai berikut:

$$\begin{aligned} \Delta(p)y_d &= 0.02p = 0 \\ \dot{y}_d &= \begin{bmatrix} 0 & 1 \\ 0 & -0.02 \end{bmatrix} y_d = G y_d \end{aligned} \quad (3.38)$$

dimana  $Y_d$  menotasikan state referensi sistem pada posisi  $Y$ .

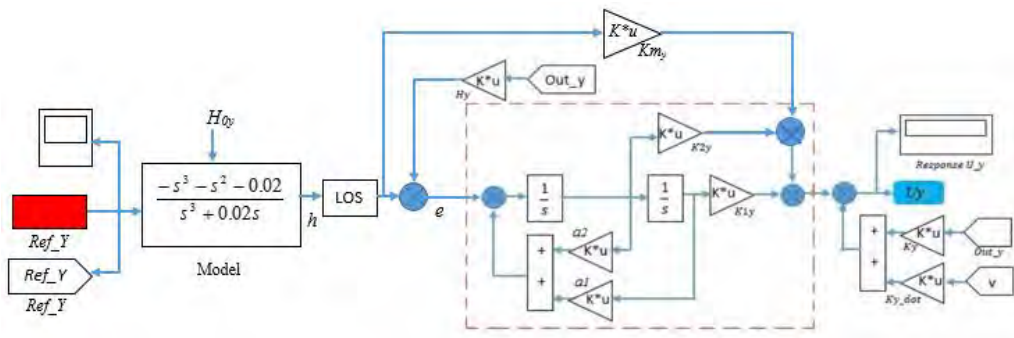
Bentuk dari modifikasi dari dinamika referensi dan dinamika sistem untuk posisi  $Y$  ditunjukkan oleh (3.20) dan (3.21) adalah sebagai berikut:

$$\dot{\tilde{x}}_y = \begin{bmatrix} \dot{\varepsilon} \\ \ddot{\varepsilon} \\ \dot{y} \\ \ddot{y} \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -0.02 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -136.2683 \\ 0 & 0 & 0 & 0 & -15 \end{bmatrix} \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ y \\ \dot{y} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 15 \end{bmatrix} U_1 \quad (3.39)$$

dan persamaan *output* menjadi

$$\tilde{Y}_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ y \\ \dot{y} \\ v \end{bmatrix} \quad (3.40)$$





Gambar 3.5 Diagram simulink kontroler untuk posisi  $Y$

Diagram Simulink untuk kontrol posisi pada sumbu  $Y$  dapat dilihat pada Gambar 3.5, dimana yang menjadi bagian dari CGT ditandai oleh garis putus-putus. Model following sudah dalam bentuk fungsi transfer (3.36). Algoritma LOS digunakan untuk menghitung referensi  $Y$  yang akan dibandingkan dengan aktual  $Y$ .

Tabel 3.3 Nilai yang digunakan sebagai parameter kontrol posisi  $X$  dan  $Y$

Parameter	Kontrol Posisi $X$	Kontrol Posisi $Y$
$K_0$	$[-9.5360 \quad -23.5205 \quad 23.2489 \quad 10.5217]$	$[10 \quad 20.1052 \quad -20.3326 \quad -10.2060]$
$R$	1	1
$Q$	diag {500, 1, 0.1, 500, 10}	diag {400, 5, 25, 40, 1}
$a$	0.01	0.01

Nilai gain kontrol dicari menggunakan metode LQ *regulator* dengan *output feedback* pada kontrol posisi  $X$  dan  $Y$ . Algoritma kontrol yang digunakan terdapat dalam Tabel 3.4 dan kondisi perlu untuk solusi permasalahan tersebut terdapat pada (2.134), sedangkan. Nilai-nilai dari parameter yang digunakan untuk mencari nilai gain  $K$  ditunjukkan dalam Tabel 3.3.

Tabel 3.4 Kontroler *Outer loop* (Posisi)

<b>Algoritma Kontroler <i>Outer loop</i> (Posisi)</b>
1. Inisialisasi nilai $Q$ , $R$ dan gain awal $K_0$ sehingga nilai $A - BK_0C$ adalah stabil <i>asymptotic</i> dengan performa indeks
$J = \frac{1}{2} \int_0^{\infty} (z^T Q' x + 2x^T W u + u^T R' u) dt$
2. Untuk setiap iterasi ke-n: Atur $A_n = A - B_n K C$ kemudian cari nilai $P_n$ dan $S_n$ dalam Persamaan (2.134)
3. Atur $J_n = tr(P_n X)$ evaluasi nilai gain $K$ dengan

---

**Algoritma Kontroler Outer loop (Posisi)**

---

$\Delta K = R^{-1}B^T PSC^T (CSC)^T - K_n$  kemudian update nilai gain  $K$  dengan  $K_{n+1} = K_n + \alpha\Delta K$

dimana  $\alpha$  dipilih sehingga  $A_n = A - B_n K C$  adalah stabil *asymptotic* dan jika,  $J_{n+1} \leq J_n$ , konvergen (cukup dekat satu sama lain), maka lanjut ke tahap 4, jika selain itu set  $n = n + 1$  dan kembali ke tahap 2.

4. Akhiri dengan set  $K = K_{n+1}, J = J_{n+1}$

---

Untuk kontroler posisi  $X$ , matriks  $P$ ,  $S$  dan  $K$  diperoleh setelah iterasi ke-13 dengan data sebagai berikut:

$$P_x = \begin{bmatrix} 996.5 & 1012.2 & -19.8 & -19.8 & 3.6 \\ 1012.2 & 1562.8 & -42.5 & -42.5 & -7.3 \\ -519.8 & -1068 & 41.8 & 41.8 & 7.9 \\ -19.8 & -42.5 & 23.2 & 23.2 & 3.8 \\ -3.6 & -7.3 & 7.9 & 3.8 & 7.3 \end{bmatrix} \quad (3.41)$$

$$S_x = \begin{bmatrix} 3.6800 & -0.5000 & 1.5178 & -0.5100 & -0.0574 \\ -0.5000 & 1.5914 & 0.5100 & 1.5643 & -0.0187 \\ 1.5178 & 0.5100 & 1.5340 & -0.5000 & -0.1325 \\ -0.5100 & 1.5643 & -0.5000 & 3.6122 & -0.0183 \\ -0.0574 & -0.0187 & -0.1325 & -0.0183 & 0.9462 \end{bmatrix} \quad (3.42)$$

$$\tilde{K}_x = [-4.4518 \quad -9.6604 \quad 9.4440 \quad 5.3433] \quad (3.43)$$

Untuk kontroler posisi  $Y$ , setelah iterasi ke-28 didapat nilai matriks  $P$ ,  $S$  dan  $K$  sebagai berikut

$$P_y = \begin{bmatrix} 575.5783 & 408.1571 & -141.8171 & -4.9340 & 2.6876 \\ 408.1571 & 441.7089 & -195.5831 & -7.2252 & 3.6818 \\ -141.8171 & -195.5831 & 137.8846 & 4.8847 & -2.8078 \\ -4.9340 & -7.2252 & 4.8847 & 1.8853 & -0.8695 \\ 2.6876 & 3.6818 & -2.8078 & -0.8695 & 7.9339 \end{bmatrix} \quad (3.44)$$

$$S_y = \begin{bmatrix} 3.6800 & -0.50 & 1.2450 & -0.4753 & 0.0160 \\ -0.50 & 1.2350 & 0.4753 & 2.1803 & 0.0033 \\ 1.2450 & 0.4753 & 2.1708 & -0.50 & 0.0706 \\ -0.4770 & 2.1803 & -0.50 & 9.6152 & 0.0037 \\ 0.0160 & 0.0033 & 0.0706 & 0.0037 & 0.3017 \end{bmatrix} \quad (3.45)$$

$$\tilde{K}_y = [15.9326 \quad 23.4931 \quad -15.8108 \quad -6.2181] \quad (3.46)$$

### 3.5 Blok Koreksi

Dari (2.139), akan dilakukan perhitungan matematika hingga mendapatkan persamaan input  $\phi_d$  dan  $\theta_d$  untuk *inner loop*. Pindahkan kecepatan gravitasi ke sebelah kanan persamaan dan kalikan kedua belah sisi dengan RT, maka (2.78) dapat ditulis menjadi:

$$R^T (Uv + gz_E) = \frac{F_T}{m} Z_E \quad (3.47)$$

$$\begin{pmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ -C\phi S\psi + S\phi S\theta C\psi & C\phi C\psi + S\phi S\theta S\psi & S\phi C\theta \\ S\phi S\psi + C\phi S\theta C\psi & -S\phi C\psi + C\phi S\theta S\psi & C\phi C\theta \end{pmatrix} \begin{pmatrix} U_x \\ U_y \\ U_z + g \end{pmatrix} = \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ F_T \end{pmatrix} \quad (3.48)$$

dimana  $U_z = U_1$  adalah input kontrol yang dihasilkan oleh kontroler posisi  $Z$ . Setelah melakukan perhitungan sederhana, maka didapat persamaan-persamaan berikut:

$$U_x C\theta C\psi + U_y C\theta S\psi - (U_z + g)S\theta = 0 \quad (3.49)$$

$$U_x (-C\phi S\psi + S\phi S\theta C\psi) + U_y (C\phi C\psi + S\phi S\theta S\psi) + (U_z + g)S\phi C\theta = 0 \quad (3.50)$$

$$U_x (S\phi S\psi + C\phi S\theta C\psi) + U_y (-S\phi C\psi + C\phi S\theta S\psi) + (U_z + g)C\phi C\theta = \frac{F_T}{m} \quad (3.51)$$

bagi kedua sisi dari (3.50) dengan  $\cos\theta$ , sehingga sudut *pitch* yang diinginkan  $\theta_d$  adalah sebagai berikut:

$$U_x C\psi + U_y S\psi - (U_z + g) \frac{S\theta}{C\theta} = 0; \theta_d = \arctan\left(\frac{U_x C\psi + U_y S\psi}{(U_z + g)}\right) \quad (3.52)$$

Sudut *roll* yang diinginkan dicari dengan cara mengalikan (3.51) dengan  $\cos\phi$

$$U_x (-C^2\phi S\psi + C\phi S\phi S\theta C\psi) + U_y (C^2\phi C\psi + C\phi S\phi S\theta S\psi) + (U_z + g)C\phi S\phi C\theta = 0 \quad (3.53)$$

dan mengalikan (3.52) dengan  $\sin\phi$  hasilnya adalah,

$$U_x (S^2\phi S\psi + S\phi C\phi S\theta C\psi) + U_y (-S^2\phi C\psi + S\phi C\phi S\theta S\psi) + (U_z + g)S\phi C\phi C\theta = \frac{F_T}{m} S\phi \quad (3.54)$$

kemudian kurangi (3.53) dengan (3.54), sehingga menghasilkan

$$U_x \sin\psi - U_y \cos\psi = \frac{F_T}{m} S\phi \quad (3.55)$$

Pindahkan kecepatan gravitasi pada (2.139) ke ruas sebelah kiri, kemudian mengalikan dengan transpose-nya maka akan didapat hubungan seperti dibawah ini

$$(U_v + gz_E)^T (U_v + gz_E) = \left( \frac{F_T}{m} z_E \right)^T \left( \frac{F_T}{m} z_E \right) U_x^2 + U_y^2 + (U_z + g)^2 = \left( \frac{F_T}{m} \right)^2 \quad (3.5)$$

$$\sqrt{U_x^2 + U_y^2 + (U_z + g)^2} = \frac{F_T}{m}$$

Substitusi (3.57) kedalam (3.56) maka akan dihasilkan sudut *roll*  $\phi_d$  yang diinginkan adalah

$$\phi_d = \arcsin \left( \frac{U_x \sin \psi - U_y \cos \psi}{\sqrt{U_x^2 + U_y^2 + (U_z + g)^2}} \right) \quad (3.57)$$

Persamaan (3.52) dan (3.57) adalah input sudut referensi attitude ( $\phi_d$  dan  $\theta_d$ ) untuk *inner loop* dari sistem.

### 3.6 Perancangan Kontroler *Inner loop* (Rotasi)

Persamaan LTI pada (2.106)-(2.108) pada dinamika quadcopter yang telah *dilinearisasi* dimana, sistem dikatakan memiliki performa  $H_\infty$  jika memenuhi pertidaksamaan (2.113). *Gain* kontrol akan dicari pada perancangan ini dengan teknik numerik *iterative* LMI dimana, untuk memutuskan bentuk LMI didefinisikan dengan indeks performa dari (2.113) adalah sebagai berikut:

$$J = \int_0^{\infty} (\gamma^2 w(t)^T - z(t)^T z(t)) dt \quad (3.58)$$

dengan  $\gamma > 0$ . Kuadrat norm dari *input* dan *output* didalam integral akan dikalikan dengan nilai pembobot scalar yang berlawanan tanda.

Performa  $H_\infty$  yang dimiliki pada sistem (2.106)-(2.108) jika dan hanya jika terdapat fungsi Lyapunov yang memenuhi (2.113). Selanjutnya, nilai  $V(x)$  dari (2.113) diturunkan terhadap waktu,

$$\frac{dV(x(t))}{dt} = ((A - BKC)x + Dw)^T Px + x^T P((A - BKC)x + Dw) \quad (3.59)$$

$$\frac{dV(x(t))}{dt} = x^T ((A - BKC)x + Dw)^T P + P(A - BKC)x + w^T D^T Px + x^T PDw$$

Dari kondisi tersebut dilakukan manipulasi perhitungan sehingga diperoleh,

$$\begin{aligned} \frac{dV(x(t))}{dt} &= x^T ((A - BKC)^T P + P(A - BKC))x - \left( \frac{1}{2} D^T Px - \gamma w \right)^T \left( \frac{1}{\gamma} D^T Px - \gamma w \right) \\ &+ \frac{1}{\gamma^2} x^T P D D^T Px + \gamma^2 w^T w \end{aligned} \quad (3.60)$$

Jika (3.30) disubstitusikan kedalam (2.113)

$$\frac{dV(x(t))}{dt} \leq x(t)^T \left( \frac{1}{\gamma^2} PDD^T P - Q - C^T K^T RKC \right) \quad (3.61)$$

$$\Rightarrow x(t)x(t)^T [(A - BKC)^T P + P(A - BKC) + \frac{1}{\gamma^2} PDD^T P + Q + C^T K^T RKC] x(t) \leq 0$$

$$\Rightarrow x(t)x(t)^T [A^T P + PA + Q - PBKC - C^T K^T B^T P + C^T K^T RKC + \frac{1}{\gamma^2} PDD^T P] x(t) \leq 0$$

Substitusi gain  $K$  pada (2.110) kedalam (3.33), sehingga menghasilkan pertidaksamaan sebagai berikut:

$$A^T P + PA + Q + \frac{1}{\gamma^2} PDD^T P - PBR^{-1}B^T P + L^T R^{-1}L \leq 0 \quad (3.62)$$

Dengan menerapkan metode *Schur complement*, maka bentuk *iterative* LMI dari (3.34) adalah sebagai berikut:

$$\begin{bmatrix} P_n A + A^T P_n + Q + L_n^T R^{-1} L_n & P_n B & P_n D \\ B^T P_n & R & 0 \\ D^T P_n & 0 & -\gamma^2 I \end{bmatrix} \leq 0 \quad (3.63)$$

dimana  $n$  menotasikan iterasi ke- $n$ .

Dengan mengikuti permasalahan LMI optimasi maka kontroler *output feedback* dengan performa  $H_\infty$  dapat diselesaikan secara efektif berikut:

$$\text{Minimize } \gamma \quad (3.64)$$

$$\text{Subject to } P > 0, \gamma > 0, S(P, \gamma) < 0$$

Jika permasalahan optimisasi (3.36) dicapai, dengan mendapatkan nilai matriks  $P$ , maka gain kontroler *Static Output Feedback* dinyatakan oleh

$$K = R^{-1} (B^T P + L) C^T (C C^T)^{-1} \quad (3.65)$$

Diagram blok yang didesain untuk kontrol stabilisasi ini dapat dilihat pada Gambar 3.2. *State* sistem hasil linierisasi untuk model *roll*, *pitch* dan *yaw* menjadi

$$x_{inner} = [\phi \ p \ v \ \theta \ q \ \psi \ r]^T$$

Variabel utama pada perancangan kontroler rotasi yang dikontrol adalah sudut *roll* dan *pitch*, juga tiga kecepatan sudut yaitu kecepatan sudut *roll*, *pitch* dan *yaw*. Oleh karena itu, vektor *output* dari kontroler ini adalah

$$y_{inner} = [\phi \ \theta \ \psi \ p \ q]^T.$$

Simulasi dilakukan dengan nilai parameter matriks sistem sebagai berikut

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 800 \\ 0 & 0 & -15 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 15 \end{bmatrix} \Delta u; y_\phi = C_\phi \begin{bmatrix} \phi \\ \dot{\phi} \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ v \end{bmatrix} \quad (3.66)$$

dengan nilai performa output yang diinginkan adalah

$$Z_\phi = Hx_\phi \begin{bmatrix} 0.3 & 0 & 0 \\ \phi \\ \dot{\phi} \\ \nu \end{bmatrix} \quad (3.67)$$

sedangkan matriks sistem untuk sudut *pitch* adalah

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\nu} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 800 \\ 0 & 0 & -15 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \nu \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 15 \end{bmatrix} \Delta u; y_\phi = C_\phi \begin{bmatrix} \phi \\ \dot{\phi} \\ \nu \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \nu \end{bmatrix} \quad (3.68)$$

$$Z_\phi = Hx_\phi \begin{bmatrix} 0.3 & 0 & 0 \\ \phi \\ \dot{\phi} \\ \nu \end{bmatrix} \quad (3.69)$$

State rotasi disekitar titik equilibrium merupakan nilai matriks  $H$  dipilih berdasarkan performa output yang diinginkan. Dinamika sistem untuk gerak  $\phi$  sama dengan dinamika gerak  $\theta$  dikarenakan nilai  $J_{xx} = J_{yy} = 0.03 \text{ Kg.m}^2$ .

Sistem dengan kondisi stabil asymptotic akan memenuhi performa  $H_\infty$  dimana tingkat pelemahan gangguan  $w(t)$  terhadap performa keluaran  $z(t)$  kurang dari  $\gamma$  pada sistem (3.67)–(3.69) sehingga memenuhi spesifikasi desain yang sudah ditentukan. Untuk memenuhi spesifikasi desain tersebut, maka harus terdapat matriks simetris  $P$  yang memenuhi LMI (3.64). Tabel 3.5 adalah langkah–langkah yang dilakukan dalam simulasi.

Untuk menyelesaikan LMI (3.64), dapat menggunakan LMI toolbox yang tersedia dalam *software* Matlab. Hasil terbaik diperoleh dengan nilai parameter sebagai berikut:

$$\gamma = 0.7, R = 10 \text{ dan } Q = \text{diag}\{755, 3, 0.1\}$$

Tabel 3.5 Algoritma Kontrol Rotasi [10]

<b>Algoritma Output feedback dengan performa <math>H_\infty</math></b>
1. Inisialisasi: Set nilai $n = 0$ , definisikan $L_0 = 0$ tentukan nilai $\gamma$ , nilai toleransi $\alpha$ , matriks pembobot $Q$ dan $R$
2. Penyelesaian Iterasi ke- $n$ Mencari nilai $P_n$ untuk iterasi ke- $n$ pada Persamaan (3.85) Evaluasi nilai $K$ dan update nilai $L$ $K_{n+1} = R^{-1}(B^T P_n + L_n)C^T (CC^T)^{-1}$ $L_{n+1} = RK_{n+1}C - B^T P_n$ jika nilai $K_n$ dan $K_{n+1}$ cukup dekat satu sama lain, lanjut ke langkah 3, sebaliknya, set $n = n + 1$ dan kembali ke langkah 2 (mencari $P_n$ yang <i>feasible</i> )
3. Stop: set $K = K_{n+1}$

setelah iterasi ke-12 maka, hasil matriks  $P$ ,  $K$ , dan  $L$  adalah

$$P = \begin{bmatrix} 29.8331 & 0.1572 & 0.4897 \\ 0.1572 & 0.0156 & 0.0545 \\ 0.4897 & 0.0545 & 0.3359 \end{bmatrix} \quad (3.70)$$

$$K = [0.7346 \quad 0.0817] \quad (3.71)$$

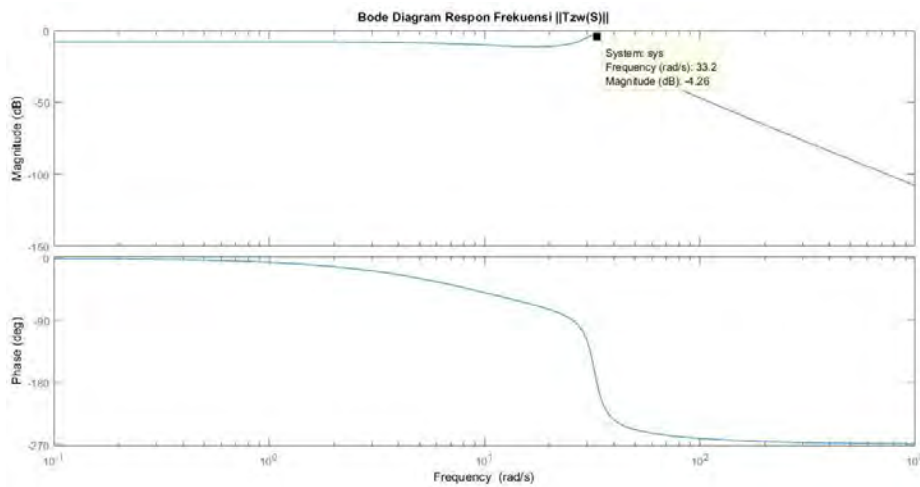
Fungsi alih lup tertutup dari  $Z(s)$  terhadap  $W(s)$  sebagai berikut

$$T_{zw}(s) = H(sI - A + BKC)^{-1}D; T_{zw}(s) = \frac{4050}{s^3 + 15s + 1102.7s + 9916.17} \quad (3.72)$$

Selanjutnya dari (3.72) dicari nilai norm- $\infty$  dari  $T_{zw}(s)$ . Berdasarkan [15], nilai norm- $\infty$  dari  $T_{zw}(s)$  adalah puncak dari nilai singular terbesar dari  $T_{zw}(j\omega)$ , Nilai  $\|T_{zw}(s)\|_{\infty}$  yaitu:

$$\|T_{zw}(s)\|_{\infty} = \gamma^* = 0.67 \quad (3.73)$$

Tingkat pelemahan ini kurang dari nilai  $\gamma$  yang dipilih yaitu  $\gamma = 0.7$  atau  $\|T_{zw}(s)\|_{\infty} < \gamma$  dimana persamaan diatas menunjukkan bahwa norm- $\infty$  dari fungsi-



Gambar 3.6 Bode diagram respon frekuensi dari  $\|T_{zw}(s)\|_{\infty}$

alih lup tertutup sistem memiliki tingkat pelemahan  $\gamma^* = 0.67$ . Nilai  $K$  akan menjamin pengaruh terburuk gangguan  $w(t)$  terhadap performa output  $z(t)$ .

### 3.7 Pemodelan Efek Hembusan Angin (*Wind gust*)

Pengaruh kondisi meteorologi harus dipertimbangkan dalam desain kontrol quadcopter. Kondisi meteorologi dapat berwujud perubahan suhu, tekanan, kerapatan udara dan pergerakan angin. Angin memiliki pengaruh besar pada dinamika terbang quadcopter. Sementara, pengaruh perubahan suhu, tekanan dan perubahan kepadatan udara tidak begitu berpengaruh pada ketinggian operasional penerbangan quadcopter.

Analisis efek angin memisahkan dua komponen dasar, yaitu komponen konstan (sistematis) dan variabel (turbulen) komponen. Komponen konstan merupakan nilai konstan kecepatan angin yang dimodelkan sebagai sinyal *step*. Komponen variabel merupakan '*gust*' atau hembusan angin sesaat. Estimasi pengaruh kondisi meteorologi utama dibuat sebagai aturan yang menggunakan model statistik dari kondisi meteorologi penyimpangan dari nilai iklimnya [18]. Kecepatan angin dipergunakan akan dibangkitkan secara acak.

Efek angin dipertimbangkan dengan sudut pandang makro. Dalam pendekatan makro, quadcopter terhubung di titik  $r$ . Titik  $r$  terletak pada titik pusat rangka quadcopter. Pengaruh angin pada semua bagian quadcopter memiliki nilai dan arah yang sama untuk setiap titik waktu.

#### 3.7.1 Pemodelan *Disturbance* Kecepatan Angin Konstan (*Step*)

Model kecepatan angin konstan '*step*' didasarkan pada [18]. Besarnya kecepatan angin konstan *step* dengan magnitude yang berbeda pada saat  $t$  dengan fungsi parsial berikut:

$$V_{si} = \begin{cases} 0, & t < t_{0i} \\ V_{mi} \left(1 - \cos\left(\frac{\pi(t-t_{0i})}{2t_{di}}\right)\right), & t_{0i} < t \leq t_{0i} + t_{di} \\ V_{mi}, & t > t_{0i} + t_{di} \end{cases} \quad (3.74)$$

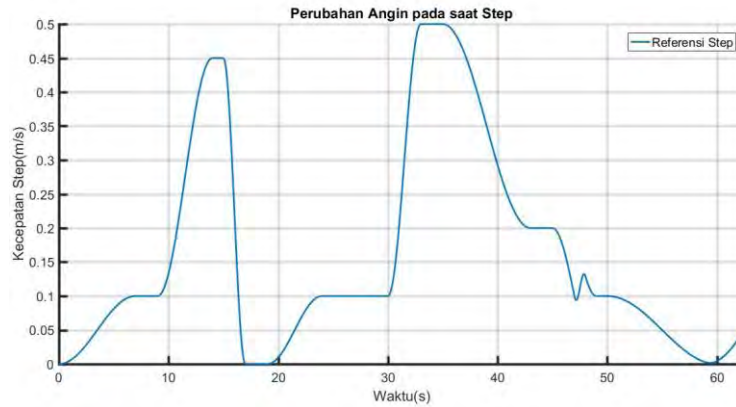
dimana  $d_n$  adalah interval dari peningkatan angin,  $V_m$  adalah *gust magnitude*,  $d_m$  adalah durasi *gust*, dan  $t_m$  waktu terbang.

Hasil simulasi pada:

- $t_0 = [0; 9; 16; 19]$ ,
- $V_m = [1; 4.5; 0.1; ]$ ,
- $d_n = [7, 5, 3, 5]$ ,
- $t_m = 25s$ ,  $V_0 = 0,5 \frac{M}{s}$

ditunjukkan pada gambar 3.7





Gambar 3.7 Perubahan kecepatan angin *step*

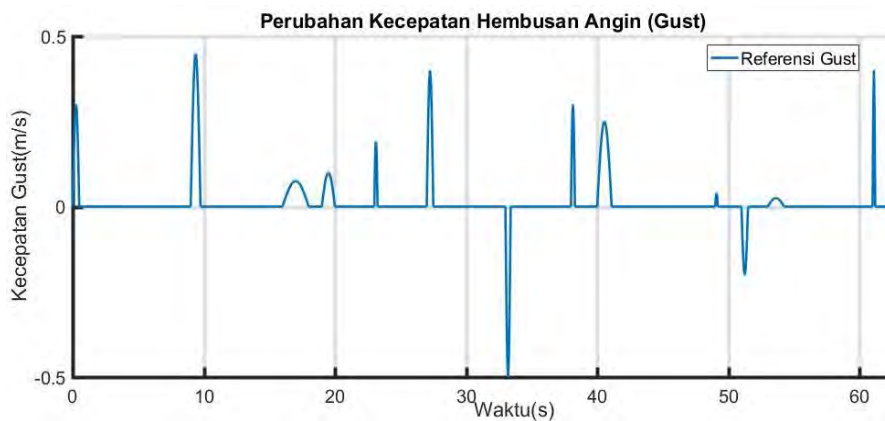
Hasil simulasi menunjukkan bahwa model (3.74) memungkinkan simulasi peningkatan dan penurunan kecepatan angin dengan mempertimbangkan kecepatan angin sebelum "*step*". Dengan menggunakan model ini komponen yang sistematis  $V_s$  kecepatan angin dapat disimulasikan.

### 3.7.2 Pemodelan *Disturbance* Kecepatan Angin *Gust*

Model kecepatan angin model '*gust*' didasarkan pada [18]. Besarnya kecepatan angin '*gust*' pada saat  $t$  dengan fungsi parsial berikut:

$$V_{gi} = \begin{cases} 0, & t < t_{0i} \\ V_{mi} \left( \sin\left(\frac{\pi(t - t_{0i})}{t_{di}}\right) \right), & t_{0i} < t \leq t_{0i} + t_{di} \\ V_{mi}, & t > t_{0i} + t_{di} \end{cases} \quad (3.75)$$

dimana  $d_n$  adalah interval dari peningkatan angin,  $V_m$  adalah *gust magnitude*,  $d_m$  adalah durasi *gust*, dan  $t_m$  waktu terbang.

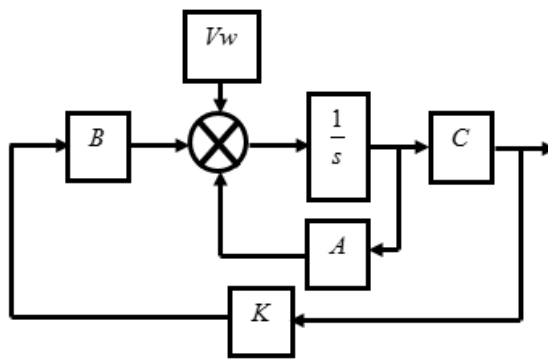


Gambar 3.8 Perubahan kecepatan angin '*gust*'

Hasil simulasi pada

- $t_0 = [0; 9; 16; 19]$ ,
- $V_m = [3; 4.5; 0.75; 1]$ ,
- $D_m = [0.5; 0.75; 2; 1]$ ,
- $t_m = 25s$

ditunjukkan pada gambar 3.8 Hasil simulasi rumus (3.75) menunjukkan bahwa simulasi dengan hembusan dibuat dengan mempertimbangkan komponen variabel dari kecepatan angin  $V_v$ .



Gambar 3.9 Diagram blok untuk simulasi dengan gangguan

## BAB 4

### HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas simulasi dari hasil perancangan sistem pada BAB III. Setelah dilakukan simulasi kemudian mendeskripsikan tiap data pada hasil simulasi apakah semua yang dilakukan sesuai dengan tujuan pada penelitian.

#### 4.1 Hasil Perancangan Sistem Kontrol

Hasil perancangan sistem kontrol didapat setelah dilakukannya perancangan kontroler pada bab 3 khususnya yang telah dijelaskan disubbab 3.4 dan 3.6. analisa dari desain pada kontroler didapat sebagai berikut.

##### 1. Hasil Perancangan kontroler rotasi (*Inner loop*)

Dikarenakan mencari nilai *eigenvalue* dari sebuah sistem rotasi dimana dinamika rotasi  $\phi$  (*Roll*) sama dengan dinamika rotasi  $\theta$  (*Pitch*) maka,  $\det(\lambda I - (A - BK)) = 0$  adalah *eigenvalue* yang didapat dari sistem *closed loop* untuk dinamika  $\phi$  dan  $\theta$ .

$$\lambda = \{-7.7623, -3.6189 \pm j31.703\} \quad (4.1)$$

##### 2. Hasil Perancangan kontroler posisi (*Outer loop*)

- **Dinamika posisi  $X$**

Mencari nilai *eigenvalue* dari sebuah sistem *closed loop* untuk dinamika Posisi  $X$

$$\tilde{A}c_x = \tilde{A}_x - \tilde{B}_x \tilde{K}_x \tilde{C}_x = 0 \quad (4.2)$$

jika nilai matriks (3.55)-(3.56) dan (3.65) disubstitusikan ke (4.1), maka diperoleh *eigenvalue* dari sistem *closed loop* ini adalah:

$$\lambda = \{-6.6157 \pm j45.992, -0.8303, -0.4732 \pm j0.9135\} \quad (4.3)$$

- **Dinamika posisi  $Y$**

Mencari nilai *eigenvalue* dari sebuah sistem *closed loop* untuk dinamika Posisi  $Y$

$$\tilde{A}c_y = \tilde{A}_y - \tilde{B}_y \tilde{K}_y \tilde{C}_y = 0 \quad (4.4)$$

jika nilai matriks (3.61)-(3.62) dan (3.68) disubstitusi ke (4.3), maka *eigenvalue* dari sistem *closed loop* untuk posisi  $Y$  adalah

$$\lambda = \{-6.22 \pm j112.41, -1.17, -0.7 \pm j1.31\} \quad (4.5)$$

Untuk dapat mengetahui matriks  $(Q, A)$  *detectability* maka, dapat dilakukan dengan melihat *rank* dari matrik tersebut,

$$Q = \begin{vmatrix} \sqrt{Q} \\ \sqrt{Q}, \tilde{A} \\ \vdots \\ \sqrt{Q}, \tilde{A}^4 \end{vmatrix} \quad (4.6)$$

Dari hasil diatas maka dapat dikategorikan sebagai kondisi *detectability* terpenuhi dikarenakan hasil substitusi (4.5) dengan nilai matriks  $Q$  pada Tabel 3.4 dan  $\tilde{A}$  pada (3.55) dan (3.61), maka diperoleh bahwa sistem memiliki  $rank = 5$ , atau rank penuh.

penempatan pole dinamika rotasi (4.6) lebih jauh dari penempatan *pole* dinamika translasi (4.2) dan (4.4) dapat dilihat pada (4.2), (4.4), dan (4.6). dari hasil tersebut maka sistem menjamin bahwa kondisi *inner loop* lebih cepat daripada *outer loop*.

## 4.2 Pengujian Sistem Kontrol Quadcopter

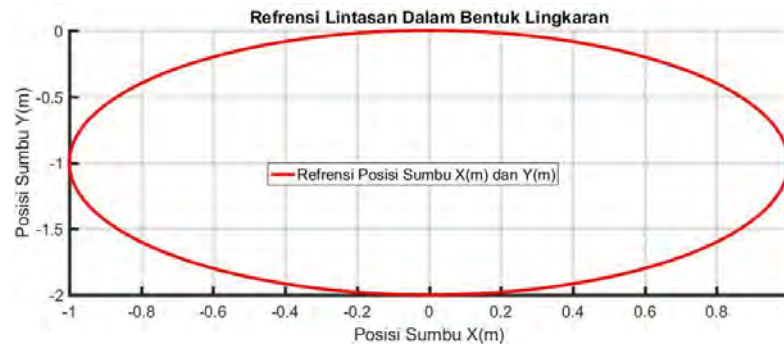
Setelah didapatkan hasil perancangan baik kontroler *Outer loop* (Kontroler Posisi) dan Kontroler *Inner loop* (Kontroler Rotasi) maka dilakukan pengujian respon dari quadcopter. Pengujian tersebut dilakukan pada lintasan yang sudah terdefinisi yang dikhususkan yaitu lintasan lingkaran. Terdapat beberapa pengujian terhadap pergerakan quadcopter dalam melakukan *Path following* yaitu pengujian tanpa adanya *Disturbance*, pengujian dengan adanya *Wind gust* (Hembusan Angin), dan pengujian dengan perubahan nilai  $\gamma$ .

### 4.2.1 Pengujian Sistem Kontrol Quadcopter Tanpa Adanya Gangguan

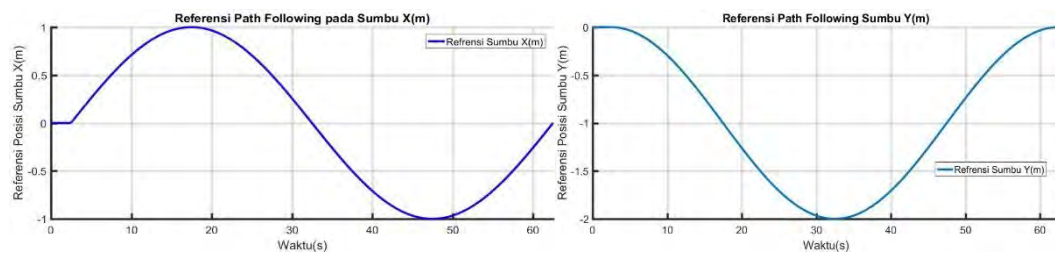
#### A. Pengujian Sistem Kontrol Untuk Gerak Translasi

Pengujian pertama adalah *generate path* dimana bentuk *path* berupa lingkaran dapat dilihat pada Gambar 4.1. Kontroler posisi yang telah didesain adalah dengan memberikan sinyal referensi yang telah terdefinisi yang harus diikuti oleh quadcopter. Referensi pada sumbu  $Z$  hanya diberikan sinyal *step* dengan nilai  $Z_d = 1$  m. Jika dilihat dalam bidang  $X$ ,  $Y$ , dan  $Z$  ilustrasi referensi lintasan dapat dilihat pada Gambar 4.2. Gambar 4.3 menunjukkan sinyal referensi  $(X_d, Y_d)$  yang diberikan kepada quadcopter dalam sumbu  $X$  dan  $Y$ , berturut-turut.

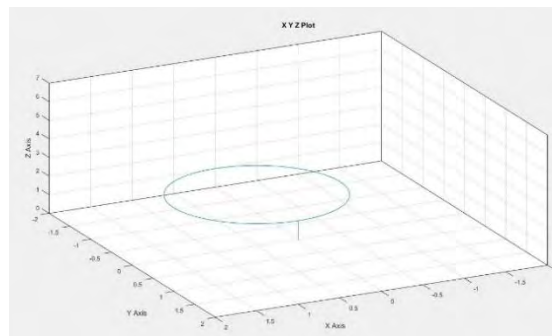
Dalam pengujian kontroler quadcopter, sinyal referensi diberikan pada saat quadcopter sudah berada pada kondisi melayang (*hover*). Pengujian pertama dilakukan dengan memilih nilai parameter  $\gamma$  yang terbaik, yaitu  $\gamma = 0.7$ . sinyal referensi lintasan diberikan pada detik ke-2.5, dimana pada saat waktu tersebut-



Gambar 4.1 *Path* lingkaran



Gambar 4.2 Sinyal referensi posisi pada sumbu  $X$  dan  $Y$



Gambar 4.3 Referensi lintasan dalam bidang  $X$ ,  $Y$ , dan  $Z$

quadcopter telah berada dalam kondisi melayang, seperti yang dapat dilihat pada Gambar 4.2.

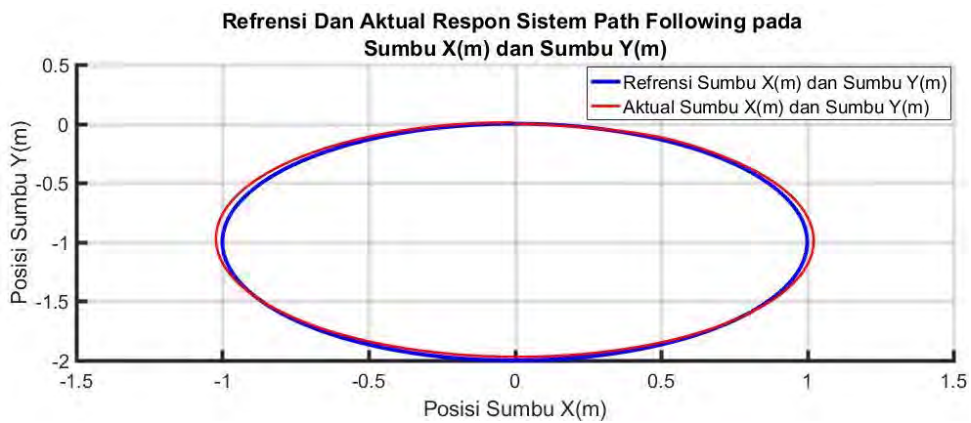
Respon *path following* sistem menunjukkan respon jika dilihat pada bidang  $X$  dan  $Y$  dimana lintasan berbentuk lingkaran terdapat pada Gambar 4.4. Respon dapat mengikuti referensi yang diberikan dengan rata-rata penyimpangan  $\pm 0.13$  m. Dengan menggunakan komputasi menggunakan *software* Matlab maka hasil nilai gain *output feedback* ( $K$ ) dapat diketahui.

Posisi quadcopter pada sumbu  $X$  dapat mengikuti referensi yang berbentuk sinyal sinusoidal ditunjukkan pada gambar 4.5. *Path following* dimulai

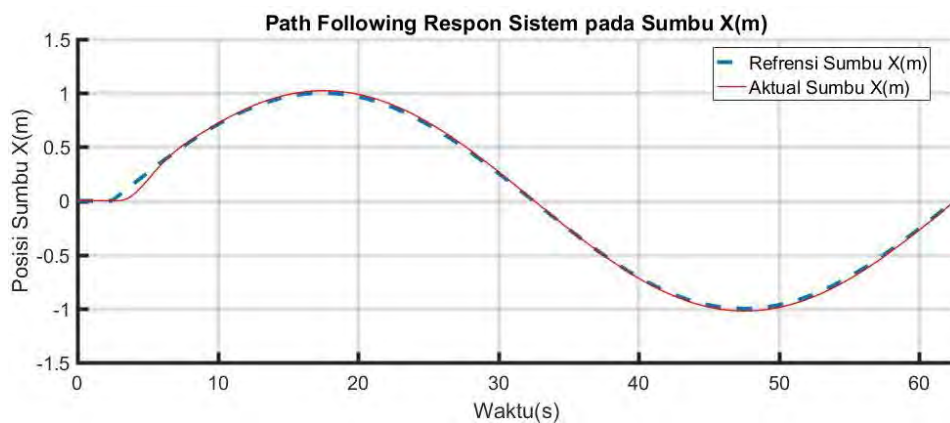
pada detik ke-2.5 (pada awal *path following*), terjadi keterlambatan respon sekitar 0.7 detik yang menyebabkan penyimpangan respon sebesar 0.14 m. Pada detik ke-6.7 respon dapat kembali berada pada lintasan referensi. Nilai Integral Square Error (ISE) *path following* pada posisi  $X$  sebesar 0.07486. Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat *delay*  $\pm 3.17$  detik.

Respon posisi translasi pada sumbu  $Y$  ditunjukkan oleh Gambar 4.6. Respon dapat mengikuti referensi lintasan yang diberikan dengan tepat yang berbentuk cosinus. Pergerakan pada sumbu  $Y$  memiliki nilai ISE sebesar 0.0615.

Sinyal referensi yang diberikan adalah sinyal *step* pada posisi sumbu  $Z$  yang ditunjukkan oleh Gambar 4.7. Secara keseluruhan, kontroler *path following* hasil rancangan dapat menghasilkan kesesuaian terhadap *path* yang sudah terdefinisi. Hasilnya, respon mencapai steady state pada detik ke-2.3 dan respon dapat mengikuti referensi dengan *overshoot* sekitar 0.2 m. Respon dapat mengikuti lintasan yang diinginkan dengan nilai ISE *path following* bernilai kurang dari 0.1.

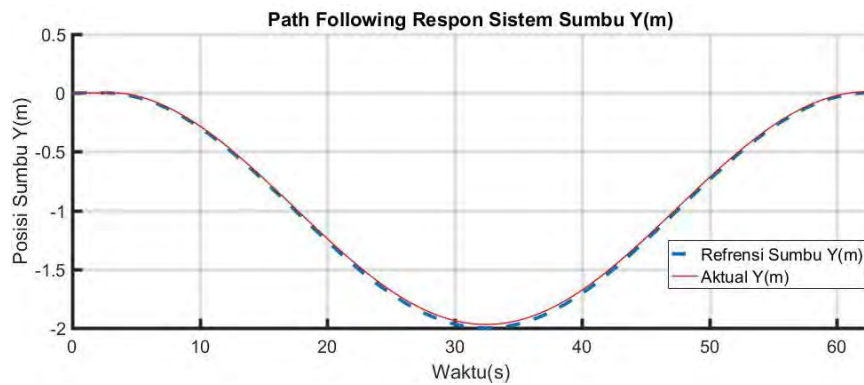


Gambar 4.4 Respon referensi dan aktual *path following* pada sumbu  $X, Y$

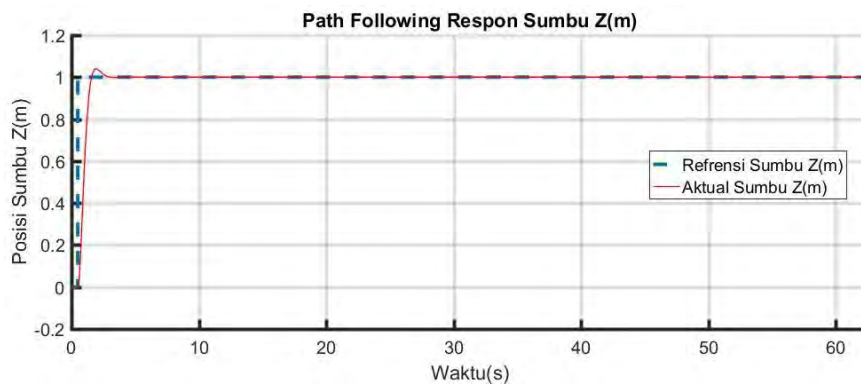


Gambar 4.5 Respon sistem *path following* pada sumbu  $X$

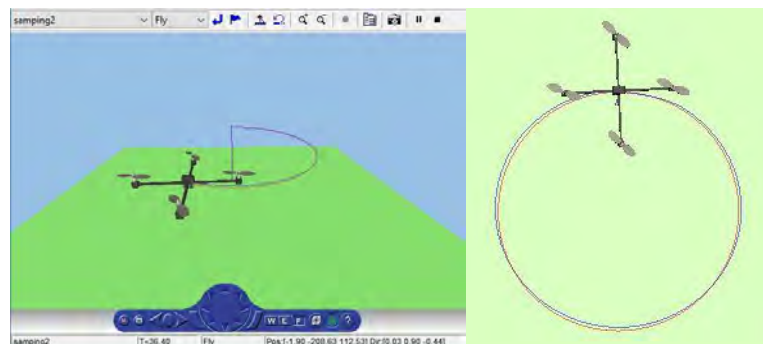
*Virtual world* merupakan blok sistem yang terdapat pada *software* Matlab. Dimana blok tersebut dapat menampilkan animasi quadcopter secara 3D yang dibangun menggunakan *3D World Editor*. Dengan *virtual world* dapat membantu memperjelas hasil simulasi. Selain itu, *Virtual world* juga mampu merepresentasikan dinamika gerak rotasi dan translasi dari quadcopter yang disajikan dalam bentuk tiga dimensi. Dalam *virtual world* ini terdapat dua buah sudut pandang (*view point*) yaitu tampak atas dan tampak samping. Gambar 4.8 merupakan *sniping tool* dari *virtual world* quadcopter ketika simulasi sedang berlangsung, garis berwarna biru merupakan *path* yang dilalui oleh quadcopter.



Gambar 4.6 Respon Sistem *path following* pada sumbu *Y*



Gambar 4.7 Respon sistem *path following* pada sumbu *Z*

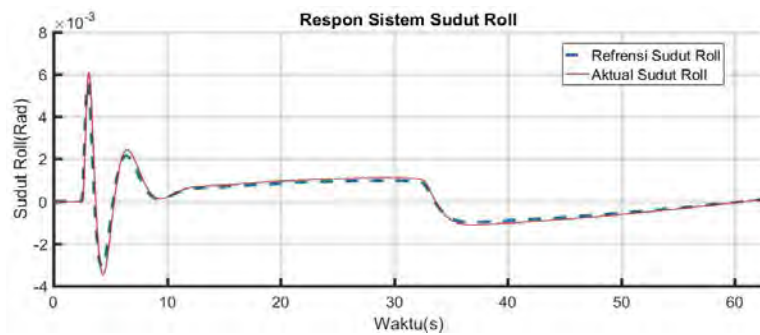


Gambar 4.8 Hasil *sniping tool* dari *virtual world* quadcopter

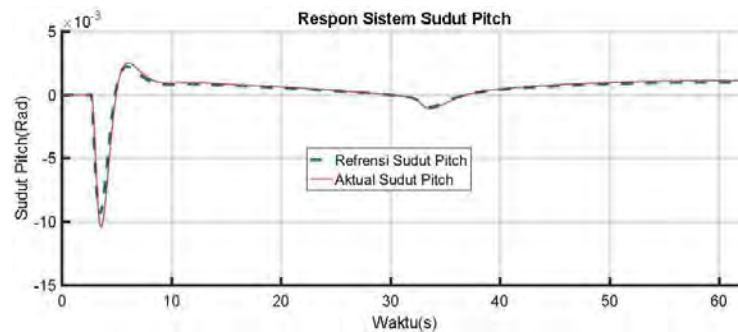
## B. Pengujian Sistem Kontrol Untuk Gerak Rotasi

Input referensi untuk gerak rotasi diperoleh dari output kontroler translasi yang dikonversi menjadi  $\phi_d$  dan  $\theta_d$ . Gambar 4.9 dan 4.10 menunjukkan sinyal referensi dan respon sistem untuk gerak rotasi. Seperti yang telah dibahas pada sub-bab 3.6, dengan rata-rata *error* sebesar 0.01794 rad dapat dilihat bahwa respon sudut *roll* dan *pitch* dapat mengikuti sinyal referensi yang diberikan dari output kontroler posisi (*outer loop*).

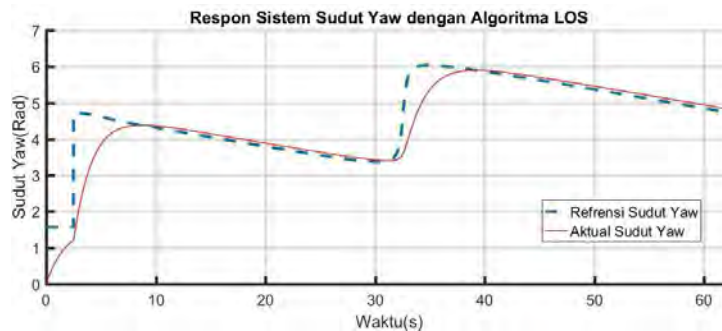
Karena menambahkan algoritma *Line Of Sight* (LOS) yang digunakan untuk menghitung arah hadap dari quadcopter maka respon sistem diambil dari sudut *yaw* juga diperhitungkan.



Gambar 4.9 Respon sistem sudut *roll*



Gambar 4.10 Respon sistem sudut *pitch*



Gambar 4.11 Respon sistem sudut *yaw* dengan algoritma LOS



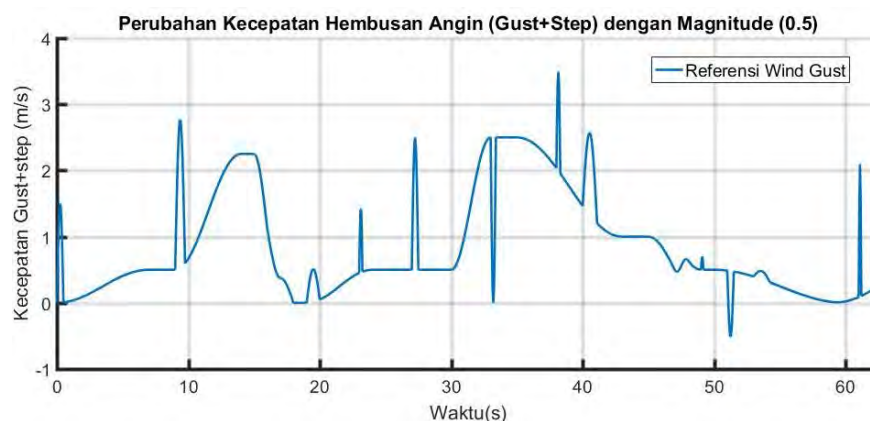
Dapat dilihat bahwa respon sudut *yaw* dapat mengikuti sinyal referensi yang diberikan dari *output* kontroler posisi (*outer loop*), dengan rata-rata *error* sebesar 0.0245 rad. Dari Gambar 4.11 dilihat perbandingan antara referensi dan aktual dari sudut *yaw*, dimana aktual menjaga arah hadap (*heading*) dan memperhalus gerakan quadcopter. Konsep LOS adalah mengambil sudut tangensial sehingga membuat dinamika bergerak menuju referensi *path following* sehingga konvergen menuju nol.

#### 4.2.2 Pengujian Sistem Kontrol Quadcopter dengan Efek *Wind gust*

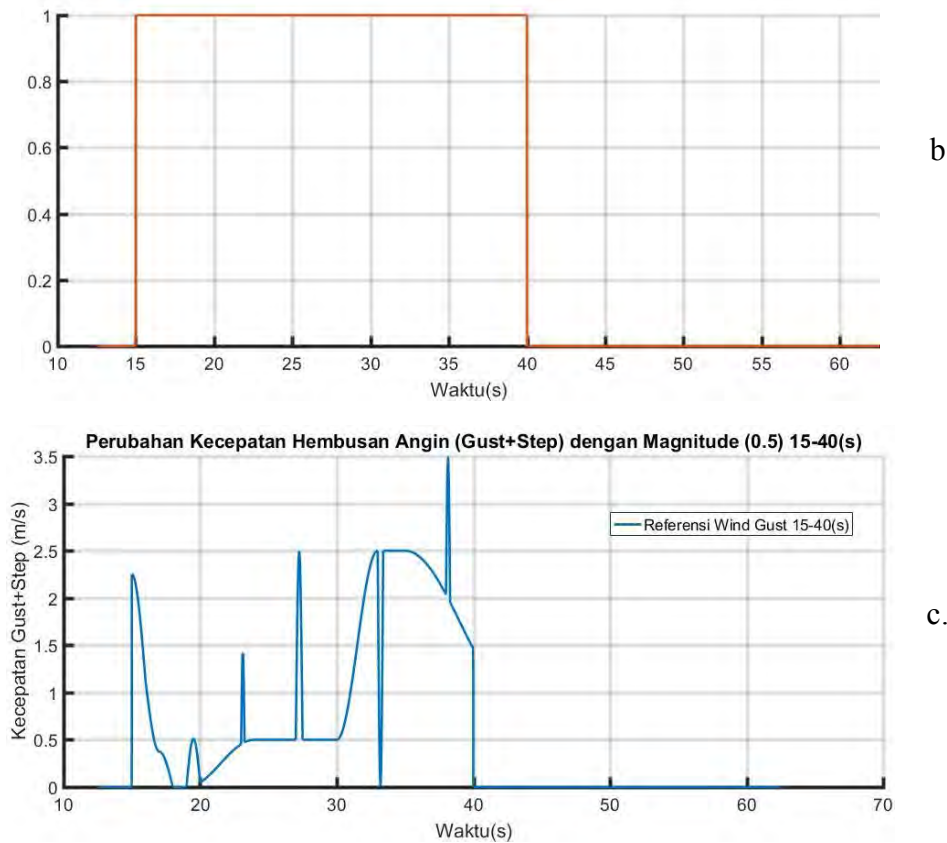
Pengujian respon sistem selanjutnya adalah dengan mensimulasikan efek dari pemodelan *wind gust* yang diberikan pada dinamika quadcopter. Pengujian dilakukan dengan empat cara yaitu pemberian *Disturbance* model angin konstan (*step*) dengan waktu (15-40s), pemberian *Disturbance* model *gust* dengan waktu (15-40s), pemberian *Disturbance* model modifikasi penggabungan *step+gust* sejak awal (0-62.5s) quadcopter bergerak dan pemberian *Disturbance* model modifikasi penggabungan *step+gust* pada waktu tertentu (15-40s). Dari penjelasan pemodelan modifikasi penggabungan *step+gust* baik dengan perubahan kecepatan angin konstan (*step*) (3.95) serta pemodelan kecepatan angin model *gust* (3.96) maka dilakukan modifikasi terhadap bentuk *wind gust*. Pemodelan modifikasi penggabungan *step+gust* tersebut dibentuk dengan cara menjumlahkan (3.95) dan (3.96).

$$V_{si} = \begin{cases} 0, t < t_{0i} & (4.7) \\ V_{mi}(1 - \cos(\frac{\pi(t-t_{0i})}{2t_{di}})), t_{0i} < t \leq t_{0i} + t_{di} \\ V_{mi}(\sin(\frac{\pi(t-t_{0i})}{t_{di}})), t_{0i} < t \leq t_{0i} + t_{di} \\ V_{mi}, t > t_{0i} + t_{di} \end{cases}$$

Dapat dilihat pada Gambar 4.12 (a). Gangguan diberikan pada detik ke-15 hingga detik ke-40 yang diasumsikan quadcopter bergerak dalam ruangan (Gambar 4.12(b)), sehingga gangguan yang masuk ke dalam sistem adalah pada Gambar 4.12 (c).



a.



Gambar 4.12 a. *Wind gust* secara keseluruhan, b. pemberian gangguan pada detik ke-15 hingga detik ke-40 dan c. *Wind gust* pada sistem

**A. Pengujian *Path following* Quadcopter dengan *Disturbance Model* Angin Konstan 15-40(s)**

Model kecepatan angin konstan ‘*step*’ didasarkan pada [11]. Besarnya kecepatan angin konstan *step* pada saat *t* dengan fungsi parsial (3.95). Dimana nilai-nilai parameter yang digunakan sebagai berikut pada

- $t_0 = [0; 9; 15; 19; 30; 35; 45; 47; 50; 59]$ ,
- $V_m = [1; 3.5; -4.5; 14; -3; -2; 1; -14]$ ,
- $d_n = [7; 2; 5; 3; 8; 4; 1; 10; 15]$ ,
- $t_m = 25s$ ,  $V_0 = 0,5 \frac{M}{s}$ .

sehingga bentuk model angin konstan ditunjukkan pada gambar 3.5.

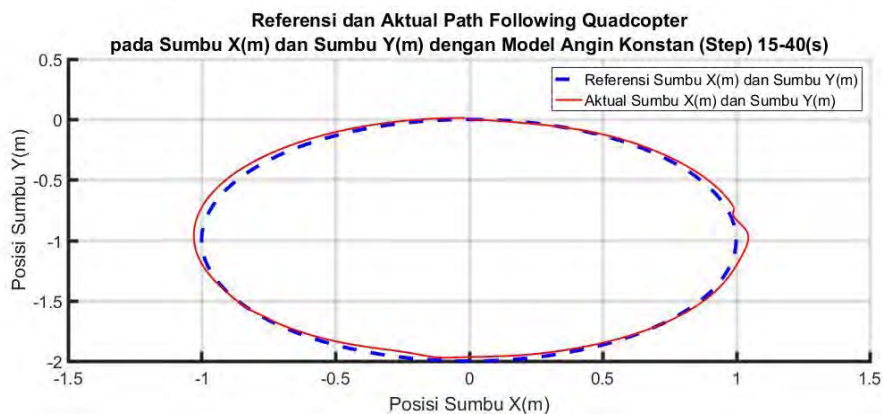
- **Pengujian Sistem Kontrol Untuk Gerak Translasi dengan *Disturbance Model* Angin Konstan 15-40(s)**

Besaran magnitude yang diberikan sebesar 1. Gambar 4.13 menunjukkan respon *path following* quadcopter pada sumbu *X* dan *Y*. Pergerakan rotasi (*Outer loop*) yang menggunakan kontroler *command-generator tracker model following*

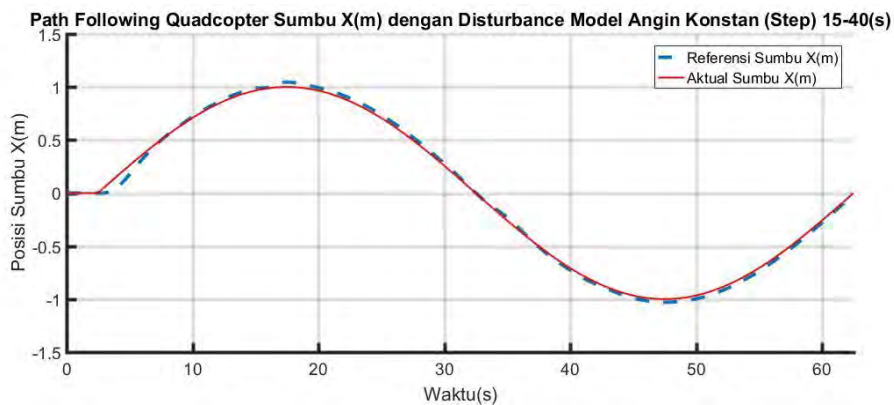
dimana kontroler tersebut mampu mengkompensasi gangguan agar pada gerakan translasi tidak terjadi penyimpangan yang besar saat melakukan *path following*.

Penyimpangan posisi pada sumbu  $X$  terjadi pada detik ke-15 dengan penyimpangan sebesar  $\pm 0.003$  m dan detik ke-40 dengan penyimpangan sebesar  $\pm 0.001$  m. Hasil aktual mengalami percepatan pada detik ke-4 sehingga mendahului referensi. Dari percepatan tersebut terjadi penyimpangan sebesar  $\pm 0.1$ m seperti yang ditunjukkan pada Gambar 4.14. Nilai Integral Square Error (ISE) *path following* pada posisi  $X$  sebesar 0.001312m. Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat percepatan  $\pm 0.04$ . Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat *delay*  $\pm 2.15$  detik.

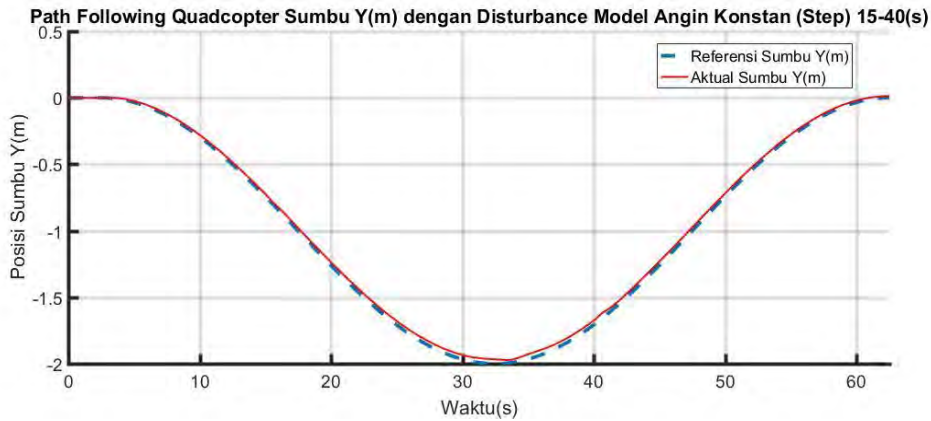
Pada respon posisi pada sumbu  $Y$  yang ditunjukkan pada Gambar 4.15, respon dapat mengikuti referensi dengan tepat. Nilai Integral Square Error (ISE) *path following* pada posisi  $Y$  sebesar 0.000677m.



Gambar 4.13 *Path following* dengan gangguan dalam sumbu  $X, Y$



Gambar 4.14 *Path following* sumbu  $X$  (m) pada saat terdapat angin konstan



Gambar 4.15 *Path following* sumbu  $Y$  (m) pada saat terdapat angin konstan

- **Pengujian Sistem Kontrol Untuk Gerak Rotasi dengan *Disturbance* Model Angin Konstan 15-40(s)**

Gambar 4.16 dan 4.17 menunjukkan respon sudut  $\phi$  dan respon sudut  $\theta$  pada saat terdapat gangguan berupa *wind gust*. Ketika angin berhembus membuat quadcopter menyimpang dari posisi *path following*nya. Pengaruh *wind gust* terhadap sudut  $\phi$  terjadi dari detik ke-15 s/d detik ke-40, pergerakan dari -0.014 rad s/d 0.034 rad. Dengan rata-rata *error* sebesar 0.0125 rad.

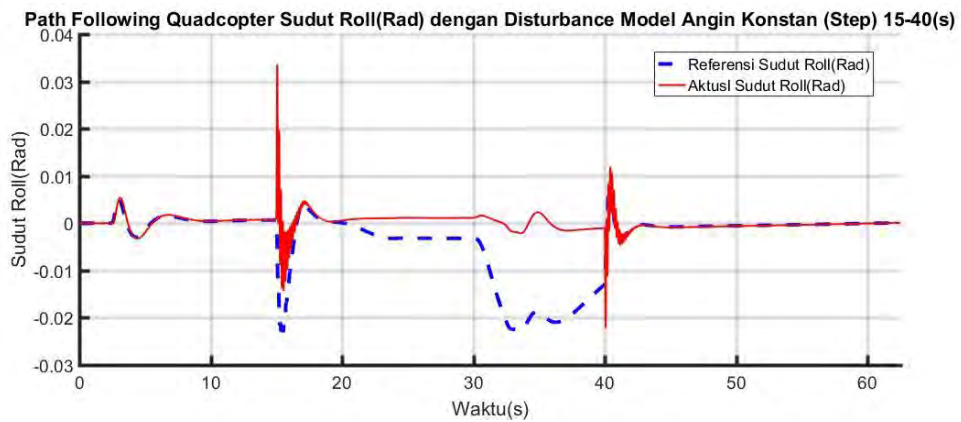
Respon sudut  $\theta$  dapat mengikuti referensinya, namun pada saat terdapat *wind gust*, respon sudut  $\theta$  berosilasi dari -0.012 rad s/d 0.032 rad. Dari hasil yang dilihat baik dari respon sistem pada sudut  $\phi$  dan  $\theta$  saat *wind gust* aktif kontroler mampu meminimalisir gangguan Hal tersebut terjadi untuk membuat quacopter tetap stabil sehingga dapat kembali mengikuti *path*. Dengan rata-rata *error* sebesar 0.001838rad.

Tabel 4.1 Nilai *Integral Square Error* (ISE) tiap perubahan *magnitude stept*

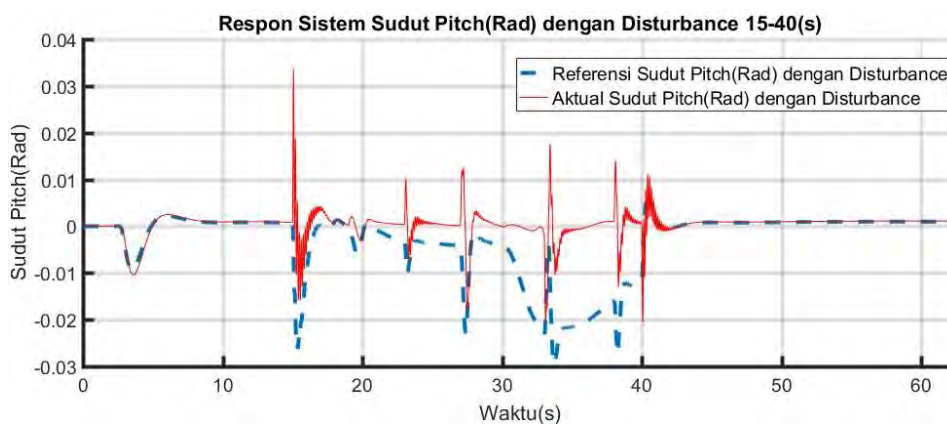
<i>Integral Square Error</i> (ISE)	<i>Wind gust</i>		
	<i>Magnitude</i> (1)	<i>Magnitude</i> (1.5)	<i>Magnitude</i> (2)
<i>Roll</i>	0.001903rad	0.002855rad	0.00386rad
<i>Pitch</i>	0.001838rad	0.002908rad	0.00387rad
<i>Yaw</i>	0.091126rad	0.13669rad	0.1822rad
Posisi X	0.001312m	0.001969m	0.002625m
Posisi Y	0.000677m	0.000451m	0.000602m

Hasil simulasi dengan sinyal *Disturbance* yang berbeda-beda telah disimpulkan pada Tabel 4.1. Sistem kontrol diuji dengan *Disturbance* dengan nilai yang berbeda-beda, yaitu D(1), D(1.5) dan D(2). Berdasarkan nilai ISE dari setiap

pergerakan quadcopter, diperoleh bahwa semakin besar nilai rata-rata *Disturbance* yang diberikan, maka semakin besar pula nilai ISE dari pergerakan quadcopter.



Gambar 4.16 *Path following* sudut *roll* pada saat terdapat angin konstan



Gambar 4.17 *Path following* sudut *pitch* pada saat terdapat angin konstan

Hal ini berpengaruh pula kepada besarnya nilai sinyal kontrol yang masuk ke *plant* quadcopter. Maksimal *Disturbance* pengukuran yang dapat diatasi oleh sistem kontrol adalah sebesar  $D(1)$ .

## B. Pengujian *Path following* Quadcopter dengan *Disturbance Model Gust* 15-40(s)

Model kecepatan angin konstan '*gust*' didasarkan pada [11]. Besarnya kecepatan angin konstan *step* pada saat  $t$  dengan fungsi parsial (3.96). Dimana nilai-nilai parameter yang digunakan sebagai berikut pada :

- $t_0 = [0; 9; 16; 19; 23; 27; 33; 38; 40; 49; 51; 53; 61]$ ,
- $V_m = [3; 4.5; 0.75; 1; 2; 4; -5; 3; 2.5; 0.4; -2; 0.25; 4]$ ,
- $d_n = [0.5; 0.75; 2; 1; 0.25; .5; .4; .3; 1.1; .2; .5; 1.25; 0.2]$ ,

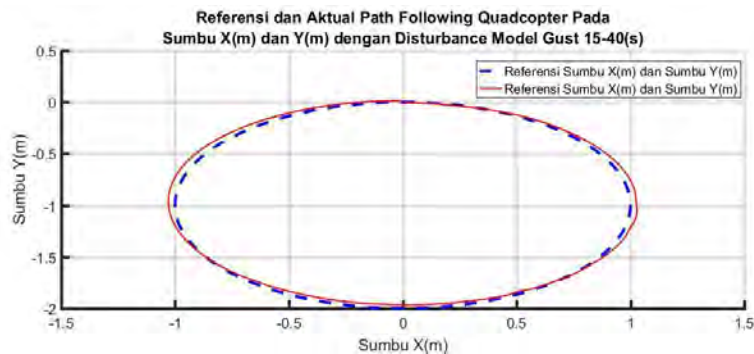
- $t_m = 25s, V_0 = 0,5 \frac{M}{s}$ .

sehingga bentuk model *gust* ditunjukkan pada gambar 3.6.

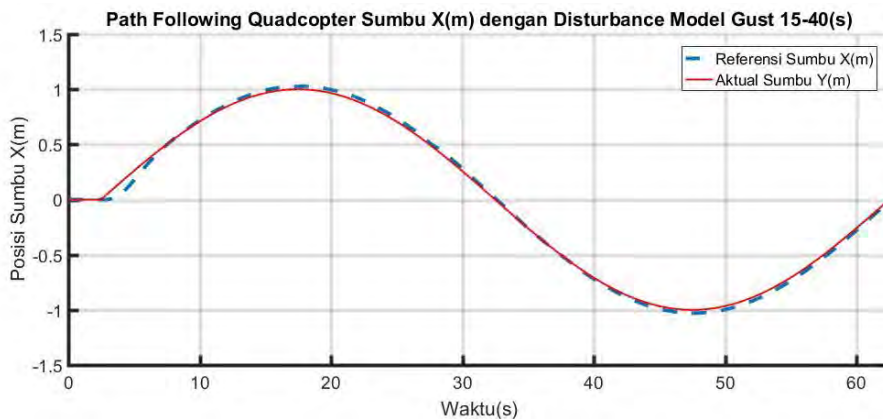
- **Pengujian Sistem Kontrol Untuk Gerak Translasi dengan *Disturbance* Model *Gust* 15-40(s)**

Gambar 4.18 menunjukkan respon *path following* quadcopter pada sumbu  $X$  dan  $Y$  saat terdapat *Disturbance* model *gust*. Besaran magnitude yang diberikan sebesar 1. Penyimpangan posisi pada sumbu  $X$  terjadi pada detik ke-15 dengan penyimpangan sebesar  $\pm 0.0035$  m dan detik ke-40 dengan penyimpangan sebesar  $\pm 0.001$  m.

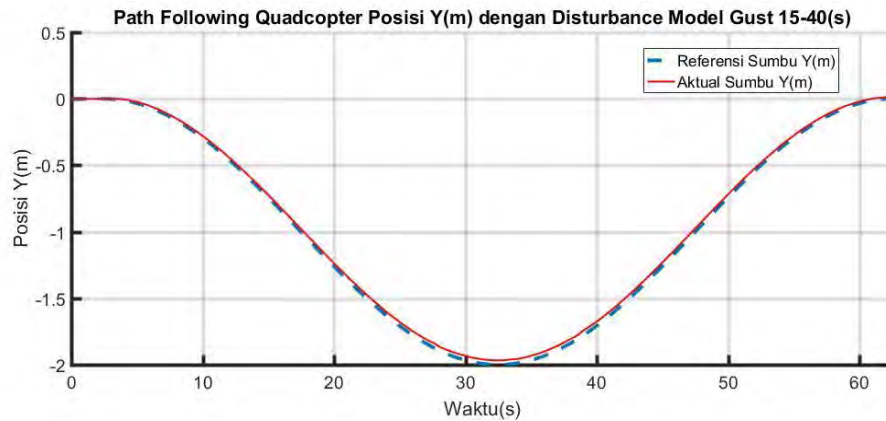
Hasil aktual mengalami percepatan pada detik ke-4 sehingga mendahului referensi. Dari percepatan tersebut terjadi penyimpangan sebesar  $\pm 0.1$ m seperti yang ditunjukkan pada Gambar 4.19. Nilai Integral Square Error (ISE) *path following* pada posisi  $X$  sebesar 0.001451. Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat *delay*  $\pm 2.17$  detik.



Gambar 4.18 *Path following* dengan gangguan dalam sumbu  $X, Y$



Gambar 4.19 *Path following* Sumbu  $X$  (m) pada saat terdapat *gust*



Gambar 4.20 *Path following* Sumbu  $Y$  (m) pada saat terdapat *gust*

Pada respon posisi pada sumbu  $Y$  yang ditunjukkan pada Gambar 4.20, respon dapat mengikuti referensi dengan tepat. Nilai Integral Square Error (ISE) *path following* pada posisi  $Y$  sebesar 0.000289.

- **Pengujian Sistem Kontrol Untuk Gerak Rotasi dengan *Disturbance Model Gust 15-40(s)***

Gambar 4.21 dan 4.22 menunjukkan respon sudut  $\phi$  dan respon sudut  $\theta$  pada saat terdapat gangguan berupa *wind gust*. Ketika angin berhembus membuat quadcopter menyimpang dari posisi *path following*nya. Pengaruh *wind gust* terhadap sudut  $\phi$  terjadi dari detik ke-15 s/d detik ke-40, pergerakan dari -0.18 rad-

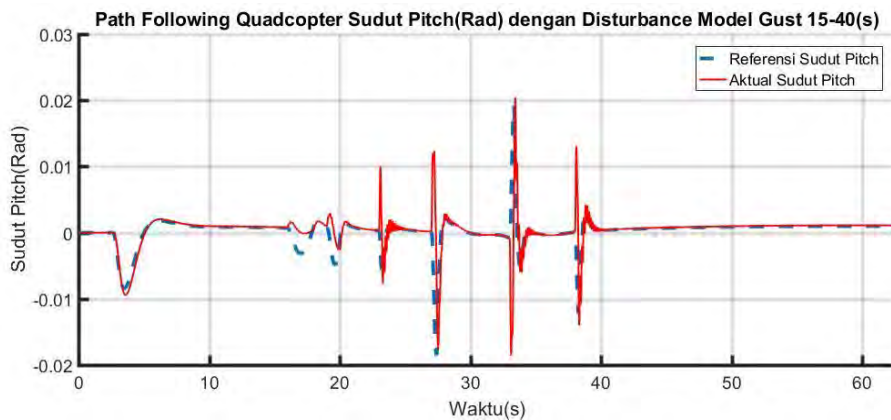
Tabel 4.2 Nilai *Integral Square Error* (ISE) tiap perubahan *magnitude gust*

<i>Integral Square Error</i> (ISE)	<i>Wind gust</i>		
	<i>Magnitude</i> (1)	<i>Magnitude</i> (1.5)	<i>Magnitude</i> (2)
<i>Roll</i>	0.0391rad	0.00587rad	0.00782rad
<i>Pitch</i>	0,01066rad	0.0016rad	0,00213rad
<i>Yaw</i>	0.10856rad	0.16284rad	0.21712rad
Posisi X	0.001451m	0.002176m	0.006528m
Posisi Y	0.000289m	0.000434m	0.000578m

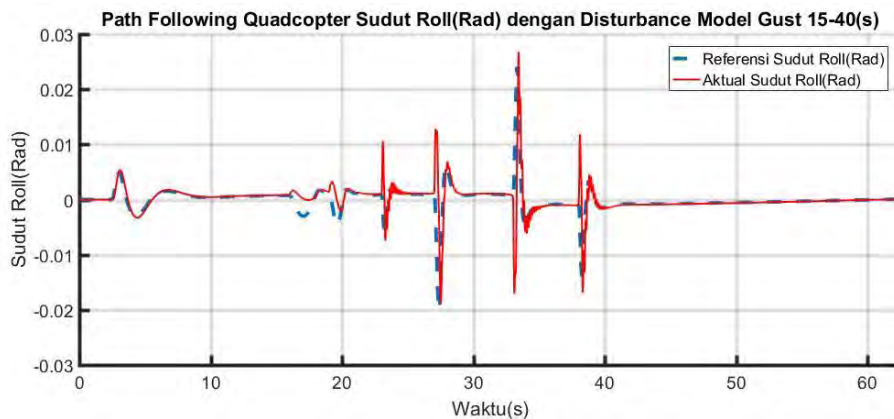
s/d 0.025 rad. Dengan rata-rata *error* sebesar 0.008 rad. Respon sudut  $\theta$  dapat mengikuti referensinya, namun pada saat terdapat *wind gust*, respon sudut  $\theta$  beresilasi dari -0.085 rad s/d 0.02 rad. Dengan rata-rata *error* sebesar 0,01066rad.

Hasil simulasi dengan sinyal *Disturbance* yang berbeda-beda telah disimpulkan pada Tabel 4.2. Sistem kontrol diuji dengan *Disturbance* dengan nilai yang berbeda-beda, yaitu D(1), D(1.5) dan D(2). Berdasarkan nilai ISE dari setiap

pergerakan quadcopter, diperoleh bahwa semakin besar nilai rata-rata *Disturbance* yang diberikan, maka semakin besar pula nilai ISE dari pergerakan quadcopter.



Gambar 4.21 *Path following* sudut *pitch* pada saat terdapat *wind gust*



Gambar 4.22 *Path following* sudut *roll* pada saat terdapat *gust*

Hal ini berpengaruh pula kepada besarnya nilai sinyal kontrol yang masuk ke *plant* quadcopter. Maksimal *Disturbance* pengukuran yang dapat diatasi oleh sistem kontrol adalah sebesar  $D(1)$ .

### C. Pengujian *Path following* Quadcopter dengan *Disturbance* Model Modifikasi Penggabungan *Step+Gust* 15-40(s)

Model kecepatan angin dengan memodifikasi Penggabungan *Step+Gust* didasarkan pada [11]. Besarnya kecepatan angin konstan *Step+Gust* pada saat  $t$  dengan fungsi parsial (4.7). Dimana nilai-nilai parameter yang digunakan adalah jumlah dari nilai parameter *step* dengan parameter *gust*. Bentuk model angin yang telah dimodifikasi ditunjukkan pada gambar 4.12c.

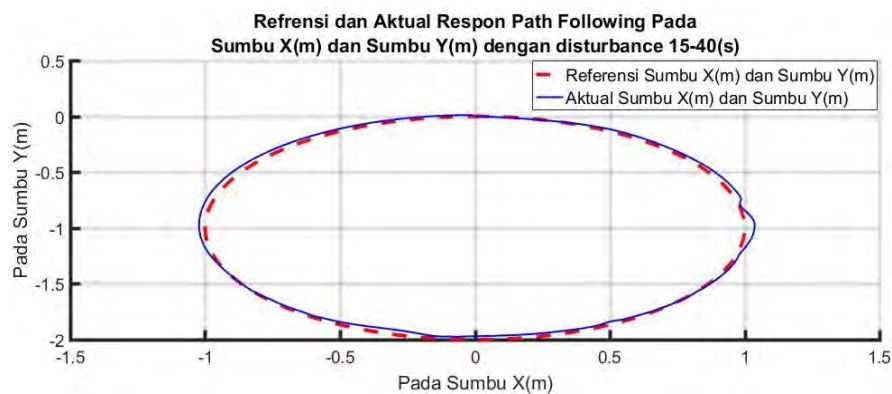


- **Pengujian Sistem Kontrol Untuk Gerak Translasi dengan *Disturbance* Model Modifikasi Penggabungan *Step+Gust* 15-40(s)**

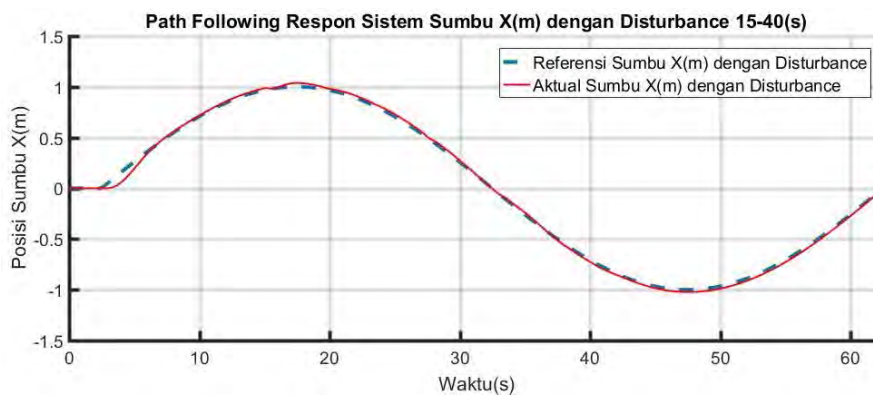
Gangguan yang paling sering terjadi pada saat quadcopter terbang adalah adanya hembusan angin. Pemberian efek *wind gust* yang telah dimodelkan dan diberikan pada dinamika quadcopter bertujuan untuk, mengetahui apakah quadcopter mampu tetap melakukan *path following* sesuai dengan referensi yang diberikan pada saat terjadi penyimpangan dari lintasan referensi.

Pada saat yang bersamaan gerakan *roll* dan *pitch* dari quadcopter akan menyesuaikan agar pengaruh *wind gust*. Besaran magnitude yang diberikan sebesar 1. Gambar 4.23 menunjukkan respon *path following* quadcopter pada sumbu  $X$  dan  $Y$ .

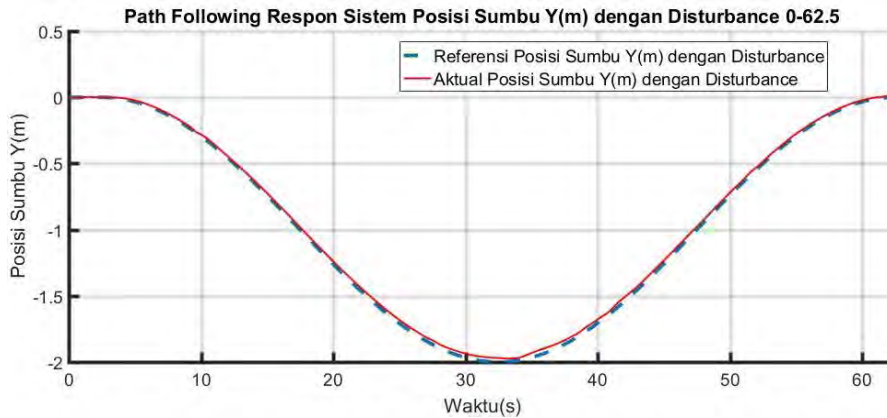
Pergerakan rotasi (*Outer loop*) yang menggunakan kontroler *command-generator tracker model following* dimana kontroler tersebut mampu mengkompensasi gangguan agar pada gerakan translasi tidak terjadi penyimpangan yang besar saat melakukan *path following*.



Gambar 4.23 *Path following* dengan gangguan dalam sumbu  $X, Y$



Gambar 4.24 *Path following* Sumbu  $X$  (m) pada saat terdapat *wind gust*



Gambar 4.25 *Path following* Sumbu  $Y$  (m) pada saat terdapat *wind gust*

Penyimpangan posisi pada sumbu  $X$  terjadi pada detik ke-15 dengan penyimpangan sebesar 0.013 m dan detik ke-40 dengan penyimpangan sebesar  $\pm 0.005$  m, seperti yang ditunjukkan pada Gambar 4.24. Pada respon posisi pada sumbu  $Y$  yang ditunjukkan pada Gambar 4.25, respon sistem detik ke-15 hingga detik ke-40 penyimpangan terjadi dengan nilai *overshoot* =  $\pm 0.002$  m. Pergerakan rotasi mengkompensasi gangguan agar pada gerakan translasi tidak terjadi penyimpangan yang besar. Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat *delay* sebesar  $\pm 3.17$  detik.

- **Pengujian Sistem Kontrol Untuk Gerak Rotasi dengan *Disturbance* Model Modifikasi Penggabungan *Step+Gust* 15-40(s)**

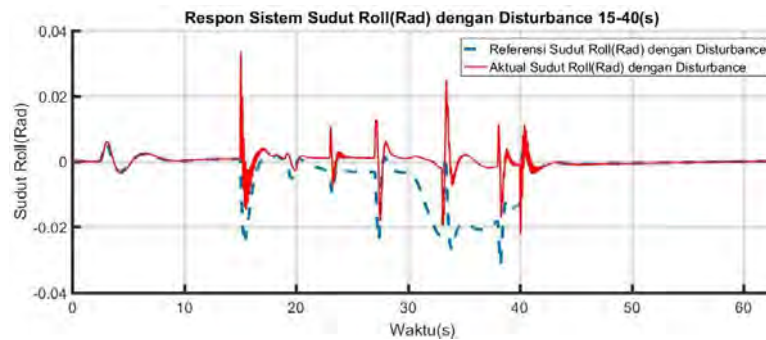
Gambar 4.26 dan 4.27 menunjukkan respon sudut  $\phi$  dan respon sudut  $\theta$  pada saat terdapat gangguan berupa *wind gust*. Ketika angin berhembus membuat quadcopter menyimpang dari posisi *path following*nya. Pengaruh *wind gust* terhadap sudut  $\phi$  terjadi dari detik ke-15 s/d detik ke-40, pergerakan dari -0.2 rad s/d 0.3 rad atau sekitar  $-11^\circ$  s/d  $18^\circ$ .

Respon sudut  $\theta$  dapat mengikuti referensinya, namun pada saat terdapat *wind gust*, respon sudut  $\theta$  beresilasi dari 0.5 rad s/d 0.2 rad atau sekitar  $-30^\circ$  s/d  $11^\circ$ . Dari hasil yang dilihat baik dari respon sistem pada sudut  $\phi$  dan  $\theta$  saat *wind gust* aktif kontroler mampu meminimalisir gangguan Hal tersebut terjadi untuk membuat quadcopter tetap stabil sehingga dapat kembali mengikuti *path*.

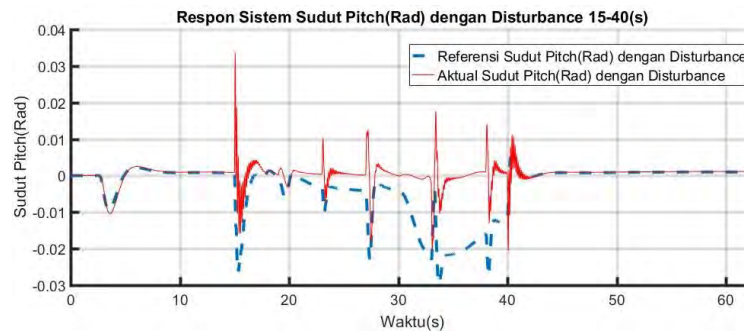
Algoritma *Line Of Sight* (LOS) yang ditambahkan pada sudut  $\psi$  untuk menghitung arah hadap dari quadcopter. Dapat dilihat bahwa respon sudut  $\psi$  dapat mengikuti sinyal referensi yang diberikan, dengan rata-rata *error* sebesar

0.09034rad. Dari Gambar 4.28 dilihat perbandingan antara referensi dan aktual dari sudut *yaw*, dimana aktual menjaga arah hadap (*heading*) dan memperhalus gerakan quadcopter. Visualisasi quadcopter pada saat *path following* dengan gangguan dapat dilihat pada Gambar 4.29. Visualisasi terjadi ketika sudah selesai melakukan *path following*, garis melingkar berwarna merah merupakan referensi dan garis berwarna biru merupakan sinyal aktual dari quadcopter.

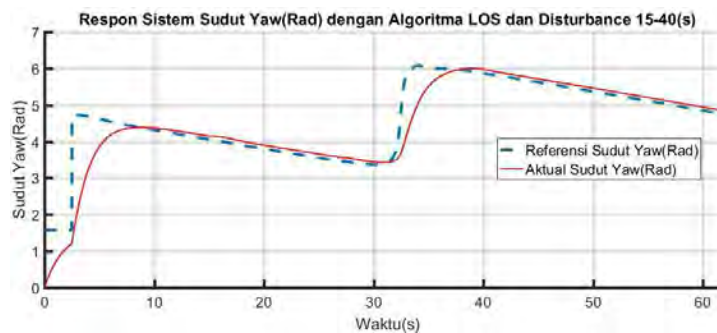
Hasil simulasi dengan sinyal *Disturbance* yang berbeda-beda telah disimpulkan pada Tabel 4.3. Sistem kontrol diuji dengan *Disturbance* dengan nilai yang berbeda-beda, yaitu D(1), D(1.5) dan D(2).



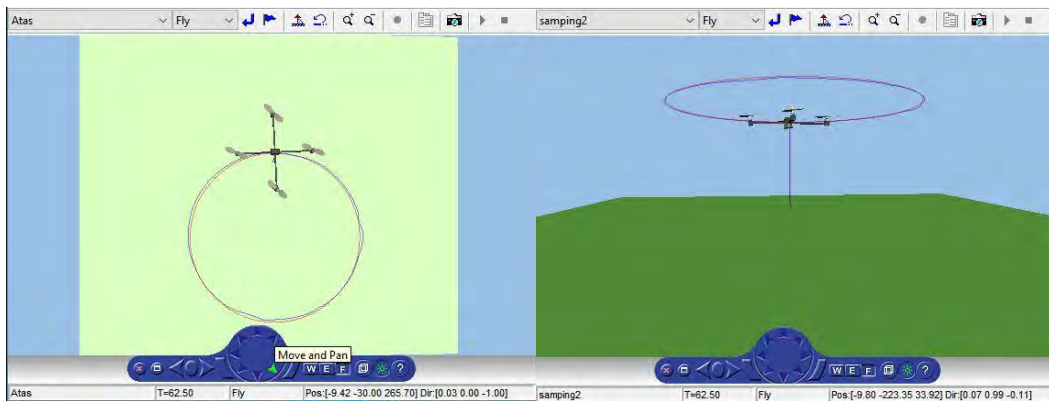
Gambar 4.26 *Path following* Sudut Roll pada saat terdapat *wind gust*



Gambar 4.27 *Path following* Sudut Pitch pada saat terdapat *wind gust*



Gambar 4.28 *Path following* Sudut Yaw pada saat terdapat *wind gust*



Gambar 4.29 Snipping Tool Virtual World saat simulasi dengan efek *wind gust* 15-40(s)

Tabel 4.3 Nilai *Integral Square Error* (ISE) tiap perubahan *magnitude step+gust* 15-40(s)

<i>Integral Square Error</i> (ISE)	<i>Wind gust</i>		
	<i>Magnitude</i> (1)	<i>Magnitude</i> (1.5)	<i>Magnitude</i> (2)
<i>Roll</i>	0.00207rad	0.00311rad	0.00414rad
<i>Pitch</i>	0,0021rad	0.00315rad	0,0042rad
<i>Yaw</i>	0.09034rad	0.13551rad	0.18068rad
Posisi X	0.01316m	0.01975m	0.02633m
Posisi Y	0.0003m	0.00045m	0.0006m

Berdasarkan nilai ISE dari setiap pergerakan quadcopter, diperoleh bahwa semakin besar nilai rata-rata *Disturbance* yang diberikan, maka semakin besar pula nilai ISE dari pergerakan quadcopter. Hal ini berpengaruh pula kepada besarnya nilai sinyal kontrol yang masuk ke *plant* quadcopter. Maksimal *Disturbance* pengukuran yang dapat diatasi oleh sistem kontrol adalah sebesar  $D(1)$ .

#### D. Pengujian *Path following* Quadcopter dengan *Disturbance Model* Modifikasi Penggabungan *Step+Gust* 0-62.5(s)

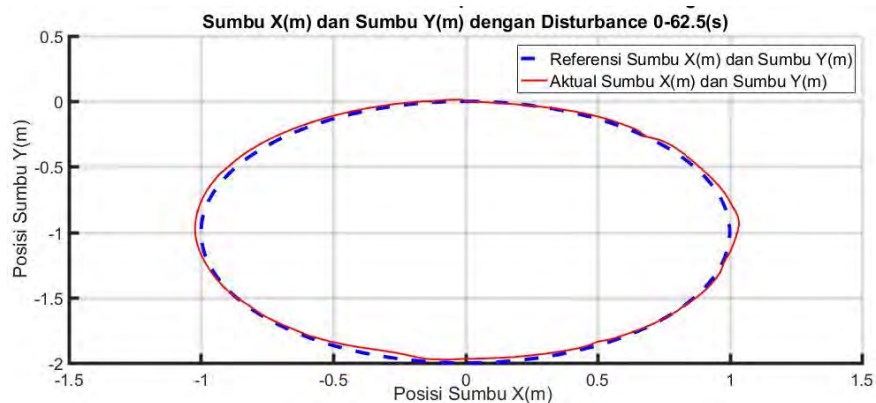
Model kecepatan angin memodifikasi Penggabungan *Step+Gust* didasarkan pada [11]. Besarnya kecepatan angin konstan *step+gust* pada saat  $t$  dengan fungsi parsial (4.7). Dimana nilai-nilai parameter yang digunakan adalah jumlah dari nilai parameter *step* dengan parameter *gust* yang diuji dari detik ke 0-62.5(s). Bentuk model angin yang telah dimodifikasi ditunjukkan pada gambar 4.12a. Besaran *magnitude* yang diberikan sebesar 1.

- Pengujian Sistem Kontrol Untuk Gerak Translasi dengan *Disturbance* Model Modifikasi Penggabungan *Step+Gust* 0-62.5(s)**

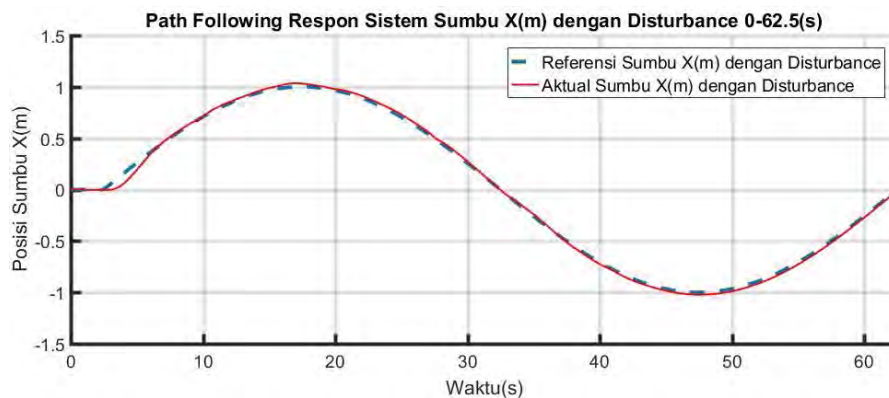
Keberadaan *wind gust* membuat quadcopter menyimpang dari lintasan referensi, Gambar 4.30 menunjukkan respon *path following* quadcopter pada sumbu  $X$  dan  $Y$  yang pada saat yang bersamaan gerakan *roll* dan *pitch* dari quadcopter menyesuaikan agar pengaruh *wind gust* membuat quadcopter dapat kembali melakukan *path following* sesuai dengan referensi yang diberikan.

Seperti yang ditunjukkan pada Gambar 4.31, penyimpangan posisi pada sumbu  $X$  terjadi pada detik ke-15 dengan penyimpangan sebesar 0.05 m dan detik ke-35 dengan penyimpangan sebesar  $\pm 0.005$  m. Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat  $\pm 3.17$  detik. Kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) terdapat *delay*  $\pm 3.17$  detik.

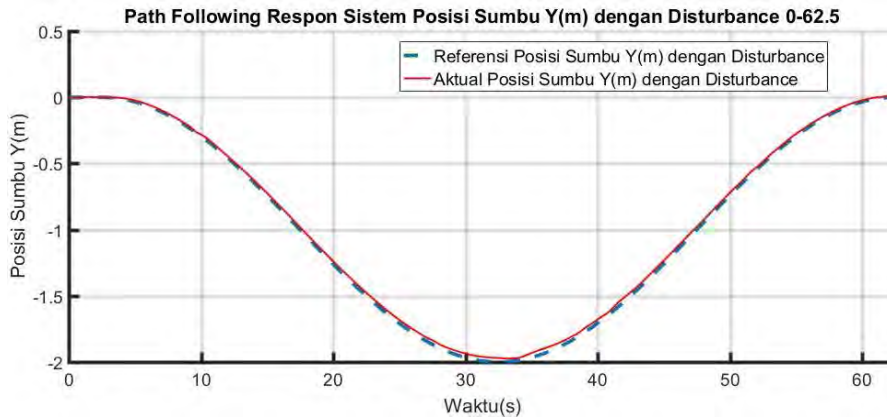
Pada respon posisi pada sumbu  $Y$  yang ditunjukkan pada Gambar 4.32, penyimpangan yang kecil terjadi pada detik ke-10 hingga detik ke-40 penyimpangan terjadi dengan nilai *overshoot* = 0.006 m dan *undershoot* = 0.001 m.



Gambar 4.30 *Path following* dengan gangguan dalam sumbu  $X, Y$



Gambar 4.31 *Path following* Sumbu  $X$  (m) pada saat terdapat *wind gust*



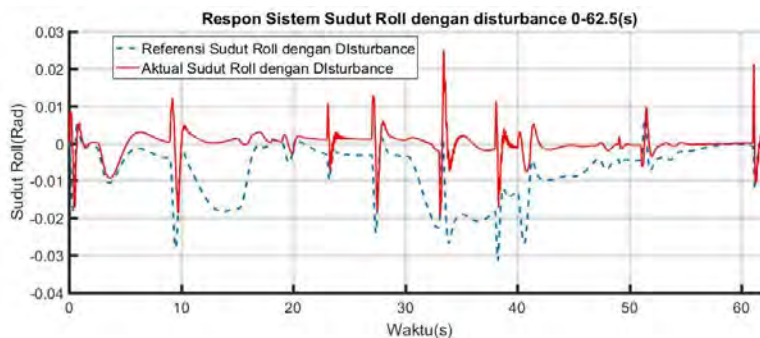
Gambar 4.32 *Path following* Sumbu  $Y$ (m) pada saat terdapat *wind gust*

- **Pengujian Sistem Kontrol Untuk Gerak Rotasi dengan *Disturbance* Model Modifikasi Penggabungan *Step+Gust* 0-62.5(s)**

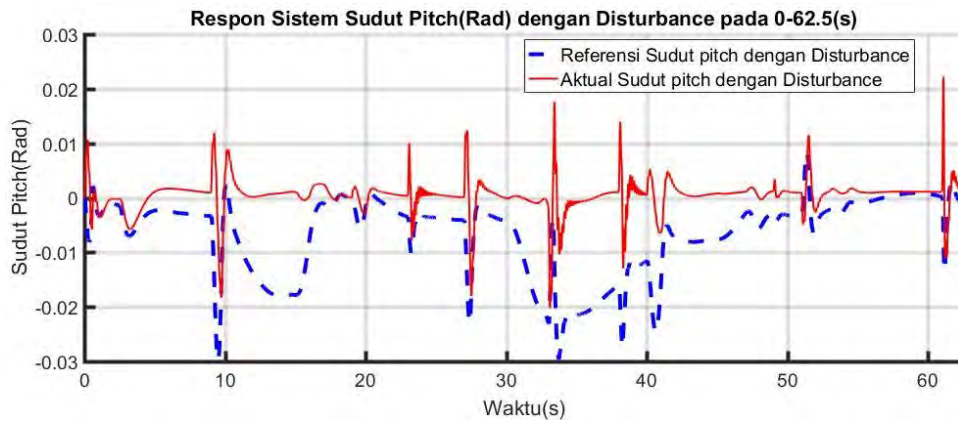
Pengaruh *wind gust* yang terjadi dari detik ke-15 s/d detik ke-40 terhadap sudut  $\phi$ , pergerakan dari 0.0059 rad s/d 0.0068rad. Respon sudut  $\theta$  dapat mengikuti referensinya, namun pada saat terdapat *wind gust*, respon sudut  $\theta$  beresilasi dari 0.0167 rad s/d 0.0160 rad. Gambar 4.33 dan 4.34 menunjukkan respon sudut  $\phi$  dan respon sudut  $\theta$  pada saat terdapat gangguan berupa *wind gust*. Ketika angin berhembus membuat quadcopter menyimpang dari posisi *path following*nya. Hal tersebut terjadi untuk membuat quadcopter tetap stabil sehingga dapat kembali mengikuti *path*.

Dapat dilihat bahwa respon sudut  $\psi$  dapat mengikuti sinyal referensi yang diberikan, dengan rata-rata *error* sebesar 0.0774rad. Dari Gambar 4.35 dilihat perbandingan antara referensi dan aktual dari sudut *yaw*, dimana aktual menjaga arah hadap (*heading*) dan memperhalus gerakan quadcopter.

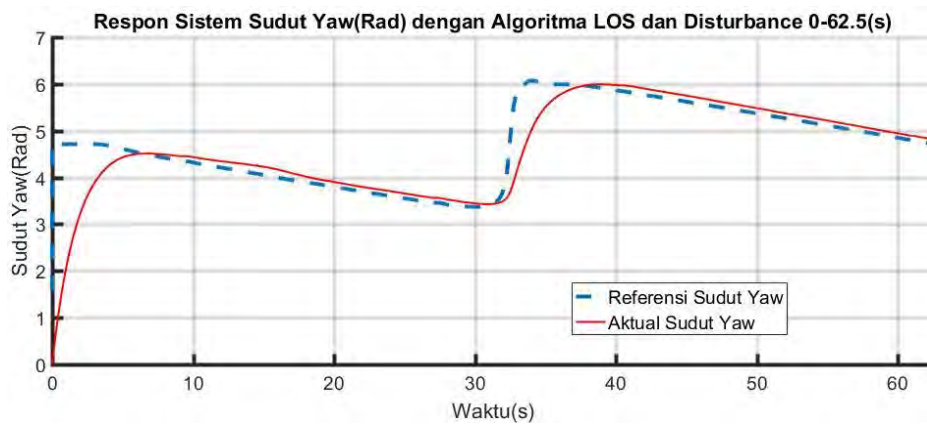
Visualisasi quadcopter pada saat melakukan *path following* dengan gangguan dapat dilihat pada Gambar 4.36 quadcopter tampak atas dan samping ketika sudah selesai melakukan *path following* dengan adanya *wind gust*.



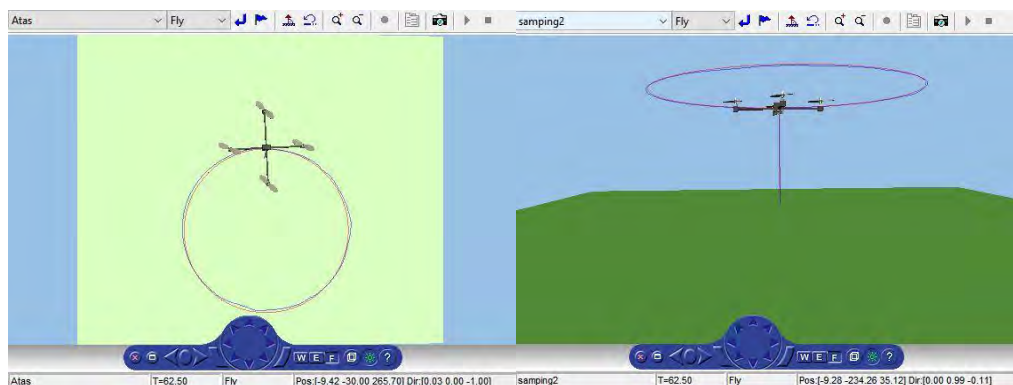
Gambar 4.33 *Path following* Sudut Roll pada saat terdapat *wind gust*



Gambar 4.34 *Path following* Sudut *Pitch* pada saat terdapat *wind gust*



Gambar 4.35 *Path following* Sudut *Yaw* pada saat terdapat *wind gust*



Gambar 4.36 *Snipping Tool Virtual World* saat simulasi dengan efek *wind gust* 0-62.5(s)

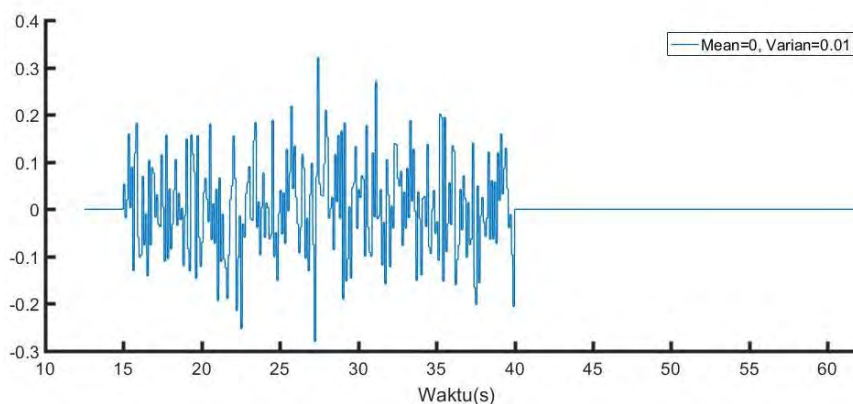
Tabel 4.4 Nilai *Integral Square Error* (ISE) tiap perubahan *magnitude step+gust* 0-62.5(s)

<i>Integral Square Error</i> (ISE)	<i>Wind gust</i>		
	<i>Magnitude</i> (1)	<i>Magnitude</i> (1.5)	<i>Magnitude</i> (2)
<i>Roll</i>	0.00007rad	0.000105rad	0.00014rad
<i>Pitch</i>	0,0000706rad	0.000106rad	0,00021rad
<i>Yaw</i>	0.0774rad	0.11623rad	0.1549rad
Posisi X	0.000014m	0.0000212m	0.000028m
Posisi Y	0.000282m	0.0004247m	0.000565m

Hasil simulasi dengan sinyal *Disturbance* yang berbeda-beda telah disimpulkan pada Tabel 4.4. Sistem kontrol diuji dengan *Disturbance* dengan nilai yang berbeda-beda, yaitu D(1), D(1.5) dan D(2). Berdasarkan nilai ISE dari setiap pergerakan quadcopter, diperoleh bahwa semakin besar nilai rata-rata *Disturbance* yang diberikan, maka semakin besar pula nilai ISE dari pergerakan quadcopter. Hal ini berpengaruh pula kepada besarnya nilai sinyal kontrol yang masuk ke *plant* quadcopter. Maksimal *Disturbance* pengukuran yang dapat diatasi oleh sistem kontrol adalah sebesar D(1).

### 4.3 Pengujian Sistem Kontrol Terhadap *Noise* Pengukuran

Dalam mengatasi *noise* pengukuran, maka dilakukan pengukuran dari sistem untuk menguji kemampuan sistem dalam mengatasi *noise*. Yang mana *noise* diberikan pada *state* pengukuran dari sistem. Pemberian *noise* pada *state* pengukuran menggunakan sinyal random Gaussian. Sinyal random gaussian diberikan nilai *mean* sebesar 0 dan *varian* sebesar 0.01 atau  $N(0,0.01)$ . *Noise* tersebut diberikan pada detik 14-40s seperti yang ditunjukkan pada Gambar 4.37.

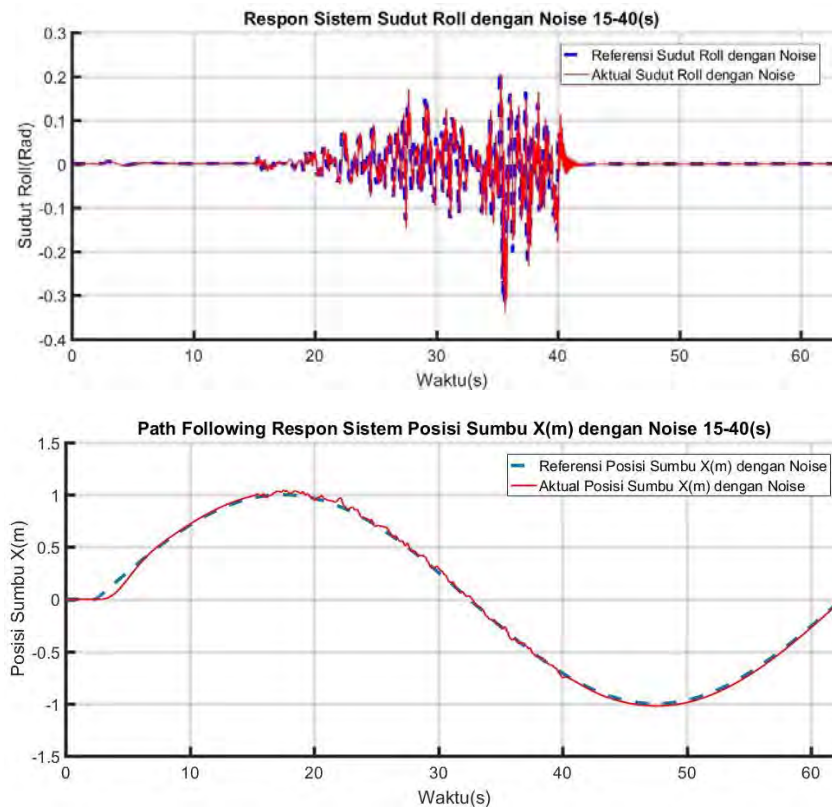


Gambar 4.37 Sinyal random Gaussian dengan  $N(0,0.01)$



## A. Pengujian Sistem Kontrol Untuk Sumbu $X$

Gambar 4.38 menunjukkan perilaku sudut *pitch* dan posisi pada sumbu  $X$  pada saat diberi *noise*. Untuk mengatasi sinyal random yang diberikan, sudut *roll* beresilasi dengan nilai *overshoot/undershoot* sebesar  $\pm 0.2$  rad atau  $12^\circ$ .



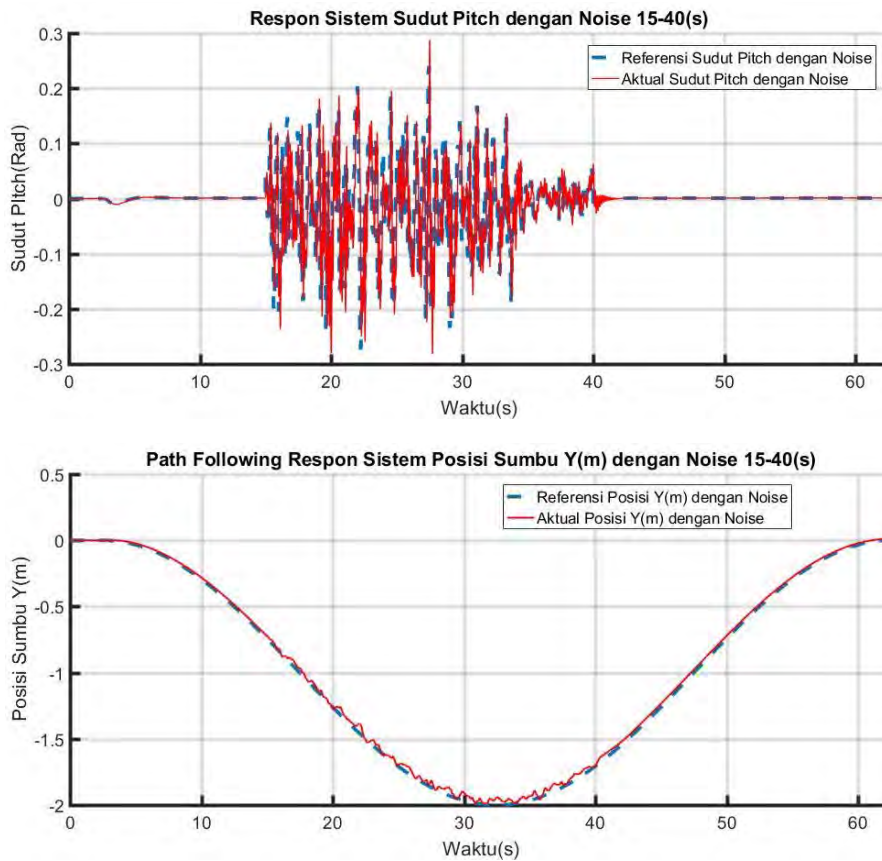
Gambar 4.38 Posisi  $X$  dan sudut *roll* saat diberi *noise*  $N(0,0.01)$

Pengaruh *noise* terhadap *path following* posisi  $X$ , terjadi ada detik ke-15 hingga detik ke-40 respon sistem tidak smooth melainkan terjadi riak kecil selama rentang waktu tersebut.

## B. Pengujian Sistem Kontrol Untuk Sumbu $Y$

Sama halnya dengan *path following* pada posisi  $X$ , pergerakan pada sumbu  $Y$  mengalami penyimpangan selama 25 detik. Gambar 4.39 menunjukkan sudut *roll* dan posisi  $Y$  saat diberi *noise*. Nilai ISE pada pergerakan ini adalah 0.03552 m. Respon sudut *pitch* beresilasi dengan nilai *overshoot/undershoot* sebesar  $\pm 0.25$  rad atau  $15^\circ$ .

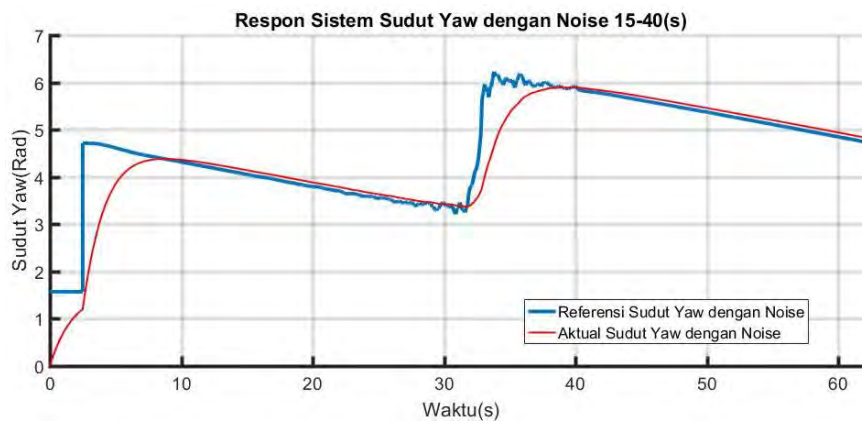
Pada saat *noise* sudah tidak diberikan pada sistem quadcopter dapat kembali melakukan *path following* dengan stabil seperti yang dapat dilihat pada detik ke-43 hingga detik ke-62.5.



Gambar 4.39 Posisi  $Y$  dan sudut *roll* saat diberi *noise*  $N(0,0.01)$

### C. Pengujian Sistem Kontrol Untuk Sudut *Yaw*

Sama halnya dengan *path following* pada posisi  $X$  dan  $Y$ , pergerakan pada sudut *yaw* tidak mengalami penyimpangan. Pada detik ke-15 hingga ke-31.9 *noise* yang terjadi di minimalisir sehingga respon sistem tahan terhadap *noise*.



Gambar 4.40 Sudut *yaw* saat diberi *noise*  $N(0,0.01)$

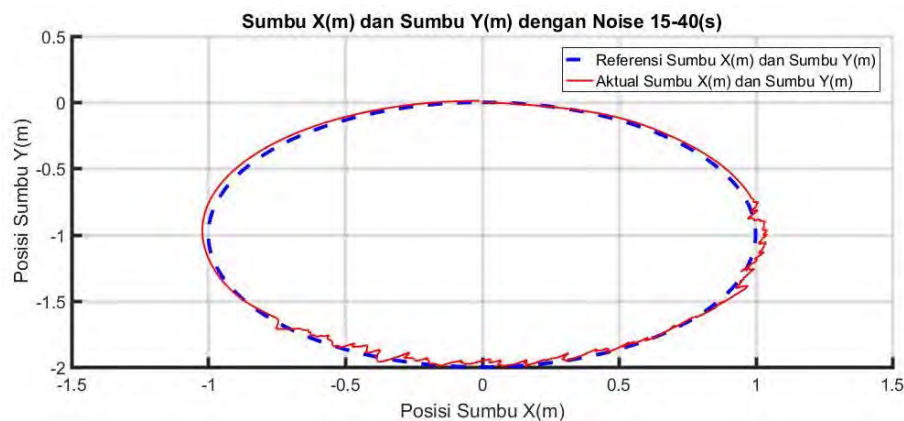
Noise yang terjadi pada detik ke-32 hingga detik ke-38 dilewati oleh aktual dari respon sistem *yaw*. Dengan menambahkan algoritma LOS dinamika quadcopter bergerak mengikuti referensi dan sudah memperkirakan titik berikutnya sehingga berbelok lebih awal dikarenakan titik ke-2 hingga ke-4 dan titik ke-4 hingga ke-6 mempunyai sudut  $90^\circ$ . Jika algoritma LOS bertemu dengan sudut  $90^\circ$  maka  $\tan(a_k) = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$  bernilai tak terhingga. Agar tidak bernilai tak

terhingga maka dilakukan pengurangan sebanyak 6m  $\left( \tan(a_k) = \frac{(y_{k+1} - y_k) - 6}{(x_{k+1} - x_k) - 6} \right)$  sehingga, pada saat kurang dari 6m maka dinamika sudah berbelok menuju titik berikutnya. Gambar 4.40 menunjukkan sudut *yaw* saat diberi *noise*.

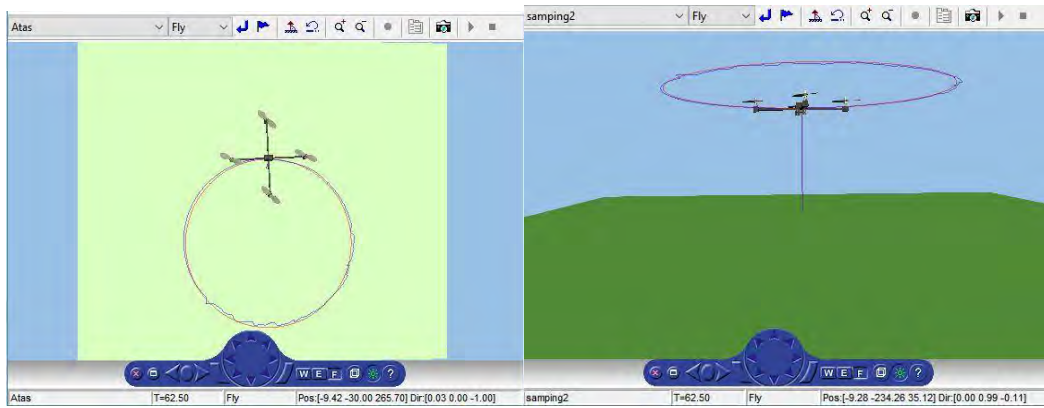
#### D. Pengujian Sistem Kontrol dalam bidang XY

Respon *path following* pada saat diberi *noise* jika dilihat dalam bidang  $X, Y$  ditunjukkan oleh Gambar 4.41 sedangkan virtual world dari quadcopter terdapat pada Gambar 4.42. Saat diberi *noise*, pergerakan quadcopter menjadi tidak smooth dimana untuk mengatasi *noise*, pergerakan rotasi quadcopter menjadi berosilasi, namun hal tersebut membuat penyimpangan pada posisi quadcopter menjadi kecil.

Hasil simulasi dengan sinyal *noise* yang berbeda-beda telah disimpulkan pada Tabel 4.5. Sistem kontrol diuji dengan *noise* dengan nilai yang berbeda-beda, yaitu  $N(0,0.001)$ ,  $N(0,0.01)$  dan  $N(0,0.05)$ . Berdasarkan nilai ISE dari setiap pergerakan quadcopter, diperoleh bahwa semakin besar nilai rata-rata amplitude dari *noise* yang diberikan, maka semakin besar pula nilai ISE dari pergerakan quadcopter. Hal ini berpengaruh pula kepada besarnya nilai sinyal kontrol yang masuk ke *plant* quadcopter. Maksimal *noise* pengukuran yang dapat diatasi oleh sistem kontrol adalah sebesar  $N(0, 0.05)$  dengan standar deviasi 0.22.



Gambar 4.41 Respon *path following* lintasan lingkaran pada bidang  $X, Y$



Gambar 4.42 *Virtual World* dari quadcopter saat simulasi diberi efek *noise*

Tabel 4.5 Nilai *Integral Square Error* (ISE) pada perubahan *Variance Noise*

<i>Integral Square Error</i> (ISE)	<i>Noise</i>		
	<i>Mean=0, Var= 0.001</i>	<i>Mean=0, Var= 0.01</i>	<i>Mean=0, Var= 0.05</i>
<i>Roll</i>	0.014rad	0.038rad	0.048rad
<i>Pitch</i>	0,011rad	0,036rad	0,043rad
<i>Yaw</i>	0.0122rad	0.0145rad	0.0145rad
Posisi X	0.001m	0.005m	0.009m
Posisi Y	0.0001m	0.002m	0.007m

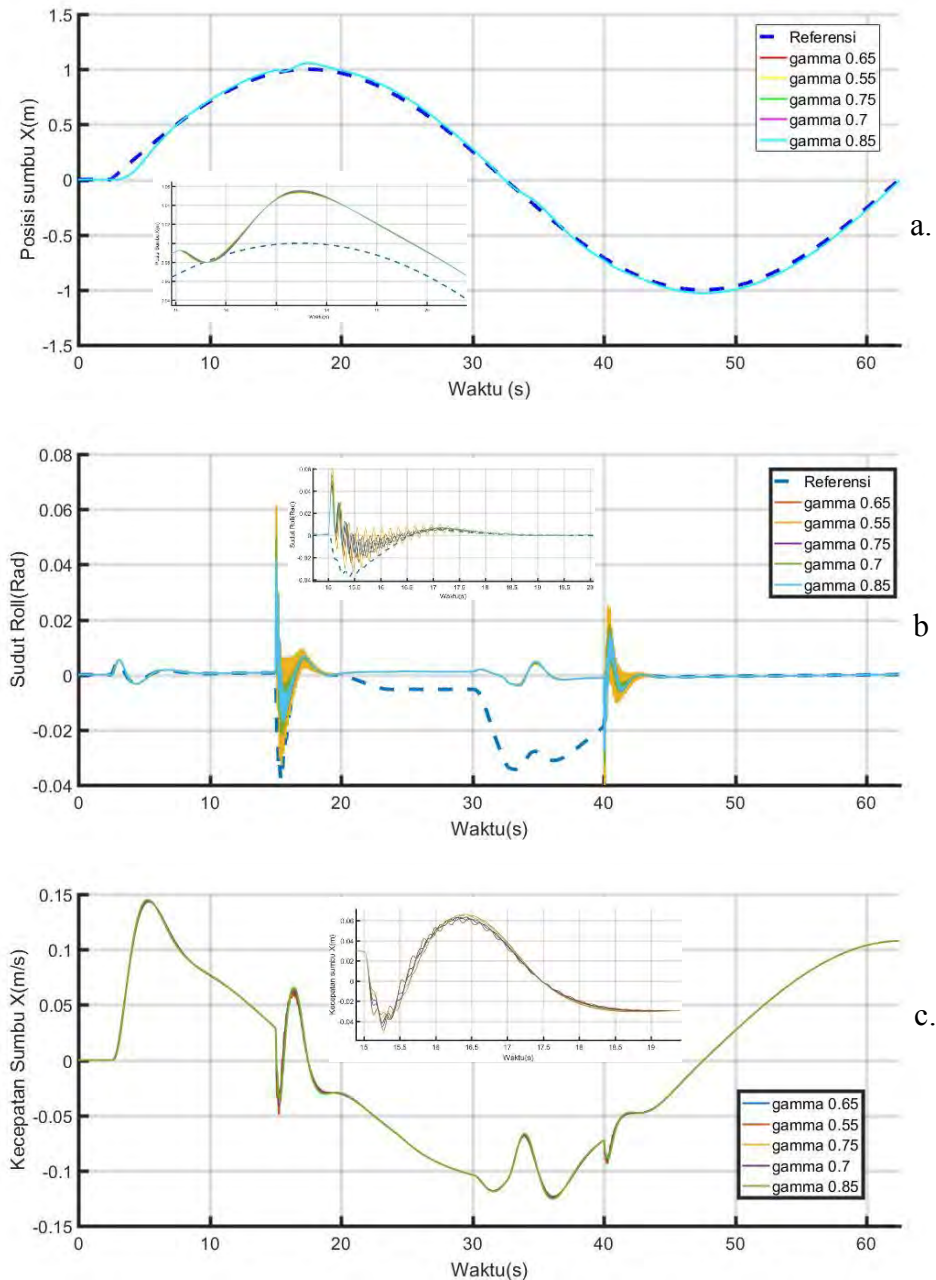
#### 4.4 Pengujian Pelemahan Terhadap Gangguan (*Disturbance*)

Untuk mengetahui sejauh mana pengaruh tingkat level pelemahan terhadap gangguan maka, dilakukan simulasi dengan nilai  $\gamma$  yang berbeda-beda ( $\gamma = 0.55, 0.65, 0.7, 0.75, \text{ dan } 0.85$ ). Hasil simulasi disajikan dalam bentuk analisa posisi, rotasi dan kecepatan quadcopter dengan berbagai nilai  $\gamma$  pada saat terdapat gangguan. Gangguan yang diujikan pada pengujian ini antara lain angin konstan (*step*) 15-40(s), *gust* 15-40(s), Model Modifikasi Penggabungan *Step+Gust* 15-40(s) dan Model Modifikasi Penggabungan *Step+Gust* 0-62.5(s)

##### A. Pengujian dengan Perubahan nilai $\gamma$ *Disturbance* Model Angin Konstan 15-40(s)

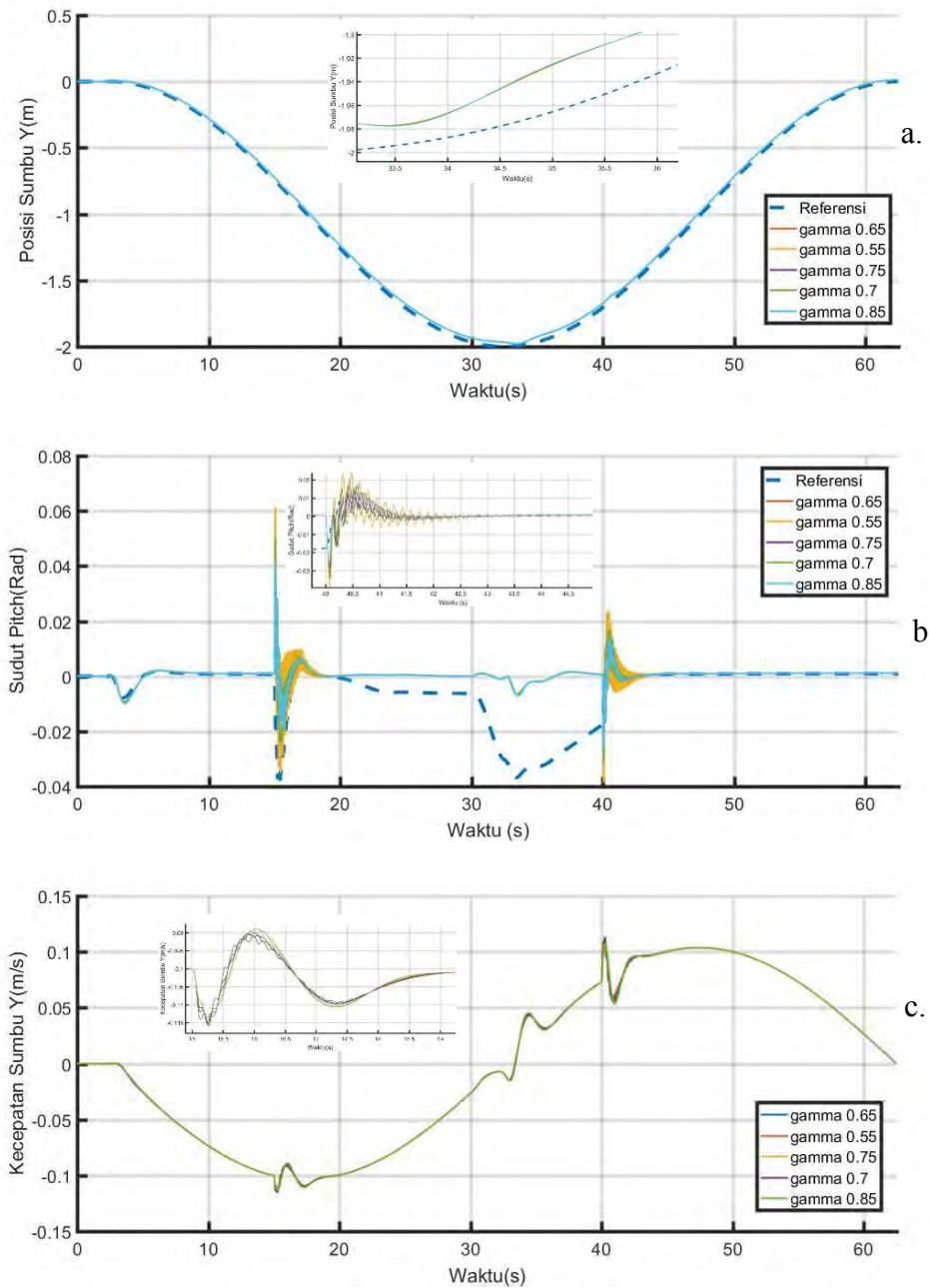
Respon posisi, sudut *roll* dan kecepatan quadcopter pada posisi X pada saat diberi gangguan *wind gust*, dapat dilihat pada Gambar 4.43. Nilai  $\gamma$  yang berbeda-beda berpengaruh terhadap seberapa besar maksimum *overshoot* pada setiap penyimpangan *path following* yang terjadi.

Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata untuk semua nilai  $\gamma$  adalah 0.00441. Pada saat  $\gamma = 0.7$  maksimum *overshoot* adalah 0.00175 m dan *undershoot* 0.0028 m, sedangkan saat  $\gamma = 0.65$  maksimum *overshoot* 0.0017 m dan *undershoot* 0.00275 m. Pada saat  $\gamma = 0.55$  *overshoot* dan *undershoot* mencapai 0.00195 m dan 0.0038 m. saat  $\gamma = 0.75$  *overshoot* dan *undershoot* mencapai 0.00295 m dan 0.0028 m. Respon dengan penyimpangan terbesar terjadi ketika nilai  $\gamma = 0.85$ , dengan maksimum *overshoot* 0.003 m dan *undershoot* 0.0045 m.



Gambar 4.43a. Respon posisi  $X$ , b. Respon sudut *roll*, dan c. Respon kecepatan sumbu  $X$  dengan berbagai nilai  $\gamma$  saat *step*

Gambar 4.43a,b dan c menunjukkan posisi, sudut *roll* dan kecepatan quadcopter pada posisi X. Pada saat  $\gamma = 0.55$  kecepatan berkisar antara  $-0.54$  m/s hingga  $0.64$  m/s. Pada  $\gamma = 0.85$  maksimum *overshoot* dan *undershoot* adalah  $0.75$  m/s dan  $-8.2$  m/s.



Gambar 4.44 Respon posisi  $Y$ , b. Respon sudut *Roll*, dan c. Respon kecepatan sumbu  $Y$  dengan berbagai nilai  $\gamma$  saat *step*

Gambar 4.44a, b dan c adalah respon posisi, sidut *pitch* dan kecepatan quadcopter pada posisi Y dengan berbagai nilai  $\gamma$ . Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata 0.0212 m. Untuk  $\gamma = 0.55$ , maksimum *overshoot* dan *undershoot* bernilai 0.093 m dan 0.13 m sedangkan  $\gamma = 0.65$  *undershoot* = 0.012 dan *overshoot* = 0.01m.

Untuk  $\gamma = 0.7$  maksimum *overshoot* = 0.0232m dan *undershoot* = 0.032 m.  $\gamma = 0.75$  maksimum *overshoot* = 0.00332m dan *undershoot* = 0.0132m. Serta  $\gamma = 0.85$  maksimum *overshoot* = 0.0542m dan *undershoot* = 0.232m. Kecepatan pada sumbu Y ditunjukkan oleh Gambar 4.44c. Kecepatan maksimum untuk  $\gamma = 0.85$  adalah  $\pm 0.6$  m/s, untuk  $\gamma = 0.75$  adalah  $\pm 0.3$  m/s, untuk  $\gamma = 0.7$  adalah  $\pm 0.26$  m/s, untuk  $\gamma = 0.65$  adalah  $\pm 0.22$  m/s dan untuk  $\gamma = 0.55$  adalah  $\pm 0.16$  m/s.

Nilai  $\gamma$  yang berbeda-beda, akan mempengaruhi besar nilai matriks P dan gain K. Oleh karena itu, respon yang dihasilkan akan berbeda untuk setiap nilai  $\gamma$  yang diberikan. Berdasarkan seluruh respon sistem saat simulasi dengan efek gangguan, menunjukkan bahwa perbedaan nilai  $\gamma$  akan mempengaruhi penyimpangan posisi serta rotasi quadcopter. Besarnya penyimpangan posisi dan rotasi yang terjadi pada quadcopter pada nilai  $\gamma$  yang berbeda-beda dapat dilihat pada Tabel. 4.6 dan Tabel 4.7.

Penyimpangan terkecil terjadi ketika  $\gamma = 0.7$ , jika dilihat pada Tabel 4.8, bahwa nilai  $\gamma^*$  mendekati  $\gamma$  yaitu 0.69. Semakin nilai  $\gamma^*$  mendekati  $\gamma$  ( $\gamma^* = \gamma$ ), maka pada kondisi ini menghasilkan ketahanan sistem terhadap gangguan dengan performa yang baik.

Tabel 4.6 Penyimpangan posisi quadcopter pada sumbu  $X$  dan  $Y$  saat *step*

$\gamma$	Penyimpangan	Penyimpangan	ISE	ISE
	Posisi $X$	Posisi $Y$	Posisi $X$	Posisi $Y$
0.55	0.004416	0.0209105	0.001950	0.00043725
0.65	0.004426	0.0212547	0.001958	0.00045176
0.7	0.004437	0.0212548	0.001969	0.00045177
0.75	0.004439	0.0212552	0.001971	0.00045178
0.85	0.004444	0.9469284	0.001974	0.89667339

Tabel 4.7 Penyimpangan posisi quadcopter pada sudut *roll* dan *pitch* saat *step*

$\gamma$	Penyimpangan	Penyimpangan	ISE	ISE
	Sudut <i>roll</i>	Sudut <i>pitch</i>	Sudut <i>roll</i>	Sudut <i>pitch</i>
0.55	0.0053428	0.005392	0.002848	0.00290042
0.65	0.0053434	0.005393	0.002854	0.00290507
0.7	0.0053470	0.005384	0.002855	0.00290822
0.75	0.0053371	0.005385	0.002859	0.00290858
0.85	0.0053484	0.005389	0.002860	0.00289937

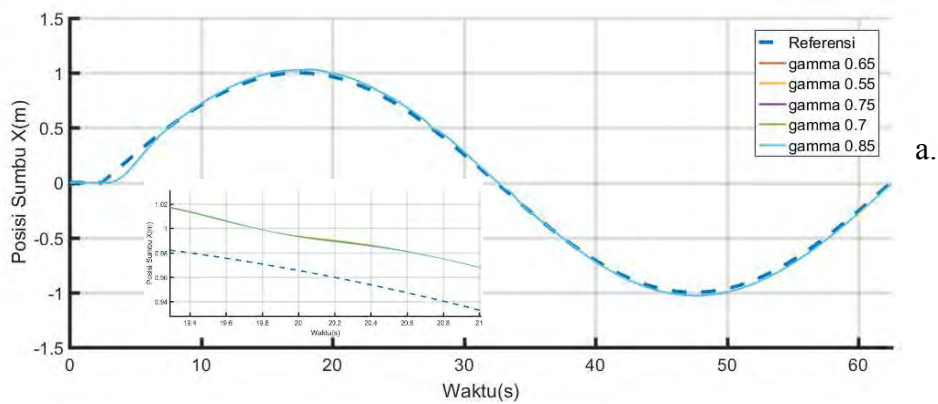
Tabel 4.8 Nilai gain  $K$  pada *inner loop* dan norm- $\infty$  sistem untuk berbagai nilai  $\gamma$

$\gamma$	$K$	$\ T_{zw}\ _{\infty} = \gamma^*$
0.55	[0.5161 0.0590]	0.54
0.65	[0.6100 0.0698]	0.64
0.7	[0.6570 0.0752]	0.69
0.75	[0.7036 0.0805]	0.73
0.85	[0.8255 0.0912]	0.82

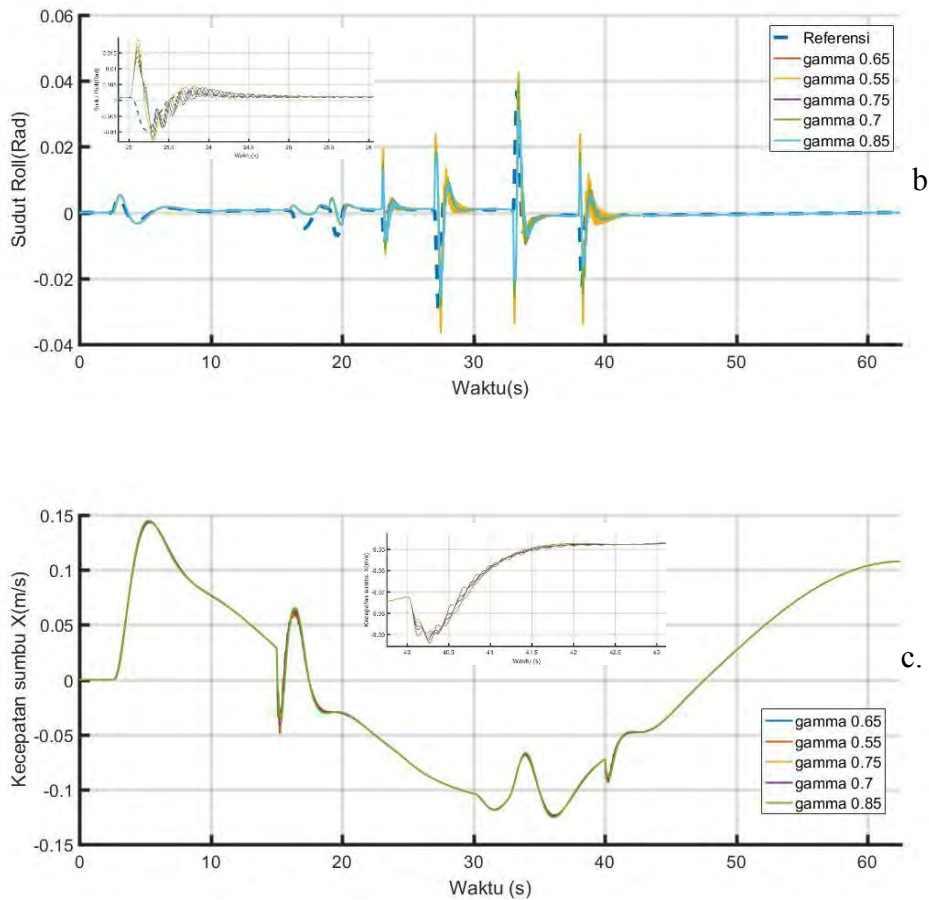
### B. Pengujian dengan Perubahan nilai $\gamma$ *Disturbance Model Gust 15-40(s)*

Selanjutnya dilakukan simulasi dengan nilai  $\gamma$  yang berbeda-beda ( $\gamma = 0.55, 0.65, 0.7, 0.75$  dan  $0.85$ ) untuk mengetahui sejauh mana pengaruh tingkat level pelemahan terhadap gangguan. Hasil simulasi disajikan dalam bentuk analisa posisi dan kecepatan quadcopter dengan berbagai nilai  $\gamma$  pada saat terdapat gangguan. Seperti yang terdapat pada Gambar 4.45 – 4.46.

Respon posisi, sudut *roll* dan kecepatan posisi quadcopter pada sumbu X pada saat diberi gangguan *wind gust*, dapat dilihat pada Gambar 4.45. Nilai  $\gamma$  yang berbeda-beda berpengaruh terhadap seberapa besar maksimum *overshoot* pada setiap penyimpangan *path following* yang terjadi. Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata untuk semua nilai  $\gamma$  adalah 0.00456. Pada saat  $\gamma = 0.7$  maksimum *overshoot* adalah 0.00165 m dan *undershoot* 0.0128 m, sedangkan saat  $\gamma = 0.65$  maksimum *overshoot* 0.0327 m dan *undershoot* 0.00175 m. Pada saat  $\gamma = 0.55$  *overshoot* dan *undershoot* mencapai 0.00195 m dan 0.0038 m. Saat  $\gamma = 0.75$  *overshoot* dan *undershoot* mencapai 0.00295 m dan 0.0028 m. Respon dengan penyimpangan terbesar terjadi ketika nilai  $\gamma = 0.85$ , dengan maksimum *overshoot* 0.003 m dan *undershoot* 0.0045 m.



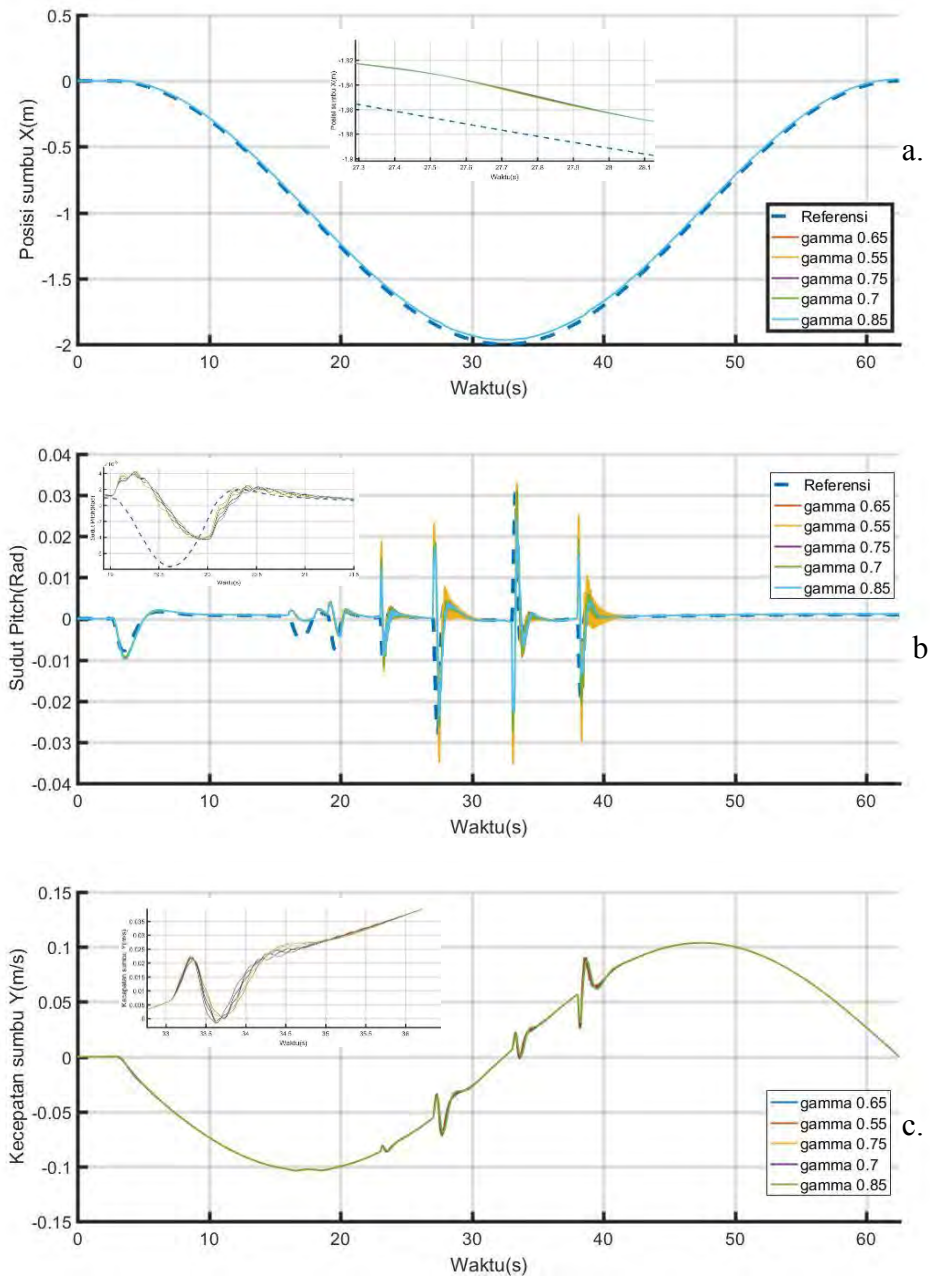




Gambar 4.45 Respon posisi  $X$ , b. Respon sudut *roll*, dan c. Respon kecepatan sumbu  $X$  dengan berbagai nilai  $\gamma$

Gambar 4.45a,b dan c menunjukkan posisi, sudut *roll* dan kecepatan quadcopter pada posisi  $X$ . Pada saat  $\gamma = 0.55$  kecepatan berkisar antara  $-0.44$  m/s hingga  $0.64$  m/s. Pada  $\gamma = 0.85$  maksimum *overshoot* dan *undershoot* adalah  $0.45$  m/s dan  $-3.2$  m/s.

Gambar 4.46a, b, dan c adalah respon posisi, sudut *pitch* dan kecepatan quadcopter pada posisi  $Y$  dengan berbagai nilai  $\gamma$ . Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata  $0.0208$  m. Untuk  $\gamma = 0.65$ , maksimum *overshoot* dan *undershoot* bernilai  $0.09$  m dan  $0.13$  m. Untuk  $\gamma = 0.7$  maksimum *overshoot* =  $0.0232$ m dan *undershoot* =  $0.042$  m.  $\gamma = 0.75$  maksimum *overshoot* =  $0.0032$ m dan *undershoot* =  $0.013$ m. Serta  $\gamma = 0.85$  maksimum *overshoot* =  $0.052$ m dan *undershoot* =  $0.32$ m. Kecepatan pada sumbu  $Y$  ditunjukkan oleh Gambar 4.46c. Kecepatan maksimum untuk  $\gamma = 0.85$  adalah  $\pm 0.9$  m/s, untuk  $\gamma = 0.75$  adalah  $\pm 0.86$  m/s, untuk  $\gamma = 0.7$  adalah  $\pm 0.82$  m/s, untuk  $\gamma = 0.65$  adalah  $\pm 0.78$  m/s dan untuk  $\gamma = 0.55$  adalah  $\pm 0.72$  m/s.



Gambar 4.46 Respon posisi  $Y$ , b. Respon sudut  $Roll$ , dan c. Respon kecepatan sumbu  $Y$  dengan berbagai nilai  $\gamma$

Nilai  $\gamma$  yang berbeda-beda, akan mempengaruhi besar nilai matriks  $P$  dan gain  $K$ . Oleh karena itu, respon yang dihasilkan akan berbeda untuk setiap nilai  $\gamma$  yang diberikan. Berdasarkan seluruh respon sistem saat simulasi dengan efek gangguan, menunjukkan bahwa perbedaan nilai  $\gamma$  akan mempengaruhi penyimpangan posisi quadcopter. Besarnya penyimpangan posisi yang terjadi pada quadcopter pada nilai  $\gamma$  yang berbeda-beda dapat dilihat pada Tabel. 4.9 dan 4.10.

Penyimpangan terkecil terjadi ketika  $\gamma = 0.7$ , jika dilihat pada Tabel 4.11, bahwa nilai  $\gamma^*$  mendekati  $\gamma$  yaitu 0.67. Semakin nilai  $\gamma^*$  mendekati  $\gamma$  ( $\gamma^* = \gamma$ ), maka pada kondisi ini menghasilkan ketahanan sistem terhadap gangguan dengan performa yang baik.

Tabel 4.9 Penyimpangan posisi quadcopter pada sumbu  $X$  dan  $Y$  saat *gust*

$\gamma$	Penyimpangan Posisi $X$	Penyimpangan Posisi $Y$	ISE Posisi $X$	ISE Posisi $Y$
0.55	0.0046652	0.02084038	0.0021764	0.000434322
0.65	0.0046654	0.02084113	0.0021766	0.000434353
0.7	0.0046656	0.02084139	0.0021767	0.000434364
0.75	0.0046658	0.02084142	0.0021770	0.000434365
0.85	0.0046661	0.02084144	0.0021773	0.000434366

Tabel 4.10 Penyimpangan posisi quadcopter pada sudut *roll* dan *pitch* saat *gust*

$\gamma$	Penyimpangan Sudut <i>roll</i>	Penyimpangan Sudut <i>pitch</i>	ISE Sudut <i>roll</i>	ISE Sudut <i>pitch</i>
0.55	0.00094	0.00011	0.00325	0.0002
0.65	0.00570	0.00012	0.00408	0.0012
0.7	0.00639	0.00015	0.00587	0.0016
0.75	0.00766	0.00061	0.00645	0.0037
0.85	0.00834	0.00079	0.00889	0.0063

Tabel 4.11 Gain  $K$  pada *inner loop* dan norm- $\infty$  sistem untuk berbagai nilai  $\gamma$

$\gamma$	$K$	$\ T_{zw}\ _{\infty} = \gamma^*$
0.55	[0.5776 0.0641]	0.52
0.65	[0.6821 0.0758]	0.62
0.7	[0.7346 0.0817]	0.67
0.75	[0.7870 0.0875]	0.71
0.85	[0.8919 0.0991]	0.804

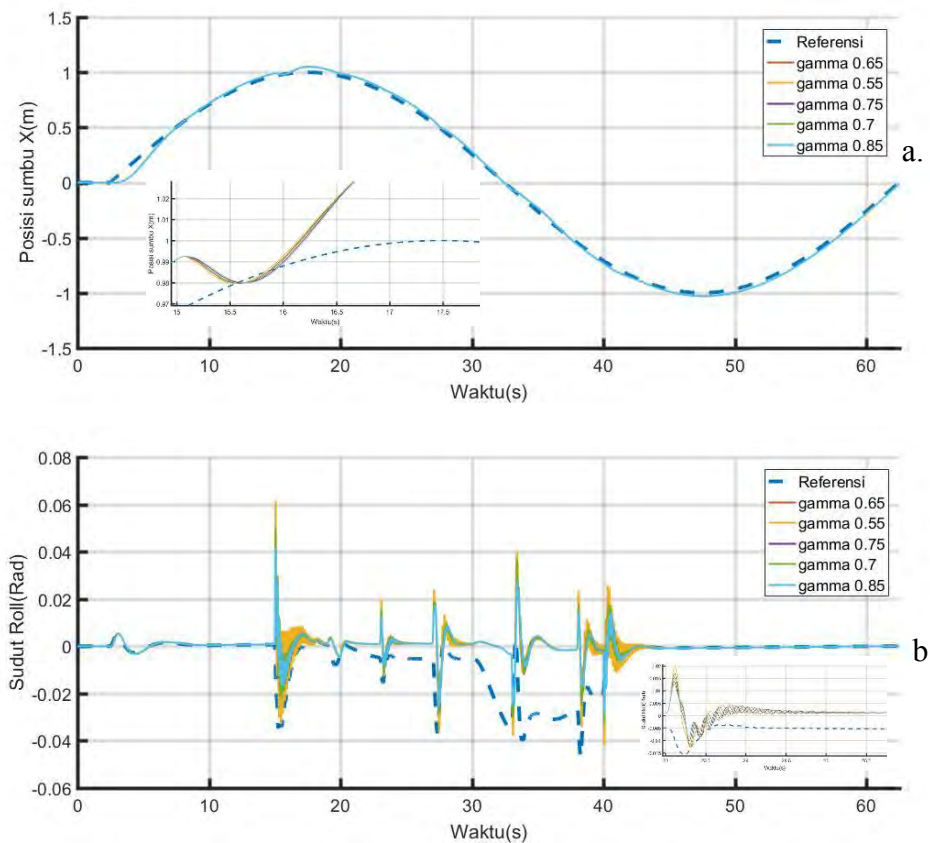
### C. Pengujian dengan Perubahan nilai $\gamma$ *Disturbance Model Modifikasi Penggabungan Step+Gust 15-40(s)*

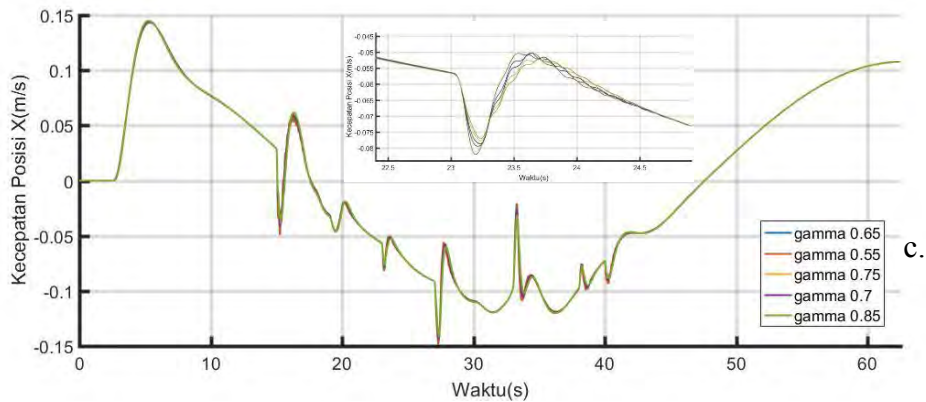
Selanjutnya dilakukan simulasi dengan nilai  $\gamma$  yang berbeda-beda ( $\gamma = 0.55, 0.65, 0.7, 0.75$  dan  $0.85$ ) untuk mengetahui sejauh mana pengaruh tingkat level pelemahan terhadap gangguan. Hasil simulasi disajikan dalam bentuk analisa posisi dan kecepatan quadcopter dengan berbagai nilai  $\gamma$  pada saat terdapat gangguan. Seperti yang terdapat pada Gambar 4.47 – 4.48.

Respon posisi, sudut *roll* dan kecepatan posisi quadcopter pada sumbu X pada saat diberi gangguan *wind gust*, dapat dilihat pada Gambar 4.47. Nilai  $\gamma$  yang berbeda-beda berpengaruh terhadap seberapa besar maksimum *overshoot* pada setiap penyimpangan *path following* yang terjadi. Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata untuk semua nilai  $\gamma$  adalah 0.00456 m. Pada saat  $\gamma = 0.7$  maksimum *overshoot* adalah 0.00165 m dan *undershoot* 0.0128 m, sedangkan saat  $\gamma = 0.65$  maksimum *overshoot* 0.0327 m dan *undershoot* 0.00175 m. Pada saat  $\gamma = 0.55$  *overshoot* dan *undershoot* mencapai 0.00195 m dan 0.0038 m.

Gambar 4.47a,b dan c menunjukkan posisi, sudut *roll* dan kecepatan quadcopter pada posisi X. Pada saat  $\gamma = 0.55$  kecepatan berkisar antara -0.54 m/s hingga 0.64 m/s. Pada  $\gamma = 0.85$  maksimum *overshoot* dan *undershoot* adalah 0.45 m/s dan -3.2 m/s.

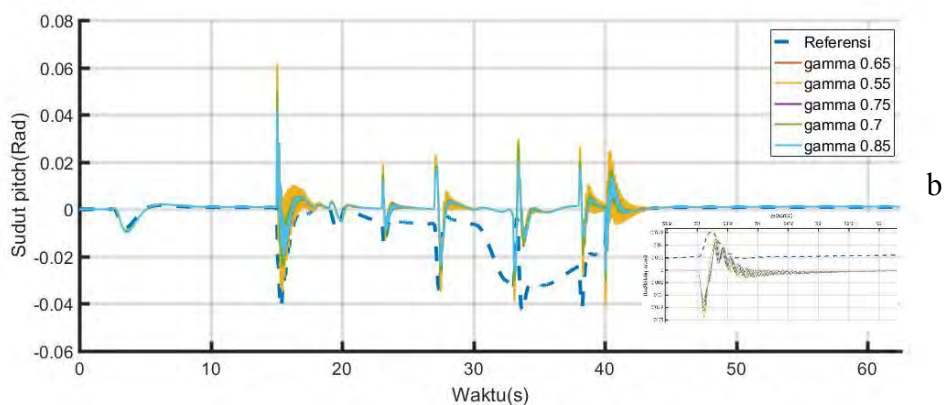
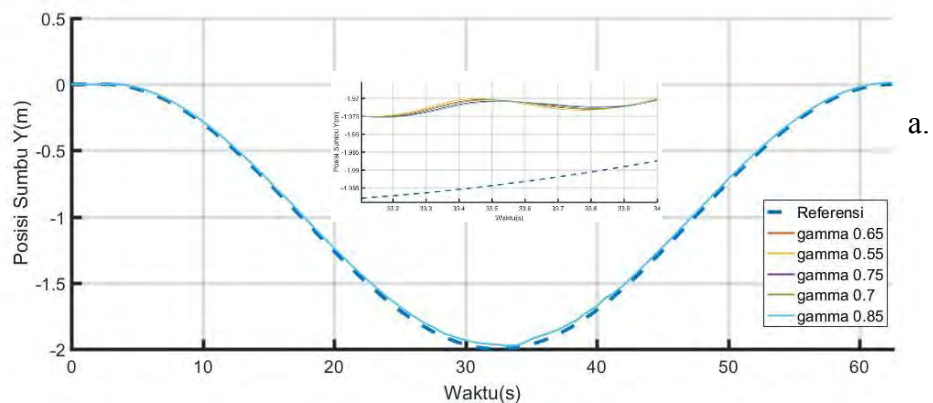
Gambar 4.48a, b, dan c adalah respon posisi, sudut *pitch* dan kecepatan quadcopter pada posisi Y dengan berbagai nilai  $\gamma$ . Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata 0.028 m. Untuk  $\gamma = 0.65$ , maksimum *overshoot* dan *undershoot* bernilai 0.045 m dan 0.13 m.

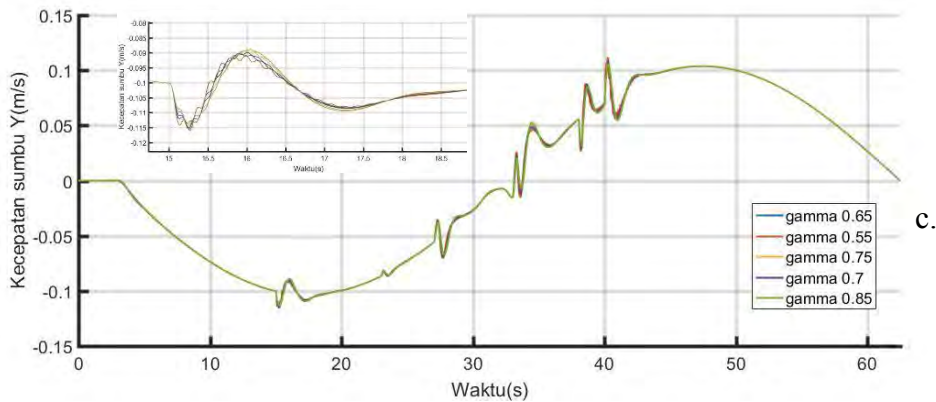




Gambar 4.47 Respon posisi  $X$ , b. Respon sudut *roll*, dan c. Respon kecepatan sumbu  $X$  dengan berbagai nilai  $\gamma$  saat *Step+Gust* 15-40(s)

Untuk  $\gamma = 0.7$  maksimum *overshoot* = 0.0332m dan *undershoot* = 0.242 m.  $\gamma = 0.75$  maksimum *overshoot* = 0.0132m dan *undershoot* = 0.0153m. Serta  $\gamma = 0.85$  maksimum *overshoot* = 0.0512m dan *undershoot* = 0.342m. Kecepatan pada sumbu  $Y$  ditunjukkan oleh Gambar 4.48c. Kecepatan maksimum untuk  $\gamma = 0.85$  adalah  $\pm 0.4$  m/s, untuk  $\gamma = 0.35$  adalah  $\pm 0.31$  m/s, untuk  $\gamma = 0.7$  adalah  $\pm 0.27$  m/s, untuk  $\gamma = 0.65$  adalah  $\pm 0.22$  m/s dan untuk  $\gamma = 0.55$  adalah  $\pm 0.18$  m/s.





Gambar 4.48 Respon posisi  $Y$ , b. Respon sudut *Roll*, dan c. Respon kecepatan sumbu  $Y$  dengan berbagai nilai  $\gamma$  saat *Step+Gust* 15-40(s)

Nilai  $\gamma$  yang berbeda-beda, akan mempengaruhi besar nilai matriks P dan gain K. Oleh karena itu, respon yang dihasilkan akan berbeda untuk setiap nilai  $\gamma$  yang diberikan. Berdasarkan seluruh respon sistem saat simulasi dengan efek gangguan, menunjukkan bahwa perbedaan nilai  $\gamma$  akan mempengaruhi penyimpangan posisi quadcopter. Besarnya penyimpangan posisi yang terjadi pada quadcopter pada nilai  $\gamma$  yang berbeda-beda dapat dilihat pada Tabel. 4.12 dan 4.13.

Tabel 4.12 Penyimpangan posisi quadcopter pada sumbu  $X$  dan  $Y$  saat *Step+Gust* 15-40(s)

$\gamma$	Penyimpangan Posisi $X$	Penyimpangan Posisi $Y$	ISE Posisi $X$	ISE Posisi $Y$
0.55	0.004434	0.0212540	0.01957	0.00045173
0.65	0.004442	0.0212547	0.01966	0.00045176
0.7	0.004445	0.0212553	0.01975	0.00045178
0.75	0.004447	0.0212560	0.01977	0.00045182
0.85	0.004451	0.0212566	0.01981	0.00045184

Tabel 4.13 Penyimpangan posisi quadcopter pada sudut *roll* dan *pitch* saat *Step+Gust* 15-40(s)

$\gamma$	Penyimpangan Sudut <i>roll</i>	Penyimpangan Sudut <i>pitch</i>	ISE Sudut <i>roll</i>	ISE Sudut <i>pitch</i>
0.55	0.005550	0.005516	0.003080	0.003094
0.65	0.005564	0.005562	0.003096	0.003120
0.7	0.005580	0.005633	0.003114	0.003154
0.75	0.005589	0.005651	0.003124	0.003173
0.85	0.005595	0.005686	0.003130	0.003193

Tabel 4.14 Gain  $K$  pada *inner loop* dan norm- $\infty$  sistem untuk berbagai nilai  $\gamma$

$\gamma$	$K$	$\ T_{zw}\ _{\infty} = \gamma^*$
0.55	[0.7102 0.7852]	0.428
0.65	[0.8394 0.0928]	0.506
0.7	[0.9040 0.1000]	0.545
0.75	[0.9685 0.1071]	0.583
0.85	[1.0976 0.1213]	0.660

Penyimpangan terkecil terjadi ketika  $\gamma = 0.7$ , jika dilihat pada Tabel 4.14, bahwa nilai  $\gamma^*$  mendekati  $\gamma$  yaitu 0.545. Semakin nilai  $\gamma^*$  mendekati  $\gamma$  ( $\gamma^* = \gamma$ ), maka pada kondisi ini menghasilkan ketahanan sistem terhadap gangguan dengan performa yang baik..

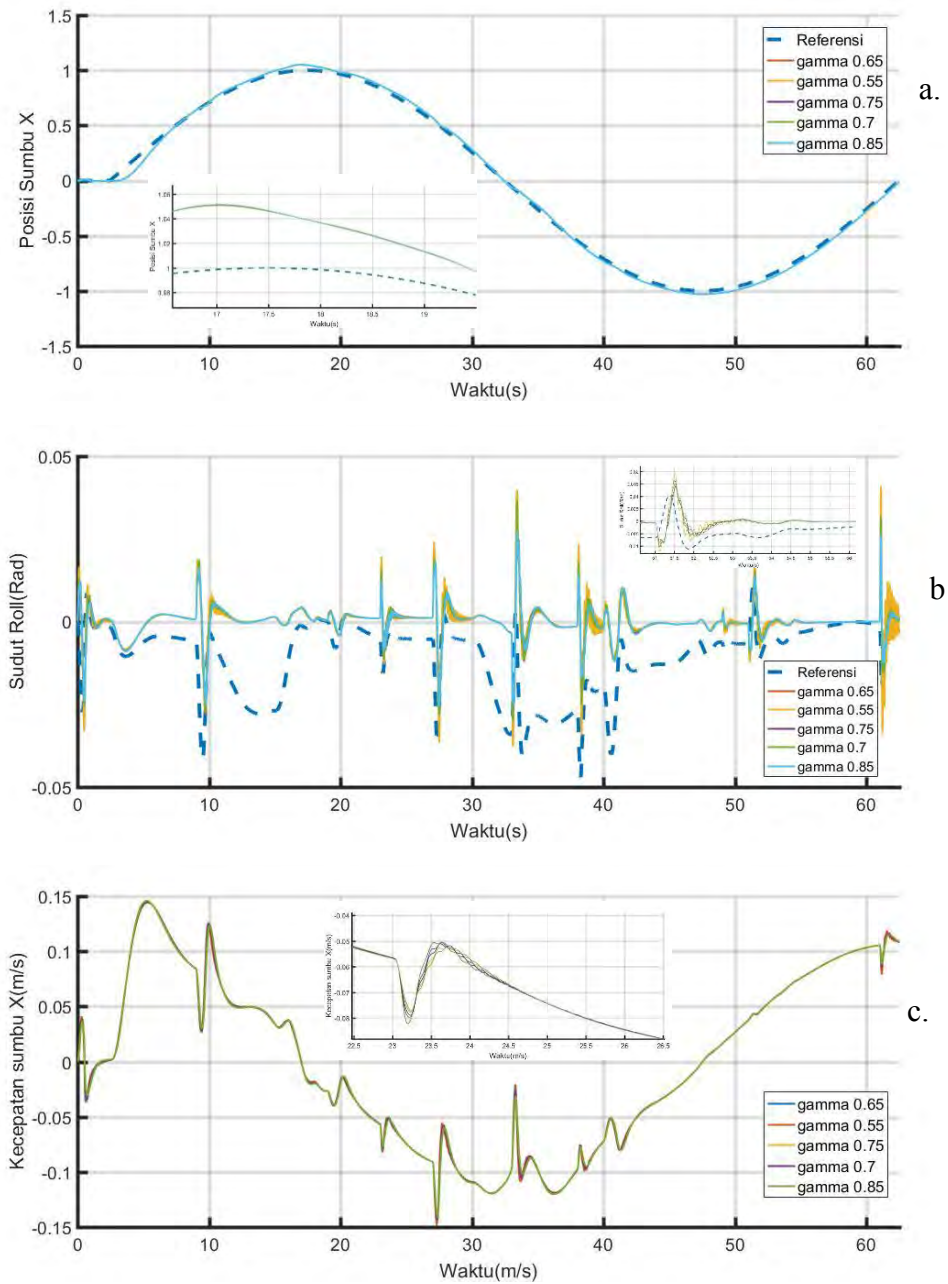
#### D. Pengujian dengan Perubahan nilai $\gamma$ *Disturbance Model Modifikasi Penggabungan Step+Gust 0-62.5(s)*

Selanjutnya dilakukan simulasi dengan nilai  $\gamma$  yang berbeda-beda ( $\gamma = 0.55, 0.65, 0.7, 0.75$  dan  $0.85$ ) untuk mengetahui sejauh mana pengaruh tingkat level pelemahan terhadap gangguan. Hasil simulasi disajikan dalam bentuk analisa posisi dan kecepatan quadcopter dengan berbagai nilai  $\gamma$  pada saat terdapat gangguan. Seperti yang terdapat pada Gambar 4.49 – 4.50.

Respon posisi, sudut *roll* dan kecepatan posisi quadcopter pada sumbu X pada saat diberi gangguan *wind gust*, dapat dilihat pada Gambar 4.47. Nilai  $\gamma$  yang berbeda-beda berpengaruh terhadap seberapa besar maksimum *overshoot* pada setiap penyimpangan *path following* yang terjadi. Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata untuk semua nilai  $\gamma$  adalah 0.00456. Pada saat  $\gamma = 0.7$  maksimum *overshoot* adalah 0.00145 m dan *undershoot* 0.0228 m, sedangkan saat  $\gamma = 0.65$  maksimum *overshoot* 0.0127 m dan *undershoot* 0.00175 m. Pada saat  $\gamma = 0.544$  *overshoot* dan *undershoot* mencapai 0.00195 m dan 0.0238 m.

Gambar 4.49a,b dan c menunjukkan posisi, sudut *roll* dan kecepatan quadcopter pada posisi X. Pada saat  $\gamma = 0.55$  kecepatan berkisar antara -0.34 m/s hingga 0.64 m/s. Pada  $\gamma = 0.85$  maksimum *overshoot* dan *undershoot* adalah 0.25 m/s dan -1.2 m/s.

Gambar 4.50a, b, dan c adalah respon posisi, sudut *pitch* dan kecepatan quadcopter pada posisi Y dengan berbagai nilai  $\gamma$ . Penyimpangan pada detik ke-15 cukup kecil yaitu rata-rata 0.0408 m. Untuk  $\gamma = 0.65$ , maksimum *overshoot* dan *undershoot* bernilai 0.39 m dan 0.17 m. Untuk  $\gamma = 0.7$  maksimum *overshoot* = 0.0152m dan *undershoot* = 0.0427 m.

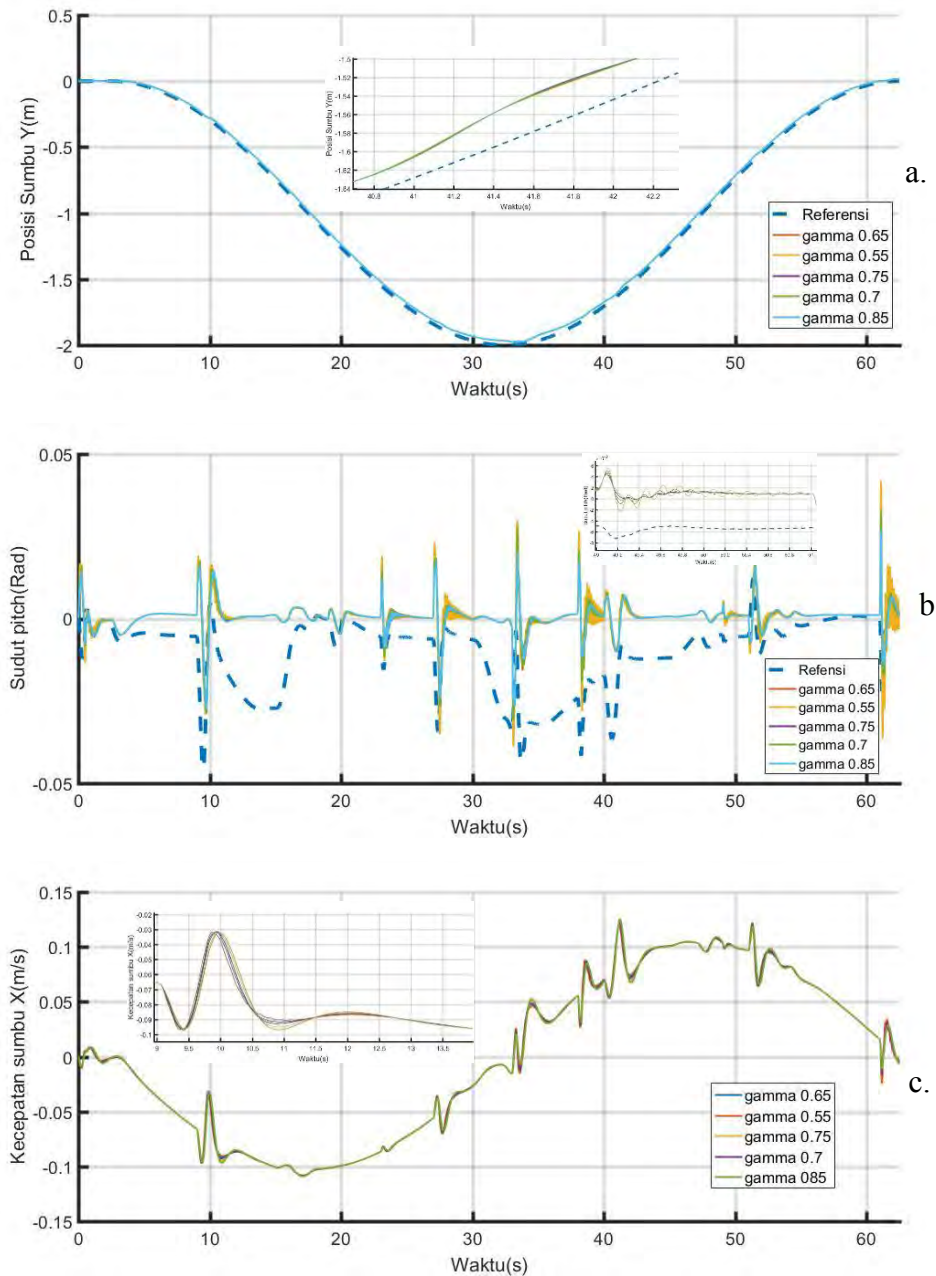


Gambar 4.49 Respon posisi  $X$ , b. Respon sudut roll, dan c. Respon kecepatan sumbu  $X$  dengan berbagai nilai  $\gamma$  dengan *Disturbance Step+Gust* 0-62.5(s)

$\gamma = 0.75$  maksimum *overshoot* = 0.032m dan *undershoot* = 0.135m. Serta  $\gamma = 0.85$  maksimum *overshoot* = 0.0532m dan *undershoot* = 0.232m. Kecepatan pada sumbu Y ditunjukkan oleh Gambar 4.50c. Kecepatan maksimum untuk  $\gamma = 0.85$  adalah  $\pm 0.5$  m/s, untuk  $\gamma = 0.75$  adalah  $\pm 0.46$  m/s, untuk  $\gamma = 0.7$  adalah  $\pm 0.42$  m/s, untuk  $\gamma = 0.65$  adalah  $\pm 0.38$  m/s dan untuk  $\gamma = 0.55$  adalah  $\pm 0.32$  m/s.



Nilai  $\gamma$  yang berbeda-beda, akan mempengaruhi besar nilai matriks P dan gain K. Oleh karena itu, respon yang dihasilkan akan berbeda untuk setiap nilai  $\gamma$  yang diberikan. Berdasarkan seluruh respon sistem saat simulasi dengan efek gangguan,-



Gambar 4.50 Respon posisi  $Y$ , b. Respon sudut  $Roll$ , dan c. Respon kecepatan sumbu  $Y$  dengan berbagai nilai  $\gamma$  dengan  $Disturbance\ Step+Gust\ 0-62.5(s)$

menunjukkan bahwa perbedaan nilai  $\gamma$  akan mempengaruhi penyimpangan posisi quadcopter. Besarnya penyimpangan posisi yang terjadi pada quadcopter pada nilai  $\gamma$  yang berbeda-beda dapat dilihat pada Tabel. 4.15 dan Tabel. 4.16.

Penyimpangan terkecil terjadi ketika  $\gamma = 0.7$ , jika dilihat pada Tabel 4.17, bahwa nilai  $\gamma^*$  mendekati  $\gamma$  yaitu 0.5081. Semakin nilai  $\gamma^*$  mendekati  $\gamma$  ( $\gamma^* = \gamma$ ), maka pada kondisi ini menghasilkan ketahanan sistem terhadap gangguan dengan performa yang baik.

Tabel 4.15 Penyimpangan posisi quadcopter pada sumbu  $X$  dan  $Y$  saat *Step+Gust* 0-62.5(s)

$\gamma$	Penyimpangan Posisi $X$	Penyimpangan Posisi $Y$	ISE Posisi $X$	ISE Posisi $Y$
0.55	0.0046041	0.0206086	0.00002119	0.00042471
0.65	0.0046070	0.0206098	0.00002122	0.00042476
0.7	0.0046104	0.0206099	0.00002125	0.00042477
0.75	0.0046106	0.0206105	0.00002125	0.00042479
0.85	0.0046109	0.0206119	0.00002126	0.00042485

Tabel 4.16 Penyimpangan posisi quadcopter pada sudut *roll* dan *pitch* saat *Step+Gust* 0-62.5(s)

$\gamma$	Penyimpangan Sudut <i>roll</i>	Penyimpangan Sudut <i>pitch</i>	ISE Sudut <i>roll</i>	ISE Sudut <i>pitch</i>
0.55	0.010231	0.010235	0.0001046	0.000104
0.65	0.010245	0.010258	0.0001049	0.000105
0.7	0.010254	0.010296	0.0001051	0.000106
0.75	0.010260	0.010315	0.0001052	0.000106
0.85	0.010265	0.010344	0.0001053	0.000107

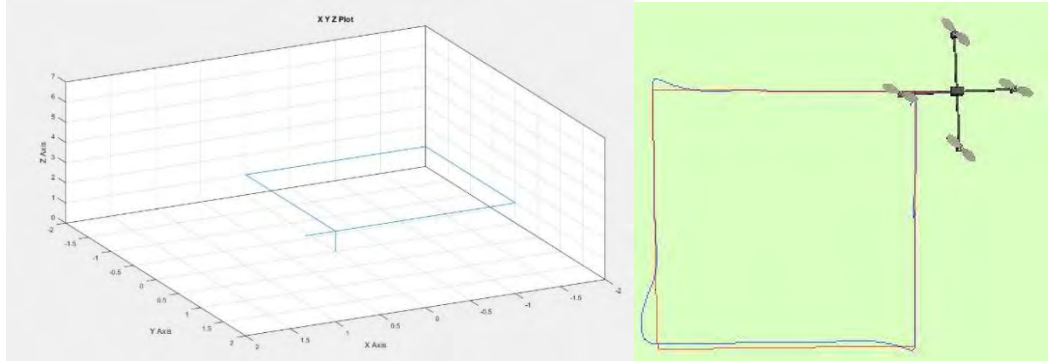
Tabel 4.17 Gain  $K$  pada *inner loop* dan norm- $\infty$  sistem untuk berbagai nilai  $\gamma$

$\gamma$	$K$	$\ T_{zw}\ _{\infty} = \gamma^*$
0.55	[0.7549 0.0833]	0.3992
0.65	[0.8922 0.0985]	0.4718
0.7	[0.9609 0.1061]	0.5081
0.75	[1.0295 0.1136]	0.5443
0.85	[1.1667 0.1287]	0.6168

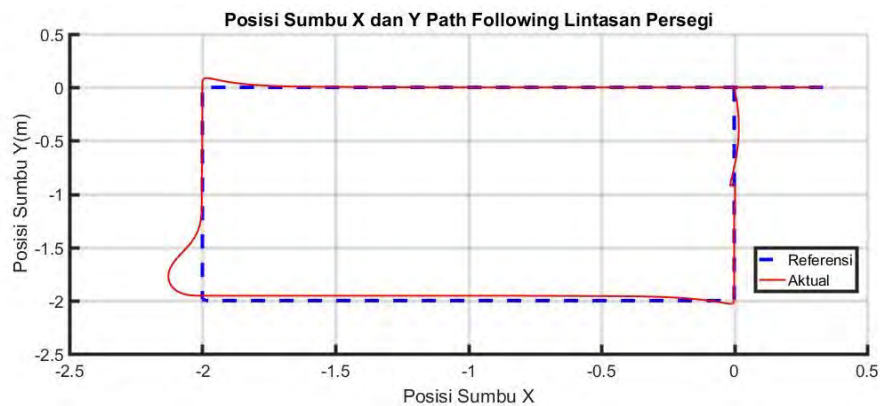
#### 4.5 Simulasi dengan Sinyal Referensi yang Berbeda-beda

Pengujian dilakukan dengan memberikan sinyal referensi yang berbeda-beda terhadap sistem kontrol yang telah didesain tanpa mengubah parameter

kontrol apapun. Seperti pada Gambar 4.51, referensi yang diberikan adalah sinyal berbentuk persegi. Hasilnya, sistem dapat melakukan *path following* namun terdapat *overshoot* posisi sebesar 1.7 m pada posisi  $X$  dan 0.9 m pada posisi  $Y$ .



Gambar 4.51 *Path following* Posisi dengan Lintasan Persegi

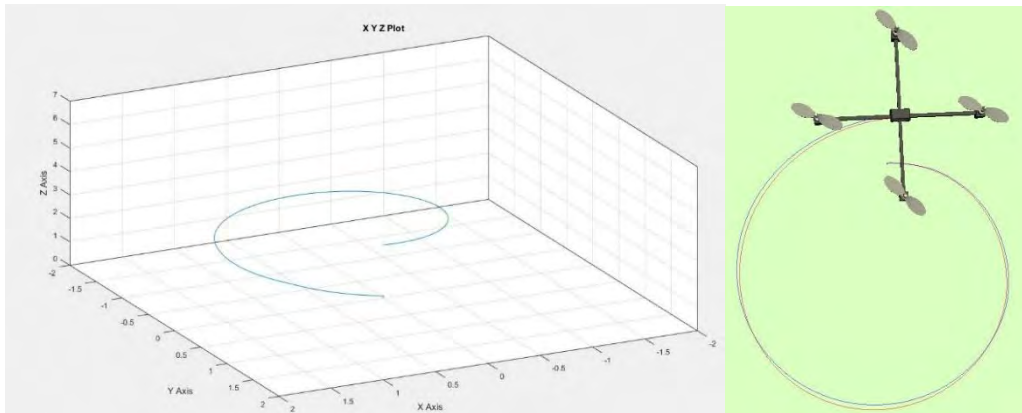


Gambar 4.52 Posisi X dan Y Pada Saat *Path following* Lintasan Berbentuk Persegi

Hal ini dikarenakan dinamika pada saat berbelok sudut sebesar  $90^\circ$  (tegak lurus) berusaha mendekati model dari bentuk persegi. Dengan menambah algoritma LOS yang mengatur arah hadap dari quadcopter maka dinamika mampu berbelok pada kondisi sudut sebesar  $90^\circ$  (tegak lurus) dapat dilihat pada Gambar 4.52. Sehingga memperbaiki kekurangan penelitian sebelumnya dimana terjadi penyimpangan posisi sebesar 0.18 m pada posisi  $X$  dan 0.16 m pada posisi  $Y$  dan tidak dapat berbelok dengan ketajaman sudut sebesar  $90^\circ$ .

Selain itu, sinyal referensi dengan bentuk persegi juga diberikan kepada sistem, seperti yang ditunjukkan oleh Gambar 4.53, referensi yang diberikan adalah sinyal berbentuk spiral (*helix*) dengan  $x_d = \sin(0.07\pi t)$ ,  $Y_d = -1 + \cos(0.07\pi t)$ , dan

$Z_d = t \text{ zd} = t$ . Hasilnya, sistem quadcopter dapat melakukan *path following* mengikuti referensi yang diberikan.



Gambar 4.53 *Path following* Posisi dengan Lintasan Spiral (*Helix*)

#### 4.6 Perbandingan Kontroler *Output Feedback Command-Generator Tracker* (CGT) dengan Hasil Desain

Pada sub-bab ini, hasil simulasi dari kontroler *tracking* menggunakan struktur *Output Feedback Command-Generator Tracker* (CGT) [19] yang didesain pada tesis sebelumnya ini akan dibandingkan dengan kontroler *path following* *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS). Perbandingan hasil simulasi yang akan disajikan yaitu hasil *tracking* dan *path following* tanpa gangguan dan hasil *tracking* dan *path following* dengan adanya *wind gust* (*step+gust*) pada lintasan lingkaran dan persegi.

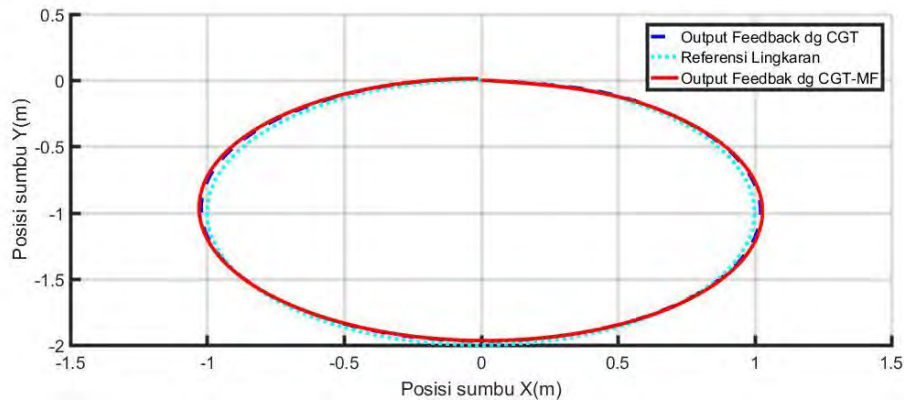
##### 4.6.1 Pengujian Lintasan Lingkaran

Sinyal referensi yang diberikan pada kedua kontroler adalah sama yaitu, lintasan berbentuk lingkaran. Referensi untuk posisi X dan Y diberikan setelah quadcopter berada pada posisi melayang, yaitu sekitar detik ke-2.5. Gambar 4.54 menunjukkan perbandingan respon *tracking* dan *path following* ketika tidak ada gangguan.

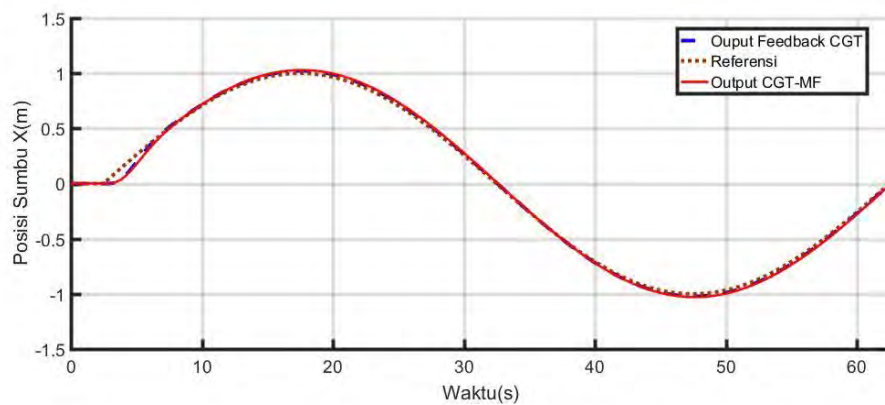
Terlihat bahwa respon dengan kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) dapat melakukan *path following* melingkar namun, terdapat sedikit pergeseran *path* dari referensinya di beberapa koordinat. Kontroler *output feedback* dengan *Output Feedback Command-Generator Tracker* (CGT) menghasilkan *tracking* lintasan yang tepat sesuai dengan referensi.

Pada Gambar 4.55 yaitu *tracking* dan *path following* posisi sumbu x dimana *tracking* dan *path following* dimulai pada detik ke-2.5. Kontroler *Output Feedback Command-Generator Tracker* (CGT) pada gambar 4.57 terdapat *delay*

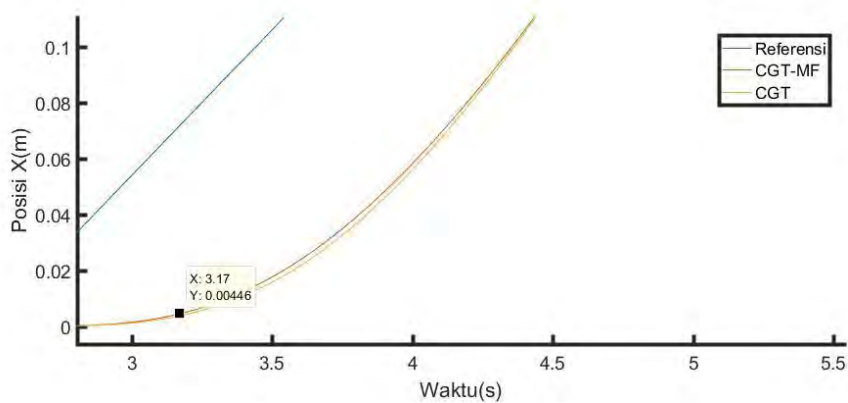
$\pm 3.21$  detik sedangkan kontroler *Output Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) pada gambar 4.56 terdapat *delay*  $\pm 3.17$  detik.



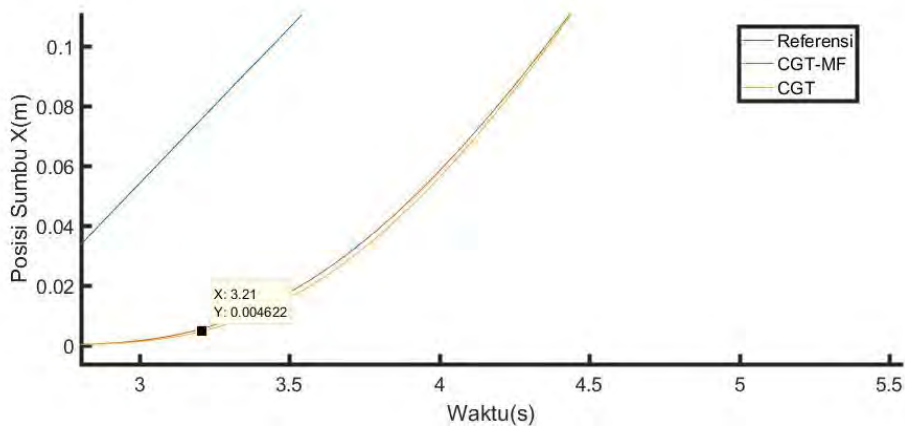
Gambar 4.54 *Tracking* dan *path following* Lintasan Lingkaran tanpa gangguan



Gambar 4.55 Perbandingan dalam posisi sumbu X(m)

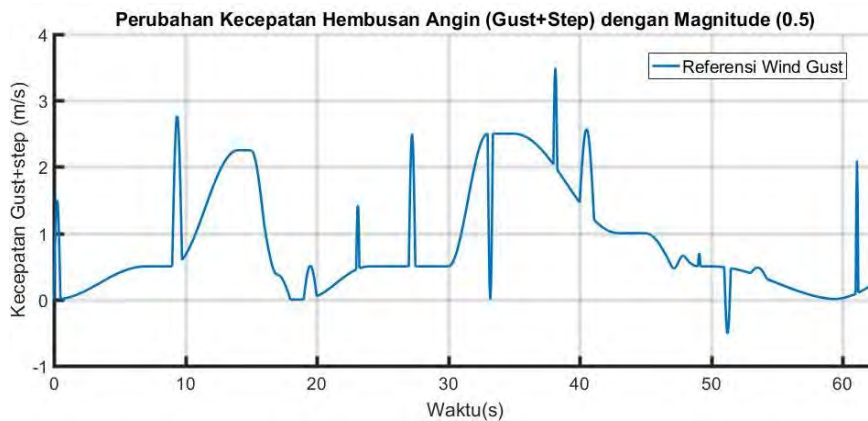


Gambar 4.56 Waktu *delay* yang dibutuhkan kontroler CGT-MF

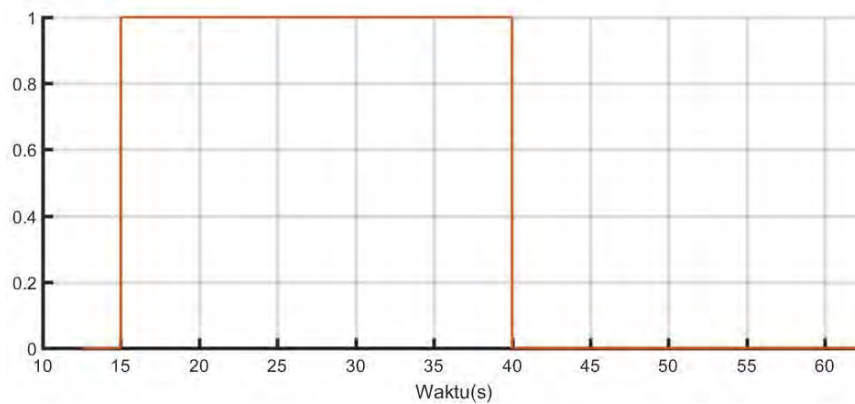


Gambar 4.57 Waktu *delay* yang dibutuhkan kontroler CGT

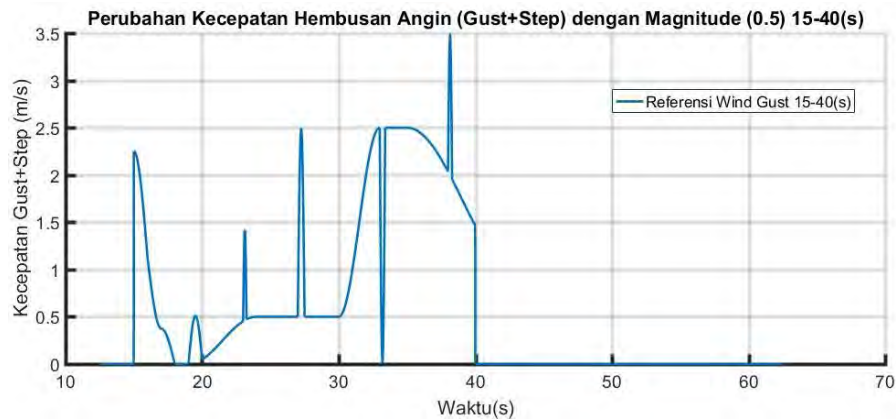
Selanjutnya disimulasikan untuk membandingkan ketahanan sistem terhadap gangguan *wind gust* yang diberikan. Untuk membandingkan kedua metode kontrol, perlakuan yang sama diterapkan baik pada dinamika *plant* maupun kontroler dari kedua metode kontrol.



a.



b.

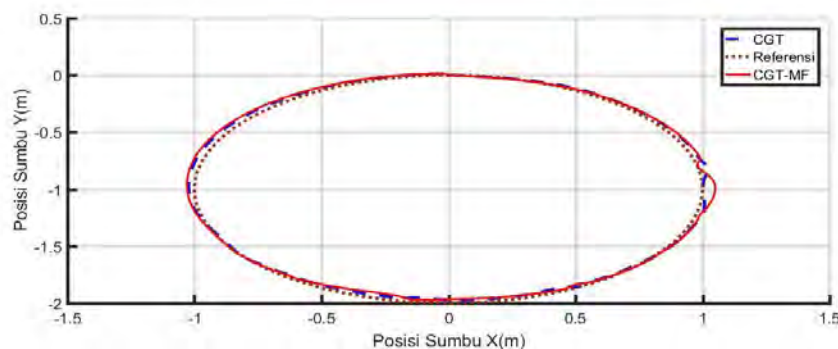


c.

Gambar 4.58 a. *Wind gust* secara keseluruhan, b. pemberian gangguan pada detik ke-15 hingga detik ke-40 dan c. *Wind gust* pada sistem

Pada [19], gangguan hanya dapat diberikan dengan maksimum amplitude dari *wind gust* sebesar 1.5 m/s, sedangkan pada desain kontroler yang diajukan dapat mencapai 3 m/s. Oleh karena itu, perbandingan dua buah kontroler ini dilakukan dengan besar amplitude dari gangguan seperti pada Gambar 4.56. Dimana *wind gust* hanya diberikan dalam rentang waktu detik ke-15 hingga ke-40 dengan amplitude 1.5 m/s.

Gambar 4.59 menunjukkan respon *tracking* dan *path following* dengan *output feedback* dengan CGT dan *output feedback* dengan CGT-MF jika dilihat pada bidang X dan Y. Untuk respon dengan CGT-MF menghasilkan *path following* yang sesuai dengan referensi dengan penyimpangan sekitar 0.524 m pada saat diberi *wind gust*. Kedua sistem kontrol tersebut dapat mengatasi gangguan yang diberikan. Penyimpangan *tracking* dan *path following* yang terjadi sangat besar, yaitu maksimum penyimpangan mencapai 1.04 m. Untuk lebih jelas, hasil *tracking* disajikan dalam sumbu X dan Y secara terpisah seperti pada Gambar 4.60 dan 4.61.

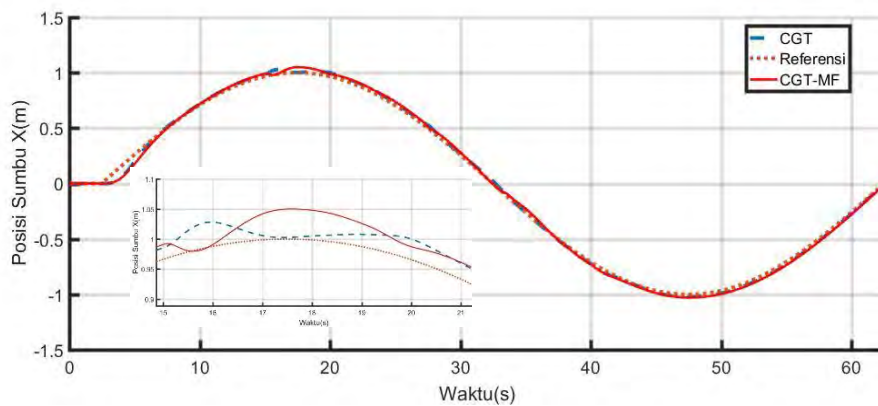


Gambar 4.59 Respon quadcopter pada sumbu X dan Y dengan gangguan

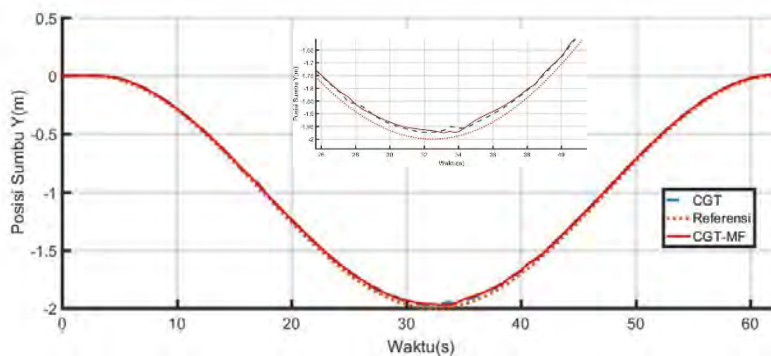
Gambar 4.60 menunjukkan perbandingan respon posisi pada sumbu X. Respon kontroler *output feedback* dengan CGT menyimpang dari referensi hanya pada detik ke-15 dimana membutuhkan waktu kurang dari 5 detik untuk dapat kembali *tracking* sesuai referensinya. Penyimpangan terjadi sekitar 3.17 m. Penyimpangan respon untuk kontroler *output feedback* dengan CGT-MF sangat besar dan terjadi selama rentang detik ke-15. Maksimum penyimpangan posisi X mencapai 1.04 m.

Gambar 4.61 menunjukkan respon posisi pada sumbu Y. Pengaruh gangguan yang diberikan pada detik ke-30 cukup kecil, yaitu membuat penyimpangan pada kedua kontroler sebesar  $\pm 0.0096$  m. Untuk kontroler *output feedback* dengan CGT penyimpangan terjadi sebesar 0.011129 m, sedangkan untuk kontroler *output feedback* dengan CGT-MF penyimpangan mencapai 0.008116 m.

Kontroler *output feedback* dengan CGT-MF dapat membawa quadcopter mengikuti referensi yang diberikan, dan menghasilkan *path following* yang kecil pada saat terdapat gangguan. Ketika terdapat gangguan, penyimpangan tetap ada, namun sistem kontrol yang diajukan pada tesis ini dapat dengan cepat membuat quadcopter kembali pada lintasan referensi.



Gambar 4.60 Perbandingan respon posisi X dengan gangguan



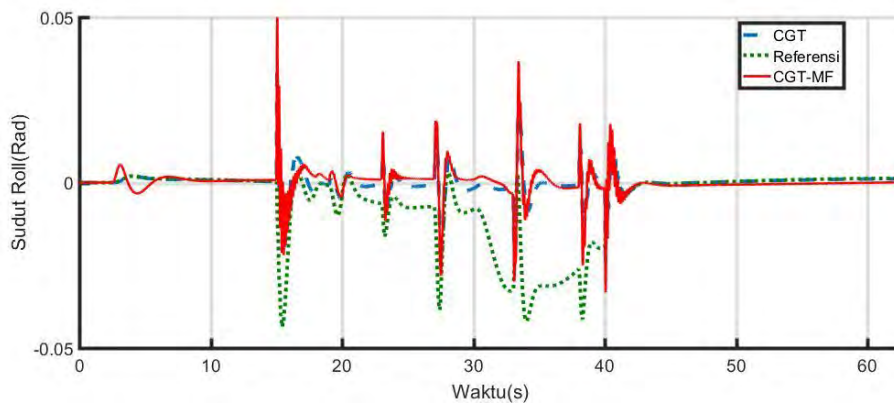
Gambar 4.61 Perbandingan respon posisi Y dengan gangguan



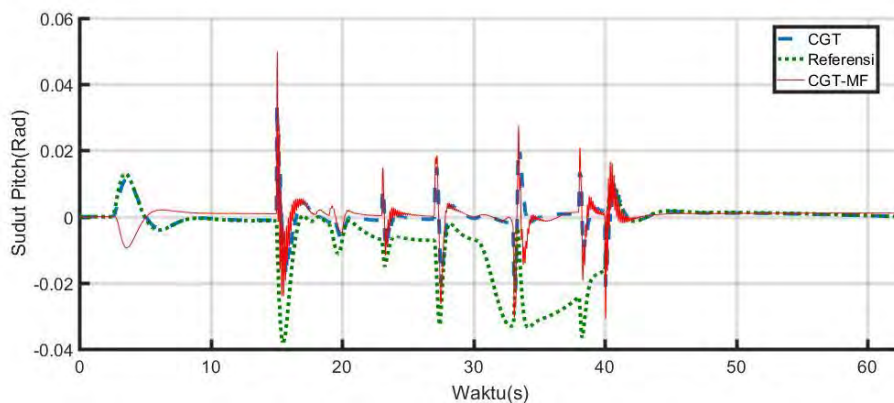
Gambar 4.62 dan Gambar 4.63 menunjukkan respon sudut *pitch* dan *roll* hasil perbandingan kedua metode kontrol. Respon sudut *roll* dan *pitch* dari kontroler output feedback dengan CGT terjadi *overshoot/undershoot* hingga  $\pm 0.0025271$  rad dan kembali pada posisi 0 rad dengan cepat yaitu kurang dari 5 detik.

Perilaku sudut *roll* dan *pitch* dari *output feedback* dengan CGT-MF memiliki maksimum amplitude  $\pm 0.00250269$  rad dan terjadi dalam rentang waktu yang cukup lama yaitu sekitar 5.5 detik. Hal tersebut menyebabkan respon posisi quadcopter menyimpang dan cukup lama untuk dapat kembali pada lintasan referensinya

Dari tabel 4.18 dapat dilihat perbandingan penyimpangan yang terjadi pada kedua kontroler pada saat adanya gangguan. Respon sistem kontroler CGT-MF memiliki nilai lebih kecil penyimpangannya daripada kontroler CGT terkecuali pada sudut *roll*. Sedangkan pada perbandingan kecepatan dalam mengatasi adanya gangguan. Kontroler CGT lebih cepat mengatasi adanya gangguan dibanding dengan kontroler CGT-MF pada sudut *roll*.



Gambar 4.62 Perbandingan respon sudut *Roll* dengan gangguan



Gambar 4.63 Perbandingan respon sudut *Pitch* dengan gangguan

Tabel 4.18 Perbandingan rata-rata penyimpangan dan kecepatan kedua metode dalam lintasan lingkaran

Penyimpangan dan kecepatan Posisi/Sudut/Kecepatan	CGT-MF		CGT	
	Tanpa <i>Disturbance</i>	Dengan <i>Disturbance</i>	Tanpa <i>Disturbance</i>	Dengan <i>Disturbance</i>
Posisi X	-0.00332 m	0.000901 m	-0.00337 m	0.002422 m
ISE Posisi X	1.1 m	8.12 m	1.14 m	5.87 m
Posisi Y	-0.92523 m	-0.9272 m	-0.93973 m	-0.94391 m
ISE Posisi Y	0.856052 m	0.859701 m	0.883087 m	0.89096 m
Sudut roll	0.00016 rad	0.000192 rad	3.79 rad	-0.00567 rad
ISE Sudut roll	2.55 rad	3.69 rad	1.44 rad	3.22 rad
Sudut pitch	0.000388 rad	0.00035855 rad	0.000174 rad	0.00021132 rad
ISE Sudut pitch	1.5 rad	1.2856 rad	3.04 rad	4.4658 rad
Sudut yaw	4.392352 rad	4.432552 rad	5.42 rad	0.028174 rad
ISE Sudut yaw	19.29275 rad	19.64752 rad	2.93 rad	0.000794 rad
Kecepatan X	-0.0001	-0.0005	-0.00012	-0.00039
ISE Kecepatan X	1.01	2.52	1.34	1.52
Kecepatan Y	0.000181	-0.00023	0.000183	-0.0005
ISE Kecepatan Y	3.27	5.44	3.35	2.47
Kecepatan Z	0.015533	0.015344	0.015982	0.015677
ISE Kecepatan Z	0.000241	0.000235	0.000255	0.000246

#### 4.6.2 Pengujian Lintasan Persegi

Sinyal referensi yang diberikan pada kedua kontroler adalah sama yaitu, lintasan berbentuk persegi. Referensi untuk posisi X dan Y diberikan setelah quadcopter berada pada posisi melayang, yaitu sekitar detik ke-2.5. Gambar 4.62 menunjukkan perbandingan respon *tracking* dan *path following* ketika tidak ada gangguan.

Terlihat bahwa respon dengan kontroler Output *Feedback Command-Generator Tracker Model Following* (CGT-MF) dengan *Line Of Sight* (LOS) dapat melakukan *tracking* dan *path following* lintasan persegi namun, terdapat *overshoot* dari referensinya di beberapa koordinat. Kontroler output *feedback* dengan *Output Feedback Command-Generator Tracker* (CGT) menghasilkan *tracking* lintasan

yang tepat sesuai dengan referensi namun, terdapat pergeseran *tracking* dari referensinya.

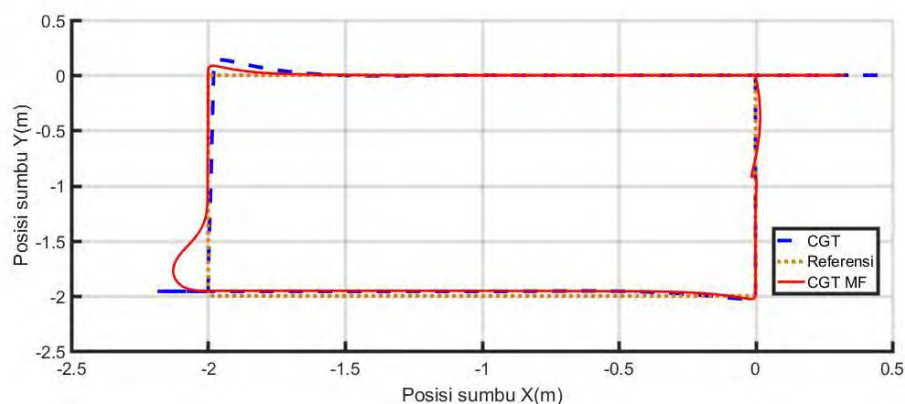
Selanjutnya disimulasikan untuk membandingkan ketahanan sistem terhadap gangguan *wind gust* yang diberikan. Untuk membandingkan kedua metode kontrol, perlakuan yang sama diterapkan baik pada dinamika *plant* maupun kontroler dari kedua metode kontrol.

Pada [19], gangguan hanya dapat diberikan dengan maksimum amplitude dari *wind gust* sebesar 1.5 m/s, sedangkan pada desain kontroler yang diajukan dapat mencapai 3 m/s. Oleh karena itu, perbandingan dua buah kontroler ini dilakukan dengan besar amplitude dari gangguan seperti pada Gambar 4.63. Dimana *wind gust* hanya diberikan dalam rentang waktu detik ke-15 hingga ke-40 dengan amplitude 1.5 m/s.

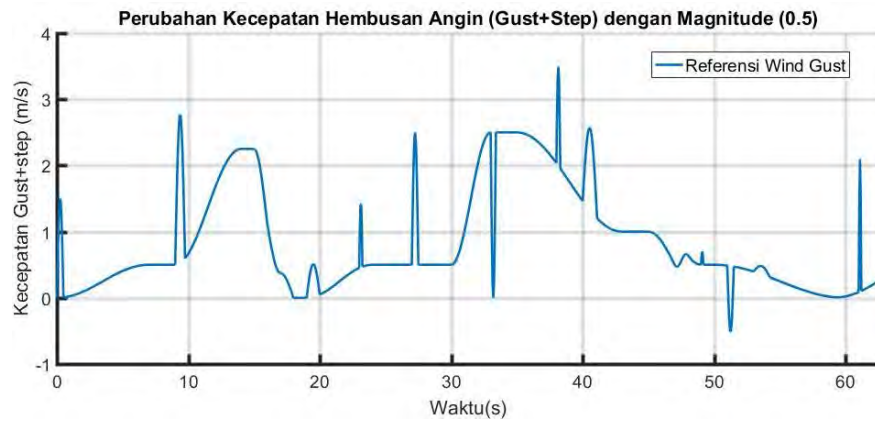
Gambar 4.64 menunjukkan respon *tracking* dan *path following* dengan *output feedback* dengan CGT dan *output feedback* dengan CGT-MF jika dilihat pada bidang X dan Y. Untuk respon dengan CGT-MF menghasilkan *path following* yang sesuai dengan referensi dengan *overshoot* sekitar 0.249013 m sedangkan kontroler CGT memiliki *overshoot* sebesar 0.489082 m pada saat diberi *wind gust*. Kedua sistem kontrol tersebut dapat mengatasi gangguan yang diberikan.

Penyimpangan *tracking* dan *path following* yang terjadi sangat besar, yaitu maksimum penyimpangan mencapai 0.489082 m. Untuk lebih jelas, hasil *tracking* dan *path following* disajikan dalam sumbu X dan Y secara terpisah seperti pada Gambar 4.65 dan 4.66.

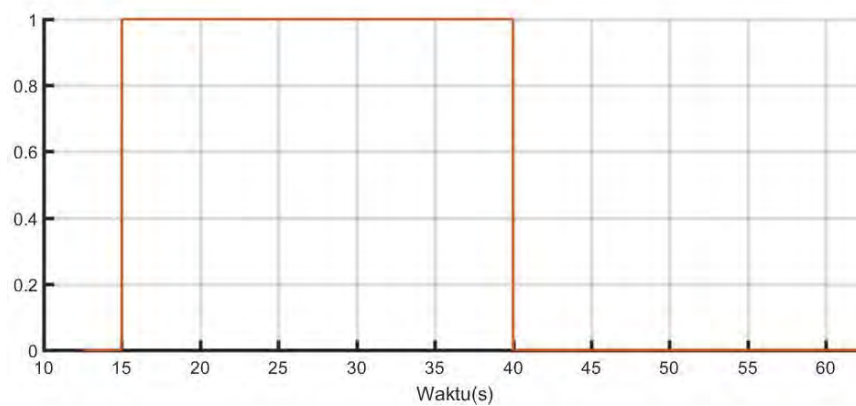
Gambar 4.66 menunjukkan perbandingan respon posisi pada sumbu X. Respon kontroler *output feedback* dengan CGT menyimpang dari referensi hanya pada detik ke-20, ke-30 dan ke-50 dimana membutuhkan waktu kurang dari 5 detik untuk dapat kembali *tracking* sesuai referensinya. Penyimpangan terjadi sekitar 0.489082 m. Penyimpangan respon untuk kontroler *output feedback* dengan CGT-



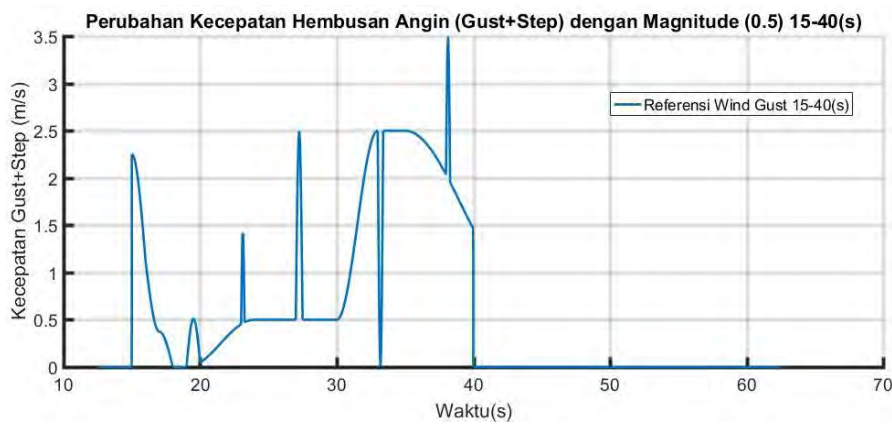
Gambar 4.64 *Tracking* dan *path following* Lintasan Persegi tanpa gangguan



a.



b.

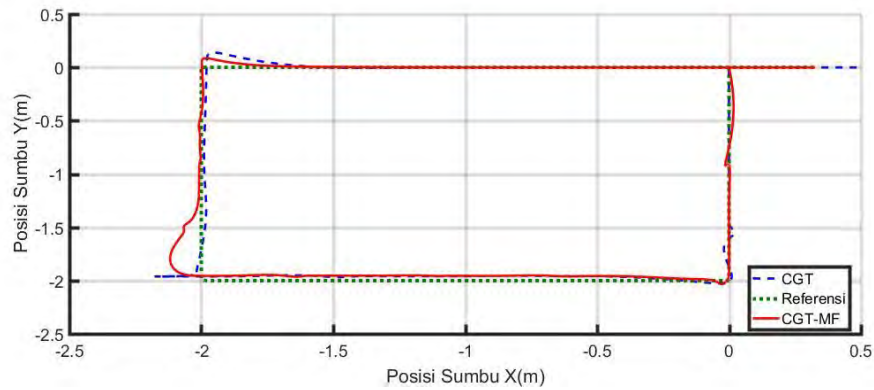


c.

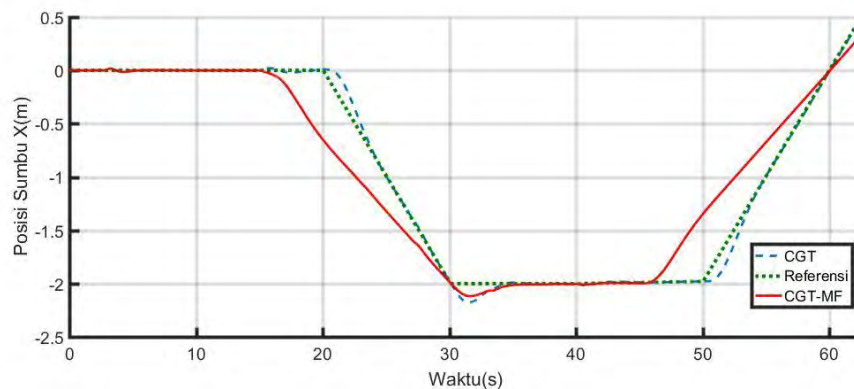
Gambar 4.65 a. *Wind gust* secara keseluruhan, b. pemberian gangguan pada detik ke-15 hingga detik ke-40 dan c. *Wind gust* pada sistem

MF sangat besar dan terjadi selama rentang detik ke-15 hingga ke-35. Maksimum penyimpangan posisi  $X$  mencapai 0.249013 m. Penyimpangan terjadi dikarenakan terdapat algoritma LOS yang membantu menjaga arah hadap. Algoritma LOS memiliki kendala jika terdapat sudut  $90^\circ$  dimana, LOS diambil dari nilai  $\tan_a = \frac{Y}{X}$ .

Nilai tan 90° menghasilkan nilai tak terhingga sehingga, jika dilakukan pengurang sebanyak 6 m sebelum berbelok menuju sudut 90°.



Gambar 4.66 Respon quadcopter pada sumbu X dan Y dengan gangguan



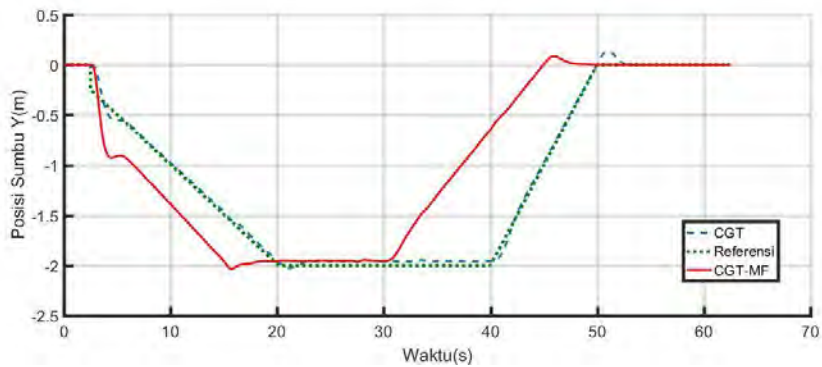
Gambar 4.67 Perbandingan respon posisi X dengan gangguan

Gambar 4.59 menunjukkan respon posisi pada sumbu Y. Pengaruh gangguan yang diberikan pada detik ke-5, ke-20 serta detik ke-45 cukup kecil pada kontroler *output feedback* CGT-MF sebesar  $\pm 0.056285$  m. Algoritma LOS yang diitambahkan pada kontroler *output feedback* CGT-MF membuat respon quadcopter terdapat jarak dengan sinyal referensi. Untuk kontroler *output feedback* dengan CGT penyimpangan terjadi sebesar -0.62972 m.

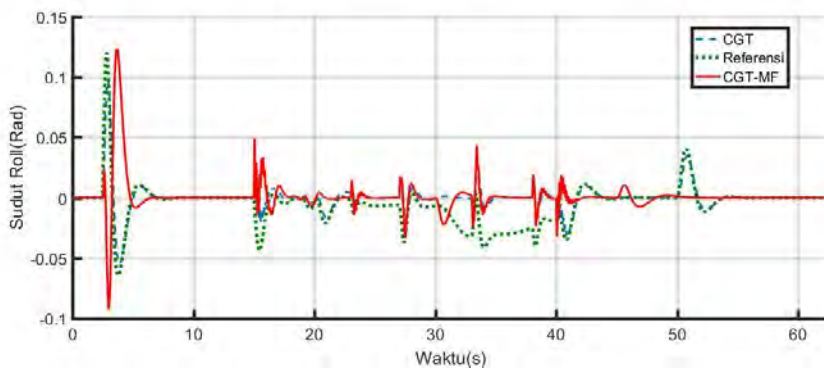
Kontroler *output feedback* dengan CGT-MF dapat membawa quadcopter mengikuti referensi yang diberikan, dan menghasilkan *path following* yang kecil pada saat terdapat gangguan. Ketika terdapat gangguan, penyimpangan tetap ada, namun sistem kontrol yang diajukan pada tesis ini dapat dengan cepat membuat quadcopter kembali pada lintasan referensi. Gambar 4.68 dan Gambar 4.69 menunjukkan respon sudut *pitch* dan *roll* hasil perbandingan kedua metode kontrol. Respon sudut *roll* dan *pitch* dari kontroler *output feedback* dengan CGT terjadi

*overshoot/undershoot* hingga  $\pm 0.0092$  rad dan kembali pada posisi 0 rad dengan cepat yaitu kurang dari 5 detik.

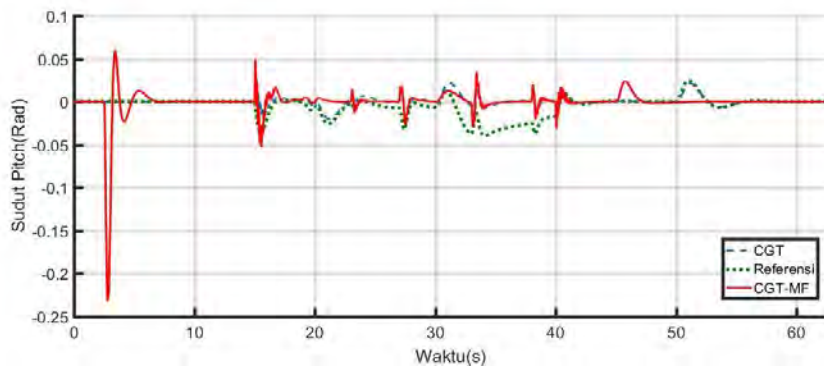
Perilaku sudut *roll* dan *pitch* dari *output feedback* dengan CGT-MF memiliki maksimum amplitude  $\pm 0.092869$  rad dan terjadi dalam rentang waktu yang cukup lama yaitu sekitar 5 detik.



Gambar 4.68 Perbandingan respon posisi Y dengan gangguan



Gambar 4.69 Perbandingan respon sudut Roll dengan gangguan



Gambar 4.70 Perbandingan respon sudut Pitch dengan gangguan

Hal tersebut menyebabkan respon posisi quadcopter menyimpang dan cukup lama untuk dapat kembali pada lintasan referensinya.

Dari tabel 4.19 dapat dilihat perbandingan penyimpangan yang terjadi pada kedua kontroler pada saat adanya gangguan. Respon sistem kontroler CGT-MF memiliki nilai lebih kecil penyimpangannya daripada kontroler CGT pada posisi sumbu *X* dan *Y* tetapi, pada sudut *roll* dan *pitch* dalam melemahkan gangguan kontroler CGT lebih baik dari pada CGT-MF. Sedangkan pada perbandingan kecepatan dalam mengatasi adanya gangguan. Kontroler CGT-MF lebih cepat mengatasi adanya gangguan dibanding dengan kontroler CGT terkecuali pada pada sumbu *Y*.

Tabel 4.19 Perbandingan rata-rata penyimpangan dan kecepatan kedua metode dalam lintasan persegi

Penyimpangan Posisi/Sudut/ Kecepatan	CGT-MF		CGT	
	Tanpa <i>Disturbance</i>	Dengan <i>Disturbance</i>	Tanpa <i>Disturbance</i>	Dengan <i>Disturbance</i>
Posisi <i>X</i>	-0.93931 m	-0.93786 m	-0.96366 m	-0.96145 m
ISE Posisi <i>X</i>	0.882294 m	0.879584 m	0.92864 m	0.924389 m
Posisi <i>Y</i>	-0.96596 m	-0.97038 m	-1.10358 m	-1.11494 m
ISE Posisi <i>Y</i>	0.933076 m	0.941636 m	1.217886 m	1.2431 m
Sudut <i>roll</i>	0.000743 Rad	0.00078 Rad	-2.4 Rad	-2.3 Rad
ISE Sudut <i>roll</i>	5.52 Rad	6.09 Rad	5.56 Rad	5.39 Rad
Sudut <i>pitch</i>	-0.00149 Rad	-0.00155 Rad	0.000328 Rad	0.000378 Rad
ISE Sudut <i>pitch</i>	2.22 Rad	2.42 Rad	1.08 Rad	1.43 Rad
Sudut <i>yaw</i>	5.024437 Rad	5.070504 Rad	6.46 Rad	-0.02853 Rad
ISE Sudut <i>yaw</i>	25.24497 Rad	25.71001 Rad	4.17 Rad	0.000814 Rad
Kecepatan <i>X</i>	0.003765	0.003736	0.007936	0.007863
ISE Kecepatan <i>X</i>	1.42	1.4	6.3	6.18
Kecepatan <i>Y</i>	-0.00175	-0.00216	2.85	-0.00114
ISE Kecepatan <i>Y</i>	3.07	4.69	8.09	1.29
Kecepatan <i>Z</i>	0.015395	0.015254	0.001695	0.001662
ISE Kecepatan <i>Z</i>	0.000237	0.000233	2.87	2.76

*--Halaman ini sengaja dikosongkan--*



## BAB 5

### KESIMPULAN

#### 5.1 Kesimpulan

Tesis ini menyajikan strategi kontrol untuk menyelesaikan permasalahan *path following* pada quadcopter menggunakan kontrol optimal *output feedback*. Terdiri dari kontroler *inner loop* untuk menjaga kestabilan *attitude* dan kontroler *outer loop* untuk melakukan *path following* referensi posisi  $x$ ,  $y$ , dan  $z$ .

Metode kontrol *output feedback* dengan performa  $H_\infty$  digunakan pada *inner loop* sedangkan struktur kontrol *Command-Generator Tracker Model Following* dengan menambahkan algoritma *Line Of Sight* digunakan pada *outer loop*. Hasil simulasi menunjukkan bahwa sistem kontrol dapat menstabilkan sudut *attitude* dari quadcopter dan dapat membuat quadcopter mampu melakukan *path following* terhadap sinyal referensi yang diberikan.

Nilai maksimum ISE *path following* posisi  $X$  dan  $Y$  tanpa gangguan adalah 0.001 m dan pada saat ada gangguan sebesar 0.013168 m pada posisi  $X$  serta 0.0003 m pada posisi  $Y$ . Rata-rata nilai ISE dari sudut *yaw* untuk permasalahan arah hadap sebesar 0.024 Rad tanpa gangguan dan pada saat ada gangguan sebesar 0.0903 Rad. Kontroler ini mampu mengatasi efek *delay* dengan waktu  $\pm 3.17$  detik. Kontroler juga dapat membuat quadcopter mampu melakukan *path following* dengan referensi lintasan berbentuk persegi maupun spiral (*helix*).

#### 5.2 Saran

Untuk pengembangan penelitian selanjutnya, penulis menyarankan untuk permasalahan *path following* dengan menggunakan *output feedback* sebaiknya merumuskan kondisi dijadikan dalam dua persamaan sehingga, dapat mengurangi jumlah parameter yang dicari. Serta dapat mempertimbangkan permasalahan perubahan parameter baik kecepatan dan massa pada quadcopter pada waktu tertentu. menambah kontrol lain seperti kontrol adaptif yang diharapkan dapat mempermudah memperoleh gain  $K$  yang lebih akurat dibanding dengan *trial and error*.

*--Halaman ini sengaja dikosongkan--*

## DAFTAR PUSTAKA

- [1] David Cabecian, Rita Cunha, and Carlos Silvestre., “*A Globally Stabilizing Path Following Controller for Rotorcraft With Wind Disturbance Rejection*”, IEEE Transactions on Control Systems Technology Volume 23 issue 2, 2015
- [2] E. C. Suicmez and A. T. Kutay., “*Optimal path tracking control of a quadcopter UAV*”, International Conference on Unmanned Aircraft System, May 2014.
- [3] E. C. Vivas G, M. V. Pena G and C. I. Rodriguez F, "Simulation of The Quadrotor Controller with LQR with Integral Effect," ABCM Symposium Series in Mechatronics, vol. 5, pp. 390-399, 2012.
- [4] F. Yacef, O. Bouhali, H. Khebbache and F. Boudjema., “Takagi-Sugeno model for quadcopter modelling and control using *nonlinear state feedback controller*”, International Journal of Control Theory and Computer Modelling (IJCTCM), May 2012
- [5] Trihastuti Agustinah, Feni Isdaryani, Mohammad Nuh,” *Tracking Control of Quadrotor Using Static Output Feedback with Modified Command-Generator Tracker*,” International Review of Automatic Control (I.R.E.A.CO.), Vol. 9, N. 4, July,2016.
- [6] J. Gadewadikar, F. L. Lewis, K. Subbarao and B. M. Chen, "Design of  $H_{\infty}$  Command and Control Loops for Unmanned Aerial Vehicles using Static Output-Feedback," in Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, 2007
- [7] K. Ogata., *Modern Control Engineering*, Prentice-Hall, New Jersey, 1997
- [8] K. Khalil, *Nonlinear System* 3rd Edition, New Jersey: Prentice Hall, 2000
- [9] Frank. L. Lewis dan V. L. Syrmos., “*Optimal Control System – 2<sup>nd</sup> ed*”, John Wiley & Sons, Inc., Canada, 1995
- [10] J. Li, H. O. Wang and D. Niemann, "Relations Between LMI (Linear Matrix Inequality) and ARE (Algebraic Riccati Equatin) with their applications to Absolute Stability Criteria, Robustness Analysis and Optimal Control," 1998
- [11] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Philadelphia: Society for Industrial and Applied Mathematics, 1994

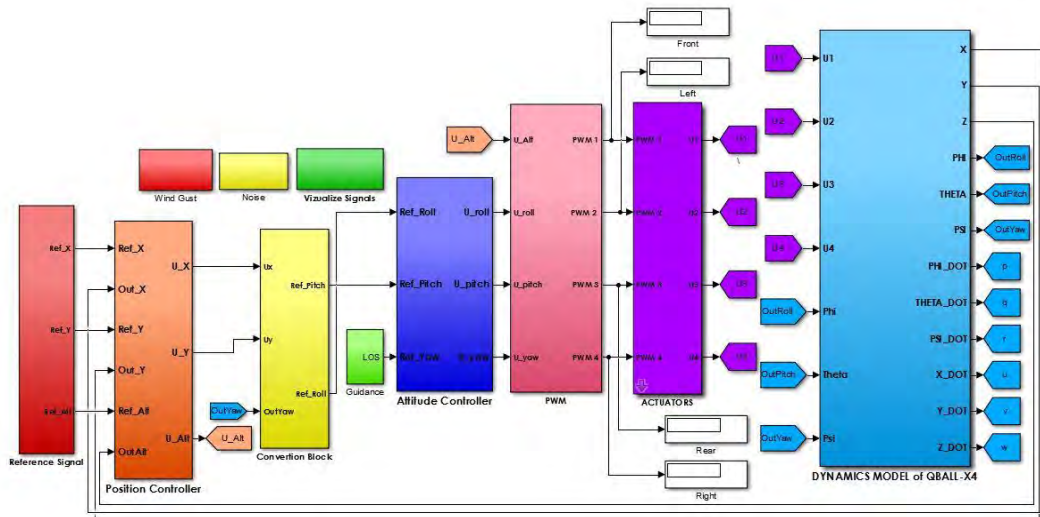
- [12] “Unmanned Flight”, IEEE *Control System Magazine Volume 34 number 1*, February 2014
- [13] Antonio Tsourdos, Brian A. White and madhavan Shanmugavel., “Cooperative *path* planning of Unmanned Aerial Vehicles”, John Wiley & Sons, Ltd, 2011
- [14] ..., Quanser Q-Ball X-4 User Manual, Quanser Innovate Educate, 2010, Doc. 888 Rev.2 pp. i-46.
- [15] M. M. de Almeida Neto., “Control strategies of a tilt-rotor UAV for load transportation”, Master Thesis, Universidade Federal de Minas Gerais, August 2014
- [16] Frank. L. Lewis, B. L. Stevens dan Eric N. Johnson., “*Aircraft Control and Simulation – 3<sup>nd</sup> ed*”, John Wiley & Sons, Inc., Canada, 2016
- [17] Thor I. Fossen., “Handbook of Marine Craft Hydrodynamics and Motion Control”, John Wiley & Sons, Inc., Canada, 2011
- [18] V. Solovyev Viktor, I. Finaev Valery, A. Zargaryan Yuri, O. Shapovalov Igor and A. Beloglazov Denis, "Simulation of Wind Effect on A Quadrotor Flight," *ARPN Journal Engineering and Apllied Sciences*, vol. 10, pp. 1535-1538, 2015
- [19] Isdaryani, Feni,,” Kontrol *Tracking* untuk Quadrotor Menggunakan Metode *Optimal Output Feedback* dengan *Command generator tracker*,” Tesis Master, Institut Teknologi Sepuluh Nopember,2016.

# LAMPIRAN

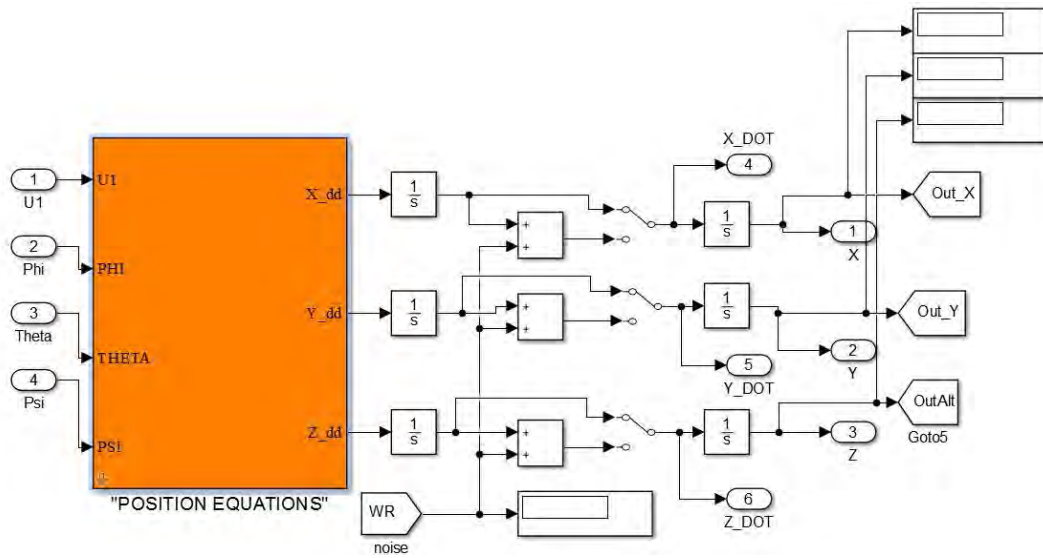
- LAMPIRAN A

## Blok Diagram Simulink

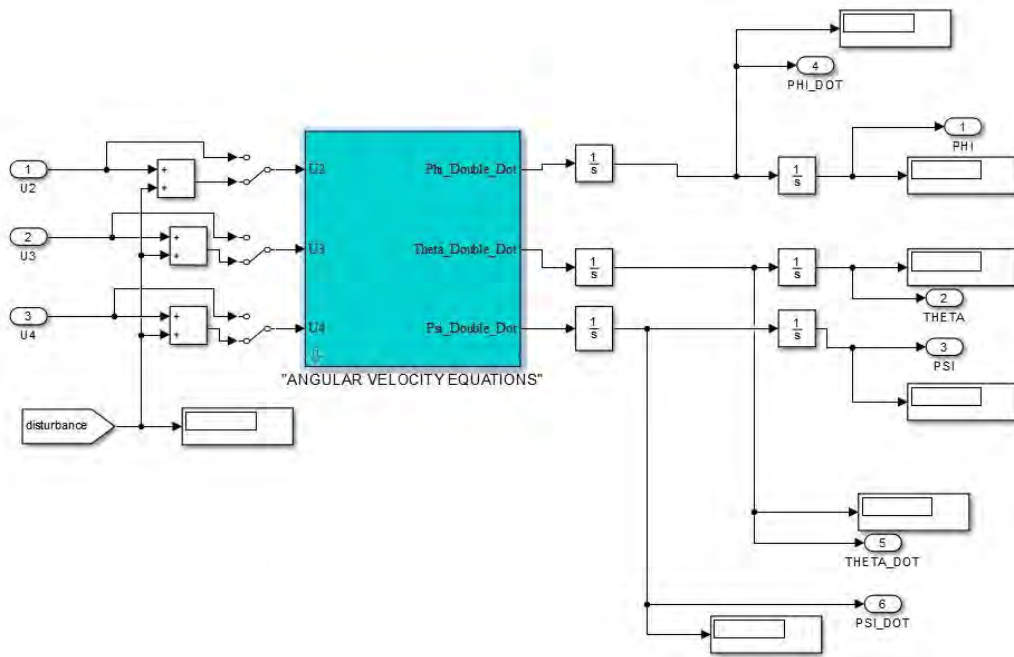
### 1. Desain Kontroler Secara Keseluruhan



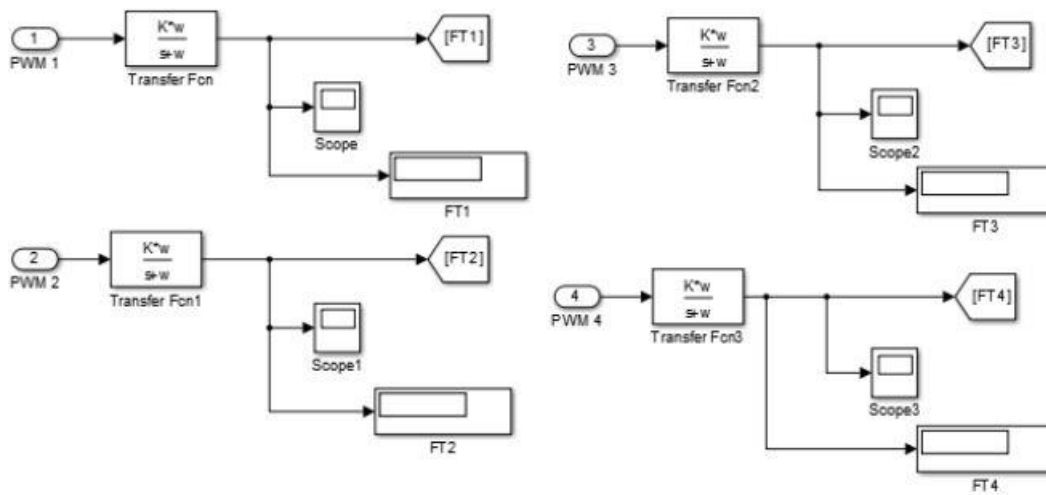
### 2. Blok Dinamika Translasi



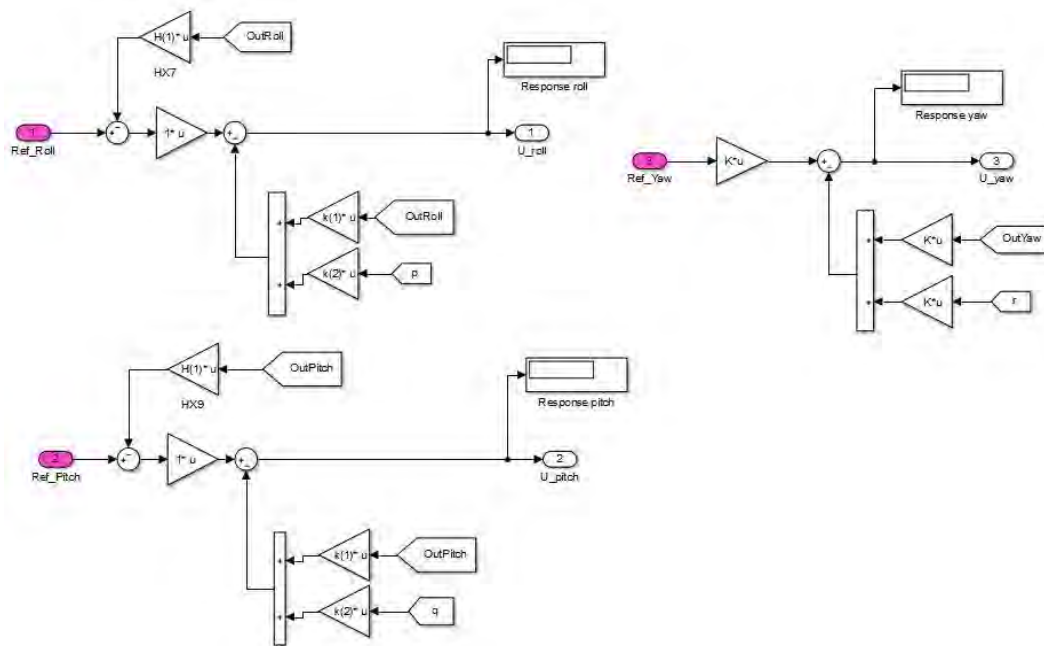
### 3. Blok Dinamika Rotasi



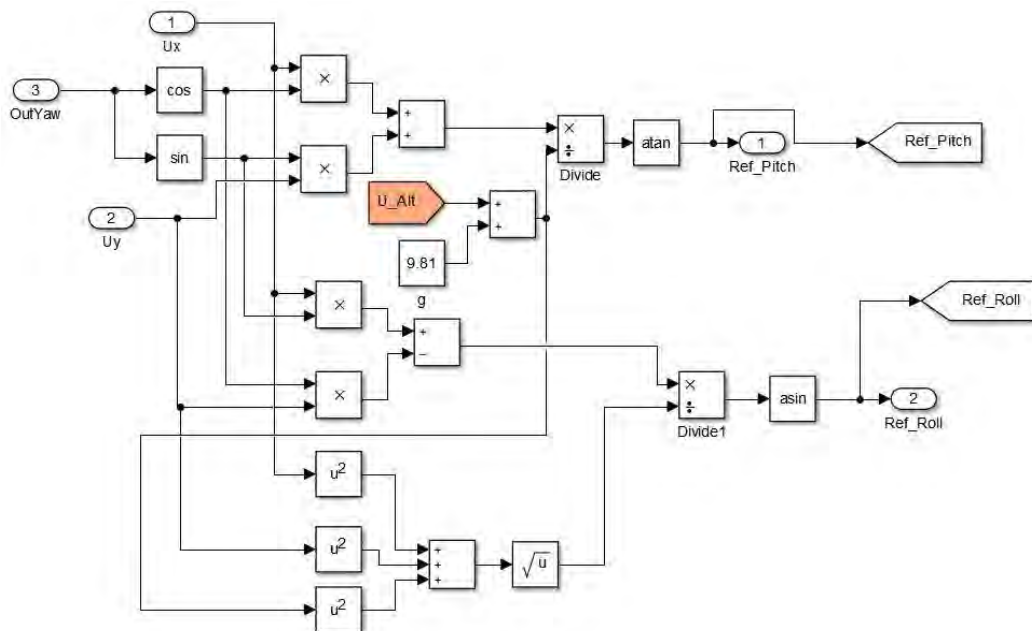
### 4. Blok Model Aktuator



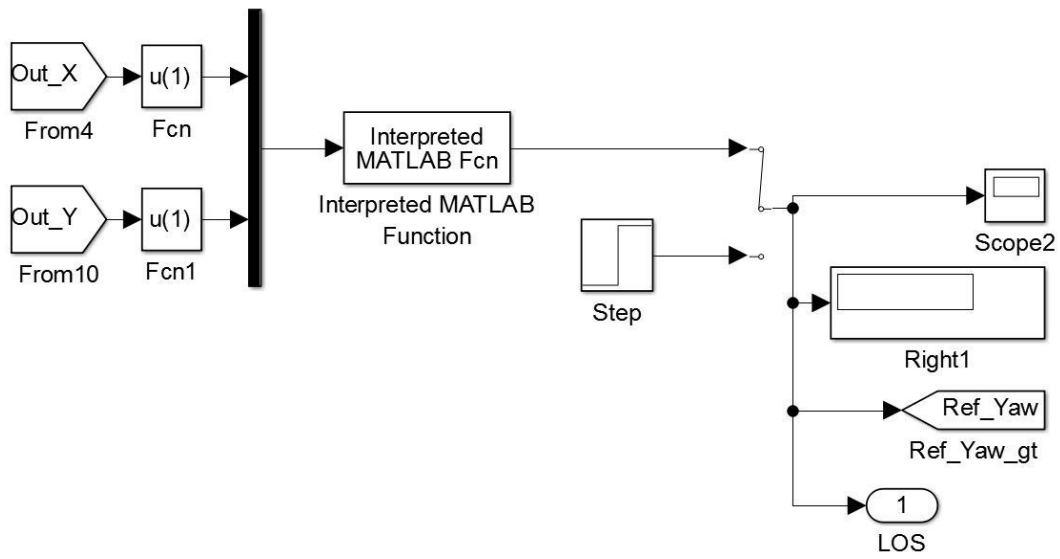
## 5. Blok Kontroler Roll, Pitch dan Yaw



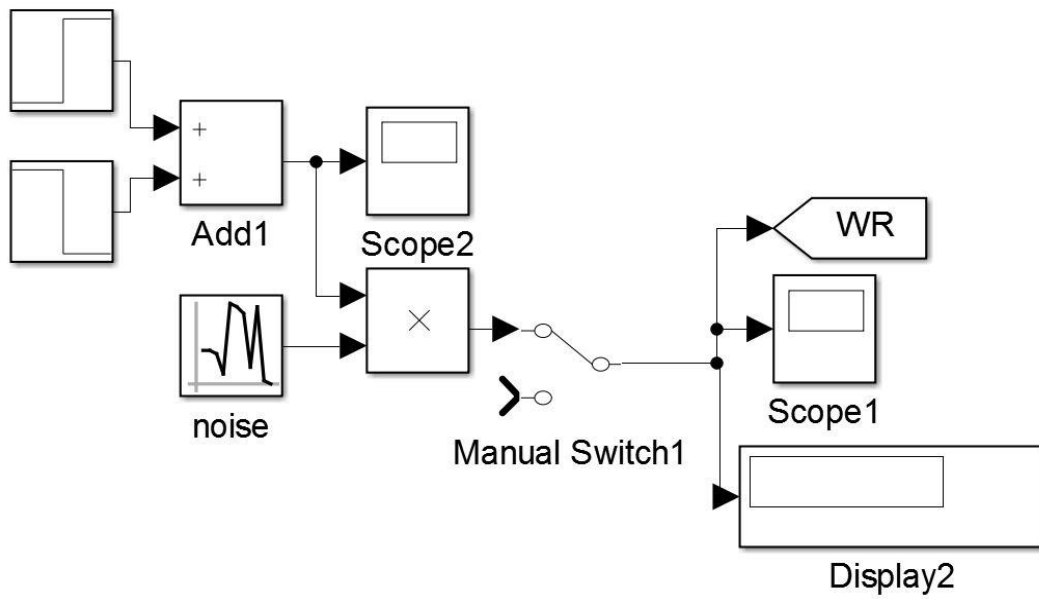
## 6. Blok Koreksi Sinyal Kontrol



### 7. Blok Algoritma LOS

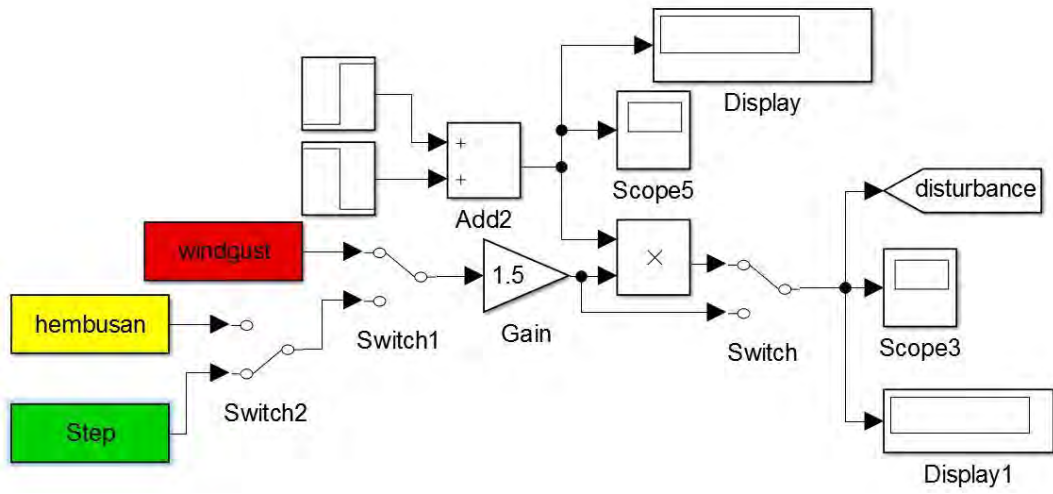


### 8. Blok Pemodelan Noise

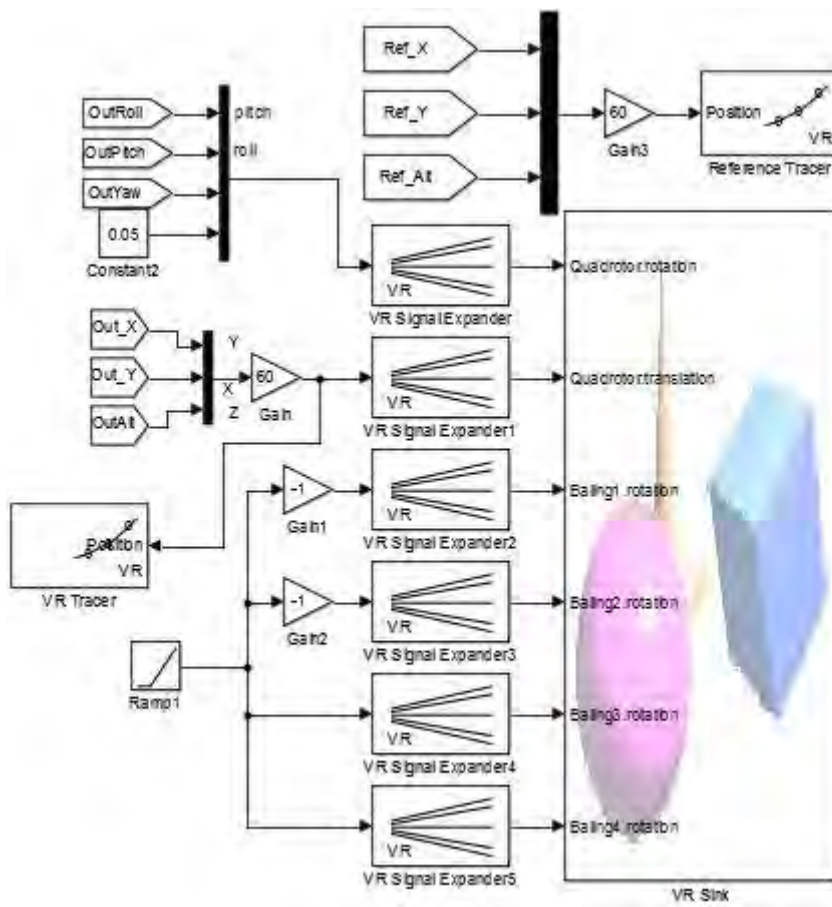




### 9. Blok Pemodelan *Wind Gust*



### 10. Blok Virtual Reality



- **LAMPIRAN B**

Pemograman

Program *Iterative* LMI dengan performa  $H_\infty$  pada permasalahan Stabilisasi

```
function
K=Hitung_OPFB_Coba(Am,Bm,Cm,Bwm,Qm,Rm,N,n,gam,toler,H,Ppos)
%inisialisasi %program mencari H-inf OPFB tanpa inisial Gain
% n = 0;
L0 = [0.1 .1 .1];
Ln = [0 0 0]; %Jangan Ln namanya, Ln mah nama Gain itu rumus Ln
ada di baris 43
K = 0;
Kn = 0;
a=0;
toler = 0.01; %nilai epsilon(toleransi)
gam = 0.7;
Rr = inv(Rm);

%for I=1:N,
while abs(Ln(1)-L0(1))>=toler || abs(Ln(2)-L0(2))>=toler ||
abs(Ln(3)-L0(3))>=toler
a=a+1;
L0 = Ln;
K=Kn;

%Insialisasi LMI 1

setlmis([]);
Y=lmivar(1,[3 1]);

lmiterm([1 1 1 Y],1,-Am,'s');
lmiterm([1 1 1 0],-Qm);
lmiterm([1 1 1 0],-L0'*Rr*L0);
lmiterm([1 2 1 Y],Bm',1);
lmiterm([1 2 2 0],-Rm);
lmiterm([1 3 1 Y],Bwm',1);
lmiterm([1 3 3 0],-(gam^2));

lmiterm([-2 1 1 Y],1,1); %nilai Y>0

lmis=getlmis;
[tm,xf]=feasp(lmis);

%Hasil matriks P
Y=dec2mat(lmis,xf,Y);
P=Y
Kn = (inv(Rm))*(Bm'*P+L0)*Cm'*(inv(Cm*Cm')); %Gain Kn
Ln = Rm*Kn*Cm-(Bm'*P) %Gain Ln
L0;
end
K = Kn;
```

## Program Kontroler *Altitude* dan *Attitude*

```

%titik kerja
phi = 0.2;
tetha = 0.2;
psi = 0.2;
%parameter quadcopter
m = 3.499; l=0.225; g=9.81; Kmt=120; w=15;
d = 3.23*(10^-7); b=2.98*(10^5); Jr=2.84*(10^-5);
Ix = 0.03; Iy = 0.03; Iz=0.04;
% ===== Mencari Gain Roll dan Pitch =====
Am      = [0 1 0;0 0 (Kmt*l)/Ix;0 0 -15];
Bm      = [0 0 15]';
H = [0.3 0 0]; %performansi output
Cm = [1 0 0;0 1 0];
Bwm = Bm;
%Cek controllability dan Observability dari sistem open loop
ccx = ctrb(Am,Bm);
controllability = rank(ccx);
Cob = obsv(Am,Cm);
unob = length(Am)-rank(Cob);
observability = rank(Cob);
nilai_eigen= eig(Am);
%Inisialisasi
N = 100; K0=0; L0=0;
gam = 0.7;
Rm = 10;
Qm =[505 0 0;
      0 3 0;
      0 0 .1];
% nilai epsilon(toleransi)
toler = 0.1;
k = Hitung_OPFB_Coba(Am,Bm,Cm,Bwm,Qm,Rm,N,gam,toler,L0,H)
Ei=Am-Bm*k*Cm;
ei=eig(Ei)
% condition for convergen
AQob = obsv(sqrt(Qm),Am);
AQunob = length(Am)-rank(AQob);
AQobservability = rank(AQob)
% ===== Menghitung Gain Yaw =====
Ky      = 4;
Jy      = 0.04;
Amy     = [0 1
           0 0];
Bmy     = [0 4*Ky/Jy]';
Cmy     = [1 0];
Dmy     = [0];
co = ctrb(Amy,Bmy);
controllability = rank(co);
Qy      = diag([1 3]); %[4 3]
Ry      = 1;

[Kmy] = lqr(Amy,Bmy,Qy,Ry) ; %gain feedback
Nbar = rscale(Amy,Bmy,Cmy,Dmy,Kmy) ; %gain precompensator

```

```

%===== Menghitung Gain Z =====
Az = [0 1 0;0 0 0;1 0 0];
Bz = [0 4*Kmt/m 0]';
Cz = eye(3);
Qz = diag([0.5 1 100]);
% Qz = diag([.5 5 100]);
Rz = 1;
[Kz] = lqr(Az,Bz,Qz,Rz);
%Matriks gain state-feedback K
Kzi = Kz(3);
Kzz=[Kz(1) Kz(2)];

```

### Kontrol *Path following* sumbu $X$

```

%parameter quadcopter
m = 3.522462; l=0.225; g=9.81; Kmt=120; w=15;
d = 3.23*(10^-7); b=2.98*(10^5); Jr=2.84*(10^-5);
Ix = 0.03; Iy = 0.03; Iz=0.04;
%Matriks Sistem posisi X dan Y
Ax = [0 1 0;0 0 (4*Kmt/m)*0.2;0 0 -w];
Bx = [0 0 w]';
Hx = [1 0 0]; %performansi output
%matriks referensi
Arx = [0 1;-0.02 0];
hb1 = [1 0];
%Augmented sistem
Brx = [zeros(1,3);-Hx];
Agx = [Arx Brx;zeros(3,2) Ax];
Bgx = [0;0;Bx];
Cgx = [1 0 0 0 0;0 1 0 0 0;0 0 1 0 0;0 0 0 1 0.0001];
R0x = .1;
Q0x=diag([10 10 5 10 0.1]);
[kox]=lqr(Agx,Bgx,Q0x,R0x)
k0x = [kox(1) kox(2) kox(3) kox(4)]
% inisialisasi
v=1;
tolerx=0.01;
alphax=.1;
Nx=100;
r0x =1;
Rx = 1;
Qx=diag([250 300 0.1 500 10]);
kx=optk2(Agx,Bgx,Cgx,k0x,Rx,Qx,Nx,tolerx,alphax)
Acx=Agx-Bgx*kx*Cgx; e=eig(Acx)
ob_posX = obsv(Agx,sqrt(Qx));
unob_posX = length(Agx)-rank(ob_posX);
observability_posX = rank(ob_posX)

```

## Kontrol *Path following* Sumbu *Y*

```
%parameter quadcopter
m = 3.522462; l=0.225; g=9.81;
d = 3.23*(10^-7); b=2.98*(10^5); Jr=2.84*(10^-5);
Kmt = 120; w =15;
Ix = 0.03; Iy = 0.03; Iz=0.04;
%Matriks Sistem posisi X dan Y
Ay = [0 1 0;0 0 -4*Kmt/m;0 0 -w];
By = [0 0 w]';
Hy = [1 0 0]; %performansi output
%matriks referensi
Ary = [0 1;0 -0.02];
%Augmented system
Bry = [zeros(1,3);-Hy];
Agy = [Ary Bry;zeros(3,2) Ay];
Bgy = [0;0;By];
Cgy = [1 0 0 0 0;0 1 0 0 0;0 0 1 0 0;0 0 0 1 0.0001];
%Matriks Pembobot
R0y = 1;
Q0y =diag([100 10 5 100 10]);
[koy]=lqr(Agy,Bgy,Q0y,R0y);
k0y=[koy(1) koy(2) koy(3) koy(4)]
% inisialisasi
v=1;
tolery=0.001;
alphay=.1;
Ny=100;
r0y =1;
Ry = 1;
Qy=diag([400 300 25 40 1]);
ky=optk2_Y(Agy,Bgy,Cgy,k0y,Ry,Qy,Ny,tolery,alphay)*3
Acy=Agy-Bgy*ky*Cgy; e=eig(Acy)
```

## Pemodelan *Disturbance* Angin konstan (*Step*)

```
%produceWind
tt=0.05;
tf=62.5;
time=(0:tt:tf)';
Step.time=time;
%---STEP---%
teph=zeros(1,length(time));
ts0=[0 9 15 19 30 35 45 47 50 59];
Vs=[1 3.5 -4.5 1 4 -3 -2 1 -1 4];
ds=[7 5 2 5 3 8 4 1 10 15];
for i=1:length(ts0)
    for j=1:length(time)
        t=time(j);
        if t>=ts0(i) && t<=ts0(i)+ds(i)
            teph(j)=teph(j)+Vs(i)/2*(1-cos(pi*(t-
ts0(i))/ds(i)));
```

```

        elseif t>ts0(i)+ds(i)
            teph(j)=teph(j)+Vs(i);
        end
    end
end
Step.signals.values=(teph)';
Step.signals.dimensions=1;

```

### Pemodelan *Disturbance Gust*

```

%produceWind
tt=0.05;
tf=62.5;
time=(0:tt:tf)';
hembusan.time=time;
%---GUST---%
gust=zeros(1,length(time));
t0=[0 9 16 19 23 27 33 38 40 49 51 53 61];
Vm=[3 4.5 0.75 1 2 4 -5 3 2.5 0.4 -2 0.25 4];
dm=[0.5 0.75 2 1 0.25 .5 .4 .3 1.1 .2 .5 1.25 0.2];
for i=1:length(t0)
    for j=1:length(time)
        t=time(j);
        if t>=t0(i) && t<=t0(i)+dm(i)
            gust(j)=gust(j)+Vm(i)*sin(pi*(t-t0(i))/dm(i));
        end
    end
end
hembusan.signals.values=(gust)';
hembusan.signals.dimensions=1;

```

### Pemodelan *Disturbance Modifikasi (Gust+Step)*

```

%produceWind
tt=0.05;
tf=62.5;
time=(0:tt:tf)';
windgust.time=time;
%---GUST---%
gust=zeros(1,length(time));
t0=[0 9 16 19 23 27 33 38 40 49 51 53 61];
Vm=[3 4.5 0.75 1 2 4 -5 3 2.5 0.4 -2 0.25 4];
dm=[0.5 0.75 2 1 0.25 .5 .4 .3 1.1 .2 .5 1.25 0.2];
for i=1:length(t0)
    for j=1:length(time)
        t=time(j);
        if t>=t0(i) && t<=t0(i)+dm(i)
            gust(j)=gust(j)+Vm(i)*sin(pi*(t-t0(i))/dm(i));
        end
    end
end
%---STEP---%
teph=zeros(1,length(time));

```

```

ts0=[0 9 15 19 30 35 45 47 50 59];
Vs=[1 3.5 -4.5 1 4 -3 -2 1 -1 4];
ds=[7 5 2 5 3 8 4 1 10 15];
for i=1:length(ts0)
    for j=1:length(time)
        t=time(j);
        if t>=ts0(i) && t<=ts0(i)+ds(i)
            teph(j)=teph(j)+Vs(i)/2*(1-cos(pi*(t-
ts0(i))/ds(i)));
        elseif t>ts0(i)+ds(i)
            teph(j)=teph(j)+Vs(i);
        end
    end
end
end
windgust.signals.values=(gust+teph)';
windgust.signals.dimensions=1;

```

• LAMPIRAN C

Daftar penelitian berkaitan quadcopter yang telah dilakukan di Institut Teknologi Sepuluh Nopember Departemen Teknik Elektro Bidang Keahlian Teknik Sistem Pengaturan.

No	Tesis	
1.	Nama	Jumiyatun
	Tahun	2013
	Judul	PENGATURAN KESTABILAN <i>HOVER</i> PADA QUADROTOR MENGGUNAKAN KONTROL <i>BACKSTEPPING</i>
	Rumusan masalah	Bagaimana merancang sistem kontrol quadrotor yang stabil pada kondisi <i>hover</i> dan mampu mengatasi adanya gangguan.
	Metode kontrol	<i>Backstepping</i>
	Kelebihan	<ul style="list-style-type: none"> <li>• Mampu melakukan gerak <i>hover</i> dengan tingkat <i>error</i> sebesar 0.0156m</li> <li>• Mampu menjaga kestabilan pada orientasinya.</li> </ul>
	Kekurangan	<ul style="list-style-type: none"> <li>• Terdapat <i>overshoot</i> yang besar baik <i>roll</i> dan <i>pitch</i> sebelum mencapai <i>set point</i>.</li> <li>• Pada saat diberi gangguan <i>overshoot</i> yang didapat cukup besar pada sudut <i>roll</i> dan sudut <i>yaw</i>.</li> </ul>
	Saran	<ul style="list-style-type: none"> <li>• Memahami benar pemodelan dan <i>plant</i>nya sehingga akan lebih mudah dalam menentukan metode yang digunakan.</li> <li>• Meningkatkan kontrol <i>backstepping</i> ini dengan menambah kontrol lain seperti kontrol adaptif yang diharapkan dapat memperoleh parameter <math>\alpha</math> yang lebih akurat dibanding dengan <i>trial</i> dan <i>error</i>.</li> </ul>
No	Tesis	
2.	Nama	RUDY KURNIAWAN
	Tahun	2013
	Judul	PERANCANGAN KONTROL NON-LINIER <i>SLIDING MODE</i> UNTUK MENJAGA POSISI KETINGGIAN <i>HOVER</i> PADA UAV TRICOPTER
	Rumusan masalah	Bagaimana merancang desain kontrol tricopter yang mampu mengatasi momen pada sudut <i>yaw</i> dan mampu menjaga kestabilan dalam kondisi <i>hover</i> .
	Metode kontrol	<ul style="list-style-type: none"> <li>• <i>Sliding mode control</i></li> <li>• PID</li> </ul>
	Kelebihan	Mampu meminimalisir adanya gangguan serta menjaga kestabilan saat melakukan <i>hover</i>
	Kekurangan	<ul style="list-style-type: none"> <li>• Masih terdapat osilasi pada sudut <i>pitch</i></li> <li>• Masih membutuhkan waktu hingga 8 detik untuk mencapai <i>set point</i> yang berikan</li> </ul>
	Saran	Adapun saran yang bisa diberikan adalah sistem kontrol yang digunakan masih bisa dikembangkan lagi untuk melakukan pengendalian kestabilan sudut <i>roll</i> , <i>pitch</i> dan <i>yaw</i> dengan menggunakan metode SMC. Selanjutnya, hasil perancangan kontroler SMC dapat diujikan dengan cara mengimplementasikan kontroler dengan <i>plant</i> UAV Tricopter dan dilakukan pengujian secara <i>real time</i> . Penggunaan metode SMC dikombinasikan dengan metode kontrol yang lainnya sebaiknya dilakukan untuk keberlanjutan penelitian ini.



No	Tesis	
3.	Nama Tahun Judul Rumusan masalah Metode kontrol Kelebihan Kekurangan Saran	Nofria Hanafi 2014 PENDARATAN OTOMATIS QUADCOPTER PADA <i>PLATFORM</i> YANG BERGERAK MENGGUNAKAN NEURO FUZZY Bagaimana merancang quadcopter mampu melakukan <i>landing</i> dengan stabil pada saat ada atau tidaknya gangguan serta <i>platform</i> yang bergerak Neuro Fuzzy Penggunaan kontroler PD mampu menstabilkan gerak <i>roll</i> dan <i>pitch</i> hingga 0° Masih membutuhkan waktu hingga 10 detik dalam melakukan pendaratan <ul style="list-style-type: none"> <li>• Perlu dikembangkan metode <i>safety landing</i> yang lebih akurat dengan ditambahkan kamera sebagai pengunci tempat pendaratan.</li> <li>• Pembacaan sensor untuk pendaratan di tempat yang bergerak ritmik sebaiknya dikirimkan langsung dari tempat pendaratan.</li> </ul>
No	Tesis	
4.	Nama Tahun Judul Rumusan masalah Metode kontrol Kelebihan Kekurangan Saran	Mokhamad Khoizin 2014 <i>WAYPOINT TRACKING CONTROL</i> PADA <i>MICRO</i> QUADROTOR UNMANNED AERIAL VEHICLE (UAV) MENGGUNAKAN <i>BLOCK BACKSTEPPING CONTROL</i> DENGAN <i>ADAPTIVE PARAMETER APPROXIMATION</i> Bagaimana merancang struktur kontrol <i>micro</i> quadcopter dalam melakukan <i>waypoint tracking</i> yang stabil dan mampu mengatasi nonlinearitas dan adanya gangguan eksternal <ul style="list-style-type: none"> <li>• <i>Backstepping</i></li> <li>• <i>Adaptive backstepping</i></li> </ul> Kelebihan <ul style="list-style-type: none"> <li>• Menggunakan <i>block backstepping control</i> dengan <i>adaptive parameter approximation</i> sebagai kontroler, pergerakan <i>micro quadrotor</i> cukup baik walaupun terdapat gangguan eksternal. Metode kontrol ini mampu mengatasi adanya gangguan eksternal sehingga mencegah adanya deviasi posisi yang besar dari <i>micro quadrotor</i> terhadap posisi yang diharapkan</li> </ul> Kekurangan <ul style="list-style-type: none"> <li>• Masih membutuhkan waktu hingga 4 detik untuk mencapai <i>set point</i> yang diinginkan</li> <li>• Pada saat diberi gangguan eksternal berupa <i>step</i>, <i>plant</i> masih belum mampu meminimalisir adanya gangguan tersebut</li> </ul> Saran -
No	Tesis	
5.	Nama Tahun Judul Rumusan masalah Metode kontrol	Swadexi Istiqphara 2015 PERANCANGAN SISTEM <i>WAYPOINT TRACKING CONTROL</i> PADA QUADROTOR MENGGUNAKAN <i>SELF TUNING SLIDING MODE CONTROL</i> Bagaimana merancang quadrotor mampu melakukan <i>waypoint tracking</i> yang stabil dan tahan terhadap gangguan serta mampu mengatasi perubahan sistem <ul style="list-style-type: none"> <li>• <i>Sliding mode Control</i></li> <li>• <i>Self tuning Sliding mode Control</i></li> </ul>

		<ul style="list-style-type: none"> <li>• Complementary Filter ( Highpass+Lowpass)</li> <li>• Least square Parameter Estimator</li> </ul>
Kelebihan		<ul style="list-style-type: none"> <li>• Dalam mengatasi perubahan parameter sistem saat melakukan <i>tracking waypoint</i>, metode <i>Self Tuning Sliding mode Control</i> mampu melakukan perubahan parameter <i>gain K</i> dengan cepat.</li> <li>• Pengaruh perubahan parameter massa pada saat melakukan <i>waypoint tracking</i> mempengaruhi perubahan nilai gain pada kontroler sumbu Z. Hal ini menunjukkan perubahan nilai massa quadrotor tidak mempengaruhi geraktranslasi sumbu X dan sumbu Y.</li> </ul>
Kekurangan		<ul style="list-style-type: none"> <li>• Terdapat fenomena chattering pada permukaan luncur (Sliding Surface)</li> <li>• Membutuhkan waktu cukup lama untuk mencapai <i>set point</i></li> <li>• Pada saat adanya perubahan parameter pada beban <i>controller</i> masih terdapat fenomena chattering serta belum mampu mencapai <i>set point</i> yang diberikan</li> </ul>
Saran		Metode kontrol yang digunakan hanya diterapkan pada perangkat simulasi, untuk penelitian selanjutnya diharapkan dapat dilakukan implementasi untuk menguji kualitas metode <i>control Self Tuning Sliding mode Control</i> pada quadrotor

No	Tesis	
6.	Nama	Hanum Arrosida
	Tahun	2015
	Judul	PERANCANGAN NONLINEAR DECOUPLING DAN NONLINEAR PD UNTUK KESTABILAN DAN PENGENDALIAN GERAK CRUISE PADA QUADROTOR
	Rumusan masalah	Bagaimana merancang sistem <i>control</i> quadcopter yang mampu melakukan gerak cruise dengan stabil pada trayektori yang ditentukan.
	Metode kontrol	Nonlinear PD Nonlinear decoupling ditambah dengan kontroler PID
Kelebihan		<ul style="list-style-type: none"> <li>• Pengaruh metode <i>control</i> nonlinear PD berdasarkan fungsi invers model dari sinyal <i>control plant</i> mampu memperbaiki performansi kontroler dengan metode nonlinear PD berdasarkan dari karakteristik <i>error</i> sistem</li> <li>• Mampu merubah sistem yang coupling menjadi decoupling</li> </ul>
Kekurangan		<ul style="list-style-type: none"> <li>• Pada saat melakukan trayektori dalam bidang persegi masih terdapat <i>overshoot</i>.</li> <li>• Membutuhka waktu sebesar 3detik untuk mencapai <i>set point</i> yang ditentukan</li> </ul>
Saran		Mekanisme pengendalian arah atau orientasi <i>quadrotor</i> dan uji <i>robustness</i> dari kontroler yang telah diterapkan.

No	Tesis	
7.	Nama	Feni Isdaryani
	Tahun	2016
	Judul	KONTROL <i>TRACKING</i> UNTUK QUADROTOR MENGGUNAKAN METODE OPTIMAL OUTPUT <i>FEEDBACK</i> DENGAN COMMAND-GENERATOR TRACKER
	Rumusan masalah	Bagaimana merancang struktur kontrol <i>tracking</i> untuk quadrotor yang dapat memperoleh kesesuaian <i>tracking</i> sambil menjaga stabilitas <i>attitude</i> walau terdapat gangguan eksternal
	Metode kontrol	<ul style="list-style-type: none"> <li>• Metode kontrol <i>output feedback</i> dengan performa <math>H_{\infty}</math> digunakan pada <i>inner loop</i></li> <li>• Struktur kontrol <i>Command-Generator Tracker</i> digunakan pada <i>outer loop</i>.</li> </ul>

Kelebihan	<ul style="list-style-type: none"> <li>• LMI (<i>Linear Matrix Inequality</i>)</li> <li>• LQR (<i>Linear Quadratic Regulator</i>)</li> </ul>
Kekurangan	<ul style="list-style-type: none"> <li>• Dapat membuktikan sistem kontrol <i>command-generator tracking</i> dapat melakukan <i>tracking</i> tanpa memperdulikan adanya perubahan orde pada integrator</li> <li>• Sistem kontrol dapat menstabilkan sudut attitude dari quadrotor dan dapat membuat quadrotor men-<i>track</i></li> <li>• Quadrotor melakukan <i>tracking</i> dengan referensi lintasan berbentuk persegi dan spiral (<i>helix</i>).</li> </ul>
Saran	<ul style="list-style-type: none"> <li>• Terdapat <i>delay</i> sebesar 0.05 detik pada saat track sumbu X</li> <li>• Masih terdapat <i>overshoot</i> pada saat awal dibangkitkan <i>windgust</i> dan ditiadakannya <i>windgust</i>.</li> <li>• Terdapat <i>overshoot</i> pada <i>tracking</i> lintasan persegi pada sudut 90 derajat</li> </ul> <p>Merumuskan kondisi perlu dari permasalahan <i>tracking</i> dengan kontrol <i>optimal output feedback</i> menjadi dua persamaan saja sehingga mengurangi jumlah parameter yang perlu dicari. Serta dapat mempertimbangkan permasalahan <i>control</i> orientasi pada quadrotor</p>

No	Tesis	
----	-------	--

8.	Nama	Anisa Ulya Darajat
	Tahun	2016
	Judul	KOMPENSASI KESALAHAN AKTUATOR QUADROTOR PADA <i>CONTROL ASSEMBLER</i> MENGGUNAKAN MODIFIED LEAST SQUARE
	Rumusan masalah	Bagaimana merancang algoritma kontrol toleransi kesalahan berbasis estimasi dan kompensasi kesalahan <i>actuator</i> (rotor) pada quadrotor yang menyebabkan quadrotor tidak dapat mengikuti sinyal referensi.
	Metode kontrol	<ul style="list-style-type: none"> <li>• <i>Backstepping</i></li> <li>• Least square</li> </ul>
	Kelebihan	Metode <i>modified least square</i> mampu melakukan perubahan parameter <i>gain</i> $\eta$ dengan cepat
	Kekurangan	Masih membutuhkan waktu dalam mencapai <i>set point</i> yang sudah ditentukan
	Saran	Diharapkan dapat dilakukan implementasi untuk menguji kualitas metode kontrol modified least square pada quadrotor

No	Tesis	
----	-------	--

9.	Nama	Mohammad Nur
	Tahun	2017
	Judul	KONTROL <i>TRACKING</i> PADA QUADROTOR MENGGUNAKAN <i>NONLINEAR QUADRATIC TRACKING</i> DENGAN <i>EXTENDED KALMAN FILTER</i>
	Rumusan masalah	Bagaimana mendesain Kontrol NLQT dengan matriks B pada sistem translasi tetap dipertahankan dalam bentuk <i>nonlinear</i> namun tetap stabil dalam melakukan <i>tracking</i> .
	Metode kontrol	<ul style="list-style-type: none"> <li>• <i>Linear Quadratic Tracking</i></li> <li>• Extended kalman Filter</li> </ul>
	Kelebihan	Mampu mempertahankan nilai nonlinear pada matrik B pada permasalahan Nonlinear quadratic <i>tracking</i> Dapat melakukan <i>tracking</i> dengan baik sesuai trajectory yang diinginkan

Kekurangan	Masih terdapat <i>overshoot</i> dan membutuhkan waktu untuk kembali ke nol. Dapat terlihat pada saat quadcopter berpindah dari titik satu ke titik berikutnya.
Saran	<ul style="list-style-type: none"> <li>• Penggunaan Metode NLQT bisa digunakan pula pada system rotasi</li> <li>• Uji coba pada gangguan eksternal belum dilakukan</li> </ul>

No	Tesis	
----	-------	--

10.	Nama	Chalidia Nurin Hamdani
	Tahun	2017
	Judul	PERANCANGAN AUTONOMOUS VTOL PADA QUADCOPTER DENGAN MENGGUNAKAN <i>FEEDBACK LINEARIZATION</i> DAN FUZZY TAKAGISUGENO
	Rumusan masalah	bagaimana merancang sebuah sistem kontrol yang mampu memenuhi kriteria untuk sistem gerak rotasi dapat distabilkan di 0 rad dengan <i>settling time</i> yang cepat, dan stabil secara semi-global serta sistem gerak translasi sumbu $z$ ( <i>altitude</i> ) dapat mengikuti sinyal referensi yang diberikan, stabil secara semi-global dan mampu mengatasi gangguan terbatas yang diberikan dengan nilai kesalahan <i>tracking</i> yang sekecil mungkin.
	Metode kontrol	<ul style="list-style-type: none"> <li>• <i>Feedback Linearization</i></li> <li>• Fuzzy takagi sugeno</li> <li>• Performa <math>H_\infty</math></li> <li>• <i>Linear Matrix Inequality</i> (LMI)</li> </ul>
	Kelebihan	Mampu melemahkan adanya <i>noise</i> yang diberikan pada gerak orientasi dan mampu menjaga kestabilan <i>altitude</i>
	Kekurangan	Masih membutuhkan waktu hingga 7 detik dan terdapat <i>overshoot</i> saat diberi gangguan dari luar pada gerak <i>altitude</i>
	Saran	penggabungan sistem kontrol gerak translasi $x$ , $y$ dan $z$ menggunakan metode linierisasi state dependent dengan tetap menggunakan $H_\infty$ sehingga sistem gerak translasi dapat terjamin kekokohannya secara keseluruhan. Kemudian dengan membuat kontrol <i>tracking</i> pada sistem gerak rotasi, maka quadcopter dapat melakukan <i>aerobatic take off and landing</i> . Selain itu, dapat dipertimbangkan juga mengenai penambahan kontrol orientasi.

No	Tesis	
----	-------	--

11.	Nama	Anggara Trisna Nugraha
	Tahun	2017
	Judul	Desain Kontrol <i>Path following</i> Quadcopter menggunakan output <i>feedback</i> dengan command generator tracker model following
	Rumusan masalah	Bagaimana merancang stuktur kontrol untuk quadcopter yang dapat melakukan <i>path following</i> terhadap <i>path</i> lingkaran walau terdapat gangguan eksternal
	Metode kontrol	<ul style="list-style-type: none"> <li>• Performa <math>H_\infty</math></li> <li>• <i>Linear Matrix Inequality</i> (LMI)</li> <li>• <i>Line Of Sight</i></li> <li>• <i>Static Output Feedback</i></li> <li>• <i>Command Generator Tracker Model Following</i></li> </ul>
	Kelebihan	<ul style="list-style-type: none"> <li>• Mampu melakukan <i>path following</i> dalam path berbentuk helix, persegi dan lingkaran</li> <li>• Mampu menjaga arah hadap dengan penambahan algoritma <i>Line Of Sight</i> pada sudut <i>yaw</i></li> </ul>

Kekurangan	<ul style="list-style-type: none"><li>• Mampu melemahkan adanya gangguan dan <i>noise</i> yang diberikan.</li><li>• Masih terdapat <i>delay</i> sebesar 0.046 detik pada posisi X</li><li>• Terdapat <i>overshoot</i> pada saat melakukan <i>path following</i> terhadap lintasan persegi</li><li>• Membutuhkan waktu sebesar 4 detik untuk mencapai <i>set point</i> yang diberikan pada sudut <i>yaw</i></li></ul>
Saran	Permasalahan <i>path following</i> dengan menggunakan <i>output feedback</i> sebaiknya merumuskan kondisi dijadikan dalam dua persamaan sehingga, dapat mengurangi jumlah parameter yang dicari. Serta dapat mempertimbangkan permasalahan perubahan parameter baik kecepatan dan massa pada quadcopter pada waktu tertentu. menambah kontrol lain seperti kontrol adaptif yang diharapkan dapat mempermudah memperoleh gain K yang lebih akurat dibanding dengan <i>trial</i> dan <i>error</i> .

---

*--Halaman ini sengaja dikosongkan--*

## BIODATA PENULIS



**Anggara Trisna Nugraha**, putra pertama dari dua bersaudara dari pasangan Drs. Syaiful Anam dan Ir. Ismatul Mas'adah. Penulis biasa dipanggil dengan nama Angga. Penulis lahir dikota Gresik pada tanggal 7 November 1993. Penulis memulai pendidikannya dari Playgroup Petrokimia Gresik kemudian melanjutkan studi di TK. Islam Bhakti 3, SDNU 1 Gresik, SMP Negeri 2 Gresik, dan SMA Negeri 1 Manyar. Setelah lulus dari SMA pada tahun 2011, penulis melanjutkan studi di Jurusan Teknik Elektro, program studi Teknik Elektro

dengan konsentrasi Elektronika dan Sistem Kendali, Fakultas Teknik, Universitas Negeri Jember dan lulus pada Juni 2015. Kemudian penulis melanjutkan program magister di Departemen Teknik Elektro, bidang keahlian Teknik Sistem Pengaturan, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya hingga saat ini. Penulis selama kuliah S2 menjadi member di Laboratorium Sistem Biocybernetic dan tergabung dalam tim riset quadrotor. Penulis dapat dihubungi melalui alamat email [anggaratnugraha@yahoo.co.id](mailto:anggaratnugraha@yahoo.co.id).

*--Halaman ini sengaja dikosongkan--*