



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

PERANCANGAN PEMANDUAN PELURU KENDALI MENGUNAKAN METODE *SLIDING CURVE*

Leonard Wihardi
NRP 0711154000096

Dosen Pembimbing
Ir. Rusdianto Effendi A.K., MT.
Zulkufli Hidayat ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - EE 184801

**PERANCANGAN PEMANDUAN PELURU KENDALI
MENGUNAKAN METODE *SLIDING CURVE***

Leonard Wihardi
NRP 0711154000096

Dosen Pembimbing
Ir. Rusdianto Effendi A.K., MT.
Zulkifli Hidayat ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - EE 184801

**MISSILE GUIDANCE DESIGN USING SLIDING CURVE
METHOD**

Leonard Wihardi
NRP 0711154000096

Supervisor

Ir. Rusdianto Effendi A.K., MT.
Zulkifli Hidayat ST., M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul

“Perancangan Pemanduan Peluru Kendali Menggunakan Metode *Sliding Cruve*”

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, Juni 2019



Leonard Wihardi

Nrp 0711154000096

[Halaman ini sengaja dikosongkan]

**PERANCANGAN PEMANDUAN PELURU KENDALI
MENGUNAKAN METODE *SLIDING CURVE***

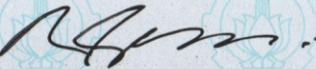
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui

Dosen Pembimbing I,

Dosen Pembimbing II,



Ir. Rusdhianto Effendi AK., M.T.
NIP. 195704241985021001



Zulkifli Hidayat ST., M.Sc.
NIP. 197012251999031002



[Halaman ini sengaja dikosongkan]

PERANCANGAN PEMANDUAN PELURU KENDALI MENGUNAKAN METODE *SLIDING CURVE*

Leonard Wihardi

0711154000096

Dosen Pembimbing I: Ir. Rusdianto Effendi A K., MT.

Dosen Pembimbing II: Zulkifli Hidayat ST., M.Sc.

ABSTRAK

Peluru kendali (rudal) dirancang agar bisa dikendalikan secara otonom dengan tujuan untuk mencapai sasaran yang diinginkan. Perancangan pemanduan rudal menjadi sangat penting untuk dikembangkan karena tanpa adanya pemanduan yang baik, pergerakan rudal akan menjadi tidak akurat dan bahkan tidak bisa mengenai sasaran yang diinginkan. Pemanduan rudal yang dirancang pada tugas akhir ini menggunakan metode *sliding curve*, dimana sistem pemanduan akan berusaha untuk mengikuti trayektori tertentu yang telah ditentukan dan memberi referensi sudut orientasi pada sistem *autopilot* rudal. Penentuan sudut referensi pada pemanduan rudal didapatkan berdasarkan posisi relatif rudal terhadap trayektori referensi. Perencanaan trayektori rudal dengan *Dubins curve* akan digunakan untuk mendapatkan trayektori referensi yang lebih mulus. Selisih jarak orthogonal antara posisi rudal dengan trayektori disebut sebagai *cross-track error*. Sistem pemanduan ini akan aktif selama rudal berada di luar *sliding surface* dan akan non-aktif ketika rudal telah berada di dalam *sliding surface*. Simulasi pengejaran rudal menuju sasaran adalah menggunakan *waypoint* statis. Hasil simulasi menunjukkan bahwa dengan *waypoint* statis, pemanduan *sliding curve* mampu memandu rudal untuk bergerak sepanjang trayektori referensi dengan nilai *Root-Mean-Square-Error* (RMSE) dari *cross-track error* sebesar 0.1160 meter dan *vertical-track error* sebesar 0.2972 meter..

Kata Kunci: *Dubins Curve*, Peluru Kendali, Sistem Pemanduan, *Sliding Curve*

[Halaman ini sengaja dikosongkan]

**MISSILE GUIDANCE DESIGN USING SLIDING CURVE
METHOD**

Leonard Wihardi
0711154000096

Supervisor I: Ir. Rusdianto Effendi A K., MT.

Supervisor II: Zulkifli Hidayat ST., M.Sc.

ABSTRACT

Autonomous guided missile needs to be guided well to reach its target. Missile guidance design has become a pivotal part to be designed for without accurate guidance, the missile cannot reach its target accurately or even does not reach the target at all. In this final project the design of missile guidance is based on the sliding curve method, in which the guidance will guide the missile according to a determined trajectory by giving an orientation angle reference to the missile's autopilot system. The orientation angle reference is acquired through calculating the relative position between missile and the trajectory itself. Dubins curve is used to create a smooth reference trajectory for the missile. Distance between missile and reference trajectory is called cross-track error. The designed missile guidance will only be active as long as the missile is out of the sliding curve boundaries. The simulation held in this final project will use a static waypoints. The result of the simulation shows that the designed missile guidance could guide the missile along the determined trajectory with Root-Mean-Square-Value (RMSE) of cross-track error 0.1160 metre and vertical-track error 0.2972 metre.

Key Words: *Dubins Curve, Guided Missile, Guidance System, Sliding Curve*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT yang atas berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Tugas Akhir berjudul “**Perancangan Pemanduan Peluru Kendali Menggunakan Metode *Sliding Curve***” untuk memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.

Buku Laporan tugas akhir ini dapat terselesaikan berkat oleh penulis berkat bimbingan dan bantuan dari Allah SWT. Penulis ingin mengucapkan rasa terima kasih kepada Ibu yang selalu sabar dan mendukung penulis dengan semua bantuan yang dapat diberikan, baik berupa materil maupun non-materil, berkatnya penulis dapat menyelesaikan buku laporan. Terima kasih kepada Bapak Rusdhianto Effendi AK MT., dan Bapak Zulkifli Hidayat ST., M.Sc selaku dosen pembimbing atas masukan, arahan dan ilmu yang disalurkan kepada penulis dalam menyelesaikan buku laporan ini. Dosen-dosen Bidang Studi Teknik Sistem Pengaturan atas ilmu-ilmu yang disalurkan, baik melalui perkuliahan maupun non-perkuliahan. Para senior S2 Teknik Elektro Bidang Studi Teknik Sistem Pengaturan, mas Mochamad Nur Qomarudin, dan keluarga Laboratorium Sistem dan Sibernetik B204. Tommy Pratama, M. Farih, Ananta Muhammad, dan teman-teman yang telah membantu serta berjuang bersama-sama penulis.

Penulis berharap buku laporan ini dapat memberikan manfaat kepada pembaca, baik secara langsung maupun tak langsung. Laporan ini memang masih jauh dari sempurna, oleh karenanya penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Surabaya, Juni 2019

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	v
LEMBAR PENGESAHAN	Error! Bookmark not defined.
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	1
1.3. Batasan Masalah	2
1.4. Tujuan	2
1.5. Metodologi	2
1.6. Sistematika	4
1.6.1. BAB 1 Pendahuluan	4
1.6.2. BAB 2 Dasar Teori	4
1.6.3. BAB 3 Perancangan Simulasi Sistem	4
1.6.4. BAB 4 Hasil Simulasi dan Analisa	4
1.6.5. BAB 5 Kesimpulan dan Saran	4
1.7. Relevansi	4
BAB 2 DASAR TEORI	5
2.1. Peluru Kendali (Rudal)	5
2.2. Sistem Koordinat	7
2.2.1. Transformasi Vektor	8
2.2.2. Sudut Euler	9
2.3. Model Matematik Rudal	11
2.3.1. Kinematika Rudal	11
2.3.2. Dinamika Rudal	12
2.4. Dubins Curve	19
2.5. Metode Pemanduan <i>Sliding Curve</i>	22
2.6. Sistem <i>Autopilot</i> Rudal	25
2.7. Kontroler PID	27
BAB 3 PERANCANGAN SISTEM	29
3.1. Gambaran Umum Sistem	29
3.2. Perancangan Model Rudal	29

3.3. Perancangan Trayektori Dubins Curve	37
3.4. Perancangan Pemandu Rudal.....	43
3.5. Perancangan <i>Autopilot</i> Rudal.....	49
3.5.1. <i>Autopilot Roll</i>	50
3.5.2. <i>Autopilot Yaw</i>	51
3.5.3. <i>Autopilot Pitch</i>	52
3.5.4. Mekanisasi Sirip Rudal	53
3.6. Perancangan Pengujian Sistem.....	56
3.6.1. Perancangan Pengujian Model Matematik Rudal	56
3.6.2. Perancangan Pengujian Trayektori <i>Dubins Curve</i>	56
3.6.3. Perancangan Pengujian Sistem <i>Autopilot</i> Rudal	56
3.6.4. Perancangan Pengujian Sistem Pemanduan Rudal	56
3.6.5. Perancangan Pengujian Sistem Keseluruhan	57
BAB 4 HASIL DAN ANALISA	59
4.1. Pengujian Model Matematik Rudal.....	59
4.2. Pengujian Trayektori <i>Dubins Curve</i>	61
4.3. Pengujian Sistem <i>Autopilot</i>	64
4.3.1. Pengujian Inner Loop <i>Autopilot Pitch</i>	66
4.3.2. Pengujian Inner Loop <i>Autopilot Roll</i>	68
4.3.3. Pengujian Inner Loop <i>Autopilot Yaw</i>	70
4.3.4. Pengujian Outer Loop <i>Autopilot Pitch</i>	72
4.3.5. Pengujian Outer Loop <i>Autopilot Roll</i>	73
4.3.6. Pengujian Outer Loop <i>Autopilot Yaw</i>	73
4.4. Pengujian Sistem Pemanduan Rudal.....	74
4.4.1. Pengujian Sistem Pemanduan Garis Lurus	74
4.4.2. Pengujian Sistem Pemanduan Lingkaran	78
4.4.3. Pengujian Sistem Pemanduan Gabungan	81
BAB 5 PENUTUP	85
5.1 Kesimpulan	87
5.2 Saran	88
DAFTAR PUSTAKA	89
LAMPIRAN	91
L1. Blok Simulink untuk Model Matematik Rudal	91
L2. Blok Simulink untuk Pemandu dan <i>Autopilot</i>	91
RIWAYAT HIDUP PENULIS	93

DAFTAR GAMBAR

Gambar 2.1	Diagram Alur Sistem Rudal Otonom.....	6
Gambar 2.2	Sistem Koordinat Rudal [1]	7
Gambar 2.3	Sistem Koordinat Rudal [1]	9
Gambar 2.4	Transformasi koordinat dengan rotasi pada setiap sumbu	11
Gambar 2.5	Diagram vektor koefisien aerodinamik rudal.....	18
Gambar 2.6	Kombinasi Lintasan pada <i>Dubins curve</i> [5].....	20
Gambar 2.7	Lintasan <i>Dubins</i> RSR [1].....	21
Gambar 2.8	Pemanduan Sliding Curve [4].....	23
Gambar 2.9	Pergerakan rudal di dalam selubung <i>sliding surface</i>	25
Gambar 2.10	Defleksi Sirip Rudal [2].....	26
Gambar 2.11	Konfigurasi Sirip Rudal <i>plus</i> (kiri) dan <i>cross</i> (kanan) ...	26
Gambar 2.12	Diagram blok kontroler PID	27
Gambar 3.1	Blok fungsi komponen gaya dan momen aerodinamik	31
Gambar 3.2	Blok fungsi gabungan dinamika dan kinematika rudal	32
Gambar 3.3	Grafik gaya dorong <i>thruster</i> rudal	33
Gambar 3.4	Blok fungsi perhitungan massa dan inersia.....	33
Gambar 3.5	Blok fungsi model lengkungan rudal.....	34
Gambar 3.6	Peletakan lingkaran pada bidang 3 dimensi	38
Gambar 3.7	Ilustrasi vektor pada ruang 3 dimensi	39
Gambar 3.8	Pencarian titik pusat lingkaran baru.....	40
Gambar 3.9	Mencari garis singgung terhadap lingkaran baru	41
Gambar 3.10	Trayektori busur lingkaran yang menghubungkan <i>Ten</i> dan <i>Tex</i>	41
Gambar 3.11	Blok fungsi <i>waypoint</i> untuk memberikan titik <i>waypoint</i> referensi pada sistem pemanduan.....	43
Gambar 3.12	Blok fungsi sistem pemanduan rudal	44
Gambar 3.13	Diagram alur strategi sistem pemanduan rudal	47
Gambar 3.14	Menghitung jarak orthogonal titik terhadap vektor	48
Gambar 3.15	Blok diagram <i>autopilot roll</i>	50
Gambar 3.16	Blok Simulink <i>autopilot roll</i>	51
Gambar 3.17	Blok diagram <i>autopilot yaw</i>	52
Gambar 3.18	Blok Simulink <i>autopilot yaw</i>	52
Gambar 3.19	Blok diagram <i>autopilot pitch</i>	53
Gambar 3.20	Blok Simulink <i>autopilot pitch</i>	53

Gambar 3.21	Sirip rudal dengan susunan <i>cross</i>	54
Gambar 3.22	Blok Simulink dari Dinamika Aktuator Sirip Rudal	55
Gambar 4.1	Posisi rudal (a) x_e , (b) y_e dan (c) z_e	60
Gambar 4.2	Trayektori Garis Lurus dengan 4 buah <i>waypoints</i>	62
Gambar 4.3	Trayektori dengan <i>Dubins curve</i>	63
Gambar 4.4	Titik belok pada <i>waypoint</i> 1	63
Gambar 4.5	Titik belok pada <i>waypoint</i> 2	64
Gambar 4.6	Titik belok pada <i>waypoint</i> 3	64
Gambar 4.7	Respon sudut <i>pitch</i> rudal dengan kontroler proporsional	67
Gambar 4.8	Respon sudut <i>pitch</i> rudal dengan kontroler PI.....	68
Gambar 4.9	Respon sudut <i>roll</i> rudal dengan kontroler proporsional ...	69
Gambar 4.10	Respon sudut <i>roll</i> rudal dengan kontroler PI.....	70
Gambar 4.11	Respon sudut <i>yaw</i> rudal dengan kontroler proporsional	72
Gambar 4.12	Respon sudut <i>yaw</i> rudal dengan kontroler PI	72
Gambar 4.13	Sistem pemanduan <i>pitch</i> garis lurus (a) tampilan keseluruhan dan (b) tampilan diperbesar pada saat terjadinya <i>overshoot</i>	75
Gambar 4.14	Sistem pemanduan <i>yaw</i> garis lurus (a) tampilan keseluruhan dan (b) tampilan diperbesar pada saat terjadinya <i>overshoot</i>	76
Gambar 4.15	Grafik (a) <i>cross-track error</i> dan (b) <i>vertical-track error</i> rudal terhadap trayektori referensi	77
Gambar 4.16	Hasil pengujian sistem pemanduan <i>yaw</i> lintasan lingkaran (a) tampilan keseluruhan dan (b) tampilan diperbesar saat posisi $x = 6500$	80
Gambar 4.17	<i>Cross-Track error</i> sistem pemanduan <i>yaw</i> lingkaran dengan nilai λ yang berbeda-beda.	80
Gambar 4.18	Hasil simulasi sistem pemanduan gabungan dengan 4 titik <i>waypoints</i> (termasuk sasaran	82
Gambar 4.19	Grafik <i>cross-track error</i> pemanduan gabungan	83
Gambar 4.20	Grafik <i>Vertical-Track error</i> pemanduan gabungan	83

DAFTAR TABEL

Tabel 2.1 Komponen aerodinamik, koordinat, dan kecepatan rudal	19
Tabel 2.2 Karakteristik respon kontroler PID.....	28
Tabel 3.1 Nilai parameter rudal yang digunakan.....	36
Tabel 3.2 Algoritma Sistem pemanduan garis lurus <i>yaw</i>	45
Tabel 3.3 Algoritma sistem pemanduan garis lurus <i>pitch</i>	45
Tabel 3.4 Algoritma sistem pemanduan lingkaran <i>yaw</i>	46
Tabel 3.5 Algoritma sistem pemanduan lingkaran <i>pitch</i>	46
Tabel 3.6 Parameter kontroler <i>cascade</i> P-PI pada <i>autopilot roll</i>	51
Tabel 3.7 Parameter kontroler <i>autopilot yaw</i>	52
Tabel 3.8 Parameter kontroler <i>autopilot pitch</i>	53
Tabel 4.1 Konfigurasi Awal Rudal	59
Tabel 4.2 Titik <i>Waypoints</i> pengujian trayektori garis lurus.....	61
Tabel 4.3 Kontroler proporsional <i>autopilot pitch</i>	66
Tabel 4.4 Kontroler proporsional-integral <i>autopilot pitch</i>	67
Tabel 4.5 Kontroler proporsional <i>autopilot roll</i>	68
Tabel 4.6 Kontroler proporsional-integral <i>autopilot roll</i>	69
Tabel 4.7 Kontroler proporsional <i>autopilot yaw</i>	71
Tabel 4.8 Kontroler proporsional-integral <i>autopilot yaw</i>	71
Tabel 4.9 Kontroler proporsional <i>autopilot pitch</i>	73
Tabel 4.10 Kontroler proporsional <i>autopilot roll</i>	73
Tabel 4.11 Kontroler proporsional <i>autopilot yaw</i>	73
Tabel 4.12 <i>Cross-Track error</i> pemanduan garis lurus.....	77
Tabel 4.13 <i>Vertical-Track error</i> pemanduan garis lurus	77
Tabel 4.14 Selisih jarak rudal ketika melewati <i>waypoint</i>	78
Tabel 4.15 <i>Cross-Track error</i> pemanduan lingkaran	81
Tabel 4.16 Pengujian sistem pemanduan gabungan	81
Tabel 4.17 <i>Cross-Track error</i> pemanduan gabungan.....	82
Tabel 4.18 <i>Vertical-Track error</i> pemanduan gabungan	83
Tabel 4.19 <i>Miss Distance</i> rudal terhadap <i>waypoint</i>	84

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan teknologi di bidang militer yang sangat pesat, sangatlah penting untuk turut menguasai dan mengembangkan teknologi tersebut. Salah satunya adalah teknologi di bidang sistem pertahanan udara. Sistem pertahanan udara menjadi penting pada saat ada objek di daratan, laut, maupun udara milik musuh atau pihak lain yang ingin menimbulkan konflik memasuki wilayah teritorial suatu negara tanpa izin.

Salah satu sistem pertahanan udara yang utama adalah peluru kendali (rudal). Rudal bisa menghancurkan objek di darat, laut, maupun udara membuatnya menjadi sistem pertahanan udara yang sangat baik. Untuk dapat mencapai suatu sasaran tertentu, rudal harus dikendalikan baik itu secara langsung maupun secara otonom. Sistem pemanduan dan sistem *autopilot* perlu dirancang dengan baik agar rudal bisa bergerak dan mencapai sasaran yang telah ditentukan secara akurat.

Telah banyak penelitian yang sudah dilakukan untuk menyelesaikan permasalahan sistem pemanduan rudal seperti algoritma *Carrot-Chasing Algorithm*, *Non-Linear Guidance Law*, dan *Vector Field* [4]. Kebanyakan algoritma di atas hanya mengimplementasikan pada *plant* berupa *Unmanned Aerial Vehicle* (UAV) dan juga algoritma yang digunakan hanya untuk gerakan 2 dimensi saja yaitu pada bidang lateral sumbu x - y , sedangkan untuk pengaturan ketinggian digunakan *altitude control*.

Dalam tugas akhir ini, *sliding curve guidance* digunakan karena dapat melakukan pemanduan pada ruang 3 dimensi serta cocok untuk diimplementasikan dengan trayektori referensi deterministik dimana rudal harus bergerak melewati trayektori tersebut dan mengenai titik-titik *waypoint* yang telah ditentukan sebelumnya.

1.2. Perumusan Masalah

Pada penelitian ini masalah yang akan dibahas adalah mengenai bagaimana suatu sistem pemanduan rudal dapat memandu pergerakan rudal untuk dapat bergerak mengikuti trayektori yang telah ditentukan dengan nilai kesalahan yang minimum sehingga pergerakan rudal dapat akurat dan mengenai sasaran yang telah ditentukan.

Trayektori yang akan dilalui rudal merupakan suatu kurva yang dibangkitkan berdasarkan *waypoints* deterministik yang telah ditentukan sebelum rudal diluncurkan di udara. Ketika rudal bergerak di udara mengikuti trayektori tersebut, selisih jarak posisi rudal pada tiap saat tidak boleh melampaui nilai tertentu, karena apabila selisih jarak tersebut sangat besar maka dapat dikatakan bahwa sistem pemanduan rudal belum bekerja dengan baik.

1.3. Batasan Masalah

Dalam sistem pemanduan rudal yang sebenarnya, terdapat banyak banyak masalah dan kendala yang harus juga dipertimbangkan dalam melakukan perancangan sistem. Oleh karena itu diperlukan batasan masalah sebagai berikut:

1. Menggunakan rudal *surface to surface* (SSM).
2. Manuver rudal merupakan jenis *skid to turn*.
3. Pergerakan *roll* pada rudal diasumsikan terjaga konstan
4. Pendorong rudal (*thrust*) tidak dapat dikendalikan.
5. Pemanduan rudal berada pada fase *midcourse*.
6. Badan rudal merupakan benda tegar dan tidak mengalami efek aeroelastis.

1.4. Tujuan

Tujuan yang ingin dicapai pada tugas akhir ini adalah dapat merancang sistem pemanduan rudal yang mampu mendapatkan sudut orientasi referensi rudal pada setiap saat agar rudal bisa bergerak mengikuti lintasan trayektori dengan akurat. Pergerakan rudal yang akurat dapat diartikan sebagai selisih jarak relatif rudal terhadap trayektori yang diinginkan adalah minimum dan rudal harus melewati titik *waypoint* yang telah ditentukan pada posisi-posisi tertentu.

Keluaran dari sistem pemanduan rudal ini akan menjadi masukan dari sistem *autopilot*, sehingga pemanduan yang akurat dan efisien sangatlah penting. Keluaran dari sistem pemanduan ini nantinya berupa informasi sudut, sehingga sistem *autopilot* akan mengolah informasi sudut yang harus dicapai tersebut menjadi pergerakan aktuator atau sirip-sirip rudal.

1.5. Metodologi

Metodologi yang akan diterapkan pada penelitian tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur yang dilakukan meliputi memperoleh persamaan matematik rudal, metode pemanduan rudal, dan metode-metode penunjang lainnya. Studi literatur ini bersumber dari buku, jurnal ilmiah, serta laporan penelitian Laboratorium Sistem dan Sibernetika Institut Teknologi Sepuluh Nopember.

2. Pemodelan Sistem

Membuat model persamaan matematik rudal yang didapatkan dari sumber literatur pada metodologi tahap studi literatur. Hasil pemodelan rudal pada tahap ini akan menjadi objek penelitian tugas akhir pada tahap selanjutnya.

3. Perancangan Trayektori

Membuat kurva berdasarkan titik-titik *waypoint* yang telah ditentukan. Metode untuk membuat kurva dari *waypoint* adalah menggunakan *Dubins curve*. Hasil kurva yang dihasilkan tersebut harus melewati titik *waypoint* yang ada.

4. Perancangan Sistem Pemanduan

Setelah lintasan dibuat, akan dirancang sebuah sistem pemanduan pada rudal yang berfungsi untuk mendapatkan informasi sudut orientasi yang harus dicapai rudal agar bisa mengikuti lintasan yang ada. Sistem pemanduan yang dirancang ini menggunakan algoritma *sliding curve* atau pada referensi lain juga kerap kali disebut sebagai *carrot-chasing* atau *rabbit-chasing* [4].

5. Pengujian dan Analisa

Setelah sistem pemanduan rudal selesai dirancang, maka akan dilakukan pengujian sistem pemanduan rudal ketika diberikan trayektori referensi tertentu dengan sasaran statis. Analisa dilakukan terhadap kinerja sistem pemanduan dengan mendapatkan *mean squared error* (MSE) jarak relatif posisi rudal dengan lintasan.

6. Kesimpulan dan Saran

Setelah pengujian dan analisis dilakukan, maka akan ditarik beberapa kesimpulan terkait dengan hasil tersebut. Saran juga akan diberikan sebagai potensi pengembangan lebih lanjut dari penelitian yang telah dilakukan.

7. Penyusunan Buku Tugas Akhir

Melakukan penulisan buku tugas akhir yang berbentuk laporan tertulis dari penelitian tugas akhir yang telah dilakukan.

1.6. Sistematika

Penulisan laporan tugas akhir ini dibagi menjadi 5 bab yang akan diuraikan secara singkat sebagai berikut:

1.6.1. BAB 1 Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang dari pemilihan judul tugas akhir, permasalahan yang ada, tujuan tugas akhir, metodologi, sistematika penulisan laporan tugas akhir, dan relevansi tugas akhir ini.

1.6.2. BAB 2 Dasar Teori

Bab ini menjelaskan teori-teori penunjang dan teori-teori dasar yang digunakan dalam pelaksanaan tugas akhir ini. Teori yang ada bersumber dari tinjauan pustaka yang menjadi acuan dalam pelaksanaan tugas akhir ini.

1.6.3. BAB 3 Perancangan Simulasi Sistem

Pada bab ini dijelaskan langkah-langkah sistematis mengenai perancangan pemanduan rudal mulai dari pemodelan dinamika rudal, perancangan trayektori deterministik, dan perancangan sistem pemanduan pada rudal.

1.6.4. BAB 4 Hasil Simulasi dan Analisa

Bab ini berisi hasil implementasi sistem pemanduan rudal yang telah dirancang melalui simulasi yang berbasis pada software *Matlab*. Setelah simulasi dijalankan, dilanjutkan dengan tahap menganalisa hasil simulasi dari sistem pemanduan yang telah dirancang.

1.6.5. BAB 5 Kesimpulan dan Saran

Bab ini berisikan kesimpulan yang didapatkan dari hasil simulasi dan analisa yang telah dilakukan pada bab sebelumnya serta diberikan juga saran untuk penelitian selanjutnya yang berkaitan dengan topik tugas akhir ini.

1.7. Relevansi

Hasil dari penelitian tugas akhir ini diharapkan dapat memberi manfaat maupun menjadi referensi yang bisa digunakan dalam penelitian lainnya di bidang sistem pertahanan udara khususnya sistem peluru kendali otonom. Selain itu hasil dari tugas akhir diharapkan dapat menjadi pembandingan untuk metode lain yang dapat diterapkan pada topik penelitian yang sama.

BAB 2

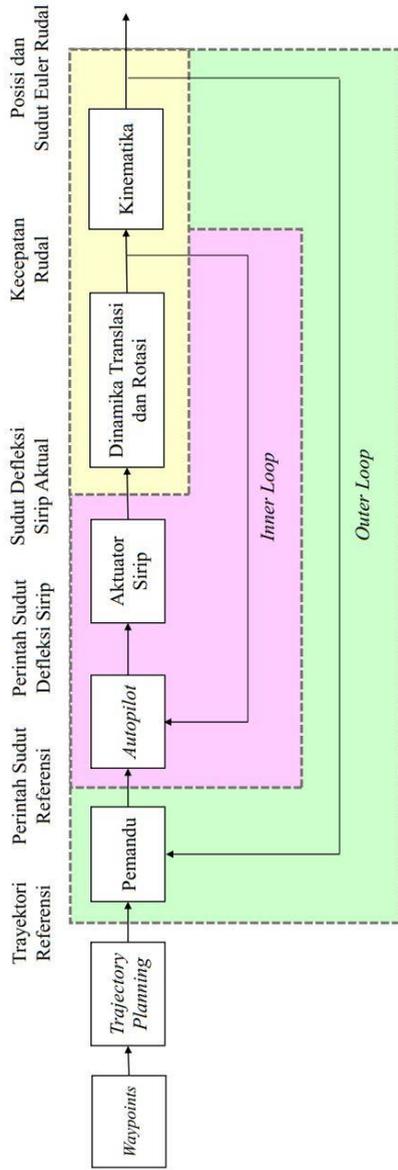
DASAR TEORI

2.1. Peluru Kendali (Rudal)

Peluru kendali atau yang biasa disingkat menjadi rudal merupakan salah satu jenis kendaraan tanpa awak yang bergerak di udara yang lintasan penerbangannya dapat dikendalikan. Peluru kendali diklasifikasikan berdasarkan tempat peluncuran rudal dan tempat sasaran secara fisik yaitu rudal *surface to surface*, *surface to air*, *air to surface*, dan *air to air*. Sesuai dengan namanya, rudal *surface to surface* adalah rudal yang diluncurkan dari permukaan atau daratan untuk mencapai sasaran yang juga di permukaan. *Surface to air* adalah rudal yang diluncurkan dari permukaan untuk mencapai sasaran di udara dan *air to surface* adalah kebalikannya. Rudal *air to air* adalah rudal yang diluncurkan dari udara dan mengejar sasaran yang juga di udara.

Berdasarkan cara rudal melakukan manuver di udara, terdapat 2 skema metode pengendalian gerak belok rudal yaitu *skid-to-turn* dan *bank-to-turn*. Rudal dengan skema gerak belok *skid-to-turn* memiliki karakteristik yaitu tidak adanya gerakan roll pada rudal sehingga bidang *pitch* dan *yaw* pada rudal memiliki perilaku respon yang identik. Selain itu, *skid-to-turn* memiliki hasil produk *cross* dari kopling inersia antara *roll*, *pitch*, dan *yaw* yang dapat diabaikan. Sedangkan rudal dengan skema gerak belok *bank-to-turn* dalam melakukan manuver akan melakukan rotasi *roll* terlebih dahulu sebelum *yaw*.

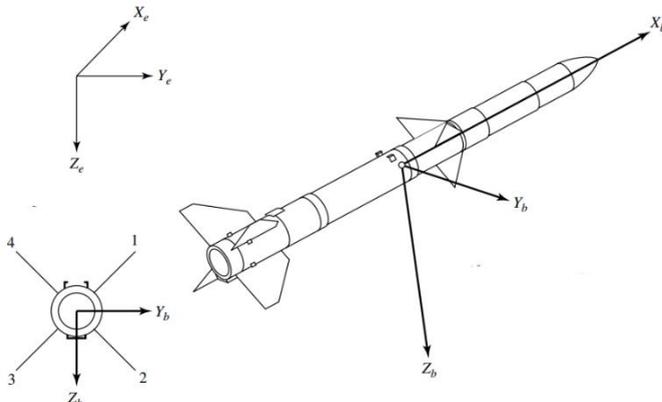
Dalam merancang sistem pemanduan maupun sistem *autopilot* rudal, tahapan penting yang harus dilakukan yang pertama adalah mendapatkan model dinamika rudal. Pada pemodelan dinamika rudal ini, diupayakan untuk mendapatkan model yang merepresentasikan hubungan antar variabel input dan output. Setelah melakukan pemodelan, maka dapat dirancang sistem pemanduan untuk rudal. Sistem pemanduan rudal merupakan algoritma perhitungan agar rudal dapat bergerak mencapai sasaran yang diinginkan. Tahap selanjutnya adalah perancangan komponen kontrol rudal atau sistem *autopilot* rudal. Sistem kontrol ini berfungsi untuk memberikan sinyal perintah kepada aktuator berdasarkan informasi yang diperoleh dari sistem pemanduan.



Gambar 2.1 Diagram Alur Sistem Rudal Otonom

2.2. Sistem Koordinat

Sebelum melakukan pemodelan matematik rudal, penting untuk dipahami terlebih dahulu mengenai sistem koordinat yang digunakan pada penelitian rudal ini. Sistem koordinat yang akan diadopsi dalam tugas akhir ini adalah sistem koordinat tangan kanan, dimana sumbu x berada pada sumbu longitudinal badan rudal, sumbu y berada pada positif ke arah kanan rudal, dan sumbu z berada pada positif ke arah bawah badan rudal. Dari sistem koordinat tangan kanan ini, sumbu z merupakan hasil *cross product* dari sumbu x dan sumbu y . Sistem koordinat ini juga dapat disebut sebagai sistem koordinat *north-east-down* (NED) yang berkaitan dengan perjanjian tanda pada *north-east-down* inersia.



Gambar 2.2 Sistem Koordinat Rudal [1]

Dapat dilihat dari Gambar 2.2 terdapat 2 sistem koordinat yang digunakan dalam penelitian ini, yaitu sistem koordinat badan rudal (*Missile Body-Fixed*) dan sistem koordinat tetap bumi (*Earth Fixed*). Sistem koordinat bumi memiliki notasi X_e, Y_e, Z_e . Seperti yang telah dijelaskan sebelumnya, pada sistem koordinat tangan kanan ini sumbu $X_e - Y_e$ merupakan di bidang horizontal dan sumbu Z_e menghadap ke bawah secara vertical searah dengan gravitasi. Sedangkan sistem koordinat badan rudal memiliki notasi X_b, Y_b, Z_b . Koordinat badan rudal ini bersifat tetap terhadap rudal, sehingga sistem koordinat turut bergerak mengikuti pergerakan rudal. Pada sistem koordinat badan rudal, sumbu X_b positif berada pada sumbu longitudinal rudal yang ke arah maju. Sumbu Y_b positif berada di sebelah kanan dari sumbu X_b dan berada pada

bidang horizontal. Sumbu ini merupakan sumbu *pitch*. Sumbu Z_b merupakan sumbu *yaw* yang menunjuk ke vertikal ke arah bawah.

2.2.1. Transformasi Vektor

Pada sistem koordinat kartesius atau rektanguler, suatu vektor dapat dispesifikasikan berdasarkan komponen-komponennya. Komponen dari vektor tersebut memiliki orientasi tersendiri pada sistem koordinat dan vektor yang sama dapat dibentuk dari beberapa kombinasi komponen triplet yang berbeda. Suatu vektor kemudian dapat didefinisikan sebagai susunan dari 3 angka yang bertransformasi seiring dengan perubahan koordinat suatu titik ketika sumbu-sumbu sistem dirotasi. Penggunaan notasi untuk merepresentasikan vektor seringkali menggunakan notasi a_1, a_2, a_3 agar lebih memudahkan dibandingkan penggunaat notasi huruf x, y, z . Penulisan notasi vektor tersebut biasanya secara keseluruhan adalah a_i , dimana indeks i memiliki nilai 1, 2, atau 3.

Rotasi dari sebuah sistem koordinat terhadap titik referensinya bisa direpresentasikan oleh 9 jumlah sudut kosinus antara sumbu i di satu posisi koordinat dengan sumbu j di posisi lainnya yang diberi notasi C_{ij} . Masing-masing dari 9 nilai tersebut memberikan sudut antar sumbu di satu posisi terhadap sumbu lainnya serta merupakan koefisien dari transformasi koordinat suatu titik. Dalam bentuk matriks, dapat dinyatakan pada Persamaan (2.1).

$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (2.1)$$

Karena C_{ij} dapat dianggap sebagai komponen dalam sumbu j pada satu sistem koordinat dari unit vektor pada sumbu i dari sistem koordinat lainnya, maka:

$$C_{i1}^2 + C_{i2}^2 + C_{i3}^2 = \sum_j C_{ij}^2 = 1 \quad (2.2)$$

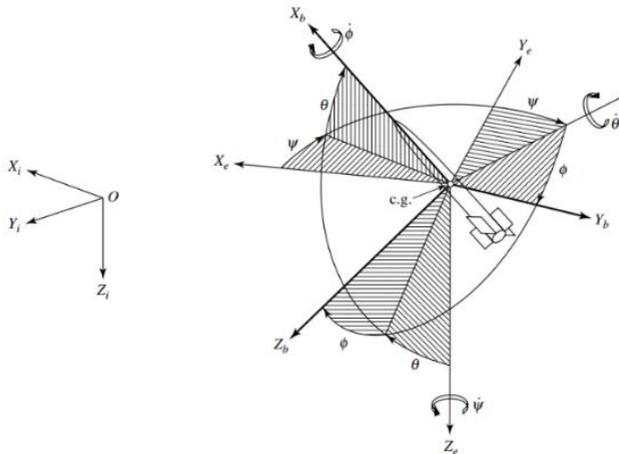
Komponen dari suatu vektor atau koordinat dari suatu titik dapat ditransformasikan dari satu sistem koordinat ke sistem koordinat lainnya dengan menggunakan Persamaan (2.3).

$$\mathbf{b}_i = C_{i1}\mathbf{a}_1 + C_{i2}\mathbf{a}_2 + C_{i3}\mathbf{a}_3 = \mathbf{C}_{ij}\mathbf{a}_j \quad (2.3)$$

dimana \mathbf{a}_j merupakan representasi dari komponen vektor \mathbf{a} dalam satu sistem koordinat dan \mathbf{b}_i merupakan representasi dari komponen di sistem koordinat lainnya.

2.2.2. Sudut Euler

Sudut Euler adalah salah satu metode yang umum digunakan dalam melakukan transformasi koordinat 3 sumbu satu dengan lainnya. Sudut Euler merupakan sebutan konvensional dari sistem dengan sumbu bergerak terhadap sistem sumbu tetap, yang sering digunakan pada mekanisasi dan/atau simulasi peluru kendali dan pesawat terbang. Notasi yang digunakan dalam sudut Euler adalah *roll* (ϕ), *pitch* (θ), dan *yaw* (ψ). Metode Euler dapat digunakan untuk melakukan transformasi dari sistem koordinat rudal ke sistem koordinat bumi dengan melakukan rotasi pada tiap sumbu dari sistem koordinat badan rudal ke sistem koordinat bumi. Perlu diperhatikan dalam melakukan rotasi tiap sumbu tersebut urutannya berpengaruh, karena urutan yang terbalik dapat menghasilkan matriks transformasi yang berbeda.



Gambar 2.3 Sistem Koordinat Rudal [1]

Dibandingkan dengan metode transformasi sistem koordinat yang lainnya, metode Euler ini memiliki kelebihan tersendiri yaitu sudut Euler dan laju perubahan sudut Euler dapat diinterpretasikan dengan mudah secara fisik, sehingga proses mekanisasi yang relatif sederhana pada simulasi menggunakan komputer.

Dalam tugas akhir ini, urutan rotasi yang dilakukan terlebih dahulu untuk melakukan transformasi koordinat dari sistem koordinat bumi ke sistem koordinat rudal adalah rotasi pada sumbu $-z$ (*yaw*), dilanjutkan

dengan rotasi sumbu $-y$ (*pitch*), dan diakhiri dengan rotasi sumbu $-x$ (*roll*). Rotasi pada sumbu $-z$ menghasilkan matriks transformasi seperti pada Persamaan (2.4).

$$\begin{bmatrix} X'_e \\ Y'_e \\ Z'_e \end{bmatrix} = \mathbf{A} \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} \quad (2.4)$$

Setelah mendapatkan koordinat baru yang telah dirotasi terhadap sumbu $-z$, dilanjutkan dengan rotasi terhadap sumbu $-y$ dengan menggunakan Persamaan (2.5)

$$\begin{bmatrix} X''_e \\ Y''_e \\ Z''_e \end{bmatrix} = \mathbf{B} \begin{bmatrix} X'_e \\ Y'_e \\ Z'_e \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} X'_e \\ Y'_e \\ Z'_e \end{bmatrix} \quad (2.5)$$

Rotasi selanjutnya adalah sumbu $-x$ dengan menggunakan Persamaan (2.6)

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \mathbf{D} \begin{bmatrix} X''_e \\ Y''_e \\ Z''_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X''_e \\ Y''_e \\ Z''_e \end{bmatrix} \quad (2.6)$$

Setelah melakukan proses rotasi pada tahap sebelumnya, maka telah didapatkan matriks rotasi sistem koordinat yang dapat dilihat pada Persamaan (2.7).

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \mathbf{DBA} \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} = \mathbf{C} \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} \quad (2.7)$$

Dimana:

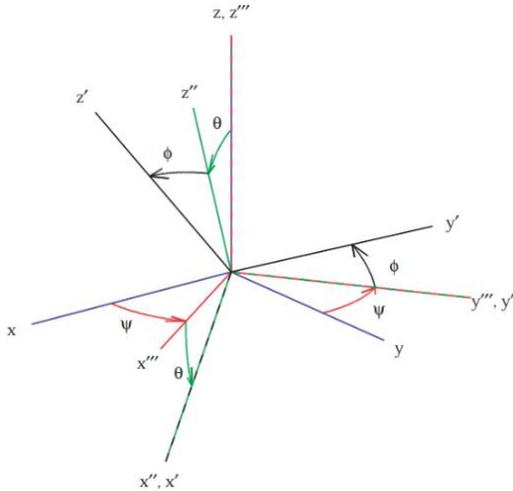
X_e, Y_e, Z_e = Sistem koordinat bumi

X_b, Y_b, Z_b = Sistem koordinat rudal

$$\mathbf{C} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\psi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix} \quad (2.8)$$

($s = \sinus$ dan $c = \cosinus$)

Ilustrasi tahap-tahap untuk mendapatkan matriks rotasi pada masing-masing sumbu dalam sistem koordinat dapat dilihat pada Gambar (2.4).



Gambar 2.4 Transformasi koordinat dengan rotasi pada setiap sumbu

2.3. Model Matematik Rudal

Persamaan matematik dari rudal bisa diturunkan dari hukum Newton. Dalam menurunkan Persamaan matematik dari rudal ini, perlu diasumsikan beberapa hal terhadap keadaan rudal. Asumsi yang pertama adalah rudal merupakan benda tegar, sehingga rudal tidak akan mengalami perubahan ukuran maupun bentuk selama pergerakannya. Selain itu, akan juga diasumsikan badan rudal bisa dianggap sebagai suatu partikel yang massanya berpusat pada titik pusat massa badan rudal.

Dalam tugas akhir ini, pemodelan matematik rudal diasumsikan memiliki enam derajat kebebasan (*degree of freedom*), dimana tiga diantaranya adalah derajat kebebasan gerak translasi dan tiga derajat kebebasan lainnya pada gerak rotasi pada sumbu badan rudal (X_b, Y_b, Z_b). Gerakan translasi pada rudal diberi notasi (u, v, w) dan gerak rotasi memiliki notasi (p, q, r) .

2.3.1. Kinematika Rudal

Kinematika merupakan cabang dari ilmu mekanika klasik yang membahas mengenai pergerakan dari suatu objek yang meliputi posisi,

kecepatan, dan percepatan objek tersebut tanpa memperhatikan gaya-gaya yang menyebabkan gerakan tersebut. Kinematika memiliki hubungan yang erat dengan transformasi koordinat, karena kinematika menjelaskan posisi, kecepatan atau percepatan dari suatu objek terhadap referensi tertentu yang kemudian ditransformasikan ke dalam sistem koordinat yang lainnya.

Kinematika rudal terbagi menjadi dua macam yakni kinematika translasi dan kinematika rotasi. Kedua kinematika pada rudal tersebut menjelaskan mengenai kecepatan translasi dan kecepatan rotasi pada sistem koordinat rudal terhadap sistem koordinat bumi. Menggunakan Persamaan (2.8), kinematika translasi bisa didapatkan seperti dapat dilihat pada Persamaan (2.9).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.9)$$

Dimana

$\dot{x}, \dot{y}, \dot{z}$ = Kecepatan translasi terhadap koordinat bumi

u, v, w = Kecepatan translasi terhadap koordinat badan rudal

ϕ, θ, ψ = Representasi sudut euler dari rotasi koordinat badan rudal terhadap koordinat bumi

Sedangkan kinematika rotasi didapatkan melalui transformasi kordinat badan rudal terhadap kordinat bumi seperti yang ditunjukkan pada Persamaan (2.10)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.10)$$

2.3.2. Dinamika Rudal

Dinamika juga merupakan cabang dari ilmu mekanika yang membahas pergerakan objek, meliputi posisi, kecepatan, dan percepatan. Berbeda dengan kinematika, pada dinamika gaya-gaya yang bekerja pada objek tersebut dipertimbangkan. Hukum Newton akan digunakan untuk menurunkan Persamaan dinamika suatu objek.

Sama seperti kinematika rudal, dinamika rudal juga terbagi menjadi dua macam yaitu dinamika translasi dan dinamika rotasi. Dinamika translasi menyatakan perubahan kecepatan maupun percepatan translasi dari rudal yang disebabkan oleh gaya-gaya tertentu. Sedangkan

dinamika rotasi menyatakan perubahan kecepatan sudut maupun percepatan sudut akibat momen-momen yang ada. Kecepatan translasi dari rudal direpresentasikan dengan \mathbf{V}_b , meliputi kecepatan rudal u, v, w dan kecepatan rotasi atau kecepatan sudut rudal direpresentasikan dengan $\boldsymbol{\omega}_b$ yang meliputi kecepatan sudut rudal p, q, r . Secara matematik, Persamaan \mathbf{V}_b dan $\boldsymbol{\omega}_b$ dapat dilihat pada Persamaan (2.11) dan (2.12)

$$\mathbf{V}_b = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} \quad (2.11)$$

$$\boldsymbol{\omega}_b = p\mathbf{i} + q\mathbf{j} + r\mathbf{k} \quad (2.12)$$

Dimana $\mathbf{i}, \mathbf{j}, \mathbf{k}$ adalah vektor satuan pada tiap sumbu sistem koordinat badan rudal (X_b, Y_b, Z_b). Magnitudo kecepatan transasi rudal dapat dihitung menggunakan Persamaan (2.13).

$$V_m = \|\mathbf{V}_b\| = \sqrt{u^2 + v^2 + w^2} \quad (2.13)$$

2.3.2.1. Dinamika Translasi Rudal

Persamaan matematik dinamika kecepatan atau percepatan translasi rudal dalam sistem badan rudal dapat dirumuskan menggunakan hukum Newton kedua, yaitu jumlahan dari gaya-gaya yang bekerja pada tiap sumbu sistem koordinat rudal seperti pada Persamaan (2.14).

$$\sum \mathbf{F} = \mathbf{F}_b = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k} \quad (2.14)$$

Berdasarkan hukum Newton tersebut, penjumlahan semua gaya eksternal yang bekerja pada badan adalah sama dengan momentum badan terhadap waktu. Dengan demikian, Persamaan kecepatan translasi rudal dalam sistem koordinat dapat dituliskan seperti pada Persamaan (2.15) dan (2.16).

$$\mathbf{F}_b = m\dot{\mathbf{V}}_b + \boldsymbol{\omega}_b \times \mathbf{V}_b \quad (2.15)$$

$$\frac{d\mathbf{V}_b}{dt} = \frac{\mathbf{F}_b}{m} - \boldsymbol{\omega}_b \times \mathbf{V}_b \quad (2.16)$$

dimana:

\mathbf{F}_b = Gaya yang bekerja pada rudal

\mathbf{V}_b = Kecepatan translasi rudal

$\boldsymbol{\omega}_b$ = Kecepatan sudut rudal

m = Massa rudal

\dot{m} = Perubahan Massa Rudal

Dari Persamaan (2.15), $(\boldsymbol{\omega}_b \times \mathbf{V}_b)$ yang merupakan *cross product* dari kecepatan translasi dan rotasi rudal dapat dijabarkan ke dalam Persamaan (2.17) dan (2.18).

$$\boldsymbol{\omega}_b \times \mathbf{V}_b = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ p & q & r \\ u & v & w \end{bmatrix} \quad (2.17)$$

$$\boldsymbol{\omega}_b \times \mathbf{V}_b = (qw - rv)\mathbf{i} + (ru - pw)\mathbf{j} + (pv - qu)\mathbf{k} \quad (2.18)$$

Percepatan translasi rudal pada setiap sumbu dalam sistem koordinat rudal kemudian dapat didapatkan seperti pada Persamaan (2.19) hingga Persamaan (2.21).

$$\frac{du}{dt} = \frac{F_x}{m} + rv - qw \quad (2.19)$$

$$\frac{dv}{dt} = \frac{F_y}{m} + pw - ru \quad (2.20)$$

$$\frac{dw}{dt} = \frac{F_z}{m} + qu - pv \quad (2.21)$$

Adapun gaya-gaya yang dialami rudal pada sepanjang sistem koordinat rudal (F_x, F_y, F_z) seperti pada Persamaan (2.14) adalah gaya *thruster* atau roket pendorong pada rudal, gaya aerodinamik serta gaya gravitasi. Gaya-gaya yang dialami rudal ini bersifat tetap pada titik pusat gravitasi atau *center of gravity (cog)* dari rudal. Secara matematik, gaya-gaya tersebut dapat dinyatakan seperti pada Persamaan (2.22) hingga Persamaan (2.25).

$$F_x = T - mg \sin \theta + qSC_D \quad (2.22)$$

$$F_y = mg \sin \phi \cos \theta + qSC_S \quad (2.23)$$

$$F_z = mg \cos \phi \cos \theta + qSC_L \quad (2.24)$$

$$q = \frac{1}{2} \rho V_m^2 \quad (2.25)$$

Dimana m adalah massa rudal, g adalah percepatan gravitasi bumi, T adalah gaya dorong *thruster*, ϕ adalah sudut *roll*, θ adalah sudut *pitch*, q tekanan dinamis arus bebas pada *airfoil*, ρ adalah kerapatan atmosfer, S adalah luas permukaan salah satu *airfoil*, C_D , C_S , dan C_L adalah koefisien aerodinamik pada masing-masing sumbu di koordinat badan rudal. Koefisien aerodinamik untuk sumbu x , y , dan z tersebut secara berurutan dinamakan koefisien *drag*, *sde force*, dan *lift*. *Drag* itu sendiri merupakan gaya aerodinamik yang sejajar dengan arah angin dan *lift* adalah gaya aerodinamik yang orthogonal terhadap arah angin.

Sedangkan *side force* adalah gaya aerodinamik yang tegak lurus terhadap gaya aerodinamik *drag* dan *lift*. Persamaan untuk koefisien aerodinamik dapat dinyatakan pada Persamaan (2.26) hingga persamaan (2.28).

$$C_D = C_{D_1}\alpha + C_{D_2}M \quad (2.26)$$

$$C_S = C_{S_1}\beta + C_{S_2}M + C_{S_3}\delta_r \quad (2.27)$$

$$C_L = C_{L_1}\alpha + C_{L_2}M + C_{L_3}\delta_e \quad (2.28)$$

Dapat dilihat dari persamaan di atas bahwa C_D , C_S , dan C_L merupakan fungsi dari *angle of attack* (α), *sideslip* (β), sudut defleksi *rudder* (δ_r), sudut defleksi *elevator* (δ_e), dan *mach number* (M). Secara matematik, *angle of attack* (α), *sideslip* (β), dan *mach number* (M) dapat dihitung menggunakan Persamaan (2.29) hingga (2.31).

$$\alpha = \tan^{-1} \frac{w}{u} \quad (2.29)$$

$$\beta = \sin^{-1} \frac{v}{\sqrt{u^2 + v^2 + w^2}} = \sin^{-1} \frac{v}{V} \quad (2.30)$$

$$M = \frac{\sqrt{u^2 + v^2 + w^2}}{\text{kecepatan suara}} = \frac{V}{\text{kecepatan suara}} \quad (2.31)$$

Untuk mencari model kompak dinamika translasi dari rudal, dapat dilakukan dengan menggabungkan Persamaan (2.19) hingga (2.31) sehingga didapatkan Persamaan (2.32) hingga (2.34).

$$\frac{du}{dt} = \frac{T - mg \sin \theta + \bar{q}S(C_{D_1}\alpha + C_{D_2}M)}{m} + rv - qw \quad (2.32)$$

$$\frac{dv}{dt} = \frac{mg \sin \phi \cos \theta + \bar{q}S(C_{S_1}\beta + C_{S_2}M + C_{S_3}\delta_r)}{m} + pw - ru \quad (2.33)$$

$$\frac{dw}{dt} = \frac{mg \cos \phi \cos \theta + \bar{q}S(C_{L_1}\alpha + C_{L_2}M + C_{L_3}\delta_e)}{m} + qu - pv \quad (2.34)$$

2.3.2.2. Dinamika Rotasi Rudal

Pada dinamika rotasi rudal, gaya yang bekerja pada tiap simbu sistem koordinat rudal adalah momen. Momen-momen yang bekerja pada tiap sumbu koordinat rudal dapat dinyatakan pada Persamaan (2.35).

$$\mathbf{M}_b = M_x \hat{i} + M_y \hat{j} + M_z \hat{k} \quad (2.35)$$

Berdasarkan hukum Newton kedua, sama seperti pada dinamika translasi rudal, penjumlahan semua momen eksternal yang bekerja pada

badan adalah sama dengan momentum sudut terhadap waktu. Dengan demikian, persamaan dinamika rotasi rudal secara umum bisa didapatkan seperti pada Persamaan (2.36) dan (2.37).

$$\mathbf{M}_b = \mathbf{I}\boldsymbol{\omega}_b + \boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b \quad (2.36)$$

$$\frac{d\boldsymbol{\omega}_b}{dt} = \mathbf{I}^{-1}(\mathbf{M}_b - \boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b) \quad (2.37)$$

Dimana:

$\boldsymbol{\omega}_b$ = Kecepatan sudut rotasi rudal

\mathbf{I} = Momen inersia rudal

\mathbf{M}_b = Momen rotasi rudal

Momen inersia pada objek dengan tiga sumbu dalam sistem koordinatnya memiliki persamaan berbentuk matrix seperti pada Persamaan (2.38).

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \quad (2.38)$$

Dari matriks inersia tersebut elemen diagonalnya merupakan momen inersia aksial, sedangkan elemen selain diagonalnya disebut sebagai produk inersia. Semua produk inersia tersebut memiliki nilai nol dikarenakan sumbu utama sistem koordinat rudal berimpit dengan sumbu sistem koordinat rudal sehingga pada matriks inersia hanya tersisa momen inersia aksial. Oleh karena itu, momen inersia pada rudal dapat dinyatakan pada Persamaan (2.39).

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.39)$$

Momen inersia aksial dapat ditentukan dengan mengambil asumsi bahwa rudal merupakan suatu benda padat berbentuk tabung dan massanya terdistribusi secara rata. Melalui pendekatan objek tabung ini, matriks inersia rudal dapat ditentukan dengan rumus pada Persamaan (2.40).

$$\mathbf{I} = \begin{bmatrix} \frac{1}{8}md^2 & 0 & 0 \\ 0 & \frac{1}{12}ml^2 + \frac{1}{16}md^2 & 0 \\ 0 & 0 & \frac{1}{12}ml^2 + \frac{1}{16}md^2 \end{bmatrix} \quad (2.40)$$

Dimana:

- m = Massa rudal
 d = Diameter rudal
 l = Panjang rudal

Setelah mendapatkan matriks inersia rudal pada Persamaan (2.40) maka bisa *cross product* ($\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b$) pada Persamaan (2.37) dapat dijabarkan ke dalam Persamaan (2.41) hingga Persamaan (2.43).

$$\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.41)$$

$$\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ p & q & r \\ I_{xx}p & I_{yy}q & I_{zz}r \end{bmatrix} \quad (2.42)$$

$$\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b = \begin{bmatrix} (I_{zz} - I_{yy})qr \mathbf{i} \\ (I_{xx} - I_{zz})pr \mathbf{j} \\ (I_{xx} - I_{yy})pq \mathbf{k} \end{bmatrix} \quad (2.43)$$

Terdapat tiga macam momen yang bekerja pada sumbu referensi badan rudal, yaitu momen *rolling* (M_X), momen *pitching* (M_Y), momen *yawing* (M_Z). Adanya momen *rolling* diakibatkan oleh perbedaan gaya angkat (*lift*) di *aileron* pada sumbu longitudinal rudal dan nilai positif dari momen *roll* adalah ketika rudal melakukan *roll* ke arah kanan. Momen *pitching* disebabkan oleh gaya angkat (*lift*) dan gaya seret (*drag*) yang bekerja di sumbu lateral rudal dan momen *pitching* positif adalah ketika arah hadap rudal bergerak ke atas. Sedangkan momen *yawing* terjadi di sumbu vertikal rudal, dimana nilai yang positif adalah ketika hidung rudal berotasi ke arah kanan. Momen yang bekerja pada rudal berasal dari adanya momen aerodinamika rudal, dimana momen-momen tersebut terbagi menjadi momen yang diciptakan oleh beban aerodinamik yang terdistribusi serta gaya dorong yang tidak bekerja di titik pusat gravitasi atau *center of gravity* (*cog*). Momen yang bekerja pada rudal diberikan pada Persamaan (2.44) hingga Persamaan (2.46)

$$M_X = qSdC_R \quad (2.44)$$

$$M_Y = qSdC_P \quad (2.45)$$

$$M_Z = qSdC_Y \quad (2.46)$$

Dimana:

- q = Tekanan dinamis udara
 S = Luas permukaan rudal

- d = Diameter rudal
- C_R = Koefisien momen aerodinamika pada sumbu x
- C_P = Koefisien momen aerodinamika pada sumbu y
- C_Y = Koefisien momen aerodinamika pada sumbu z

Koefisien momen aerodinamika C_R, C_P dan C_Y merupakan fungsi dari *angle of attack* (α), *sideslip* (β), sudut defleksi *aileron* (δ_a), sudut defleksi *elevator* (δ_e), sudut defleksi *rudder* (δ_r), dan kecepatan sudut rudal (p, q, r). Persamaan koefisien momen aerodinamik ini dinyatakan rumus pada Persamaan (2.47) hingga (2.49).

$$C_R = C_{R_1}\alpha + C_{R_2}\beta + C_{R_3}p + C_{R_4}\delta_a \quad (2.47)$$

$$C_P = C_{P_1}\alpha + C_{P_2}q + C_{P_3}\delta_e \quad (2.48)$$

$$C_Y = C_{Y_1}\beta + C_{Y_2}r + C_{Y_3}\delta_r \quad (2.49)$$

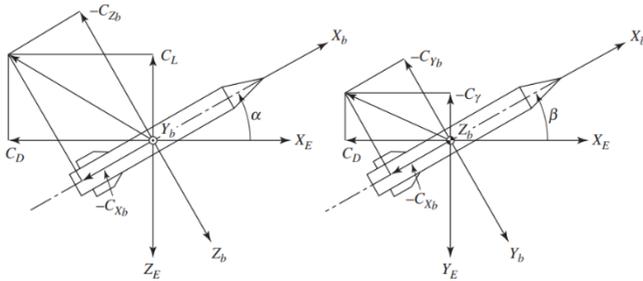
Untuk mencari model matematik kompak dinamika rotasi dari rudal, dapat dilakukan dengan menggabungkan Persamaan (2.35) hingga Persamaan (2.49) seperti yang dapat dilihat pada Persamaan (2.50) hingga Persamaan (2.52)

$$\frac{dp}{dt} = \frac{1}{I_{xx}} (qSd(C_{R_1}\alpha + C_{R_2}\beta + C_{R_3}p + C_{R_4}\delta_a) - (I_{zz} - I_{yy})qr) \quad (2.50)$$

$$\frac{dq}{dt} = \frac{1}{I_{yy}} (\bar{q}Sd(C_{P_1}\alpha + C_{P_2}q + C_{P_3}\delta_e) - (I_{xx} - I_{zz})pr) \quad (2.51)$$

$$\frac{dr}{dt} = \frac{1}{I_{zz}} (\bar{q}Sd(C_{Y_1}\beta + C_{Y_2}r + C_{Y_3}\delta_r) - (I_{yy} - I_{xx})pr) \quad (2.52)$$

Secara keseluruhan, ilustrasi diagram vektor koefisien aerodinamik yang bekerja pada rudal dapat dilihat pada Gambar (2.5) dan tabel komponen-komponen momen aerodinamik, koordinat, dan kecepatan rudal dapat dilihat pada Tabel (2.1).



Gambar 2.5 Diagram vektor koefisien aerodinamik rudal

Tabel 2.1 Komponen aerodinamik, koordinat, dan kecepatan rudal

	Sumbu x (<i>Roll</i>)	Sumbu y (<i>Pitch</i>)	Sumbu z (<i>Yaw</i>)
Kecepatan sudut	p	q	r
Kecepatan translasi	u	v	w
Gaya aerodinamis	F_x	F_y	F_z
Koefisien gaya aerodinamis	C_D	C_S	C_L
Koefisien momen aerodinamis	C_R	C_P	C_Y

2.4. Dubins Curve

Metode *Dubins curve* merupakan salah satu teknik untuk merancang trayektori atau lintasan. Trayektori deterministik yang dihasilkan dari *Dubins curve* ini kemudian akan menjadi trayektori yang akan dilalui oleh rudal dari sejak titik keadaan awal sampai dengan titik terakhir dari lintasan yang telah diketahui. Pada metode *Dubins curve* ini, trayektori yang dibangun terdiri dari gabungan trayektori garis lurus dan busur lingkaran atau *arc*. Lintasan *Dubins* terpendek dapat dicapai ketika radius busur lingkaran minimum dan garis merupakan garis lurus. Dalam geometri *euclidean*, *Dubins curve* dihasilkan dari menarik garis singgung atau *tangent* antar 2 busur lingkaran. Garis singgung ini dapat menghubungkan 2 busur lingkaran secara eksternal maupun internal (diagonal).

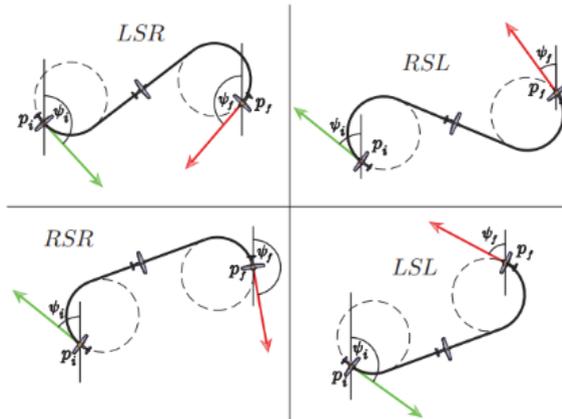
Tahap pertama pada metode *Dubins curve* adalah menentukan tipe lintasan apa yang akan digunakan. Terdapat empat macam kombinasi lintasan dari *Dubins curve* untuk bisa membangun trayektori yang diinginkan, dimana masing-masing kombinasi tersusun dari 2 segmen melingkar dan 1 garis lurus seperti dapat dilihat pada Gambar (2.6).

Dapat dilihat pula dari Gambar (2.6) bahwa 4 macam lintasan *Dubins* adalah LSR, RSL, RSR dan LSL. LSR merupakan lintasan *left straight right* yang sesuai namanya merupakan lintasan berbelok kiri, kemudian lurus, diakhiri dengan dengan belok kanan. RSL adalah lintasan *right*

straight left yaitu lintasan kanan, lurus, diakhiri dengan belok kiri. RSR adalah lintasan *right straight right* yaitu lintasan berbelok kanan, lurus, dan kanan. LSL merupakan *left straight left* dimana lintasan yang terbentuk adalah berbelok kiri, kemudian lurus, dan diakhiri belokan ke kiri.

Setelah menentukan bentuk lintasan yang diinginkan, kemudian mendapatkan konfigurasi awal dan akhir dari objek bergerak, yang dalam penelitian ini merupakan rudal. Konfigurasi yang dimaksud adalah posisi awal P_s dan akhir P_f . Akan diambil lintasan RSR sebagai contoh langkah-langkah membangun lintasan *Dubins* seperti pada Gambar (2.7).

1. Menentukan posisi awal P_s dan posisi akhir P_f dalam koordinat (x, y) , sudut arah atau *heading* rudal (θ) , dan radius putar minimum ρ . Secara matematis, konfigurasi awal dan akhir adalah $P_s(x_s, y_s, \theta_s)$ dan $P_f(x_f, y_f, \theta_f)$.



Gambar 2.6 Kombinasi Lintasan pada *Dubins curve* [5]

2. Menentukan titik pusat dari lingkaran awal $O_s(x_{cs}, y_{cs})$ dan akhir $O_f(x_{cf}, y_{cf})$ dengan jari-jari minimum dengan arah putaran searah jarum jam atau ke arah kanan menggunakan rumus pada Persamaan (2.53) dan (2.54).

$$(x_{cs}, y_{cs}) = \left(x_s \pm \rho_s \cos \left(\theta_s \pm \frac{\pi}{2} \right), y_s \pm \rho_s \sin \left(\theta_s \pm \frac{\pi}{2} \right) \right) \quad (2.53)$$

$$(x_{cf}, y_{cf}) = \left(x_f \pm \rho_f \cos \left(\theta_f \pm \frac{\pi}{2} \right), y_f \pm \rho_f \sin \left(\theta_f \pm \frac{\pi}{2} \right) \right) \quad (2.54)$$

3. Mencari jarak antar titik pusat lingkaran menggunakan Persamaan (2.55).

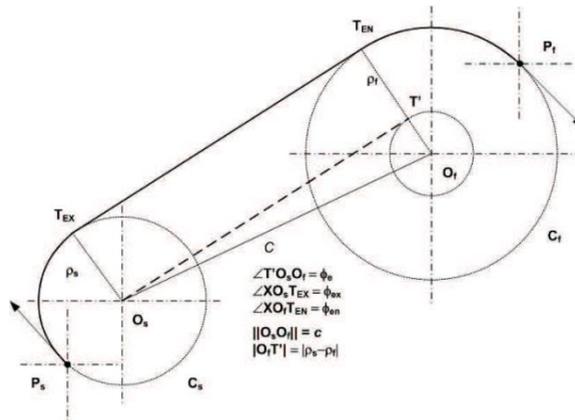
$$|c| = \sqrt{(x_{cs} - x_{cf})^2 + (y_{cs} - y_{cf})^2} \quad (2.55)$$

4. Menghitung sudut kemiringan dari garis antar titik pusat lingkaran menggunakan rumus pada Persamaan (2.56).

$$\psi = \arctan \left(\frac{y_{cf} - y_{cs}}{x_{cf} - x_{cs}} \right) \quad (2.56)$$

Dan sudut antara garis $O_s O_f$ dan $O_s T'$ menggunakan Persamaan (2.57).

$$\phi_e = \arcsin \left(\frac{\rho_f - \rho_s}{|c|} \right) \quad (2.57)$$



Gambar 2.7 Lintasan Dubins RSR [1]

5. Menentukan titik T_{ex} (*tangent exit*) yang merupakan titik akhir dari lintasan busur lingkaran awal sebelum memasuki lintasan garis lurus serta titik T_{en} (*tangent entry*) yang merupakan titik awal dari lintasan busur lingkaran akhir setelah lintasan garis lurus menggunakan rumus pada Persamaan (2.58) dan (2.59).

$$T_{ex} = (x_{cs} + \rho_s \cos(\phi_{ex}), y_{cs} + \rho_s \sin(\phi_{ex})) \quad (2.58)$$

$$T_{en} = (x_{cf} + \rho_f \cos(\phi_{en}), y_{cf} + \rho_f \sin(\phi_{en})) \quad (2.59)$$

Dimana ϕ_{ex} merupakan sudut *exit* dan ϕ_{en} adalah sudut *entry*. Kedua sudut ini merupakan sudut hadap rudal ketika berada di titik T_{ex} dan T_{en} dan dapat didapatkan melalui rumus pada Persamaan (2.60) dan (2.61).

$$\phi_{ex} = \phi_{es} + \frac{\pi}{2} + \psi \quad (2.60)$$

$$\phi_{en} = \phi_{ef} + \frac{\pi}{2} + \psi \quad (2.61)$$

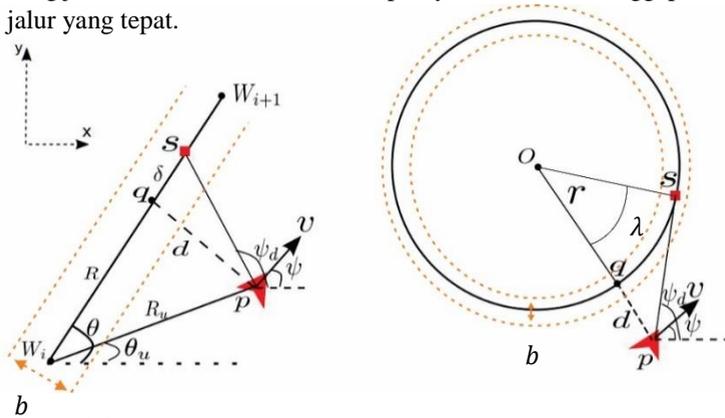
6. Hubungkan titik P_s dan T_{ex} menggunakan busur lingkaran dengan jari-jari ρ_s serta titik P_f dan T_{en} menggunakan busur lingkaran dengan jari-jari ρ_f .
7. Lintasan RSR yang telah terbentuk merupakan gabungan dari busur lingkaran awal $P_s T_{ex}$, dilanjutkan dengan garis singgung luar $T_{ex} T_{en}$ dan diakhiri dengan lintasan busur lingkaran akhir $T_{en} P_f$.

Semua sudut yang digunakan merupakan positif apabila berlawanan dengan arah jarum jam. Pada lintasan LSL, prosedur yang sama seperti RSR dapat digunakan, hanya saja memiliki perbedaan arah lintasan busur lingkaran yang berlawanan dengan jarum jam atau ke arah kiri. Pada lintasan RSL dan LSR, garis singgung yang didapat antara lingkaran awal dan lingkaran akhir adalah garis singgung dalam atau internal (diagonal).

2.5. Metode Pemanduan *Sliding Curve*

Metode *sliding curve* merupakan suatu metode *trajectory following* yang memanfaatkan informasi sudut arah hadap (*heading*) rudal serta adanya sasaran *virtual* yang ikut bergerak sepanjang lintasan *sliding surface* sehingga dapat digunakan untuk memandu pergerakan rudal agar bisa mengikuti trayektori yang telah ditentukan. Tujuan utama dari metode pemanduan *sliding curve* ini adalah meminimumkan *cross-track error* atau selisih jarak antara posisi rudal setiap saat terhadap trayektori yang ada serta menyamakan sudut arah hadap rudal dengan sudut lintasan. Dengan *cross-track error* yang minimum serta sudut arah hadap rudal yang sama dengan trayektori, maka akan didapatkan pergerakan rudal yang mengikuti trayektori tersebut. Metode dari *sliding curve* ini dikembangkan dari metode *carrot-chasing* [4], dimana algoritma dasarnya menyerupai hanya saja kurva lintasan yang menjadi acuan

memiliki toleransi tertentu sehingga apabila rudal masih berada pada rentang jarak toleransi tersebut terhadap trayektori masih dianggap berada di jalur yang tepat.



Gambar 2.8 Pemanduan Sliding Curve [4]

Algoritma pemanduan *sliding curve* ini terpisah menjadi 2 macam yaitu untuk lintasan garis lurus dan lintasan lingkaran atau *loiter* seperti pada Gambar (2.8) ψ adalah sudut *heading* atau arah hadap rudal, d adalah *cross-track error*, W_i adalah *waypoint* saat ke i , W_{i+1} adalah *waypoint* saat ke $i + 1$, q adalah titik tegak lurus dari posisi rudal ke trayektori, b adalah lebar *sliding surface*, s adalah sasaran *virtual*, dan δ adalah parameter jarak yang mengatur jauh dekatnya titik s . Dapat dilihat juga bahwa metode *sliding curve* ini menggerakkan sasaran virtual dari satu *waypoint* ke *waypoint* selanjutnya, sehingga lintasan yang akan diikuti merupakan susunan dari titik-titik sepanjang trayektori. Susunan titik-titik yang berada di sepanjang trayektori dapat dikatakan sebagai *waypoint virtual*, karena *waypoint* sesungguhnya yang harus dilewati rudal hanyalah 4 buah titik sehingga titik-titik tersebut haruslah melalui *waypoint* yang sesungguhnya. Perlu diperhatikan untuk pemilihan δ sebaiknya tidak melebihi jarak antara *waypoint*. Adapun langkah-langkah yang harus dilakukan pada metode pemanduan *sliding curve* adalah sebagai berikut:

Lintasan Garis Lurus

1. Menentukan posisi *waypoint* saat ke i dan saat ke $i + 1$, serta konfigurasi posisi dan arah hadap rudal awal.

$$W_i = (x_i, y_i) \quad (2.62)$$

$$W_{i+1} = (x_{i+1}, y_{i+1}) \quad (2.63)$$

$$p = (x_i, y_i), \psi \quad (2.64)$$

2. Menghitung jarak antara posisi sekarang terhadap W_i dan sudut kemiringan garis (θ) yang menghubungkan W_i dengan W_{i+1} .

$$R_u = ||W_i - p|| \quad (2.65)$$

$$\theta = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i) \quad (2.66)$$

3. Mencari sudut antara W_i dengan posisi sekarang serta sudut β

$$\theta_u = \text{atan2}(y - y_i, x - x_i) \quad (2.67)$$

$$\beta = \theta - \theta_u \quad (2.68)$$

4. Mencari jarak dari W_i ke titik q

$$R = \sqrt{R_u^2 - (R_u \sin(\beta))^2} \quad (2.69)$$

5. Melakukan *update* dari posisi sasaran atau *virtual target*

$$(x'_t, y'_t) \leftarrow ((R + \delta)\cos\theta, (R + \delta)\sin\theta) \quad (2.70)$$

6. Menghitung sudut arah hadap yang harus dicapai rudal untuk mengejar sasaran virtual

$$\psi_d = \text{atan2}(y'_t - y, x'_t - x) \quad (2.71)$$

7. Memberikan aksi kontrol untuk membelokkan arah hadap rudal

$$u = k(\psi_d - \psi) \quad (2.72)$$

Sinyal kontrol u diasumsikan sebagai $\dot{\psi}$ yaitu kecepatan sudut *yaw* dari rudal. Sinyal kontrol ini menggunakan kontroler proporsional dengan gain $k > 0$. Sedangkan untuk lintasan *loiter* atau lingkaran adalah sebagai berikut:

1. Menentukan titik pusat lingkaran, jari-jari lingkaran, posisi dan sudut arah hadap rudal kondisi awal.

$$O = (x_i, y_i), r \quad (2.73)$$

$$p = (x_i, y_i), \psi \quad (2.74)$$

2. Menghitung jarak orthogonal dari posisi rudal saat ini terhadap lingkaran

$$d = ||O - p|| - r \quad (2.75)$$

3. Mencari sudut antara titik posisi sekarang dengan titik sasaran *virtual*

$$\theta = \text{atan2}(y - y_i, x - x_i) \quad (2.76)$$

4. Melakukan *update* posisi sasaran *virtual*

$$(x'_t, y'_t) \leftarrow (r\cos(\theta + \lambda), r\sin(\theta + \lambda)) \quad (2.77)$$

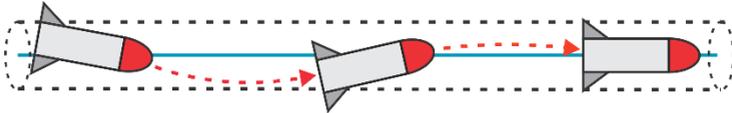
5. Menghitung sudut arah hadap yang harus dicapai rudal untuk mengejar sasaran virtual

$$\psi_d = \text{atan2}(y'_t - y, x'_t - x) \quad (2.78)$$

6. Memberikan aksi kontrol untuk membelokkan arah hadap rudal

$$u = k(\psi_d - \psi) \quad (2.79)$$

Kedua algoritma di atas merupakan algoritma untuk pemanduan *yaw* (ψ) pada rudal, atau dalam sumbu kartesian adalah untuk bidang x - y . Pada sistem pemanduan rudal ini, akan digunakan juga algoritma yang sama untuk pemanduan *pitch* (θ) atau bidang x - z . Gerakan *roll* diasumsikan tidak berubah selama rudal bergerak di udara. Ilustrasi pergerakan rudal dengan pemanduan *sliding curve* di dalam selubung *sliding surface* dapat dilihat pada Gambar (2.9).



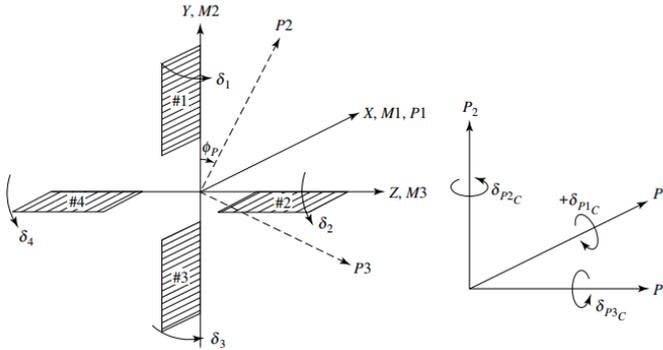
Gambar 2.9 Pergerakan rudal di dalam selubung *sliding surface*

2.6. Sistem *Autopilot* Rudal

Sistem *autopilot* rudal merupakan suatu sistem *loop* tertutup yang berada dalam sub-sistem pemanduan rudal. Sistem *autopilot* rudal ini bertujuan untuk memastikan bahwa rudal mencapai sudut orientasi yang sesuai dengan yang diperintahkan serta menjaga kestabilan sistem rudal. Sistem *autopilot* terdiri dari *autopilot roll*, *pitch*, dan *yaw*. Sistem *autopilot* rudal bisa menstabilkan rudal dan mencapai sudut orientasi referensi dengan mengatur defleksi sirip-sirip rudal, yang menyebabkan badan rudal untuk berotasi dan bertranslasi. Masukan dari sistem *autopilot* adalah informasi sudut referensi dari sistem pemanduan rudal dan keluarannya adalah sudut defleksi sirip rudal (δ) untuk bisa mencapai sasaran posisi tertentu.

Dalam mengendalikan defleksi sirip rudal, dapat dilihat pada gambar 3 bahwa terdapat 4 buah sirip yang dapat dikendalikan menggunakan aktuator motor *servo*. Apabila sirip 2 dan 4 terhubung secara mekanik maka keduanya dapat disebut sebagai *elevator*, sedangkan apabila masing-masing sirip 2 dan 4 memiliki servo tersendiri maka disebut sebagai *aileron*. Sedangkan pada sirip 1 dan 3 disebut sebagai *rudder* apabila keduanya terhubung secara mekanik. Aktuator

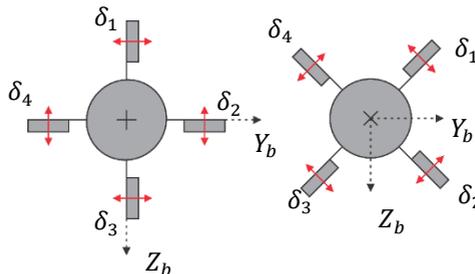
motor *servo* yang digunakan pada rudal akan diasumsikan memiliki respons orde dua seperti pada Persamaan (2.80).



Gambar 2.10 Defleksi Sirip Rudal [2]

$$\delta_{actual} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \delta_{cmd} \quad (2.80)$$

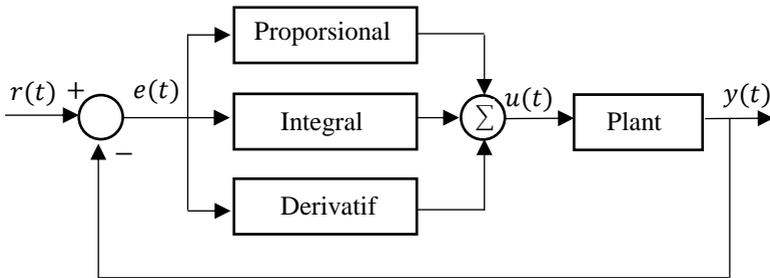
Konfigurasi dari 4 buah sirip rudal juga dapat dibedakan menjadi konfigurasi *plus* dan konfigurasi *cross* seperti yang dapat dilihat pada Gambar (2.11). Sesuai dengan namanya, konfigurasi sirip *plus* memiliki bentuk seperti tanda “+” dan konfigurasi sirip *cross* memiliki bentuk menyilang seperti tanda “×”. Pada konfigurasi *plus*, dua sirip rudal horizontal (*elevator*) akan mengendalikan sudut *pitch* rudal dan dua sirip vertikal (*rudder*) untuk mengendalikan sudut *yaw* rudal. Berbeda dengan konfigurasi *plus*, pada konfigurasi *cross* beban kerja masing-masing sirip rudal akan terbagi secara merata.



Gambar 2.11 Konfigurasi Sirip Rudal *plus* (kiri) dan *cross* (kanan)

2.7. Kontroler PID

Kontroler proporsional integral derivatif atau yang biasa disingkat menjadi kontroler PID merupakan kontroler yang memiliki tiga komponen utama dan masing-masing memiliki nilai *gain* tersendiri yaitu komponen proporsional (K_p), komponen integral (K_i), dan komponen derivatif (K_d). Masing-masing komponen ini memiliki peran masing-masing untuk bisa memberikan sinyal keluaran yang akan menjadi sinyal perintah kepada aktuator sirip rudal. Diagram blok kontroler PID secara umum dapat dilihat pada Gambar (2.12).



Gambar 2.12 Diagram blok kontroler PID

Sedangkan untuk kalkulasi sinyal keluaran dari kontroler PID dapat dilihat pada Persamaan (2.81).

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.81)$$

dimana:

$u(t)$ = Sinyal kontrol

$e(t)$ = Sinyal error

K_p = *Gain* proporsional

K_i = *Gain* integral

K_d = *Gain* Derivatif

Dalam kontroler PID ini, apabila nilai K_p dinaikkan maka *rise time* dan *error steady state* akan berkurang namun *overshoot* akan bertambah. *Settling time* hanya akan mengalami sedikit perubahan ketika terjadi perubahan nilai *gain* K_p . Hampir sama seperti K_p , pada saat nilai *gain* K_i dinaikkan maka *rise time* dan *error steady state* akan berkurang namun

overshoot akan bertambah. Tetapi pada saat nilai K_i dinaikkan, nilai *settling time* akan bertambah pula. Sedangkan pada saat nilai K_d dinaikkan maka nilai *overshoot* dan *settling time* akan berkurang. Perubahan pada nilai *gain* K_d hanya akan mengubah sedikit nilai *rise time* dan tidak mempengaruhi besarnya *error steady state*. Karakteristik dari masing-masing komponen kontroler PID dapat dilihat pada Tabel (2.2).

Tabel 2.2 Karakteristik respon kontroler PID

	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Error Steady State</i>
K_p	Berkurang	Bertambah	Sedikit Perubahan	Berkurang
K_i	Berkurang	Bertambah	Bertambah	Berkurang
K_d	Sedikit Perubahan	Berkurang	Berkurang	Tidak Berubah

BAB 3

PERANCANGAN SISTEM

3.1. Gambaran Umum Sistem

Secara umum, sistem yang akan dirancang secara keseluruhan terbagi menjadi 4 tahap perancangan. Perancangan yang pertama adalah perancangan model rudal dengan menggunakan persamaan matematik yang telah didapatkan. Tahap selanjutnya adalah merancang *Dubins curve* yang akan menjadi trayektori pergerakan rudal. *Dubins curve* akan menghasilkan trayektori referensi berupa titik-titik yang berurutan sebagai *waypoint virtual* yang harus dilalui oleh rudal sedangkan *waypoint utamanya* terdapat 4 titik, dimana *waypoint virtual* yang dibangkitkan dari *Dubins curve* tersebut harus melalui titik *waypoint* sesungguhnya. Setelah dibangun trayektori yang diinginkan, maka dilanjutkan dengan perancangan sistem pemanduan rudal.

Pada perancangan sistem pemanduan rudal ini menggunakan algoritma *sliding curve*, dimana masukan dari sistem pemanduan adalah titik-titik *waypoint*, posisi rudal, dan sudut orientasi rudal. Keluaran dari sistem pemanduan ini merupakan sudut orientasi referensi yang harus dicapai agar rudal bisa mengikuti trayektori yang ada. Pada tugas akhir ini, gerakan *roll* pada rudal diasumsikan terjaga tetap bernilai 0, sehingga yang akan menjadi keluaran dari sistem pemanduan hanyalah sudut *pitch* (θ) dan *yaw* (ψ). Informasi sudut *pitch* dan *yaw* referensi ini kemudian diterima oleh sistem *autopilot* yang merupakan tahap perancangan sistem selanjutnya. Pada perancangan sistem *autopilot*, akan diimplementasikan kontroler *cascade* P-PI yang dapat mengubah sinyal masukan berupa sudut referensi menjadi sinyal kontrol kepada aktuator, yaitu *rudder* dan *elevator* rudal.

3.2. Perancangan Model Rudal

Perancangan sistem secara keseluruhan dimulai dengan merancang model matematik rudal yang terdiri dari model dinamika dan kinematika rudal. Persamaan matematik yang telah didapatkan pada Bab sebelumnya akan digunakan sebagai model yang merepresentasikan rudal berjenis *skid-to-turn*. Model dinamika rudal yang tersusun oleh dinamika translasi dan dinamika rotasi adalah dalam sistem koordinat badan rudal sehingga untuk melakukan transformasi ke sistem koordinat bumi dibutuhkan model kinematika rudal. Oleh karena itu, perancangan model rudal akan

dimulai dengan perancangan dinamika barulah dilanjutkan dengan perancangan kinematika.

Pada perancangan dinamika rudal dengan enam derajat kebebasan ini dibangun berdasarkan persamaan yang didapatkan menggunakan hukum Newton kedua yaitu jumlahan dari gaya-gaya yang bekerja pada tiap sumbu sistem koordinat rudal. Gaya-gaya yang bekerja terdiri dari gaya aerodinamika, gaya gravitasi, dan gaya hasil interaksi antara kecepatan translasi rudal dengan kecepatan sudut rudal. Persamaan gaya translasi yang didapatkan adalah seperti pada Persamaan (3.1) hingga (3.3).

$$F_x = T - mg \sin \theta + qSC_D \quad (3.1)$$

$$F_y = mg \sin \phi \cos \theta + qSC_S \quad (3.2)$$

$$F_z = mg \cos \phi \cos \theta + qSC_L \quad (3.3)$$

dimana

$$q = \frac{1}{2} \rho V_m^2 \quad (3.4)$$

$$C_S = C_{D_1} \alpha + C_{D_2} M \quad (3.5)$$

$$C_S = C_{S_1} \beta + C_{S_2} M + C_{S_3} \delta_r \quad (3.6)$$

$$C_L = C_{L_1} \alpha + C_{L_2} M + C_{L_3} \delta_e \quad (3.7)$$

Sedangkan untuk dinamika rotasi merupakan jumlahan dari momen-momen rotasi yang bekerja pada rudal dan persamaannya bisa dinyatakan pada Persamaan (3.8) hingga (3.10).

$$M_x = qSdC_R \quad (3.8)$$

$$M_y = qSdC_P \quad (3.9)$$

$$M_z = qSdC_Y \quad (3.10)$$

dimana

$$C_R = C_{R_1} \alpha + C_{R_2} \beta + C_{R_3} p + C_{R_4} \delta_a \quad (3.11)$$

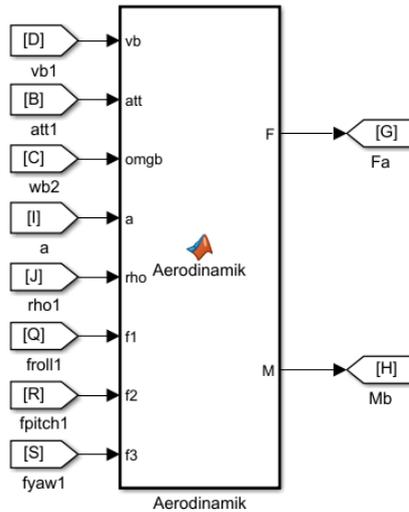
$$C_P = C_{P_1} \alpha + C_{P_2} q + C_{P_3} \delta_e \quad (3.12)$$

$$C_Y = C_{Y_1} \beta + C_{Y_2} r + C_{Y_3} \delta_r \quad (3.13)$$

Persamaan-persamaan dinamika translasi dan rotasi yang telah didapatkan kemudian akan dimasukkan ke dalam satu blok fungsi pada Matlab. Gambar (3.1) merupakan rancangan blok fungsi untuk menghitung komponen gaya dan momen aerodinamik seperti pada Persamaan (3.14) dan (3.15).

$$\mathbf{F}_a = \begin{bmatrix} F_{ax} \\ F_{ay} \\ F_{az} \end{bmatrix} = \begin{bmatrix} qsC_D \\ qsC_S \\ qsC_L \end{bmatrix} \quad (3.14)$$

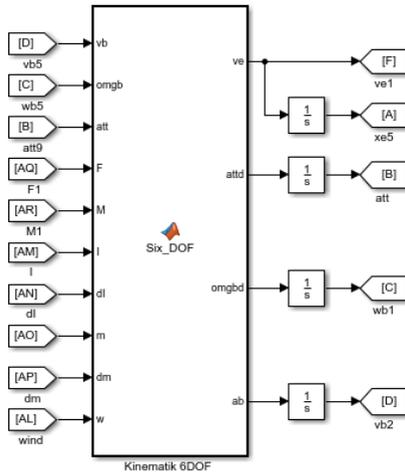
$$\mathbf{M}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} qsdC_R \\ qsdC_P \\ qsdC_Y \end{bmatrix} \quad (3.15)$$



Gambar 3.1 Blok fungsi komponen gaya dan momen aerodinamik

Pada blok fungsi komponen gaya dan momen aerodinamik ini, masukannya adalah kecepatan translasi rudal, kecepatan sudut rudal, sudut orientasi rudal pada sistem koordinat badan rudal, kecepatan suara, dan massa jenis udara. Semua masukan ini kemudian digunakan untuk mendapatkan keluaran berupa gaya (\mathbf{F}_a) dan momen (\mathbf{M}_b) aerodinamik.

Setelah merancang blok fungsi untuk gaya dan momen aerodinamik, maka dilanjutkan dengan merancang blok fungsi untuk seluruh gaya dan momen yang bekerja pada rudal. Blok fungsi ini juga akan menjadi satu dengan kinematika, dimana persamaan kinematika rudal yang akan digunakan adalah pada Persamaan (2.9) dan (2.10). Hasil perancangan blok fungsi ini dapat dilihat pada Gambar (3.2).

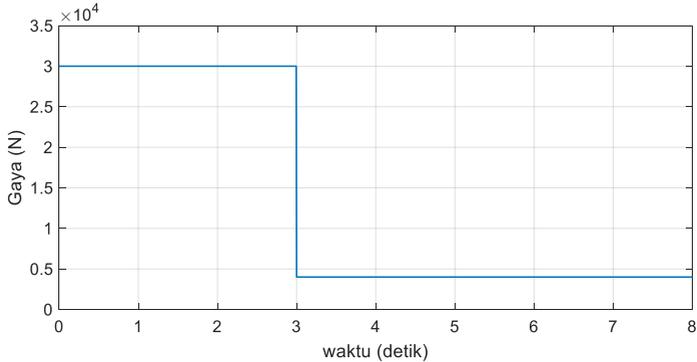


Gambar 3.2 Blok fungsi gabungan dinamika dan kinematika rudal

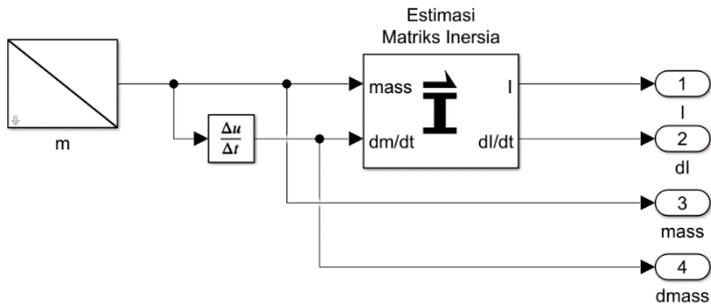
Masukan pada blok fungsi ini adalah kecepatan translasi rudal, kecepatan sudut rudal, sudut orientasi rudal, gaya dan momen total yang bekerja pada rudal, matriks inersia beserta perubahannya terhadap waktu ketika massa rudal berkurang, massa rudal beserta perubahan massa rudal, dan kecepatan angin pada atmosfer tempat rudal bergerak. Gaya dan momen total yang menjadi masukan pada blok fungsi ini merupakan gaya dan momen aerodinamik pada blok fungsi aerodinamik dengan gaya *thruster* atau pendorong rudal. Model *thruster* yang digunakan pada rudal terdiri dari dua fase. Fase pertama adalah kondisi ketika *booster* dan *sustainer* beroperasi bersamaan sedangkan fase kedua adalah fase *sustain* yaitu ketika rudal hanya didorong oleh *sustainer*. Fase pertama dimulai ketika rudal baru diluncurkan dan berakhir setelah fase *booster* selesai yaitu selama 3 detik pertama. Pada pengujian ini gaya dorong *thruster* adalah 30000 Newton dan pada fase *sustain* sebesar 4000 Newton. Grafik keluaran gaya dorong *thruster* yang digunakan dapat dilihat pada Gambar (3.3).

Perhitungan matriks inersia, massa rudal, perubahan massa rudal, dan perubahan inersia selama rudal bergerak di udara dilakukan pada blok fungsi massa dan inersia seperti pada Gambar (3.4). Pada blok fungsi ini masukannya berupa massa rudal yang akan berkurang terhadap waktu seiring rudal bergerak di udara. Berdasarkan massa tetap rudal serta

perubahannya maka bisa didapatkan estimasi matriks inersia dan perubahan matriks inersia.

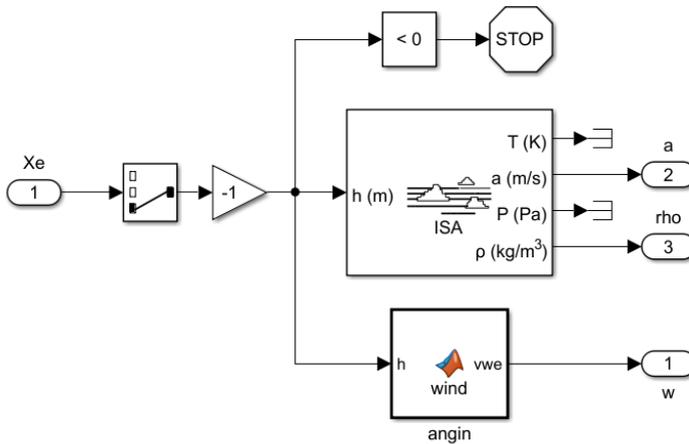


Gambar 3.3 Grafik gaya dorong *thruster* rudal



Gambar 3.4 Blok fungsi perhitungan massa dan inersia

Setelah merancang blok fungsi massa dan inersia, maka akan dilanjutkan dengan blok fungsi lingkungan atmosfer yang akan memberi masukan berupa kecepatan angin pada blok fungsi dinamika rudal serta massa jenis udara dan kecepatan suara dengan kondisi atmosfer sekitar kepada blok fungsi aerodinamik. Blok fungsi model lingkungan rudal dapat dilihat pada Gambar (3.5).



Gambar 3.5 Blok fungsi model lingkungan rudal

Pada blok fungsi model lingkungan rudal ini digunakan *library* dari *software* Simulink yaitu *International Standard Atmosphere (ISA)*. Keluaran dari model lingkungan ISA ini adalah cepat rambat suara di udara dengan kondisi lingkungan rudal serta massa jenis udara pada lingkungan rudal. Kecepatan angin yang akan diterapkan hanyalah kecepatan angin pada sumbu x dengan arah yang berlawanan dengan arah gerak rudal. Selain itu pada blok fungsi model lingkungan rudal juga diberi logika simulasi harus berhenti ketika ketinggian rudal kurang dari nol atau rudal jatuh ke permukaan bumi.

Keluaran utama dari blok fungsi gabungan kinematika dan dinamika rudal sesuai pada Gambar (3.2) adalah kecepatan translasi dan kecepatan sudut rudal terhadap sistem koordinat bumi, percepatan sudut dan percepatan translasi rudal terhadap sistem koordinat badan rudal. Dari keluaran blok fungsi tersebut kemudian bisa didapatkan posisi rudal terhadap sistem koordinat bumi dengan melakukan integral kecepatan translasi rudal. Demikian juga bisa didapatkan sudut orientasi rudal terhadap koordinat bumi, kecepatan sudut dan kecepatan translasi rudal terhadap koordinat badan rudal dengan melakukan integral dari keluaran utama blok fungsi. Persamaan dinamika percepatan translasi rudal dalam sistem koordinat badan rudal bisa didapatkan dengan menggunakan

hukum Newton II yang hasilnya didapatkan seperti pada Persamaan (3.16) hingga (3.21).

$$\frac{du}{dt} = \frac{F_x}{m} + rv - qw \quad (3.16)$$

$$\frac{dv}{dt} = \frac{F_y}{m} + pw - ru \quad (3.17)$$

$$\frac{dw}{dt} = \frac{F_z}{m} + qu - pv \quad (3.18)$$

dimana

$$C_D = C_{D_1}\alpha + C_{D_2}M \quad (3.19)$$

$$C_S = C_{S_1}\beta + C_{S_2}M + C_{S_3}\delta_r \quad (3.20)$$

$$C_L = C_{L_1}\alpha + C_{L_2}M + C_{L_3}\delta_e \quad (3.21)$$

Sedangkan percepatan rudal dalam sistem koordinat bumi bisa didapatkan dengan menggunakan Persamaan (3.22) hingga (3.24).

$$a_x = \frac{F_x - \dot{m}u}{m} \quad (3.22)$$

$$a_y = \frac{F_y - \dot{m}v}{m} \quad (3.23)$$

$$a_z = \frac{F_z - \dot{m}w}{m} \quad (3.24)$$

Sama seperti dinamika translasi, untuk mendapatkan dinamika rotasi juga menggunakan hukum Newton II yang menyatakan bahwa momen-momen yang bekerja pada rudal dapat menyebabkan percepatan sudut pada rudal. Persamaan percepatan rotasi rudal dinyatakan dalam Persamaan (3.25) hingga (3.30)

$$\frac{dp}{dt} = \frac{1}{I_{xx}}(qSdC_R - (I_{zz} - I_{yy})qr) \quad (3.25)$$

$$\frac{dq}{dt} = \frac{1}{I_{yy}}(qSdC_P - (I_{xx} - I_{zz})pr) \quad (3.26)$$

$$\frac{dr}{dt} = \frac{1}{I_{zz}}(qSdC_Y - (I_{yy} - I_{xx})pr) \quad (3.27)$$

dimana

$$C_R = C_{R_1}\alpha + C_{R_2}\beta + C_{R_3}p + C_{R_4}\delta_a \quad (3.28)$$

$$C_P = C_{P_1}\alpha + C_{P_2}q + C_{P_3}\delta_e \quad (3.29)$$

$$C_Y = C_{Y_1}\beta + C_{Y_2}r + C_{Y_3}\delta_r \quad (3.30)$$

Untuk mendapatkan nilai *angle of attack* (α), *sideslip angle* (β), *mach number* (M), dan tekanan udara (q) dapat digunakan Persamaan (3.31) hingga Persamaan (3.34)

$$\alpha = \tan^{-1} \frac{w}{u} \quad (3.31)$$

$$\beta = \sin^{-1} \frac{v}{\sqrt{u^2 + v^2 + w^2}} = \sin^{-1} \frac{v}{V} \quad (3.32)$$

$$M = \frac{\sqrt{u^2 + v^2 + w^2}}{\text{kecepatan suara}} = \frac{V}{\text{kecepatan suara}} \quad (3.33)$$

$$\bar{q} = \frac{1}{2} \rho V_m^2 \quad (3.34)$$

Seluruh nilai parameter rudal seperti koefisien aerodinamis, massa, serta parameter lainnya yang digunakan pada dinamika rudal bisa dilihat pada Tabel (3.1). Nilai parameter ini didapatkan berdasarkan asumsi spesifikasi parameter rudal.

Tabel 3.1 Nilai parameter rudal yang digunakan

Parameter	Nilai	Satuan
Massa Rudal (m)	340	Kg
Diameter (d)	0.28	m
Luas Permukaan (S)	0.2622	m ²
Momen Inersia I_{xx}	2.8028	Kg m ²
Momen Inersia I_{yy}	323.2855	Kg m ²
Momen Inersia I_{zz}	323.2855	Kg m ²
Koefisien C_{D_1}	-0.35	-
Koefisien C_{D_2}	-0.04	-
Koefisien C_{S_1}	$-90/\pi$	-
Koefisien C_{S_2}	-0.025	-
Koefisien C_{S_3}	1.5	-
Koefisien C_{L_1}	$-90/\pi$	-
Koefisien C_{L_2}	-0.025	-
Koefisien C_{L_3}	1.5	-
Koefisien C_{R_1}	-0.1	-
Koefisien C_{R_2}	-0.1	-
Koefisien C_{R_3}	-1	-

Koefisien C_{R_4}	1.5	-
Koefisien C_{P_1}	$-120/\pi$	-
Koefisien C_{P_2}	-1	-
Koefisien C_{P_3}	1.5	-
Koefisien C_{Y_1}	$120/\pi$	-
Koefisien C_{Y_2}	-1	-
Koefisien C_{Y_3}	1.5	-

3.3. Perancangan Trayektori Dubins Curve

Langkah awal dari merancang trayektori adalah menentukan titik-titik *waypoint* (x_i, y_i, z_i) utama yang harus dilalui oleh rudal. Dari titik-titik *waypoint* tersebut kemudian saling dihubungkan dari *waypoint* pertama sampai dengan *waypoint* terakhir sehingga terbentuk kurva garis lurus. Setelah mendapatkan kurva garis lurus inilah titik-titik belok antar *waypoint* akan dihaluskan menggunakan metode *Dubins Curve* pendekatan geometris.

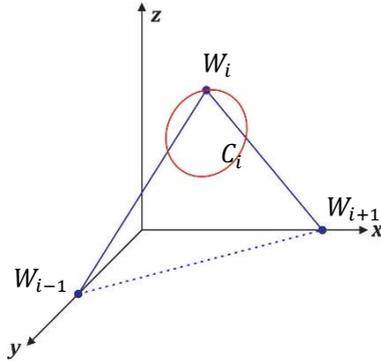
Untuk mendapatkan lintasan *Dubins Curve*, yang harus dilakukan pertama menentukan posisi-posisi lingkaran *Dubins*. Pada ruang 3 dimensi, lingkaran ini harus berada pada bidang yang sama dengan bidang yang dibentuk dari 3 buah titik yaitu titik *waypoint* ke $i - 1, i$ dan $i + 1$. Selain harus berada pada bidang tersebut, titik *waypoint* ke i juga harus berada di titik singgung lingkaran dan lingkaran harus dipotong oleh kedua garis yang mengapit titik *waypoint* tersebut. Ilustrasi dari peletakan lingkaran pada bidang 3 dimensi ini dapat dilihat pada Gambar (3.6).

Untuk membuat lingkaran yang berada pada bidang yang terbentuk dari 3 titik tersebut, pertama-tama akan dibuat lingkaran datar pada bidang xy dengan titik pusat pada W_i dengan $z = 0$ dan radius r menggunakan Persamaan (3.35).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_{W_i} \\ y_{W_i} \\ 0 \end{bmatrix} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \\ 0 \end{bmatrix} \quad (3.35)$$

Setelah itu lingkaran pada bidang xy tersebut harus dirotasi terlebih dahulu terhadap sumbu x , sumbu y , dan sumbu z . Rotasi terhadap sumbu x dilakukan sebesar α , rotasi terhadap sumbu y sebesar β , rotasi terhadap sumbu z sebesar γ . Pencarian sudut-sudut rotasi α, β , dan γ ini dilakukan dengan mencari titik potong setiap vektor garis yang menghubungkan *waypoints* terhadap bidang xy , bidang xz , dan bidang yz . Untuk

mendapatkan titik potong terhadap masing-masing bidang tersebut perlu dicari terlebih dahulu persamaan garis pada ruang 3 dimensi.



Gambar 3.6 Peletakan lingkaran pada bidang 3 dimensi

Berdasarkan Gambar (3.7) persamaan garis dalam bentuk vektor secara umum dapat dilihat pada Persamaan (3.36).

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \end{bmatrix} t \quad (3.36)$$

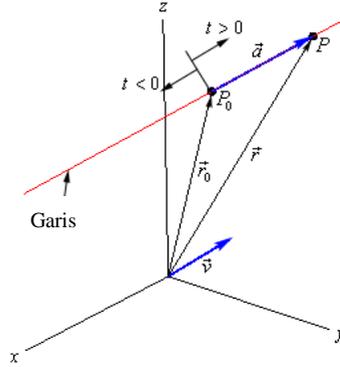
Dari persamaan di atas kemudian bisa didapatkan bentuk persamaan parametris seperti pada Persamaan (3.37) hingga (3.39)

$$x = x_0 + at \quad (3.37)$$

$$y = y_0 + bt \quad (3.38)$$

$$z = z_0 + ct \quad (3.39)$$

Setelah mendapatkan persamaan parametris maka dapat dicari titik potong garis terhadap bidang xz dengan memasukkan nilai $y = 0$ dan didapatkan nilai t tertentu. Nilai t pada saat $y = 0$ ini kemudian digunakan untuk mencari nilai x dan nilai z . Hal yang serupa juga dilakukan untuk mencari perpotongan garis terhadap bidang xy dan bidang yz yaitu dengan memasukkan nilai $z = 0$ dan $x = 0$ secara berurutan. Setelah mendapatkan titik potong masing-masing garis terhadap bidang, maka sudut α, β , dan γ dari tiap bidang dapat dicari berdasarkan 2 garis yang melalui bidang tersebut. Dengan asumsi adanya 3 *waypoint* dan 1 sasaran maka akan terdapat 4 buah garis dan 3 buah bidang secara keseluruhan. Persamaan (3.40) hingga (3.42).



Gambar 3.7 Ilustrasi vektor pada ruang 3 dimensi

$$\alpha_k = \arctan\left(\frac{z_{yz_{k+1}} - z_{yz_k}}{y_{yz_{k+1}} - y_{yz_k}}\right) \quad (3.40)$$

$$\beta_k = \arctan\left(\frac{z_{xz_{k+1}} - z_{xz_k}}{x_{xz_{k+1}} - x_{xz_k}}\right) \quad (3.41)$$

$$\gamma_k = \arctan\left(\frac{x_{xy_{k+1}} - x_{xy_k}}{y_{xy_{k+1}} - y_{xy_k}}\right) \quad (3.42)$$

dimana $k = 1,2,3,4$ adalah indeks untuk bidang ke k , $(y, z)_{yz_k}$ adalah titik potong garis k pada bidang yz , $(x, z)_{xz_k}$ adalah titik potong garis k pada bidang xz , dan $(x, y)_{xy_k}$ adalah titik potong k pada bidang xy . Setelah mendapatkan sudut rotasi tersebut, maka lingkaran yang semula berada pada bidang xy akan dirotasi dengan menggunakan matriks rotasi seperti pada Persamaan (3.43) hingga Persamaan (3.45).

$$\mathbf{R}_{X_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_k & \sin \alpha_k \\ 0 & -\sin \alpha_k & \cos \alpha_k \end{bmatrix} \quad (3.43)$$

$$\mathbf{R}_{Y_k} = \begin{bmatrix} \cos \beta_k & 0 & -\sin \beta_k \\ 0 & 1 & 0 \\ \sin \beta_k & 0 & \cos \beta_k \end{bmatrix} \quad (3.44)$$

$$\mathbf{R}_{Z_k} = \begin{bmatrix} \cos \gamma_k & \sin \gamma_k & 0 \\ -\sin \gamma_k & \cos \gamma_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.45)$$

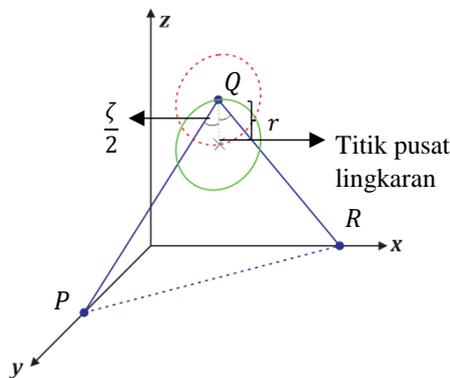
Dengan matriks rotasi R_X , R_Y , R_Z yang telah didapatkan maka dengan menggabungkan Persamaan (3.36) dan matriks rotasi menghasilkan persamaan lingkaran baru (x_b, y_b, z_b) dengan rumus seperti pada Persamaan (3.46).

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = R_{X_k} R_{Y_k} R_{Z_k} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.46)$$

Titik pusat lingkaran ditentukan pertama-tama dengan mencari sebuah titik yang berada pada bidang yang sama dengan lingkaran dan mengapit garis (W_{i-1}, W_i) dan (W_i, W_{i+1}) dengan sudut yang sama. Selain itu titik pusat lingkaran ini akan diletakkan sejauh jari-jari dari lingkaran awal yang semula memiliki titik pusat di *waypoint* ke i . Sudut antara 2 garis yang berpotongan pada titik *waypoint* bisa dicari dengan menggunakan *dot product* seperti pada Persamaan (3.47). Ilustrasi dari pencarian titik pusat lingkaran ini dapat dilihat pada Gambar (3.8).

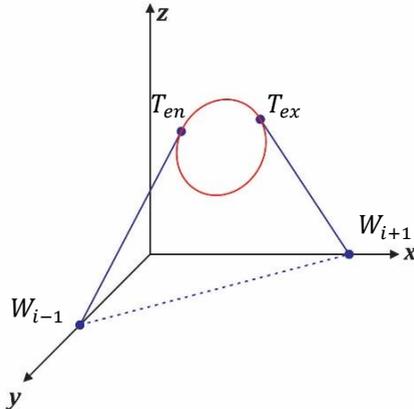
$$\zeta = \arccos\left(\frac{QP \cdot QR}{|QP| |QR|}\right) \quad (3.47)$$

Jumlah lingkaran yang harus diletakkan adalah sesuai dengan jumlah *waypoint* yang telah ditentukan dan bidang yang terbentuk pada masing-masing *waypoint* akan berbeda sehingga posisi dan sudut kemiringan lingkaran juga akan berbeda-beda. Setelah meletakkan lingkaran C_i pada titik pusat yang baru, maka kemudian akan dicari garis singgung yang baru yang berada dalam bidang yang sama dengan lingkaran baru seperti pada Gambar (3.9).

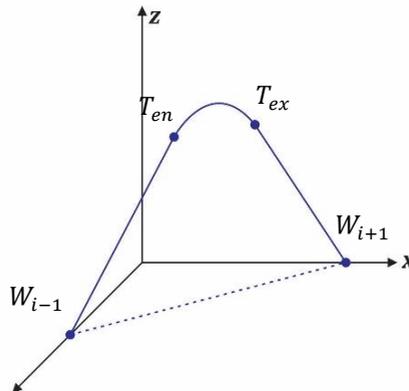


Gambar 3.8 Pencarian titik pusat lingkaran baru

Tangent Entry (T_{en}) merupakan titik singgung antara lingkaran C_i dengan titik W_{i-1} dan *Tangent Exit* (T_{ex}) merupakan titik singgung antara lingkaran C_i dengan titik W_{i+1} . Langkah selanjutnya adalah mendapatkan busur lingkaran C_i terpendek yang menghubungkan T_{en} dengan T_{ex} sehingga didapatkan lintasan seperti pada Gambar (3.10). Busur lingkaran inilah nantinya yang akan menjadi trayektori referensi rudal.



Gambar 3.9 Mencari garis singgung terhadap lingkaran baru



Gambar 3.10 Trayektori busur lingkaran yang menghubungkan T_{en} dan T_{ex}

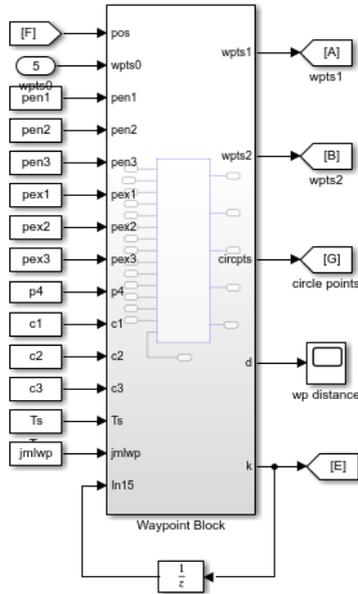
Seluruh algoritma penghitungan trayektori seperti metode yang telah dijelaskan akan dilakukan menggunakan Software Matlab dan

Simulink. Dalam program Matlab, akan dilakukan inisialisasi titik-titik *waypoint* utama yang harus dilalui oleh rudal. Dari titik-titik tersebut kemudian program akan menghitung posisi lingkaran pada tiap *waypoint*. Setelah mendapatkan posisi lingkaran maka program akan menghitung titik-titik *tangent entry* dan *tangent exit* pada tiap lingkaran. Keluaran dari program Matlab ini kemudian disimpan dalam *workspace* untuk kemudian digunakan pada blok fungsi “waypoint” pada Simulink yang dapat dilihat pada Gambar (3.11).

Pada blok fungsi “waypoint” ini, akan memiliki masukan berupa posisi rudal, titik *waypoint* awal “wpts0”, titik *tangent entry* “pen”, titik *tangent exit* “pex”, titik target “p4”, titik pusat lingkaran “c”, waktu *sampling* sistem “Ts”, dan jumlah keseluruhan *waypoints* yang ada “jmlwp”. Tujuan utama dari blok fungsi ini adalah menghasilkan keluaran berupa 2 buah titik *waypoints* yang akan digunakan untuk sistem pemanduan rudal. Kedua titik *waypoints* tersebut secara berurutan adalah titik W_i (“wpts1”) dan W_{i+1} (“wpts2”). Nilai keluaran dari “wpts1” dan “wpts2” akan berubah apabila posisi rudal dianggap telah mencapai titik “wpts2” pada bagian lintasan tertentu. Apabila titik tersebut telah dicapai, maka titik *waypoint* sasaran akan diperbarui. Posisi rudal dianggap telah mencapai titik *waypoint* selanjutnya dapat dihitung menggunakan persamaan jarak antara dua titik pada ruang tiga dimensi seperti pada Persamaan (3.48). Selisih jarak relatif antara posisi rudal dengan *waypoint* sasaran diberi notasi d . Siklus perbaruan titik *waypoint* “wpts1” dan “wpts2” ini akan terus berlangsung sampai rudal mencapai titik sasaran utama pada titik “p4”.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3.48)$$

Selain dua titik *waypoint*, keluaran dari blok fungsi “waypoint” ini juga ada posisi titik pusat lingkaran (x,y,z) yang harus dicapai oleh rudal pada algoritma pemanduan lingkaran atau *loiter*. Posisi titik pusat lingkaran ini juga akan terus diperbarui sampai seluruh lintasan lingkaran telah dilalui oleh rudal. Seluruh pembaruan titik-titik “wpts1”, “wpts2”, dan “circpts” ini mengikuti indeks “k” yang akan naik ketika nilai “d” telah mendekati 0.

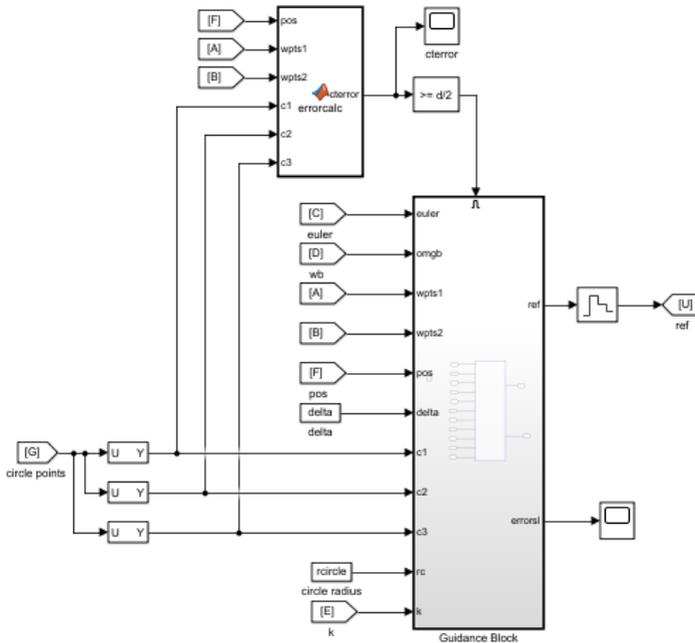


Gambar 3.11 Blok fungsi *waypoint* untuk memberikan titik *waypoint* referensi pada sistem pemanduan

3.4. Perancangan Pemandu Rudal

Setelah trayektori yang diinginkan telah dibuat pada perancangan *Dubins Curve*, maka rudal memerlukan sistem pemanduan untuk bisa mengikuti trayektori tersebut. Pada perancangan sistem pemanduan rudal ini, akan digunakan algoritma *sliding curve guidance*. Pemandu rudal akan menerima informasi berupa posisi rudal saat ini dan *waypoint* sasaran. Informasi tersebut kemudian akan diproses menjadi perhitungan sudut arah hadap rudal yang harus dicapai agar rudal bisa bergerak mengikuti trayektori yang diinginkan.

Secara keseluruhan, terdapat dua macam komponen pemandu yaitu masing-masing pemandu untuk gerakan *yaw* (ψ) dan *pitch* (θ). Pemandu untuk gerakan *yaw* menerima informasi berupa sudut *yaw* dan posisi rudal saat ini dalam koordinat x dan y sistem koordinat bumi. Sedangkan untuk pemanduan *pitch* menggunakan informasi posisi x dan z . Blok Simulink untuk sistem pemanduan dapat dilihat pada Gambar (3.12).



Gambar 3.12 Blok fungsi sistem pemanduan rudal

Informasi sudut Euler, kecepatan sudut badan rudal, dan posisi rudal terhadap koordinat bumi diterima dari komponen *gyro* yang berada di dalam rudal. Sedangkan input *waypoint* saat ke i “wpts1” dan $i + 1$ “wpts2” diterima dari blok fungsi *waypoint* pada bagian sebelumnya. Selain itu, sistem pemanduan ini juga menerima informasi letak titik pusat lingkaran beserta jari-jarinya untuk kebutuhan algoritma pemanduan lingkaran atau *loiter*.

Dapat dilihat pada Gambar (3.13), strategi pemanduan rudal secara keseluruhan dibagi menjadi 2 macam, yaitu algoritma untuk garis lurus dan algoritma untuk lintasan lingkaran atau *loiter*. Pada saat awal rudal diluncurkan dan telah melewati fase *boost*, maka trayektori referensi yang akan dilalui rudal merupakan garis lurus. Setelah titik akhir dari lintasan garis lurus tersebut tercapai, maka rudal akan berbelok dengan titik sasaran *virtual* yang bergerak pada busur lingkaran. Titik akhir dari lintasan garis lurus juga merupakan titik *tangent entry* pada busur

lingkaran. Pada saat berbelok sepanjang busur lingkaran, sistem pemanduan akan menggunakan algoritma pemanduan lingkaran. Algoritma lingkaran ini digunakan pada saat rudal memasuki *tangent entry* dan akan selesai pada saat rudal telah mencapai titik *tangent exit* pada busur lingkaran. Tahap pergantian algoritma ini terus berlanjut sampai semua *waypoints* termasuk sasaran utama rudal telah tercapai.

Urutan algoritma sistem pemanduan garis lurus untuk *yaw* dapat dilihat pada Tabel (3.2) dan sistem pemanduan garis lurus untuk *pitch* dapat dilihat pada Tabel (3.3). Sedangkan algoritma sistem pemanduan lingkaran untuk *yaw* dapat dilihat pada Tabel (3.4) dan sistem pemanduan lingkaran untuk *pitch* dapat dilihat pada Tabel (3.5).

Tabel 3.2 Algoritma Sistem pemanduan garis lurus *yaw*

Algoritma		Keterangan
1	$W_i = (x_i, y_i)$	Waypoint ke i
2	$W_{i+1} = (x_{i+1}, y_{i+1})$	Waypoint ke $i + 1$
3	$p = (x_i, y_i), \psi$	Posisi dan orientasi rudal pada bidang xy
4	$R_u = W_i - p $	Jarak W_i terhadap p
5	$\theta_y = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i)$	θ pada bidang xy
6	$\theta_{uy} = \text{atan2}(y - y_i, x - x_i)$	θ_u pada bidang xy
7	$\beta_y = \theta_y - \theta_{uy}$	β pada bidang xy
8	$R_y = \sqrt{R_u^2 - (R_u \sin(\beta_y))^2}$	R pada bidang xy
9	$x'_{ty} \leftarrow (R_y + \delta) \cos \theta_y$ $y'_{ty} \leftarrow (R_y + \delta) \sin \theta_y$ $s_y = (x'_{ty}, y'_{ty})$	Posisi sasaran <i>virtual</i> (s) pada lintasan
10	$\psi_d = \text{atan2}(y'_{ty} - y, x'_{ty} - x)$	ψ referensi (ψ_{ref})

Tabel 3.3 Algoritma sistem pemanduan garis lurus *pitch*

Algoritma		Keterangan
1	$W_i = (x_i, z_i)$	Waypoint ke i
2	$W_{i+1} = (x_{i+1}, z_{i+1})$	Waypoint ke $i + 1$
3	$p = (x_i, z_i), \theta$	Posisi dan orientasi rudal pada bidang xz
4	$R_u = W_i - p $	Jarak W_i terhadap p

5	$\theta_p = \text{atan2}(z_{i+1} - z_i, x_{i+1} - x_i)$	θ pada bidang xz
6	$\theta_{up} = \text{atan2}(z - z_i, x - x_i)$	θ_u pada bidang xz
7	$\beta_p = \theta_p - \theta_{up}$	β pada bidang xz
8	$R_p = \sqrt{R_u^2 - (R_u \sin(\beta_p))^2}$	R pada bidang xz
9	$x'_{tp} \leftarrow (R_p + \delta) \cos \theta_p$ $z'_{tp} \leftarrow (R_p + \delta) \sin \theta_p$ $s_p = (x'_{tp}, z'_{tp})$	Posisi sasaran <i>virtual</i> (s) pada lintasan
10	$\theta_d = \text{atan2}(z'_{tp} - z, \sqrt{(x'_{tp} - x)^2 + (y'_{ty} - y)^2})$	θ referensi (θ_{ref})

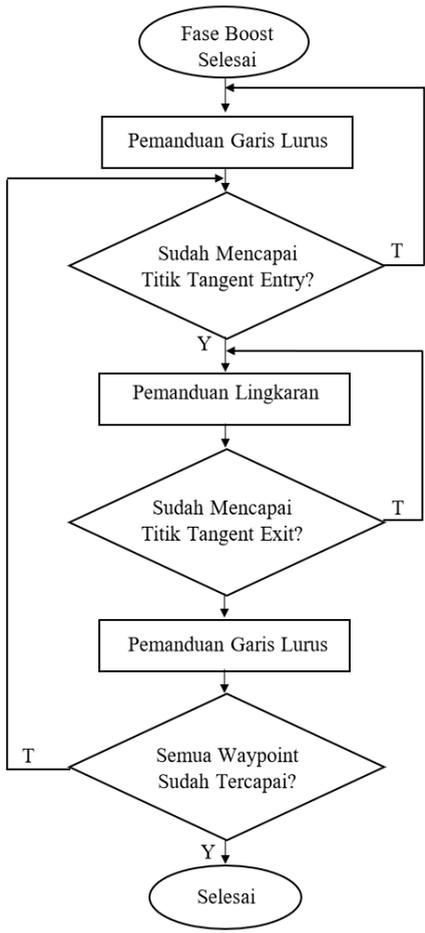
Tabel 3.4 Algoritma sistem pemanduan lingkaran *yaw*

Algoritma		Keterangan
1	$O = (x_i, y_i), r$	Titik pusat lingkaran
2	$p = (x_i, y_i), \psi$	Posisi dan orientasi rudal pada bidang xy
3	$d = O - p - r$	Jarak orthogonal rudal terhadap lingkaran
4	$\theta_y = \text{atan2}(y - y_i, x - x_i)$	θ pada bidang xy
5	$x'_{ty} \leftarrow (r \cos(\theta_y + \lambda))$ $y'_{ty} \leftarrow (r \sin(\theta_y + \lambda))$ $s_y = (x'_{ty}, y'_{ty})$	Posisi sasaran <i>virtual</i> (s) pada lintasan lingkaran
6	$\psi_d = \text{atan2}(y'_{ty} - y, x'_{ty} - x)$	ψ_d referensi (ψ_{ref})

Tabel 3.5 Algoritma sistem pemanduan lingkaran *pitch*

Algoritma		Keterangan
1	$O = (x_i, z_i), r$	Titik pusat lingkaran
2	$p = (x_i, z_i), \theta$	Posisi dan orientasi rudal pada bidang xz
3	$d = O - p - r$	Jarak orthogonal rudal terhadap lingkaran
4	$\theta_p = \text{atan2}(z - z_i, x - x_i)$	θ pada bidang xz

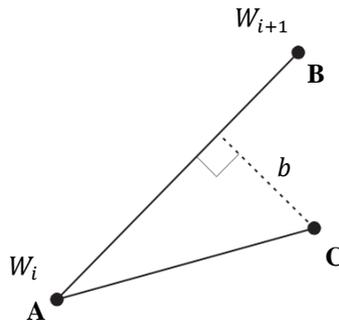
5	$x'_{tp} \leftarrow (r \cos(\theta_p + \lambda))$ $z'_{tp} \leftarrow (r \sin(\theta_p + \lambda))$ $s_p = (x'_{tp}, z'_{tp})$	Posisi sasaran <i>virtual</i> (s) pada lintasan lingkaran
6	$\theta_d = \text{atan2}(z'_{tp} - z, x'_{tp} - x)$	θ_d referensi (θ_{ref})



Gambar 3.13 Diagram alur strategi sistem pemanduan rudal

Pada sistem pemanduan lintasan lingkaran, titik sasaran *virtual* akan bergerak sepanjang busur lingkaran yang telah ditransformasikan menggunakan matriks rotasi masing-masing pada bidang xy dan xz pada perencanaan trayektori *Dubins curve*.

Penentuan *sliding surface* pada sistem pemanduan *sliding curve* didapatkan dari jarak absolut rudal terhadap trayektori referensi. Jarak absolut ini adalah jarak orthogonal rudal (dihitung dari titik pusat badan rudal) terhadap trayektori referensi. Sehingga dalam kata lain jarak absolut terhadap trayektori referensi inilah yang menentukan lebar *sliding surface* sistem pemanduan *sliding curve*. Pada sistem pemanduan ini lebar *sliding surface* yang ditentukan adalah sebesar dua kali diameter rudal yaitu sebesar 0.56 meter sehingga jarak orthogonal maksimal yang diambil adalah sebesar jari-jari rudal yaitu 0.14 meter. Jadi sistem pemanduan secara keseluruhan akan bekerja ketika rudal berada di luar *sliding surface* atau jarak orthogonal relatif rudal terhadap trayektori telah melebihi 0.14 meter dan sistem pemanduan akan dinonaktifkan selama rudal berada di dalam *sliding surface*. Aktivasi dan non-aktivasi sistem pemanduan rudal secara keseluruhan dikendalikan oleh blok fungsi 'errorcalc'. Berdasarkan Gambar (3.14) penghitungan jarak orthogonal rudal terhadap trayektori dapat dicari menggunakan *cross product*.



Gambar 3.14 Menghitung jarak orthogonal titik terhadap vektor

Diasumsikan terdapat titik **A** pada posisi W_i , titik **B** pada posisi W_{i+1} , dan titik **C** sebagai posisi rudal maka untuk mencari jarak orthogonal (b) titik **C** terhadap vektor garis **AB** bisa menggunakan Persamaan (3.46).

$$b = \frac{|AC \times AB|}{|AB|} \quad (3.49)$$

Selain algoritma pemanduan seperti yang telah dipaparkan pada Tabel (3.3) sampai dengan Tabel (3.6), sudut referensi *pitch* dan *yaw* yang dihasilkan dari sistem pemanduan perlu ditambahkan dengan nilai sudut tertentu untuk mengatasi *error steady state* yang akan timbul ketika nilai β telah mendekati nol dan menyebabkan pemandu rudal menjadi tidak aktif meskipun telah keluar dari selubung *sliding surface*. Nilai penambahan sudut referensi untuk *pitch* dapat dilihat pada Persamaan (3.50) dan untuk *yaw* pada Persamaan (3.51).

$$\theta'_{ref} = \frac{e_z}{\delta} \quad (3.50)$$

$$\psi'_{ref} = \frac{e_y}{\delta} \quad (3.51)$$

dimana:

$$e_z = R_u \sin(\beta_p) \quad (3.52)$$

$$e_y = R_u \sin(\beta_y) \quad (3.53)$$

Dari Persamaan (3.50) dan Persamaan (3.51) dapat dilihat bahwa kalkulasi sudut referensi tambahan ini adalah berdasarkan besarnya sudut yang dibentuk antara *cross-track error* dan *vertical-track error* yang ada pada rudal dengan jarak sasaran *virtual* (δ) yang didapatkan pada sistem pemanduan garis lurus *yaw* dan *pitch* secara berurutan. Oleh karena itu, besarnya θ'_{ref} dan ψ'_{ref} bergantung pada nilai *cross-track* dan *vertical-track error* relatif rudal terhadap trayektori referensi pada tiap saat.

3.5. Perancangan *Autopilot Rudal*

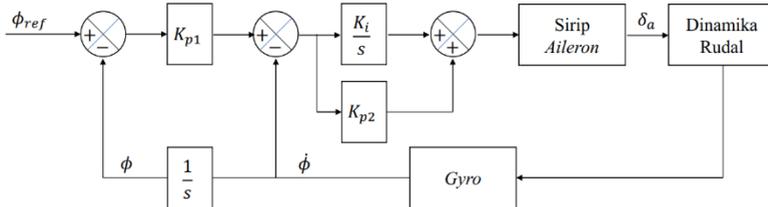
Sistem *autopilot* pada rudal ini akan menerima masukan berupa sudut *yaw* dan *pitch* referensi dari sistem pemanduan rudal. Sudut referensi tersebut kemudian akan menjadi masukan dari kontroler *cascade* P-PI pada rudal. Tujuan utama dari *autopilot* ini adalah memberikan keluaran berupa sinyal kontrol kepada sirip-sirip rudal sebagai aktuator agar rudal bisa memiliki sudut *roll*, *pitch*, dan *yaw* yang sama dengan sudut referensi.

Pada perancangan *autopilot* rudal ini, masing-masing sudut *roll*, *pitch*, dan *yaw* memiliki *autopilot* yang terpisah. *Autopilot pitch* mengendalikan gerakan pada bidang *xz* dan sistem *autopilot yaw* mengendalikan gerakan pada bidang *xy* sehingga kedua *autopilot* ini juga

bisa disebut sebagai *autopilot lateral*. Sedangkan *autopilot roll* disebut sebagai *autopilot longitudinal*. Secara keseluruhan, masing-masing sistem *autopilot* ini terdiri dari pengaturan *inner loop* dan *outer loop*. *Inner loop* merupakan *loop* pengaturan kecepatan sudut untuk menghasilkan redaman kecepatan sudut rudal, sedangkan *outer loop* merupakan *loop* utama pengaturan sudut orientasi rudal.

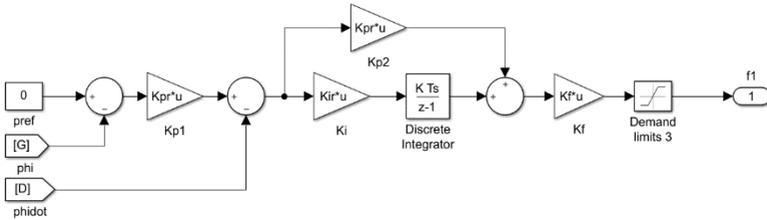
3.5.1. Autopilot Roll

Fungsi dari *autopilot roll* adalah menstabilkan kecepatan *roll* sehingga bisa didapatkan kestabilan sudut *roll* pada sumbu longitudinal rudal. Hal ini dilakukan dengan menerima informasi pengukuran kecepatan *roll* dari *gyro rate* dan menghasilkan sinyal kontrol kepada sirip-sirip rudal untuk bisa mencapai *setpoint* sudut *roll* yang diinginkan. Dalam *autopilot roll* ini, sirip rudal yang digunakan adalah sirip *aileron*. Blok diagram *autopilot roll* dapat dilihat pada Gambar (3.15).



Gambar 3.15 Blok diagram *autopilot roll*

Berdasarkan diagram blok pada Gambar (3.15), maka kemudian akan diimplementasikan pada blok fungsi Simulink seperti pada Gambar (3.16). Masukan dari *autopilot roll* ini adalah sudut *roll* referensi yang kemudian dikurangi dengan *feedback* sudut *roll* rudal dari *gyro* dan didapatkan selisih atau *error* sudut *roll* yang kemudian dimasukkan pada kontroler proporsional pada *outer loop*. Keluaran dari kontroler proporsional ini kemudian dikurangi dengan kecepatan *roll* rudal dari *gyro* sehingga didapatkan selisih atau *error* kecepatan sudut *roll* yang dimasukkan pada kontroler PI pada *inner loop*. Keluaran dari kontroler PI inilah yang kemudian menjadi sinyal kontrol kepada sirip *aileron* dengan diberi batasan defleksi sirip maksimum sesuai spesifikasi mekanik sirip rudal. Pada *autopilot roll*, kecepatan *roll* referensi akan selalu ditetapkan nilai 0 dengan tujuan agar tidak timbul gerak *roll* pada rudal. Parameter pada *autopilot roll* dapat dilihat pada Tabel (3.6)..



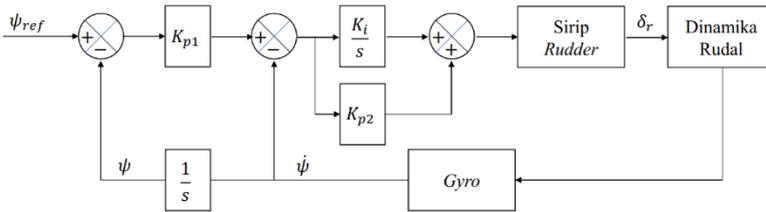
Gambar 3.16 Blok Simulink *autopilot roll*

Tabel 3.6 Parameter kontroler *cascade P-PI* pada *autopilot roll*

Parameter	Keterangan
K_{p1}	Gain proporsional <i>outer loop</i>
K_{p2}	Gain proporsional <i>inner loop</i>
K_i	Gain integral <i>inner loop</i>
K_f	Gain proporsional

3.5.2. *Autopilot Yaw*

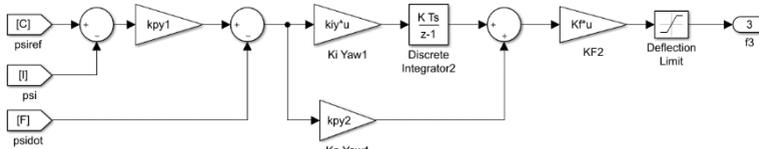
Fungsi dari *autopilot yaw* adalah mengendalikan sirip *rudder* agar rudal dapat bergerak sesuai dengan sudut *yaw* seperti yang diinginkan. Masukan dari *autopilot yaw* adalah sudut *yaw* referensi yang berasal dari sistem pemanduan dan kemudian dikurangi dengan sudut *yaw* rudal untuk mendapatkan selisih atau *error* sudut *yaw*. Sinyal *error yaw* tersebut kemudian dimasukkan pada kontroler proporsional *outer loop*. Keluaran dari kontroler proporsional ini kemudian dikurangi dengan kecepatan sudut *yaw* dari *gyro* dan dimasukkan pada kontroler PI *inner loop*. Kontroler PI kemudian menghasilkan sinyal kontrol yang akan diterima oleh sirip *rudder* pada rudal. Sama seperti *autopilot roll*, sinyal kontrol akan diberi batasan sudut defleksi sirip tertentu sesuai dengan spesifikasi mekanik sirip rudal. Pada rudal yang digunakan pada simulasi ini, sudut defleksi sirip *rudder* memiliki rentang dari -10° sampai dengan 10° . Pembatasan defleksi sirip rudal harus dilakukan untuk mencegah kerusakan struktural atau *angle of attack* yang terlalu besar sehingga menaikkan koefisien *drag* pada rudal dan menghilangkan kemampuan akselerasi. Blok diagram dan Simulink dari *autopilot yaw* dapat dilihat pada Gambar (3.17) dan (3.18). Parameter untuk *autopilot yaw* dapat dilihat pada Tabel (3.7).



Gambar 3.17 Blok diagram *autopilot yaw*

Tabel 3.7 Parameter kontroler *autopilot yaw*

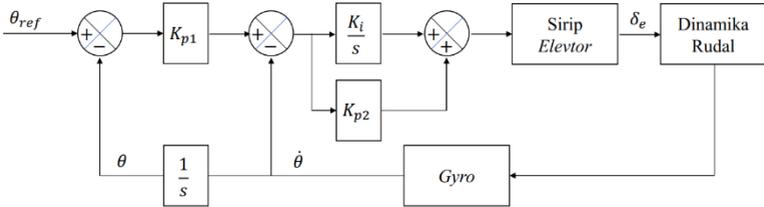
Parameter	Keterangan
K_{p1}	Gain proporsional <i>outer loop</i>
K_{p2}	Gain proporsional <i>inner loop</i>
K_i	Gain integral <i>inner loop</i>
K_f	Gain proporsional



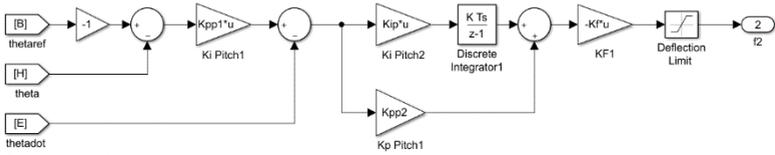
Gambar 3.18 Blok Simulink *autopilot yaw*

3.5.3. *Autopilot Pitch*

Sistem *autopilot pitch* memiliki prinsip kerja yang sama dengan *autopilot yaw*. Perbedaan pada *Autopilot pitch* adalah yang dikendalikan merupakan sirip *elevator* agar rudal dapat bergerak sesuai dengan sudut *pitch* yang diinginkan. Masukan dari *autopilot pitch* adalah sudut *pitch* referensi dari sistem pemanduan. Pada sistem *autopilot pitch* ini juga digunakan kontroler PI dan pada keluarannya diberi batasan sudut defleksi sirip *elevator* dengan rentang dari -10° sampai dengan 10° . Blok diagram sistem *autopilot pitch* beserta implementasi rancangan pada Simulink dapat dilihat pada Gambar (3.19) dan (3.20). Parameter pada sistem *autopilot pitch* bisa dilihat pada Tabel (3.8).



Gambar 3.19 Blok diagram *autopilot pitch*



Gambar 3.20 Blok Simulink *autopilot pitch*

Perlu diperhatikan bahwa sudut *pitch* referensi dari sistem pemanduan merupakan hasil perhitungan dengan asumsi sumbu z positif adalah ke arah atas menjauhi bumi, sehingga untuk bisa diproses pada sistem z positif yang mengarah ke pusat bumi (sistem koordinat NED) maka perlu dibalikkan arahnya dengan mengalikan dengan -1 . Demikian juga keluaran dari kontroler PI perlu dikalikan kembali dengan -1 agar sinyal kontrol yang diberikan pada sirip *elevator* sesuai dengan sistem koordinat NED.

Tabel 3.8 Parameter kontroler *autopilot pitch*

Parameter	Keterangan
K_{p1}	Gain proporsional <i>inner loop</i>
K_{p2}	Gain proporsional <i>outer loop</i>
K_i	Gain integral <i>inner loop</i>
K_f	Gain proporsional

3.5.4. Mekanisasi Sirip Rudal

Rudal yang memiliki 4 sirip dan tersusun secara *plus* memiliki beban kerja yang terbagi menjadi 2, yaitu dua sirip horizontal untuk mengendalikan *pitch* dan dua sirip vertikal mengendalikan *yaw*. Sedangkan rudal yang digunakan pada tugas akhir ini adalah rudal empat

sirip dengan susunan *cross* sehingga beban kerja sirip terbagi menjadi 4 dan perlu dicari mekanisasi masing-masing sirip rudal.

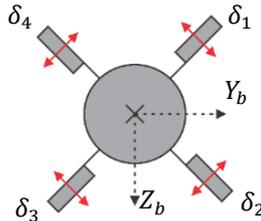
Keluaran dari sistem *autopilot roll, pitch, dan yaw* merupakan sinyal kontrol terhadap sirip *aileron* (δ_a), *elevator* (δ_e), dan *rudder* (δ_r) secara berurutan sedangkan sinyal kontrol yang harus diterima pada masing-masing sirip adalah besarnya sudut defleksi yang harus dilakukan oleh aktuator *servo* untuk menggerakkan sirip. Berdasarkan urutan sirip dari 1 hingga 4 seperti pada Gambar (3.21), konversi dari δ_a, δ_e , dan δ_r menjadi $\delta_1, \delta_2, \delta_3$, dan δ_4 bisa dilihat pada Persamaan (3.54) – (3.57).

$$\delta_1 = \delta_a + \delta_e - \delta_r \quad (3.54)$$

$$\delta_2 = \delta_a + \delta_e + \delta_r \quad (3.55)$$

$$\delta_3 = \delta_a - \delta_e + \delta_r \quad (3.56)$$

$$\delta_4 = \delta_a - \delta_e - \delta_r \quad (3.57)$$



Gambar 3.21 Sirip rudal dengan susunan *cross*

Setelah mendapatkan persamaan sudut defleksi sirip pada masing-masing aktuator sirip maka kemudian dilakukan konversi kembali ke sudut-sudut *aileron, elevator, dan rudder* yang akan menjadi masukan pada dinamika gerak rudal dengan menggunakan Persamaan (3.58) – (3.60).

$$\delta_a = \frac{\delta_1 + \delta_2 + \delta_3 + \delta_4}{4} \quad (3.58)$$

$$\delta_e = \frac{\delta_1 + \delta_2 - \delta_3 - \delta_4}{4} \quad (3.59)$$

$$\delta_r = \frac{-\delta_1 + \delta_2 + \delta_3 - \delta_4}{4} \quad (3.60)$$

Mengingat bahwa motor *servo* yang digunakan pada rudal sebagai aktuator sirip rudal adalah memiliki respon sistem orde dua maka harus

ditentukan juga kriteria desainnya. Diharapkan respon sirip yang didapatkan sangatlah cepat sehingga ditentukan kriteria desain dimana respon sistem memiliki *setting time* ($\pm 2\%$) sebesar 0.037 detik dan *overshoot* maksimum sebesar 4.3%. Setelah menentukan kriteria desain maka kemudian akan dicari nilai ζ dan ω_n dengan menggunakan Persamaan (3.61) dan Persamaan (3.62).

$$\zeta = \frac{\left| \ln \left(\frac{PO}{100} \right) \right|}{\sqrt{\pi^2 + \left(\frac{PO}{100} \right)^2}} \quad (3.61)$$

$$\omega_n = \frac{4}{\zeta t_s} \quad (3.62)$$

dimana:

ζ = Rasio redaman

ω_n = Frekuensi natural

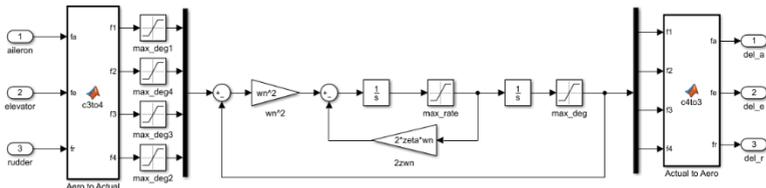
PO = Maksimum *overshoot*

t_s = *Settling time*

Dengan menggunakan Persamaan (3.61) dan Persamaan (3.62) didapatkan nilai $\omega_n = 150$ dan $\zeta = 0.707$. Setelah mendapatkan nilai-nilai parameter persamaan respon orde dua maka dapat dituliskan dalam persamaan seperti dalam Persamaan (3.63).

$$\frac{\delta_{actual}(s)}{\delta_{cmd}(s)} = \frac{22500}{s^2 + 1060.5s + 22500} \quad (3.63)$$

Dinamika aktuator sirip rudal ini kemudian dimasukkan ke dalam blok Simulink yang dapat dilihat pada Gambar (3.22).



Gambar 3.22 Blok Simulink dari Dinamika Aktuator Sirip Rudal

3.6. Perancangan Pengujian Sistem

Setelah seluruh perancangan komponen sistem telah dilakukan, maka akan dilanjutkan dengan tahap pengujian sistem. Pengujian yang pertama adalah pada model dinamika rudal. Setelah itu akan dilakukan pembuatan trayektori menggunakan sistem yang telah dirancang dan dilanjutkan dengan pengujian sistem pemanduan rudal. Pengujian yang ketiga adalah pengujian sistem *autopilot* rudal. Setelah ketiga pengujian sistem tersebut, maka pengujian sistem secara keseluruhan akan dilakukan yang melibatkan seluruh sistem yang telah dirancang.

3.6.1. Perancangan Pengujian Model Matematik Rudal

Pada pengujian ini akan dilakukan pengujian terhadap rudal ketika diberikan beberapa konfigurasi awal tertentu. Pengujian ini memiliki tujuan untuk mencari pada konfigurasi awal berapakah rudal tidak akan jatuh setelah fase *booster* selesai, karena selama fase *booster* sistem pemanduan dan autopilot rudal tidak diaktifkan.

3.6.2. Perancangan Pengujian Trayektori *Dubins Curve*

Pada pengujian ini akan dilakukan pembuatan trayektori berdasarkan sistem yang telah dirancang. Diharapkan dengan sistem yang telah dirancang, trayektori yang dibangkitkan berdasarkan algoritma *Dubins curve* dapat menghasilkan trayektori pada ruang 3 dimensi yang nantinya akan menjadi trayektori referensi bagi sistem pemanduan rudal.

3.6.3. Perancangan Pengujian Sistem *Autopilot* Rudal

Dalam pengujian ini masing-masing sistem *autopilot roll*, *pitch*, dan *yaw* akan diuji apakah ketika diberi masukan referensi berupa suatu nilai konstan, *autopilot* dapat memberi sinyal kendali yang mengikuti nilai referensi tersebut dengan baik. Selain itu juga akan dilakukan pencarian parameter *gain* yang menghasilkan sinyal keluaran terbaik dari kontroler *cascade* P-PI sistem *autopilot* dengan menggunakan metode *tuning* prosedural.

3.6.4. Perancangan Pengujian Sistem Pemanduan Rudal

Pengujian sistem pemanduan rudal ini dilakukan untuk melihat apakah sistem pemanduan yang telah dirancang mampu memberikan sudut referensi ϕ_{ref} , θ_{ref} dan ψ_{ref} terhadap sasaran *virtual* yang bergerak sepanjang trayektori sehingga rudal dapat bergerak mengikuti

trayektori dengan baik. Jarak relatif sasaran *virtual* (δ dan λ) terhadap rudal akan diubah-ubah kemudian dilakukan simulasi dan juga analisa terhadap dampak perubahan yang terjadi terhadap pergerakan rudal. Nilai δ dan λ yang terbaik nantinya akan ditentukan berdasarkan *cross-track* dan *vertical-track error* yang terkecil. Trayektori referensi yang akan digunakan untuk pengujian sistem pemanduan ini masih berupa trayektori garis lurus sederhana dan trayektori lingkaran saja.

3.6.5. Perancangan Pengujian Sistem Keseluruhan

Pada pengujian ini, akan dilakukan pengujian sistem keseluruhan yang melibatkan sistem pemanduan, sistem *autopilot*, serta dinamika rudal itu sendiri. Pada pengujian ini rudal akan disimulasikan diluncurkan dengan konfigurasi awal tertentu yang telah didapatkan dari pengujian model matematik rudal dan rudal diharapkan mampu bergerak mengikuti trayektori yang telah dibuat sampai mengenai sasaran pada titik *waypoint* yang paling akhir.

[Halaman ini sengaja dikosongkan]

BAB 4 HASIL DAN ANALISA

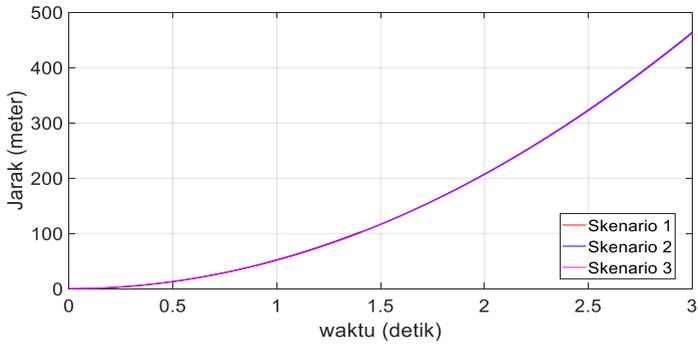
4.1. Pengujian Model Matematik Rudal

Pengujian terhadap model matematik rudal yang telah dirancang perlu dilakukan untuk mendapatkan konfigurasi awal rudal yang dibutuhkan agar rudal tidak jatuh ke permukaan selama fase *booster* berlangsung, yaitu selama 3 detik dan setelah fase *booster* selesai diharapkan rudal memiliki arah hadap ke titik *waypoint* pertama. Konfigurasi awal rudal yang dimaksud adalah posisi awal rudal (x_e, y_e, z_e) , kecepatan translasi rudal (u, v, w) , sudut orientasi rudal (ϕ, θ, ψ) , dan kecepatan sudut rotasi (p, q, r) . Untuk mendapatkan konfigurasi awal yang tepat, akan dilakukan beberapa skenario untuk dibandingkan. Pengujian ini dilakukan dengan memberi masukan gaya dorong *thrust* dan sudut defleksi sirip δ_a, δ_e , dan δ_r diberi nilai nol. Adapun skenario yang akan diuji dapat dilihat pada Tabel (4.1).

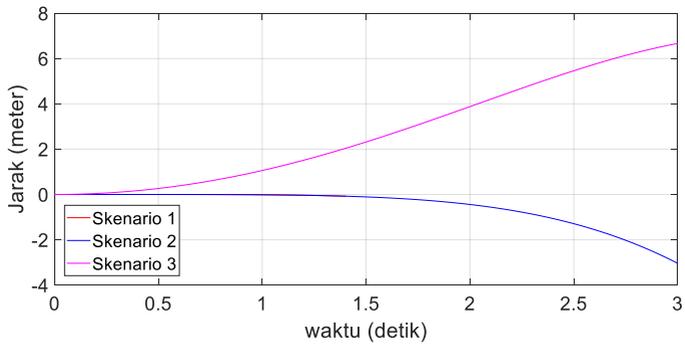
Tabel 4.1 Konfigurasi Awal Rudal

Kondisi awal	Skenario 1	Skenario 2	Skenario 3
x_e, y_e, z_e	[0 0 -5]	[0 0 -5]	[0 0 -5]
u, v, w	[0.0001 0 0]	[0.0001 0 0]	[0.0001 0 0]
ϕ, θ, ψ	[0 0.1 0]	[0 0.2 0]	[0 0.245 0.0251]
p, q, r	[0 0 0]	[0 0 0]	[0 0 0]

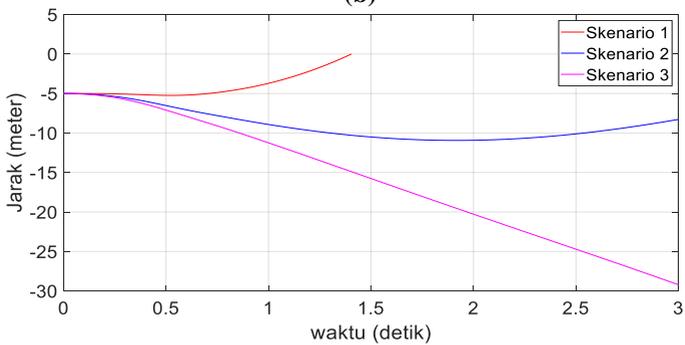
Setelah menentukan konfigurasi awal rudal dalam tiga skenario yang berbeda, kemudian dilakukan simulasi pengujian model matematik rudal menggunakan Simulink seperti yang dapat dilihat pada Gambar (4.1). Dari ketiga skenario yang dilakukan, nilai kecepatan translasi dan kecepatan sudut rotasi tidak dilakukan perubahan. Sedangkan sudut orientasi awal rudal yang diubah adalah hanya sudut *pitch* (θ) dan *yaw* (ψ). Nilai z awal adalah -5 meter karena diasumsikan pada kondisi awal rudal akan dilepaskan ke udara terlebih dahulu menggunakan pelontar dari permukaan bumi baru kemudian *booster* aktif dan rudal mulai bergerak dari ketinggian tersebut.



(a)



(b)



(c)

Gambar 4.1 Posisi rudal (a) x_e , (b) y_e dan (c) z_e

Berdasarkan gambar di atas dapat dilihat bahwa pada skenario 1 rudal akan jatuh sebelum fase *booster* selesai dengan melihat nilai $z = 0$ setelah rudal diluncurkan selama 1.04 detik. Sedangkan pada skenario 2 dan 3 rudal belum jatuh setelah fase *booster* selesai. Dengan asumsi bahwa nantinya titik *waypoint* pertama yang akan digunakan terletak pada $[10000 \ 100 \ -300]$ maka diharapkan rudal bisa memiliki sudut orientasi yang mendekati sudut-sudut kemiringan lintasan garis referensi dari titik awal ke titik *waypoint* pertama yaitu $[0 \ 0.03 \ 0.01]$ pada saat fase *booster* telah berakhir. Dengan demikian, pada simulasi rudal selanjutnya kondisi awal yang akan digunakan adalah yang didapatkan dari skenario 3, karena pada skenario ini rudal memiliki sudut orientasi $[0 \ 0.0306 \ 0.01]$ pada posisi $[462 \ 8.55 \ -29.13]$ pada saat $t = 3$ detik.

4.2. Pengujian Trayektori *Dubins Curve*

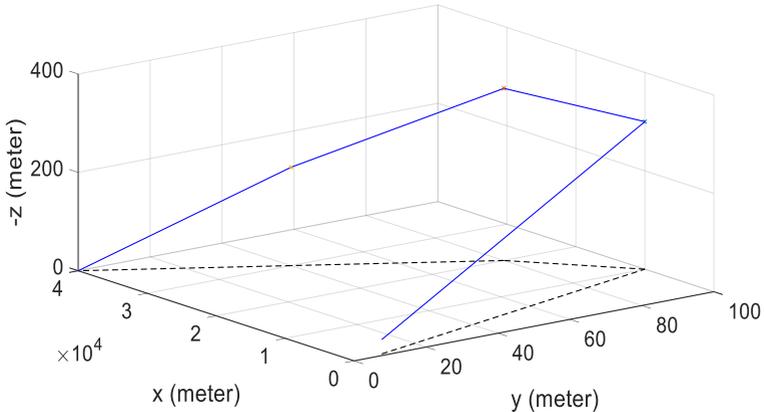
Tujuan dari pengujian ini adalah menghasilkan trayektori berdasarkan algoritma *Dubins curve* yang telah dibahas pada Bab sebelumnya. Pengujian dilakukan pertama-tama dengan membuat trayektori garis lurus dengan 4 buah *waypoint* termasuk sasaran. Setelah membuat trayektori garis lurus maka akan dilanjutkan dengan pembuatan trayektori *Dubins curve*. Trayektori garis lurus dibuat dengan cara menghubungkan titik-titik *waypoint* dari paling awal sampai titik paling akhir yaitu sasaran. Titik-titik *waypoints* yang akan digunakan untuk membangun trayektori dapat dilihat pada Tabel 4.2.

Tabel 4.2 Titik *Waypoints* pengujian trayektori garis lurus

<i>Waypoint</i>	Posisi
0	$[462 \ 8.55 \ -29.13]$
1	$[10000 \ 100 \ -300]$
2	$[20000 \ 80 \ -350]$
3	$[30000 \ 40 \ -200]$
4	$[40000 \ 0 \ 0]$

Titik *waypoint* yang pertama atau *waypoint* 0 yang digunakan bukan titik $(0,0,0)$ melainkan menggunakan posisi terakhir rudal setelah fase *booster* selesai dan sistem pemanduan serta *autoiplot* baru diaktifkan. Hal ini dilakukan dengan tujuan bahwa setelah fase *booster* selesai, trayektori referensi yang harus dilalui rudal tidak berjarak terlalu jauh dari posisi

rudal saat itu sehingga dapat menyebabkan pergerakan rudal yang lebih tidak stabil dan gerakan yang tidak *smooth* akibat momentum rudal yang timbul. Hasil trayektori garis lurus yang dibuat berdasarkan titik *waypoints* pada Tabel (4.2) dapat dilihat pada Gambar (4.2).

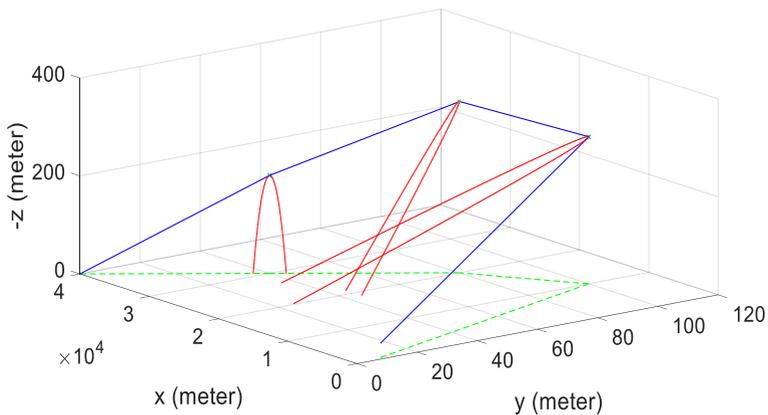


Gambar 4.2 Trayektori Garis Lurus dengan 4 buah *waypoints*

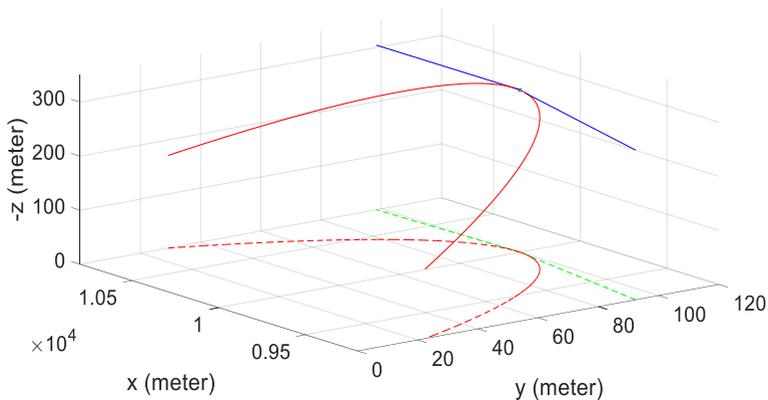
Setelah mendapatkan trayektori garis lurus dengan 4 *waypoints*, akan dilakukan pembuatan trayektori dengan menggunakan metode *Dubins curve* dengan titik-titik *waypoints* yang sama dengan Tabel (4.2). Hasil perancangan trayektori dengan *Dubins curve* dapat dilihat pada Gambar (4.3). Dapat dilihat bahwa secara sepintas lingkaran tampak seperti garis dan oval tetapi hanyalah karena sudut pandang yang berbeda pada ruang 3 dimensi beserta skala sumbu x dan sumbu y yang tidak sama. Apabila dilihat secara lebih dekat, titik belok pada *waypoints* yang telah mengimplementasikan *Dubins curve* dapat dilihat pada Gambar (4.4) hingga Gambar (4.6).

Dapat dilihat dari Gambar (4.3) bahwa trayektori dubins memiliki trayektori yang mirip dengan pada trayektori garis lurus pada Gambar (4.2) tetapi sebenarnya keduanya berbeda pada titik belok pada *waypoint*. Pada trayektori garis lurus terjadi bentuk trayektori yang patah dan tidak mulus yang nantinya dapat menyebabkan pergerakan rudal yang tidak mulus juga dan *error* jarak relatif rudal terhadap trayektori juga bertambah. Dengan pengujian ini, dapat dilihat bahwa algoritma

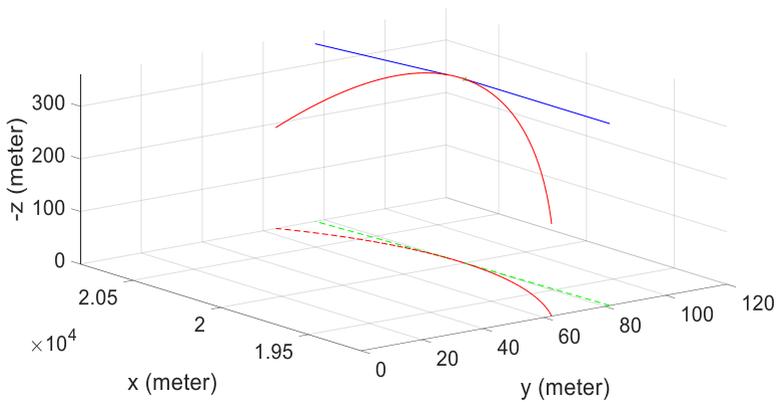
perencanaan trayektori menggunakan *Dubins curve* bisa dilakukan dan bisa digunakan sebagai trayektori referensi bagi sistem pemanduan rudal.



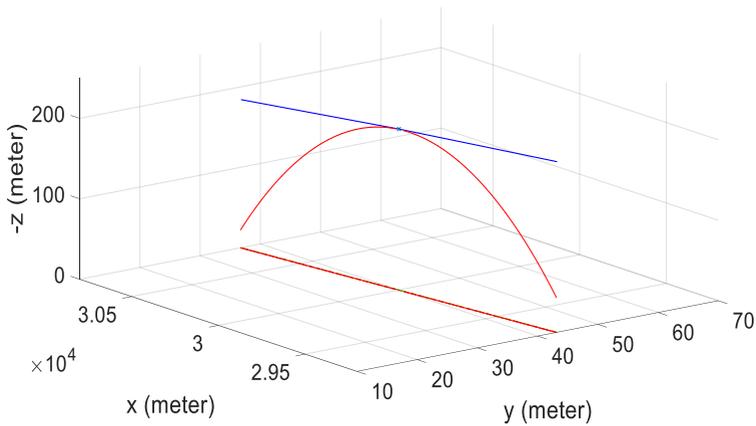
Gambar 4.3 Trayektori dengan *Dubins curve*



Gambar 4.4 Titik belok pada *waypoint 1*



Gambar 4.5 Titik belok pada *waypoint 2*



Gambar 4.6 Titik belok pada *waypoint 3*

4.3. Pengujian Sistem *Autopilot*

Pada pengujian ini akan diuji apakah sistem *autopilot* rudal yang telah dirancang dapat bekerja sesuai yang diinginkan. Karena pada sistem *autopilot* ini menggunakan kontroler *cascade* P-PI maka harus terlebih dahulu didapatkan nilai-nilai parameter *gain* kontroler tersebut. Untuk mendapatkan nilai parameter kontroler, akan dilakukan *tuning* dengan metode prosedural. Metode *tuning* prosedural ini akan diawali dengan *tuning* kontroler *inner loop* baru kemudian dilanjutkan dengan *tuning*

kontroler *outer loop*. Pada tahap *tuning inner loop*, *gain* kontroler *outer loop* akan diberi nilai 1. Dalam melakukan *tuning* prosedural ini, rudal akan memiliki posisi awal pada $[0 \ 0 \ -5]$ dan sudut orientasi $[0 \ 0.245 \ 0.0251]$ serta sistem *autopilot* akan dijalankan setelah fase *booster* selesai. Adapun prosedur *tuning inner loop* yang dilakukan sebagai berikut:

1. Melakukan *tuning* kontroler PI untuk *autopilot pitch* terlebih dahulu dengan kontroler *yaw* dan *roll* diberi *setpoint* nol. Pada *tuning* kontroler *autopilot pitch* ini pertama-tama akan dilakukan analisa respon *closed loop* sistem ketika diberi nilai *gain* proporsional yang diubah-ubah dengan masukan berupa nilai konstan sudut *pitch*. Apabila kemudian apabila terdapat *error steady state* maka ditambahkan dengan *gain integral*.
2. Melakukan *tuning* kontroler PI untuk *autopilot roll* dengan *setpoint* nol karena yang ingin didapatkan adalah *autopilot roll* dapat menjaga kecepatan *roll* pada nilai nol. Pada *tuning autopilot roll* ini, *setpoint pitch* harus lebih dari nol agar rudal bisa selalu terbang di atas permukaan dan *tuning* kontroler PI pada *autopilot roll* dapat dilakukan. Sama seperti pada *autopilot pitch*, akan dilakukan analisa terlebih dahulu terhadap respon *closed loop* dengan nilai *gain* proporsional yang diubah-ubah. Setelah itu untuk meminimumkan *error steady state* akan diberikan *gain integral*.
3. Melakukan *tuning* kontroler PI untuk *autopilot yaw*. Pada saat melakukan *tuning* pada *autopilot yaw* ini akan diberikan *setpoint* konstan pada *autopilot pitch* dan *roll*. Demikian juga pada *autopilot yaw*, akan dianalisa respon *closed loop* sistem dengan *gain* proporsional yang diubah-ubah ketika diberi masukan *setpoint* nilai sudut *yaw* konstan. *Gain integral* kemudian ditambahkan untuk mengurangi *error steady state*.

Setelah melakukan *tuning* prosedural pada *inner loop* sistem *autopilot*, maka akan dilanjutkan dengan *tuning* prosedural untuk *outer loop*. Pada *outer loop* ini yang digunakan hanyalah kontroler proporsional untuk mempercepat respon sistem. Prosedur *tuning outer loop* adalah sebagai berikut.

1. Melakukan *tuning* kontroler proporsional pada *autopilot pitch* dengan menetapkan *gain* kontroler proporsional *yaw* dan *roll* dengan nilai 1. Nilai *gain* proporsional ini akan diperbesar sampai respon sudut *pitch* mengalami *overshoot*, karena kriteria desain yang diinginkan adalah tidak timbul *overshoot* pada respon sistem. Nilai

gain proporsional yang memberikan respon tercepat dan tanpa menimbulkan *overshoot* adalah yang akan digunakan pada *autopilot pitch*.

2. Melakukan *tuning* kontroler proporsional pada *autopilot roll* dengan cara mengubah-ubah nilai parameter *gain* proporsional. Nilai *gain* proporsional yang memberikan respon tercepat adalah yang akan digunakan pada *autopilot roll*.

3. Melakukan *tuning* kontroler proporsional pada *autopilot yaw* dengan cara memperbesar *gain* sampai respon sudut *yaw* mengalami *overshoot*. Nilai *gain* proporsional yang memberikan respon tercepat dan tanpa menimbulkan *overshoot* adalah yang akan digunakan pada *autopilot yaw*.

Dalam melakukan analisa respon pada pengujian sistem *autopilot* ini, kriteria *settling time* yang digunakan adalah *settling time* $\pm 2\%$.

4.3.1. Pengujian Inner Loop Autopilot Pitch

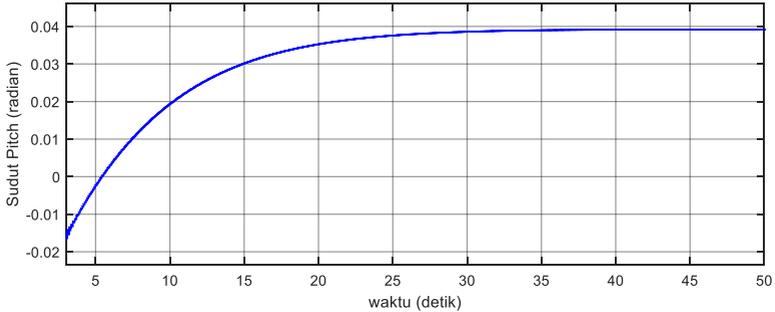
Pengujian *autopilot pitch* dilakukan pertama-tama dengan memberi masukan *setpoint* sudut *pitch* sebesar 0.05 radian. Pada tahap awal, akan diuji sistem *autopilot* dengan hanya kontroler proporsional saja. *Gain* kontroler akan diubah-ubah senilai 3, 5, 7, dan 9. Nilai *gain* kontroler proporsional K_p yang diambil memiliki nilai terkecil adalah 3 karena dengan *gain* K_p dibawah 3 tidak cukup untuk menaikkan sudut *pitch* rudal sehingga rudal terjatuh. Sedangkan untuk nilai K_p lebih dari 10 akan menghasilkan osilasi dan ketidakstabilan pada sistem *autopilot*. Hasil pengujian kontroler proporsional dapat dilihat pada Tabel (4.3).

Tabel 4.3 Kontroler proporsional *autopilot pitch*

K_p	<i>Settling Time</i> (Detik)	<i>Error Steady State</i> (%)
3	60.12	56.70
5	41.70	36.8
7	32.99	27.28
9	23.8	21.7

Dapat dilihat berdasarkan Tabel (4.3) bahwa pemilihan K_p yang terlalu kecil memberikan *settling time* yang paling lama dan *error steady state* yang paling besar. Sedangkan dengan nilai $K_p = 9$ memberikan respon yang paling baik ditunjukkan dengan *settling time* paling kecil dan *error steady state* juga paling kecil. Hasil respon sudut *pitch* rudal dengan

penggunaan kontroler proporsional dengan $K_p = 9$ dapat dilihat pada Gambar (4.7).



Gambar 4.7 Respon sudut *pitch* rudal dengan kontroler proporsional

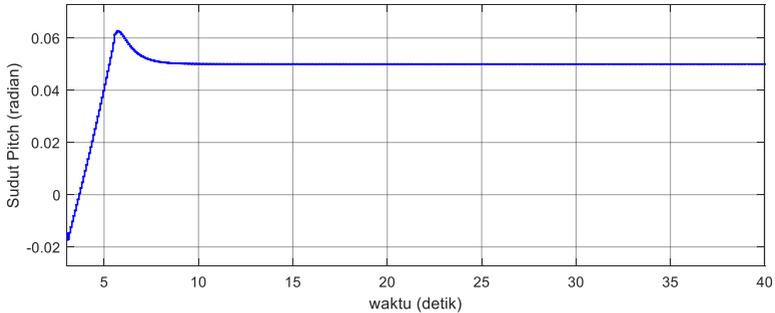
Dari hasil pengujian kontroler proporsional ini maka akan ditetapkan nilai *gain* proporsional yang akan digunakan adalah $K_p = 9$. Untuk mengurangi *error steady state* maka pada sistem *autopilot pitch* ini akan ditambahkan dengan kontroler integral dengan *gain* K_i . Hasil pengujian kontroler proporsional integral (PI) dapat dilihat pada Tabel (4.4).

Tabel 4.4 Kontroler proporsional-integral *autopilot pitch*

K_p	K_i	<i>Settling Time</i> (Detik)	<i>Error Steady State</i> (%)	<i>Overshoot</i> (%)
9	1	36.97	230×10^{-4}	18.89
9	50	8.63	13.2×10^{-4}	20.96
9	100	7.24	5.8×10^{-4}	21.58
9	150	6.65	3.4×10^{-4}	23.14
9	200	6.47	2.6×10^{-4}	25.45

Penggunaan nilai *gain* integral $K_i = 1$ memberikan *settling time* yang berbeda signifikan dengan penggunaan nilai *gain* lainnya. Dapat dilihat bahwa memperbesar *gain* integral akan mempercepat *settling time* dan mengurangi *error steady state* meskipun *percent overshoot* juga ikut bertambah. Dengan pertimbangan bahwa sistem *autopilot pitch* ini membutuhkan kontroler PI dengan *error steady state* terkecil dan *settling time* paling cepat maka nilai K_i yang digunakan adalah $K_i = 200$. Dengan masih adanya *overshoot* seharusnya bisa diatasi dengan

penggunaan kontroler derivatif namun pada sistem *autopilot* rudal ini penambahan kontroler derivatif dengan *gain* berapapun tidak dapat menghilangkan *overshoot* yang timbul pada sistem tetapi menambah osilasi dan *overshoot* pada respon sistem. Hasil respon sudut *pitch* rudal dengan menggunakan kontroler PI dengan *gain* yang telah dipilih dapat dilihat pada Gambar (4.8).



Gambar 4.8 Respon sudut *pitch* rudal dengan kontroler PI

4.3.2. Pengujian *Inner Loop Autopilot Roll*

Pada pengujian *autopilot roll* ini nilai *setpoint* kecepatan *roll* akan selalu diberi nilai 0 radian untuk mencegah terjadinya gerak *roll* pada rudal. Bersamaan dengan pengujian *autopilot roll* ini, nilai *setpoint* sudut *pitch* rudal akan diberi nilai 0.05 radian untuk mencegah rudal jatuh selama pengujian *autopilot roll*. Pengujian sistem *autopilot roll* ini akan dilakukan hanya menggunakan kontroler proporsional terlebih dahulu. Nilai *gain* proporsional K_p yang akan diuji adalah 1, 5, 10 dan 16 dimana pemilihan *gain* hanya sampai 16 adalah karena selebihnya akan menimbulkan respon sistem yang berosilasi dan tidak stabil. Hasil dari pengujian kontroler proporsional pada sistem *autopilot roll* dapat dilihat pada Tabel (4.5).

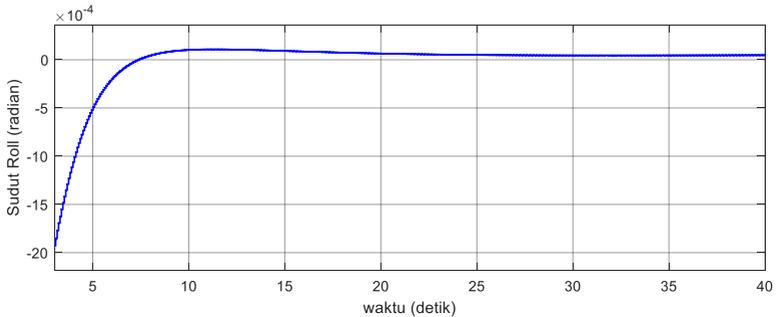
Tabel 4.5 Kontroler proporsional *autopilot roll*

K_p	Settling Time (Detik)	Error Steady State (%)	Overshoot (%)
1	50.39	5.19×10^{-2}	14.43
5	46.05	1.04×10^{-2}	91.32
10	22.94	0.52×10^{-2}	106.78

16	44.62	0.32×10^{-2}	108.87
----	-------	-----------------------	--------

Dapat dilihat bahwa dengan menaikkan nilai K_p maka *error steady state* dan *settling time* akan semakin berkurang kecuali ketika *gain* dinaikkan pada nilai 16. Tetapi seiring dinaikkannya nilai K_p maka *overshoot* yang terjadi akan semakin besar pula. Dengan pertimbangan bahwa kontroler yang dibutuhkan adalah yang memiliki *settling time* tercepat dan *error steady state* yang paling minimum maka nilai *gain* proporsional yang digunakan adalah $K_p = 10$. Hasil pengujian kontroler proporsional *autopilot roll* dengan $K_p = 10$ dapat dilihat pada Gambar (4.9).

Sama seperti pada *autopilot pitch*, untuk memperbaiki kesalahan *error steady state* akan ditambahkan dengan kontroler integral sehingga kontroler yang ada di dalam sistem *autopilot roll* adalah juga kontroler proporsional integral. Hasil pengujian *autopilot roll* dengan kontroler PI dapat dilihat pada Tabel (4.6).



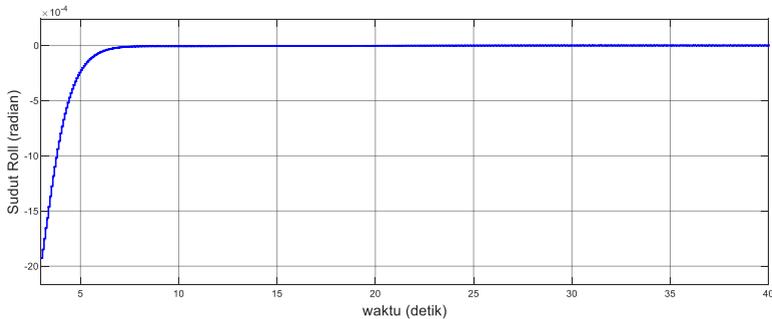
Gambar 4.9 Respon sudut *roll* rudal dengan kontroler proporsional

Tabel 4.6 Kontroler proporsional-integral *autopilot roll*

K_p	K_i	<i>Settling Time</i> (Detik)	<i>Error Steady State</i> (%)	<i>Overshoot</i> (%)
10	1	77.78	2.571×10^{-7}	99.80
10	15	58.57	0.107×10^{-7}	54.29
10	30	26.44	0.05×10^{-7}	73.63
10	50	25.50	0.0058×10^{-7}	1.010

Dari hasil pengujian didapatkan bahwa dengan menaikkan nilai *gain* integral K_i akan didapatkan *settling time* yang lebih cepat dan *error*

steady state yang semakin kecil pula. *Overshoot* secara keseluruhan menjadi lebih berkurang dengan adanya kontroler integral apabila dibandingkan dengan *overshoot* respon yang hanya menggunakan kontroler proporsional. Dari hasil pengujian ini nilai $K_i = 50$ adalah yang akan digunakan dengan pertimbangan *settling time* tercepat, *error steady state* yang paling minimum dan *overshoot* yang terkecil. Hasil respon sudut *roll* rudal dengan menggunakan kontroler PI pada *autopilot roll* ini dengan menggunakan nilai parameter *gain* yang telah dipilih dapat dilihat pada Gambar (4.10). Selain itu, *settling time* pada *autopilot roll* ini terlihat lebih lambat namun karena *error steady state* yang sangat kecil sehingga bukan menjadi masalah bagi sistem *autopilot roll*.



Gambar 4.10 Respon sudut *roll* rudal dengan kontroler PI

4.3.3. Pengujian *Inner Loop Autopilot Yaw*

Pengujian *autopilot yaw* dilakukan dengan memberikan masukan nilai *setpoint* sudut *yaw* konstan sebesar 0.05 radian. Pengujian *autopilot yaw* ini dapat dilakukan dengan menjaga sudut *pitch* sebesar 0.05 radian dan sudut *roll* sebesar 0 radian dengan menggunakan *autopilot pitch* dan *yaw* yang telah didapatkan parameter kontrolernya. Hal ini dilakukan agar rudal selalu berada di udara dan tidak memiliki gerak *roll*. Sama seperti *autopilot* lainnya, akan dicari terlebih dahulu parameter *gain* kontroler proporsional pada *autopilot yaw* ini barulah ditambahkan dengan kontroler integral. Nilai K_p yang akan diuji pada *autopilot yaw* ini adalah 3, 5, 7, dan 11. Nilai paling rendah adalah 3 karena di bawah nilai 3 tidak cukup untuk mengendalikan sudut *yaw* rudal. Sedangkan nilai di atas 11 akan menghasilkan respon sudut *yaw* yang berosilasi dan tidak stabil. Hasil pengujian kontroler proporsional pada *autopilot yaw* dapat dilihat

pada Tabel (4.7). Hasil pengujian *autopilot yaw* dengan kontroler proporsional ini dapat dilihat pada Gambar (4.11).

Tabel 4.7 Kontroler proporsional *autopilot yaw*

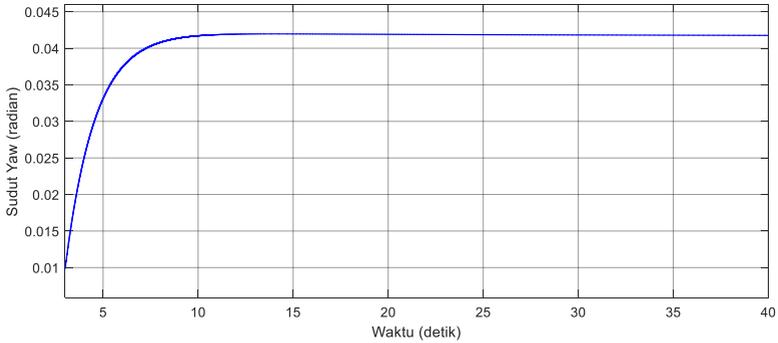
K_p	<i>Settling Time</i> (Detik)	<i>Error Steady State</i> (%)
3	11.90	64.0
5	11.35	36.27
7	10.10	25.92
11	8.98	16.5

Dapat dilihat berdasarkan Tabel (4.7) bahwa dengan menaikkan *gain* proporsional maka *settling time* akan menjadi lebih cepat dan *error steady state* berkurang. Nilai K_p yang akan digunakan untuk *autopilot yaw* adalah $K_p = 11$. Untuk mengurangi *error steady state* akan maka kontroler akan ditambahkan komponen integral dengan *gain* K_i . Hasil pengujian *autopilot yaw* dengan kontroler proporsional integral dapat dilihat pada Tabel (4.8).

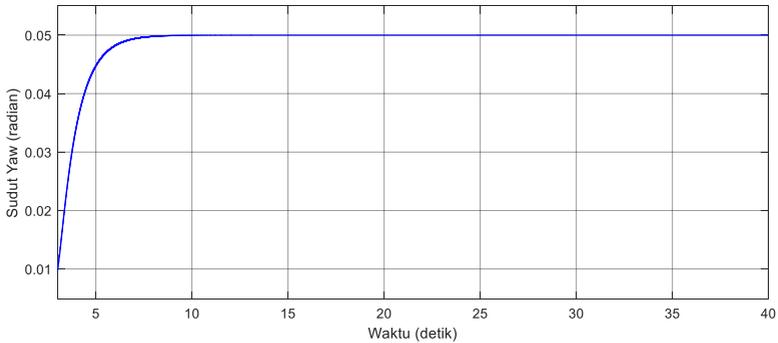
Tabel 4.8 Kontroler proporsional-integral *autopilot yaw*

K_p	K_i	<i>Settling Time</i> (Detik)	<i>Error Steady State</i> (%)
11	1	9.35	0.15
11	50	5.65	0.25×10^{-4}
11	100	5.76	0.12×10^{-4}
11	200	6.02	0.06×10^{-4}

Dari hasil pengujian, didapatkan data bahwa nilai $K_i = 1$ memiliki *settling time* yang paling lambat dan *error steady state* yang paling besar. Sedangkan mulai dari nilai *gain* $K_i = 50$ ke atas, semakin besar nilai K_i akan memberikan *settling time* yang semakin besar namun *error steady state* yang mengecil. Dengan pertimbangan *settling time* yang cukup cepat serta *error steady state* yang cukup kecil maka yang akan digunakan adalah nilai $K_i = 100$. Hasil pengujian *autopilot yaw* dengan kontroler PI dapat dilihat pada Gambar (4.12).



Gambar 4.11 Respon sudut *yaw* rudal dengan kontroler proporsional



Gambar 4.12 Respon sudut *yaw* rudal dengan kontroler PI

4.3.4. Pengujian *Outer Loop Autopilot Pitch*

Setelah mendapatkan parameter kontroler PI pada *inner loop* masing-masing *autopilot* maka dilanjutkan dengan mencari nilai parameter *gain* proporsional pada *outer loop* dari masing-masing *autopilot*. Sama seperti pada *tuning inner loop*, akan dimulai dari *autopilot pitch*. Pada pengujian ini akan diuji ketika *gain* proporsional diubah-ubah seperti pada Tabel (4.9). Berdasarkan Tabel (4.9) nilai *gain* K_p yang dipilih adalah $K_p = 1.5$ karena memberikan *settling time* yang paling cepat.

Tabel 4.9 Kontroler proporsional *autopilot pitch*

K_p	Settling Time (Detik)
1	5.32
1.3	4.63
1.5	4.32

4.3.5. Pengujian *Outer Loop Autopilot Roll*

Sama seperti *autopilot pitch*, pada pengujian ini akan dicari nilai *gain* proporsional yang memberikan respon *settling time* tercepat. Nilai *gain* proporsional yang telah diuji dapat dilihat pada Tabel (4.10). Dari hasil pengujian akan dipilih nilai *gain* $K_p = 1.5$.

Tabel 4.10 Kontroler proporsional *autopilot roll*

K_p	Settling Time (Detik)
1	25.50
1.3	24.95
1.5	24.20

4.3.6. Pengujian *Outer Loop Autopilot Yaw*

Pada pengujian ini juga akan dilakukan pengubahan nilai *gain* proporsional dari *autopilot yaw* pada *outer loop*. Nilai *gain* yang telah diuji dapat dilihat pada Tabel (4.11). Dari hasil pengujian yang telah dilakukan, nilai *gain* proporsional yang akan digunakan adalah $K_p = 3$.

Tabel 4.11 Kontroler proporsional *autopilot yaw*

K_p	Settling Time (Detik)
1	6.02
2	4.55
3	3.91

Dalam melakukan pengujian sistem *autopilot* ini seharusnya dilakukan analisa terhadap *time constant* dari *inner loop* dan juga *outer loop* untuk mendapatkan respon sistem yang stabil, dimana *time constant* dari *inner loop* harus sekurang-kurangnya 3 kali lebih cepat daripada *time constant outer loop*. Namun pada sistem *autopilot* rudal yang digunakan ini tidak dapat dilakukan pengujian secara terpisah antara *inner loop* dan *outer loop*, karena *plant* rudal yang ingin dikendalikan sifatnya memiliki

orde tinggi dan *nonlinier*, sehingga *inner* dan *outer loop* harus selalu bekerja secara bersamaan.

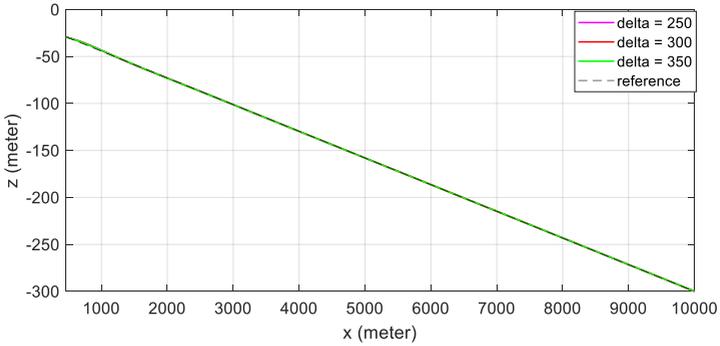
4.4. Pengujian Sistem Pemanduan Rudal

Pengujian sistem pemanduan rudal akan dilakukan pengujian sistem pemanduan pada trayektori berupa garis lurus, trayektori lingkaran, dan trayektori gabungan. Tujuan dari pengujian ini adalah memeriksa apakah sistem pemanduan mampu mendapatkan sudut-sudut orientasi rudal referensi yang tepat sehingga rudal dapat bergerak mengikuti trayektori yang ada. Keluaran dari sistem pemanduan ini akan dimasukkan menjadi *set point* pada sistem *autopilot* yang telah diuji sebelumnya untuk mengendalikan sirip-sirip rudal. Pada pengujian ini parameter jarak sasaran *virtual* (δ dan λ) akan diubah-ubah untuk melihat pengaruhnya terhadap pergerakan rudal.

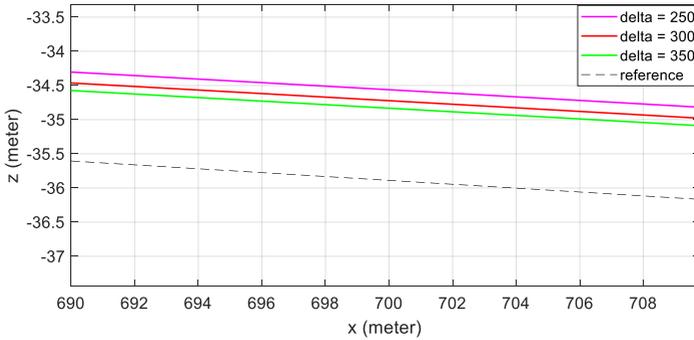
4.4.1. Pengujian Sistem Pemanduan Garis Lurus

Pada pengujian garis lurus, akan digunakan konfigurasi awal rudal yang sama yaitu posisi awal rudal tidak dari posisi [0 0 0] melainkan dari posisi dimulainya sistem pemanduan setelah fase *booster* selesai. Oleh karena itu, akan digunakan konfigurasi awal skenario 3 pada subbab sebelumnya dimana rudal berada pada posisi [462 8.55 -29.13] dan rudal memiliki sudut orientasi [0 0.0306 0.01]. Posisi dan orientasi rudal yang didapatkan ini didapatkan berdasarkan kondisi awal rudal sebelum diluncurkan pada posisi [0 0 -5] dan sudut orientasi [0 0.245 0.0251]. Setelah menentukan posisi dan orientasi pada saat $t = 3$ detik, maka bisa ditentukan titik *waypoint* awal (W_1) rudal adalah [462 8.55 -29.13] dan *waypoint* akhir rudal (W_2) adalah [10000 100 -300]. Hasil pengujian sistem pemanduan garis lurus ini dapat dilihat pada Gambar (4.13) dan Gambar (4.14).

Dapat dilihat dari Gambar (4.13a) dan Gambar (4.13b) bahwa pemanduan sudut *pitch* sudah dapat memberikan pemanduan sudut yang baik kepada sistem *autopilot pitch* sehingga rudal dapat bergerak mengikuti lintasan garis lurus. Nilai parameter δ yang diambil dan diuji adalah 250, 300, dan 350. *Overshoot* paling besar terjadi pada saat $\delta = 250$ dan $\delta = 350$ memberikan *overshoot* yang paling minimum.



(a)



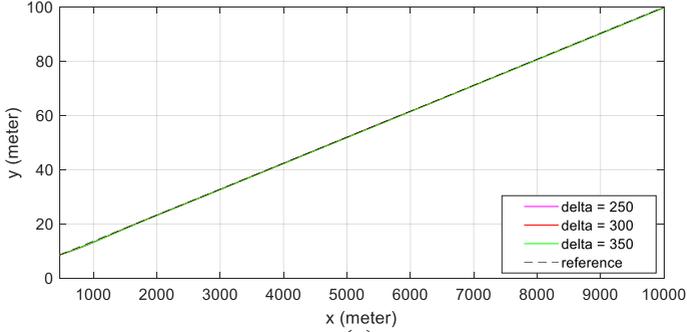
(b)

Gambar 4.13 Sistem pemanduan *pitch* garis lurus (a) tampilan keseluruhan dan (b) tampilan diperbesar pada saat terjadinya *overshoot*

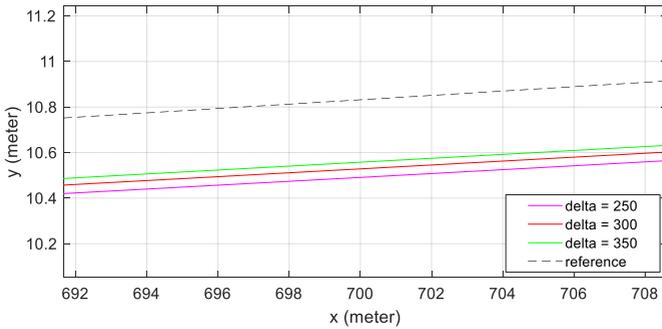
Sedangkan berdasarkan Gambar (4.14a) dan Gambar (4.14b) dapat dilihat bahwa pemanduan sudut *yaw* juga telah memberikan pemanduan sudut yang baik kepada sistem *autopilot yaw* sehingga rudal dapat mengikuti lintasan garis lurus. Sama seperti pemanduan *autopilot pitch*, $\delta = 250$ memberikan hasil *overshoot* yang paling besar dan $\delta = 350$ memberikan *overshoot* yang paling minimum.

Dari kedua pemanduan *yaw* dan *pitch* garis lurus, dapat dilihat bahwa nilai δ yang semakin kecil akan menimbulkan *overshoot* yang lebih besar. Hal ini dikarenakan letak sasaran *virtual* yang bergerak sepanjang lintasan memiliki jarak yang semakin dekat terhadap posisi

rudal sehingga pemanduan akan mengeluarkan referensi sudut yang lebih besar. Dengan referensi sudut yang lebih besar, maka akan timbul osilasi yang lebih besar pula.



(a)



(b)

Gambar 4.14 Sistem pemanduan yaw garis lurus (a) tampilan keseluruhan dan (b) tampilan diperbesar pada saat terjadinya *overshoot*

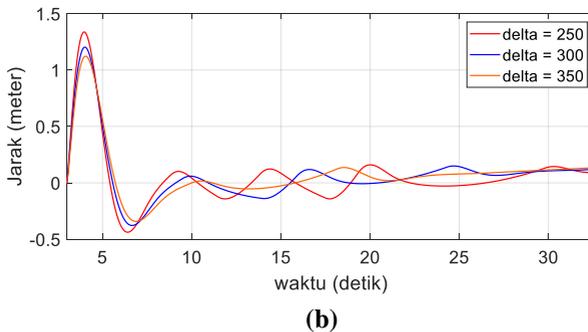
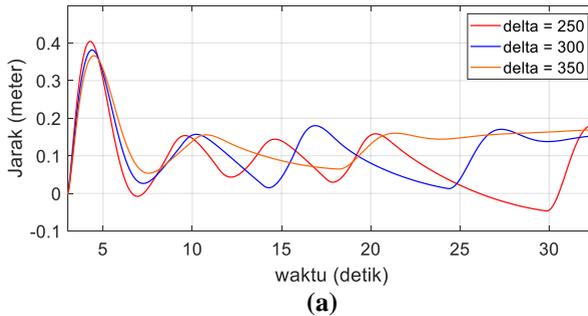
Dari pengujian yang telah dilakukan, untuk menentukan nilai δ yang paling bagus untuk digunakan akan dianalisa *error* atau jarak relatif rudal secara orthogonal terhadap lintasan referensi pada bidang yaw dan *pitch* berdasarkan perhitungan *root mean squared error* (RMSE). *Error yaw* disebut dengan *cross-track error* dan *error pitch* disebut *vertical-track error*. Hasil perhitungan *cross-track error* dan *vertical-track error* dapat dilihat pada Tabel (4.9) dan (Tabel 4.10) dan Gambar (4.15).

Tabel 4.12 *Cross-Track error* pemanduan garis lurus

δ (meter)	RMSE (meter)	<i>Error steady state</i> (meter)
250	0.1244	0.1298
300	0.1381	0.1472
350	0.1524	0.1671

Tabel 4.13 *Vertical-Track error* pemanduan garis lurus

δ (meter)	RMSE (meter)	<i>Error steady state</i> (meter)
250	0.2851	0.0919
300	0.2653	0.1197
350	0.2521	0.1344



Gambar 4.15 Grafik (a) *cross-track error* dan (b) *vertical-track error* rudal terhadap trayektori referensi

Berdasarkan Tabel (4.9) dapat dilihat bahwa pada sistem pemanduan *yaw*, nilai RMSE, *error steady state*, dan *settling time* akan meningkat ketika nilai δ yang semakin besar. Sedangkan pada Tabel (4.10) menunjukkan bahwa pada sistem pemanduan *pitch* dengan menaikkan nilai δ , *error steady state* dan *settling time* akan naik akan tetapi nilai RMSE semakin mengecil. Dari kedua pemanduan *yaw* dan *pitch* ini dapat dilihat bahwa nilai δ yang semakin besar akan menimbulkan *error steady state* yang semakin besar yang dikarenakan jarak sasaran virtual terhadap rudal lebih besar sehingga pemanduan akan memberikan sudut referensi yang semakin kecil pada saat *steady state*.

Dapat dilihat pula dari Gambar (4.15). *cross-track* dan *vertical-track error* rudal akan membesar seiring pergerakan rudal menuju *steady state* pada saat menggunakan nilai δ yang lebih besar meskipun telah keluar dari *sliding surface*. Hal ini disebabkan karena pada saat rudal telah berada dalam kondisi *steady state*, nilai sudut referensi yang dihasilkan pemanduan untuk mencapai titik sasaran sangatlah kecil sehingga ketika sistem *autopilot* rudal menerima sudut referensi tersebut akan dianggap mendekati nol dan sirip-sirip rudal tidak diberikan sinyal kontrol.

Secara keseluruhan, berdasarkan nilai RMSE yang didapatkan dari semua nilai δ yang digunakan dapat dikatakan sistem pemanduan telah bekerja dengan baik mengingat pada algoritma *sliding curve* ini lebar *sliding surface* adalah sebesar diameter rudal yaitu 0.28 meter dimana rudal akan menyimpang dengan jarak tertentu dari *waypoint* yang telah ditentukan. Selisih jarak rudal terhadap *waypoint* (*miss distance*) pada saat melintasi *waypoint* dapat dilihat pada Tabel 4.11. Selisih jarak terkecil terdapat pada nilai $\delta = 250$ dan yang terbesar pada $\delta = 350$.

Tabel 4.14 Selisih jarak rudal ketika melewati *waypoint*

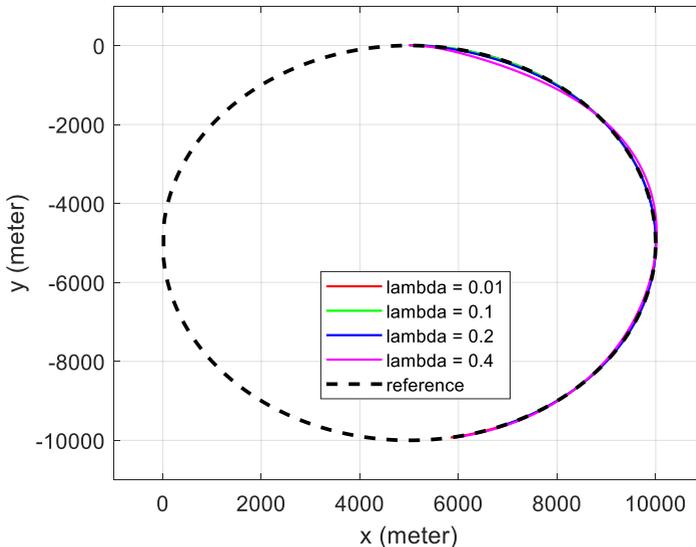
δ (meter)	Selisih Jarak rudal-waypoint (meter)
250	0.0909
300	0.1691
350	0.2760

4.4.2. Pengujian Sistem Pemanduan Lingkaran

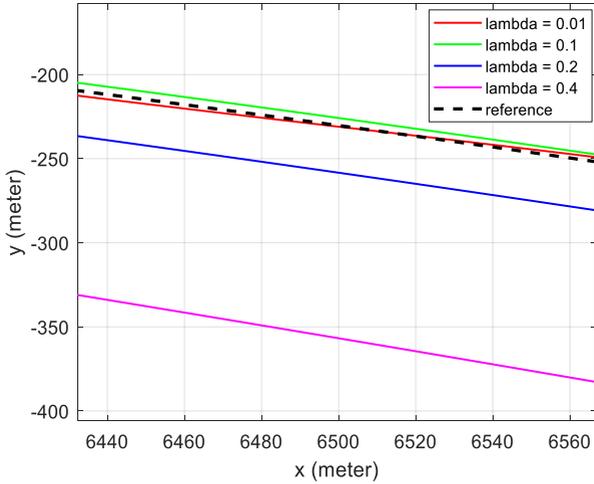
Dalam melakukan pengujian sistem pemanduan lingkaran ini akan diasumsikan bahwa rudal hanya akan melakukan gerak belok maksimum sebesar 180° sehingga pengujian dilakukan terhadap trayektori setengah

lingkaran. Sebelum melakukan pengujian sistem pemanduan lingkaran, akan dicari terlebih dahulu radius putar terkecil dari rudal dengan cara memberi masukan *setpoint* sudut 180° pada *autopilot rudal*. Didapatkan radius putar minimal rudal adalah sebesar 1500 meter dengan rentang defleksi sirip -10° hingga 10° . Dengan radius putar ini, akan diasumsikan pula rudal tidak mungkin melakukan manuver gerak *pitch* mengikuti trayektori lingkaran radius 1500 meter. Oleh karena itu, yang akan dilakukan pengujian adalah untuk gerak *yaw* rudal.

Pengujian sistem pemanduan lingkaran ini akan menggunakan trayektori referensi berupa lingkaran dengan radius 5000 meter dan titik pusatnya berada pada $[5000 \ -5000 \ -10]$, titik awal rudal berada pada $[5000 \ 0 \ -10]$ dan sudut orientasi rudal adalah $[0 \ 0 \ 0.001]$. Nilai parameter λ pada sistem pemanduan akan diubah-ubah untuk melihat perbedaan yang terjadi pada pergerakan rudal. Nilai λ yang digunakan dalam pengujian sistem pemanduan lingkaran ini adalah 0.01, 0.1, 0.2, 0.4. Hasil pengujian sistem pemanduan lingkaran dapat dilihat pada Gambar (4.16) dan *cross-track error* pada Gambar (4.17).

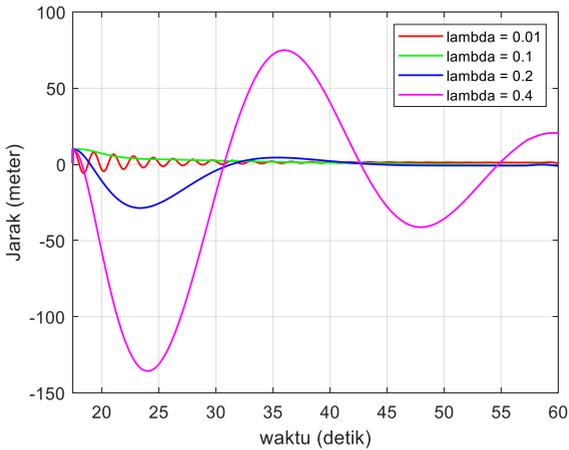


(a)



(b)

Gambar 4.16 Hasil pengujian sistem pemanduan yaw lintasan lingkaran (a) tampilan keseluruhan dan (b) tampilan diperbesar saat posisi $x = 6500$



Gambar 4.17 Cross-Track error sistem pemanduan yaw lingkaran dengan nilai λ yang berbeda-beda.

Tabel 4.15 *Cross-Track error* pemanduan lingkaran

λ (radian)	RMSE (meter)	<i>Error steady state</i> (meter)
0.01	1.8635	1.220
0.1	2.6438	0.3917
0.2	8.9988	0.8221
0.4	50.6154	—

Dari gambar di atas dapat dilihat bahwa nilai λ yang terlalu kecil akan menyebabkan osilasi kecil pada pemanduan karena titik sasaran *virtual* yang harus dicapai oleh rudal memiliki sudut fase yang hampir sama dalam lingkaran sedangkan untuk nilai λ yang terlalu besar akan menyebabkan rudal berosilasi dan menyimpang jauh dari lintasan referensi. Berdasarkan nilai RMSE, nilai $\lambda = 0.01$ adalah yang terkecil namun memiliki *error steady state* yang paling besar dan *settling time* paling lambat. Sedangkan pada nilai $\lambda = 0.04$ nilai RMSE sangat besar karena rudal bergerak menyimpang dari lintasan referensi dan tidak mencapai *steady state* karena cenderung berosilasi. Dengan asumsi bahwa yang dibutuhkan adalah nilai *error steady state* yang terkecil, RMSE cukup kecil, dan *settling time* yang cukup cepat maka nilai $\lambda = 0.1$ adalah yang terbaik dan akan digunakan pada pemanduan lingkaran rudal.

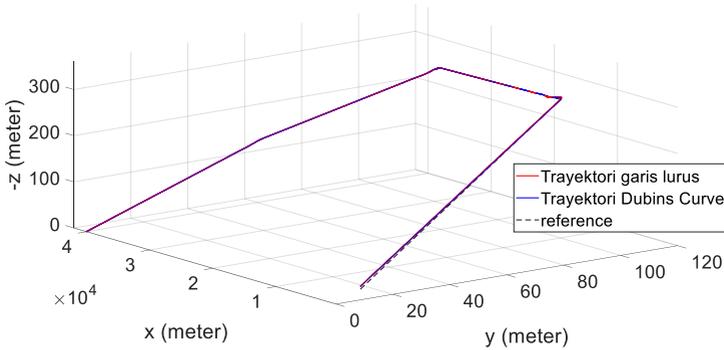
4.4.3. Pengujian Sistem Pemanduan Gabungan Dengan Trayektori Referensi yang Berbeda

Setelah menguji sistem pemanduan pada garis lurus dan lingkaran, akan dilakukan pengujian sistem pemanduan gabungan atau keseluruhan. Pengujian ini dilakukan dengan menggunakan 4 buah *waypoint* seperti pada Tabel (4.13). Nilai δ dan λ yang digunakan pada pengujian keseluruhan ini adalah $\delta = 250$ meter dan $\lambda = 0.1$ radian. Hasil pengujian sistem pemanduan keseluruhan ini kemudian akan dibandingkan ketika lintasan referensinya berupa lintasan garis lurus dan ketika lintasan referensinya adalah *Dubins curve*. Hasil simulasi sistem pemanduan gabungan dapat dilihat pada Gambar (4.18).

Tabel 4.16 Pengujian sistem pemanduan gabungan

<i>Waypoint</i>	Posisi
0	[462 8.55 -23.19]

1	[10000 150 -300]
2	[20000 80 -350]
3	[30000 40 -200]
4	[40000 0 0]

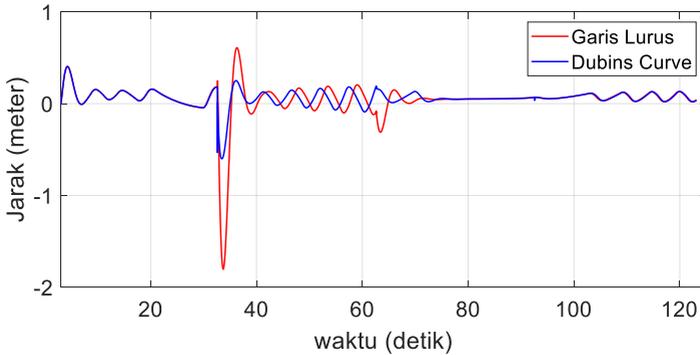


Gambar 4.18 Hasil simulasi sistem pemanduan gabungan dengan 4 titik *waypoints* (termasuk sasaran)

Dapat dilihat bahwa sistem pemanduan sudah dapat memandu rudal untuk bisa bergerak mengikuti lintasan referensi yang ada. Pada saat lintasan referensinya berupa garis lurus saja, akan terjadi *overshoot* yang lebih besar ketika rudal melakukan manuver belok pada masing-masing *waypoint*. Hal ini disebabkan karena lintasan referensi yang seolah-olah patah ketika rudal telah mencapai titik *waypoint* tersebut. Sedangkan *overshoot* pada lintasan referensi *Dubins curve* memberikan gerakan yang lebih mulus, karena rudal akan mengikuti lintasan lingkaran ketika melewati titik *tangent entry* dari lingkaran *Dubins* referensi pada masing-masing *waypoint*. Untuk melihat lebih jelas dari perbedaan antara kedua lintasan referensi ini dapat dilihat dari *cross-track error* dan *vertical-track error* rudal terhadap lintasan referensi pada Gambar (4.19) dan Gambar (4.20) beserta nilai RMSE-nya pada Tabel (4.14) dan Tabel (4.15).

Tabel 4.17 *Cross-Track error* pemanduan gabungan

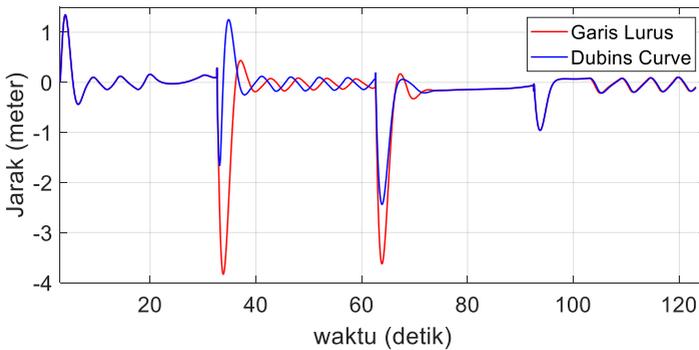
Jenis Lintasan Referensi	RMSE (meter)
Garis Lurus	0.2147
<i>Dubins curve</i>	0.1160



Gambar 4.19 Grafik *cross-track error* pemanduan gabungan dengan trayektori referensi yang berbeda

Tabel 4.18 *Vertical-Track error* pemanduan gabungan

Jenis Lintasan Referensi	RMSE (meter)
Garis Lurus	0.6429
<i>Dubins curve</i>	0.2972



Gambar 4.20 Grafik *Vertical-Track error* pemanduan gabungan dengan trayektori referensi yang berbeda

Dapat berdasarkan Tabel (4.14) dan Tabel (4.15) dilihat bahwa pada saat menggunakan lintasan referensi berupa *Dubins curve* memberikan nilai RMSE yang lebih kecil dibandingkan dengan garis lurus. Dari pengujian yang telah dilakukan membuktikan bahwa dengan

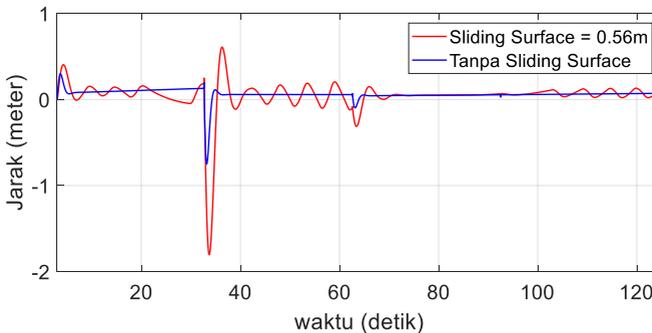
pengubahan bentuk trayektori referensi rudal akan memberi pengaruh yang cukup signifikan terhadap pergerakan rudal.

Tabel 4.19 *Miss Distance* rudal terhadap *waypoint*

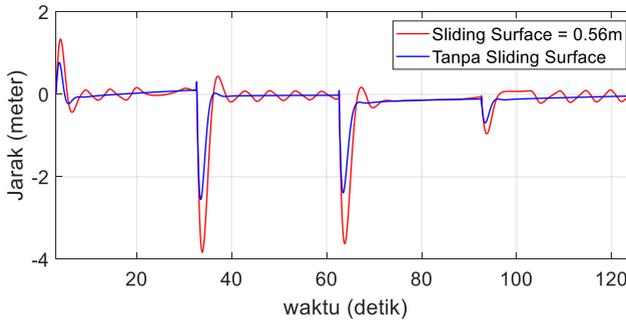
<i>Waypoint</i>	Garis Lurus	<i>Dubins Curve</i>
1	0.4468	0.4139
2	0.7516	0.6206
3	0.5387	0.2449
4	0.4967	0.2398
Rata-Rata	0.5585	0.3798

4.4.4. Pengujian Sistem Pemanduan Gabungan tanpa *Sliding Surface* dan dengan *Sliding Surface*

Setelah melakukan pengujian dengan trayektori referensi yang berbeda, maka akan kemudian akan dilakukan pengujian ketika sistem pemanduan yang digunakan memiliki *sliding surface* dengan lebar 0.56 meter dan dengan lebar 0 meter atau bisa dikatakan tanpa *sliding surface*. Pada pengujian ini, trayektori referensi yang digunakan adalah sama yaitu menggunakan trayektori garis lurus untuk bisa melihat pengaruh lebar *sliding surface* yang ditentukan. Hasil pengujian ini dapat dilihat pada Gambar (4.21) dan Gambar (4.22) serta *cross-track error* dan *vertical-track error* pada Tabel (4.20) dan Tabel (4.21).



Gambar 4.21 Grafik *Cross-Track error* dengan lebar *sliding surface* yang berbeda



Gambar 4.22 Grafik *Vertical-Track error* dengan lebar *sliding surface* yang berbeda

Tabel 4.20 *Cross-Track error* pemanduan gabungan

Lebar <i>Sliding Surface</i> (meter)	RMSE (meter)
0.00	0.0941
0.56	0.2147

Tabel 4.21 *Vertical-Track error* pemanduan gabungan

Lebar <i>Sliding Surface</i> (meter)	RMSE (meter)
0.00	0.3861
0.56	0.6429

Dapat dilihat bahwa pada saat menggunakan *sliding surface* dengan lebar 2 kali diameter, rudal memiliki pergerakan yang cenderung berosilasi di dalam selubung *sliding surface* tersebut. Sedangkan pada saat tidak menggunakan *sliding surface* memberikan pergerakan rudal yang lebih stabil yang tidak berosilasi. Hal ini tentu disebabkan oleh sistem pemanduan rudal yang selalu aktif apabila tidak adanya *sliding surface*, dan ketika ada *sliding surface* maka sistem pemanduan rudal hanya akan aktif ketika rudal berada di dalam *sliding surface*.

Pada pengujian ini nilai *root-mean-square-error* (RMSE) dari *cross-track error* dan *vertical-track error* yang telah didapatkan, keduanya lebih kecil pada saat tidak menggunakan *sliding surface* apabila dibandingkan dengan penggunaan *sliding surface* 0.56 meter. Untuk *cross-track error* memiliki selisih sebesar 0.1206 meter dan 0.2568 meter untuk *vertical-track error*.

[Halaman ini sengaja dikosongkan]

BAB 5 PENUTUP

5.1 Kesimpulan

Dari hasil simulasi yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut.

1. Sistem pemanduan *sliding curve* dapat digunakan untuk pemanduan rudal dengan sasaran statis dan trayektori referensi deterministik. Hal ini dapat dilihat dari gambar hasil simulasi menggunakan Simulink dan nilai *miss distance* rata-rata yang kecil yaitu 0.5585 meter pada trayektori garis lurus dan 0.3798 meter pada trayektori *Dubins curve*.
2. Pemilihan lebar *sliding surface* pada sistem pemanduan *sliding curve* berpengaruh terhadap mulus tidaknya pergerakan rudal sepanjang trayektori, semakin lebar *sliding surface* maka pergerakan rudal akan semakin beresilasi. Hal ini dapat dilihat berdasarkan pergerakan rudal yang cenderung beresilasi pada radius *sliding surface* senilai 0.28 meter.
3. Penentuan jarak δ dan λ mempengaruhi osilasi pergerakan rudal serta *error steady state* rudal terhadap trayektori referensi. Nilai δ dan λ yang terlalu kecil menimbulkan osilasi berlebih pada rudal karena titik sasaran terlalu dekat dengan posisi rudal sedangkan apabila nilainya terlalu besar maka rudal membutuhkan waktu yang lama untuk mencapai trayektori referensi. Hal ini dibuktikan dari hasil simulasi pada Gambar (4.15).
4. Perencanaan trayektori berpengaruh terhadap kinerja sistem pemanduan. Apabila perencanaan trayektori mampu menghasilkan titik belok yang mulus maka sistem pemanduan akan bisa memandu rudal dengan pergerakan yang mulus pula. Apabila pada perencanaan trayektori terdapat titik belok yang patah antar 2 *waypoint* maka rudal akan mengalami *overshoot* pada titik belok tersebut. Hal ini didapatkan dari melihat nilai rata-rata *miss distance* yang dihasilkan dari *Dubins curve* adalah lebih kecil 0.1787 meter dibandingkan dengan menggunakan trayektori garis lurus.
5. Penggunaan kontroler *cascade* P-PI dapat digunakan sebagai pengendali pada sistem *autopilot* rudal dan memiliki kelebihan bahwa bisa dilakukan *tuning* prosedural terhadap *plant* dengan

orde tinggi dan *nonlinier*. Hal ini ditunjukkan dari hasil simulasi pengujian bahwa *autopilot* rudal memiliki *settling time* yang cukup cepat dan *error steady state* yang mendekati nol.

5.2 Saran

Untuk pengembangan selanjutnya pada topik penelitian rudal, terdapat beberapa saran yang diberikan, antara lain:

1. Melakukan metode pengendalian *autopilot* yang berbeda dari kontroler proporsional-integral biasa seperti kontroler prediktif.
2. Menggunakan metode pemanduan rudal yang berbeda untuk mendapatkan hasil yang lebih baik.
3. Menggunakan koefisien aerodinamika serta parameter rudal yang sesuai untuk mendapatkan simulasi rudal yang akurat.

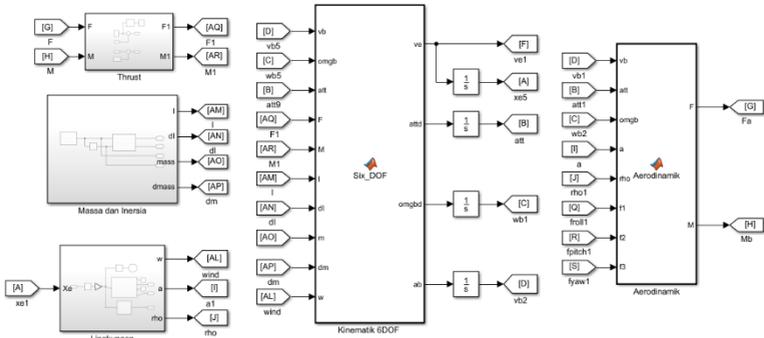
DAFTAR PUSTAKA

- [1] Shanmugavel, Madhavan. "Path planning of multiple autonomous vehicles." (2007).
- [2] Siouris, George M. *Missile guidance and control systems*. Springer Science & Business Media, 2004.
- [3] Özkan, Bülent. "Dynamic modeling, guidance, and control of homing missiles." *The Graduate School of Natural and Applied Sciences of Middle East Technical University* (2005).
- [4] Sujit, P. B., Srikanth Saripalli, and Joao Borges Sousa. "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles." *IEEE Control Systems Magazine* 34.1 (2014): 42-59.
- [5] Lugo-Cárdenas, Israel, et al. "Dubins path generation for a fixed wing UAV." *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014.
- [6] Dubins, Lester E. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents." *American Journal of Mathematics* 79.3 (1957): 497-516.
- [7] Yanushevsky, Rafael. *Modern missile guidance*. CRC Press, 2007.
- [8] Xavier, Daniel M., BF Natassya Silva, and Kalinka RLJC Branco. "Comparison of path-following algorithms for loiter paths of Unmanned Aerial Vehicles." *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018.
- [9] AK, Rusdhianto Effendie, Muhamad Rafif Prasetyo, and Zulkifli Hidayat. "Designing an Optimal Control LQT for Controlling and Guidance of Missile." *JAREE (Journal on Advanced Research in Electrical Engineering)* 2.1 (2018).
- [10] B. Medagoda, E.D. and Gibbens, P.W., 2010. Synthetic-waypoint guidance algorithm for following a desired flight trajectory. *Journal of guidance, control, and dynamics*, 33(2), pp.601-606.

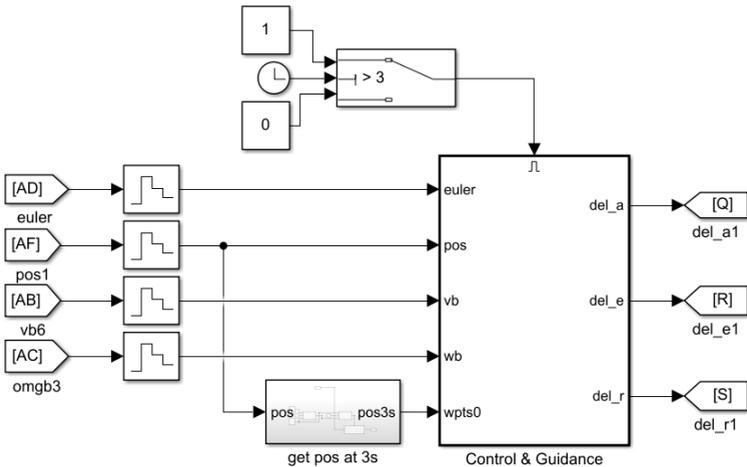
[Halaman ini sengaja dikosongkan]

LAMPIRAN

Berikut ini adalah lampiran berupa blok Simulink secara keseluruhan yang digunakan pada Tugas Akhir ini:



L1. Blok Simulink untuk Model Matematik Rudal



L2. Blok Simulink untuk Pemandu dan Autopilot

[Halaman ini sengaja dikosongkan]

RIWAYAT HIDUP PENULIS



Leonrd Wihardi lahir pada tanggal 6 Juli 1997 di kota Surabaya, Jawa Timur, Indonesia. Setelah menyelesaikan pendidikan sekolah dasar pada tahun 2009, sekolah menengah pada tahun 2012, dan sekolah menengah atas pada tahun 2015, penulis meneruskan pendidikannya di Institut Teknologi Sepuluh Nopember, Surabaya di departemen Teknik Elektro, Fakultas Teknologi Elektro. Penulis memiliki minat pada teknologi otomasi, robotika, teknik pengaturan, dan teknik sistem membuat penulis memilih teknik Sistem pengaturan sebagai bidang studi. Hingga saat ini, penulis terus berusaha agar menjadi seorang yang bermanfaat bagi orang lain, bangsa dan negara.