



TUGAS AKHIR - TE141599

**PERANCANGAN KENDALI OPTIMAL LQT UNTUK
PENGENDALIAN DAN PEMANDUAN PADA RUDAL**

Muhamad Rafif Prasetyo
NRP 0711134000129

Dosen Pembimbing
Ir. Rusdhianto Effendie AK, M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE141599

***DESIGNING AN OPTIMAL CONTROL LQT FOR
CONTROLLING AND GUIDANCE OF MISSILE***

Muhamad Rafif Prasetyo
NRP 07111340000129

Supervisor
Ir. Rusdhianto Effendie AK, M.T.

***ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Perancangan Kendali Optimal LQT untuk Pengendalian dan Pemanduan pada Rudal”** 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 mandiri.

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, Januari 2018

Muhamad Rafif Prasetyo
NRP 0711134000129

Halaman ini sengaja dikosongkan

**PERANCANGAN KENDALI OPTIMAL LQT UNTUK
PENGENDALIAN DAN PEMANDUAN
PADA RUDAL**

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,



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



Halaman ini sengaja dikosongkan

PERANCANGAN KENDALI OPTIMAL LQT UNTUK PENGENDALIAN DAN PEMANDUAN PADA RUDAL

Nama : Muhamad Rafif Prasetyo
Pembimbing : Ir. Rusdhianto Effendie AK, M.T.

ABSTRAK

Peluru kendali (rudal) perlu dikendalikan dan dibuat untuk selalu mengikuti komando yang diberikan pemandu agar rudal dapat bergerak menuju sasarannya. Pengendalian rudal menjadi lebih rumit dikarenakan rudal memiliki persamaan dinamika yang *non-linear* dan *coupled*. *Linear Quadratic Tracking* (LQT) merupakan salah satu metode kendali optimal dengan tujuan membuat keluaran sistem mengikuti referensi yang diberikan sedekat mungkin dengan memperhatikan suatu indeks performansi. Sistem yang ingin dikendalikan oleh LQT harus sistem yang linier, namun rudal memiliki karakteristik yang *non-linear*, sehingga diperlukan linearisasi agar rudal dapat dikendalikan dengan LQT. Dalam tugas akhir ini, dirancang sebuah *autopilot* rudal yang terdiri dari *non-linear state feedback decoupler* dan pengendali LQT. Pemandu yang digunakan adalah Pemandu Pengejar (*Pursuit Guidance*). Simulasi pengejaran rudal menuju sasaran dirancang dengan menggunakan dua jenis sasaran, yaitu sasaran statis dan sasaran dinamis. Hasil simulasi menunjukkan dengan sasaran statis, rata-rata jarak terdekat antara rudal – sasaran sebesar 0.45 meter. Sedangkan pada sasaran dinamis, rata-rata jarak terdekat antara rudal – sasaran sebesar 2.562 meter.

Kata Kunci : Peluru Kendali, *Non-Linear State Feedback Decoupler*, *Linear Quadratic Tracking* (LQT), *Pursuit Guidance*

Halaman ini sengaja dikosongkan

DESIGNING AN OPTIMAL CONTROL LQT FOR CONTROLLING AND GUIDANCE OF MISSILE

Name : Muhamad Rafif Prasetyo
Supervisor : Ir. Rusdhianto Effendie AK, M.T.

ABSTRACT

Missile needs to be controlled and made to always follow the commanded guidance in order to make it engage the target. Controlling the missile has become more complex because missile has a non-linear and coupled dynamic equation. Linear Quadratic Tracking (LQT) is one method of optimal control theory where its objective is to make the output of a system tracks its reference as close as possible while minimize or maximize a performance index. The system which wants to be controlled with LQT must be a linear system, but the missile itself has a non-linear system, so the missile needs to be linearized in order to control it using LQT. In this final project, a missile's autopilot is designed which consists of non-linear state feedback decoupler and LQT controller. Pursuit Guidance is used for the guidance law. A missile – target engagement simulation is made and done using 2 kinds of targets; static target and dynamic target. From the simulation result's by using static target, the mean of the closest distance between the missile and the target is 0.45 meters and from the simulation result's by using dynamic target, the mean of the closest distance between the missile and the target is 2.562 meters.

Keywords : *Guided Missile, Non-linear State Feedback Decoupler, Linear Quadratic Tracking (LQT), Pursuit Guidance*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah Robbil 'Alamin, segala puji bagi Allah SWT atas limpahan rahmat dan karunianya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul :

PERANCANGAN KENDALI OPTIMAL LQT UNTUK PENGENDALIAN DAN PEMANDUAN PADA RUDAL

Adapun tujuan dari penyusunan tugas akhir ini adalah sebagai salah satu persyaratan untuk menyelesaikan studi tahap sarjana pada Bidang Studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro. Institut Teknologi Sepuluh Nopember.

Pada kesempatan ini, penulis mengucapkan banyak terima kasih kepada pihak – pihak yang telah banyak berjasa terutama yang telah membantu terselesaikannya tugas akhir ini, antara lain:

1. Segeanap keluarga tercinta yang berada di rumah, Ayah dan Ibu, yang selalu tidak kenal lelah memberi do'a, semangat, dan dukungan yang tiada henti untuk keberhasilan penulis.
2. Bapak Ir. Rusdhianto Effendie AK, M.T. selaku Dosen pembimbing yang telah banyak memberikan saran dan bantuan sehingga tugas akhir ini dapat terselesaikan dengan baik.
3. Bapak Zulkifli Hidayat S.T., M.Sc. dan mas Mochamad Nur Qomarudin yang telah memberikan banyak masukan dan saran selama penyelesaian tugas akhir ini.
4. Seluruh rekan Nindya, Irfan, Anang, Yasya, Hendri, Maliki, dan teman – teman bidang studi Teknik Sistem Pengaturan angkatan 2013 lainnya yang telah menemani penulis selama masa perkuliahan dan penyelesaian tugas akhir di perguruan tinggi tercinta ini.
5. Seluruh rekan asisten Laboratorium Sistem dan Sibernatika AJ204 dan B405 yang telah memberikan bantuan dan dukungan selama penulis menyelesaikan tugas akhir ini.
6. Seluruh keluarga besar Departemen Teknik Elektro, para dosen, dan karyawan atas dukungan dan kerjasamanya selama masa perkuliahan dan penyelesaian tugas akhir ini.

Demikian yang dapat penulis sampaikan. Penulis menyadari masih banyaknya kekurangan dalam laporan tugas akhir ini. Oleh karena itu, penulis sangat terbuka dengan kritik dan saran dari pembaca laporan tugas akhir ini. Terakhir, semoga laporan tugas akhir ini dapat bermanfaat untuk banyak pihak.

Surabaya, Januari 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	v
HALAMAN PENGESAHAN	vii
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 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	4
1.7 Relevansi.....	4
BAB 2 TINJAUAN PUSTAKA	5
2.1 Sistem Koordinat.....	5
2.1.1 Sifat Transformasi pada Vektor	6
2.1.2 Sudut Euler.....	7
2.2 Peluru Kendali (Rudal)	9
2.3 Model Matematis Rudal	12
2.3.1 Kinematika Rudal.....	12
2.3.2 Dinamika Rudal	14
2.4 <i>Autopilot</i> Rudal	19
2.4.1 <i>Non-Linear State Feedback Decoupler</i>	20
2.4.2 Kendali Optimal	21
2.5 Pemandu Rudal	24
2.5.1 Pemandu Pengejar (<i>Pursuit Guidance</i>)	28
2.6 Prediksi Posisi Sasaran.....	29
BAB 3 PERANCANGAN SISTEM	35
3.1 Gambaran Umum Sistem	35
3.2 Perancangan Model Rudal	35

3.3 Perancangan <i>Non-Linear State Feedback Decoupler</i>	40
3.4 Perancangan Linear Quadratic Tracking (LQT)	42
3.5 Perancangan Pemandu Rudal.....	45
3.6 Perancangan Pergerakan Sasaran.....	47
3.7 Perancangan Prediksi Posisi Sasaran	48
BAB 4 HASIL DAN ANALISA	53
4.1 Gambaran Umum Pengujian Simulasi Rudal	53
4.2 Pengujian Model Dinamika Rudal.....	54
4.3 Pengujian <i>Non-linear State Feedback Decoupler</i>	56
4.4 Pengujian Pengendali LQT.....	58
4.5 Pengujian Pemandu Rudal	60
4.6 Pengujian Prediksi Posisi Sasaran	63
4.7 Simulasi Pengejaran Rudal – Sasaran.....	65
4.7.1 Sasaran Statis	65
4.7.2 Sasaran Dinamis	68
BAB 5 PENUTUP	73
5.1 Kesimpulan	73
5.2 Saran	74
DAFTAR PUSTAKA.....	75
LAMPIRAN A	77
A1. Program Dinamika Rudal	77
A2. Program Kinematika Rudal.....	79
A3. Program <i>Non-linear State Feedback Decoupler</i>	79
A4. Program Pemandu Rudal	81
A5. Program Prediksi Posisi Sasaran.....	82
A6. Program Menentukan Jarak Relatif.....	82
RIWAYAT HIDUP PENULIS	85

DAFTAR GAMBAR

Gambar 2.1	Sistem koordinat bumi dan sistem koordinat rudal 5
Gambar 2.2	Orientasi sumbu rudal terhadap sumbu bumi..... 7
Gambar 2.3	Pemutaran terhadap sebuah sumbu (a) terhadap sumbu-z, (b) terhadap sumbu-y, dan (c) terhadap sumbu-x 9
Gambar 2.4	Diagram alur mekanisme kerja rudal 11
Gambar 2.5	Rudal dengan 6 derajat kebebasan 13
Gambar 2.6	Rudal dengan tempat sirip (kiri) hidung rudal, (tengah) badan rudal, dan (kanan) ekor rudal 19
Gambar 2.7	Susunan sirip pada rudal (kiri) susunan <i>plus</i> , dan (kanan) susunan <i>cross</i> 20
Gambar 2.8	Diagram blok <i>Non-Linear State Feedback Decoupler</i> . 21
Gambar 2.9	Diagram Blok sistem dengan LQT 24
Gambar 2.10	<i>Command to Line-of-Sight Guidance</i> 25
Gambar 2.11	<i>Beam Riding Guidance</i> 26
Gambar 2.12	<i>Active Homing</i> 27
Gambar 2.13	<i>Semi-active Homing</i> 27
Gambar 2.14	<i>Passive Homing</i> 27
Gambar 2.15	Sudut relatif rudal – sasaran..... 29
Gambar 2.16	Posisi sasaran pada waktu sample tertentu 31
Gambar 2.17	Posisi sasaran saat waktu t_{go} 32
Gambar 3.1	Diagram alir perancangan simulasi rudal..... 36
Gambar 3.2	Blok Simulink Model Dinamika Rudal 37
Gambar 3.3	Model <i>Thurst</i> yang digunakan pada masukan model rudal 39
Gambar 3.4	Blok Simulink model dinamika dan kinematika rudal. 40
Gambar 3.5	Blok Simulink linearisasi dengan <i>decoupler</i> 42
Gambar 3.6	Blok Simulink pengendali LQT 45
Gambar 3.7	Blok Simulink pengendali LQT yang dihubungkan dengan <i>decoupler</i> 45
Gambar 3.8	Batas keluaran fungsi $atan2(a,b)$ pada MATLAB 46
Gambar 3.9	Blok Simulink pemandu rudal 47
Gambar 3.10	Blok Simulink pergerakan sasaran..... 48
Gambar 3.11	Diagram alir prediksi posisi sasaran 50
Gambar 3.12	Blok Simulink untuk memprediksi posisi sasaran 51
Gambar 4.1	Posisi rudal terhadap bumi tanpa <i>Thrust</i> 54
Gambar 4.2	Pengujian model rudal dengan <i>Thurst</i> 55

Gambar 4.3	Respon kecepatan sudut rudal dengan masukan [0.5 0 0]	56
Gambar 4.4	Respon kecepatan sudut rudal dengan masukan [0 0.5 0]	57
Gambar 4.5	Respon kecepatan sudut rudal dengan masukan [0 0 0.5]	57
Gambar 4.6	Pengujian pengendalian sudut pitch rudal dengan masukan sinusoidal	59
Gambar 4.7	Pengujian pengendalian sudut yaw dengan masukan sinusoidal	60
Gambar 4.8	Jarak relatif rudal - sasaran saat posisi sasaran [1000 0 0]	61
Gambar 4.9	Jarak relatif rudal - sasaran saat posisi sasaran [1000 0 1000]	61
Gambar 4.10	Jarak relatif rudal – sasaran saat posisi sasaran [1000 1000 1000]	62
Gambar 4.11	Jarak relatif rudal – sasaran saat posisi sasaran [1500 1400 1300]	62
Gambar 4.12	Pengujian prediksi posisi sasaran dengan sasaran statis	64
Gambar 4.13	Pengujian prediksi posisi sasaran dengan sasaran dinamis	64
Gambar 4.14	Hasil simulasi dengan sasaran statis pada skenario 1, 2, dan 3	66
Gambar 4.15	Hasil simulasi dengan sasaran statis pada skenario 4,5, dan 6	67
Gambar 4.16	Hasil simulasi dengan sasaran statis pada skenario 7,8, dan 9	67
Gambar 4.17	Hasil simulasi sasaran dinamis pada skenario 1 dan 2	69
Gambar 4.18	Hasil simulasi sasaran dinamis pada skenario 3 dan 4	70
Gambar 4.19	Hasil simulasi sasaran dinamis pada skenario 5 dan 6	70

DAFTAR TABEL

Tabel 3.1	Parameter – parameter spesifikasi rudal	38
Tabel 4.1	Karakteristik respon kecepatan sudut rudal	58
Tabel 4.2	Hasil pengujian pemandu rudal	63
Tabel 4.3	Posisi awal sasaran pada simulasi sasaran statis.....	66
Tabel 4.4	Hasil simulasi dengan sasaran statis	68
Tabel 4.5	Kondisi awal sasaran dinamis	69
Tabel 4.6	Hasil simulasi dengan sasaran dinamis	71

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Bab ini menjelaskan latar belakang dari penulisan tugas akhir ini, perumusan masalah yang ditentukan, batasan – batasan masalah yang ada, tujuan penelitian tugas akhir, metodologi pelaksanaan tugas akhir, sistematika penulisan laporan tugas akhir, dan relevansi.

1.1 Latar Belakang

Saat ini sistem pertahanan udara sedang mengalami perkembangan. Apabila terdapat objek tidak teridentifikasi di udara tentu dapat membuat masyarakat menjadi resah. Hal yang paling buruk dapat terjadi yaitu objek tidak teridentifikasi tersebut adalah sebuah pesawat tempur bukan milik pertahanan negara yang membawa bahan peledak dan sedang dalam perjalanan untuk meledakkan suatu tempat ataupun daerah. Salah satu cara terakhir agar dapat mencegah hal yang tidak diinginkan ini yaitu dengan menembak jatuh objek yang tidak teridentifikasi tersebut.

Peluru Kendali (rudal) merupakan sebuah senjata jenis roket, dalam dunia militer, yang dapat dikendalikan secara langsung atau dikendalikan secara otomatis dalam mencari sasarannya dan pergi menuju sasarannya tersebut. Agar rudal dapat bergerak menuju sasaran, rudal perlu dikendalikan dengan baik. Rudal memiliki karakteristik yang *non-linear* dan persamaan dinamika yang *coupled*, sehingga pengendalian rudal terlihat cukup rumit. Umumnya variabel yang dikendalikan pada rudal adalah percepatan lateral rudal ataupun sudut orientasi rudal. Dengan memanfaatkan gaya aerodinamika, posisi sudut sirip rudal akan menentukan kemana rudal akan bergerak.

Kendali optimal merupakan salah satu teori kendali modern. Tujuan dari kendali optimal adalah menentukan sinyal kendali yang tepat agar keluaran sistem dapat menjaga atau mengikuti referensi yang diberikan dan disaat yang sama, meminimalkan atau memaksimalkan suatu indeks performansi yang diberikan. Dilihat dari tujuannya, kendali optimal terbagi menjadi 2 kasus, yaitu kasus *regulator* (dikenal sebagai *Linear Quadratic Regulator*) dan kasus *tracking* (dikenal sebagai *Linear Quadratic Tracking*). Pada kasus *tracking*, keluaran sistem dijaga agar mengikuti referensi yang diberikan. Pengendali *Linear Quadratic* ini memerlukan sistem yang linier dan indeks performansi yang diinginkan dibuat dalam bentuk kuadrat.

1.2 Permasalahan

Pemandu memberikan referensi orientasi yang diinginkan ke rudal agar rudal dapat bergerak tepat menuju sasaran. Seiring Bergeraknya rudal di udara, nilai referensi ini akan selalu berubah. Selain itu, agar rudal dapat mengenai sasaran dengan baik, kesalahan orientasi rudal dengan referensi yang diberikan harus sekecil mungkin. Oleh karena itu, diperlukan suatu pengendali yang dapat orientasi rudal mengikuti nilai referensi dengan nilai kesalahan sekecil mungkin. Hal ini dapat diselesaikan dengan menggunakan pengendali LQT. Sistem yang ingin dikendalikan menggunakan LQT harus linier, tetapi persamaan dinamika rotasi rudal memiliki karakteristik yang *non-linear* dan *coupled*, sehingga pengendali LQT tidak dapat langsung digunakan dengan rudal.

1.3 Batasan Masalah

Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut.

- Penelitian ini menggunakan rudal taktis SAM jenis *Skid-to-Turn*.
- Rudal diasumsikan sebagai benda tegar.
- Rudal tidak mengalami efek aeroelastis.
- Nilai percepatan gravitasi bumi dan kepadatan udara dianggap konstan.
- Efek aerodinamis pada sumbu *roll* dianggap simetri.
- Massa rudal tidak berubah terhadap waktu.
- Rudal memiliki 6 *degrees of freedom* (6-DOF).
- Informasi yang ada bersifat deterministik.

1.4 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah membuat suatu pengendali agar rudal dapat mengikuti referensi orientasi yang diberikan dari pemandu dengan kesalahan antara orientasi rudal dengan referensi sekecil mungkin. Oleh sebab itu, digunakan pengendali LQT agar rudal dapat dikendalikan dan dipandu untuk bergerak menuju sarannya.

Sistem yang ingin dikendalikan oleh LQT harus dalam bentuk linier, sedangkan rudal sendiri memiliki karakteristik yang *non-linear* dan *coupled*, sehingga diperlukan sebuah *non-linear state feedback decoupler* agar sistem rudal bersifat linier dan *decoupled*. Dengan demikian, rudal dapat dikendalikan oleh pengendali LQT.

1.5 Metodologi

Tahapan penyelesaian tugas akhir ini adalah sebagai berikut.

- a) Studi Literatur
Melakukan studi literatur dalam mencari persamaan matematis dari rudal. Sumber literatur berasal dari buku, jurnal ilmiah, dan. Lalu dilakukan studi literatur mengenai perancangan pengendali LQT. Sumber literatur berasal dari buku acuan dan catatan kuliah dari mata kuliah Sistem Pengaturan Optimal.
- b) Pemodelan Matematis dari Rudal
Membuat model persamaan matematis *non-linear* dari rudal yang telah didapat dari sumber literatur. Selanjutnya model rudal ini yang akan dijadikan objek kajian dalam tugas akhir ini.
- c) Linearisasi Persamaan Matematis pada Rudal
Linearisasi dilakukan dengan membuat *decoupler* pada rudal guna menghilangkan sifat *non-linear* pada rudal, sehingga rudal dapat dikendalikan dengan pengendali LQT.
- a) Perancangan *Linear Quadratic Tracking*
Melakukan perancangan pengendali *Linear Quadratic Tracking* sebagai pengendali posisi sirip dari rudal. Pengujiannya dilakukan dengan diberikan masukan referensi sinusoidal. Pengendalian dikatakan berhasil apabila keluaran rudal yang telah dilinearisasi dapat mengikuti masukan yang diberikan.
- b) Perancangan Pemandu dan Prediksi Posisi Sasaran
Pemandu digunakan sebagai pemberi referensi ke pengendali LQT pada rudal. Lalu Prediksi posisi sasaran diperlukan agar rudal dapat bergerak ke posisi di mana akan terjadinya *impact*.
- c) Hasil dan Analisa
Setelah semua tahapan sebelumnya selesai, dilakukan simulasi rudal mengejar sasaran. Simulasi dilakukan dengan menggunakan sasaran statis dan sasaran dinamis.
- d) Kesimpulan dan Saran
Penarikan kesimpulan dilakukan setelah melakukan analisa dari simulasi yang telah dijalankan sebelumnya. Selanjutnya pemberian saran berupa seperti apa penelitian yang dapat dilanjutkan dari tugas akhir ini.
- e) Penyusunan Buku Tugas Akhir
Penulisan buku laporan tugas akhir dilakukan sebagai bentuk laporan tertulis dari pelaksanaan tugas akhir ini.

1.6 Sistematika Penulisan

Laporan tugas akhir ini dibagi menjadi lima bab dengan uraian singkat masing – masing bab diberikan sebagai berikut.

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang dari pelaksanaan tugas akhir, permasalahan yang ada, tujuan pelaksanaan tugas akhir, metodologi pelaksanaan, sistematika penulisan laporan tugas akhir, dan relevansi tugas akhir ini.

BAB II TINJAUAN PUSTAKA

Dalam bab ini dijelaskan teori – teori penunjang yang digunakan dalam penyelesaian tugas akhir ini meliputi transformasi koordinat, model matematis rudal, pemandu rudal, linearisasi menggunakan *decoupler*, *Linear Quadratic Tracking*, dan prediksi posisi sasaran.

BAB III PERANCANGAN SISTEM

Dalam bab ini dijelaskan langkah – langkah pembuatan simulasi rudal, mulai dari pembuatan model matematis rudal, perancangan *decoupler* pada rudal, perancangan *Linear Quadratic Tracking*, perancangan pemandu rudal, perancangan prediksi posisi sasaran.

BAB IV HASIL DAN ANALISA

Bab ini berisi pengujian dari simulasi perancangan simulasi rudal yang telah dibuat. Analisa dilakukan dengan membuat parameter awal sasaran yang berbeda – beda.

BAB V PENUTUP

Bab ini berisi kesimpulan yang diambil dari hasil simulasi beserta analisa simulasi, serta saran untuk pelaksanaan penelitian selanjutnya dari tugas akhir ini.

1.7 Relevansi

Hasil dari pelaksanaan tugas akhir ini diharapkan dapat memberi manfaat sebagai salah satu referensi yang dapat digunakan dalam penelitian sistem pertahanan udara khususnya dalam sistem pertahanan udara Indonesia. Selain itu dapat membuka topik kajian baru dengan menggunakan pemandu atau pengendali yang berbeda.

BAB 2

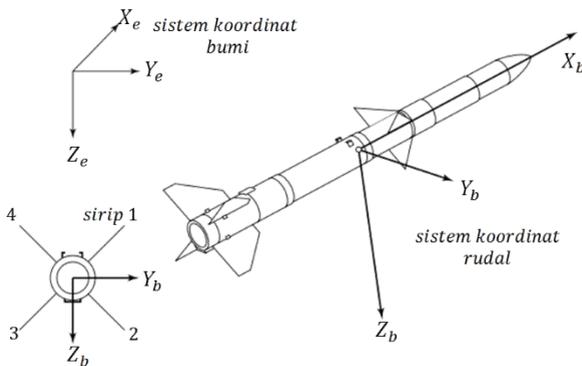
TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai dasar teori yang digunakan dalam pelaksanaan tugas akhir ini, yakni mengenai model matematis dari peluru kendali yang digunakan, *autopilot* dari rudal, metode pemandu (*guidance*) yang digunakan, dan prediksi posisi sasaran di saat mendarat.

2.1 Sistem Koordinat[1]

Sistem koordinat merupakan suatu sistem yang terdiri dari beberapa bilangan untuk menunjukkan posisi suatu titik atau benda dalam ruang berdimensi-N. Dalam dunia nyata, sistem koordinat yang digunakan adalah 3 dimensi. Jika suatu sistem koordinat memiliki titik referensi atau lokasi awal yang berbeda dengan sistem koordinat lainnya, maka nilai yang menunjukkan posisi suatu titik atau benda dalam sistem koordinat tersebut akan berbeda satu sistem koordinat dengan sistem koordinat lainnya. Oleh karena itu, diperlukan suatu transformasi agar sistem koordinat dengan referensi tertentu dapat dipetakan ke sistem koordinat dengan referensi yang berbeda.

Sistem koordinat yang diadaptasikan dalam tugas akhir ini adalah sistem tangan kanan, di mana sumbu- x positif ada di sepanjang sumbu longitudinal rudal, sumbu- y positif ke kanan rudal, dan sumbu- z positif ke arah bawah (sumbu- z didefinisikan sebagai perkalian *cross* antara sumbu- x dan sumbu- y). Sistem ini juga dikenal sebagai *north-east-down* (NED).



Gambar 2.1 Sistem koordinat bumi dan sistem koordinat rudal [1]

Sistem koordinat yang digunakan ada 2, yaitu sistem koordinat bumi dan sistem koordinat rudal. Untuk sistem koordinat terhadap bumi dinotasikan (X_e, Y_e, Z_e) . Sumbu $X_e - Y_e$ merupakan bidang horizontal mengikuti permukaan laut, sedangkan sumbu Z_e mengarah ke pusat bumi mengikuti arah gravitasi. Lalu sistem koordinat rudal dinotasikan sebagai (X_b, Y_b, Z_b) . Sistem koordinat berpusat di titik pusat massa rudal dan tidak berubah terhadap rudal, sehingga sistem koordinat ini ikut bergerak mengikuti rudal.

2.1.1 Sifat Transformasi pada Vektor

Dalam sistem koordinat, sebuah vektor dapat ditunjukkan dari komponen-komponen penyusunnya. Komponen-komponen ini tentunya tergantung dari orientasi pada sistem koordinat yang digunakan, serta vektor yang sama dapat dengan komponen lainnya yang tersusun dari sistem koordinat yang lain. Dalam ruang 3 dimensi, 3 komponen yang menunjukkan suatu vektor pada suatu sistem koordinat akan berhubungan dengan komponen yang ada pada sistem koordinat lainnya. Sehingga dapat dikatakan komponen skalar dari suatu vektor berubah mengikuti perubahan koordinatnya.

Sebuah vektor dapat didefinisikan berupa suatu himpunan terdiri dari 3 angka yang berubah mengikuti koordinat dari suatu titik saat sumbunya diputar. Untuk menunjukkan sumbu koordinat umumnya menggunakan angka, sehingga komponen suatu vektor menjadi $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$. Penulisan sebuah vektor secara keseluruhan adalah \mathbf{a}_i di mana i merupakan nilai 1,2 atau 3.

Perubahan dari satu sistem koordinat ke suatu sistem koordinat lainnya terhadap referensinya dapat direpresentasikan oleh sembilan nilai C_{ij} , di mana C_{ij} , merupakan kosinus sudut antara sumbu- i pada satu koordinat dengan sumbu- j pada koordinat lainnya. sembilan nilai ini dapat menunjukkan sudut yang dibentuk pada setiap sumbu dari suatu sistem koordinat dengan sumbu pada sistem koordinat lainnya. Nilai ini juga merupakan koefisien dalam persamaan transformasi koordinat dari suatu titik. Nilai kosinus dapat ditulis dalam bentuk matriks yang dapat dilihat 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 dikatakan sebagai komponen dalam sumbu- j dari satu sistem koordinat dari unit vektor dalam sumbu- i dari sistem koordinat lainnya, maka berlaku hubungan seperti pada Persamaan (2.2).

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

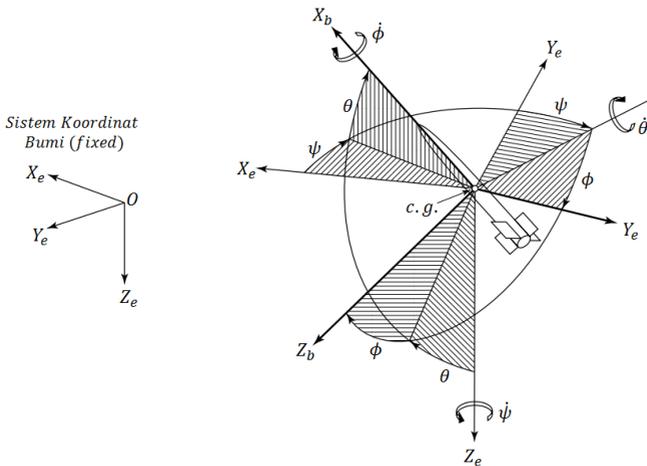
Komponen dari suatu vektor, atau titik pada suatu sistem koordinat, dapat ditransformasikan ke sistem koordinat lainnya dengan menggunakan Persamaan (2.3).

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

Di mana \mathbf{a}_j menunjukkan komponen vektor \mathbf{a} dalam satu sistem koordinat, dan \mathbf{b}_i adalah komponen dari sistem koordinat lainnya.

2.1.2 Sudut Euler[2]

Sudut Euler merupakan salah satu metode transformasi sistem koordinat yang umum digunakan dalam penerbangan maupun simulasi dari pesawat udara atau peluru kendali. Sudut Euler digunakan untuk menunjukkan hubungan sistem koordinat yang bergerak dengan sistem koordinat yang tetap. Notasi sudut yang digunakan dalam metode ini adalah ϕ untuk gerakan *roll*, θ untuk gerakan *pitch*, dan ψ untuk gerakan *yaw*. Sudut – sudut ini dapat mendefinisikan posisi dari sistem koordinat rudal (X_b, Y_b, Z_b) ke sistem koordinat bumi (X_e, Y_e, Z_e), maupun sebaliknya.



Gambar 2.2 Orientasi sumbu rudal terhadap sumbu bumi [1]

Metode ini dapat memutar sistem koordinat rudal ke sistem koordinat bumi dengan menggunakan informasi dari Sudut Euler yang ada. Transformasi dilakukan dengan memutar setiap sumbu dari satu sistem koordinat ke sistem koordinat lainnya secara berurutan. Urutan pemutaran sumbu harus selalu konsisten (misal pertama diputar terhadap sumbu-z (*yaw*), lalu diputar terhadap sumbu-y (*pitch*), terakhir diputar terhadap sumbu-x (*roll*)) karena matriks transformasi suatu urutan pemutaran dengan urutan pemutaran lainnya berbeda. Dalam tugas akhir ini digunakan pemutaran terhadap sumbu-z, lalu sumbu-y, lalu sumbu-x.

Langkah pertama transformasi koordinat dari sistem koordinat bumi ke sistem koordinat rudal dilakukan dengan memutar sistem koordinat rudal terhadap sumbu-z (*yaw*) menggunakan Persamaan (2.4).

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

Koordinat baru ini selanjutnya diputar terhadap sumbu-y (*pitch*) dengan menggunakan Persamaan (2.5).

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

Terakhir koordinat ini diputar terhadap sumbu-x (*roll*) dengan menggunakan Persamaan (2.6).

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

Dari ketiga proses pemutaran sumbu ini, didapatkan matriks rotasi sistem koordinat seperti pada Persamaan (2.7).

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = [D][B][A] \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} = [C] \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} \quad (2.7)$$

Di mana:

X_b, Y_b, Z_b = Sistem koordinat rudal

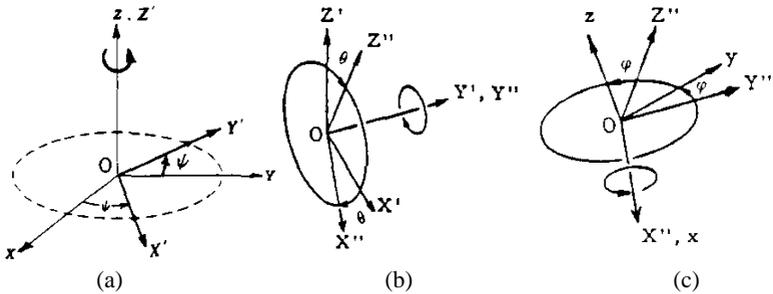
X_e, Y_e, Z_e = Sistem koordinat bumi

$[C]$ = Matriks rotasi

Dengan nilai dari matriks C dapat dilihat pada Persamaan (2.8).

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

Di mana c adalah *cosinus* dan s adalah *sinus*.



Gambar 2.3 Pemutaran terhadap sebuah sumbu (a) terhadap sumbu-z, (b) terhadap sumbu-y, dan (c) terhadap sumbu-x [2]

Visualisasi pemutaran setiap sumbu dapat dilihat pada Gambar 2.3. Kelebihan dari metode ini yaitu mudah diinterpretasikan sehingga mudah digunakan pada simulasi. Ini dikarenakan Sudut Euler merupakan sudut yang dibentuk oleh sumbu – sumbu pada rudal terhadap sumbu – sumbu pada bumi. Kekurangan dari metode ini yaitu terdapat singularitas apabila sudut *pitch* mencapai 90° . Karena apabila terjadi, maka arah sumbu putar *yaw* akan sama dengan arah sumbu putar *roll*.

2.2 Peluru Kendali (Rudal)[3]

Peluru kendali (rudal) merupakan sebuah senjata militer jenis roket yang dapat dikendalikan atau tidak dalam mencari sasaran dan bergerak menuju sasaran tersebut. Perkembangan rudal ini dimulai pada awal abad 20, yakni saat Perang Dunia I, lalu penggunaannya dilakukan pertama kali oleh Jerman pada Perang Dunia II.

Rudal dapat dikategorikan ke dalam beberapa kategori. Dilihat dari jarak jelajahnya, rudal dikategorikan ke dalam 2 grup, yaitu Rudal Taktis (*Tactical Missiles*) dan Rudal Balistik (*Ballistic Missiles*). Rudal Taktis digunakan pada jarak jelajah dekat dan umumnya dipandu dengan sensor seperti *seeker*. Rudal balistik berbeda dengan Rudal Taktis karena rudal

jenis ini dapat menjelajah lebih jauh dan memang dirancang untuk menyerang sasaran yang tidak bergerak dan lokasinya diketahui dengan tepat. Dalam tugas akhir ini rudal yang dikaji adalah Rudal Taktis.

Dilihat dari tempat diluncurkan dan sasarannya, rudal dibedakan menjadi 4 jenis.

- *Surface-to-Surface Missiles (SSM)*
- *Surface-to-Air Missiles (SAM)*
- *Air-to-Air Missiles (AAM)*
- *Air-to-Surface Missiles (ASM)*

Rudal SSM diluncurkan dari permukaan atau daratan, bergerak menuju sasaran yang terletak di permukaan juga, seperti kendaraan atau tank. Jika sasaran berada di udara, maka rudal tersebut termasuk jenis SAM. Jika rudal diluncurkan di udara, misal dari pesawat tempur, dan sasarannya berada di permukaan atau daratan, maka rudal tersebut termasuk jenis ASM. Jika sasaran berada di udara, rudal tersebut termasuk jenis AAM.

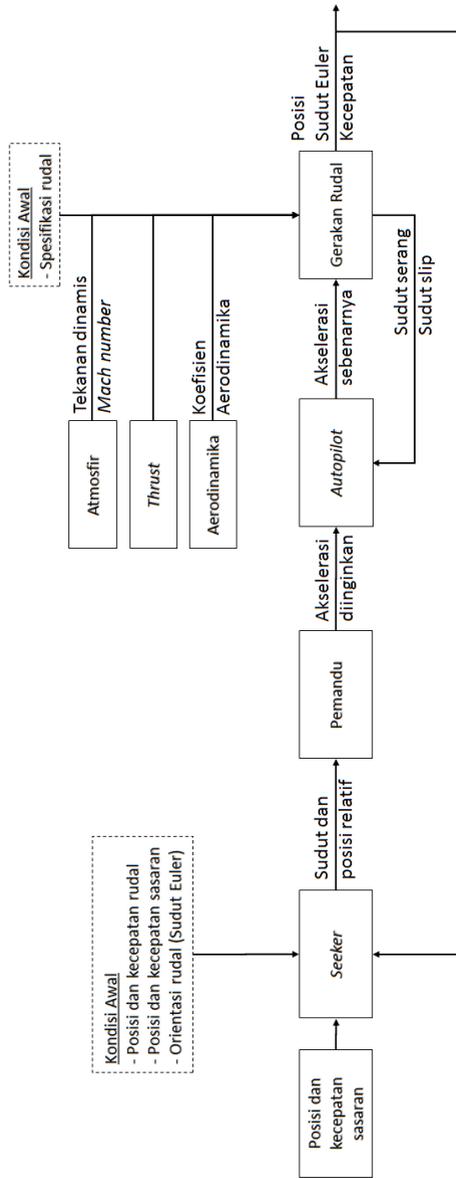
Dari cara rudal melakukan *maneuver*, rudal dibagi menjadi 2 jenis, yaitu rudal jenis *Skid-to-Turn* dan *Bank-to-Turn*. Pada rudal jenis *Skid-to-Turn*, rudal dipertahankan untuk tidak melakukan gerakan rotasi *roll* selama rudal bergerak di udara. Sedangkan pada jenis *Bank-to-Turn*, rudal melakukan gerakan rotasi *roll* dahulu untuk melakukan gerakan rotasi *yaw*.

Dalam perancangan simulasi rudal, terdapat 4 tahap dasar yang perlu diperhatikan.

- Model matematis dinamika rudal
- Pemandu rudal
- *Autopilot* rudal
- Prediksi posisi sasaran

Pada tahap pemodelan dinamis, rudal dimodelkan sedemikian agar didapatkan hubungan masukan dan keluaran yang digunakan. Selanjutnya algoritma pemandu dirancang agar dapat memandu rudal bergerak ke lokasi pengecatan sasaran. *Autopilot* dirancang berdasarkan model dinamis yang digunakan agar rudal dapat bergerak sesuai sinyal yang dibuat oleh pemandu. Lalu tahap terakhir dirancang sebuah sistem untuk memprediksi trayektori sasaran. Sistem ini umumnya menggunakan algoritma *state estimator* seperti *Kalman Filter*.

Perancangan rudal harus selalu dimulai dari pemodelan dinamis dari rudal yang akan digunakan. Selanjutnya dapat merancang pemandunya dahulu lalu *autopilot*, atau *autopilot* dahulu lalu pemandu. Tidak terikat



Gambar 2.4 Diagram alur mekanisme kerja rudal [1]

apakah harus pemandu atau pengendali dahulu, karena kedua tahap ini berdasarkan model dinamis yang telah diturunkan. Yang perlu diperhatikan adalah sinyal keluaran pemandu harus sesuai dengan sinyal masukan pengendali yang digunakan. Lalu komponen untuk memprediksi gerakan sasaran dapat dirancang agar posisi dan kecepatan sasaran dapat diperkirakan. Diagram aliran informasi untuk simulasi rudal dapat dilihat pada Gambar 2.4.

2.3 Model Matematis Rudal[1]

Persamaan matematis dari rudal dapat diturunkan menggunakan Hukum Newton. Ada beberapa asumsi yang digunakan dalam model rudal yang digunakan. Pertama, rudal merupakan benda tegar. Rudal dianggap sebagai partikel di mana massa rudal berpusat di titik pusat massa dari rudal sendiri. Rudal tidak mengalami efek aeroelastis. Sehingga persamaan gerakan rudal dijelaskan sebagai gerakan translasi dan rotasi terhadap titik pusat gravitasi. Kedua, efek aerodinamis pada sumbu *roll* dianggap simetri. Gaya dan momen yang bekerja untuk sumbu *roll* dianggap invarian terhadap posisi *roll* dari rudal. Ketiga, massa dari rudal dianggap tidak berubah terhadap waktu.

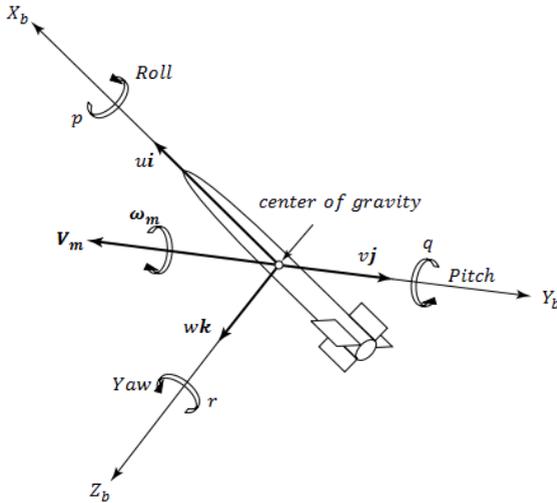
Beberapa asumsi lain yang sering digunakan diantaranya adalah persamaan gerakan rudal ditulis terhadap sistem koordinat rudal (bukan terhadap sistem koordinat bumi). Besar percepatan gravitasi dianggap konstan, sehingga nilai ini tidak bervariasi terhadap ketinggian rudal dari permukaan bumi. Kepadatan udara dianggap konstan, karena pada dasarnya kepadatan udara berkurang seiring bertambahnya ketinggian rudal dari permukaan bumi. Permukaan bumi diasumsikan datar, karena rudal yang digunakan adalah Rudal Taktis.

Dalam tugas akhir ini, rudal diasumsikan memiliki enam derajat kebebasan. Enam derajat kebebasan ini yakni tiga pada gerakan translasi dan tiga lainnya pada gerakan rotasi setiap sumbu pada sistem koordinat rudal. Gerakan ini dapat dilihat pada Gambar 2.5, di mana kecepatan translasi direpresentasikan dalam (u, v, w) dan kecepatan rotasi direpresentasikan dalam (p, q, r) .

2.3.1 Kinematika Rudal

Kinematika merupakan salah satu cabang dari mekanika yang membahas gerakan suatu objek tanpa memperhatikan gaya ataupun massa pada objek tersebut. Kinematika erat kaitannya dengan posisi, kecepatan,

atau percepatan suatu objek terhadap suatu referensi, sehingga kinematika ini erat kaitannya transformasi koordinat.



Gambar 2.5 Rudal dengan 6 derajat kebebasan [1]

Terdapat 2 kinematika pada rudal, yaitu kinematika translasi rudal dan kinematika rotasi rudal. Kinematika translasi rudal menyatakan kecepatan translasi rudal dalam sistem koordinat rudal terhadap sistem koordinat bumi. Sedangkan kinematika rotasi rudal menyatakan kecepatan rotasi rudal dalam sistem koordinat rudal terhadap sistem koordinat bumi. Transformasi kecepatan translasi rudal dinyatakan menggunakan Persamaan (2.8) dalam Persamaan (2.9).

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

Di mana:

- $\dot{x}, \dot{y}, \dot{z}$ = Kecepatan translasi rudal dalam sistem koordinat bumi.
- u, v, w = Kecepatan translasi rudal dalam sistem koordinat rudal.
- c, s, t = Masing-masing singkatan dari *cosinus*, *sinus*, dan *tangent*.

2.3.2 Dinamika Rudal[1], [4]

Jika sebelumnya kinematika tidak memperhatikan gaya atau massa yang bekerja pada suatu objek, dinamika mengkaji gerakan suatu objek dengan memperhatikan gaya yang bekerja serta massa yang ada pada suatu objek. Seperti halnya kinematika, dinamika juga membahas posisi, kecepatan, dan percepatan objek, namun yang mempengaruhi hal-hal tersebut adalah gaya-gaya yang bekerja pada objek tersebut. Dalam kegiatan sehari-hari, persamaan dinamika suatu benda dapat diturunkan dari Hukum Newton.

Seperti halnya pada kinematika, dinamika pada rudal terbagi menjadi 2 jenis, yaitu dinamika translasi dan dinamika rotasi. Dinamika translasi rudal menyatakan gaya-gaya yang bekerja pada rudal sedemikian rupa rudal mengalami perubahan kecepatan translasi (percepatan translasi). Sedangkan dinamika rotasi menyatakan momen-momen yang bekerja pada rudal sedemikian rupa rudal mengalami perubahan kecepatan sudut (percepatan sudut). Persamaan percepatan ini didapatkan dari penurunan Hukum Newton kedua.

Kecepatan pada rudal terbagi menjadi 2, yaitu kecepatan translasi dan kecepatan rotasi. Kecepatan translasi dari rudal (\mathbf{V}_b) merupakan kecepatan rudal u, v, w pada sumbu sistem koordinat rudal (X_b, Y_b, Z_b). Sedangkan kecepatan rotasi rudal ($\boldsymbol{\omega}_b$) merupakan kecepatan sudut rudal p, q, r dalam sumbu sistem koordinat rudal. Kecepatan translasi rudal dan kecepatan sudut rudal secara matematis dapat ditulis dalam bentuk seperti pada Persamaan (2.10) dan Persamaan (2.11).

$$\mathbf{V}_b = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} \quad (2.10)$$

$$\boldsymbol{\omega}_b = p\mathbf{i} + q\mathbf{j} + r\mathbf{k} \quad (2.11)$$

Di mana u, v, w dan p, q, r masing-masing merupakan kecepatan translasi rudal dan kecepatan rotasi rudal, sedangkan $\mathbf{i}, \mathbf{j}, \mathbf{k}$ merupakan vektor satuan pada masing-masing sumbu sistem koordinat rudal X_b, Y_b, Z_b . Magnitudo dari kecepatan rudal ditunjukkan pada Persamaan (2.12).

$$V_m = |\mathbf{V}_b| = \sqrt{u^2 + v^2 + w^2} \quad (2.12)$$

2.3.2.1 Dinamika Translasi[4]

Persamaan percepatan translasi rudal terhadap sistem koordinat rudal dapat diturunkan dari Hukum Newton kedua. Karena rudal memiliki gerakan rotasi, maka persamaan percepatan rudal akan dipengaruhi

percepatan tangensial yang terjadi pada rudal. Dengan demikian, persamaan percepatan translasi rudal dalam sistem koordinat rudal dapat dituliskan seperti pada Persamaan (2.13).

$$\frac{d\mathbf{V}_b}{dt} = \frac{\mathbf{F}_b}{m} - (\boldsymbol{\omega}_b \times \mathbf{V}_b) \quad (2.13)$$

Di mana:

- \mathbf{V}_b = Kecepatan translasi rudal
- $\boldsymbol{\omega}_b$ = Kecepatan sudut rudal
- \mathbf{F}_b = Gaya – gaya yang bekerja pada rudal
- m = Massa rudal

Gaya – gaya yang bekerja pada rudal merupakan gaya – gaya yang bekerja pada masing- masing sumbu pada sistem koordinat rudal. Gaya – gaya ini dapat dituliskan dalam Persamaan (2.14).

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

Dari Persamaan (2.13), bentuk $(\boldsymbol{\omega}_m \times \mathbf{V}_m)$ dapat dijabarkan menjadi Persamaan (2.15).

$$\boldsymbol{\omega}_m \times \mathbf{V}_m = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ p & q & r \\ u & v & w \end{bmatrix} \quad (2.15)$$

$$\boldsymbol{\omega}_m \times \mathbf{V}_m = (qw - rv)\mathbf{i} + (ru - pw)\mathbf{j} + (pv - qu)\mathbf{k} \quad (2.16)$$

Dengan demikian, percepatan translasi rudal pada setiap sumbu dalam sistem koordinat rudal dapat dinyatakan seperti pada Persamaan (2.17a) hingga Persamaan (2.17c).

$$\frac{du}{dt} = \frac{F_x}{m} + rv - qw \quad (2.17a)$$

$$\frac{dv}{dt} = \frac{F_y}{m} + pw - ru \quad (2.17b)$$

$$\frac{dw}{dt} = \frac{F_z}{m} + qu - pv \quad (2.17c)$$

Gaya – gaya yang dialami oleh rudal sendiri terdiri dari gaya pendorong pada rudal (*Thrust*), dan gaya gravitasi, gaya aerodinamika. Gaya gravitasi harus proyeksikan ke setiap sumbu sistem koordinat rudal agar nilai gaya tersebut dapat terlihat secara eksplisit dalam setiap sumbu. Gaya aerodinamika umumnya sering dinyatakan dengan dengan sebuah koefisien gaya aerodinamika tidak berdimensi, tekanan dinamis udara,

dan luas permukaan. Dengan demikian, gaya yang dialami rudal pada setiap sumbu dalam sistem koordinat rudal dapat dilihat pada Persamaan (2.18a) hingga Persamaan (2.18c).

$$F_x = Thrust - mg \sin \theta + C_x QS \quad (2.18a)$$

$$F_y = mg \sin \phi \cos \theta + C_y QS \quad (2.18b)$$

$$F_z = mg \cos \phi \cos \theta + C_z QS \quad (2.18c)$$

Di mana:

$Thrust$ = Gaya dorong pada rudal

m = massa rudal

g = percepatan gravitasi

Q = Tekanan dinamis udara $\left(\frac{1}{2}\rho V_m^2\right)$

ρ = Massa jenis udara

S = Luas permukaan rudal

C_x = Koefisien gaya hambatan (*drag force*)

C_y = Koefisien gaya samping (*side force*)

C_z = Koefisien gaya angkat (*lift force*)

Koefisien gaya aerodinamika pada sumbu sistem koordinat rudal C_x, C_y, C_z merupakan fungsi dari sudut serang (α), sudut slip (β), *Mach number* (M), besar sudut defleksi *elevator* (δ_e), dan besar sudut defleksi *rudder* (δ_r). *Mach number* merupakan perbandingan antara kecepatan rudal dengan kecepatan suara (*Speed of Sound, SoS*). Dengan demikian, persamaan koefisien gaya aerodinamika dapat dituliskan seperti pada Persamaan (2.19a) hingga Persamaan (2.19c).

$$C_x = C_{x1}\alpha + C_{x2}M \quad (2.19a)$$

$$C_y = C_{y1}\beta + C_{y2}M + C_{y3}\delta_r \quad (2.19b)$$

$$C_z = C_{z1}\alpha + C_{z2}M + C_{z3}\delta_e \quad (2.19c)$$

Di mana:

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right)$$

$$\beta = \sin^{-1}\left(\frac{v}{V_m}\right)$$

$$M = \frac{V_m}{SoS}$$

2.3.2.2 Dinamika Rotasi[4]

Persamaan percepatan sudut rudal dinyatakan dalam Persamaan (2.20).

$$\frac{d\boldsymbol{\omega}_b}{dt} = \mathbf{I}^{-1}(\mathbf{M}_b - (\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b)) \quad (2.20)$$

Di mana:

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

\mathbf{I} = Momen Inersia

\mathbf{M}_b = Momen yang bekerja pada rudal

Gaya – gaya yang bekerja pada rudal merupakan gaya – gaya yang bekerja pada masing- masing sumbu pada sistem koordinat rudal yang dapat dinyatakan dengan Persamaan (2.21).

$$\mathbf{M}_b = L\mathbf{i} + M\mathbf{j} + N\mathbf{k} \quad (2.21)$$

Momen inersia pada suatu benda yang memiliki 3 sumbu pada sistem koordinatnya memiliki bentuk seperti pada Persamaan (2.22).

$$\mathbf{I} = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{yx} & I_y & I_{yz} \\ I_{zx} & I_{zy} & I_z \end{bmatrix} \quad (2.22)$$

Elemen diagonal pada matriks momen inersia dari Persamaan (2.22) disebut sebagai momen inersia aksial, sedangkan elemen selain bagian diagonal disebut sebagai produk inersia. Karena sumbu utama sistem koordinat rudal berimpit dengan sumbu sistem koordinat rudal, maka semua produk inersia pada rudal menjadi nol dan hanya menyisakan momen inersia aksial. Dengan demikian, momen inersia pada rudal dapat dilihat pada Persamaan (2.23).

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (2.23)$$

Momen yang bekerja pada rudal berasal dari gaya aerodinamika. Dengan demikian, momen yang bekerja pada rudal diberikan dengan Persamaan (2.24a) hingga Persamaan (2.24c).

$$L = QSdC_l \quad (2.24a)$$

$$M = QSdC_m \quad (2.24b)$$

$$N = QSdC_n \quad (2.24c)$$

Di mana:

Q = Tekanan dinamis udara

S = Luas permukaan rudal

d = diameter rudal

C_l = Koefisien momen aerodinamika pada sumbu X_b

C_m = Koefisien momen aerodinamika pada sumbu Y_b

C_n = Koefisien momen aerodinamika pada sumbu Z_b

Koefisien momen aerodinamika merupakan fungsi dari sudut serang (α), sudut slip (β), kecepatan sudut rudal (p, q, r), besar sudut defleksi *ailirion* (δ_a), besar sudut defleksi *elevator* (δ_e), dan besar sudut defleksi *rudder* (δ_r). Persamaan koefisien momen aerodinamika ini dinyatakan dengan Persamaan (2.25a) hingga Persamaan (2.25c).

$$C_l = C_{l1}\alpha + C_{l2}\beta + C_{l3}p + C_{l4}\delta_a \quad (2.25a)$$

$$C_m = C_{m1}\alpha + C_{m2}q + C_{m3}\delta_e \quad (2.25b)$$

$$C_n = C_{n1}\beta + C_{n2}r + C_{n3}\delta_r \quad (2.25c)$$

Dari Persamaan (2.20), bentuk $(\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b)$ dapat dijabarkan menjadi Persamaan (2.26).

$$(\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b) = \begin{bmatrix} p\mathbf{i} \\ q\mathbf{j} \\ r\mathbf{k} \end{bmatrix} \times \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} p\mathbf{i} \\ q\mathbf{j} \\ r\mathbf{k} \end{bmatrix} \quad (2.26)$$

$$(\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b) = \begin{bmatrix} p\mathbf{i} \\ q\mathbf{j} \\ r\mathbf{k} \end{bmatrix} \times \begin{bmatrix} I_x p\mathbf{i} \\ I_y q\mathbf{j} \\ I_z r\mathbf{k} \end{bmatrix} \quad (2.27)$$

$$(\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b) = (I_z qr - I_y qr)\mathbf{i} + (I_x pr - I_z pr)\mathbf{j} \dots \\ \dots + (I_x pq - I_y pq)\mathbf{k} \quad (2.28)$$

$$(\boldsymbol{\omega}_b \times \mathbf{I}\boldsymbol{\omega}_b) = (I_z - I_y)qr\mathbf{i} + (I_x - I_z)pr\mathbf{j} \dots \\ \dots + (I_y - I_x)pq\mathbf{k} \quad (2.29)$$

Percepatan sudut rudal pada setiap sumbu sistem koordinat rudal dapat ditulis menjadi Persamaan (2.30a) hingga Persamaan (2.30c).

$$\frac{dp}{dt} = \frac{QSd(C_{l1}\alpha + C_{l2}\beta + C_{l3}p + C_{l4}\delta_a) + (I_y - I_z)qr}{I_x} \quad (2.30a)$$

$$\frac{dq}{dt} = \frac{QSd(C_{m1}\alpha + C_{m2}q + C_{m3}\delta_e) + (I_z - I_x)pr}{I_y} \quad (2.30b)$$

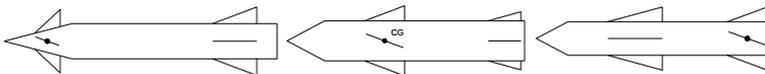
$$\frac{dr}{dt} = \frac{QSd(C_{n1}\beta + C_{n2}r + C_{n3}\delta_r) + (I_x - I_y)pq}{I_z} \quad (2.30c)$$

2.4 Autopilot Rudal[3]

Tugas dari sistem kendali rudal, atau biasa disebut sebagai *autopilot* rudal, adalah untuk memastikan kestabilan dan performa rudal saat terbang sesuai dengan perintah yang dikeluarkan oleh pemandu. Agar jarak meleset dapat berkurang, respon dari *autopilot* harus secepat mungkin. Yakni, *time constant* dari respon *autopilot* rudal bernilai kecil.

Pada umumnya, variabel yang dikendalikan pada rudal adalah percepatan lateral dari rudal. Dilihat dari jenis – jenis pemandu yang ada, selain percepatan lateral, variabel lainnya yang dapat dikendalikan yaitu kecepatan sudut rudal (p, q, r) atau sudut orientasi rudal (ϕ, θ, ψ). Dalam tugas akhir ini, variabel yang dikendalikan adalah sudut orientasi rudal.

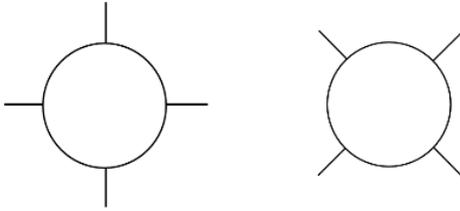
Pada rudal yang dipandu, khususnya rudal taktis, objek yang dikendalikan oleh *autopilot* adalah permukaan yang mengalami gaya aerodinamika, yakni sirip dari rudal. Sirip untuk rudal dapat ditempatkan di 3 tempat, yaitu hidung rudal (disebut sebagai *canard-controlled*), badan rudal (disebut sebagai *wing-controlled*), dan ekor rudal (disebut sebagai *tail-controlled*).



Gambar 2.6 Rudal dengan tempat sirip (kiri) hidung rudal, (tengah) badan rudal, dan (kanan) ekor rudal [3]

Jumlah sirip yang digunakan pada rudal umumnya sebanyak 4 buah. Dari keempat sirip ini, terdapat 2 cara untuk menyusunnya pada rudal, yaitu (1) *plus* dan (2) *cross*. Pada susunan *plus*, beban kerja sirip rudal terbagi menjadi 2, dua sirip untuk mengendalikan *pitch* rudal, dua sirip lainnya untuk mengendalikan *yaw* rudal. Sedangkan pada susunan *cross*, beban kerja sirip terbagi rata ke setiap sirip pada rudal.

Besarnya defleksi dari masing – masing sirip pada rudal akan menentukan besarnya gaya dan momen aerodinamika yang diterima rudal, alhasil orientasi dan kecepatan rudal terhadap bumi dapat berubah. Sehingga rudal dapat dikendalikan dengan mengubah besarnya sudut defleksi dari sirip rudal.



Gambar 2.7 Susunan sirip pada rudal (kiri) susunan *plus*, dan (kanan) susunan *cross* [3]

2.4.1 Non-Linear State Feedback Decoupler[5]

Dari pilihan variabel dapat yang dikendalikan pada rudal, dapat dilihat bahwa rudal merupakan sistem yang memiliki banyak masukan dan banyak keluaran (*Multiple Input Multiple Output*, MIMO). Pada persamaan dinamika rudal, rudal memiliki persamaan matematis yang *non-linear* serta satu persamaan dengan persamaan lainnya *coupled*.

Decoupling digunakan untuk menghilangkan hubungan antar persamaan pada rudal yang saling berinteraksi, sehingga satu sistem rudal keseluruhan yang memiliki banyak masukan dan banyak keluaran (MIMO) dapat diubah menjadi beberapa sistem yang memiliki satu masukan dan satu keluaran (*Single Input Single Output*, SISO). Lalu agar rudal dapat dikendalikan dengan LQT, rudal dilakukan linearisasi. Proses linearisasi ini dilakukan dengan cara mengumpukan balikkan *state* dari rudal (*state feedback*), sehingga proses *decoupling* ini seperti memaksa menghilangkan interaksi antar sub-sistem dan membuat persamaan dinamika rudal menjadi linier. Dalam *state space*, suatu sistem *non-linear* dapat dinyatakan dengan Persamaan (2.31) dan Persamaan (2.32).

$$\dot{\mathbf{x}}(t) = \mathbf{G}(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t)\mathbf{u} \quad (2.31)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (2.32)$$

Di mana:

$\mathbf{G}(\mathbf{x}, t)$ = Matriks *state* sistem *non-linear* dan *coupled*

$\mathbf{F}(\mathbf{x}, t)$ = Matriks masukan yang memiliki *inverse* atau *pseudoinverse*

\mathbf{C} = Matriks keluaran sistem

Didefinisikan suatu matriks sistem \mathbf{A} yang bernilai konstan dan *decoupled* (terpisah dan independen) dan matriks masukan \mathbf{B} yang juga merupakan matriks konstan sedemikian rupa sehingga dapat dibentuk suatu persamaan *state* baru dengan bentuk Persamaan (2.33).

$$\dot{x}(t) = G(x, t) + F(x, t)u + Ax - Ax + B\hat{u} - B\hat{u} \quad (2.33)$$

Di mana:

\hat{u} = Masukan baru

Persamaan (2.33) dapat diatur ulang menjadi Persamaan (2.34).

$$\dot{x}(t) = Ax + B\hat{u} + G(x, t) + F(x, t)u - Ax - B\hat{u} \quad (2.34)$$

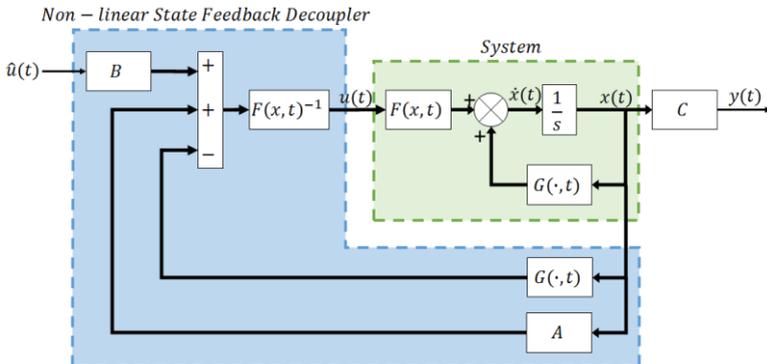
Agar sistem yang baru menjadi sistem yang linier dan *decoupled*, maka Persamaan (2.35) harus dipenuhi.

$$G(x, t) + F(x, t)u - Ax - B\hat{u} = 0 \quad (2.35)$$

Jika matriks $F(x, t)$ memiliki *inverse* atau *pseudoinverse*, agar sistem menjadi linier dan *decoupled*, Persamaan (2.36) harus menjadi masukan sistem *non-linear* dan *coupled*.

$$u = F(x, t)^{-1}(Ax + B\hat{u} - G(x, t)) \quad (2.36)$$

Sistem yang linier dan *decoupled* dengan menggunakan *non-linear state feedback* dapat digambarkan seperti pada Gambar 2.8.



Gambar 2.8 Diagram blok *Non-Linear State Feedback Decoupler*

2.4.2 Kendali Optimal[6]

Dalam sistem pengaturan kontinu, terdapat 2 kasus pengendalian; kasus *regulator* dan kasus *tracking*. Pada kasus *regulator*, masukan sistem berupa nilai konstan. Variabel *state* sistem adalah perubahan pada sekitar nilai *steady state*. Sinyal kendali merupakan sinyal yang dapat mengubah *state* dari sistem. Aksi kendali diperlukan untuk mengembalikan *state* sistem kembali pada nilai *steady state*-nya apabila

state sistem mengalami perubahan karena gangguan. Sedangkan pada kasus *tracking*, masukan sistem berubah terhadap waktu. Dalam kasus ini, sinyal kendali berubah sedemikian rupa sehingga nilai respon keluaran sedekat-dekatnya dengan nilai referensi yang diberikan.

Tujuan dari kendali optimal adalah menentukan sinyal kendali yang tepat agar sistem dapat menjaga nilai *steady state*-nya (pada kasus *regulator*) ataupun mengikuti referensinya (pada kasus *tracking*), dan di saat yang sama, meminimalkan atau memaksimalkan suatu indeks performansi. Umumnya sistem merupakan sistem yang linier dan indeks performansi memiliki bentuk kuadratik, sehingga dikenal istilah *Linear Quadratic Regulator* (LQR) dan *Linear Quadratic Tracking* (LQT).

Karena indeks performansi dibuat minimal atau maksimal, maka kendali dilakukan dalam waktu terbatas (*finite-time case*). Permasalahan dapat dipecah menjadi 2 kondisi, yaitu untuk *state* akhir yang bebas (*free final state*) dan *state* akhir yang terikat (*fixed final state*). Namun kasus ini dapat diperluas menjadi waktu akhir tidak terbatas (*infinite-time case*).

2.4.2.1 Linear Quadratic Tracking (LQT)

Pada *Linear Quadratic Tracking* (LQT), tujuan yang dicapai adalah mengendalikan sistem sedemikian rupa sehingga keluaran dari sistem mengikuti keluaran yang diinginkan sedekat mungkin dengan memperhatikan indeks performansi yang diinginkan. Suatu sistem *Linear Time-Invariant* (LTI) dinyatakan dalam *state space* dapat dinyatakan dengan Persamaan (2.37) dan Persamaan (2.38)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.37)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (2.38)$$

Di mana:

$\mathbf{x}(t)$ = Vektor *state* dari sistem

$\mathbf{u}(t)$ = Vektor sinyal kendali masukan ke sistem

$\mathbf{y}(t)$ = Vektor keluaran dari sistem

\mathbf{A} = Matriks *state* dengan ukuran $n \times n$

\mathbf{B} = Matriks masukan dengan ukuran $n \times r$

\mathbf{C} = Matriks keluaran dengan ukuran $r \times n$

Lalu definisikan sebuah vektor kesalahan dari sistem seperti pada Persamaan (2.39).

$$\mathbf{e}(t) = \mathbf{z}(t) - \mathbf{y}(t) \quad (2.39)$$

Di mana:

$\mathbf{e}(t)$ = Vektor kesalahan dari sistem

$\mathbf{z}(t)$ = Vektor keluaran yang diinginkan / referensi pada sistem

Indeks performansi untuk kasus waktu akhir tidak terbatas diberikan dengan Persamaan (2.40).

$$\lim_{t_f \rightarrow \infty} J = \lim_{t_f \rightarrow \infty} \frac{1}{2} \int_0^{\infty} [\mathbf{e}'(t)\mathbf{Q}\mathbf{e}(t) + \mathbf{u}'(t)\mathbf{R}\mathbf{u}(t)] dt \quad (2.40)$$

Dengan \mathbf{Q} merupakan matriks ukuran $n \times n$ semi-definit positif. Sedangkan \mathbf{R} merupakan matriks ukuran $r \times r$ definit positif. Pada indeks performansi, determinan matriks \mathbf{Q} menentukan seberapa kecil kesalahan dari sistem. Nilai determinan \mathbf{Q} yang besar akan membuat kesalahan keluaran dari sistem sekecil mungkin. Sedangkan determinan matriks \mathbf{R} menentukan seberapa kecil penggunaan *energy drive* dari sistem. Semakin besar nilai determinan \mathbf{R} , penggunaan *energy drive* akan menjadi kecil. Sehingga dengan indeks performansi dari Persamaan (2.40), akan dibuat sinyal kendali dengan nilai kesalahan keluaran yang kecil, namun di sisi lain diinginkan juga penggunaan *energy drive* yang kecil.

Berikut langkah – langkah yang diperlukan untuk membuat sebuah sistem menggunakan LQT.

1. Dapatkan parameter matriks \mathbf{A} , \mathbf{B} , \mathbf{C} dari sistem.
2. Tentukan orde dan nilai matriks \mathbf{Q} dan \mathbf{R} .
Orde matriks \mathbf{Q} mengikuti jumlah banyaknya keluaran, sedangkan orde matriks \mathbf{R} mengikuti jumlah banyaknya masukan. Pertimbangan untuk menentukan nilai matriks \mathbf{Q} dan \mathbf{R} yaitu jika ingin lebih meminimalkan *energy drive* dibandingkan kesalahan keluaran, maka nilai $|\mathbf{R}|$ dibuat lebih besar dari $|\mathbf{Q}|$. Sedangkan jika ingin lebih meminimalkan kesalahan keluaran dibandingkan dengan *energy drive*, maka nilai $|\mathbf{Q}|$ dibuat lebih besar dari $|\mathbf{R}|$.
3. Cari solusi Persamaan Aljabar Riccati (*Algebraic Riccati Equation*) \mathbf{P} menggunakan Persamaan (2.41).

$$\mathbf{A}'\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{C}'\mathbf{Q}\mathbf{C} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}'\mathbf{P} = 0 \quad (2.41)$$

4. Membuat matriks *Kalman Gain* \mathbf{K} dengan menggunakan Persamaan (2.42).

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}'\mathbf{P} \quad (2.42)$$

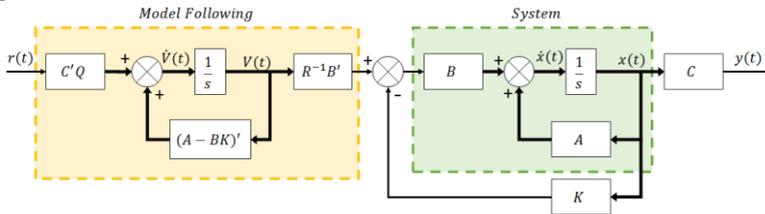
5. Membuat *model following* dengan persamaan *state* seperti pada Persamaan (2.43).

$$\dot{V}(t) = -(A - BK)'V(t) - C'Qr(t) \quad (2.43)$$

6. Sinyal kendali optimal yang menjadi masukan ke sistem adalah Persamaan (2.44).

$$u^*(t) = -Kx(t) + R^{-1}B'V(t) \quad (2.44)$$

Diagram blok dari sistem dengan LQT yang dibentuk dapat dilihat pada Gambar 2.9.



Gambar 2.9 Diagram Blok sistem dengan LQT

2.5 Pemandu Rudal[3]

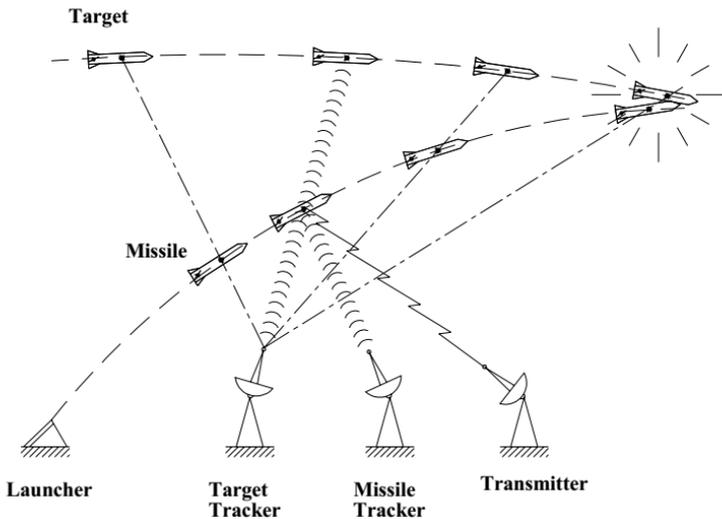
Pemandu pada rudal dapat didefinisikan sebagai strategi untuk menggerakkan rudal ke titik di mana sasaran berada. Di sisi lain, pengendali pada rudal didefinisikan sebagai taktik dalam menggunakan aktuator pada rudal agar dapat merealisasikan strategi yang dihasilkan oleh pemandu. Fungsi utama dari elemen yang digunakan pada sistem pemandu adalah *sensing*, mengolah informasi yang didapat, mengkoreksi.

Dalam mencari letak sasaran, pemandu dapat dibagi ke dalam 2 grup, yaitu pemandu yang tidak berhubungan dengan sasaran (pemanduan secara tidak langsung) dan pemandu yang berhubungan dengan sasaran (pemanduan secara langsung). Pada pemanduan secara tidak langsung, rudal digerakkan ke suatu titik yang telah ditentukan sebelumnya. Sehingga rudal tidak mengetahui persis di mana sasarannya. Rudal hanya mengikuti trayektori yang telah diberikan sebelumnya. Pada pemanduan secara langsung, rudal bergerak berdasarkan informasi mengenai sasaran yang didapat dari *seeker*. Namun, dikarenakan kesalahan internal maupun eksternal, rudal dapat tidak persis mencapai titik pencegatan dengan sasaran, sehingga tembakan rudal dapat meleset.

Berdasarkan cara mengikuti sasaran, pemanduan secara langsung dapat dibedakan menjadi 3 grup:

- *Command to Line-of-Sight Guidance (CLOS Guidance)*
- *Beam Riding Guidance (BRG)*
- *Homing Guidance (HG)*

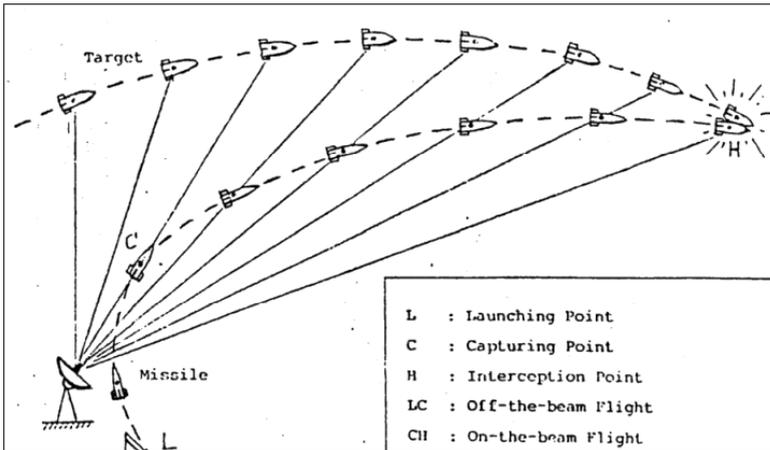
Pemandu CLOS merupakan satu dari dua metode pemanduan jenis LOS. Metode Pemandu LOS disebut juga sebagai metode tiga titik karena tempat peluncuran rudal, rudal, dan sasaran membentuk tiga titik seperti sebuah segitiga. Metode ini dapat dilihat pada Gambar 2.10. Metode pemanduan jenis LOS lainnya ialah BRG. Pada BRG, rudal dipandu untuk bergerak menuju sasaran oleh sebuah radar atau sebuah laser yang diarahkan ke sasaran secara terus menerus, sehingga rudal selalu segaris lurus dengan sasaran. Jika laser diarahkan terus menerus ke sasaran, dan rudal selalu mengarah ke sasaran, maka rudal pasti akan mengenai sasaran. Metode BRG dapat dilihat pada Gambar 2.11.



Gambar 2.10 *Command to Line-of-Sight Guidance* [2]

Homing Guidance didefinisikan sebagai sebuah metode di mana rudal mengendalikan dirinya bergerak menuju sasaran dikarenakan instrumen pemandu berada dalam rudal memberikan komando pergerakan rudal yang diinginkan berdasarkan karakteristik dari sasaran (posisi, sudut relatif, dst). Secara singkat, rudal dengan pemandu ini

beroperasi berdasarkan sinyal yang didapat dari sasaran. Pemandu jenis ini lebih praktis dibandingkan dengan pemandu lainnya karena semua instrumen yang diperlukan oleh pemandu sudah ada dalam rudal, yakni instrumen pengukuran gerakan rudal (*accelerometer* dan *gyroscope*) dan instrumen untuk mengikuti sasaran (*seeker*). Kelebihan lainnya pada pemandu HG dibandingkan dengan CLOS maupun BRG adalah akurasi pengukuran gerakan relatif antara rudal dengan sasaran saat rudal bergerak mendekat menuju sasaran meningkat, sedangkan pada pemandu jenis lainnya berkurang.



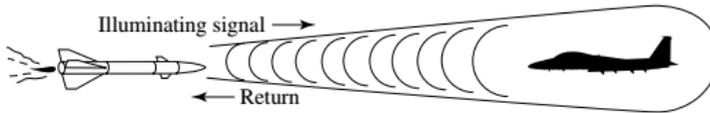
Gambar 2.11 *Beam Riding Guidance* [2]

Pada HG, instrumen untuk mengikuti sasaran yang ada dalam rudal memerlukan sebuah metode *homing* agar dapat sasaran dapat terdeteksi. Terdapat empat metode *homing* yang bisa digunakan untuk mendeteksi sasaran ini, yaitu:

- *Active Homing*
- *Semiactive Homing*
- *Passive Homing*

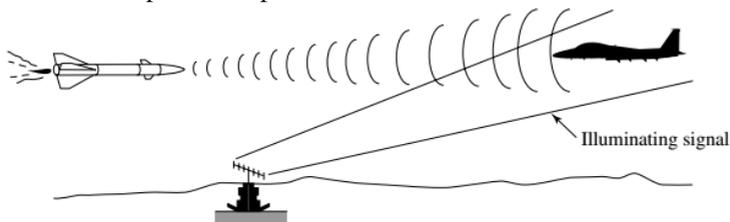
Pada metode *active homing*, rudal membawa dua alat, yaitu sebuah pemancar untuk memancarkan suatu radiasi dan penerima untuk menerima pantulan radiasi pemancar dari sasaran. Rudal mendeteksi sasaran dengan memancarkan sebuah radiasi, lalu radiasi ini mengenai sasaran dan dipantulkan olehnya. Hasil pantulan diterima kembali oleh rudal. Selanjutnya rudal akan bergerak menuju posisi sasaran yang telah

terdeteksi. Jenis radiasi yang digunakan bervariasi mulai dari gelombang radio, cahaya tampak, sinar inframerah, bahkan gelombang suara. Metode ini dapat dilihat pada Gambar 2.12.



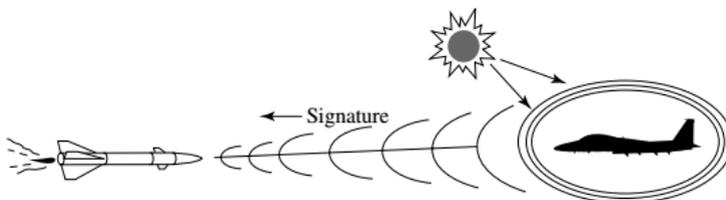
Gambar 2.12 Active Homing [1]

Pada metode *semiactive homing*, radiasi pemancar tidak dipancarkan dari rudal, melainkan dari sumber lainnya, sehingga rudal hanya perlu membawa penerima saja. Pemancar radiasi ini umumnya terletak pada tempat di mana rudal diluncurkan. Dengan menggunakan radar dengan radiasi yang kuat, jarak pendeteksian sasaran juga lebih luas. Metode ini dapat dilihat pada Gambar 2.13.



Gambar 2.13 Semi-active Homing [1]

Pada metode *passive homing*, alat pemancar tidak diperlukan karena sumber radiasi berasal dari sasaran sendiri, sehingga rudal hanya membawa alat penerima saja. Metode jenis ini merupakan metode paling mudah dan banyak digunakan. Pada metode ini, pemancar dari sasaran yang mudah terdeteksi adalah antenna radar sasaran, pemancar radio sasaran, dan suhu panas pesawat terbang sasaran. metode ini dapat dilihat pada Gambar 2.14.



Gambar 2.14 Passive Homing [1]

Beberapa metode HG yang ada pada literatur antara lain:

- Pemandu Konvensional
- Pemandu berdasarkan Teori Kendali Optimal
- Pemandu berdasarkan Teori Permainan
- Pemandu berdasarkan Teknik Kendali *Robust*
- Pemandu berdasarkan *Artificial Intelligence*

2.5.1 Pemandu Pengejar (*Pursuit Guidance*)[1]

Pemandu ini merupakan jenis Pemandu Konvensional. Pemandu Pengejar bekerja dengan cara meniadakan sudut *Line-of-Sight* (LOS) relatif antara rudal dengan sasaran. Namun, pemandu ini memiliki kemampuan terbatas saat berhadapan dengan sasaran yang memiliki kemampuan *maneuver* lebih cepat dan lincah. Pemandu jenis ini memiliki dua variasi:

- *Attitude Pursuit Guidance* (APG)
- *Velocity Pursuit Guidance* (VPG)

Pada pemandu APG, sumbu longitudinal rudal (sumbu X_b pada rudal) dibuat agar selalu mengarah ke sasaran. Agar dapat mengikuti sasaran, *seeker* pada rudal harus memiliki *Field-of-View* (FOV) yang luas. Ide konseptual dari VPG adalah membuat rudal selalu bergerak ke posisi sasaran saat ini. Dengan kata lain, vektor kecepatan rudal dijaga agar selalu mengarah ke sasaran. Strategi ini mirip seperti seekor pemangsa yang mengejar mangsanya (misal seekor macan mengejar seekor kijang). Pengejaran ini akan berakhir dengan pencegatan mangsa dengan syarat kecepatan pemangsa lebih cepat daripada mangsanya.

Informasi yang diperlukan dalam pemandu jenis ini adalah posisi sasaran dengan rudal terhadap koordinat bumi. Dengan diketahuinya posisi relatif ini, dapat ditentukan sudut relatif yang dibentuk antara rudal dengan sasaran relatif terhadap bumi. Sehingga sudut ini yang akan dikeluarkan dari pemandu untuk dilaksanakan oleh *autopilot*.

Posisi sasaran terhadap koordinat bumi dinotasikan sebagai x_t, y_t, z_t dan posisi rudal terhadap koordinat bumi dinotasikan sebagai x_m, y_m, z_m . Posisi sasaran relatif terhadap rudal pada masing – masing sumbu pada sistem koordinat bumi dapat dilihat pada Persamaan (2.45a) hingga Persamaan (2.45c).

$$x_r = x_t - x_m \quad (2.45a)$$

$$y_r = y_t - y_m \quad (2.45b)$$

$$z_r = z_t - z_m \quad (2.45c)$$

Dari Persamaan (2.45a) hingga Persamaan (2.45c), dapat dicari sudut relatif yang dibentuk antara rudal dengan sasaran (visualisasi sudut relatif ini dapat dilihat pada Gambar 2.15). Dengan menggunakan sifat trigonometri, persamaan yang digunakan untuk mencari sudut relatif ini dapat dilihat pada Persamaan (2.46) dan Persamaan (2.47).

$$\theta_r = \tan^{-1} \left(\frac{z_r}{\sqrt{x_r^2 + y_r^2}} \right) \quad (2.46)$$

$$\psi_r = \tan^{-1} \left(\frac{y_r}{x_r} \right) \quad (2.47)$$

Di mana:

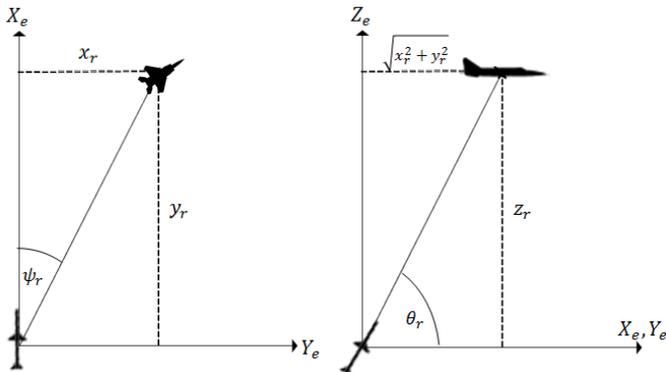
θ_r = Sudut ketinggian relatif rudal - sasaran

ψ_r = Sudut azimuth relatif rudal - sasaran

x_r = Posisi relatif rudal – sasaran pada sumbu x bumi

y_r = Posisi relatif rudal – sasaran pada sumbu y bumi

z_r = Posisi relatif rudal – sasaran pada sumbu z bumi



Gambar 2.15 Sudut relatif rudal – sasaran

2.6 Prediksi Posisi Sasaran

Pencegatan dengan sasaran yang bergerak tentu lebih sulit dibandingkan dengan sasaran yang diam. Pada *seeker* sendiri, perubahan sudut LOS antara rudal – sasaran akan membesar seiring dengan jarak rudal – sasaran yang mengecil, dengan kecepatan rudal yang tetap maupun rudal masih memiliki akselerasi. Sehingga terdapat kemungkinan bahwa pencegatan sasaran meleset.

Setelah rudal ditembakkan, rudal akan memprediksi posisi sasaran di saat yang akan datang. Selama di udara, rudal akan selalu memprediksi posisi sasaran. Rudal memerlukan informasi mengenai posisi sasaran untuk memprediksi posisi sasaran di saat mendatang. Rudal memiliki *sampling time* (t_s) untuk mengambil informasi posisi sasaran, sehingga informasi posisi sasaran ada secara periodik.

Karena pemandu yang digunakan memiliki masukan berupa posisi sasaran, maka keluaran dari prediksi posisi sasaran adalah prediksi posisi sasaran di masa mendatang. Lebih tepatnya hasil prediksi posisi di mana pencegatan akan terjadi. Prediksi dilakukan dengan menggunakan informasi posisi, kecepatan, dan percepatan sasaran. Ketiga informasi ini perlu dimiliki sesaat sebelum rudal ditembakkan. Dengan menggunakan 3 informasi posisi sasaran terakhir, dapat ditentukan posisi, kecepatan, dan percepatan sasaran 2 waktu *sample* sebelumnya. Lalu dari hasil ini, dapat diprediksi posisi sasaran di saat mendatang dengan waktu tertentu.

Posisi sasaran yang didapat oleh rudal setiap waktu *sample* mulai dari rudal diluncurkan dapat dilihat pada persamaan (2.48).

$$x(n) = p + vt_s n + a(t_s n)^2 \quad (2.48)$$

Di mana:

- $x(n)$ = Posisi sasaran saat n
- p = Posisi awal sasaran
- v = Kecepatan awal sasaran
- a = Percepatan awal sasaran
- t_s = *Sampling time*

Sesaat rudal baru diluncurkan, rudal tidak dapat mengetahui kecepatan dan percepatan sasaran. setelah 3 waktu *sample* setelah rudal diluncurkan, rudal memiliki 3 informasi posisi sasaran, yaitu posisi sasaran saat ini, posisi sasaran 1 waktu *sample* sebelum saat ini, dan 2 waktu *sample* sebelum saat ini. Dengan waktu *sample* sebesar t_s , posisi sasaran saat $x(k-2)$, $x(k-1)$, dan $x(k)$, di mana $k-2$ merupakan waktu saat posisi sasaran diketahui pertama kali, dapat dilihat pada Persamaan (2.49a) hingga Persamaan (2.49c).

$$x(k-2) = p + vt_s + at_s^2 \quad (2.49a)$$

$$x(k-1) = p + 2vt_s + 4at_s^2 \quad (2.49b)$$

$$x(k) = p + 3vt_s + 9at_s^2 \quad (2.49c)$$

Persamaan (2.49a) hingga Persamaan (2.49c) ini dapat ditulis ulang dalam bentuk matriks menjadi Persamaan (2.50).

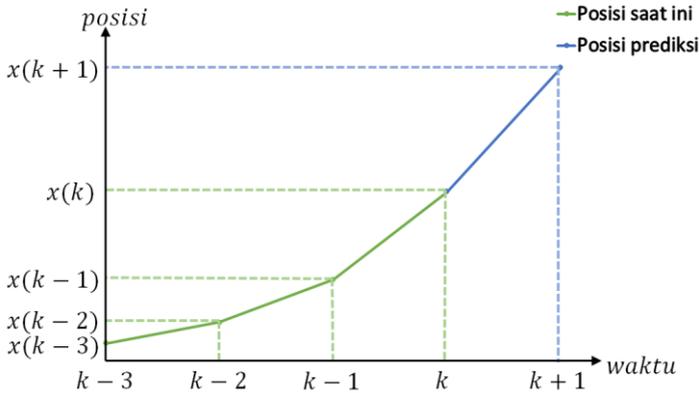
$$\begin{bmatrix} x(k-2) \\ x(k-1) \\ x(k) \end{bmatrix} = \begin{bmatrix} 1 & t_s & t_s^2 \\ 1 & 2t_s & 4t_s^2 \\ 1 & 3t_s & 9t_s^2 \end{bmatrix} \begin{bmatrix} p \\ v \\ a \end{bmatrix} \quad (2.50)$$

Posisi, kecepatan, dan percepatan sasaran saat posisi sasaran diketahui pertama kali, yakni saat $(k-2)$ dapat ditentukan dengan menggunakan Persamaan (2.51).

$$\begin{bmatrix} p \\ v \\ a \end{bmatrix} = \begin{bmatrix} 1 & t_s & t_s^2 \\ 1 & 2t_s & 4t_s^2 \\ 1 & 3t_s & 9t_s^2 \end{bmatrix}^{-1} \begin{bmatrix} x(k-2) \\ x(k-1) \\ x(k) \end{bmatrix} \quad (2.51)$$

Dengan demikian, posisi sasaran saat $(k+1)$ dapat diprediksi menggunakan Persamaan (2.52).

$$x(k+1) = p + 4vt_s + 16at_s^2 \quad (2.52)$$



Gambar 2.16 Posisi sasaran pada waktu *sample* tertentu

Persamaan (2.52) menunjukkan bahwa posisi sasaran 1 waktu *sample* mendatang dapat diprediksi dengan menggunakan 3 informasi terakhir posisi sasaran. Representasi grafis prediksi posisi ini dapat dilihat pada Gambar 2.16. Lalu dengan diketahui kecepatan rudal dan kecepatan sasaran saat ini, dapat diketahui *closing velocity* antara rudal – sasaran dengan menggunakan Persamaan (2.53).

$$V_c = \sqrt{(v_{tx} - v_{mx})^2 + (v_{ty} - v_{my})^2 + (v_{tz} - v_{mz})^2} \quad (2.53)$$

Di mana:

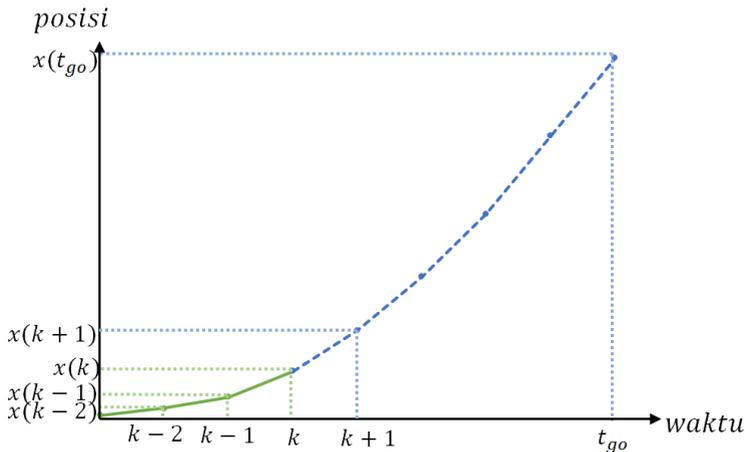
- v_{tx} = Kecepatan sasaran pada sumbu x bumi
- v_{ty} = Kecepatan sasaran pada sumbu y bumi
- v_{tz} = Kecepatan sasaran pada sumbu z bumi
- v_{mx} = Kecepatan rudal pada sumbu x bumi
- v_{my} = Kecepatan rudal pada sumbu y bumi
- v_{mz} = Kecepatan rudal pada sumbu z bumi

Time-to-go rudal, yaitu waktu yang diperlukan rudal untuk bergerak menuju posisi sasaran saat ini, dapat diketahui dengan menggunakan Persamaan (2.54).

$$t_{go} = \frac{\sqrt{x_r^2 + y_r^2 + z_r^2}}{V_c} \quad (2.54)$$

Dengan:

- x_r = Posisi relatif rudal – sasaran pada sumbu x bumi
- y_r = Posisi relatif rudal – sasaran pada sumbu y bumi
- z_r = Posisi relatif rudal – sasaran pada sumbu z bumi



Gambar 2.17 Posisi sasaran saat waktu t_{go}

Posisi sasaran pada saat t_{go} detik dari posisi sasaran saat 2 waktu *sample* sebelumnya didapatkan dengan Persamaan (2.55).

$$x(t_{go}) = p + v(t_{go}) + a(t_{go})^2 \quad (2.55)$$

Posisi di mana rudal akan *impact* dengan sasaran didapat dengan mensubstitusikan hasil Persamaan (2.54) ke Persamaan (2.55). Representasi grafis dari prediksi posisi sasaran saat t_{go} detik mendatang dapat dilihat pada Gambar 2.17. Perlu diketahui bahwa Persamaan (2.48), Persamaan (2.52), dan Persamaan (2.55) hanya berlaku dalam 1 sumbu sistem koordinat bumi, sehingga diperlukan 3 persamaan yang sama persis untuk sumbu – sumbu lainnya.

Halaman ini sengaja dikosongkan

BAB 3

PERANCANGAN SISTEM

Bab ini menjelaskan tahap – tahap pembuatan simulasi dari rudal mengejar sasaran. Pertama membahas pembuatan model matematis dinamika rudal, lalu perancangan linearisasi model rudal dengan *decoupler*, selanjutnya perancangan pengendali LQT, kemudian perancangan pemandu untuk rudal, lalu perancangan pergerakan sasaran, dan diakhiri dengan perancangan prediksi posisi sasaran.

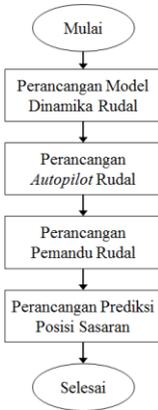
3.1 Gambaran Umum Sistem

Terdapat 4 tahap yang perlu diperhatikan dalam merancang simulasi pengejaran rudal menuju sasaran. Pertama model matematis dinamika rudal harus dimiliki. Selanjutnya dari model matematis yang ada, dapat dirancang pengendali yang diinginkan. Penelitian tugas akhir ini menggunakan pengendali LQT dan variabel yang dikendalikan adalah sudut orientasi rudal. Agar pengendalian rudal dapat dilakukan, digunakan *decoupler* untuk menghilangkan interaksi antar persamaan dinamika rotasi rudal.

Pemandu rudal yang sesuai dengan masukan dari pengendali dirancang setelah model dinamika rudal dan pengendali LQT dirancang. Karena variabel yang dikendalikan adalah sudut orientasi rudal, maka pemandu yang digunakan yaitu pemandu yang mengeluarkan sudut orientasi yang diinginkan. Selanjutnya dirancang prediksi posisi sasaran agar kemungkinan tembakan rudal meleset berkurang. Diagram alir dari perancangan simulasi rudal ini dapat dilihat pada Gambar 3.1.

3.2 Perancangan Model Rudal

Perancangan model matematis rudal merupakan tahap awal dari perancangan sistem pengendalian rudal. Hukum Newton kedua menjadi landasan penurunan persamaan dinamika rudal. Hukum Newton kedua menyatakan bahwa percepatan suatu benda berbanding lurus dengan jumlah gaya yang bekerja pada benda tersebut, dan berbanding terbalik dengan massanya. Arah percepatan sama dengan arah total gaya yang bekerja pada benda tersebut. Dari hukum newton kedua ini, dapat diturunkan persamaan percepatan rudal.



Gambar 3.1 Diagram alir perancangan simulasi rudal

Persamaan dinamika rudal terdiri dari 2 jenis, yaitu dinamika translasi dan dinamika rotasi. Persamaan dinamika translasi merupakan persamaan yang menunjukkan gaya – gaya yang bekerja pada rudal sehingga percepatan rudal dapat berubah. Gaya – gaya yang bekerja pada rudal terdiri dari gaya gravitasi, gaya aerodinamika, dan gaya hasil interaksi antara kecepatan translasi rudal dengan kecepatan sudut rudal. Dengan demikian, persamaan percepatan translasi rudal yang digunakan dapat dilihat pada Persamaan (3.1a) hingga Persamaan (3.1c).

$$\frac{du}{dt} = \frac{Thrust - mg \sin \theta + C_x QS}{m} + Rv - Qw \quad (3.1a)$$

$$\frac{dv}{dt} = \frac{mg \sin \phi \cos \theta + C_y QS}{m} + Pw - Ru \quad (3.1b)$$

$$\frac{dw}{dt} = \frac{mg \cos \phi \cos \theta + C_z QS}{m} + Qu - Pv \quad (3.1c)$$

Dengan

$$C_x = C_{x1}\alpha + C_{x2}M \quad (3.2a)$$

$$C_y = C_{y1}\beta + C_{y2}M + C_{y3}\delta_r \quad (3.2b)$$

$$C_z = C_{z1}\alpha + C_{z2}M + C_{z3}\delta_e \quad (3.2c)$$

Persamaan dinamika rudal merupakan persamaan yang menunjukkan momen – momen yang bekerja pada rudal sedemikian rupa sehingga percepatan sudut rudal dapat berubah. Momen – momen yang bekerja pada rudal terdiri dari momen aerodinamika dan momen

dikarenakan interaksi antara kecepatan sudut rudal dengan kecepatan translasi rudal. Dengan demikian, persamaan percepatan rotasi rudal yang digunakan dapat dilihat pada Persamaan (3.3a) hingga Persamaan (3.3c).

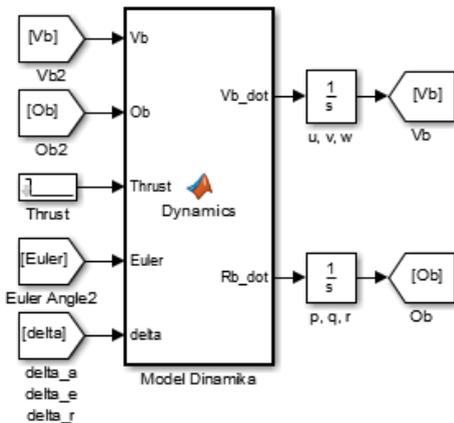
$$\frac{dp}{dt} = \frac{QSD(C_{l1}\alpha + C_{l2}\beta + C_{l3}p + C_{l4}\delta_a) + (I_y - I_z)qr}{I_x} \quad (3.3a)$$

$$\frac{dq}{dt} = \frac{QSD(C_{m1}\alpha + C_{m2}q + C_{m3}\delta_e) + (I_z - I_x)pr}{I_y} \quad (3.3b)$$

$$\frac{dr}{dt} = \frac{QSD(C_{n1}\beta + C_{n2}r + C_{n3}\delta_r) + (I_x - I_y)pq}{I_z} \quad (3.3c)$$

Model rudal dibuat dengan menggunakan fitur Simulink dalam perangkat lunak MATLAB. Blok yang digunakan pada Simulink adalah MATLAB *Function* dari kategori *User-Defined Functions*. Blok ini diberi nama “Model Dinamika” dan dapat dilihat pada Gambar 3.2. Persamaan dinamika translasi rudal dan persamaan dinamika rotasi rudal dituliskan dalam blok ini.

Sesuai dengan persamaan dinamika rudal, blok ini memiliki keluaran percepatan translasi rudal dalam sistem koordinat rudal dan percepatan sudut rudal dalam sistem koordinat rudal. Dengan memberikan integrator pada keluaran blok, akan didapatkan kecepatan translasi dan kecepatan sudut rudal. Masukan pada blok ini adalah kecepatan translasi rudal, kecepatan sudut rudal, gaya pendorong rudal, Sudut orientasi rudal (Sudut Euler), dan besar sudut defleksi sirip rudal.



Gambar 3.2 Blok Simulink Model Dinamika Rudal

Pada persamaan dinamika rotasi rudal, sudut serang dan sudut slip bisa didapatkan dari fungsi trigonometri kecepatan translasi rudal. Lalu momen inersia dan koefisien aerodinamika sudah didefinisikan sebelumnya.

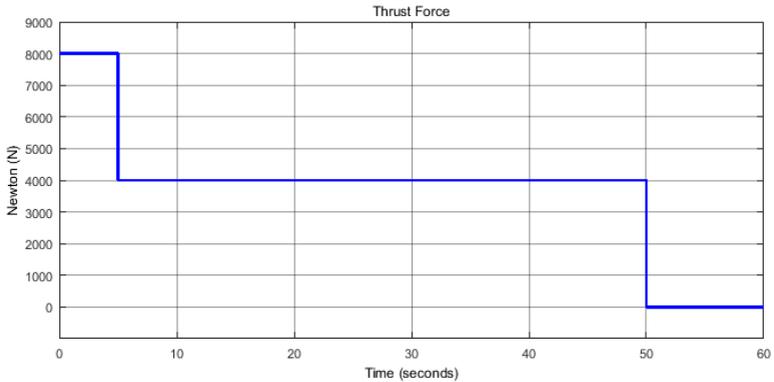
Parameter massa, diameter, dan luas permukaan rudal yang ada pada model matematis rudal diperoleh dari rudal *RHAN I22B Guided*. Parameter – parameter tersebut langsung dimasukkan ke dalam blok model dinamika rudal. Nilai momen inersia didapatkan dengan asumsi bahwa rudal berbentuk tabung. Sedangkan parameter – parameter koefisien aerodinamika menggunakan asumsi. Nilai dari parameter – parameter yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Parameter – parameter spesifikasi rudal

Parameter	Nilai	Satuan
Massa (m)	142	kg
Percepatan gravitasi (g)	9.8	m/s ²
Diameter (d)	0.118	m
Luas permukaan (S)	1.1711	m ²
Kepadatan udara (ρ)	1.2251	kg/m ³
Kecepatan suara (SoS)	340	m/s
Momen Inersia X_b (I_x)	0.2475	kg.m ²
Momen Inersia Y_b (I_y)	24.8004	kg.m ²
Momen Inersia Z_b (I_z)	24.8004	kg.m ²
Koefisien C_{x1}	-0.35	-
Koefisien C_{x2}	-0.05	-
Koefisien C_{y1}	$-90/\pi$	-
Koefisien C_{y2}	-0.3	-
Koefisien C_{y3}	-3	-
Koefisien C_{z1}	$-90/\pi$	-
Koefisien C_{z2}	-0.3	-
Koefisien C_{z3}	-1.5	-
Koefisien C_{l1}	-0.1	-
Koefisien C_{l2}	-0.1	-
Koefisien C_{l3}	-1	-
Koefisien C_{l4}	5	-
Koefisien C_{m1}	$-120/\pi$	-
Koefisien C_{m2}	-2	-

Parameter	Nilai	Satuan
Koefisien C_{m3}	2	-
Koefisien C_{n1}	$90/\pi$	-
Koefisien C_{n2}	-9	-
Koefisien C_{n3}	9	-

Model *Thrust* yang digunakan pada rudal memiliki 2 fase. Fase pertama yaitu fase *boost* dengan durasi 5 detik setelah rudal ditembakkan. Pada fase ini, rudal melakukan akselerasi hingga rudal memiliki kecepatan nominalnya. Fase kedua yaitu fase *sustain* dengan durasi 45 detik setelah fase *boost* berakhir. Dalam fase ini, rudal telah memiliki kecepatan nominalnya dan bergerak menuju sarasannya. Grafik model Thrust yang digunakan dapat dilihat pada Gambar 3.3.

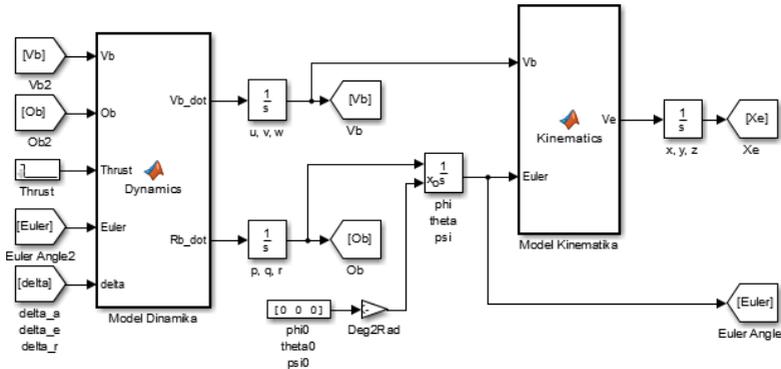


Gambar 3.3 Model *Thrust* yang digunakan pada masukan model rudal

Sebuah transformasi koordinat dari sistem koordinat rudal menjadi sistem koordinat bumi diperlukan agar rudal dapat direpresentasikan dalam sistem koordinat bumi. Transformasi ini dilakukan dengan menggunakan Persamaan (3.4).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & \dots \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & \dots \\ -s\theta & s\phi c\theta & \dots \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.4)$$

Posisi rudal dalam sistem koordinat bumi didapat dengan mengintegalkan kecepatan translasi rudal dalam sistem koordinat bumi. Sedangkan orientasi rudal dalam sistem koordinat bumi didapat dengan mengintegalkan kecepatan sudut rudal dalam sistem koordinat rudal. Ini dikarenakan jika *gyroscope* pada rudal diatur mengikuti sistem koordinat bumi sebelum rudal diluncurkan, maka sudut orientasi rudal sudah mengikuti sistem koordinat bumi.



Gambar 3.4 Blok Simulink model dinamika dan kinematika rudal

Blok kinematika rudal dibuat dengan menggunakan blok MATLAB *Function* seperti halnya blok dinamika rudal. Blok kinematika rudal ini diberi label “Model Kinematika”. Blok ini selanjutnya dihubungkan dengan blok dinamika rudal. Sehingga blok model rudal yang dihasilkan dapat dilihat pada Gambar 3.4. Pada gambar tersebut, $phi0$, $theta0$, $psi0$ merupakan orientasi awal rudal.

3.3 Perancangan *Non-Linear State Feedback Decoupler*

Persamaan yang difokuskan adalah persamaan dinamika rotasi rudal, mengingat bahwa variabel yang dikendalikan adalah sudut orientasi rudal. Dari persamaan dinamika rotasi rudal, yakni Persamaan (3.3a) hingga Persamaan (3.3c), terlihat bahwa percepatan sudut pada suatu sumbu dipengaruhi oleh momen – momen yang merupakan fungsi sudut serang (α), sudut slip (β), kecepatan sudut pada sumbu tersebut, sudut defleksi sirip rudal untuk sumbu tersebut, dan interaksi silang antar kecepatan sudut selain pada sumbu tersebut. Hal ini menyebabkan persamaan dinamika rotasi rudal memiliki sifat *non-linear* dan *coupled*. Maka

dilakukan linearisasi menggunakan *Non-Linear State Feedback* untuk menghilangkan sifat *non-linear* pada rudal sehingga persamaan dinamika rotasi rudal menjadi linier. Selanjutnya digunakan *decoupler* untuk menghilangkan interaksi antar persamaan dinamika rotasi.

Linearisasi dilakukan dengan memaksa menghilangkan momen – momen yang menyebabkan rudal memiliki sifat *non-linear*. Masukan pada persamaan dinamika rotasi rudal adalah besarnya sudut defleksi dari sirip rudal, yakni $\delta_a, \delta_e, \delta_r$. Masukan dari persamaan dinamika rotasi rudal merupakan momen – momen yang menyebabkan rudal menjadi *non-linear* agar persamaan rudal menjadi linier. Parameter baru dengan *time constant* sesuai keinginan ditambahkan agar dinamika rotasi rudal memiliki respon seperti orde satu. Rudal memerlukan respon yang cepat, oleh karena itu ditentukan *time constant* yang baru sebesar 0.1 detik. Kemudian parameter masukan baru ditambahkan agar rudal dapat dikendalikan. Parameter masukan baru ini yang akan menjadi referensi gerakan rudal. Dengan demikian, masukan pada dinamika rotasi rudal adalah Persamaan (3.5a) hingga Persamaan (3.5c).

$$\delta_a = \frac{1}{C_{l4}} \left(-C_{l1}\alpha - C_{l2}\beta - C_{l3}p - \frac{(I_y - I_z)qr}{QSD} \dots \right. \\ \left. - \frac{\lambda_p I_x p}{QSD} + \frac{\lambda_p I_x U_p}{QSD} \right) \quad (3.5a)$$

$$\delta_e = \frac{1}{C_{m3}} \left(-C_{m1}\alpha - C_{m2}q - \frac{(I_z - I_x)pr}{QSD} - \frac{\lambda_q I_y q}{QSD} \dots \right. \\ \left. + \frac{\lambda_q I_y U_q}{QSD} \right) \quad (3.5b)$$

$$\delta_r = \frac{1}{C_{n3}} \left(-C_{n1}\beta - C_{n2}r - \frac{(I_x - I_y)pq}{QSD} - \frac{\lambda_r I_z r}{QSD} \dots \right. \\ \left. + \frac{\lambda_r I_z U_r}{QSD} \right) \quad (3.5c)$$

Dengan:

$$\lambda_p = \frac{1}{\tau_p} = 10 \quad (3.6a)$$

$$\lambda_q = \frac{1}{\tau_q} = 10 \quad (3.6b)$$

$$\lambda_r = \frac{1}{\tau_r} = 10 \quad (3.6c)$$

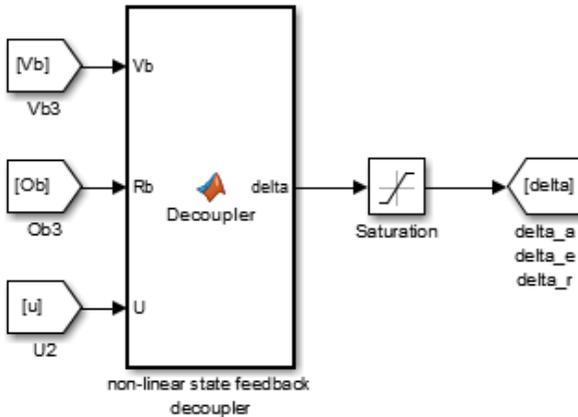
Persamaan dinamika rotasi rudal yang linier dan *decoupled* dapat dilihat pada Persamaan (3.7a) hingga Persamaan (3.7c).

$$\frac{dp}{dt} = -\lambda_p p + \lambda_p U_p \quad (3.7a)$$

$$\frac{dq}{dt} = -\lambda_q q + \lambda_q U_q \quad (3.7b)$$

$$\frac{dr}{dt} = -\lambda_r r + \lambda_r U_r \quad (3.7c)$$

Pada perancangan simulasi, blok model rudal dibiarkan tetap tidak linier. Proses linearisasi dan *decoupling* terjadi diluar model matematis rudal. Blok *decoupler* dirancang menggunakan blok *MATLAB Function*. Blok ini memiliki masukan kecepatan translasi rudal, kecepatan sudut rudal, serta masukan referensi. Sedangkan keluarannya berupa sudut defleksi sirip rudal yang ditentukan setelah dilakukan linearisasi dan *decoupling*. Blok ini diberi label “*non-linear state feedback decoupler*”. Keluaran blok ini diberi blok saturasi dikarenakan sudut yang dibentuk defleksi sirip rudal memiliki nilai maksimum dan minimum sebesar [60 - 60] derajat atau [$\pi/3$ - $\pi/3$] radian. Blok Simulink *Non-Linear State Feedback Decoupler* ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Blok Simulink linearisasi dengan *decoupler*

3.4 Perancangan *Linear Quadratic Tracking (LQT)*

Langkah awal yang diperlukan dalam merancang pengendali LQT adalah membuat *state space* dari model yang digunakan. Sistem rudal

yang sebenarnya memiliki sifat *non-linear* telah dilakukan linearisasi sebelumnya, sehingga model yang digunakan adalah model yang telah dilinearisasi. Karena variabel yang dikendalikan ialah sudut orientasi rudal, maka variabel ini masuk ke dalam *state space* yang dirancang. Persamaan *state space* yang dibentuk dari model rudal yang telah linier dan *decoupled* dapat dilihat dilihat pada Persamaan (3.7a) dan Persamaan (3.7b).

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \\ \dot{\theta} \\ \dot{q} \\ \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_q & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_r \end{bmatrix} \begin{bmatrix} \phi \\ p \\ \theta \\ q \\ \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \lambda_p & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \lambda_q & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_r \end{bmatrix} \begin{bmatrix} U_p \\ U_q \\ U_r \end{bmatrix} \quad (3.7a)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ p \\ \theta \\ q \\ \psi \\ r \end{bmatrix} \quad (3.7b)$$

Dari Persamaan (3.7a) dan Persamaan (3.7b), matriks **A**, **B**, **C** yang dibentuk dapat dilihat pada Persamaan (3.8), Persamaan (3.9), dan Persamaan (3.10).

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_q & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_r \end{bmatrix} \quad (3.8)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ \lambda_p & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \lambda_q & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_r \end{bmatrix} \quad (3.9)$$

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Langkah selanjutnya yaitu menentukan orde dan nilai matriks \mathbf{Q} dan \mathbf{R} . Ukuran matriks \mathbf{Q} mengikuti jumlah banyaknya keluaran, sehingga ukuran matriks \mathbf{Q} sebesar 3×3 . Ukuran matriks \mathbf{R} mengikuti jumlah banyaknya masukan yang ada, sehingga ukuran matriks \mathbf{R} adalah 3×3 . Kesalahan keluaran diharuskan seminimal mungkin agar kemungkinan tembakan rudal meleset mengecil, sehingga nilai $|\mathbf{Q}|$ harus lebih besar dari $|\mathbf{R}|$. Oleh karena itu, matriks \mathbf{Q} dan matriks \mathbf{R} yang digunakan seperti pada Persamaan (3.11) dan Persamaan (3.12).

$$\mathbf{Q} = \begin{bmatrix} 10000 & 0 & 0 \\ 0 & 10000 & 0 \\ 0 & 0 & 10000 \end{bmatrix} \quad (3.11)$$

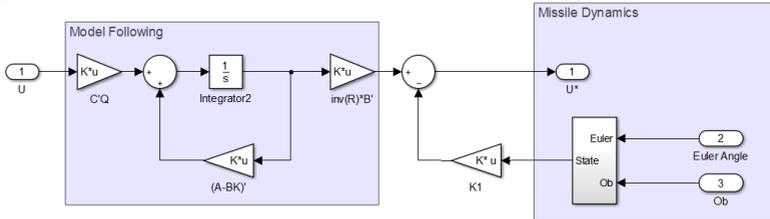
$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Langkah selanjutnya adalah mencari solusi Persamaan Aljabar Riccati dengan menggunakan matriks $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Q}, \mathbf{R}$ yang telah ditentukan sebelumnya. Dalam perangkat lunak MATLAB, digunakan fungsi *care* untuk mencari solusi Persamaan Aljabar Riccati dan *Kalman Gain*. Matriks *Kalman Gain* yang dihasilkan dapat dilihat pada Persamaan (3.13).

$$\mathbf{K} = \begin{bmatrix} 100 & 4.4949 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 4.4949 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 4.4949 \end{bmatrix} \quad (3.13)$$

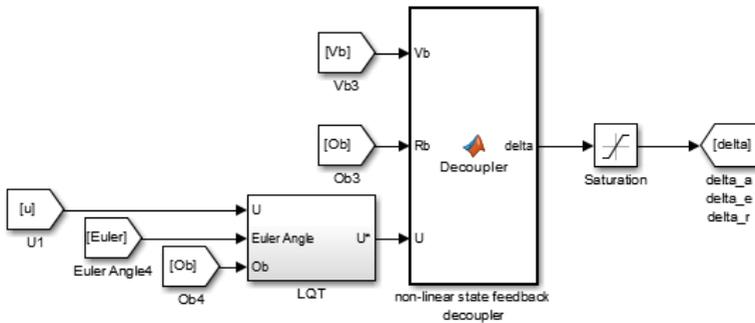
Pengendali LQT memiliki masukan berupa sudut orientasi yang diinginkan, kecepatan sudut rudal, dan sudut orientasi rudal saat ini.

Keluarannya berupa sinyal kendali yang optimal. Sinyal kendali ini yang menjadi masukan ke blok linearisasi dengan *decoupling* sebelumnya. Blok Simulink pengendali LQT yang telah dirancang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Blok Simulink pengendali LQT

Pada perangkat lunak MATLAB, parameter matriks A, B, C, Q, R disimpan pada *file* yang berbeda dari simulasi Simulink. Parameter ini tersimpan pada *workspace* pada MATLAB. Sedangkan pengendali LQT dibuat menggunakan Simulink. Saat simulasi dijalankan, Simulink akan mengambil parameter yang ada pada *workspace* MATLAB. Pengendali LQT dikumpulkan menjadi sebuah subsistem dengan label “LQT”. Dengan demikian, blok Simulink *autopilot* rudal keseluruhan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Blok Simulink pengendali LQT yang dihubungkan dengan *decoupler*

3.5 Perancangan Pemandu Rudal

Pemandu rudal berfungsi sebagai pemberi perintah agar rudal dapat bergerak menuju sasarannya. Pemandu akan menerima informasi posisi rudal saat ini dan posisi sasaran. Lalu pemandu akan mengoreksi berapa

sudut yang diperlukan agar rudal menghadap kepada sasaran. Dengan pemandu jenis Pemandu Pengejar (*Pursuit Guidance*), pemandu akan selalu membuat rudal selalu mengarah ke sasaran.

Sudut relatif yang dibentuk antara rudal dengan sasaran dapat diketahui dengan menggunakan informasi posisi rudal dan posisi sasaran pada sistem koordinat bumi. Sudut relatif yang dibentuk dengan diketahuinya posisi relatif ini dinyatakan dengan Persamaan (3.14) dan Persamaan (3.15).

$$\theta_r = \tan^{-1} \left(\frac{z_r}{\sqrt{x_r^2 + y_r^2}} \right) \quad (3.14)$$

$$\psi_r = \tan^{-1} \left(\frac{y_r}{x_r} \right) \quad (3.15)$$

Di mana:

θ_r = Sudut ketinggian relatif rudal - sasaran

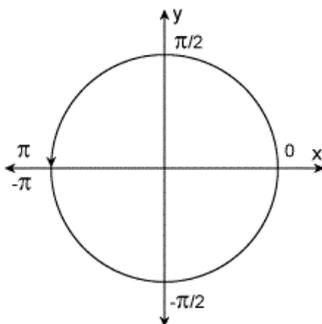
ψ_r = Sudut azimuth relatif rudal - sasaran

x_r = Posisi relatif rudal – sasaran pada sumbu x bumi

y_r = Posisi relatif rudal – sasaran pada sumbu y bumi

z_r = Posisi relatif rudal – sasaran pada sumbu z bumi

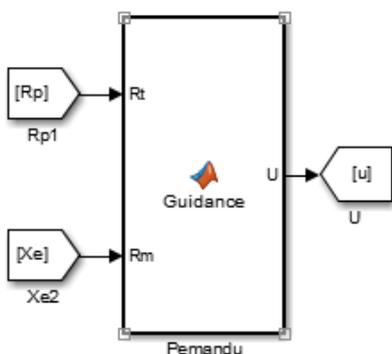
Persamaan (3.14) dan Persamaan (3.15) ini mengalkulasi invers trigonometri dari 2 buah nilai. Pada perangkat lunak MATLAB, fungsi invers trigonometri memiliki batas keluaran. Invers trigonometri dari fungsi *tangent* pada MATLAB memiliki nama *atan2(a,b)* di mana *a* adalah nilai *numerator* dan *b* adalah nilai *denominator*. Fungsi ini memiliki batas keluaran $[-\pi, \pi]$. Visualisasi batasan ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 Batas keluaran fungsi *atan2(a,b)* pada MATLAB

Pada Persamaan (3.14), *denominator* fungsi invers trigonometri *tangent* memiliki bentuk akar kuadrat. Sehingga persamaan tersebut akan selalu mengeluarkan nilai positif dan batas keluaran persamaan tersebut akan menjadi $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Blok pemandu dirancang dengan menggunakan blok MATLAB *Function*. Sesuai dengan persamaan yang diperlukan, blok ini memiliki masukan posisi sasaran dan posisi rudal saat ini. Keluarannya berupa sudut relatif antara rudal – sasaran. Keluaran ini yang menjadi masukan pada blok LQT sebelumnya. Blok pemandu ini diberi label “Pemandu”. Blok Simulink pemandu yang telah dirancang dapat dilihat pada Gambar 3.9.



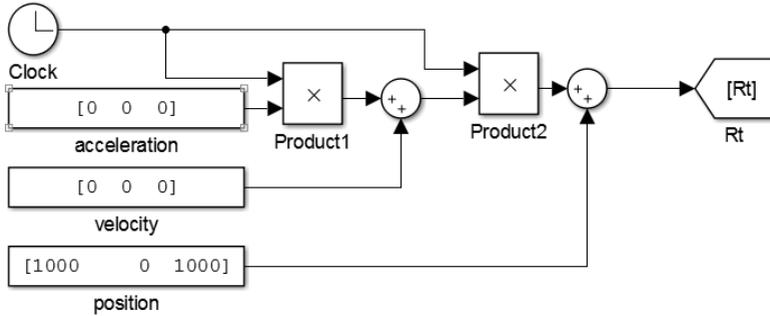
Gambar 3.9 Blok Simulink pemandu rudal

3.6 Perancangan Pergerakan Sasaran

Sasaran rudal dianggap berupa sebuah titik yang berada dalam ruang 3 dimensi. Posisi sasaran dalam ruang 3 dimensi ini dipengaruhi posisi, kecepatan, dan percepatan awal sasaran. Jika kecepatan awal dan percepatan awal sasaran bernilai nol, maka sasaran tidak bergerak, dan sebaliknya.

Pergerakan sasaran ini dibuat dengan menggunakan blok *Math Functions*. Informasi yang diperlukan sasaran pada kondisi awal adalah posisi, kecepatan, dan percepatan awal sasaran. Lalu dengan serangkaian operasi matematika, bisa didapatkan posisi sasaran yang dapat berubah terhadap waktu dalam ruang 3 dimensi. Terdapat 9 nilai informasi yang dibutuhkan sebagai masukan pergerakan sasaran dikarenakan sasaran berada dalam 3 dimensi. Keluaran dari serangkaian blok matematika ini

berupa posisi sasaran dalam ruang 3 dimensi. Keluaran ini dihubungkan langsung dengan pemandu. Blok Simulink pergerakan sasaran yang telah dirancang dapat dilihat pada Gambar 3.10.



Gambar 3.10 Blok Simulink pergerakan sasaran

3.7 Perancangan Prediksi Posisi Sasaran

Seiring dengan mengecilnya posisi relatif rudal – sasaran, perubahan sudut LOS antara rudal – sasaran juga membesar. Orientasi rudal dikendalikan oleh sirip rudal. Semakin besar sudut sirip yang dibentuk, kecepatan sudut orientasi rudal akan membesar. Karena sudut sirip rudal memiliki nilai saturasi, rudal memiliki batas kecepatan sudut yang dapat dihasilkan. Suatu saat akan ada di mana perubahan kecepatan sudut orientasi rudal tidak dapat mengikuti perubahan sudut LOS, sehingga terdapat kemungkinan tembakan rudal meleset. Oleh karena itu, dirancang suatu sistem yang dapat memprediksi posisi sasaran di saat yang akan datang.

Prediksi posisi sasaran ini menggunakan informasi 3 posisi sasaran terakhir. Dari 3 informasi ini akan ditentukan posisi, kecepatan, dan percepatan sasaran 2 waktu *sample* sebelumnya. Penentuan posisi, kecepatan, dan percepatan sasaran ini dinyatakan dalam Persamaan (3.16).

$$\begin{bmatrix} p \\ v \\ a \end{bmatrix} = \begin{bmatrix} 1 & t_s & 1t_s^2 \\ 1 & 2t_s & 4t_s^2 \\ 1 & 3t_s & 9t_s^2 \end{bmatrix}^{-1} \begin{bmatrix} x(k-2) \\ x(k-1) \\ x(k) \end{bmatrix} \quad (3.16)$$

Di mana:

- p = Estimasi posisi sasaran saat ini
- v = Estimasi kecepatan sasaran saat ini
- a = Estimasi percepatan sasaran saat ini

- $x(k)$ = Informasi sasaran terbaru
- $x(k - 1)$ = Informasi sasaran 1 *sample* yang lalu
- $x(k - 2)$ = Informasi sasaran 2 *sample* yang lalu

Prediksi posisi sasaran 1 waktu *sample* mendatang dinyatakan dalam Persamaan (3.17).

$$x(k + 1) = p + 4vt_s + 16at_s^2 \quad (3.17)$$

Di mana:

- $x(k + 1)$ = Prediksi posisi 1 *sample* mendatang
- t_s = *Sampling time*

Time-to-go, yaitu waktu yang dibutuhkan rudal agar rudal dapat mencapai sasaran, bisa didapatkan dengan diketahuinya posisi relatif antara rudal – sasaran dan *closing velocity* antara rudal - sasaran dengan menggunakan Persamaan (3.18).

$$t_{go} = \frac{\sqrt{x_r^2 + y_r^2 + z_r^2}}{\sqrt{(v_{tx} - v_{mx})^2 + (v_{ty} - v_{my})^2 + (v_{tz} - v_{mz})^2}} \quad (3.18)$$

Di mana:

- x_r = Posisi relatif rudal – sasaran pada sumbu *x* bumi
- y_r = Posisi relatif rudal – sasaran pada sumbu *y* bumi
- z_r = Posisi relatif rudal – sasaran pada sumbu *z* bumi
- v_{tx} = Kecepatan sasaran pada sumbu *x* bumi
- v_{ty} = Kecepatan sasaran pada sumbu *y* bumi
- v_{tz} = Kecepatan sasaran pada sumbu *z* bumi
- v_{mx} = Kecepatan rudal pada sumbu *x* bumi
- v_{my} = Kecepatan rudal pada sumbu *y* bumi
- v_{mz} = Kecepatan rudal pada sumbu *z* bumi

Posisi sasaran saat t_{go} detik dari posisi sasaran saat 2 waktu *sample* sebelumnya didapatkan dengan menggunakan Persamaan (3.19).

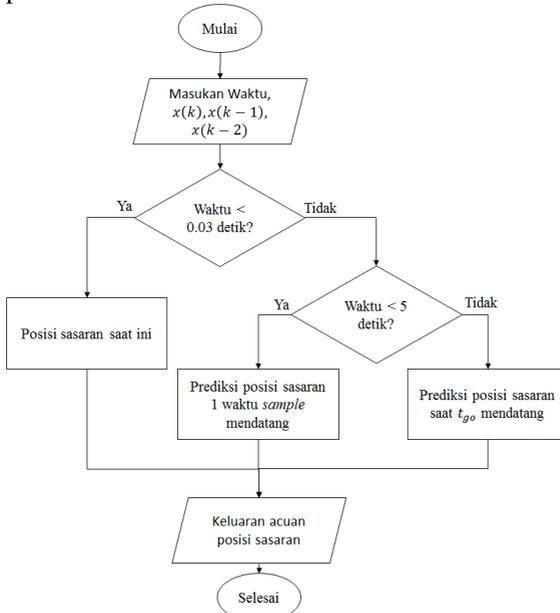
$$x(t_{go}) = p + v(t_{go}) + a(t_{go})^2 \quad (3.19)$$

Perancangan blok prediksi posisi sasaran menggunakan blok *MATLAB Function*. Masukan yang diperlukan yaitu 3 informasi posisi sasaran sebelumnya. Ketiga informasi berasal dari posisi sasaran yang melewati blok *delay* sebanyak 3 kali; 1 untuk *sampling* dan 2 lainnya

untuk menunda. Keluaran blok ini adalah prediksi posisi sasaran yang akan diteruskan menuju blok pemandu rudal. Waktu *sample* yang digunakan sebesar 0.01 detik.

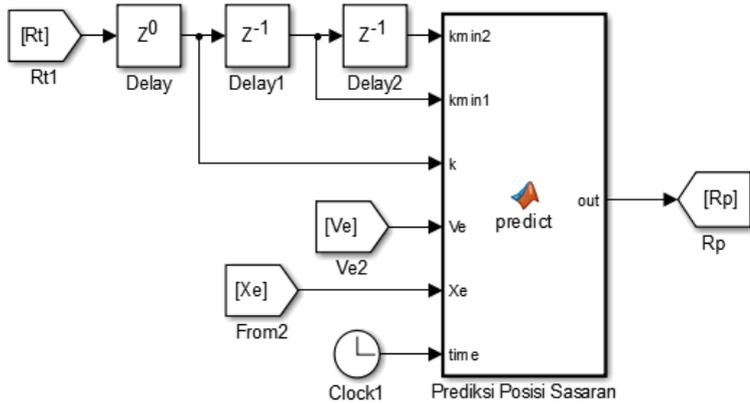
Sesaat rudal baru diluncurkan, sistem tidak dapat menentukan posisi, kecepatan, dan percepatan sasaran karena rudal memerlukan setidaknya 3 informasi sasaran. Di sisi lain, sesaat 3 informasi posisi sasaran dimiliki, rudal belum memiliki kecepatan yang besar, sehingga *closing velocity* akan bernilai kecil dan t_{go} akan sangat besar. Oleh karena itu, diperlukan strategi untuk menghasilkan prediksi posisi sasaran yang konsisten agar gerakan rudal juga konsisten saat rudal baru diluncurkan.

Strategi yang diusulkan yaitu sesaat rudal diluncurkan, blok ini mengeluarkan posisi sasaran saat ini. Lalu setelah 3 informasi posisi sasaran dimiliki, atau saat 0.03 detik setelah peluncuran, blok ini mengeluarkan hasil prediksi posisi sasaran 1 waktu *sample* mendatang. Kemudian setelah *Thrust* rudal memasuki fase *sustain*, atau 5 detik setelah peluncuran, blok ini mengeluarkan hasil prediksi posisi sasaran saat t_{go} detik dari saat ini. Keluaran blok ini akan diteruskan menuju pemandu rudal.



Gambar 3.11 Diagram alir prediksi posisi sasaran

Strategi yang diusulkan ini dapat dilihat pada Gambar 3.11. Sedangkan blok Simulink prediksi posisi sasaran yang telah dirancang dapat dilihat pada Gambar 3.12.



Gambar 3.12 Blok Simulink untuk memprediksi posisi sasaran

Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN ANALISA

Bab ini menjelaskan pengujian dari masing masing blok yang telah dirancang dan hasil simulasi dari pergerakan rudal menuju sasaran. Komponen yang dilakukan pengujian adalah model dinamika rudal, *decoupler* pada rudal, pengendali LQT, pemandu rudal, pergerakan sasaran, serta prediksi posisi sasaran.

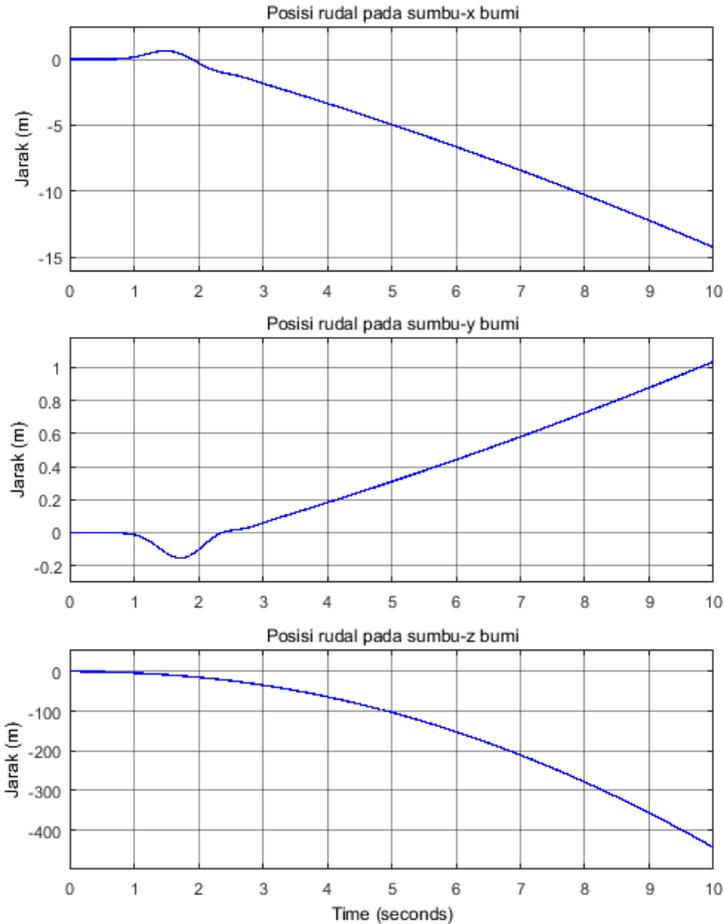
4.1 Gambaran Umum Pengujian Simulasi Rudal

Sebelum simulasi pengejaran rudal menuju sasaran dilakukan, komponen simulasi yang telah dirancang sebelumnya dilakukan pengujian dahulu. Tahapan pengujian yang dilakukan adalah sebagai berikut.

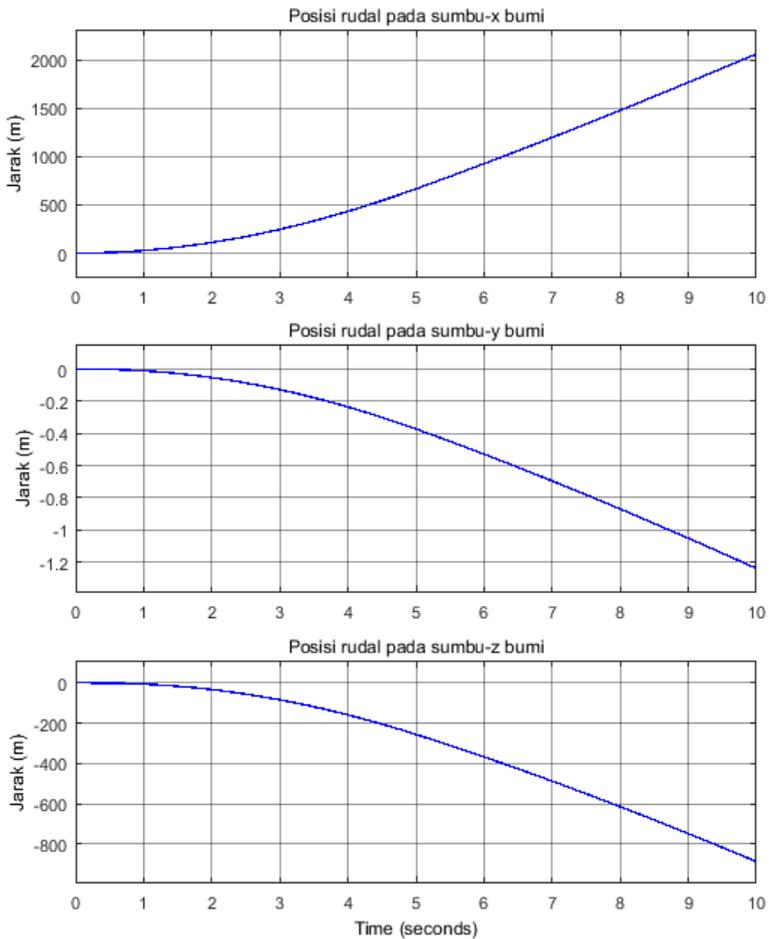
1. Pengujian model dinamika rudal
Pengujian pada model dinamika rudal dilakukan dengan mengadakan dan meniadakan *Thrust* pada rudal.
2. Pengujian *non-linear state feedback decoupler*
Pengujian dilakukan dengan menghubungkan blok Simulink ini dengan model dinamika rudal. Lalu blok Simulink diberi suatu nilai referensi dan diamati sudut orientasi yang dibentuk rudal. Pengujian ini dilakukan untuk membuktikan bahwa interaksi antar persamaan dinamika rotasi rudal dapat hilang.
3. Pengujian pengendali LQT
Pengujian ini dilakukan dengan menghubungkan pengendali LQT dengan 2 komponen sebelumnya. Selanjutnya pengendali LQT diberi masukan referensi sinusoidal.
4. Pengujian pemandu rudal
Pengujian pemandu rudal dilakukan dengan menghubungkan pemandu dengan 3 komponen sebelumnya, lalu pemandu diberi masukan berupa posisi sasaran yang statis.
5. Pengujian pergerakan sasaran dan prediksi posisi sasaran
Dalam pengujian ini, dilakukan pengujian pergerakan sasaran dan prediksi posisi sasaran secara bersamaan. Pengujian pergerakan sasaran dilakukan dengan memberikan posisi, kecepatan, dan percepatan awal rudal yang berbeda – beda lalu diamati di manakah posisinya dalam selang waktu tertentu, serta pengujian prediksi posisi sasaran dilakukan dengan masukan dari pergerakan sasaran.

4.2 Pengujian Model Dinamika Rudal

Pada model dinamika rudal, pengujian dilakukan dengan mengadakan dan meniadakan *Thrust* pada rudal serta masukan rudal $\delta_a, \delta_e, \delta_r$ dijaga nol. Dengan tidak adanya *Thrust* dan masukan pada rudal, diharapkan rudal mengalami gerakan gerak jatuh bebas. Gerakan jatuh bebas ini dapat dilihat pada keluaran posisi rudal terhadap bumi, yaitu posisi rudal pada sumbu Z_e akan selalu turun. Lalu dengan adanya *Thrust* pada rudal, diharapkan rudal dapat bergerak maju.



Gambar 4.1 Posisi rudal terhadap bumi tanpa *Thrust*



Gambar 4.2 Pengujian model rudal dengan *Thrust*

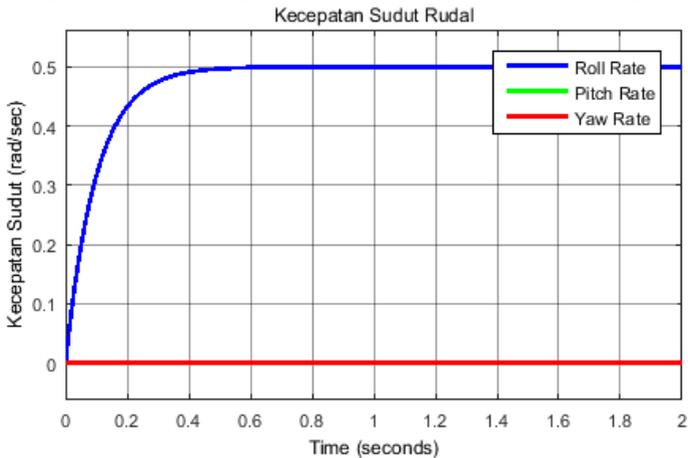
Pada pengujian model rudal tanpa *Thrust*, di mana dapat dilihat pada Gambar 4.1, pada sumbu X_e dan Y_e , perubahan posisi rudal tidak signifikan. Pada sumbu Z_e , terlihat rudal mengalami penurunan ketinggian. Sehingga rudal dapat dikatakan mengalami gerak jatuh bebas. Pada pengujian model rudal dengan *Thrust*, di mana dapat dilihat pada Gambar 4.2, terlihat rudal bergerak searah dengan sumbu X_e dan

bergerak jatuh pada sumbu Z_e . Dari pengujian model rudal tanpa *Thrust*, terlihat bahwa posisi rudal pada sumbu Y_e dan X_e mengalami perubahan. Menurut logika, gerak jatuh bebas hanya akan mempengaruhi ketinggian suatu benda. Sehingga pada model rudal yang telah dibuat terdapat interaksi antara persamaan dinamika rudal.

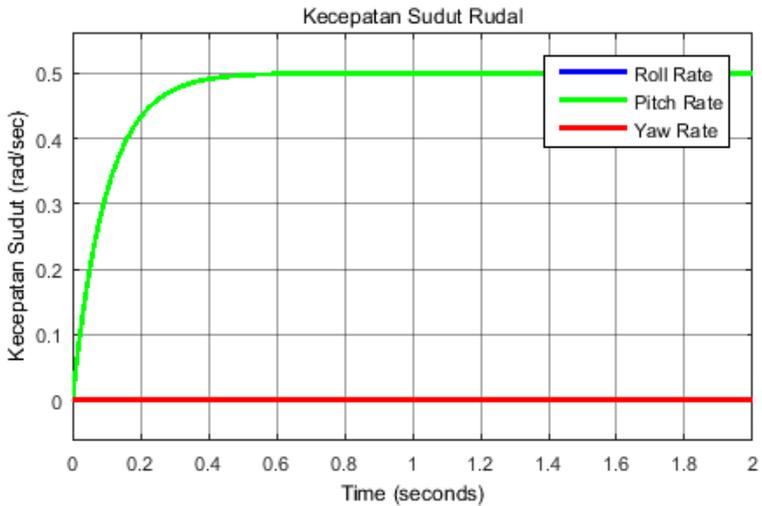
4.3 Pengujian *Non-linear State Feedback Decoupler*

Tujuan dari pengujian ini adalah untuk memastikan bahwa melalui *non-linear state feedback*, model dinamika rotasi rudal dapat dilakukan linearisasi. Selain itu, tujuan lainnya yaitu memastikan hubungan interaksi antar persamaan dinamika rotasi rudal dapat dihilangkan dengan proses *decoupling*. Nilai masukan dari blok *decoupler* ini adalah kecepatan sudut. Sehingga parameter keluaran dari model rudal yang perlu diperhatikan adalah kecepatan sudut rudal. Pada pengujian *decoupler* ini, *Thrust* rudal digunakan.

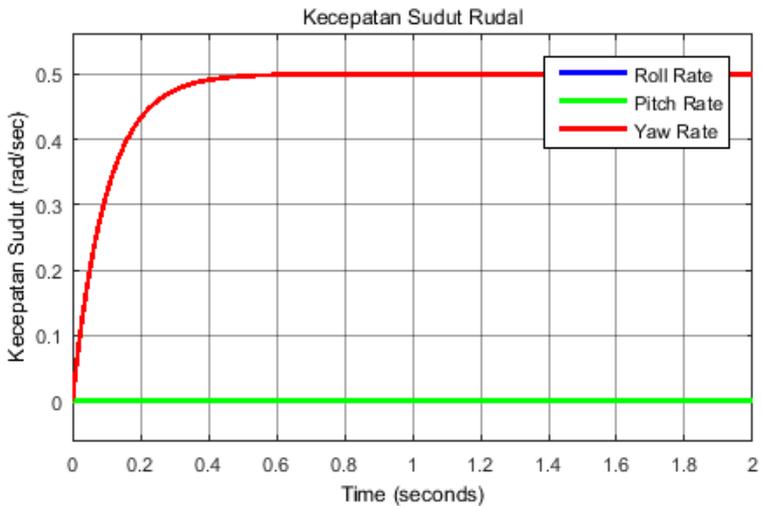
Pengujian ini dilakukan dengan menggabungkan blok model dinamika rudal dengan blok *decoupler*. Karena sudut orientasi rudal dapat berubah karena adanya momen aerodinamika, dan momen aerodinamika sendiri dipengaruhi oleh besarnya magnitudo kecepatan rudal, maka dalam pengujian ini, rudal diatur agar memiliki kecepatan awal sesuai dengan kecepatan nominalnya. Pengujian dilakukan sebanyak 3 kali mengingat pada rudal terdapat 3 sumbu rotasi. Masukan yang diberikan berupa sinyal *step* dengan nilai *set point* sebesar 0.5 radian per detik.



Gambar 4.3 Respon kecepatan sudut rudal dengan masukan [0.5 0 0]



Gambar 4.4 Respon kecepatan sudut rudal dengan masukan [0 0.5 0]



Gambar 4.5 Respon kecepatan sudut rudal dengan masukan [0 0 0.5]

Dari hasil respon pengujian *decoupling* untuk kecepatan sudut *roll* (Gambar 4.3), *pitch* (Gambar 4.4), dan *yaw* (Gambar 4.5), dicari nilai *time*

constant dari setiap respon dengan menggunakan fitur *cursor measurements* yang ada pada Simulink. Nilai *time constant* dari setiap respon disajikan dalam Tabel 4.1.

Tabel 4.1 Karakteristik respon kecepatan sudut rudal

Karakteristik	Kecepatan Sudut		
	<i>Roll</i>	<i>Pitch</i>	<i>Yaw</i>
<i>Time constant</i> (detik)	0.1	0.1	0.1
<i>Offset Error</i> (%)	0	0	0

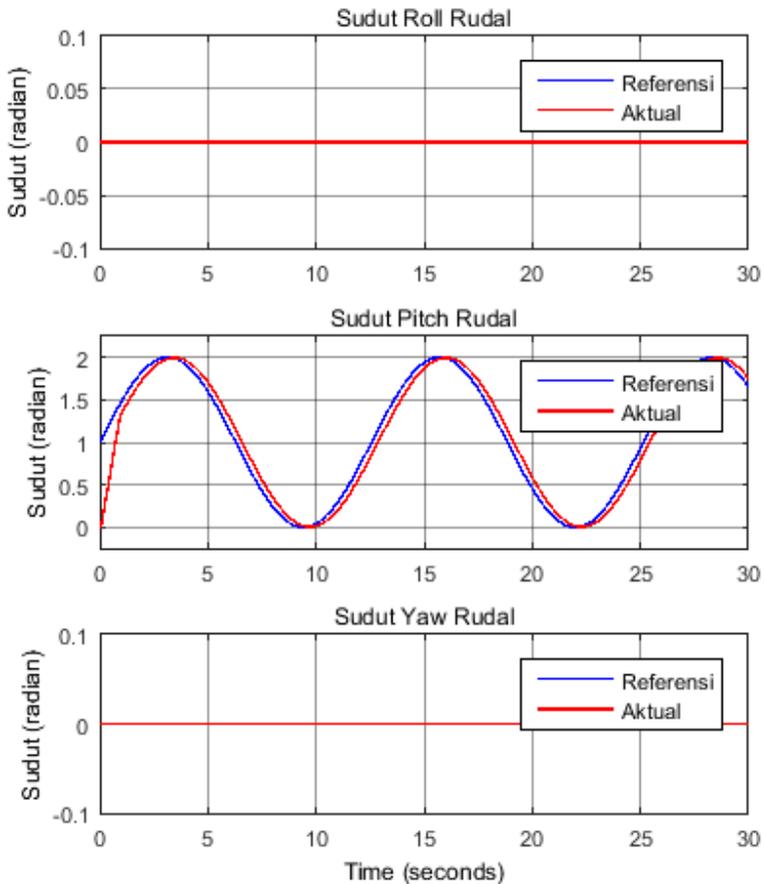
Dari Tabel 4.1 ini, terlihat bahwa *time constant* kecepatan sudut rudal sesuai dengan nilai *time constant* yang telah diinginkan. Sehingga blok decoupler ini bekerja dengan baik. Namun perlu diketahui bahwa sirip rudal memiliki nilai minimal dan maksimal besar sudut defleksi yang dapat dibentuknya, yakni $[60 -60]$ derajat. Sehingga pasti terdapat nilai maksimal kecepatan sudut yang dapat dilakukan oleh rudal.

4.4 Pengujian Pengendali LQT

Pengujian ini bertujuan untuk memeriksa bahwa pengendali LQT yang telah dirancang dapat bekerja dengan baik. Pengendali LQT menerima masukan berasal dari pemandu rudal. Masukan dari pengendali LQT ini adalah sudut referensi yang diinginkan pemandu dalam satuan radian.

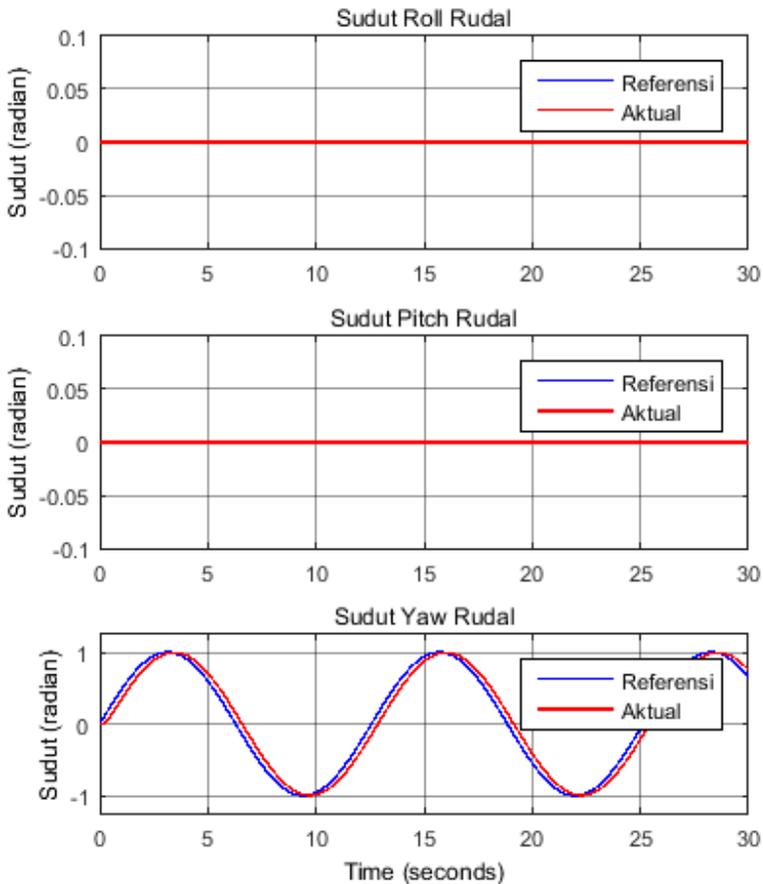
Karena variabel yang dikendalikan adalah sudut orientasi rudal, maka terdapat 3 masukan pada pengendali LQT ini, yaitu sudut referensi *roll*, sudut referensi *pitch*, dan sudut referensi *yaw*. Pengujian pengendali LQT ini dilakukan dengan menggunakan masukan berupa sinyal sinusoidal dengan frekuensi 0.5 radian per detik. Khusus masukan sudut referensi *roll*, masukan berupa nilai konstan nol karena pada rudal jenis *Skid-to-Turn*, rudal dijaga agar tidak melakukan gerak rotasi *roll*. Sehingga terdapat 2 pengujian, yaitu pengujian untuk pengendalian sudut *pitch* rudal dan pengendalian sudut *yaw* rudal.

Dari Gambar 4.6, sudut *pitch* rudal baru dapat mengejar referensi yang ada setelah 1.91 detik. Ini dikarenakan referensi awal yang diberikan mulai dari 1 radian, sedangkan sudut awal *pitch* rudal adalah 0 radian. Setelah itu, sudut *pitch* rudal dapat mengikuti referensi yang diberikan. Sehingga pengendali LQT untuk sudut *pitch* rudal dapat dikatakan bekerja dengan baik.



Gambar 4.6 Pengujian pengendalian sudut *pitch* rudal dengan masukan sinusoidal

Lalu pada respon sudut *yaw* rudal, yakni Gambar 4.7, sudut *yaw* rudal dapat mengikuti referensi sinusoidal yang diberikan, sehingga pengendali LQT untuk sudut *yaw* rudal dapat dikatakan bekerja dengan baik. Perlu diketahui juga, bahwa dengan adanya batas minimal dan maksimal sudut defleksi sirip rudal, perubahan sudut orientasi rudal dapat terbatas juga pada nilai tertentu.

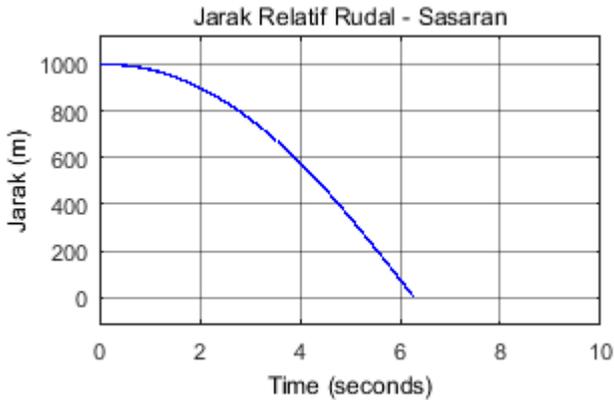


Gambar 4.7 Pengujian pengendalian sudut yaw dengan masukan sinusoidal

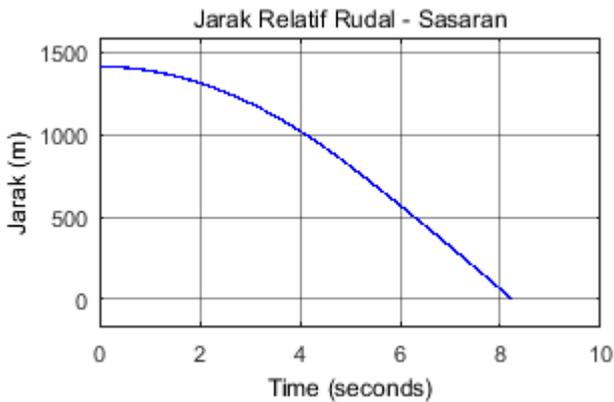
4.5 Pengujian Pemandu Rudal

Pengujian pemandu rudal dilakukan untuk memeriksa bahwa pemandu dapat memberikan sudut relatif rudal – sasaran, sehingga rudal bergerak menuju sasaran dengan baik. Pengujian dilakukan dengan menghubungkan keluaran pemandu dengan pengendali LQT. Masukan pada pemandu adalah posisi sasaran yang statis. Pengujian dilakukan sebanyak 4 kali, yaitu saat sasaran berada pada posisi [1000 0 0], posisi

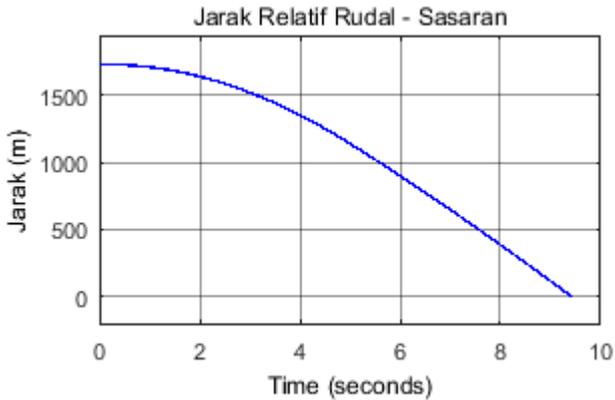
[1000 0 1000], posisi [1000 1000 1000], dan posisi [1500 1400 1300]. Lalu sudut orientasi awal rudal ($[\phi_0 \theta_0 \psi_0]$) diberikan masing – masing sebesar [0 45 0] derajat. Simulasi dihentikan saat rudal telah mencapai posisi terdekatnya dengan sasaran. Hal yang diamati pada pengujian ini adalah jarak relatif antara rudal – sasaran. Hasil pengujian ini dapat dilihat pada Gambar 4.8, Gambar 4.9, Gambar 4.10, dan Gambar 4.11.



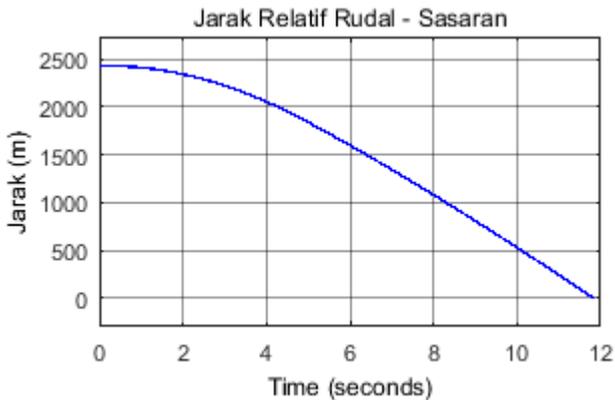
Gambar 4.8 Jarak relatif rudal - sasaran saat posisi sasaran [1000 0 0]



Gambar 4.9 Jarak relatif rudal - sasaran saat posisi sasaran [1000 0 1000]



Gambar 4.10 Jarak relatif rudal – sasaran saat posisi sasaran [1000 1000 1000]



Gambar 4.11 Jarak relatif rudal – sasaran saat posisi sasaran [1500 1400 1300]

Dari hasil pengujian ini, parameter yang diamati adalah jarak terdekat antara rudal-sasaran yang dibentuk dan durasi perjalanan rudal menuju sasaran. Parameter ini dapat dilihat pada Tabel 4.2. Dari keempat pengujian ini, *miss distance* rudal tidak melebihi 1 meter. Sehingga pemandu rudal dapat dikatakan bekerja dengan baik dengan sasaran statis.

Tabel 4.2 Hasil pengujian pemandu rudal

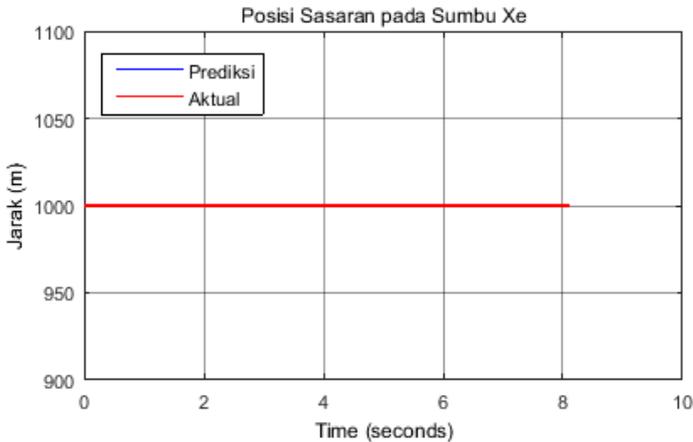
Posisi Sasaran [X_e Y_e Z_e]	Durasi Perjalanan (detik)	Jarak Terdekat (meter)
[1000 0 0]	6.28	0.0506
[1000 0 1000]	8.23	0.0440
[1000 1000 1000]	9.42	0.3770
[1500 1400 1300]	11.80	0.3167
Rata-rata	8.93	0.1971

4.6 Pengujian Prediksi Posisi Sasaran

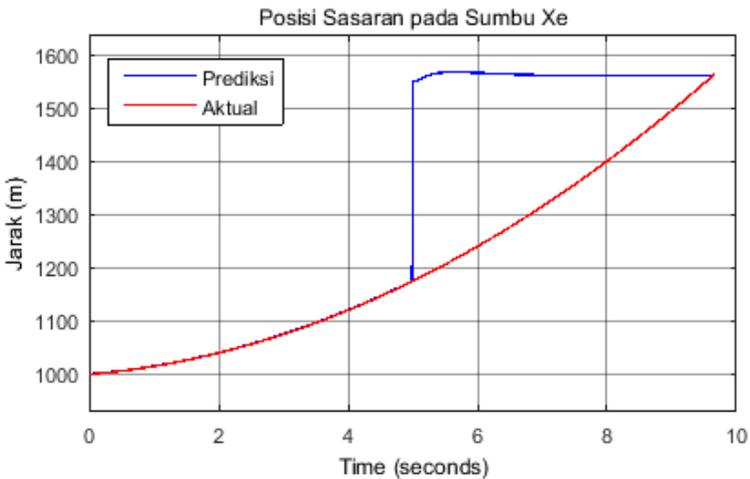
Pengujian ini bertujuan untuk menguji strategi prediksi posisi sasaran yang diajukan apakah bekerja dengan baik atau tidak. Pengujian prediksi posisi sasaran dimulai dengan menunda informasi posisi sasaran saat ini sebanyak 2 kali. Masing – masing informasi posisi sasaran yang tertunda masuk ke dalam blok Simulink prediksi posisi sasaran. Terdapat 3 informasi yang masuk ke dalam blok prediksi ini, yaitu informasi posisi sasaran saat ini, informasi posisi sasaran 1 waktu *sample* sebelumnya, dan informasi posisi sasaran 2 waktu *sample* sebelumnya. Keluaran dari blok ini adalah hasil prediksi posisi sasaran di waktu mendatang dengan menggunakan 3 informasi yang masuk ini.

Pengujian blok tidak dapat dilakukan secara mandiri dikarenakan blok ini memerlukan masukan berupa posisi dan kecepatan rudal pada koordinat bumi, sehingga blok ini memerlukan blok model rudal. Namun, jika blok model rudal tidak dikendalikan, hasil prediksi akan kacau, sehingga diperlukan blok autopilot dan blok pemandu. Dengan demikian, secara tidak langsung pengujian ini adalah simulasi pengejaran rudal – sasaran yang akan dilakukan selanjutnya.

Pengujian dilakukan sebanyak 2 kali. Pertama dengan menggunakan sasaran statis dan kedua dengan sasaran dinamis. Posisi sasaran statis yang digunakan adalah [1000 0 1000]. Kondisi awal untuk sasaran dinamis yang digunakan adalah posisi [1000 0 1000], kecepatan [10 0 0], dan percepatan [5 0 0]. Parameter yang diperhatikan adalah posisi hasil prediksi dengan posisi sasaran sebenarnya pada salah satu sumbu. Parameter yang diamati dari pengujian ini adalah hasil prediksi posisi sasaran mendatang dengan posisi sasaran sebenarnya. Hasil pengujian dengan sasaran statis dapat dilihat pada Gambar 4.12, sedangkan pengujian dengan sasaran dinamis dapat dilihat pada Gambar 4.13.



Gambar 4.12 Pengujian prediksi posisi sasaran dengan sasaran statis



Gambar 4.13 Pengujian prediksi posisi sasaran dengan sasaran dinamis

Dari hasil pengujian dengan sasaran statis, yakni dari Gambar 4.12, hasil prediksi posisi sasaran berimpit dengan posisi sasaran sebenarnya. Ini dikarenakan posisi sasaran tidak berubah dari waktu ke waktu, sehingga hasil prediksi akan sama dengan posisi sasaran saat ini. Pada

pengujian dengan sasaran dinamis, yakni dari Gambar 4.13, dengan pertimbangan *closing velocity* dan *time-to-go* rudal, rudal dapat memprediksi di mana posisi akan terjadinya *impact* dengan sasaran, sehingga referensi yang masuk ke dalam pemandu adalah posisi di mana *impact* akan terjadi. Alhasil rudal bergerak menuju titik di mana *impact* akan terjadi. Pada waktu 9.94 detik, simulasi berhenti dikarenakan rudal mencapai jarak terdekatnya dengan sasaran, yakni sebesar 1.89 meter. Oleh karena itu, prediksi posisi sasaran di saat mendatang dapat dikatakan bekerja dengan baik.

4.7 Simulasi Pengejaran Rudal – Sasaran

Setelah semua komponen simulasi telah diuji, selanjutnya mensimulasikan pengejaran rudal menuju sasaran. Simulasi dilakukan dengan menggunakan 2 jenis sasaran; sasaran statis dan sasaran dinamis. Sasaran statis yaitu sasaran yang tidak berpindah posisi dari posisi awalnya. Dengan kata lain, sasaran jenis ini diam di tempat. Sedangkan sasaran dinamis yaitu sasaran yang memiliki kecepatan maupun percepatan. Dengan kata lain, posisi sasaran jenis ini bergerak seiring dengan bertambahnya waktu. Keluaran yang diamati dari simulasi ini adalah jarak terdekat relatif antara rudal – sasaran.

4.7.1 Sasaran Statis

Pada simulasi dengan sasaran statis, simulasi dilakukan sebanyak 9 skenario dengan posisi sasaran yang berbeda – beda. Simulasi ini dilakukan untuk melihat bagaimana posisi sasaran mempengaruhi jarak terdekat yang dapat dibuat oleh rudal ke sasaran.

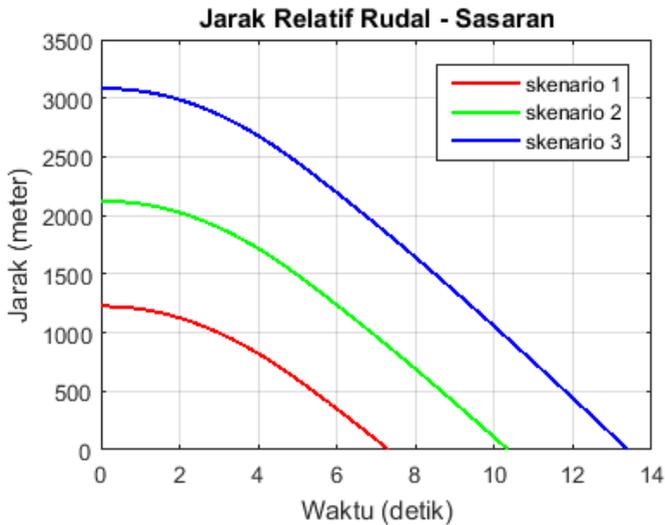
Skenario 1, 2, dan 3 menunjukkan posisi sasaran pada sumbu X_e yang berbeda – beda, namun posisi pada sumbu Y_e dan Z_e tetap. Ketiga skenario ini dikategorikan dalam kasus 1. Skenario 4, 5, dan 6 menunjukkan posisi sasaran pada sumbu Y_e yang berbeda – beda, namun posisi pada sumbu X_e dan Z_e yang tetap. Ketiga skenario ini dikategorikan sebagai kasus 2. Skenario 7, 8, dan 9 menunjukkan posisi sasaran pada sumbu Z_e yang berbeda – beda, namun posisi pada sumbu Y_e dan Z_e yang tetap. Ketiga skenario ini dikategorikan sebagai kasus 3.

Dalam simulasi dengan sasaran statis ini, semua sudut orientasi awal rudal adalah [0 45 45] derajat. Simulasi dihentikan saat rudal mencapai jarak terdekatnya dengan sasaran. Posisi awal sasaran disajikan dalam Tabel 4.3.

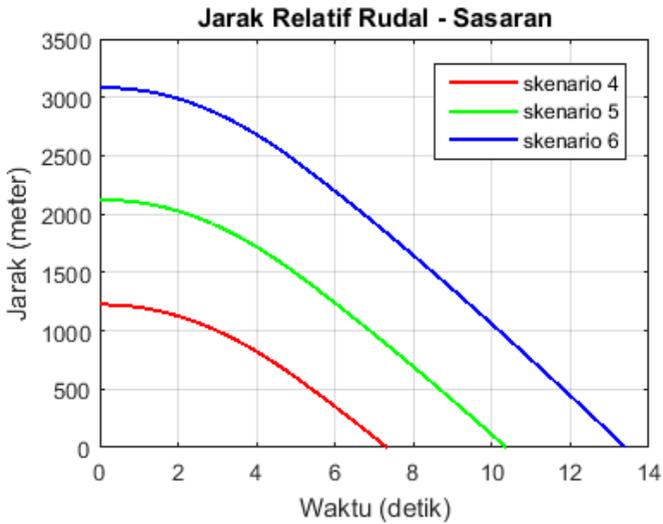
Tabel 4.3 Posisi awal sasaran pada simulasi sasaran statis

Skenario ke-	Posisi Sasaran [X_e Y_e Z_e]	Keterangan
1	[1000 500 500]	Posisi pada X_e berubah, Y_e dan Z_e tetap
2	[2000 500 500]	
3	[3000 500 500]	
4	[500 1000 500]	Posisi pada Y_e berubah, X_e dan Z_e tetap
5	[500 2000 500]	
6	[500 3000 500]	
7	[500 500 1000]	Posisi pada Z_e berubah, X_e dan Y_e tetap
8	[500 500 2000]	
9	[500 500 3000]	

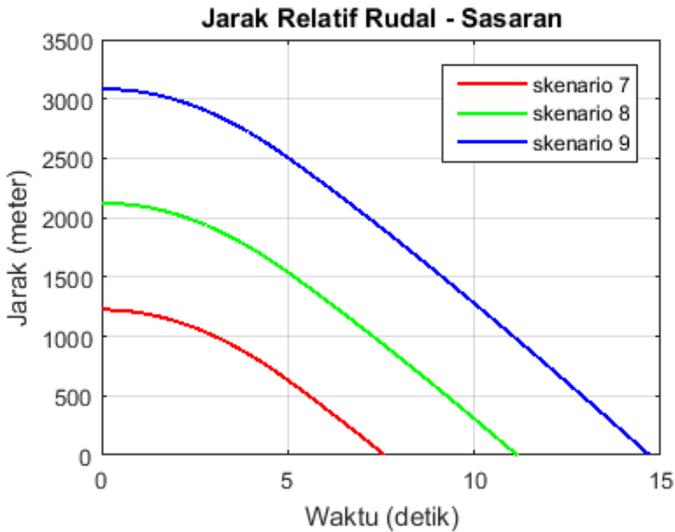
Hasil dari simulasi kasus 1 diberikan pada Gambar 4.14. Hasil simulasi kasus 2 diberikan pada Gambar 4.15. Hasil simulasi kasus 3 diberikan pada Gambar 4.16.



Gambar 4.14 Hasil simulasi dengan sasaran statis pada skenario 1, 2, dan 3



Gambar 4.15 Hasil simulasi dengan sasaran statis pada skenario 4,5, dan 6



Gambar 4.16 Hasil simulasi dengan sasaran statis pada skenario 7,8, dan 9

Jarak terdekat relatif antara rudal – sasaran serta durasi perjalanan rudal pada masing – masing skenario dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil simulasi dengan sasaran statis

Skenario	Durasi Perjalanan (detik)	Jarak Terdekat (meter)
1	7.32	0.3655
2	10.35	0.6177
3	13.40	0.2676
4	7.33	0.6534
5	10.37	1.2010
6	13.41	0.0429
7	7.60	0.4572
8	11.15	0.0937
9	14.70	0.3542
Rata-rata	10.63	0.4504

Dari hasil simulasi dengan sasaran statis ini, yaitu pada Tabel 4.4, grafik yang dihasilkan cenderung mirip antara satu dengan lainnya. Ini dikarenakan jarak menjauhnya posisi sasaran pada setiap percobaan adalah sama besar. Namun yang pasti, seiring dengan menjauhnya posisi sasaran, durasi perjalanan rudal menuju sasaran juga membesar. Didapatkan juga rata – rata jarak terdekat yang dapat dibentuk antara rudal dengan sasaran yaitu sebesar 0.794 meter.

4.7.2 Sasaran Dinamis

Pada simulasi dengan sasaran dinamis, simulasi dilakukan sebanyak 6 skenario dengan kondisi awal sasaran yang berbeda – beda. Orientasi awal rudal yang digunakan adalah [0 60 0] derajat.

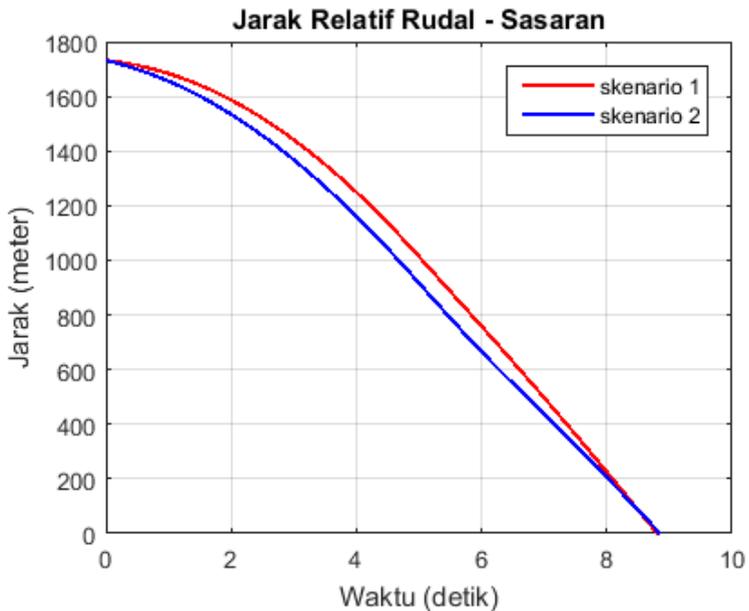
Kasus 1, yaitu skenario 1 dan 2, adalah simulasi dengan sasaran memiliki kecepatan awal yang berbeda. Kasus 2, yaitu skenario 3 dan 4, adalah simulasi dengan sasaran memiliki percepatan berbeda. Dan kasus 3, yaitu skenario 5 dan 6, adalah simulasi dengan sasaran yang bergerak berputar dan menjauhi rudal. Kondisi awal dari simulasi dengan sasaran dinamis ini dapat dilihat pada

Tabel 4.5.

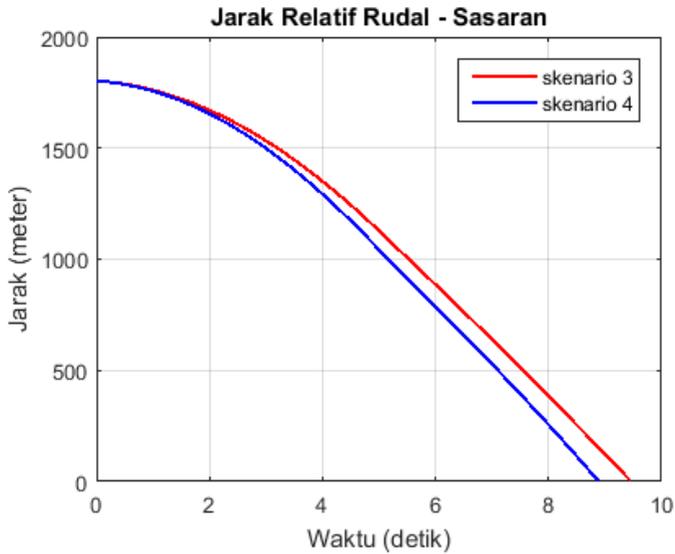
Tabel 4.5 Kondisi awal sasaran dinamis

Skenario	Posisi Awal (m)	Kecepatan Awal (m/s)	Percepatan Awal (m/s ²)
1	[1000 1000 1000]	[0 -50 0]	[0 0 0]
2	[1000 1000 1000]	[0 -100 0]	[0 0 0]
3	[1500 0000 1000]	[-50 0 40]	[0 0 0]
4	[1500 0000 1000]	[-50 0 40]	[-5 0 0]
5	[1000 1000 1000]	[-50 -50 30]	[0 0 0]
6	[1000 1000 1000]	[-50 -50 30]	[8 8 0]

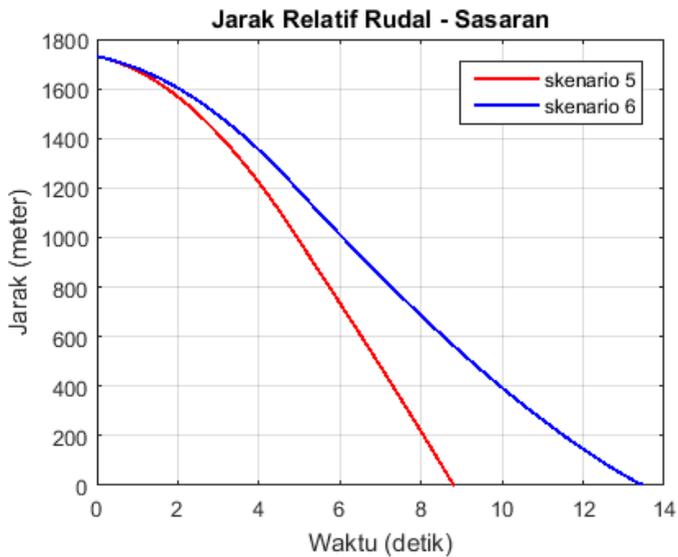
Hasil dari simulasi dengan sasaran dinamis ini dengan kasus 1, kasus 2, dan kasus 3 masing – masing dapat dilihat dapat dilihat pada Gambar 4.17, Gambar 4.18, dan Gambar 4.19.



Gambar 4.17 Hasil simulasi sasaran dinamis pada skenario 1 dan 2



Gambar 4.18 Hasil simulasi sasaran dinamis pada skenario 3 dan 4



Gambar 4.19 Hasil simulasi sasaran dinamis pada skenario 5 dan 6

Jarak terdekat relatif antara rudal – sasaran yang dibentuk serta durasi perjalanan rudal menuju sasaran pada masing – masing skenario dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil simulasi dengan sasaran dinamis

Skenario	Durasi (detik)	Jarak Terdekat (meter)
1	8.81	1.4060
2	8.85	0.6871
3	9.46	2.1270
4	8.89	4.3470
5	8.81	2.7220
6	13.47	4.0820
Rata-rata	9.72	2.5619

Dari hasil simulasi dengan sasaran dinamis ini, dalam skenario 1 dan 2, rudal bergerak menuju sasaran dengan baik. Hal ini dapat dilihat pada jarak terdekat antara rudal – sasaran sebesar 1.406 meter pada skenario 1 dan 0.6871 meter pada skenario 2. Dalam skenario 3 dan 4, rudal juga bergerak menuju sasaran dengan baik. Hal ini dapat dilihat pada jarak terdekat antara rudal – sasaran sebesar 2.127 meter pada skenario 3 dan 4.347 meter pada skenario 4. Dalam skenario 5 dan 6, rudal bergerak menuju sasaran dengan jarak terdekat antara rudal – sasaran yang dibentuk adalah 2.722 meter pada skenario 5 dan 4.082 meter pada skenario 6.

Hal menarik yang dapat diperhatikan ada pada skenario 6 di mana sasaran bergerak berputar menghindari rudal dan melakukan akselerasi. Durasi perjalanan rudal menuju sasaran relatif lebih lama dibandingkan dengan skenario lainnya. Ini dikarenakan sasaran bergerak menjauhi rudal dengan melakukan akselerasi. Namun sasaran masih dapat dicegat oleh rudal.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

Bab ini berisi kesimpulan dari simulasi yang telah dijalankan serta saran pengembangan yang dapat dilakukan berdasarkan simulasi rudal yang telah dilakukan ini.

5.1 Kesimpulan

Dari simulasi pengejaran rudal menuju sasaran, dapat diambil beberapa kesimpulan sebagai berikut.

- a. Persamaan dinamika rotasi rudal yang *non-linear* dapat dilinearisasi dan *decoupled* dengan menggunakan *Nonlinear State Feedback Decoupler*. Hal ini ditunjukkan dengan respon perubahan kecepatan sudut orientasi rudal sesuai dengan *time constant* yang diinginkan, yaitu 0.1 detik pada gerak rotasi *roll*, *pitch*, dan *yaw* rudal.
- b. Pengendali LQT yang telah dirancang bekerja dengan baik. Hal ini dapat dibuktikan pada saat pengujian pengendali LQT, sudut *pitch* dan *yaw* rudal dapat mengikuti referensi sinusoidal yang diberikan.
- c. Penggunaan pemandu pengejar pada simulasi rudal berhasil. Hal ini dibuktikan dengan rudal yang selalu bergerak menuju sasaran, dapat dilihat pada grafik pengujian pemandu, dan nilai rata – rata jarak terdekat rudal – sasaran sebesar 0.1971 meter.
- d. Prediksi posisi sasaran disaat waktu mendarang bekerja dengan baik. Hal ini dapat dilihat dari hasil pengujiannya. Pada sasaran statis, hasil prediksi berimpit dengan posisi sebenarnya. Sedangkan pada sasaran dinamis, dapat diprediksi posisi saat sasaran tercegat oleh rudal.
- e. Pada sasaran statis, simulasi rudal bekerja dengan baik. Hal ini dibuktikan dengan nilai rata – rata jarak terdekat antara rudal – sasaran adalah 0.4504 meter.
- f. Sedangkan pada sasaran dinamis, simulasi rudal bekerja dengan baik. Hal ini dapat dilihat pada nilai rata – rata jarak terdekat antara rudal – sasaran untuk sasaran dinamis sebesar 2.5619 meter.

5.2 Saran

Untuk pengembangan kedepannya, terdapat beberapa saran yang diberikan, antara lain:

- a. Melakukan simulasi dengan jenis pengendali atau pemandu yang berbeda.
- b. Melakukan simulasi dengan menimbang bahwa massa total rudal berkurang selama rudal terbang di udara.
- c. Melakukan simulasi dengan menggunakan koefisien aerodinamika yang lebih akurat sehingga hasil simulasi akan lebih akurat.

DAFTAR PUSTAKA

- [1] G. M. Siouris, *Missile Guidance and Control Systems*. Springer New York, 2010.
- [2] G. M. Siouris, *Aerospace Avionics Systems: A Modern Synthesis*. Academic Press, 1993.
- [3] B. Özkan, “Dynamic Modeling, Guidance, and Control of Homing Missiles,” Dec. 2017.
- [4] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*. American Institute of Aeronautics and Astronautics, 2007.
- [5] A. K. R. Effendi, M. Rameli, E. Iskandar, and M. Baihaqi, “Linearization and decoupling controller for quadruple tank,” in *2017 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2017, pp. 233–237.
- [6] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, Third. New Jersey: John Wiley & Sons, 2012.
- [7] W. Buzantowicz, “Matlab Script for 3D Visualization of Missile and Air Target Trajectories,” vol. 5, pp. 419–422, Sep. 2016.

Halaman ini sengaja dikosongkan

LAMPIRAN A

A1. Program Dinamika Rudal

```
function [Vb_dot,Rb_dot] = Dynamics(Vb,Ob,Thrust,Euler,delta)
%% Input
u = Vb(1);
v = Vb(2);
w = Vb(3);
p = Ob(1);
q = Ob(2);
r = Ob(3);
phi = Euler(1);
theta = Euler(2);
% psi = Euler(3);
delta_a = delta(1);
delta_e = delta(2);
delta_r = delta(3);

%% Data
m = 142.2; % massa
g = 9.8; % gravitasi
d = 0.118; % diameter rudal
S = 1.1711; % Luas permukaan misil (bentuk tabung)
rho = 1.2251; % Kepadatan udara

Vm = sqrt(u^2+v^2+w^2);
mach = Vm/340; % magnitudo rudal dalam mach
alpha = atan2(w,u); % Angle of Attack
beta = asin(v/Vm); % Sideslip Angle

% Momen Inersia
Ix = 0.2475;
Iy = 24.8004;
Iz = Iy;

%% Aerodinamika
% Koefisien Aerodinamika untuk Gerakan Translasi
Cx1 = -0.35;
Cx2 = -0.05;
```

```
Cy1 = -90/pi;  
Cy2 = -0.3;  
Cy3 = -3;
```

```
Cz1 = -90/pi;  
Cz2 = -0.3;  
Cz3 = -1.5;
```

```
% Koefisien Aerodinamika untuk Gerakan Rotasi
```

```
Cl1=-0.1;  
Cl2=-0.1;  
Cl3=-1;  
Cl4=5;
```

```
Cm1=-(120/pi);  
Cm2=-2;  
Cm3=2;
```

```
Cn1=(90/pi);  
Cn2=-9;  
Cn3=9;
```

```
% Koefisien untuk Gaya Aerodinamika
```

```
Cx = Cx1*alpha + Cx2*mach;  
Cy = Cy1*beta + 0*Cy2*mach + Cy3*delta_r;  
Cz = Cz1*alpha + 0*Cz2*mach + Cz3*delta_e;
```

```
% Dynamic Pressure
```

```
Q = 0.5*rho*Vm^2;
```

```
%% Update Gerakan Rudal
```

```
% Gerakan Translasi
```

```
du = Thrust/m - g*sin(theta) + Q*S*Cx/m + r*v - q*w;  
dv = g*sin(phi)*cos(theta) + Q*S*Cy/m + p*w - r*u;  
dw = g*cos(phi)*cos(theta) + Q*S*Cz/m + q*u - p*v;  
Vb_dot = [du dv dw]';
```

```
% Gerakan Rotasi
```

```
dp = (Q*S*d*(Cl1*alpha + Cl2*beta + Cl3*p + Cl4*delta_a) + (Iy-Iz)*q*r) / Ix;
```

```
dq = (Q*S*d*(Cm1*alpha + 0 + Cm2*q + Cm3*delta_e) + (Iz-Ix)*p*r) / Iy;
```

```
dr = (Q*S*d*(0 + Cn1*beta + Cn2*r + Cn3*delta_r) + (Ix-Iy)*p*q) / Iz;
```

```
Rb_dot = [dp dq dr]';
```

A2. Program Kinematika Rudal

```
function Ve = Kinematics(Vb,Euler)
```

```
%% Input dan Data
```

```
phi = Euler(1);
```

```
theta = Euler(2);
```

```
psi = Euler(3);
```

```
%% Transformasi Sistem Koordinat
```

```
% Kecepatan rudal terhadap bumi
```

```
C = [cos(theta)*cos(psi) sin(phi)*sin(theta)*cos(psi)-cos(phi)*sin(psi)  
cos(phi)*sin(theta)*cos(psi)+sin(phi)*sin(psi);  
cos(theta)*sin(psi) sin(phi)*sin(theta)*sin(psi)+cos(phi)*cos(psi)  
cos(phi)*sin(theta)*sin(psi)-sin(phi)*cos(psi);  
-sin(theta) sin(phi)*cos(theta)  
cos(phi)*cos(theta)];
```

```
Ve = C*Vb;
```

A3. Program Non-linear State Feedback Decoupler

```
function delta = Decoupler(Vb,Rb,U)
```

```
%% Input
```

```
% Velocity
```

```
u = Vb(1);
```

```
v = Vb(2);
```

```
w = Vb(3);
```

```
% Angular Rate
```

```
p = Rb(1);
```

```
q = Rb(2);
```

```
r = Rb(3);
```

```
% Desired control
```

```
Up = U(1);
```

```
Uq = U(2);
```

```
Ur = U(3);
```

```
%% Constant and Coefficient
```

```
d = 0.118; % diameter rudal
```

```
s = 1.1711; % Luas permukaan misil (bentuk tabung)
```

```
rho = 1.2251; % Air density
```

```
Ix = 0.2475; % Momen Inersia sumbu x
```

```
Iy = 24.8004; % Momen Inersia sumbu y
```

```
Iz = Iy; % Momen Inersia sumbu z
```

```
Vm = sqrt(u^2+v^2+w^2);
```

```
Q = 0.5*rho*Vm^2;
```

```
alpha = atan2(w,u);
```

```
beta = asin(v/Vm);
```

```
Cl1=-0.1;
```

```
Cl2=-0.1;
```

```
Cl3=-1;
```

```
Cl4=5;
```

```
Cm1=-(120/pi);
```

```
Cm2=-2;
```

```
Cm3=2;
```

```
Cn1=(90/pi);
```

```
Cn2=-9;
```

```
Cn3=9;
```

```
%New time constant
```

```
lamda_p = 10;
```

```
lamda_q = 10;
```

```
lamda_r = 10;
```

```
%% missile input
```

```
c = Q*s*d;
```

```

delta_a = (-C11*alpha - C12*beta - C13*p - (Iy-Iz)*q*r/c -
lamda_p*Ix*p/c + lamda_p*Ix*Up/c) / C14;
delta_e = (-Cm1*alpha - Cm2*q - (Iz-Ix)*p*r/c - lamda_q*Iy*q/c
+ lamda_q*Iy*Uq/c) / Cm3;
delta_r = ( - Cn1*beta - Cn2*r - (Ix-Iy)*p*q/c - lamda_r*Iz*r/c +
lamda_r*Iz*Ur/c) / Cn3;

delta = [delta_a delta_e delta_r]';

```

A4. Program Pemandu Rudal

```

function U = Guidance(Rt,Rm,Euler)
%% Input
% Posisi sasaran
Xt = Rt(1);
Yt = Rt(2);
Zt = Rt(3);
% Posisi rudal
Xm = Rm(1);
Ym = Rm(2);
Zm = -Rm(3);
% Sudut rudal
phi_m = Euler(1);

%% Mengolah Data
Xr = Xt - Xm; % Posisi relatif sb-x sasaran - rudal
Yr = Yt - Ym; % Posisi relatif sb-y sasaran - rudal
Zr = Zt - Zm; % Posisi relatif sb-z sasaran - rudal

theta_r = atan2(Zr,norm([Xr Yr]));
psi_r = atan2(Yr,Xr);

%% Demanded Angular Rate
Up = 0 - phi_m;
Uq = theta_r;
Ur = psi_r;
U = [Up Uq Ur]';

```

A5. Program Prediksi Posisi Sasaran

```
function kplus1 = fcn(kmin2,kmin1,k)
%% Input
x2 = kmin2(1); x1 = kmin1(1); x0 = k(1);
y2 = kmin2(2); y1 = kmin1(2); y0 = k(2);
z2 = kmin2(3); z1 = kmin1(3); z0 = k(3);

%% Mengolah data
x = [x2 x1 x0]';
y = [y2 y1 y0]';
z = [z2 z1 z0]';

ts = 1;          % sampling time
m = [1  ts  ts^2;
     1 2*ts 4*ts^2;
     1 3*ts 9*ts^2];

x_est = m\x;
y_est = m\y;
z_est = m\z;

%% k+1 prediction
kplus1 = [x_est(1) + 4*ts*x_est(2) + 16*ts^2*x_est(3);
          y_est(1) + 4*ts*y_est(2) + 16*ts^2*y_est(3);
          z_est(1) + 4*ts*z_est(2) + 16*ts^2*z_est(3)];
```

A6. Program Menentukan Jarak Relatif

```
function out = predict(kmin2,kmin1,k,Ve,Xe,time)
%% Input
x2 = kmin2(1); x1 = kmin1(1); x0 = k(1);
y2 = kmin2(2); y1 = kmin1(2); y0 = k(2);
z2 = kmin2(3); z1 = kmin1(3); z0 = k(3);

%% Mengolah data
x = [x2 x1 x0]';
y = [y2 y1 y0]';
z = [z2 z1 z0]';
```

```

ts = .01;           % sampling time
m = [1  ts  ts^2;
     1 2*ts 4*ts^2;
     1 3*ts 9*ts^2];

x_est = m\x;
y_est = m\y;
z_est = m\z;

% Closing velocity
Vc = norm([x_est(2)-Ve(1) y_est(2)-Ve(2) z_est(2)+Ve(3)]);
% time-to-go
tgo = norm([x0-Xe(1) y0-Xe(2) z0+Xe(3)])/Vc;

%% Outputs
first = k;

second = [
    x_est(1) + 4*ts*x_est(2) + 16*ts^2*x_est(3);
    y_est(1) + 4*ts*y_est(2) + 16*ts^2*y_est(3);
    z_est(1) + 4*ts*z_est(2) + 16*ts^2*z_est(3)];

third = [
    x_est(1) + x_est(2)*tgo + x_est(3)*tgo^2;
    y_est(1) + y_est(2)*tgo + y_est(3)*tgo^2;
    z_est(1) + z_est(2)*tgo + z_est(3)*tgo^2];

if time < 0.03
    out = first;
elseif time < 5
    out = second;
else
    out = third;
end

```

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS



Muhamad Rafif Prasetyo dilahirkan di Sungai Liat, Bangka Belitung, pada tanggal 6 Mei 1996. Penulis merupakan putra pertama dari tiga bersaudara Putra dari pasangan Suhito Aryo Prasetyo dan Nina Rosyina ini memulai jenjang pendidikannya di TK Aisyah Palembang, lalu dilanjutkan ke SDN Cilandak Barat 08 pagi, Jakarta. Sebelum menyelesaikan jenjang sekolah dasarnya, penulis pindah ke kota Karawang dikarenakan urusan pekerjaan kedua orang tuanya. Penulis melanjutkan jenjang sekolah dasarnya di SDIT Al-Irsyad Al-Islamiyyah Karawang, lalu dilanjutkan ke SMP Islam Al-Azhar 16 Cikarang, dan jenjang SMA di Darul Hikam *International School* hingga lulus pada tahun 2013. Selanjutnya penulis melanjutkan pendidikan ke jenjang perguruan tinggi di Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember. Selama masa perkuliahan, penulis aktif sebagai asisten dari Laboratorium Sistem dan Sibernetika. Selama menjadi asisten, penulis aktif sebagai Koordinator Praktikum Sistem Linier dan Koordinator Laboratorium Sistem dan Sibernetika ruang B405. Penulis dapat dihubungi melalui surel *prasetyorafif@gmail.com*.

Halaman ini sengaja dikosongkan