



TUGAS AKHIR - EE 184801

**PERANCANGAN PEMANDUAN *WAYPOINT* RUDAL
MENGUNAKAN KENDALI PREDIKTIF BERBASIS
MODEL**

Tommy Pratama
NRP 0711154000088

Dosen Pembimbing
Ir. Rusdhianto Effendi A.K., M.T.
Zulkifli Hidayat, S.T., M.Sc.

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

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - EE 184801

**PERANCANGAN PEMANDUAN WAYPOINT RUDAL
MENGUNAKAN KENDALI PREDIKTIF BERBASIS
MODEL**

Tommy Pratama
NRP 0711154000088

Dosen Pembimbing
Ir. Rusdhianto Effendi A.K., M.T.
Zulkifli Hidayat, S.T., M.Sc.

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

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - EE 184801

***WAYPOINT GUIDANCE DESIGN OF MISSILE USING
MODEL BASED PREDICTIVE CONTROL***

Tommy Pratama
NRP 0711154000088

Supervisor
Ir. Rusdhianto Effendi A.K., M.T.
Zulkifli Hidayat, S.T., M.Sc.

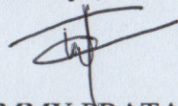
***DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019***

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Perancangan Pemanduan Waypoint Rudal Menggunakan Kendali Prediktif Berbasis Model**" adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2019



TOMMY PRATAMA
Nrp 0711154000088

[Lampiran ini wajib dikembalikan]

LAMBAR PENGESAHAN

[Halaman ini sengaja dikosongkan]

**PERANCANGAN PEMANDUAN WAYPOINT RUDAL
MENGUNAKAN KENDALI PREDIKTIF
BERBASIS MODEL**

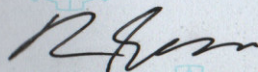
TUGAS AKHIR

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

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,



**Ir. Rusdhianto Effendi A.K., M.T.
NIP. 19570424 198502 1001**

**Zulkiffi Hidayat, S.T., M.Sc.
NIP. 19701225 199903 1002**



[Halaman ini sengaja dikosongkan]

PERANCANGAN PEMANDUAN WAYPOINT RUDAL DENGAN MENGGUNAKAN KENDALI PREDIKTIF BERBASIS MODEL

Nama : Tommy Pratama
NRP : 0711154000088
Dosen Pembimbing I : Ir. Rusdianto Effendi A.K., M.T.
Dosen Pembimbing II: Zulkifli Hidayat, S.T., M.Sc.

ABSTRAK

Perancangan autopilot dengan menggunakan pengaturan prediktif disajikan dalam riset ini. Rudal adalah sistem yang non-linier, *multi input and multi output* (MIMO) dan *coupled*. Perancangan pengaturan yang diajukan menggunakan *discrete model predictive control* (DMPC). Model sistem dibutuhkan oleh DMPC untuk memprediksi perilaku sistem yang akan datang yang digunakan untuk menyelesaikan algoritma optimisasi untuk mendapatkan sinyal kendali optimal yang menggerakkan keluaran proses. Sebelumnya, model linier didapatkan dengan *nonlinear feedback linearization*. Tujuan utamanya adalah tercapainya kemampuan rudal menabrak sasarannya dengan mengikuti akselerasi yang diperintahkan oleh sistem pemandu. Pada tugas akhir ini, simulasi pengejaran rudal sasaran dengan mengikuti lintasan menggunakan pemanduan *nonlinear guidance logic*, dan mengejar sasaran bergerak menggunakan *proportional navigation* telah ditunjukkan dengan perancangan pengaturan yang diajukan. Simulasi menunjukkan pemanduan waypoint rudal memiliki *miss distance* rata-rata terkecil rudal terhadap *waypoint* yaitu 0.0872 m serta RMSE rudal terhadap lintasannya yaitu 0.1497 m dan 0.7263 m untuk bidang horizontal dan vertikal. Sedangkan pengejaran sasaran menunjukkan *miss distance* terdekat rata-rata antara sasaran dan rudal adalah 0.3232 m.

Kata Kunci: non-linier, *MIMO*, *coupled system*, autopilot, *nonlinear feedback linearization*, *discrete model predictive control*, *nonlinear guidance logic*, *proportional navigation*

[Halaman ini sengaja dikosongkan]

WAYPOINT GUIDANCE OF MISSILE DESIGN USING MODEL BASED PREDICTIVE CONTROL

Name : Tommy Pratama
Register Number : 0711154000088
Supervisor I : Ir. Rusdianto Effendi A.K., M.T.
Supervisor II : Zulkifli Hidayat, S.T., M.Sc.

ABSTRACT

The autopilot of missile design using predictive controller is presented in this research paper. The missile is nonlinear, Multi Input and Multi Output (MIMO) and coupled system. The proposed controller design is based on discrete model predictive control (DMPC). A model of system is needed by MPC to predict system's future behaviour and used to solve optimization algorithm to find optimal control action driving the process output. In advance, the linear model is obtained by nonlinear feedback linearization. The main goal is the achievement of the ability of missile colliding its target by following the commanded acceleration by guidance system. In this paper, the missile-target engagement simulation by following the path using nonlinear guidance logic guidance, and pursuing the moving target using proportional navigation has been illustrated with proposed controller design. The simulation results the smallest average of waypoint guidance miss distance is 0.0872 meter and RMSE of missile-path are 0.1497 meter and 0.7263 meter on horizontal and vertical plan, respectively. While the missile-target engagement shows the average of nearest miss distance of missile and the target is 0.3232 meter.

Key Words: *nonlinear, MIMO, coupled system, autopilot, nonlinear feedback linearization, discrete model predictive control, nonlinear guidance logic, proportional navigation*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan berkah dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir ini. Tugas akhir ini diusulkan sebagai lanjutan dari penelitian tentang peluru kendali dan sebagai prasyarat untuk mendapatkan gelar Sarjana Teknik dari Departemen Teknik Elektro ITS. Penulis juga ingin menyampaikan terima kasih kepada:

1. Keluarga penulis, khususnya ayah dan ibu yang senantiasa memberikan dukungan baik dukungan material maupun non material.
2. Bapak Dr. Eng. Ardyono Priyadi, S.T., M.Eng. selaku Kepala Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
3. Bapak Ir. Rushianto Effendie A.K., M.T. dan Bapak Zulkifli Hidayat, S.T., M.Sc. selaku dosen pembimbing dan Mas Mochammad Nur Qomarudin yang selalu memberikan motivasi, ilmu dan saran serta bantuan dalam penelitian ini.
4. Bapak dan ibu dosen pengajar Departemen Teknik Elektro, khususnya bidang studi Sistem Pengaturan yang telah memberikan bekal ilmu kepada penulis.
5. Seluruh teman-teman penulis teman seperjuangan AJ204 dan teman angkatan e55..
6. Seluruh teman-teman dan dosen pembimbing Tim Robotika ITS khususnya tim ICHIRO.
7. Jessica yang selalu senantiasa memberikan dukungan, motivasi dan tempat berbagi cerita suka maupun duka selama proses pengerjaan tugas akhir ini.

Kesempurnaan hanya milik Tuhan Yang Maha Esa, penyusunan tugas akhir ini tentu masih banyak kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun untuk penelitian ini. Semoga penelitian ini dapat memberikan manfaat terutama penelitian peluru kendali. Semoga dengan penelitian ini dapat memberikan kontribusi yang berarti pada penelitian laboratorium.

Surabaya, Juni 2019
Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xxi
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 [1].....	5
2.2. Sudut Euler [1],[2].....	6
2.3. Peluru Kendali [1],[3]	8
2.4. Model Matematika Rudal [4]	10
2.5. Kinematika Rudal [4].....	11
2.6. Dinamika Rudal [1],[4]	13
2.6.1. Dinamika Translasi [1].....	13
2.6.2. Dinamika Rotasi [1].....	16
2.7. Autopilot Rudal [1],[3],[11]	19
2.8. <i>Nonlinear Feedback Linearization</i> [5],[12]	21
2.9. <i>Discrete Model Predictive Control</i> [6],[8].....	23
2.10. <i>Path Planning</i> [9].....	28
2.11. Pemanduan Rudal [1],[3],[7].....	29
2.11.1. <i>Line of sight</i> (LOS) [13].....	29
2.11.2. <i>Nonlinear Guidance Logic</i> (NLGL) [9],[10]	30
2.11.3. <i>Proportional Navigation</i> (PN) [13],[14]	32
BAB 3 PERANCANGAN SISTEM.....	35
3.1. Gambaran Umum Sistem	35
3.2. Perancangan Model Rudal.....	35

3.2.1. Perancangan Dinamika Rudal	36
3.2.2. Perancangan Kinematika Rudal	42
3.3. Perancangan <i>Nonlinear Feedback Linearization</i>	45
3.3.1. Linierisasi Kecepatan Rotasi Rudal	45
3.3.2. Linierisasi Kecepatan Translasi Rudal	48
3.4. Perancangan <i>Discrete Model Predictive Control</i> (DMPC)	51
3.5. Perancangan Prediksi Kecepatan Rudal	54
3.6. Perancangan <i>Path Planning</i>	56
3.7. Perancangan Pemanduan Navigasi NLGL	58
3.8. Perancangan Pemanduan <i>Proportional Navigation</i>	61
3.9. Perancangan Pengujian Sistem	61
3.9.1. Perancangan Pengujian Model Matematika Rudal	62
3.9.2. Perancangan Pengujian Linierisasi Dinamika Rudal	62
3.9.3. Perancangan Pengujian DMPC	62
3.9.4. Perancangan Pengujian Pemanduan NLGL	62
3.9.5. Perancangan Pengujian <i>Proportional Navigation</i>	63
3.9.6. Perancangan Pengejaran Sasaran Bergerak	63
BAB 4 HASIL DAN ANALISA DATA	65
4.1. Pengujian Model Matematika Rudal	65
4.2. Pengujian <i>Nonlinear Feedback Linearization</i>	67
4.3. Pengujian <i>Discrete Model Predictive Control</i>	72
4.4. Pengujian Pemanduan NLGL	80
4.5. Pengujian Pemanduan <i>Proportional Navigation</i>	86
4.6. Pengujian Pengejaran Sasaran Bergerak	88
BAB 5 PENUTUP	91
5.1. Kesimpulan	91
5.2. Saran	92
DAFTAR PUSTAKA	93
LAMPIRAN	95
Lampiran 1. Blok Model Matematika Rudal dengan Simulink	95
Lampiran 2. Blok Autopilot Rudal dengan Simulink	95
Lampiran 3. Blok <i>Discrete Model Predictive Control</i> dengan Simulink	96
Lampiran 4. Blok Autopilot dan Pemanduan dengan Simulink	96
RIWAYAT PENULIS	97

DAFTAR GAMBAR

Gambar 2.1. Sistem Koordinat pada Rudal [1]	5
Gambar 2.2. Sistem Koordinat pada Rudal [1]	6
Gambar 2.3. Peluru Kendali [1]	8
Gambar 2.4. Diagram Blok Sistem Rudal	9
Gambar 2.5. <i>Boost-Sustain Booster</i> [1]	11
Gambar 2.6. Sudut Euler pada Rudal [1]	12
Gambar 2.7. Rudal dengan model sirip tail [3]	20
Gambar 2.8. Sirip Ekor Rudal Tipe <i>Cross</i>	21
Gambar 2.9. Diagram Blok <i>Nonlinear Feedback Linearization</i>	23
Gambar 2.10. Pendekatan <i>Discrete Model Predictive Control</i> [8]	23
Gambar 2.11. Blok Diagram Alir <i>Model Predictive Control</i> [6]	28
Gambar 2.12. Tiga Fase Terbang Rudal [7]	29
Gambar 2.13. Geometri Planar Sudut LOS [7]	30
Gambar 2.14. Diagram <i>Nonlinear Guidance Logic</i> [9]	31
Gambar 2.15. Model Linier Pemanduan <i>Nonlinear Guidance Logic</i> Garis Lurus [9]	32
Gambar 2.16. Diagram Proportional Navigation [13]	33
Gambar 3.1 Blok MATLAB <i>Function</i> Gaya dan Momen Rudal	37
Gambar 3.2 Masukkan Thruster pada Model Rudal	38
Gambar 3.3 Blok MATLAB <i>Function</i> Dinamika Rudal	40
Gambar 3.4 Blok Simulink Massa dan Momen Inersia Rudal	41
Gambar 3.5 Blok Model Lingkungan Rudal	41
Gambar 3.6 Blok Perhitungan Sudut α , β , q dan Mach	42
Gambar 3.7 Blok Perhitungan Sudut Euler	44
Gambar 3.8 Blok Kinematika Rudal	44
Gambar 3.9 Blok Dinamika dan Kinematika Rudal	45
Gambar 3.10 Blok Diagram Simulink <i>Nonlinear Feedback Linearization</i> Dinamika Rotasi	47
Gambar 3.11 Blok Simulink Keseluruhan <i>Nonlinear Feedback</i> <i>Linearization</i> Dinamika Rudal	49
Gambar 3.12 Blok Simulink Dinamika Aktuator Sirip	50
Gambar 3.13 Blok Simulink Aktuator Sirip Rudal yang dihubungkan dengan Blok Linierisasi Dinamika	51
Gambar 3.14 Diagram Blok Perancangan DMPC	52

Gambar 3.15 Karakteristik Respon Orde Satu	52
Gambar 3.16 Blok Simulink Pengaturan DMPC.....	54
Gambar 3.17 Ilustrasi Prediksi Data Kecepatan.....	55
Gambar 3.18 Blok Prediksi Referensi Kecepatan	56
Gambar 3.19 Diagram Alir Algoritma <i>Path Planning</i> Navigasi <i>Waypoint</i>	58
Gambar 3.20 Blok Simulink Pemanduan NLGL.....	60
Gambar 3.21 Blok Simulink Pemanduan PN.....	61
Gambar 3.22 Diagram Alir Pengejaran Sasaran.....	63
Gambar 4.1 Posisi Rudal (a) x_e , (b) y_e dan (c) z_e	66
Gambar 4.2 Sudut <i>Pitch</i> Rudal (θ)	67
Gambar 4.3 Sudut Defleksi (a) <i>Aileron</i> , (b) <i>Elevator</i> , (c) <i>Rudder</i>	68
Gambar 4.4 Titik <i>Overshoot</i> Maksimum dan Waktu <i>Settling</i> dari Respon Sudut Defleksi <i>Aileron</i> (<i>Zoom In</i>).....	68
Gambar 4.5 Pengujian Respon <i>Step</i> Hasil Linierisasi Kecepatan Rotasi (a) p , (b) q dan (c) r	70
Gambar 4.6 Respon Kecepatan Translasi Rudal (a) v dan (b) w	71
Gambar 4.7 Respon Kecepatan v dan Model Pendekatan Orde Satu..	72
Gambar 4.8 (Kiri) Respon Kecepatan v dengan Masukkan <i>Step</i> Skenario (a) 1, (b) 2, (c) 3, (d) 4, (Kanan) Grafik <i>Error</i> Respon v	74
Gambar 4.9 (Kiri) Respon Kecepatan w dengan Masukkan <i>Step</i> Skenario (a) 1, (b) 2, (c) 3, (d) 4, (Kanan) Grafik <i>Error</i> Respon w	75
Gambar 4.10 Trayektori Referensi Kecepatan Rudal	77
Gambar 4.11 (a) Respon Kecepatan v Rudal, (b) <i>Zoom In</i>	78
Gambar 4.12 Respon Kecepatan w Rudal dengan DMPC.....	78
Gambar 4.13 Grafik Kesalahan Respon Kecepatan Translasi Terhadap Referensi (Kiri) v dan (Kanan) w	78
Gambar 4.14 Perbandingan Kecepatan w Aktual dan Prediksi (a) 0.05 Detik (b) 0.1 Detik.....	79
Gambar 4.15 Lintasan <i>Waypoint</i> Rudal Skenario 1.....	81
Gambar 4.16 Lintasan <i>Waypoint</i> Rudal Skenario 2.....	81
Gambar 4.17 Lintasan Rudal Skenario 1 dengan Pemandu NLGL.....	82
Gambar 4.18 Pengujian Pemanduan NLGL pada Skenario 2	82
Gambar 4.19 <i>Cross-Track Error</i> Pemanduan NLGL Skenario 1	83
Gambar 4.20 <i>Cross-Track Error</i> Pemanduan NLGL Skenario 2.....	83

Gambar 4.21 Perbandingan Percepatan Translasi Skenario 1 (a) Horizontal (b) Vertikal	84
Gambar 4.22 Perbandingan Percepatan Translasi Skenario 2 (a) Horizontal (b) Vertikal	85
Gambar 4.23 Perbandingan Percepatan Translasi (<i>Zoom In</i>)	85
Gambar 4.24 <i>Error</i> Percepatan Rudal dan Pemandu (Kiri) Horizontal (Kanan) Vertikal Skenario 1	85
Gambar 4.25 <i>Error</i> Percepatan Rudal dan Pemandu (Kiri) Horizontal (Kanan) Vertikal Skenario 2	86
Gambar 4.26 Pemanduan PN Skenario (a) 1, (b) 2, (c) 3	88
Gambar 4.27 Pengejaran Sasaran Skenario (a) 1, (b) 2	90
Gambar 4.28 <i>Closing Distance</i> Rudal dan Sasaran Skenario 1	90
Gambar 4.29 <i>Closing Distance</i> Rudal dan Sasaran Skenario 2	90

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1. Tabel Nilai Parameter Rudal	43
Tabel 4.1. Tabel Pengujian Kondisi Awal Rudal.....	65
Tabel 4.2. Tabel Karakteristik Kecepatan Rotasi Rudal	70
Tabel 4.3. Tabel Karakteristik Respon Kecepatan Translasi Rudal	72
Tabel 4.4. Tabel Pengujian Parameter DMPC.....	73
Tabel 4.5. Tabel Pengujian Parameter DMPC dengan Trayektori	77
Tabel 4.6. Tabel RMSE Kesalahan Kecepatan Translasi v dan w	79
Tabel 4.7. Tabel Daftar <i>Waypoint</i> untuk <i>Path Planning</i>	80
Tabel 4.8. Tabel <i>Miss Distance</i> Masing-Masing <i>Waypoint</i>	83
Tabel 4.9. Tabel RMSE <i>Cross-Track Error</i> Pemanduan NLGL	84
Tabel 4.10. Tabel Kesalahan Respon Percepatan Rudal dengan Referensi	86
Tabel 4.11. Tabel Gerak Sasaran untuk Pengujian PN	86
Tabel 4.12. Tabel Waktu Tempuh dan <i>Miss Distance</i>	88
Tabel 4.13. Tabel Gerak Sasaran untuk Pengejaran Sasaran Bergerak	89
Tabel 4.14. Tabel Waktu Tempuh dan <i>Miss distance</i> Pengejaran Sasaran	89

[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 tugas akhir dan relevansi.

1.1. Latar Belakang

Salah satu faktor yang mendukung keutuhan sebuah negara adalah sistem militer yang dimiliki oleh negara tersebut. Dalam dunia militer, peluru kendali (rudal) adalah sebuah senjata roket yang umum digunakan untuk menemukan dan menghancurkan sasarannya baik dikendalikan secara langsung atau otomatis. Agar tiba pada sasarannya, rudal harus melewati *waypoint* yang ditentukan. Untuk memastikan rudal tersebut mampu bergerak hingga mengenai sasaran, rudal harus mampu mengikuti referensi yang dibuat oleh pemandu. Selain itu, desain rudal harus dapat memenuhi hukum aerodinamika sehingga gaya-gaya aerodinamis yang bekerja pada sirip rudal mampu mengangkat rudal ke udara. Kemudian, sirip-sirip pada rudal inilah yang diatur untuk mengendalikan pergerakan rudal selama terbang. Pengaturan sirip ini dilakukan oleh salah satu bagian rudal yaitu kontroler atau autopilot dan didesain sedemikian rupa agar dapat menggerakkan rudal sesuai perintah pemandu.

Beberapa pendekatan telah dilakukan dengan menggunakan pengaturan PI *cascade* [15], pengaturan adaptif [16], *gain scheduling* [17] dan pengaturan optimal [18] yang telah digunakan untuk mengatasi permasalahan *tracking* sinyal referensi pemandu. Menggunakan pengaturan PI *cascade* membutuhkan banyak lup karena variabelnya yang dikontrol lebih dari satu, pengaturan adaptif membutuhkan waktu untuk adaptasi dengan referensinya, sementara *gain scheduling* adalah sederhana namun tidak ada jaminan stabil saat waktu transisi antara titik operasinya. Pengaturan prediktif merupakan salah satu teknik pengaturan modern berbasis model multivariabel yang menggunakan prediksi sistem untuk menghasilkan keluaran yang terbaik. Pengaturan prediktif juga menggunakan teknik optimasi untuk menjamin keluaran proses yang dihasilkan yang akan datang mengikuti referensi yang diinginkan. Oleh karena itu, dalam tugas akhir ini akan dirancang pengaturan prediktif yang mampu mengendalikan dan memandu rudal untuk dapat bergerak tetap pada lintasannya dengan kesalahan yang minimum.

1.2. Permasalahan

Rudal adalah sistem yang memiliki dinamika non-linier dan memiliki karakteristik *decoupled*. Saat rudal bergerak di udara pada fase *midcourse*, rudal harus mampu melewati titik-titik acuan (*waypoint*) yang memiliki lokasi yang berbeda-beda dan dilanjutkan fase *terminal* yaitu fase rudal menuju dan mengenai sasaran. Pemandu akan menerjemahkan orientasi yang harus dicapai oleh rudal menjadi percepatan referensi agar bergerak sedekat mungkin dengan lintasan yang harus dilaluinya. Dengan demikian, pemandu tentu akan memberikan referensi yang nilainya selalu berubah-ubah terhadap waktu. Oleh karena itu, diperlukan suatu pengendali autopilot yang mampu mengendalikan rudal mengikuti referensi pemandu dengan kesalahan seminimum mungkin. Permasalahan ini akan diselesaikan dengan menggunakan pengendali DMPC.

1.3. Batasan Masalah

Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Rudal diasumsikan benda tegar dan tidak mengalami aeroelastis
2. Model rudal yang digunakan adalah *skid-to-turn* dan enam derajat kebebasan gerak
3. Bahan bakar rudal tidak habis selama terbang di udara
4. Rudal terbang di udara tidak mengalami gangguan
5. Informasi yang tersedia bersifat deterministik
6. *Thruster* tidak dapat dikendalikan

1.4. Tujuan

Tujuan yang akan dicapai pada penelitian ini adalah bagaimana merancang sebuah pengendali rudal yang mampu mengendalikan kecepatan rudal tetap pada referensi percepatan yang diberikan agar tetap pada lintasan terbang yang menghubungkan *waypoint* yang akan dilewati oleh rudal hingga mengenai sasarannya. Oleh sebab itu, digunakanlah pengendali DMPC yang mampu mengendalikan rudal mengikuti referensinya. Dinamika sistem rudal yang akan dikendalikan oleh DMPC adalah non-linier sehingga digunakan metode *nonlinear feedback linearization* agar sistem rudal bersifat linier dan dapat diaplikasikan dalam pengendali DMPC.

1.5. Metodologi

Tahapan penyelesaian tugas akhir ini adalah sebagai berikut.

- a) Studi Literatur
Melakukan studi literatur dalam mendapatkan persamaan matematika dari rudal dengan menggunakan sumber seperti buku, jurnal ilmiah, dan internet. Selanjutnya adalah melakukan studi literatur tentang perancangan pengendali DMPC, dan metode pemanduan. Sumber literatur yang digunakan adalah buku acuan.
- b) Permodelan Matematis Rudal
Membuat model persamaan matematis non-linier dari rudal yang telah diperoleh dari sumber literatur yang akan dijadikan objek kajian dalam tugas akhir.
- c) Linierisasi Persamaan Rudal
Linierisasi persamaan dinamika non-linier rudal menggunakan metode *nonlinear feedback linearization*.
- d) Perancangan *Discrete Model predictive control*
Melakukan perancangan *discrete model predictive control* sebagai pengendali percepatan dari rudal.
- e) Perancangan Lintasan *Waypoint*
Melakukan perancangan lintasan *waypoint* rudal dengan menggunakan garis lurus antara *waypoint* satu ke *waypoint*.
- f) Perancangan Pemanduan
Pemanduan digunakan untuk memandu rudal menuju sasaran statis dan sasaran dinamisnya dengan memberikan sinyal perintah untuk pengaturan autopilot rudal.
- g) Hasil dan Analisa
Seluruh perancangan dianalisis dan disimulasikan menggunakan *software* Simulink MATLAB. Setiap komponen dianalisis apakah sesuai dengan dasar teori dan kriteria desain yang diinginkan. Hasil yang diharapkan yaitu rudal mampu mengikuti lintasan *waypoint* yang ditentukan dan mampu mengenai sasaran Bergeraknya dengan menggunakan metode pengaturan dan pemanduan yang dirancang.
- h) Kesimpulan dan Saran
Penarikan kesimpulan dilakukan setelah melakukan analisa dari hasil simulasi yang telah dijalankan. Selanjutnya pemberian saran berupa seperti apa penelitian yang dapat dilanjutkan dari tugas akhir ini.

- i) 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 matematika rudal, pemandu rudal, linierisasi menggunakan *feedback linearization*, *model predictive control*, dan lintasan *waypoint*.

BAB III PERANCANGAN SISTEM

Dalam bab ini dijelaskan langkah-langkah pembuatan simulasi rudal, mulai dari pembuatan model matematika rudal, perancangan linierisasi pada rudal, perancangan *model predictive control*, perancangan lintasan rudal dan pemanduan rudal.

BAB IV HASIL DAN ANALISA

Bab ini berisi pengujian dan simulasi perancangan simulasi rudal yang telah dibuat. Analisa dilakukan dengan membuat perbandingan dengan pengendali yang sebelumnya.

BAB V PENUTUP

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

1.7. Relevansi

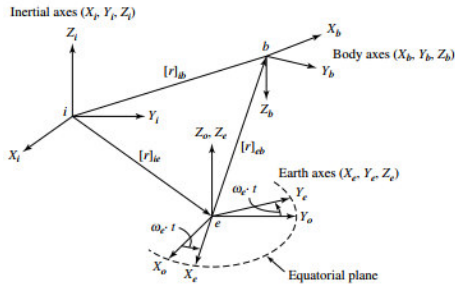
Hasil dari pelaksanaan tugas akhir ini diharapkan dapat memberikan manfaat sebagai salah satu referensi yang dapat digunakan dalam penelitian sistem pertahanan udara khususnya dalam sistem pertahanan udara Indonesia. Selain itu juga dapat membuka topik kajian baru dengan menggunakan pemandu atau pengendalian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

BAB 2 TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai dasar teori yang digunakan dalam pelaksanaan tugas akhir ini, yaitu mengenai model matematika rudal, sistem autopilot, perencanaan lintasan dan metode pemandu yang digunakan.

2.1. Sistem Koordinat [1]

Sistem koordinat adalah sistem yang menggunakan satu atau lebih angka atau koordinat untuk menentukan secara khusus letak titik atau elemen geometri dalam dimensi N ruang Euclidian. Sistem koordinat yang digunakan dalam dunia nyata adalah sistem koordinat dengan 3 dimensi. Lokasi titik pada suatu sistem koordinat bisa berbeda dengan sistem koordinat yang lainnya karena terdapat perbedaan titik referensi atau lokasi awal yang digunakan. Selain hanya merepresentasikan posisi titik pada sistem koordinat. Oleh karena itu, untuk merepresentasikan lokasi dan orientasi relatif pada sistem koordinat yang satu dengan lainnya, diperlukan kerangka koordinat pada badan dan menentukan hubungan antar kerangka koordinat dengan transformasi koordinat dari sistem koordinat yang satu ke sistem koordinat yang lainnya yang dapat dilihat pada Gambar 2.1.



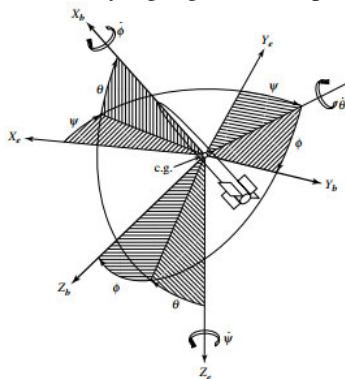
Gambar 2.1. Sistem Koordinat pada Rudal [1]

Dalam tugas akhir ini, sistem koordinat yang digunakan pada rudal adalah sistem koordinat yang menggunakan kaidah tangan kanan dimana sumbu x positif berada sepanjang sumbu longitudinal rudal, sumbu y positif berada pada sayap kanan rudal dan sumbu z positif adalah hasil *cross product* antara sumbu x dan y yaitu ke bawah. Sistem koordinat ini

juga dikenal sebagai *north-east-down* (NED) dan sistem sumbu referensi rudal berada pada titik *center of gravity* (cog).

2.2. Sudut Euler [1],[2]

Orientasi dari satu sistem koordinat yang satu terhadap sistem koordinat yang lainnya dapat selalu digambarkan oleh tiga rotasi seputar sumbu koordinat yang saling tegak lurus dan sudut rotasi itu disebut dengan sudut Euler. Sudut Euler secara umum digunakan untuk menentukan orientasi pada rudal terhadap pada sumbu koordinat bumi. Sudut Euler adalah metode umum untuk menentukan hubungan antara kerangka koordinat yang bergerak terhadap kerangka koordinat yang diam dengan sebuah matriks rotasi. Notasi yang digunakan untuk menentukan sudut pada rudal adalah ϕ yaitu gerak *roll* pada sumbu x , θ yaitu gerak *pitch* pada sumbu y , dan ψ yaitu gerak *yaw* pada sumbu z . Sudut-sudut ini sangat berguna untuk menggambarkan pergerakan sebuah badan yang berotasi pada titik yang diam, seperti pada sebuah giroskop atau sebuah badan yang berotasi pada pusat massanya (*center of mass*) seperti pesawat dan rudal yang digambarkan pada Gambar 2.2.



Gambar 2.2. Sistem Koordinat pada Rudal [1]

Untuk menentukan posisi yang baru dari sumbu utama sebagai hasil perpindahan rotasi melalui sudut Euler, perlu beberapa tahap rotasi koordinat. Transformasi yang dilakukan dengan cara memutar setiap sumbu dari satu sistem koordinat hingga menjadi sistem koordinat yang diinginkan secara berurutan. Urutan pemutarannya harus selalu konsisten karena jika urutannya berubah maka matriks transformasi pemutarannya

akan menjadi berbeda. Pada tugas akhir ini, dilakukan urutan pemutaran terhadap sumbu z , sumbu y dan sumbu x .

Transformasi koordinat dari sistem koordinat bumi ke sistem koordinat badan rudal melalui beberapa urutan transformasi rotasi. Pertama yaitu melakukan transformasi rotasi pada sumbu x menggunakan Persamaan (2.1)

$$\begin{pmatrix} X'_e \\ Y'_e \\ Z'_e \end{pmatrix} = (T_1) \begin{pmatrix} X_e \\ Y_e \\ Z_e \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} X_e \\ Y_e \\ Z_e \end{pmatrix} \quad (2.1)$$

Dimana:

X'_e, Y'_e, Z'_e = Posisi rudal dalam koordinat bumi yang dirotasi terhadap sumbu x
 X_e, Y_e, Z_e = Posisi rudal pada koordinat bumi
 ϕ = Sudut putar *roll* sumbu x

Sistem koordinat yang dihasilkan diatas adalah tetap pada bidang XY -nya. Selanjutnya yaitu dengan melakukan perputaran pada sumbu y dengan menggunakan Persamaan (2.2)

$$\begin{pmatrix} X''_e \\ Y''_e \\ Z''_e \end{pmatrix} = (T_2) \begin{pmatrix} X'_e \\ Y'_e \\ Z'_e \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} X'_e \\ Y'_e \\ Z'_e \end{pmatrix} \quad (2.2)$$

Dimana:

X''_e, Y''_e, Z''_e = Posisi rudal dalam koordinat bumi yang dirotasi terhadap sumbu y
 θ = Sudut putar *pitch* sumbu y

Terakhir yaitu dengan memutar sumbu terakhir yaitu sumbu z dengan Persamaan (2.3)

$$\begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} = (T_3) \begin{pmatrix} X''_e \\ Y''_e \\ Z''_e \end{pmatrix} = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X''_e \\ Y''_e \\ Z''_e \end{pmatrix} \quad (2.3)$$

Dimana:

X_b, Y_b, Z_b = Posisi rudal dalam koordinat badan
 ψ = Sudut putar *yaw* sumbu z

Sehingga dari ketiga proses tersebut dan bila menggabungkan persamaan, didapatkan matriks transformasi rotasi dari sistem koordinat bumi menjadi sistem koordinat badan pada Persamaan (2.4)

$$\begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} = (T_4) \begin{pmatrix} X_e \\ Y_e \\ Z_e \end{pmatrix} = (T_3)(T_2)(T_1) \begin{pmatrix} X_e \\ Y_e \\ Z_e \end{pmatrix} \quad (2.4)$$

Dengan matriks T_4 adalah Persamaan (2.5)

$$T_4 = \begin{pmatrix} c\theta \ c\psi & c\theta \ s\psi & -s\theta \\ s\phi \ s\theta \ c\psi - c\phi \ s\psi & s\phi \ s\theta \ s\psi + c\phi \ c\psi & s\phi \ c\theta \\ c\phi \ s\theta \ c\psi + s\phi \ s\psi & c\phi \ s\theta \ s\psi - s\phi \ c\psi & c\phi \ c\theta \end{pmatrix} \quad (2.5)$$

Dimana c adalah cosinus dan s adalah sinus serta:

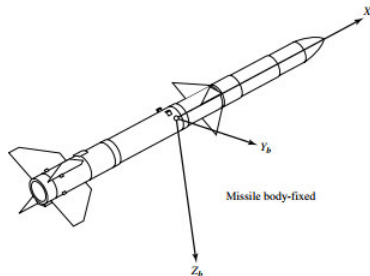
X_b, Y_b, Z_b = Sistem koordinat pada rudal

X_e, Y_e, Z_e = Sistem koordinat pada bumi

T_4 = Matriks transformasi rotasi

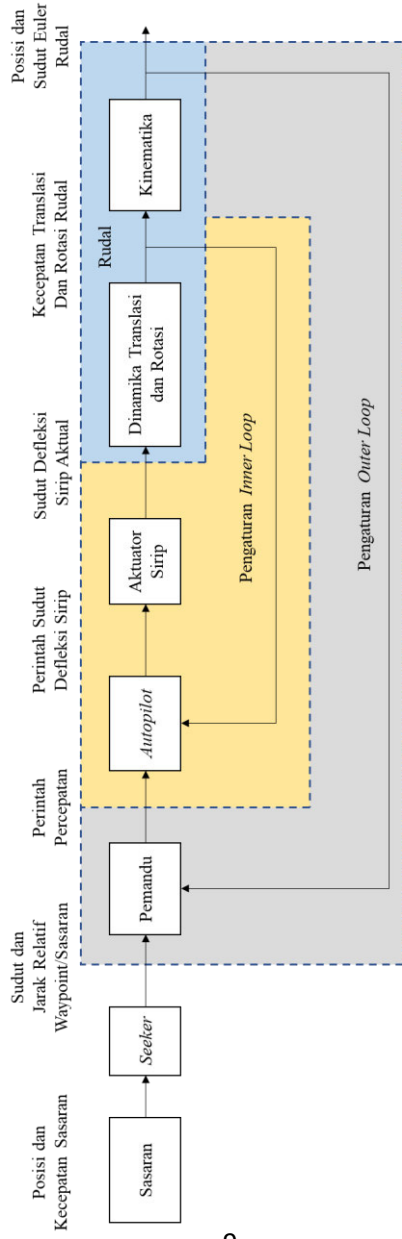
2.3. Peluru Kendali [1],[3]

Sebuah peluru kendali atau biasa disebut dengan rudal dapat didefinisikan sebagai objek apapun yang dapat diluncurkan, diproyeksikan atau didorong menuju sebuah sasaran. Dengan kata lain, rudal adalah sebuah peluru yang membawa muatan yaitu hulu ledak yang dipandu menuju sebuah sasaran dengan manual maupun otomatis. Jelasnya, rudal digunakan terutama sebagai senjata menghancurkan sasaran. Gambar rudal dapat dilihat pada Gambar 2.3.



Gambar 2.3. Peluru Kendali [1]

Pada dasarnya, rudal memiliki prinsip gerak yang sama dengan pesawat untuk terbang dengan stabil di udara. Terdapat tiga macam gerak *rolling*, *pitching* dan *yawing*. Gerak rudal ini diatur oleh bagian dari rudal yang bernama *control surface*. Prinsip kerja *control surface* adalah menghasilkan gaya aerodinamis untuk menghasilkan *attitude* yang diinginkan. *Control surface* tersebut yaitu *aileron*, *elevator* dan *rudder*.



Gambar 2.4. Diagram Blok Sistem Rudal

Aileron adalah *control surface* yang menghasilkan gerak *roll*, *elevator* untuk menghasilkan gerak *pitch* dan *rudder* untuk menghasilkan gerak *yaw*. Blok diagram keseluruhan sistem rudal digambarkan pada Gambar 2.4.

Rudal dapat diklasifikasikan menjadi beberapa kategori. Berdasarkan bagaimana mereka diarahkan ke sasarannya, ada dua kategori yaitu rudal yang dipandu dan rudal yang tidak dipandu. Rudal yang tidak dipandu baik hanya didorong diawal maupun secara berkelanjutan dapat diarahkan ke sasarannya hanya sebelum mereka ditembakkan. Setelah menembak, mereka tidak dapat dikendalikan lagi. Sebaliknya rudal yang dipandu mendapat perintah untuk mengikuti gerakan tertentu selama terbang. Rudal yang dipandu dapat dikategorikan menjadi rudal taktis (*Tactical Missiles*) dan rudal balistik (*Ballistic Missiles*) berdasarkan jarak yang ditempuh. Rudal taktis hanya digunakan pada skenario jarak pendek hingga menengah, sebaliknya rudal balistik digunakan untuk jarak yang sangat jauh. Berdasarkan misi rudal tersebut, dapat dibagi lagi menjadi empat bagian yaitu

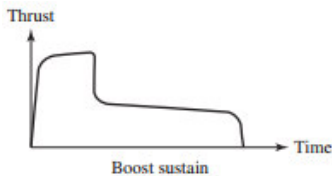
- *Surface to Surface Missiles (SSM)*
- *Surface to Air Missiles (SAM)*
- *Air to Air Missiles (AAM)*
- *Air to Surface Missiles (ASM)*

Dalam merancang rudal, permasalahan pemanduan dan pengaturan melibatkan empat tahap yaitu permodelan dinamika, pemanduan, pengaturan dan estimasi pergerakan sasaran. Dalam tahap permodelan dinamika, rudal dimodelkan sehingga hubungan antara masukan yang diberikan dan variabel keluaran diperoleh. Kemudian algoritma pemanduan dikembangkan untuk memandu rudal menuju sasaran yang diharapkan. Setelah algoritma pemanduan telah dibangun, tahap selanjutnya adalah merancang sistem pengaturan berdasarkan model dinamika dari rudal sehingga dapat model rudal dapat mengerti dan mematuhi sinyal yang diberikan. Tahap selanjutnya adalah mengestimasi lokasi dan pergerakan sasaran.

2.4. Model Matematika Rudal [4]

Gerak rudal dapat diturunkan menggunakan persamaan gerak berdasarkan Hukum Newton. Model matematika rudal adalah formulasi matematik yang menyatakan hubungan atau keterkaitan antar variabel dengan melibatkan parameter pada rudal. Dalam menurunkan persamaan gerak rudal ada beberapa asumsi yang digunakan. Pertama adalah rudal

adalah benda tegar yang tidak mengalami perubahan apapun pada ukuran dan bentuknya saat terbang. Namun, karena rudal memiliki bahan bakar dimana selama rudal terbang banyaknya bahan bakar yang digunakan akan semakin berkurang maka tentu saja massa rudal akan berkurang seiring dengan waktu. Selain massa yang berubah seiring dengan waktu, momen inersia yang memengaruhi rotasi rudal juga akan berubah dikarenakan perubahan massa rudal. Jenis pendorong rudal yang digunakan pada tugas akhir ini adalah jenis *boost-sustain*. Grafik penggunaan pendorong rudal dapat dilihat pada Gambar 2.5.

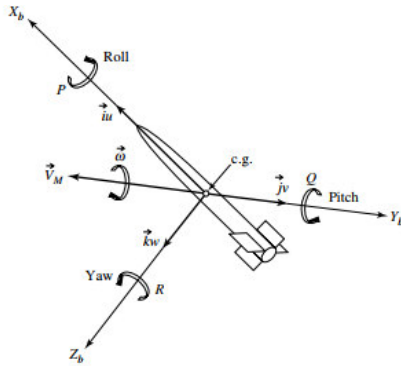


Gambar 2.5. *Boost-Sustain Booster* [1]

Pada tugas akhir ini, rudal meliputi kinematika dan dinamika rudal dengan memiliki gerak enam derajat kebebasan. Gerak enam derajat kebebasan meliputi tiga derajat kebebasan gerak translasi dan tiga derajat kebebasan gerak rotasi pada setiap sumbu pada rudal. Gerakan ini direpresentasikan sebagai kecepatan translasi (u, v, w) dan kecepatan rotasi (p, q, r) yang dapat dilihat pada Gambar 2.6. Sistem sumbu referensi rudal berada pada titik *center of gravity (cog)*. Khususnya pada tugas akhir ini, rudal yang diasumsikan adalah rudal jenis *skid-to-turn*. *Skid-to-turn* adalah jenis bagaimana sebuah rudal melakukan belok. Pada rudal jenis *skid-to-turn*, saat rudal belok rudal tidak melakukan aksi *roll*.

2.5. Kinematika Rudal [4]

Kinematika merupakan salah satu cabang mekanika yang mempelajari gerakan suatu objek tanpa mempertimbangkan gaya-gaya yang bekerja pada objek tersebut. Kinematika membahas transformasi posisi, kecepatan atau percepatan suatu objek terhadap referensi tertentu menjadi sistem koordinat lain sehingga transformasi koordinat dibutuhkan dalam mendeskripsikan kinematika. Pada tugas akhir ini, terdapat dua sistem koordinat yang digunakan yaitu sistem koordinat badan rudal dan sistem koordinat bumi.



Gambar 2.6. Sudut Euler pada Rudal [1]

Kinematika rudal dibagi menjadi dua macam yaitu kinematika translasi dan kinematika rotasi rudal. Kinematika translasi merepresentasikan dinamika translasi pada sistem koordinat badan rudal terhadap sistem koordinat bumi yang diwakili oleh transformasi koordinat translasi yang dinyatakan dengan Persamaan (2.6) berikut

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta \ c\psi & s\phi \ s\theta \ c\psi - c\phi \ s\psi & c\phi \ s\theta \ c\psi + s\phi \ s\psi \\ c\theta \ s\psi & s\phi \ s\theta \ s\psi + c\phi \ c\psi & c\phi \ s\theta \ s\psi - s\phi \ c\psi \\ -s\theta & s\phi \ c\theta & c\phi \ c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.6)$$

Dimana:

- $\dot{x}, \dot{y}, \dot{z}$ = kecepatan translasi terhadap koordinat bumi
- u, v, w = kecepatan translasi terhadap koordinat badan
- ϕ, θ, ψ = Sudut Euler yang mewakili rotasi koordinat badan terhadap koordinat bumi

Sedangkan kinematika rotasi adalah kecepatan rotasi rudal pada sistem koordinat badan rudal terhadap sistem koordinat bumi yang dilakukan transformasi rotasi koordinat badan terhadap koordinat bumi yang dinyatakan dengan Persamaan (2.7) berikut.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi \ t\theta & c\phi \ t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.7)$$

2.6. Dinamika Rudal [1],[4]

Dinamika adalah salah satu cabang dari ilmu mekanika yang menjelaskan gerakan suatu objek dengan mempertimbangkan gaya dan massa pada objek tersebut. Dinamika meliputi posisi, kecepatan dan percepatan yang dipengaruhi oleh gaya-gaya yang bekerja pada suatu objek tersebut. Secara umum, dinamika suatu objek digambarkan dengan persamaan yang mematuhi Hukum Newton.

Dinamika pada rudal dibagi menjadi dua macam yaitu dinamika translasi dan dinamika rotasi dimana dinamika translasi adalah persamaan yang menggambarkan gaya-gaya yang menyebabkan perubahan kecepatan atau percepatan translasi dan dinamika rotasi menyatakan momen-momen yang bekerja sehingga rudal dapat mengalami perubahan kecepatan rotasi atau percepatan sudut. Secara matematika kecepatan translasi dan kecepatan sudut rudal bila ditulis dalam vektor dapat dibentuk seperti Persamaan (2.8) dan (2.9)

$$\mathbf{V}_b = u\hat{i} + v\hat{j} + w\hat{k} \quad (2.8)$$

$$\boldsymbol{\omega}_b = p\hat{i} + q\hat{j} + r\hat{k} \quad (2.9)$$

Dimana $\hat{i}, \hat{j}, \hat{k}$ adalah vektor satuan sepanjang sumbu koordinat badan rudal, u, v, w adalah komponen kecepatan translasi rudal pada masing-masing sumbu kartesian x, y , dan z dan p, q, r adalah kecepatan rotasi rudal pada masing-masing sumbu yaitu *roll*, *pitch* dan *yaw*. Magnitudo atau besar kecepatan translasi relatif rudal ditunjukkan persamaan (2.10).

$$V_m = \|\mathbf{V}_b\| = \sqrt{u^2 + v^2 + w^2} \quad (2.10)$$

2.6.1. Dinamika Translasi [1]

Gaya-gaya yang bekerja pada masing-masing sumbu sistem koordinat rudal dinyatakan dalam vektor pada Persamaan (2.11) sebagai berikut.

$$\mathbf{F}_b = F_x\hat{i} + F_y\hat{j} + F_z\hat{k} \quad (2.11)$$

Seperti yang telah disebutkan diawal sebelumnya bahwa persamaan gerak rudal dapat diperoleh menggunakan prinsip Hukum Kedua Newton yang menyatakan bahwa penjumlahan semua gaya eksternal yang bekerja pada badan adalah sama dengan perubahan momentum badan terhadap waktu. Secara matematis, persamaan umum perubahan kecepatan atau

percepatan translasi rudal dalam sistem koordinat badan rudal dapat dirumuskan seperti pada Persamaan (2.12) dan (2.13)

$$\mathbf{F}_b = m\dot{\mathbf{V}}_b + \boldsymbol{\omega}_b \times \mathbf{V}_b + \dot{m}\mathbf{V}_b \quad (2.12)$$

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

Di mana:

\mathbf{V}_b = Kecepatan translasi rudal

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

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

m = massa rudal

\dot{m} = perubahan massa rudal

Persamaan diatas, rumus dengan bentuk $\boldsymbol{\omega}_b \times \mathbf{V}_b$ adalah *cross product* dari kecepatan rotasi dan kecepatan translasi rudal yang dapat dijabarkan menjadi Persamaan (2.14) dan (2.15)

$$\boldsymbol{\omega}_b \times \mathbf{V}_b = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \\ p & q & r \\ u & v & w \end{bmatrix} \quad (2.14)$$

$$\boldsymbol{\omega}_b \times \mathbf{V}_b = (qw - rv)\hat{i} + (ru - pw)\hat{j} + (pv - qu)\hat{k} \quad (2.15)$$

Dengan demikian, percepatan translasi rudal pada masing-masing sumbu sistem koordinat badan rudal dinyatakan pada Persamaan (2.16) - (2.18)

$$\frac{du}{dt} = \frac{F_x - \dot{m}u}{m} + rv - qw \quad (2.16)$$

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

$$\frac{dw}{dt} = \frac{F_z - \dot{m}w}{m} + qu - pv \quad (2.18)$$

Gaya-gaya yang dialami rudal meliputi gaya pendorong pada rudal atau *Thrust*, gaya gravitasi dan gaya aerodinamik. Gaya-gaya ini dapat diketahui sepanjang sistem sumbu koordinat rudal dan tetap pada *center of gravity (cog)* rudal. Model matematika gaya yang bekerja pada masing-masing sumbu pada koordinat rudal dirumuskan pada Persamaan (2.19) - (2.21)

$$F_x = T_m - mg \sin \theta + \bar{q}SC_x \quad (2.19)$$

$$F_y = mg \sin \phi \cos \theta + \bar{q}SC_y \quad (2.20)$$

$$F_z = mg \cos \phi \cos \theta + \bar{q}SC_z \quad (2.21)$$

Dimana:

$$\bar{q} = \frac{1}{2} \rho V_m^2 \quad (2.22)$$

Dimana T adalah gaya pendorong, m adalah massa rudal, g adalah percepatan gravitasi, ϕ adalah sudut *roll* pada rudal, θ adalah sudut *pitch*, S adalah luas permukaan permukaan dari salah satu *airfoil*, \bar{q} adalah tekanan dinamis arus bebas pada *airfoil* yang dinyatakan pada Persamaan (2.22), ρ yaitu kepadatan atmosfer serta C_x , C_y , dan C_z adalah koefisien aerodinamik. Koefisien aerodinamik pada tiap sumbu koordinat badan rudal C_x , C_y , dan C_z merupakan koefisien gaya aerodinamik *drag*, *side force* dan *lift* secara berurutan. *Lift* adalah komponen gaya aerodinamik yang tegak lurus terhadap arah angin. Sedangkan *drag* komponen gaya aerodinamik yang sejajar dengan arah angin dan *side force* adalah gaya aerodinamik yang tegak lurus terhadap gaya *drag* dan *lift*. Koefisien aerodinamik C_x , C_y dan C_z adalah fungsi dari *angle of attack* (α), *sideslip* (β), sudut defleksi *elevator* (δ_e) dan sudut defleksi *rudder* (δ_r). Dengan demikian, persamaan gaya koefisien aerodinamik secara lengkap dapat dirumuskan pada Persamaan (2.23) - (2.25).

$$C_x = C_{x_0} \quad (2.23)$$

$$C_y = C_{y\beta} \beta + C_{y\delta_r} \delta_r \quad (2.24)$$

$$C_z = C_{z\alpha} \alpha + C_{z\delta_e} \delta_e \quad (2.25)$$

Angle of attack atau sudut serang (α) dan *sideslip* atau sudut *slip* (β) dapat ditentukan oleh komponen kecepatan rudal. Sudut serang adalah sudut yang dibentuk antara garis referensi pada badan rudal yang biasanya berada pada garis chord di *airfoil* dengan vektor kecepatan udara relatif. Sudut *slip* adalah sudut yang dibentuk oleh vektor kecepatan terhadap sumbu longitudinal pada pusat massa rudal. Secara matematis, persamaan untuk sudut-sudut ini diberikan pada Persamaan (2.26) dan (2.27)

$$\alpha = \tan^{-1} \frac{w}{u} \quad (2.26)$$

$$\beta = \sin^{-1} \frac{v}{\sqrt{u^2 + v^2 + w^2}} \quad (2.27)$$

Percepatan rudal dalam sistem koordinat bumi merupakan perbandingan antara gaya yang bekerja pada rudal terhadap massanya tanpa adanya pengaruh perkalian silang antara kecepatan translasi dan kecepatan rotasi rudal. Persamaan percepatan rudal dalam sistem koordinat bumi dirumuskan pada Persamaan (2.28)-(2.30).

$$a_x = \frac{F_x - \dot{m}u}{m} \quad (2.28)$$

$$a_y = \frac{F_y - \dot{m}v}{m} \quad (2.29)$$

$$a_z = \frac{F_z - \dot{m}w}{m} \quad (2.30)$$

Sehingga jika digabung Persamaan (2.16) - (2.27), maka didapatkan model kompak dari percepatan translasi dalam sistem koordinat badan rudal secara lengkap dituliskan pada Persamaan (2.31) - (2.33).

$$\frac{du}{dt} = \frac{T_m - mg \sin \theta + \bar{q}S C_{x_0} - \dot{m}u}{m} + rv - qw \quad (2.31)$$

$$\frac{dv}{dt} = \frac{mg \sin \phi \cos \theta + \bar{q}S (C_{y_\beta} \beta + C_{y_{\delta_r}} \delta_r) - \dot{m}v}{m} + pw - ru \quad (2.32)$$

$$\frac{dw}{dt} = \frac{mg \cos \phi \cos \theta + \bar{q}S (C_{z_\alpha} \alpha + C_{z_{\delta_e}} \delta_e) - \dot{m}w}{m} + qu - pv \quad (2.33)$$

2.6.2. Dinamika Rotasi [1]

Momen-momen rotasi yang bekerja pada tiap sumbu sistem koordinat rudal dinyatakan dalam vektor dirumuskan pada Persamaan (2.34) sebagai berikut

$$\mathbf{M}_b = M_x \hat{i} + M_y \hat{j} + M_z \hat{k} \quad (2.34)$$

Sama dengan prinsip pada dinamika translasi, dinamika rotasi dipengaruhi oleh perubahan momentum sudut terhadap waktu. Secara matematis, persamaan umum dinamika rotasi rudal pada sistem koordinat rudal dirumuskan dalam Persamaan (2.35) dan (2.36)

$$\mathbf{M}_b = \mathbf{I} \dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \mathbf{I} \boldsymbol{\omega}_b + \dot{\mathbf{I}} \boldsymbol{\omega}_b \quad (2.35)$$

$$\frac{d\omega_b}{dt} = I^{-1}(M_b - \omega_b \times I\omega_b - \dot{I}\omega_b) \quad (2.36)$$

Dimana :

I = Momen inersia rudal

M_b = Momen rotasi pada rudal

ω_b = Kecepatan sudut rotasi rudal

Gerak rotasi pada suatu objek tidak akan terlepas pada momen inersia yang dimiliki pada objek tersebut sehingga momen inersia I pada rudal dilibatkan dalam dinamika rotasi. Secara umum, matriks momen inersia dinyatakan dengan matriks seperti pada Persamaan (2.37)

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} \int (y^2 + z^2) dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int (x^2 + z^2) dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (y^2 + x^2) dm \end{bmatrix} \quad (2.37)$$

Dimana elemen diagonal matriks momen inersia didefinisikan sebagai momen inersia aksial dan komponen matriks non-diagonal adalah produk inersia. Diasumsikan bahwa rudal adalah objek yang bidangnya simetris pada bidang xy , xz dan yz dan menempatkan sistem sumbu badan rudal dengan titik yang tepat maka dapat dinyatakan bahwa $I_{xy} = I_{yx} = I_{yz} = I_{zy} = I_{xz} = I_{zx} = 0$. Dengan pernyataan tersebut, matriks inersia untuk rudal disederhanakan menjadi Persamaan (2.38)

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.38)$$

Dengan menggunakan pendekatan objek tabung, matriks inersia tabung dapat dihitung dengan rumus pada Persamaan (2.39)

$$I = \begin{bmatrix} \frac{1}{8}md^2 & 0 & 0 \\ 0 & \frac{1}{12}ml^2 + \frac{1}{16}md^2 & 0 \\ 0 & 0 & \frac{1}{12}ml^2 + \frac{1}{16}md^2 \end{bmatrix} \quad (2.39)$$

Dimana:

m = Massa tabung

l = Panjang tabung

d = Diameter tabung

Dari Persamaan (2.36) diatas, rumus $\omega_b \times I\omega_b$ dapat dijabarkan menjadi Persamaan (2.40) - (2.42)

$$\omega_b \times I\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.40)$$

$$\omega_b \times I\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \\ p & q & r \\ I_{xx}p & I_{yy}q & I_{zz}r \end{bmatrix} \quad (2.41)$$

$$\omega_b \times I\omega_b = \begin{bmatrix} (I_{zz} - I_{yy})qr \hat{i} \\ (I_{xx} - I_{zz})pr \hat{j} \\ (I_{xx} - I_{yy})pq \hat{k} \end{bmatrix} \quad (2.42)$$

Momen yang bekerja pada sumbu referensi badan rudal dibagi menjadi tiga yaitu momen *rolling* (L), momen *pitching* (M), dan momen *yawing* (N). Momen *rolling* disebabkan oleh perbedaan gaya angkat (*lift*) pada *aileron* di sumbu longitudinal rudal. Momen *roll* positif menyebabkan rudal berguling ke kanan. Sedangkan momen *pitching* adalah momen yang dihasilkan dari gaya angkat (*lift*) dan gaya seret (*drag*) yang bekerja pada sumbu lateral rudal. Momen *pitching* positif menyebabkan hidung rudal mengarah ke atas. Momen *yawing* terjadi pada sumbu vertikal pada rudal yang mana nilai positif momen menyebabkan hidung rudal berotasi ke kanan. Momen rotasi yang bekerja pada rudal dipengaruhi oleh momen aerodinamik. Sama seperti gaya aerodinamik, momen pada rudal dapat dibagi menjadi momen yang diciptakan oleh distribusi beban aerodinamik dan gaya dorong yang tidak bekerja pada *center of gravity* (*cog*). Komponen momen aerodinamik dapat dinyatakan sebagai Persamaan (2.43) - (2.45) berikut.

$$M_x = \bar{q}SdC_l \quad (2.43)$$

$$M_y = \bar{q}SdC_m \quad (2.44)$$

$$M_z = \bar{q}SdC_n \quad (2.45)$$

Di mana:

C_l = Koefisien momen aerodinamik pada sumbu x

C_m = Koefisien momen aerodinamik pada sumbu y

C_n = Koefisien momen aerodinamik pada sumbu z

Koefisien momen aerodinamik C_l , C_m dan C_n merupakan fungsi dari *angle of attack* (α), *sideslip* (β), kecepatan rotasi (ω_p), sudut defleksi *aileron* (δ_a), sudut defleksi *elevator* (δ_e) dan sudut defleksi *rudder* (δ_r). Persamaan matematik koefisien aerodinamik rudal dirumuskan pada Persamaan (2.46) - (2.48)

$$C_l = C_{l\delta_a} \delta_a + C_{l_p} \frac{pd}{2V_m} \quad (2.46)$$

$$C_m = C_{m\alpha} \alpha + C_{m\delta_e} \delta_e + C_{m_q} \frac{qd}{2V_m} \quad (2.47)$$

$$C_n = C_{n\beta} \beta + C_{n\delta_r} \delta_r + C_{n_r} \frac{rd}{2V_m} \quad (2.48)$$

Dengan menggabungkan Persamaan (2.34) - (2.48), model matematika kompak dari dinamika rotasi rudal dalam sistem koordinat badan secara lengkap dinyatakan dengan Persamaan (2.49) - (2.51) sebagai berikut

$$\frac{dp}{dt} = \frac{1}{I_{xx}} \left(\bar{q} S d \left(C_{l\delta_a} \delta_a + C_{l_p} \frac{pd}{2V_m} \right) - i_{xx} p - (I_{zz} - I_{yy}) q r \right) \quad (2.49)$$

$$\begin{aligned} \frac{dq}{dt} = \frac{1}{I_{yy}} \left(\bar{q} S d \left(C_{m\alpha} \alpha + C_{m\delta_e} \delta_e + C_{m_q} \frac{qd}{2V_m} \right) - i_{yy} q \right. \\ \left. - (I_{xx} - I_{zz}) p r \right) \end{aligned} \quad (2.50)$$

$$\begin{aligned} \frac{dr}{dt} = \frac{1}{I_{zz}} \left(\bar{q} S d \left(C_{n\beta} \beta + C_{n\delta_r} \delta_r + C_{n_r} \frac{rd}{2V_m} \right) - i_{zz} r \right. \\ \left. - (I_{yy} - I_{xx}) p q \right) \end{aligned} \quad (2.51)$$

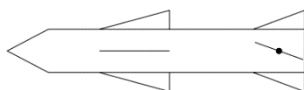
2.7. Autopilot Rudal [1],[3],[11]

Autopilot adalah sebuah sistem loop tertutup didalam sub sistem pemanduan utama yang menjaga rudal agar tetap menjaga kestabilannya mendapatkan performa terbang rudal yang sesuai dengan yang diperintahkan melalui pemandu. Sistem autopilot menjaga rudal pada

attitude, *heading* dan *altitude* secara otomatis agar meminimumkan usaha pilot. Fungsi autopilot adalah untuk menyetabilkan dan memandu rudal dengan sudut defleksi yang diminta yang menyebabkan badan rudal berotasi.

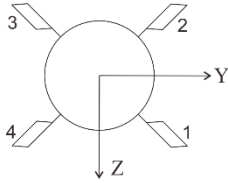
Untuk meminimumkan kegagalan jarak pada fase *terminal*, dinamika dari sistem pengaturan harus secepat mungkin. Oleh karena itu, *time constant* dari autopilot harus dapat sekecil mungkin. Hampir seluruh sistem pengaturan pada rudal, variabel yang dikendalikan disesuaikan dengan tipe dari perintah yang dihasilkan oleh pemanduan seperti kecepatan sudut (p, q, r) , sudut orientasi rudal (ϕ, θ, ψ) , kecepatan translasi (u, v, w) atau percepatannya. Kemudian, *servo* pada sirip merespon perintah yang diberikan oleh autopilot dan defleksi aktual dari sirip dihitung dengan memperhitungkan keseimbangan antara torsi motor *servo* dan momen aerodinamik.

Dalam tugas akhir ini, model letak sirip yang digunakan adalah jenis *tail*. Seperti namanya, permukaan kendali diletakkan pada ekor rudal seperti pada Gambar 2.7. Keuntungan pertama dari model sirip ini dibandingkan dengan jenis *canard* dan *wing* adalah karakteristik aerodinamik yang lebih linier karena tidak adanya *downwash effect*. Tetapi, ruang yang tersedia untuk meletakkannya terbatas dikarenakan terdapat motor roket dan saluran pembuangan. Diluar ketidakuntungan jenis sirip ini, yang terpenting adalah jenis rudal ini memiliki reaksi pertama yang berlawanan dengan perintah gerakannya. Ini adalah hasil dari karakteristik *non-minimum phase* dari rudal yang digerakkan dengan sirip ekor.



Gambar 2.7. Rudal dengan model sirip *tail* [3]

Secara umum, jumlah sirip pada rudal berjumlah empat dengan dua konfigurasi yang dapat dibentuk yaitu konfigurasi *cross* dan *plus*. Pada konfigurasi *plus*, sirip beban kerja sirip rudal terbagi menjadi dua, dua sirip untuk mengendalikan sudut *pitch* rudal, dan dua sirip lainnya menggerakkan sudut *yaw* rudal. Sedangkan pada konfigurasi *cross*, beban kerja sirip terbagi rata pada setiap sirip pada rudal. Pada tugas akhir ini, konfigurasi sirip yang digunakan adalah konfigurasi *cross* seperti pada Gambar 2.8.



Gambar 2.8. Sirip Ekor Rudal Tipe *Cross*

Besarnya defleksi dari masing-masing sirip pada rudal akan menentukan gaya dan momen aerodinamika rudal sehingga orientasi dan kecepatan rudal dapat berubah. Besarnya defleksi masing-masing sirip terhadap sudut defleksi rudal jenis sirip *cross* dirumuskan dengan Persamaan (2.52).

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \quad (2.52)$$

Diasumsikan bahwa respon gerak *servo* disederhanakan menjadi sistem dengan respons orde dua. Hubungan antara defleksi sudut yang diperintah terhadap sudut defleksi aktual dirumuskan pada Persamaan (2.53).

$$\delta_{actual} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \delta_{cmd} \quad (2.53)$$

Dimana δ_{cmd} adalah sudut defleksi $\delta_1, \delta_2, \delta_3, \delta_4$ yang diperintahkan dan δ_{actual} adalah respon sudut defleksi $\delta_1, \delta_2, \delta_3, \delta_4$ yang sebenarnya.

2.8. *Nonlinear Feedback Linearization* [5],[12]

Pada pengendalian rudal, terdapat banyak variabel keluaran dan variabel yang harus dikendalikan sehingga dinamika rudal dapat dilihat sebagai sistem *Multi input multi output (MIMO)*. Selain itu terlihat bahwa persamaan matematika rudal adalah persamaan non-linier dan tentu menjadi *coupled system* yaitu adanya keterkaitan antar masing-masing variabel sistem sehingga akan menjadi lebih rumit dalam merancang pengendaliannya.

Dalam merancang pengendalian rudal yang memiliki dinamika non-linier dan *coupled* ini, salah satu pendekatan umum yang biasanya untuk mengatasi sistem non-linier yaitu dengan cara melinierisasikan

persamaan non-linier rudal dengan menggunakan metode *nonlinear feedback linearization*. Teknik ini mentransformasikan persamaan non-linier menjadi sistem linier melalui mengumpukan balikkan seluruh *state* dari rudal dan mencocokkan sinyal input yang sesuai dengan perubahan variabel-variabel sistem. *Nonlinear feedback linearization* sangat bergantung pada informasi parameter dan variabel sistem yang akurat untuk dapat menghasilkan sistem linier yang diinginkan. Setelah sistem telah menjadi linier dengan teknik *nonlinear feedback linearization*, maka pengendali rudal dapat didesain dengan pengendali linier. Suatu sistem non-linier dapat dinyatakan dengan persamaan umum pada Persamaan (2.54)

$$\begin{aligned}\dot{x} &= f(x(t), u(t)) + g(x(t))u(t) \\ y &= h(x(t))\end{aligned}\tag{2.54}$$

Di mana:

$$\begin{aligned}f(x(t)) &= \text{Persamaan } \textit{state} \text{ non-linier} \\ g(x(t)) &= \text{Persamaan masukan non linier} \\ h(x(t)) &= \text{Persamaan keluaran sistem}\end{aligned}$$

Ditentukan bahwa suatu sistem linier yang diinginkan ditentukan pada Persamaan (2.55) yaitu

$$\begin{aligned}\dot{x} &= ax(t) + bu(t)^* \\ y(t) &= cx(t)\end{aligned}\tag{2.55}$$

Dimana a adalah koefisien sistem linier yang diinginkan, b adalah koefisien masukan dan c adalah koefisien keluaran sistem linier yang baru serta $u(t)^*$ adalah masukan baru yang diinginkan. Dengan demikian dapat dibentuk suatu persamaan *state* yang baru dengan bentuk Persamaan (2.56)

$$\dot{x} = f(x(t), u(t)) + g(x(t))u(t) + ax + bu^* - ax - bu^*\tag{2.56}$$

Persamaan (2.56) dapat diatur ulang menjadi Persamaan (2.57)

$$\begin{aligned}\dot{x} &= ax(t) + bu(t)^* + f(x(t), u(t)) + g(x(t))u(t) - \dots \\ &\dots ax(t) - bu(t)^*\end{aligned}\tag{2.57}$$

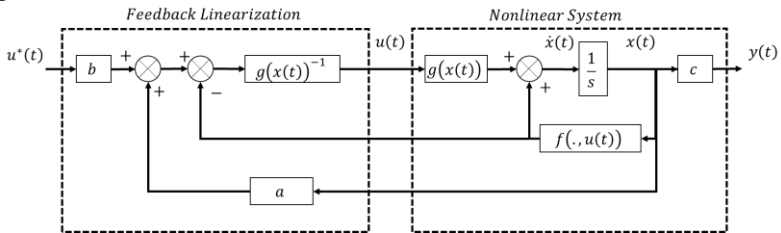
Agar Persamaan (2.57) diatas menjadi seperti Persamaan (2.55) maka Persamaan (2.58) harus dapat dipenuhi menjadi

$$f(x(t), u(t)) + g(x(t))u(t) - ax(t) - bu(t)^* = 0\tag{2.58}$$

Dengan ketentuan bahwa *inverse* atau *pseudoinverse* dari matriks $g(x(t))$ adalah *nonsingular*, maka didapatkan masukkan $u(t)$ untuk sistem non-linier agar menjadi sistem linier didefinisikan dengan Persamaan (2.59)

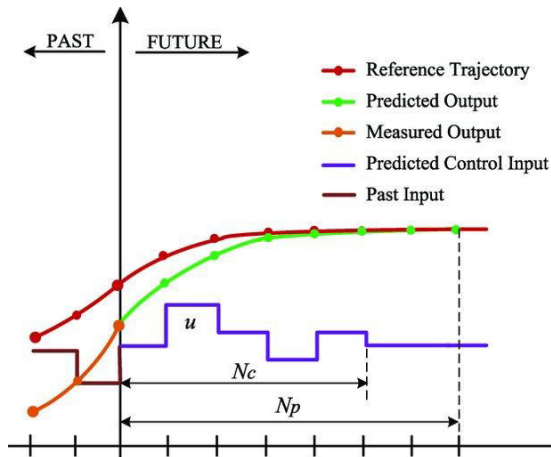
$$u(t) = g(x(t))^{-1} (ax(t) + bu(t)^* - f(x(t), u(t))) \quad (2.59)$$

Diagram blok sistem linier dengan menggunakan teknik *nonlinear feedback linearization* diturunkan dari Persamaan (2.59) digambarkan pada Gambar 2.9.



Gambar 2.9. Diagram Blok Nonlinear Feedback Linearization

2.9. Discrete Model Predictive Control [6],[8]



Gambar 2.10. Pendekatan Discrete Model predictive control [8]

DMPC merupakan algoritma pengaturan umpan balik yang menggunakan prediksi keluaran proses yang akan datang. DMPC menggunakan model sistem untuk membuat prediksi dari perilaku sistem yang akan datang. DMPC melakukan proses algoritma optimasi secara online untuk mendapatkan aksi kendali yang optimal yang menggerakkan keluaran yang diprediksi sampai ke referensi. Karena prediksi terus bergerak ke depan, maka DMPC juga disebut dengan *receding horizon control*. Beberapa alasan terkait dengan keberhasilannya pada aplikasi ini antara lain mampu mengatasi permasalahan pengaturan *multivariable* atau sistem *multi input multi output* (MIMO) yang mungkin memiliki hubungan antara masukkan dan keluarannya, mampu memerhitungkan batasan pada aktuator dan mampu melihat gambaran referensi yang akan datang menjadi permasalahan pengaturan untuk meningkatkan performa dari pengendali. Tujuan desain umum dari DMPC adalah untuk menghitung trayektori variabel yang dimanipulasi yang akan datang untuk mengoptimalkan perilaku yang akan datang dari sistem.

Sistem *Linear Time-Invariant* (LTI) dengan model *state-space* diskrit dinyatakan dalam Persamaan (2.60) dan (2.61)

$$\mathbf{x}_m(k+1) = \mathbf{A}_m \mathbf{x}_m(k) + \mathbf{B}_m \mathbf{u}(k) \quad (2.60)$$

$$\mathbf{y}(k) = \mathbf{C}_m \mathbf{x}_m(k) \quad (2.61)$$

Di mana:

$\mathbf{x}_m(k)$ = Vektor state variable dari sistem

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

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

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

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

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

Dengan menggunakan operasi selisih pada *state variable* dan selisih dari variabel sinyal kendali masukan didapatkan Persamaan (2.62) dan (2.63).

$$\Delta \mathbf{x}_m(k+1) = \mathbf{x}_m(k+1) - \mathbf{x}_m(k) \quad (2.62)$$

$$\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1) \quad (2.63)$$

Sehingga didapatkan persamaan *state-space* selisih dinyatakan dalam Persamaan (2.64).

$$\Delta \mathbf{x}_m(k+1) = \mathbf{A}_m \Delta \mathbf{x}_m(k) + \mathbf{B}_m \Delta \mathbf{u}(k) \quad (2.64)$$

Perhatikan bahwa input pada model *state-space* adalah $\Delta \mathbf{u}(k)$. Langkah selanjutnya adalah menghubungkan $\Delta \mathbf{x}(k)$ dengan keluaran $\mathbf{y}(k)$. Dengan begitu vektor *state variable* yang baru menjadi Persamaan (2.65)

$$\mathbf{x}(k) = [\Delta \mathbf{x}_m(k)^T \quad \mathbf{y}(k)^T]^T \quad (2.65)$$

Dimana superskrip T mengindikasikan transpos matriks. Perlu dicatat bahwa

$$\mathbf{y}(k+1) - \mathbf{y}(k) = \mathbf{C}_m \mathbf{A}_m \Delta \mathbf{x}_m(k) + \mathbf{C}_m \mathbf{B}_m \Delta \mathbf{u}(k) \quad (2.66)$$

Menggabungkan Persamaan (2.65) dan (2.66) menjadikan sebuah model persamaan *state-space* (2.67) dan (2.68) berikut.

$$\begin{array}{c} \mathbf{x}(k+1) \qquad \qquad \qquad A \qquad \qquad \qquad \mathbf{x}(k) \qquad \qquad \qquad B \\ \overbrace{\begin{bmatrix} \Delta \mathbf{x}_m(k+1) \\ \mathbf{y}(k+1) \end{bmatrix}}^C = \overbrace{\begin{bmatrix} \mathbf{A}_m & \mathbf{0}_m^T \\ \mathbf{C}_m \mathbf{A}_m & 1 \end{bmatrix}}^A \overbrace{\begin{bmatrix} \Delta \mathbf{x}_m(k) \\ \mathbf{y}(k) \end{bmatrix}}^{\mathbf{x}(k)} + \overbrace{\begin{bmatrix} \mathbf{B}_m \\ \mathbf{C}_m \mathbf{B}_m \end{bmatrix}}^B \Delta \mathbf{u}(k) \end{array} \quad (2.67)$$

$$\mathbf{y}(k) = \overbrace{\begin{bmatrix} \mathbf{0}_m & 1 \end{bmatrix}}^C \overbrace{\begin{bmatrix} \Delta \mathbf{x}_m(k) \\ \mathbf{y}(k) \end{bmatrix}}^{\mathbf{x}(k)} \quad (2.68)$$

Dimana $\mathbf{0}_m = [0 \ 0 \ \dots \ 0]$. Matriks A, B, C disebut dengan *augmented model* yang akan digunakan dalam mendesain pengaturan prediktif.

Setelah mendapatkan formulasi model matematika yang kompak, tahap selanjutnya adalah mengalkulasi keluaran sistem yang akan datang dengan sinyal kendali yang akan datang sebagai variabel yang dicari. Prediksi ini kemudian digambarkan pada satu jendela optimisasi. Asumsikan trayektori sinyal kendali dan variabel *state* yang akan datang dinotasikan dalam Persamaan (2.69) - (2.71)

$$\Delta \mathbf{U} = \Delta \mathbf{u}(k_i), \Delta \mathbf{u}(k_i + 1), \dots, \Delta \mathbf{u}(k_i + N_c - 1) \quad (2.69)$$

$$\mathbf{x}(k_i) = \mathbf{x}(k_i + 1|k_i), \mathbf{x}(k_i + 2|k_i), \dots, \mathbf{x}(k_i + N_p|k_i) \quad (2.70)$$

$$\mathbf{y}(k_i) = \mathbf{y}(k_i + 1|k_i), \mathbf{y}(k_i + 2|k_i), \dots, \mathbf{y}(k_i + N_p|k_i) \quad (2.71)$$

Dimana N_c adalah jumlah horison sinyal kendali yang akan datang dan N_p adalah jumlah horison *state* yang akan datang. Berdasarkan model *state-space*, variabel *state* dan keluaran sistem yang akan datang dapat

dikalkulasikan secara sekuensial menggunakan parameter kendali yang akan datang yaitu dinyatakan pada Persamaan (2.72) dan (2.73)

$$\begin{aligned} \mathbf{x}(k_i + 1|k_i) &= \mathbf{A}\mathbf{x}(k_i) + \mathbf{B}\Delta\mathbf{u}(k) \\ \mathbf{x}(k_i + 2|k_i) &= \mathbf{A}^2\mathbf{x}(k_i) + \mathbf{A}\mathbf{B}\Delta\mathbf{u}(k_i) + \mathbf{B}\Delta\mathbf{u}(k_i + 1) \\ &\vdots \end{aligned} \quad (2.72)$$

$$\begin{aligned} \mathbf{x}(k_i + N_p|k_i) &= \mathbf{A}^{N_p}\mathbf{x}(k_i) + \mathbf{A}^{N_p-1}\mathbf{B}\Delta\mathbf{u}(k_i) \\ &\quad + \dots + \mathbf{A}^{N_p-N_c}\mathbf{B}\Delta\mathbf{u}(k_i + N_c - 1) \\ \mathbf{y}(k_i + 1|k_i) &= \mathbf{C}\mathbf{A}\mathbf{x}(k_i) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k_i) \\ \mathbf{y}(k_i + 2|k_i) &= \mathbf{C}\mathbf{A}^2\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}\mathbf{B}\Delta\mathbf{u}(k_i) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k_i + 1) \\ &\vdots \\ \mathbf{y}(k_i + N_p|k_i) &= \mathbf{C}\mathbf{A}^{N_p}\mathbf{x}(k_i) + \mathbf{C}\mathbf{A}^{N_p-1}\mathbf{B}\Delta\mathbf{u}(k_i) + \\ &\quad \mathbf{C}\mathbf{A}^{N_p-2}\mathbf{B}\Delta\mathbf{u}(k_i + 1) + \\ &\quad \mathbf{C}\mathbf{A}^{N_p-N_c}\mathbf{B}\Delta\mathbf{u}(k_i + N_c - 1) \end{aligned} \quad (2.73)$$

Dengan menggabungkan persamaan diatas, didapatkan bentuk matriks kompak dengan \mathbf{F} sebagai matriks sistem baru dan $\boldsymbol{\phi}$ sebagai matriks masukan yang baru dan dinyatakan sebagai Persamaan baru (2.74)

$$\mathbf{Y} = \mathbf{F}\mathbf{x}(k_i) + \boldsymbol{\phi}\Delta\mathbf{U} \quad (2.74)$$

Dimana:

$$\mathbf{F} = \begin{bmatrix} \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \mathbf{C}\mathbf{A}^3 \\ \vdots \\ \mathbf{C}\mathbf{A}^{N_p} \end{bmatrix} \quad (2.75)$$

$$\boldsymbol{\phi} = \begin{bmatrix} \mathbf{C}\mathbf{B} & 0 & 0 & \dots & 0 \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & 0 & \dots & 0 \\ \mathbf{C}\mathbf{A}^2\mathbf{B} & \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \dots & 0 \\ \vdots & & & & \\ \mathbf{C}\mathbf{A}^{N_p-1}\mathbf{B} & \mathbf{C}\mathbf{A}^{N_p-2}\mathbf{B} & \mathbf{C}\mathbf{A}^{N_p-3}\mathbf{B} & \dots & \mathbf{C}\mathbf{A}^{N_p-N_c}\mathbf{B} \end{bmatrix} \quad (2.76)$$

Diberikan sinyal referensi $r(k)$ pada waktu sampel k_i , di dalam sebuah horison prediksi tujuan pada sistem pengaturan prediktif adalah untuk membawa keluaran sedekat mungkin dengan sinyal referensi yang diberikan, dimana kita asumsikan bahwa sinyal referensi tetap konstan dalam satu jendela optimisasi. Tujuan ini kemudian diterjemahkan ke dalam desain untuk menemukan vektor parameter kendali terbaik $\Delta\mathbf{U}$ sehingga kesalahan keluaran prediksi dapat diminimumkan.

Didefinisikan fungsi tujuan yang mencerminkan tujuan kendali sebagai Persamaan (2.77) berikut

$$J = (\mathbf{R}_s - \mathbf{Y})^T (\mathbf{R}_s - \mathbf{Y}) + \Delta \mathbf{U}^T \bar{\mathbf{R}} \Delta \mathbf{U} \quad (2.77)$$

Dimana istilah pertama menghubungkan dengan tujuan untuk meminimumkan kesalahan antara keluaran prediksi dan sinyal referensi sementara yang kedua mencerminkan bobot $\Delta \mathbf{U}$ ketika fungsi tujuan J dibuat sekecil mungkin. $\bar{\mathbf{R}}$ adalah matriks diagonal dalam dengan bentuk $\bar{\mathbf{R}} = r_w \mathbf{I}_{N_c \times N_c}$ dimana r_w digunakan untuk menentukan parameter penyeteman performansi loop tertutup yang diinginkan dengan nilai $r_w \geq 0$. Untuk menemukan $\Delta \mathbf{U}$ optimal yang akan meminimumkan J , J dinyatakan sebagai Persamaan (2.78)

$$J = (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i))^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) - 2\Delta \mathbf{U}^T \boldsymbol{\phi}^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) + \Delta \mathbf{U}^T (\boldsymbol{\phi}^T \boldsymbol{\phi} + \bar{\mathbf{R}}) \Delta \mathbf{U} \quad (2.78)$$

Agar didapatkan fungsi J yang minimum maka dicari turunan parsial terhadap variabel $\Delta \mathbf{U}$. Dari turunan pertama pada fungsi tujuan J dan kondisi minimum yang diinginkan dinyatakan sebagai Persamaan (2.79)

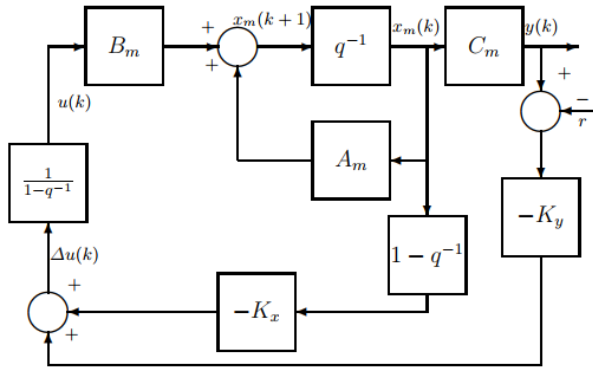
$$\begin{aligned} \frac{\partial J}{\partial \Delta \mathbf{U}} &= -2\boldsymbol{\phi}^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) + 2(\boldsymbol{\phi}^T \boldsymbol{\phi} + \bar{\mathbf{R}}) \Delta \mathbf{U} \\ \frac{\partial J}{\partial \Delta \mathbf{U}} &= 0 \end{aligned} \quad (2.79)$$

Sehingga didapatkan solusi optimal untuk sinyal kendali dinyatakan sebagai Persamaan (2.80) berikut

$$\Delta \mathbf{U} = (\boldsymbol{\phi}^T \boldsymbol{\phi} + \bar{\mathbf{R}})^{-1} \boldsymbol{\phi}^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) \quad (2.80)$$

Bila diperhatikan lebih teliti, persamaan diatas memiliki dua parameter yaitu $(\boldsymbol{\phi}^T \boldsymbol{\phi} + \bar{\mathbf{R}})^{-1} \boldsymbol{\phi}^T \mathbf{R}_s$ berhubungan dengan perubahan nilai referensi, sementara $-(\boldsymbol{\phi}^T \boldsymbol{\phi} + \bar{\mathbf{R}})^{-1} \boldsymbol{\phi}^T \mathbf{F}$ berkaitan dengan pengaturan *state feedback* atau vektor penguatan *state feedback* dalam pengaturan prediktif. Karena prinsip dari *receding horizon control*, kita hanya mengambil elemen pertama dari $\Delta \mathbf{U}$ pada waktu k_i sebagai pengaturan tambahan, maka didapatkan Persamaan (2.81)

$$\begin{aligned} \Delta u(k_i) &= [1 \ 0 \ \dots \ 0] (\boldsymbol{\phi}^T \boldsymbol{\phi} + \bar{\mathbf{R}})^{-1} \boldsymbol{\phi}^T (\mathbf{R}_s - \mathbf{F}\mathbf{x}(k_i)) \\ &= \mathbf{K}_y \mathbf{r}(k_i) - \mathbf{K}_{mpc} \mathbf{x}(k_i) \end{aligned} \quad (2.81)$$



Gambar 2.11. Blok Diagram Alir *Model predictive control* [6]

2.10. *Path Planning* [9]

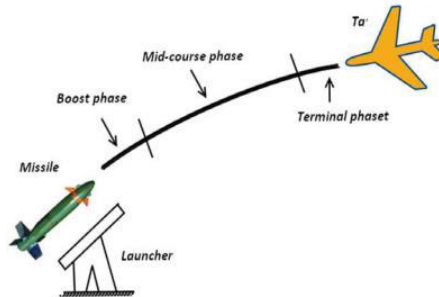
Path planning adalah penentuan lintasan yang harus dilalui rudal untuk melewati titik yang telah ditentukan dalam ruang yang sering disebut dengan *waypoint*. Secara umum, permasalahan *path planning* adalah menentukan bentuk lintasan yang menghubungkan *waypoint* tersebut dan menghasilkan lintasan terpendek atau optimal yang dikenal dengan *path generation*.

Permasalahan kemampuan kendaraan agar dapat melintasi lintasan yang melintasi *waypoint* yang telah ditentukan disebut dengan *path following*. Sistem ini merupakan salah satu metode yang digunakan oleh kendaraan tanpa awak untuk bergerak secara independen. Berbeda dengan *trajectory tracking* dimana terdapat parameter waktu dalam melintasi lintasan, *path following* hanya ada lintasan referensi yang harus dilalui berupa titik dan garis dalam ruang.

Ada dua pendekatan yang bisa dipertimbangkan sebagai solusi untuk permasalahan *path following* pada pesawat. Metode pertama yaitu dengan memisahkan pemanduan dan pengendalian kendaraan menjadi pemanduan lup luar dan pengendalian lup dalam. Metode alternatif lainnya yaitu pendekatan penggabungan yaitu menggabungkan lup luar dan lup dalam didesain secara bersamaan. Namun, pada aplikasi dunia penerbangan yang sebenarnya, pendekatan lup luar dan lup dalam secara terpisah lebih umum digunakan karena lebih sederhana dan metode desain pengaturan loop dalam yang lebih baik.

2.11. Pemanduan Rudal [1],[3],[7]

Pemanduan pada rudal dapat didefinisikan sebagai strategi mengemudi rudal menuju sasaran, sementara pengaturan dapat didefinisikan taktik dari menggunakan penggerak rudal untuk mengimplementasikan strategi yang perintahkan oleh unit pemanduan. Fungsi utama dari sistem pemandu adalah merasakan, mengolah informasi dan membetulkan perintah kepada rudal.



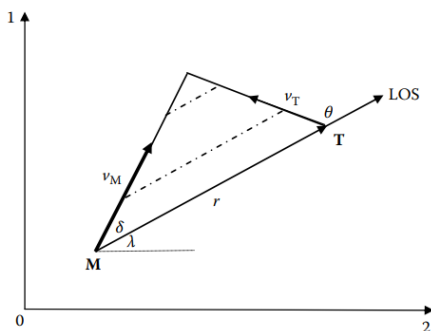
Gambar 2.12. Tiga Fase Terbang Rudal [7]

Dalam pemanduan, penerbangan rudal dapat dibagi menjadi beberapa fase yaitu *boost phase*, *midcourse phase* dan *terminal phase*. *Boost phase* mencakup penerbangan rudal dari tembakan pertama hingga akhir dari pendorong. *Midcourse phase* mulai setelah *boost phase* dan bekerja hingga sasaran dapat dideteksi oleh *seeker*. Sebenarnya, durasi dari fase ini bergantung pada deteksi sasarannya. Strategi pemanduan *midcourse* biasanya diaplikasikan pada rudal jarak jauh. Setelah target dideteksi oleh *seeker*, *terminal guidance phase* akan dimulai. Dalam fase ini, informasi keadaan sasaran didapatkan oleh *seeker*, rudal akan menuju sasarannya dengan mempertimbangkan hukum pemandu dan rudal kemudian mencoba mengikuti sasaran hingga mengenainya sesuai rencana. Tiga fase pemanduan rudal digambarkan pada Gambar 2.12.

2.11.1. Line of sight (LOS) [13]

Line of sight (LOS) didefinisikan sebagai sudut antara garis dari pusat antena *seeker* dengan sasaran. Untuk dapat melihat sebuah objek, rudal harus memandang sepanjang garis pada objek tersebut. LOS adalah konsep yang penting dalam pemanduan. Orientasinya terhadap sistem

koordinat referensi memungkinkan untuk memformulasikan aturan pemanduan dengan presisi. Salah satu tipe pemanduan yang menggunakan LOS yaitu pemanduan *Command to LOS* (CLOS). CLOS adalah jenis tipe pemanduan dan navigasi dimana rudal selalu diperintahkan berada pada garis lihatnya antara satuan yang akan disasar atau biasa disebut juga dengan pemanduan tiga titik. Pemanduan jenis CLOS hampir digunakan oleh semua jenis rudal pertahanan jarak pendek.



Gambar 2.13. Geometri Planar Sudut LOS [7]

Berdasarkan Gambar 2.13 , sudut LOS λ diukur terhadap garis referensi 02. Sudut δ pada Gambar 2.13 disebut dengan *lead angle* dan sudut $180^\circ - \theta$ disebut dengan *aspect angle*. Garis putus-putus menunjukkan posisi rudal M dan sasaran T berdasarkan hukum *parallel navigation*. Segitiga pada gambar diatas disebut dengan *collision triangle*.

2.11.2. *Nonlinear Guidance Logic* (NLGL) [9],[10]

Nonlinear guidance logic adalah salah satu pemanduan untuk *path following* terumum untuk *unmanned aerial vehicles* (UAVs), terutama karena kesederhanaannya dan kemampuannya untuk menelusuri garis lurus maupun lingkaran. Konsep ini juga telah membuat perkembangan yang lebih luas pada perencanaan terbang pesawat UAV. Ada beberapa pemandu *terminal phase* untuk rudal taktis jarak pendek yang dapat dilakukan untuk mengikuti lintasan dengan menggunakan titik bayangan sepanjang lintasan terbang yang diinginkan sebagai sasaran semu. Ini, pemandu *proportional navigation* memberikan performa terbaik dengan

usaha pengaturan yang minimum, dan secara umum diterima sebagai metode pemandu. Pemanduan NLGL menggunakan metode yang sama dengan metode *proportional navigation*, namun elemen pentingnya yaitu kecepatan sasaran semu tidak dipertimbangkan karena lintasan yang akan ditelusuri statis. Pemanduan NLGL dilakukan dengan memilih titik referensi pada lintasan yang diinginkan dan menghasilkan perintah percepatan lateral menggunakan titik referensi tersebut. Titik referensi berada pada lintasan yang diinginkan pada jarak L_1 berada di depan rudal yang dapat dilihat pada Gambar 2.14. Percepatan lateral yang diperintahkan ditentukan dengan dengan Persamaan (2.82).

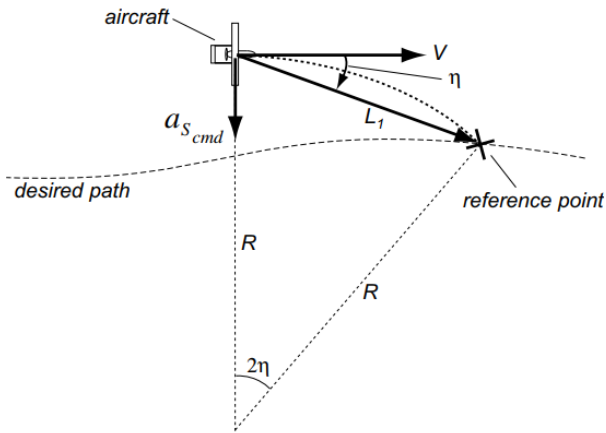
$$a_{s_{cmd}} = 2 \frac{V^2}{L_1} \sin \eta \quad (2.82)$$

Dimana:

V = Kecepatan badan pesawat

L_1 = Jarak pesawat dengan titik referensi pada lintasan terbang

η = Sudut antara kecepatan V dengan garis L_1



Gambar 2.14. Diagram *Nonlinear Guidance Logic* [9]

Bila Persamaan dilinierisasi dengan asumsi bahwa d adalah kesalahan lintasan dan nilai η bernilai kecil, maka didapatkan bahwa

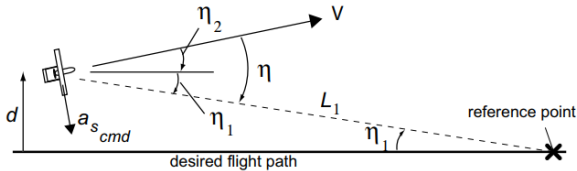
$$\sin \eta \approx \eta = \eta_1 + \eta_2 \quad (2.83)$$

$$\eta_1 \approx \frac{d}{L_1}, \quad \eta_2 \approx \frac{\dot{d}}{V} \quad (2.84)$$

Dengan menggabungkan Persamaan (2.83) dan (2.84) dengan Persamaan didapatkan hasil persamaan baru yaitu

$$a_{s_{cmd}} \approx 2 \frac{V}{L_1} \left(\dot{d} + \frac{V}{L_1} d \right) \quad (2.85)$$

Persamaan NLGL yang dilinerisasi tersebut menghasilkan sebuah pengendali *Proportional Derivative* (PD) untuk kesalahan lintasan seperti pada Gambar 2.15. Rasio antara kecepatan pesawat V dan jarak pemisah L_1 menjadi faktor penting dalam menentukan penguatan dari pengendali *proportional derivative*. Sebagai contoh, nilai L_1 yang kecil akan membawa penguatan pengaturan yang tinggi dan rasio L_1/V menentukan *time constant* dari pengendali PD.



Gambar 2.15. Model Linier Pemanduan *Nonlinear Guidance Logic* Garis Lurus [9]

Keuntungan pemandu ini mampu memberikan pengaturan periode dan redaman L_1 yang mudah dimana panjang L_1 dan penguatan dapat dihitung untuk menjaga dinamika konvergensi lintasan. Namun, terdapat dua kekurangan utama pada pemanduan ini. Pertama jika lintasannya adalah lingkaran maka konvergensi penelusuran lingkaran membutuhkan panjang L_1 yang lebih kecil atau sama dengan jari-jari R lintasan lingkaran tersebut, $L_1 \leq R$. Kondisi ini terjadi apabila kecepatan *ground* pesawat tidak sangat tinggi pada jari-jari lingkaran yang kecil. Kedua yaitu pemandu NLGL akan tidak dapat digunakan ketika kecepatan angin melebihi kecepatan udara pesawat yang umumnya dialami pada pesawat kecil dan lambat.

2.11.3. *Proportional Navigation* (PN) [13],[14]

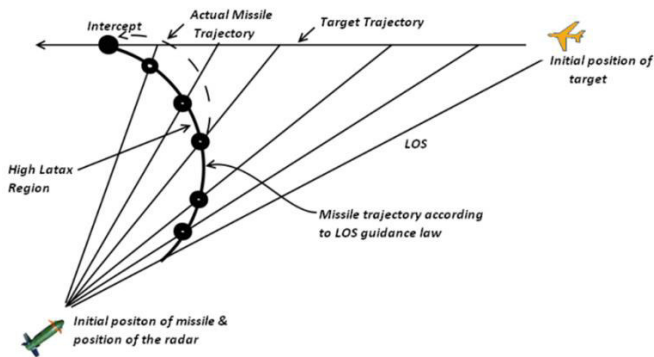
Dalam dunia praktis, pemanduan *proportional navigation* (PN) digunakan sangat luas untuk peluru kendali. Filosofi utama dibalik PN adalah percepatan rudal seharusnya menghilangkan *line-of-sight* (LOS) *rate* antara sasaran dan pengecat atau penembak. Pada hukum

permanduan, *proportional navigation* menggunakan dasar intuisi fisik *parallel navigation*. *Parallel navigation* menyatakan arah relatif terhadap LOS pada sistem koordinat inersial dijaga konstan atau *LOS rate* harus sama dengan nol. Hukum PN menyatakan bahwa percepatan yang diperintahkan adalah proporsional dengan *LOS rate*, konstan secara proporsional dapat dipecah menjadi perkalian rasio navigasi efektif N dikali dengan kecepatan mendekat rudal terhadap sasaran yang dirumuskan pada Persamaan (2.86).

$$a_c = NV_{cl}\dot{\lambda} \quad (2.86)$$

Dimana:

- a_c = Percepatan yang diperintahkan
- N = Konstanta proporsional
- V_{cl} = Kecepatan rudal mendekat sasaran
- $\dot{\lambda}$ = *LOS rate*



Gambar 2.16. Diagram *Proportional Navigation* [13]

Percepatan yang diperintahkan a_c berlaku tegak lurus terhadap LOS seperti pada Gambar 2.16. Kecepatan rudal mendekat dengan sasaran dapat didekati dengan Persamaan (2.87) berikut.

$$V_{cl} = V_M - V_T \quad (2.87)$$

Dimana:

- V_M = Kecepatan rudal
- V_T = Kecepatan sasaran

[Halaman ini sengaja dikosongkan]

BAB 3

PERANCANGAN SISTEM

Bab ini menjelaskan tahap pembuatan simulasi rudal yang mengikuti lintasan untuk menuju sasaran. Bagian pertama membahas tentang permodelan model matematika dinamika dan kinematika rudal, lalu perancangan linierisasi model rudal dengan *nonlinear feedback linearization*, selanjutnya perancangan pengaturan DMPC, perancangan lintasan *waypoint* rudal, dan yang terakhir perancangan pemandu untuk rudal untuk pengejaran sasaran.

3.1. Gambaran Umum Sistem

Dalam simulasi pemanduan rudal, ada beberapa tahap penting yang perlu diperhatikan. Pertama adalah membangun model matematika yang menggambarkan pergerakan rudal yang akan dibuat. Namun, karena model matematika rudal yang non-linier, maka perlu dirancang linierisasi sistem dengan menggunakan *nonlinear feedback decoupler* untuk membuat dinamika rudal menjadi linier. Setelah dinamika rudal menjadi linier, kemudian dirancang model matematika pengendalian yang akan dirancang yang pada tugas akhir ini, pengendali yang akan digunakan adalah DMPC. Pengendali ini kemudian menerjemahkan sinyal pemandu menjadi sinyal kendali ke aktuator untuk menggerakkan rudal. Tahap berikutnya adalah pemandu rudal yang dapat memberikan masukan ke pengendali yang akan dirancang. Selanjutnya adalah merancang lintasan yang menghubungkan titik *waypoint* yang akan dilewati rudal hingga menuju sasarnya.

3.2. Perancangan Model Rudal

Perancangan pertama adalah merancang model matematika rudal yang meliputi model dinamika dan kinematika rudal. Model matematika yang telah didapatkan digunakan sebagai model yang mewakili pergerakan rudal jenis *skid-to-turn* dengan model sirip canard. Model dinamika rudal hanya mencakup dinamika translasi dan dinamika rotasi dalam sistem koordinat badan rudal, sedangkan untuk mentransformasikan sistem koordinat badan menjadi sistem koordinat bumi dibutuhkan model kinematika rudal. Rudal dirancang Perancangan dinamika dan kinematika rudal akan diuraikan dalam sub bab berikut ini.

3.2.1. Perancangan Dinamika Rudal

Tahap pertama dalam simulasi rudal ini adalah membangun model matematika yang menggambarkan dinamika rudal bekerja. Hukum Newton kedua menjadi landasan utama sebagai penurunan persamaan dinamika rudal. Dengan menggunakan landasan hukum ini, didapatkanlah persamaan dinamika rudal yang mendeskripsikan enam derajat gerak kebebasan rudal yang dibagi menjadi tiga derajat gerak kebebasan translasi dan tiga derajat gerak kebebasan rotasi. Dinamika translasi menggambarkan percepatan translasi pada rudal yang dihasilkan oleh gaya-gaya yang memengaruhi gerak rudal sehingga percepatan rudal dapat berubah yang terdiri dari gaya gravitasi, gaya aerodinamika dan gaya dorong *thruster*. Sedangkan dinamika rotasi menggambarkan percepatan rotasi yang dipengaruhi oleh momen-momen rotasi yang dialami rudal. Dengan demikian, persamaan gaya-gaya yang bekerja pada rudal ditulis dalam Persamaan (3.1)-(3.3).

$$F_x = T_m - mg \sin \theta + \bar{q}SC_x \quad (3.1)$$

$$F_y = mg \sin \phi \cos \theta + \bar{q}SC_y \quad (3.2)$$

$$F_z = mg \cos \phi \cos \theta + \bar{q}SC_z \quad (3.3)$$

Dengan:

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

$$C_x = C_{x_0} \quad (3.5)$$

$$C_y = C_{y_\beta} \beta + C_{y_{\delta_r}} \delta_r \quad (3.6)$$

$$C_z = C_{z_\alpha} \alpha + C_{z_{\delta_e}} \delta_e \quad (3.7)$$

Serta persamaan momen rotasi yang bekerja pada rudal ditulis pada Persamaan (3.8)-(3.10).

$$M_x = \bar{q}SdC_l \quad (3.8)$$

$$M_y = \bar{q}SdC_m \quad (3.9)$$

$$M_z = \bar{q}SdC_n \quad (3.10)$$

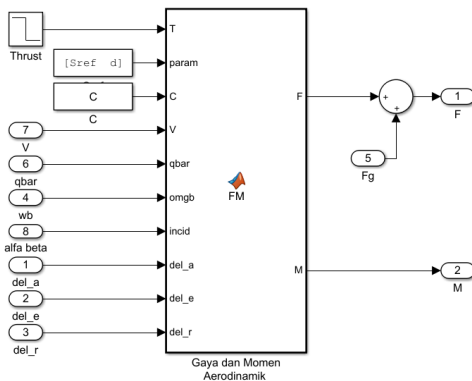
Dengan:

$$C_l = \bar{q}Sd \left(C_{l_{\delta_a}} \delta_a + C_{l_p} \frac{pd}{2V_m} \right) \quad (3.11)$$

$$C_m = \bar{q}Sd \left(C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e + C_{m_q} \frac{qd}{2V_m} \right) \quad (3.12)$$

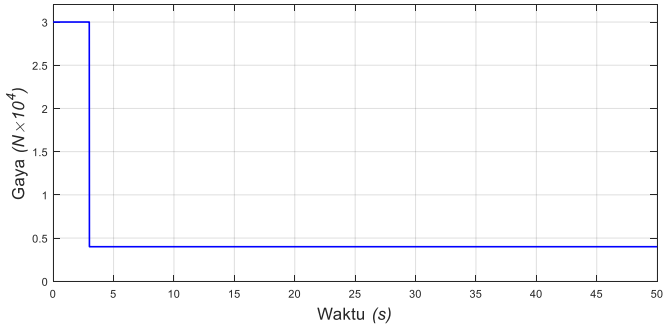
$$C_n = \bar{q}Sd \left(C_{n_\beta} \beta + C_{n_{\delta_r}} \delta_r + C_{n_r} \frac{rd}{2V_m} \right) \quad (3.13)$$

Persamaan-persamaan diatas kemudian ditulis menjadi kode program dalam Simulink pada software MATLAB. Fitur blok program yang disediakan dan digunakan pada Simulink yaitu MATLAB *Function* yang berisikan program untuk menghasilkan gaya dan momen aerodinamik yang bekerja pada rudal. Selain blok program tersebut, juga terdapat masukan berupa parameter-parameter serta gaya gravitasi. Sehingga hasil perancangan blok program untuk gaya dan momen rudal dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok MATLAB *Function* Gaya dan Momen Rudal

Nilai yang diberikan pada *thrust* selalu dibuat serupa untuk semua simulasi rudal. Gaya pendorong rudal atau *thrust* yang diberikan pada model sistem rudal ini terdapat dua fase. Fase pertama adalah fase dimana *booster* dan *sustainer* beroperasi hingga rudal mengalami akselerasi hingga rudal mencapai kecepatan nominal tertentu. Fase selanjutnya adalah fase mesin pendorong rudal menyokong rudal agar tetap terbang menuju sarannya. Gaya dorong dihasilkan *booster* selalu lebih besar dari *sustainer*. Model *thruster* yang diterapkan pada sistem rudal ini adalah fase *boost* pertama berlangsung selama 3 detik pertama dan dilanjutkan seterusnya oleh fase *sustain*. Grafik gaya dorong *thruster* yang digunakan dapat dilihat pada Gambar 3.2



Gambar 3.2 Masukkan *Thruster* pada Model Rudal

Selain itu, percepatan rudal dalam sistem koordinat badan juga dapat dipengaruhi oleh massa dan perubahannya serta hubungan antara kecepatan translasi dan kecepatan rotasi badan rudal. Oleh sebab itu, persamaan dinamika percepatan translasi rudal dalam sistem koordinat badan rudal menggunakan Persamaan (3.14) - (3.16) dibawah ini

$$\frac{du}{dt} = \frac{F_x - \dot{m}u}{m} + rv - qw \quad (3.14)$$

$$\frac{dv}{dt} = \frac{F_y - \dot{m}v}{m} + pw - ru \quad (3.15)$$

$$\frac{dw}{dt} = \frac{F_z - \dot{m}w}{m} + qu - pv \quad (3.16)$$

Sedangkan untuk percepatan rudal dalam sistem koordinat bumi datar menggunakan Persamaan (3.17) - (3.19)

$$a_x = \frac{F_x - \dot{m}u}{m} \quad (3.17)$$

$$a_y = \frac{F_y - \dot{m}v}{m} \quad (3.18)$$

$$a_z = \frac{F_z - \dot{m}w}{m} \quad (3.19)$$

Dengan prinsip yang sama pada dinamika translasi, dinamika rotasi juga menggunakan Hukum Newton II rotasi yang menunjukkan bahwa momen-momen yang bekerja sedemikian rupa pada rudal dapat menyebabkan percepatan sudut pada rudal tersebut. Momen-momen yang

bekerja pada rudal tersebut terdiri dari momen-momen aerodinamika dan momen interaksi antara kecepatan translasi rudal dengan kecepatan sudut rudal. Dengan pernyataan demikian, persamaan percepatan rotasi rudal dirumuskan dengan Persamaan (3.20) - (3.25)

$$\frac{dp}{dt} = \frac{1}{I_{xx}} (\bar{q} S d C_l - \dot{I}_{xx} p - (I_{zz} - I_{yy}) q r) \quad (3.20)$$

$$\frac{dq}{dt} = \frac{1}{I_{yy}} (\bar{q} S d C_m - \dot{I}_{yy} q - (I_{xx} - I_{zz}) p r) \quad (3.21)$$

$$\frac{dr}{dt} = \frac{1}{I_{zz}} (\bar{q} S d C_n - \dot{I}_{zz} r - (I_{yy} - I_{xx}) p q) \quad (3.22)$$

Dengan:

$$C_l = C_{l_{\delta_a}} \delta_a + C_{l_p} \frac{pd}{2V_m} \quad (3.23)$$

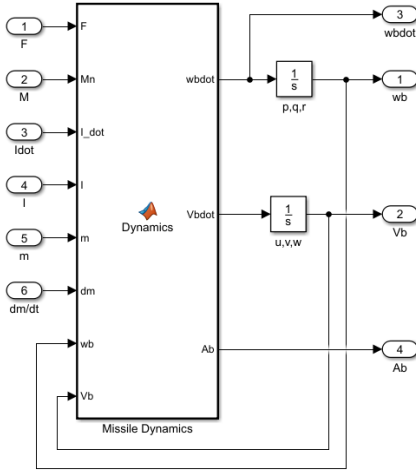
$$C_m = C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e + C_{m_q} \frac{qd}{2V_m} \quad (3.24)$$

$$C_n = C_{n_\beta} \beta + C_{n_{\delta_r}} \delta_r + C_{n_r} \frac{rd}{2V_m} \quad (3.25)$$

Dengan menggunakan fitur Simulink pada aplikasi software MATLAB, model dinamika rudal tersebut disalin dalam bentuk kode program. Untuk mempermudah menelusuri alur program rudal pada Simulink, maka dibuat beberapa blok program-program yang menggambarkan sub-sistem pada rudal. Blok yang digunakan pada fitur Simulink untuk menulis persamaan dinamika translasi dan rotasi rudal tersebut ialah dengan menggunakan blok MATLAB *Function*.

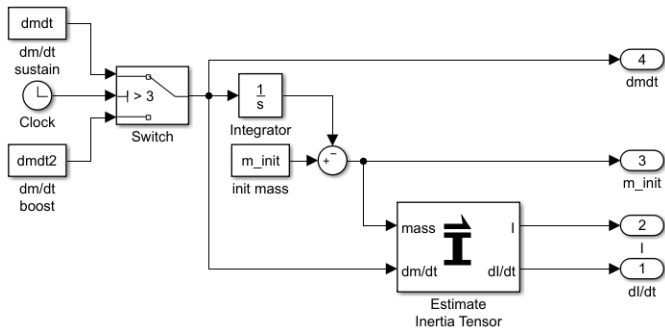
Seperti yang telah dijabarkan pada Persamaan dinamika rudal diatas, bahwa keluaran pada blok ini yaitu tiga percepatan translasi rudal ($\dot{u}, \dot{v}, \dot{w}$) dan tiga percepatan rotasi rudal ($\dot{p}, \dot{q}, \dot{r}$) dalam sistem koordinat badan rudal serta percepatan translasi dalam sistem koordinat bumi (a_x, a_y, a_z). Agar dapat menghasilkan kecepatan translasi rudal (u, v, w) dan kecepatan rotasi rudal (p, q, r) dalam sistem koordinat badan, maka dapat digunakan sebuah blok yang dapat menjalankan fungsi integral kontinyu pada Simulink yaitu dengan menggunakan blok *integrator*. Dengan demikian, blok fungsi Dinamika Rudal memiliki masukan yaitu gaya yang bekerja pada rudal, momen pada rudal, perubahan momen inersia, momen inersia, perubahan massa, massa rudal serta kecepatan

translasi dan rotasi rudal. Hasil penggunaan blok MATLAB *Function* untuk menulis Persamaan (3.14) hingga (3.25) tersebut dapat dilihat pada Gambar 3.3.



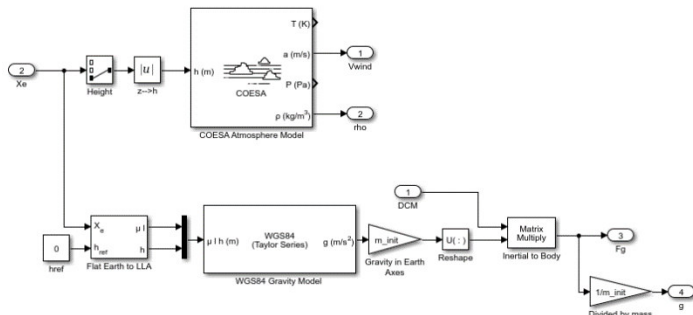
Gambar 3.3 Blok MATLAB *Function* Dinamika Rudal

Massa bahan bakar yang digunakan untuk mendorong rudal selama terbang di udara menyebabkan massa total rudal akan berkurang seiring dengan waktu. Hal ini akan memengaruhi dinamika rudal dimana percepatan translasi dipengaruhi oleh massa rudal. Namun, karena fase *boost* menghasilkan gaya dorong yang lebih besar dibandingkan dengan fase *sustain*, maka terdapat dua fase perubahan massa yang berbeda pada rudal. Oleh sebab itu, maka perancangan ini menggunakan dua konstanta perubahan massa yaitu konstanta perubahan massa untuk fase *boost* dan fase *sustain*. Berdasarkan Persamaan (2.39) dan memasukkan parameter badan rudal pada Tabel 3.1, parameter momen inersia rudal didapatkan dan nilainya dapat dilihat pada Tabel. Pada rancangan simulasi dengan Simulink, perubahan momen inersia dilakukan estimasi dengan blok Simulink *Estimate Inertia Tensor*. Hasil perancangannya dapat dilihat pada Gambar 3.4.



Gambar 3.4 Blok Simulink Massa dan Momen Inersia Rudal

Selain perancangan dinamika dan kinematika rudal dengan Simulink, parameter-parameter lingkungan juga didesain agar semirip mungkin dengan kondisi nyata saat rudal terbang di udara. Oleh karena itu, perancangan model lingkungan rudal juga dilakukan. Dengan menggunakan blok Simulink yang telah tersedia dalam *library* Simulink, model lingkungan rudal didesain menggunakan *library* COESA *Atmosphere Model* dan *WGS84 Model Gravity* dimana blok tersebut digunakan untuk mendapatkan parameter kecepatan angin, kepadatan udara dan percepatan gravitasi. Hasil perancangan blok model lingkungan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Blok Model Lingkungan Rudal

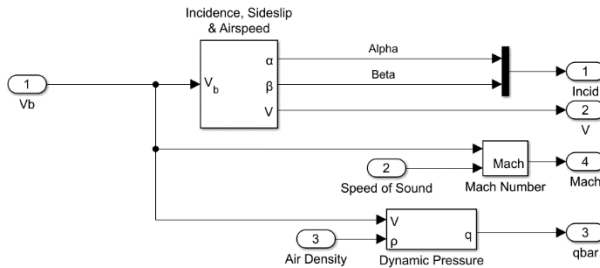
Kemudian selanjutnya adalah memperhitungkan *angle of attack* (α), *sideslip angle* (β), tekanan udara (\bar{q}) dan kecepatan rudal dalam satuan *Mach* dengan mengalikan kecepatan suara sebesar 330 m/s . Persamaan

yang digunakan untuk mendapatkan nilai *angle of attack* dan *sideslip angle* serta tekanan udara dapat dilihat pada Persamaan (3.26) - (3.28)

$$\alpha = \tan^{-1} \frac{w}{u} \quad (3.26)$$

$$\beta = \sin^{-1} \frac{v}{\sqrt{u^2 + v^2 + w^2}} \quad (3.27)$$

$$\bar{q} = \frac{1}{2} \rho V_m^2 \quad (3.28)$$



Gambar 3.6 Blok Perhitungan Sudut α , β , \bar{q} dan *Mach*

Nilai parameter-parameter pada rudal seperti massa dan lainnya juga digunakan dalam dinamika rudal. Beberapa parameter badan rudal diambil dari parameter rudal tipe C-705. Parameter lain seperti kepadatan udara dan gaya gravitasi telah dihitung pada blok lingkungan rudal, sehingga nilainya akan berubah-ubah se. Secara keseluruhan, nilai-nilai parameter yang telah ditentukan pada perancangan model matematika rudal dapat dilihat pada Tabel 3.1.

3.2.2. Perancangan Kinematika Rudal

Tahap ini merancang kinematika rudal yang mentransformasikan kecepatan translasi rudal (u, v, w) dan kecepatan rotasi rudal (p, q, r) dalam sistem koordinat badan rudal menjadi kecepatan translasi rudal ($\dot{x}, \dot{y}, \dot{z}$) dan kecepatan rotasi ($\dot{\phi}, \dot{\theta}, \dot{\psi}$) dalam sistem koordinat bumi. Oleh sebab itu, diperlukan transformasi koordinat agar sistem koordinat badan rudal dapat direpresentasikan menjadi sistem koordinat bumi. Persamaan transformasi koordinat kecepatan translasi koordinat rudal menjadi koordinat bumi dirumuskan dengan Persamaan (3.29).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.29)$$

Tabel 3.1. Tabel Nilai Parameter Rudal

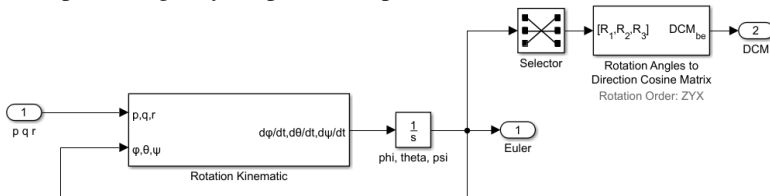
Parameter	Nilai	Satuan
Massa Awal (m)	340	Kg
Perubahan Massa Boost (\dot{m}_1)	-0.016	Kg/s
Perubahan Massa Sustain (\dot{m}_2)	-0.15	Kg/s
Diameter (d)	0.28	m
Luas Permukaan (S)	0.2622	m ²
Koefisien C_{x_0}	-0.3	-
Koefisien C_{y_β}	-0.4	-
Koefisien $C_{y_{\delta_r}}$	-0.05	-
Koefisien C_{z_α}	-0.4	-
Koefisien $C_{z_{\delta_e}}$	-0.05	-
Koefisien $C_{l_{\delta_a}}$	-0.05	-
Koefisien C_{l_p}	-0.01	-
Koefisien C_{m_α}	-0.03	-
Koefisien $C_{m_{\delta_e}}$	0.05	-
Koefisien C_{m_q}	-0.01	-
Koefisien C_{n_β}	0.03	-
Koefisien $C_{n_{\delta_r}}$	-0.05	-
Koefisien C_{n_r}	-0.01	-

Kemudian transformasi koordinat yang mengubah kecepatan rotasi sistem koordinat badan rudal menjadi sistem koordinat bumi dirumuskan dengan Persamaan (3.30)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.30)$$

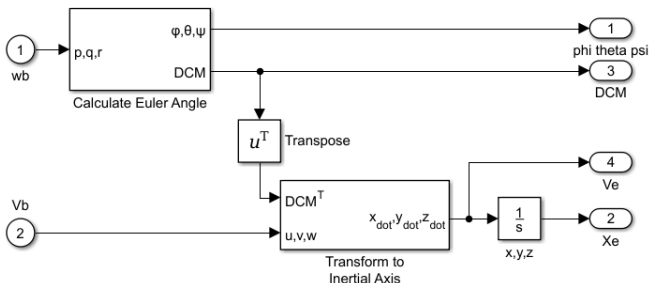
Memperoleh posisi rudal dalam sistem koordinat bumi dapat diperoleh dengan cara mengintegalkan kecepatan rudal dalam koordinat bumi pula menggunakan blok *integrator*. Cara yang sama juga diterapkan untuk mendapatkan sudut orientasi rudal atau sudut Euler dari kecepatan rotasi rudal dalam sistem koordinat bumi.

Perancangan blok kinematika rudal ini diawali dengan menghitung sudut Euler dan matriks *Direct Cosine Matrix* (DCM). Beberapa fitur blok Simulink yang dapat digunakan seperti blok *fcn* dan *Rotation Angles to DCM* dapat digunakan untuk mendapatkan sudut Euler dan matriks DCM. Hasil perancangannya dapat dilihat pada Gambar 3.7 .



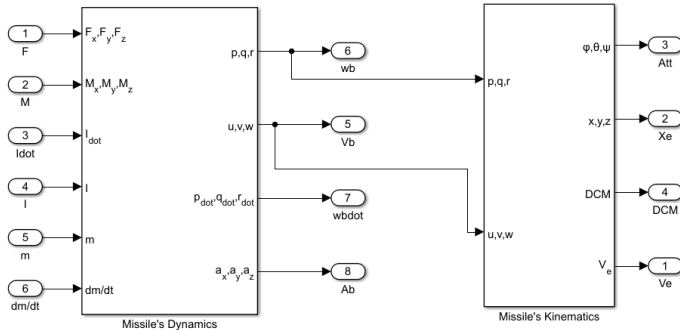
Gambar 3.7 Blok Perhitungan Sudut Euler

Selanjutnya, dengan mendapatkan informasi sudut Euler dan matriks DCM yang diinginkan, kecepatan dan posisi rudal dalam sistem koordinat bumi dapat diperoleh dengan menggunakan perkalian matriks antara kecepatan rudal dalam sistem koordinat bumi dan matriks DCM yang kemudian diimplementasikan dan digabungkan dengan blok perhitungan sudut Euler seperti pada Gambar 3.7. Hasil keseluruhan blok kinematika rudal dapat dilihat pada Gambar 3.8.



Gambar 3.8 Blok Kinematika Rudal

Setelah blok dinamika dan kinematika rudal telah dirancang, blok-blok tersebut digabungkan menjadi satu kesatuan dan hasil blok tersebut dapat dilihat pada Gambar 3.9.



Gambar 3.9 Blok Dinamika dan Kinematika Rudal

3.3. Perancangan *Nonlinear Feedback Linearization*

Perancangan ini dilakukan untuk mendapatkan persamaan yang dapat menghilangkan sifat non-linieritas sistem dan sifat *coupled* atau ketergantungan antar variabel-variabel dinamika rudal. Terlihat pada Persamaan hingga , dinamika percepatan rotasi dan translasi saling berhubungan dan berkaitan salah satu contohnya yaitu sudut serang (α) dan sudut *slip* (β) yang terdapat pada seluruh persamaan dinamika rudal. Karena pada perancangan rudal ini menggunakan autopilot pengendalian percepatan maka persamaan non-linier yang akan dilinierisasi yaitu persamaan dinamika translasi agar pengendali yang dirancang lebih sederhana dan mampu mengendalikan rudal lebih mudah. Dalam perancangan linierisasi ini, terdapat dua tahap untuk mendapatkan persamaan linier dinamika translasi yaitu tahap pertama yaitu mendapatkan persamaan linier dinamika rotasi yang kemudian dilanjutkan dengan melinierisasi dinamika translasi untuk memberikan masukan referensi untuk persamaan linier rotasi. Untuk lebih jelasnya, linierisasi sistem akan dijabarkan pada Sub Bab 3.3.1 dan 3.3.2.

3.3.1. Linierisasi Kecepatan Rotasi Rudal

Linierisasi kecepatan rotasi dilakukan dengan cara mendapatkan sinyal masukan sistem yang sesuai dan tepat sehingga sifat-sifat non-linieritas rotasi rudal dapat dihilangkan. Sinyal masukan yang dimaksud pada persamaan dinamika rotasi adalah sudut defleksi rudal yaitu δ_a, δ_e

dan δ_r . Jadi, persamaan percepatan sudut rudal dilinerisasi dengan mencari nilai δ_a, δ_e dan δ_r serta menambah parameter baru sehingga didapatkan persamaan linier orde satu baru yang *decoupled*. Parameter baru tersebut yaitu *time constant* (T) sistem orde satu yang diinginkan dan masukkan referensinya. *Time constant* dipilih yaitu 0.1 detik karena rudal harus memiliki respons yang cepat agar dapat dikendalikan. Sedangkan parameter baru satunya ialah parameter masukkan referensi yang diinginkan dari respon sistem tersebut beserta penguatan atau *gain* yang diinginkan (b). Nilai penguatan dipilih sama dengan satu per *time constant* (a) agar respons sistem memiliki nilai akhir yang sama dengan masukkan referensi yang diberikan. Dengan demikian, persamaan untuk mendapatkan masukkan sudut defleksi sirip rudal agar sistem dinamika rotasi rudal menjadi linier dirumuskan dalam Persamaan (3.31)-(3.33).

$$\delta_a = \frac{1}{C_{l\delta_a}} \left(-C_{lp} \frac{pd}{2V_m} + \frac{I_{xx}}{\bar{q}Sd} \dot{p} + \frac{(I_{zz} - I_{yy})}{\bar{q}Sd} \dot{q}r - \frac{a_p I_{xx}}{\bar{q}Sd} p + \frac{b_p I_{xx}}{\bar{q}Sd} p_{ref} \right) \quad (3.31)$$

$$\delta_e = \frac{1}{C_{m\delta_e}} \left(-C_{mq} \frac{qd}{2V_m} - C_{m\alpha} \alpha + \frac{I_{yy}}{\bar{q}Sd} \dot{q} + \frac{(I_{xx} - I_{zz})}{\bar{q}Sd} \dot{q}r - \frac{a_q I_{yy}}{\bar{q}Sd} q + \frac{b_q I_{yy}}{\bar{q}Sd} q_{ref} \right) \quad (3.32)$$

$$\delta_r = \frac{1}{C_{n\delta_r}} \left(-C_{nr} \frac{rd}{2V_m} - C_{n\beta} \beta + \frac{I_{zz}}{\bar{q}Sd} \dot{r} + \frac{(I_{yy} - I_{xx})}{\bar{q}Sd} \dot{q}r - \frac{a_r I_{zz}}{\bar{q}Sd} r + \frac{b_r I_{zz}}{\bar{q}Sd} r_{ref} \right) \quad (3.33)$$

Dengan nilai a sama dengan nilai b

$$a_p = \frac{1}{T_p} = b_p = 30 \quad (3.34)$$

$$a_q = \frac{1}{T_q} = b_q = 30 \quad (3.35)$$

$$a_r = \frac{1}{T_r} = b_r = 30 \quad (3.36)$$

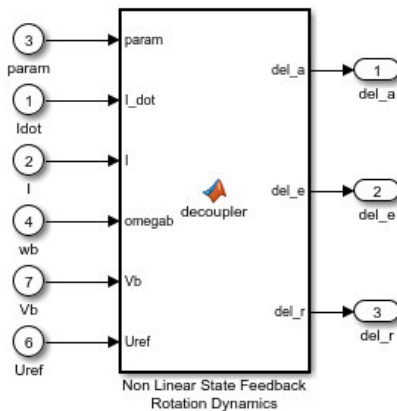
Sehingga persamaan dinamika rotasi yang telah dilinierisasi dan *decoupled* yang baru dapat dilihat pada Persamaan (3.37)-(3.39).

$$\frac{dp}{dt} = -30p + 30p_{ref} \quad (3.37)$$

$$\frac{dq}{dt} = -30q + 30q_{ref} \quad (3.38)$$

$$\frac{dr}{dt} = -30r + 30r_{ref} \quad (3.39)$$

Untuk merealisasikan perancangan dalam simulasi, persamaan masukkan sudut defleksi sirip yang telah diperoleh dari persamaan diatas kemudian dituliskan dalam *script* dalam MATLAB *Function*. Proses linierisasi dan decoupling sistem akan terjadi dengan sendirinya tanpa memodifikasi model matematika dinamika rudal yang telah dibuat sebelumnya. Blok ini membutuhkan informasi yang persis dengan informasi yang ada pada sistem rudal yang bekerja sehingga masukkan blok fungsi ini berupa kecepatan translasi dan kecepatan sudut rudal, parameter-parameter badan rudal, koefisien aerodinamika rotasi serta masukkan referensi yang baru. Keluaran yang dihasilkan pada blok ini yaitu tiga sudut defleksi sirip *aileron*, *elevator* dan *rudder* rudal. Hasil perancangan linierisasi dinamika rotasi yang dituliskan dalam MATLAB *Function* dapat dilihat pada Gambar 3.10 dibawah ini.



Gambar 3.10 Blok Diagram Simulink *Nonlinear Feedback Linearization* Dinamika Rotasi

Masukkan U_{ref} ialah masukkan referensi untuk kecepatan sudut (p, q, r) dengan catatan bahwa model belok rudal adalah *skid-to-turn* sehingga nilai *roll rate* (p) dibuat selalu bernilai 0. Dengan demikian, terdapat nilai dua variabel lainnya yaitu q dan r akan diberi masukkan dari blok linierisasi kecepatan.

3.3.2. Linierisasi Kecepatan Translasi Rudal

Sama seperti linierisasi persamaan dinamika rotasi rudal sebelumnya, melinierisasikan dinamika translasi menggunakan metode dan langkah yang sama dengan linierisasi dinamika rotasi. Dikarenakan rudal yang dirancang hanya mampu mengendalikan sudut defleksi sirip rudal dan tidak mampu mengendalikan pendorong (*booster*), maka persamaan percepatan translasi rudal sumbu x koordinat badan yaitu \dot{u} tidak digunakan dalam linierisasi. Oleh karena itu, persamaan translasi yang digunakan untuk dilinierikan yaitu hanya persamaan dinamika translasi v dan w . Dengan menentukan keluaran hasil linierisasi yaitu q dan r , maka didapatkan persamaan linierisasi untuk mendapatkan masukkan kecepatan rotasi yang sesuai agar dinamika translasi rudal menjadi linier dirumuskan dalam Persamaan (3.40)-(3.41) berikut.

$$r = \frac{1}{u} \left(\frac{\bar{q}SC_{y\beta}}{m} \beta + \frac{\bar{q}SC_{y\delta_r}}{m} \delta_r - \frac{\dot{m}}{m} v + pw + a_v v - b_v v_{ref} \right) \quad (3.40)$$

$$q = -\frac{1}{u} \left(\frac{\bar{q}SC_{z\alpha}}{m} \alpha + \frac{\bar{q}SC_{y\delta_e}}{m} \delta_e - \frac{\dot{m}}{m} w - pv + a_w w - b_w w_{ref} \right) \quad (3.41)$$

Ditentukan nilai a dua kali lebih besar dari nilai b . Pemilihan nilai a juga dipertimbangkan karena dengan linierisasi ini diharapkan respon dinamika translasi mendekati respon sistem orde satu. Maka pemilihan a dinamika translasi harus minimal dua kali lebih lambat dari nilai a dinamika rotasi. Oleh sebab itu, ditentukan nilai a_v dan a_w adalah sebagai berikut.

$$a_v = \frac{1}{T_v} = 2b_v = 10 \quad (3.42)$$

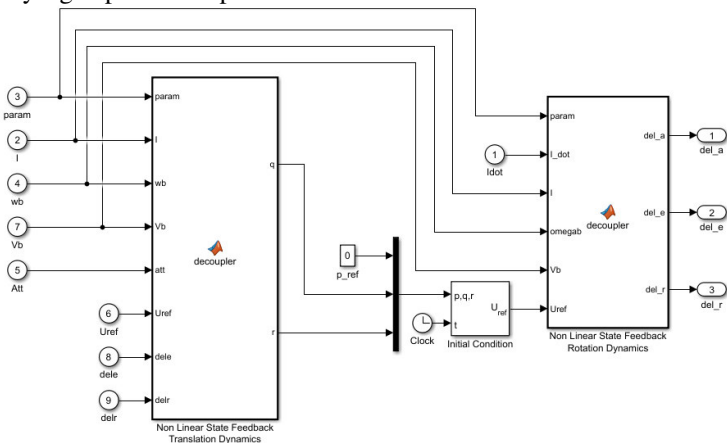
$$a_w = \frac{1}{T_w} = 2b_w = 10 \quad (3.43)$$

Dengan demikian, persamaan dinamika translasi yang *decoupled* dan linier untuk dua kecepatan translasi pada sumbu *y* dan sumbu *z* dapat dilihat pada Persamaan (3.44)-(3.45).

$$\frac{dv}{dt} = -10v + 5v_{ref} \quad (3.44)$$

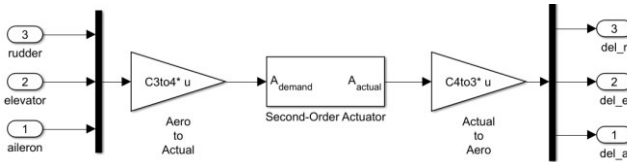
$$\frac{dw}{dt} = -10w + 5w_{ref} \quad (3.45)$$

Kemudian, perancangan simulasi dari persamaan yang telah didapatkan dari *feedback linearization* dibuat dalam script program MATLAB *Function*. Masukkan dari blok ini juga sama dengan blok linierisasi dinamika rotasi yaitu berupa parameter-parameter badan rudal, koefisien aerodinamika translasi, kecepatan translasi dan kecepatan rotasi dalam koordinat badan. Perlu dicatat bahwa persamaan diatas memiliki pembagi variabel *u* dan karena *booster* selalu bekerja, maka nilai *u* selama terbang tidak akan nol, sehingga hanya kondisi awal yang memengaruhi kejadian tersebut. Oleh sebab itu, kondisi awal dari nilai *u* diberi nilai yang kecil. Selain itu, perlu juga ditambahkan kondisi awal untuk nilai masukkan referensi kecepatan rotasi ketika digabungkan dengan blok linierisasi dinamika translasi. Blok simulasi ini kemudian digabung menjadi satu kesatuan dengan blok linierisasi kecepatan rotasi rudal yang dapat dilihat pada Gambar 3.11 .



Gambar 3.11 Blok Simulink Keseluruhan *Nonlinear Feedback Linierization* Dinamika Rudal

Aktuator pada sirip rudal memiliki keterbatasan terhadap sudut defleksi yang diperintahkan dari autopilot. Dengan begitu, simulasi rudal diberikan batas maksimum dan minimum atau saturasi dari sudut defleksi yang diberikan. Sudut defleksi sirip rudal maksimal untuk keempat sirip yaitu sebesar 10° dan minimum yaitu -10° . Selain itu, aktuator juga memiliki waktu transien untuk menggerakkan sirip rudal. Respon aktuator diasumsikan bahwa respon sistem orde dua yang digambarkan dengan Persamaan (2.53). Namun, sebelum itu perlu ditransformasikan dari tiga sudut defleksi dari autopilot menjadi defleksi empat sirip dapat dilihat pada Persamaan (2.52). Setelah mendapatkan respon masing-masing defleksi sirip, perlu dilakukan transformasi balik dari defleksi sirip menjadi defleksi sebelumnya dari autopilot. Hasilnya dapat dilihat pada Gambar 3.12.



Gambar 3.12 Blok Simulink Dinamika Aktuator Sirip

Berdasarkan Persamaan (2.53), perlu dilakukan penentuan nilai ζ dan ω_n dari respon sistem orde dua tersebut. Respon sirip harus sangat cepat sehingga ditentukan bahwa kriteria desain sistem dengan *overshoot* maksimal 4.3% dan waktu *settling* ($\pm 2\%$) 0.037 detik. Maka untuk mencari parameter sistem tersebut dapat menggunakan Persamaan berikut.

$$\zeta' = \frac{|\ln(\frac{PO}{100})|}{\sqrt{\pi^2 + (\frac{PO}{100})^2}} \quad (3.46)$$

$$\omega'_n = \frac{4}{\zeta' t_s} \quad (3.47)$$

Dimana:

ζ' = Rasio redaman yang diinginkan

ω'_n = Frekuensi natural yang diinginkan

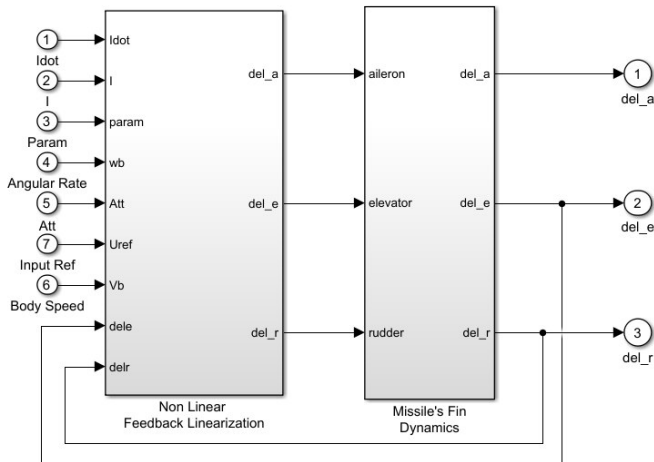
PO = Maksimum *overshoot* respon

t_s = Waktu *settling* sistem

Persamaan diatas digunakan untuk mendapatkan parameter persamaan respon orde dua. Dengan persamaan diatas, didapatkan bahwa nilai ζ sebesar 0.707 dan nilai ω_n sebesar 150. Dengan demikian, persamaan respon orde dua dari aktuator sirip tersebut menghasilkan Persamaan (3.48) berikut.

$$\frac{\delta_{actual}(s)}{\delta_{cmd}(s)} = \frac{22500}{s^2 + 1060.5s + 22500} \quad (3.48)$$

Dimana δ_{cmd} adalah sudut defleksi $\delta_1, \delta_2, \delta_3, \delta_4$ yang diperintahkan dan δ_{actual} adalah respon sudut defleksi $\delta_1, \delta_2, \delta_3, \delta_4$ yang sebenarnya. Blok respon aktuator ini digunakan untuk menyumulasikan gerakan aktual sirip rudal menggunakan servo. Hasil perancangannya yang kemudian digabungkan dengan blok linierisasi dinamika rudal dapat dilihat pada Gambar 3.13.

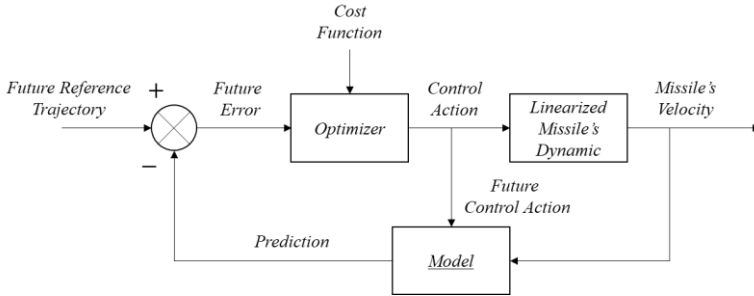


Gambar 3.13 Blok Simulink Aktuator Sirip Rudal yang dihubungkan dengan Blok Linierisasi Dinamika

3.4. Perancangan *Discrete Model Predictive Control (DMPC)*

Merancang pengendali yang berbasis model matematika sistem seperti DMPC, langkah utama yaitu mendapatkan model *state-space* dari model dinamika translasi rudal yang telah dilinierisasi. Namun, karena model dinamika rudal dilinierisasi melalui dua tahap yaitu linierisasi

dinamika rotasi dan dilanjutkan dengan linierisasi dinamika translasi, maka perlu dilakukan identifikasi sistem yang dihasilkan agar model yang dimasukkan dalam pengendali DMPC lebih akurat. Pengaturan DMPC pada perancangan ini dapat dilihat pada Gambar 3.14.

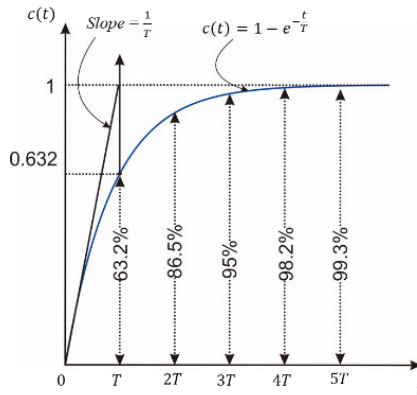


Gambar 3.14 Diagram Blok Perancangan DMPC

Tahapan untuk mendapatkan model dinamika yang akan dimasukkan dalam pengendali DMPC adalah sebagai berikut.

1. Simulasikan respon *step* untuk dinamika translasi v dan w
2. Dengan melihat respons step tersebut kemudian dapatkan parameter untuk sistem orde satu yang digambarkan pada Persamaan dan Gambar 3.15

$$\frac{Y(s)}{X(s)} = \frac{K}{s + \tau} \quad \tau = \frac{1}{T} \quad (3.49)$$



Gambar 3.15 Karakteristik Respon Orde Satu

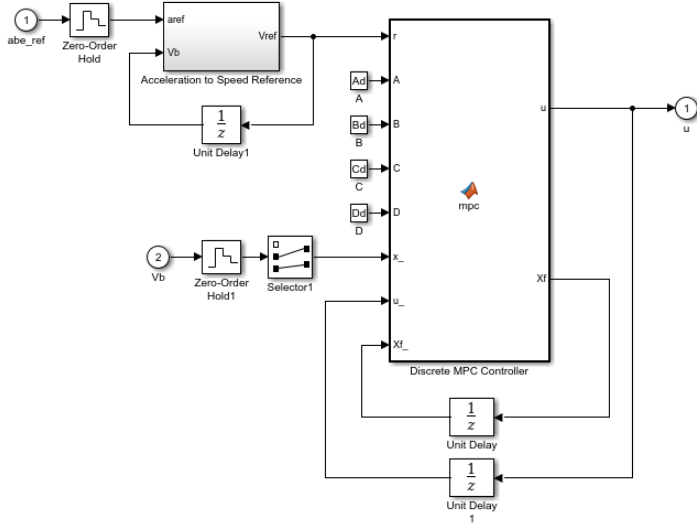
3. Dengan karakteristik sistem orde satu diatas, nilai K didapatkan dengan cara keluaran ketika keluaran sudah *steady state* dibagi dengan masukkan *step* yang diberikan, sedangkan mendapatkan nilai T dengan cara melihat waktu ketika respon berada di 0.632 dari nilai keluaran *steady state*.
4. Kemudian parameter tersebut dimasukkan dalam model *state-space* kontinyu dengan persamaan berikut

$$\begin{aligned} \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} &= \begin{bmatrix} -\tau_v & 0 \\ 0 & -\tau_w \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} + \begin{bmatrix} K_v & 0 \\ 0 & K_w \end{bmatrix} \begin{bmatrix} v_{ref} \\ w_{ref} \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \end{aligned} \quad (3.50)$$

5. Diskritisasi model *state-space* tersebut dengan waktu *sampling* T_s
6. Model *state-space* diskrit kemudian akan digunakan dalam pengendali DMPC.

Karena masukkan dari pemanduan adalah berupa percepatan yang diperintahkan untuk autopilot, maka perlu dilakukan pengubahan informasi dari percepatan yang diperintahkan menjadi informasi kecepatan. Hal tersebut dapat dilakukan dengan fungsi integral diskrit. Selanjutnya setelah mendapatkan model diskrit dari dinamika translasi linier dan informasi kecepatan, pembuatan program dengan menggunakan blok MATLAB *Function* dilakukan dan beberapa fungsi ditulis dalam script. Program untuk pengendali DMPC terlampir pada Lampiran. *Time sampling* yang digunakan yaitu 0.005 detik. Berikutnya yaitu menentukan nilai dari banyaknya sampel dalam jendela prediksi N_p , banyak sampel trayektori sinyal kendali N_c . Nilai N_p merepresentasikan panjangnya horison yang dapat diprediksi, bila nilainya besar maka semakin panjang horison yang dapat diprediksi, begitu juga untuk nilai N_c untuk menangkap trayektori sinyal kendali yang akan datang. Namun, bila terlalu besar, pengendali tidak memberikan dampak yang signifikan pada sinyal kendali yang dihasilkan karena energi optimal terdistribusi sepanjang waktu periode yang akan datang. Parameter terakhir yaitu parameter tuning \bar{R} untuk menentukan performansi lup tertutup yang diinginkan. \bar{R} adalah matriks diagonal dengan bentuk matriks kotak ukuran $N_c \times N_c$. Nilai r_w ditentukan, bila bernilai nol maka fungsi tujuan tidak mempertimbangkan seberapa besar nilai ΔU yang terjadi untuk meminimumkan kesalahan sekecil mungkin.

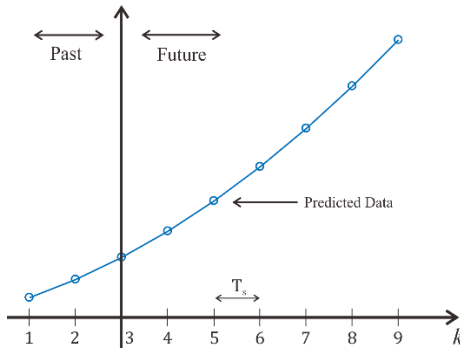
Masukkan dari blok *function* pada Simulink yaitu masukkan kecepatan translasi referensi, model *state-space* diskrit, dan kecepatan translasi rudal serta keluarannya berupa sinyal masukkan untuk blok *nonlinear feedback linearization*. Hasil perancangan pengaturan DMPC dalam Simulink disajikan pada Gambar 3.16.



Gambar 3.16 Blok Simulink Pengaturan DMPC

3.5. Perancangan Prediksi Kecepatan Rudal

Dengan adanya kemampuan prediksi perilaku sistem yang akan datang, diharapkan bahwa autopilot dapat memberikan sinyal kendali yang dapat menghasilkan respon sistem yang lebih cepat mencapai referensi dengan kemampuan prediktifnya dibandingkan dengan metode pengaturan lainnya dimana membutuhkan informasi keluaran yang kemudian diumpankanbalikkan untuk menghasilkan sinyal kendali sehingga sinyal kendalinya selalu terlambat. Oleh karena itu, masukkan referensi juga harus mengandung informasi dimana sistem harus mencapainya pada waktu sampling yang akan datang. Perancangan ini dilakukan untuk memprediksi kecepatan referensi yang akan datang berdasarkan informasi kecepatan referensi saat ini dan sebelumnya. Ilustrasi prediksi data kecepatan menggunakan tiga informasi digambarkan pada Gambar 3.17.



Gambar 3.17 Ilustrasi Prediksi Data Kecepatan

Perancangan prediksi kecepatan ini menggunakan pendekatan polinomial kuadratik deret waktu, dimana prediksi ini membutuhkan tiga informasi kecepatan sebelumnya. Untuk mendapatkan prediksi kecepatan tersebut, koefisien polinomial tersebut harus dicari terlebih dahulu. Koefisien persamaan polinomial kuadratik tersebut didapatkan dengan menggunakan Persamaan (3.51) berikut.

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} T_s^2 & T_s & 1 \\ 4T_s^2 & 2T_s & 1 \\ 9T_s^2 & 3T_s & 1 \end{bmatrix}^{-1} \begin{bmatrix} v(k-2) \\ v(k-1) \\ v(k) \end{bmatrix} \quad (3.51)$$

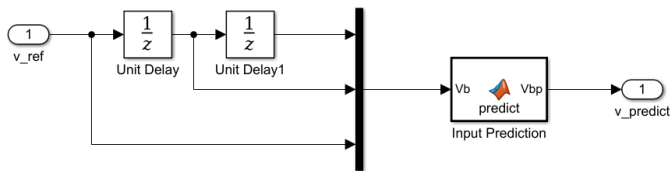
Dengan nilai koefisien tersebut, untuk mendapatkan referensi kecepatan prediksi sebanyak N_p sampel yang akan datang dirumuskan dengan Persamaan (3.52).

$$\begin{bmatrix} v(k+1) \\ v(k+2) \\ v(k+3) \\ \vdots \\ v(k+N_p) \end{bmatrix} = \begin{bmatrix} 16T_s^2 & 4T_s & 1 \\ 25T_s^2 & 5T_s & 1 \\ 36T_s^2 & 6T_s & 1 \\ \vdots & \vdots & \vdots \\ (N_p+3)^2 T_s^2 & (N_p+3)T_s & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (3.52)$$

Prediksi referensi kecepatan tersebut dilakukan pada kecepatan v dan w . Agar dapat digunakan dalam blok DMPC, referensi prediksi tersebut kemudian disusun menjadi vektor kolom dengan ketentuan berikut.

$$u_{ref} = \begin{bmatrix} v(k+1) \\ w(k+1) \\ v(k+2) \\ w(k+2) \\ \vdots \\ v(k+N_p) \\ w(k+N_p) \end{bmatrix} \quad (3.53)$$

Hasil perancangan tersebut ditulis dalam program pada MATLAB *Function* yang hasilnya dapat dilihat pada Gambar 3.18.



Gambar 3.18 Blok Prediksi Referensi Kecepatan

3.6. Perancangan *Path Planning*

Path planning menjadi hal penting dalam merencanakan perjalanan rudal agar sampai pada sasarannya. Untuk membentuk lintasan yang harus dilalui rudal, masukkan yang akan diberikan berupa titik-titik *waypoint* dalam sistem koordinat bumi. Banyaknya titik-titik ini tidak ditentukan atau bebas dan titik ini yang akan menjadi acuan rudal untuk dapat dilewatinya. Perancangan *path planning* ini akan dilakukan untuk memberikan referensi bentuk lintasan yang harus dilalui rudal agar dapat melewati *waypoint* yang telah ditentukan. Perancangan ini juga digunakan untuk menentukan lokasi titik yang berjarak L_1 terhadap rudal.

Cara termudah untuk menghubungkan antar empat buah *waypoint* ialah dengan membuat garis lurus yang menghubungkan antar *waypoint* tersebut. Garis lurus ini kemudian ditentukan persamaannya dalam ruang tiga dimensi. Dengan mendapatkan persamaan ini, kita dapat menggunakan rumus geometri untuk mendapatkan titik referensi di lintasan yang berjarak L_1 terhadap posisi rudal. Persamaan garis lurus antara dua titik dalam ruang tiga dimensi dituliskan dalam persamaan dibawah ini.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \quad (3.54)$$

Dimana x_1, y_1, z_1 adalah lokasi titik pertama dan x_2, y_2, z_2 adalah lokasi titik kedua. Dengan persamaan garis diatas, kemudian disederhanakan menjadi persamaan garis untuk bidang xy dan xz sehingga persamaan tersebut dapat ditulis ulang menjadi Persamaan (3.55).

$$\begin{aligned} y &= m_y x + k_y \\ z &= m_z x + k_z \end{aligned} \quad (3.55)$$

Dimana konstanta-konstanta persamaan diatas ialah :

$$\begin{aligned} m_y &= \frac{y_2 - y_1}{x_2 - x_1} \\ m_z &= \frac{z_2 - z_1}{x_2 - x_1} \\ k_y &= y_1 - \frac{y_2 - y_1}{x_2 - x_1} x_1 \\ k_z &= z_1 - \frac{z_2 - z_1}{x_2 - x_1} x_1 \end{aligned} \quad (3.56)$$

Kemudian untuk mendapatkan panjang atau *norm* dari dua titik yaitu titik pada lintasan dan titik lokasi rudal dalam ruang tiga dimensi sepanjang L_1 dapat menggunakan persamaan berikut.

$$(x - x_m)^2 + (y - y_m)^2 + (z - z_m)^2 = L_1^2 \quad (3.57)$$

Dengan mensubstitusikan Persamaan (3.55) ke Persamaan (3.57), didapatkan persamaan kuadrat sebagai berikut.

$$ax^2 + bx + c = 0 \quad (3.58)$$

Dengan konstanta a, b , dan c ialah

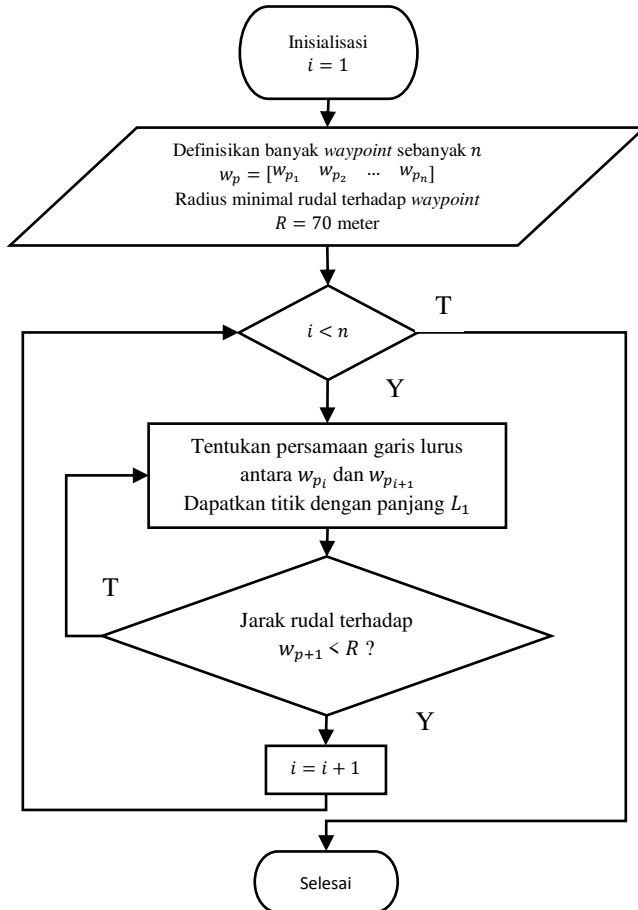
$$\begin{aligned} a &= 1 + m_y^2 + m_z^2 \\ b &= -2x_m - 2m_y(y_m - k_y) - 2m_z(z_m - k_z) \\ c &= x_m^2 + (y_m - k_y)^2 + (z_m - k_z)^2 - L_1^2 \end{aligned} \quad (3.59)$$

Selanjutnya adalah mendapatkan salah satu akar yang berada selalu di depan rudal dengan Persamaan (3.60).

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (3.60)$$

Bila menggunakan tanda negatif, akar yang didapatkan akan selalu berada di belakang rudal. Agar rudal terus bergerak ke depan, dibutuhkan titik referensi yang selalu berada di depannya. Untuk lebih jelasnya,

algoritma yang digunakan untuk menghasilkan lintasan *waypoint* rudal dapat dilihat pada diagram alir di Gambar 3.19 dibawah ini.



Gambar 3.19 Diagram Alir Algoritma *Path Planning* Navigasi *Waypoint*

3.7. Perancangan Pemanduan Navigasi NLGL

Perancangan pemanduan digunakan khusus untuk memberikan perintah yang harus dijalankan autopilot untuk menggerakkan rudal menuju sasaran statisnya. Sasaran statis tersebut berupa lintasan *waypoint* rudal. Pemandu harus mampu menerima informasi lokasi lintasan

maupun titik sarasannya dan lokasi rudal. Dengan navigasi NLGL, rudal akan dipandu oleh pemandunya agar selalu mengikuti lintasan yang harus dilaluinya.

Dari lintasan yang telah direncanakan, rudal akan menentukan titik yang berada pada lintasan di depannya dengan panjang L_1 sebagai referensinya. Untuk mendapatkan respon dengan time constant respon rudal terhadap lintasan τ yang konstan, maka rasio antara L_1 dan V_m haruslah konstan pula. Dengan demikian, panjang L_1 menjadi adaptif bergantung dari magnitudo kecepatan badan rudal. Panjang garis L_1 dapat ditentukan dengan menggunakan persamaan berikut di bawah ini.

$$L_1 = \tau_d V_m \quad (3.61)$$

Dimana τ_d adalah time constant respon rudal terhadap lintasan yang diinginkan. Selanjutnya dengan panjang L_1 ini kemudian dapat diketahui titik referensinya. Titik referensi ini kemudian dihitung sudut LOS. Sudut LOS yang dibentuk antara lintasan dan posisi bumi dalam koordinat bumi terdapat dua yaitu sudut vertikal terhadap ketinggian dan bidang horizontal yang dapat ditentukan dengan menggunakan fungsi trigonometri. Sudut-sudut LOS pada bidang horizontal dan vertikal tersebut dapat ditentukan dengan Persamaan (3.62) - (3.66).

$$\begin{aligned} e_x &= x_r - x_m \\ e_y &= y_r - y_m \end{aligned} \quad (3.62)$$

$$\begin{aligned} e_z &= z_r - z_m \\ \eta_{y_1} &= \sin^{-1} \left(\frac{e_y}{\sqrt{e_x^2 + e_y^2}} \right) \end{aligned} \quad (3.63)$$

$$\begin{aligned} \eta_y &= \eta_{y_2} - \eta_{y_1} \\ \eta_y &= \psi - \eta_{y_1} \end{aligned} \quad (3.64)$$

$$\eta_{z_1} = \sin^{-1} \left(\frac{e_z}{\sqrt{e_x^2 + e_y^2 + e_z^2}} \right) \quad (3.65)$$

$$\begin{aligned} \eta_y &= \eta_{z_2} - \eta_{z_1} \\ \eta_z &= \theta - \eta_{z_1} \end{aligned} \quad (3.66)$$

Dimana:

x_m, y_m, z_m = Posisi rudal

x_r, y_r, z_r = Posisi titik lintasan referensi dengan panjang L_1 dari rudal

e_x = Selisih jarak antara posisi rudal dan lintasan pada sumbu x

e_y = Selisih jarak antara posisi rudal dan lintasan pada sumbu y

e_z = Selisih jarak antara posisi rudal dan lintasan pada sumbu z

Setelah mendapatkan informasi sudut η_y dan η_z , pemandu NLGL mengalkulasikan berapa percepatan rudal dalam koordinat bumi yang diperintahkan untuk autopilot agar bekerja. Dalam perancangan navigasi NLGL khususnya untuk percepatan sumbu z, terdapat modifikasi untuk menghilangkan *steady-state error* rudal terhadap lintasan. Metodenya yaitu melakukan integral terhadap *error* ketinggian tersebut dan menjadikannya sebagai *offset* percepatan referensi sumbu z. Persamaan untuk mendapatkan offset percepatan tersebut dapat dilihat di bawah ini.

$$a_{offset}(k) = a_{offset}(k-1) + K_i d_z \quad (3.67)$$

Dimana:

a_{offset} : Percepatan *offset* yang ditambahkan

K_i : *Gain* integral

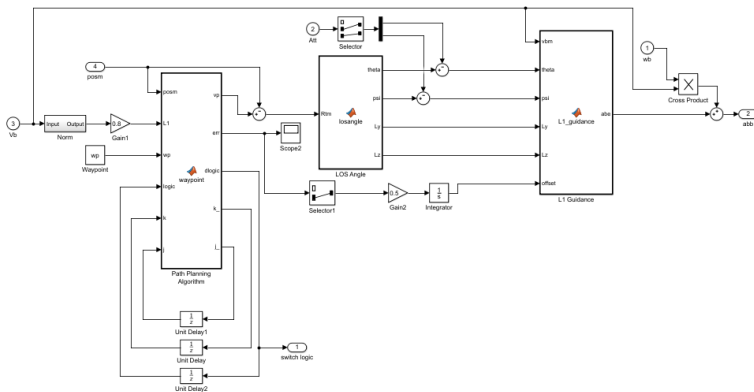
d_z : *Error* ketinggian rudal terhadap lintasan

Sehingga persamaan percepatan vertikal dapat ditulis ulang menjadi Persamaan (3.68) berikut.

$$a_z = 2 \frac{V_m^2}{L_1} \sin \eta_z + a_{offset} \quad (3.68)$$

Namun, percepatan ini kemudian harus dikonversikan menjadi percepatan dalam koordinat badan karena autopilot yang dirancang adalah autopilot kecepatan koordinat badan. Oleh karena itu, percepatan koordinat badan dapat diperoleh dengan Persamaan (3.69).

$$a_{bb} = a_{be} + \omega_b \times V_b \quad (3.69)$$

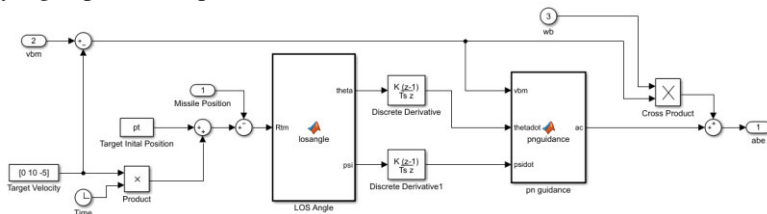


Gambar 3.20 Blok Simulink Pemanduan NLGL

3.8. Perancangan Pemanduan *Proportional Navigation*

Pemanduan *proportional navigation* dirancang khusus untuk memandu rudal untuk sasaran bergerak. Sasaran bergerak hanya dengan kecepatan konstan tertentu tanpa percepatan. Posisi dan kecepatannya diasumsikan diketahui setiap saat. Pemandu *proportional navigation* akan selalu mengarahkan rudal mendekati sasarannya.

Sama seperti pemanduan NLGL, pemanduan PN juga mengalkulasikan sudut LOS yang dibentuk antara rudal dan sasarannya. Pemanduan PN juga menggunakan informasi kecepatan rudal, namun karena sasarannya bergerak, kecepatan rudal yang dihitung adalah kecepatan relatif terhadap kecepatan sasarannya. Karena sudut LOS pada pemanduan NLGL dan PN adalah sama, maka bisa dinyatakan bahwa $\lambda = \eta$. Konstanta proporsional pemandu PN yang akan digunakan adalah 2. Kemudian untuk mendapatkan nilai LOS *rate*, nilai sudut LOS kemudian diturunkan secara diskrit. Percepatan pada koordinat bumi yang dihasilkan kemudian diubah menjadi percepatan dalam koordinat badan. Dengan demikian, pemanduan PN dirancang dalam simulasi Simulink yang dapat dilihat pada Gambar 3.21.



Gambar 3.21 Blok Simulink Pemanduan PN

Masukkan dari blok pemanduan PN yaitu kecepatan rudal, posisi rudal, kecepatan sudut rudal dan posisi sasaran bergerak. Sudut LOS dihitung dengan menggunakan MATLAB *Function* dan kemudian diturunkan untuk mendapatkan LOS *rate* dan selanjutnya diolah menjadi perintah percepatan.

3.9. Perancangan Pengujian Sistem

Setelah seluruh bagian sistem telah dirancang, maka tahap selanjutnya adalah merencanakan pengujian setiap komponen sistem apakah hasilnya bekerja dan dapat memenuhi kriteria desain yang

diinginkan. Pengujian ini dilakukan untuk melihat apakah hasil pengujian sesuai dengan perancangan yang telah dibuat sebelumnya. Pengujian masing-masing komponen tersebut juga dilakukan untuk mendapatkan hasil pengujian sistem keseluruhan yang baik.

3.9.1. Perancangan Pengujian Model Matematika Rudal

Pengujian model matematika rudal dilakukan dengan memberikan masukkan *Thrust* dengan sudut defleksi sirip yang ditentukan pada kondisi awal rudal tertentu. Pengujian dilakukan untuk menguji apakah model rudal yang didesain mampu menggambarkan gerakan rudal dalam dunia nyata dan menguji pada fase *boost* apakah rudal dapat naik tanpa terjatuh ke tanah dengan kondisi tersebut.

3.9.2. Perancangan Pengujian Linierisasi Dinamika Rudal

Pengujian linierisasi dinamika rudal terdiri dua tahapan yaitu pengujian dinamika rotasi yang kemudian dilanjutkan dengan pengujian dinamika translasi. Sinyal masukkan referensi yang digunakan adalah sinyal masukkan referensi konstan. Pengujian dilakukan untuk melihat apakah dinamika rudal telah linier dan interaksi antar dinamika telah hilang. Karakteristik hasil pengujian yang diamati yaitu *time constant*, *overshoot* dan *steady-state error*.

3.9.3. Perancangan Pengujian DMPC

Pengujian pengaturan DMPC diawali dengan identifikasi dinamika kecepatan rudal yang telah dilinierisasi dengan mengikuti langkah pada perancangan. Kemudian langkah berikutnya ialah memasukkan model matematika *state-space* diskrit sistem. Referensi yang dimasukkan pada pengujian ini adalah berupa *step* dan trayektori referensi hasil prediksi.

3.9.4. Perancangan Pengujian Pemanduan NLGL

Pengujian pemanduan NLGL dilakukan dengan menggunakan masukan *waypoint* yang akan dilewati dengan sistem pengaturan DMPC. Pengujian ini sekaligus menguji algoritma *path planning* yang telah dirancang sebelumnya yang digabung dengan pemandu NLGL. Pemandu diuji dengan melihat apakah rudal berhasil bergerak pada lintasannya melewati semua *waypoint* yang telah ditentukan berdasarkan sinyal pemandu yang dihasilkan pemanduan NLGL dengan melihat *cross-track error* lintasan dan *miss distance* pada setiap *waypoint*. *Cross-track error* adalah perbedaan posisi rudal terhadap lintasan lurus nya dan

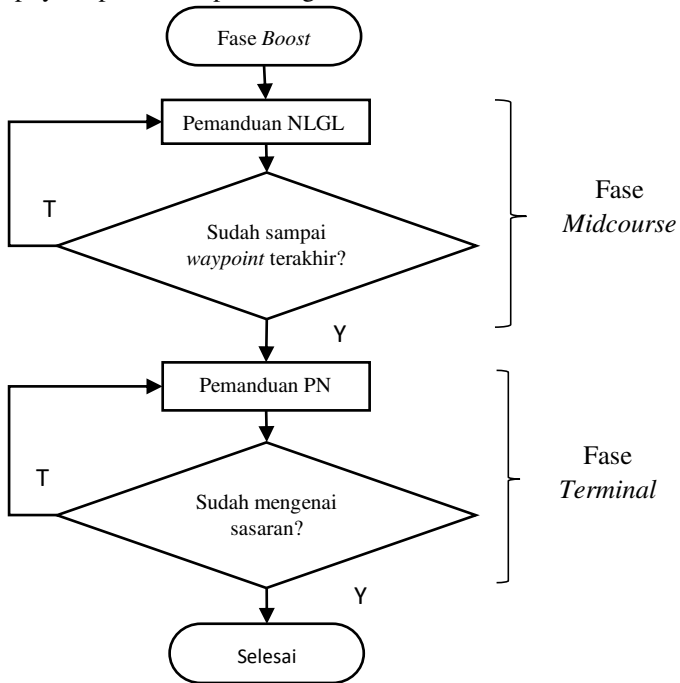
miss distance waypoint adalah jarak terdekat rudal terhadap masing-masing *waypoint*.

3.9.5. Perancangan Pengujian *Proportional Navigation*

Pengujian pemandu *proportional navigation* dilakukan dengan memberikan masukkan lokasi sasaran dinamis atau bergerak tanpa menggunakan lintasan tertentu setelah fase *boost*. Kemudian, pemandu PN diuji apakah pemandu dapat membawa rudal bergerak mengenai sasarannya dan dikatakan mengenai sasarannya apabila *miss distance* rudal terhadap sasarannya kurang dari 1 meter.

3.9.6. Perancangan Pengejaran Sasaran Bergerak

Perancangan pengujian ini untuk menyimulasikan rudal melewati seluruh fasenya yaitu fase *boost*, *midcourse* dan *terminal*. Lebih lengkapnya dapat dilihat pada diagram alir di bawah ini.



Gambar 3.22 Diagram Alir Pengejaran Sasaran

[Halaman ini sengaja dikosongkan]

BAB 4 HASIL DAN ANALISA DATA

Bab ini akan menampilkan dan menganalisis pengujian yang telah dirancang untuk masing-masing komponen sistem yang telah dibuat sebelumnya. Penjelasan untuk hasil tiap komponen tersebut akan dijabarkan pada sub bab berikut di bawah ini.

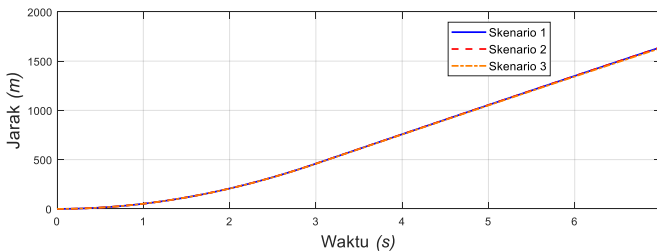
4.1. Pengujian Model Matematika Rudal

Pengujian model rudal dilakukan memiliki tujuan untuk mendapatkan kondisi awal yang tepat terutama mendapatkan sudut θ yang tepat agar rudal dapat naik dan terbang ke udara tanpa terjatuh menyentuh tanah terlebih dahulu. Masukkan pada model ini adalah masukkan gaya dorong *thrust* yang telah ditentukan dan sudut defleksi $\delta_a, \delta_e, \delta_r$ diberi nilai 0. Rudal juga diuji beberapa kondisi awal untuk mendapatkan kondisi awal yang tepat. Kondisi awal rudal yang akan diuji disajikan dalam Tabel 4.1.

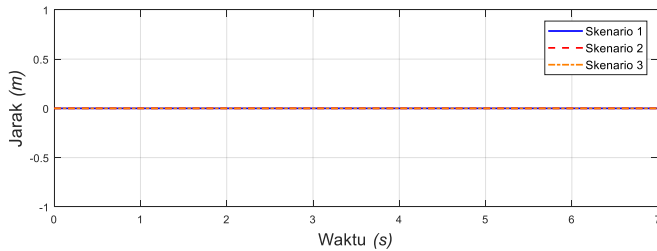
Tabel 4.1. Tabel Pengujian Kondisi Awal Rudal

Kondisi Awal	Skenario 1	Skenario 2	Skenario 3
x_e, y_e, z_e (m)	[0 0 0]	[0 0 -5]	[0 0 -5]
u, v, w (m/s)	[10^{-4} 0 0]	[10^{-4} 0 0]	[10^{-4} 0 0]
ϕ, θ, ψ (rad)	[0 0.05 0]	[0 0.05 0]	[0 0.085 0]
p, q, r (rad/s)	[0 0 0]	[0 0 0]	[0 0 0]

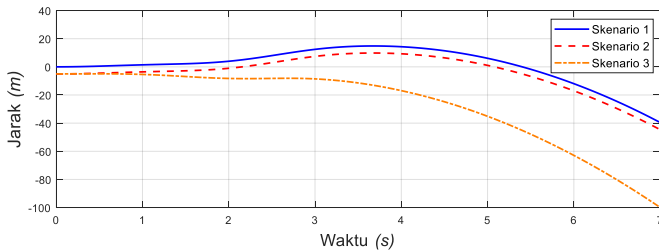
Hasil dari simulasi jarak tempuh yang dialami rudal dengan menggunakan masing-masing skenario kondisi awal yang diberikan dapat dilihat pada Gambar 4.1 (a) – (c).



(a)



(b)

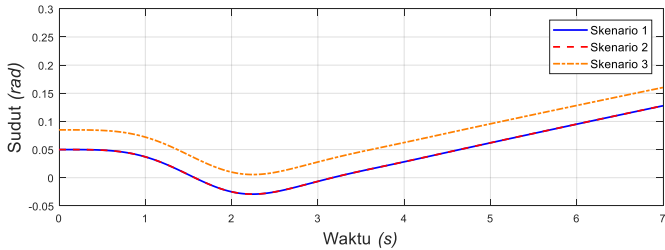


(c)

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

Gambar 4.1 diatas menunjukkan bahwa rudal dengan kondisi awal yang diberikan dapat terbang di udara, akan tetapi dua skenario kondisi awal rudal pertama menyebabkan rudal terjatuh terlebih dahulu di bawah tanah dengan ditunjukkan oleh nilai z_e yang positif. Hanya dengan kondisi awal skenario ketiga, rudal dapat terbang tanpa terjatuh. Selain melihat ketinggian rudal, posisi y rudal tidak mengalami perubahan karena masukkan yang diberikan rudal tidak memengaruhi nilai posisi y . Dengan gaya pendorong yang bekerja terus menerus menyebabkan rudal terus maju ke depan yang menyebabkan grafik x_e selalu naik. Berdasarkan hasil pengujian model matematika rudal diatas, model rudal dapat digunakan untuk menyimulasikan rudal yang sebenarnya.

Alasan mengapa rudal terjatuh sebelum terbang ke udara karena sudut θ yang digambarkan pada Gambar 4.2, menunjukkan bahwa sudut θ rudal mengalami penurunan di awal respon. Hal ini disebabkan oleh gaya gravitasi yang menarik rudal ke bawah dan rudal masih mengalami percepatan untuk mencapai kecepatan nominalnya dari gaya *thrust* rudal. Pada skenario terlihat sudut θ rudal tidak mencapai negatif yang berarti rudal tidak sampai mengarah ke bawah.



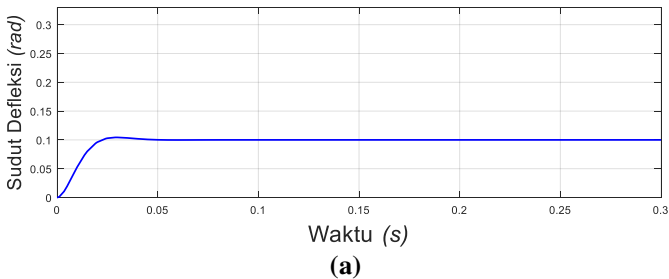
Gambar 4.2 Sudut *Pitch* Rudal (θ)

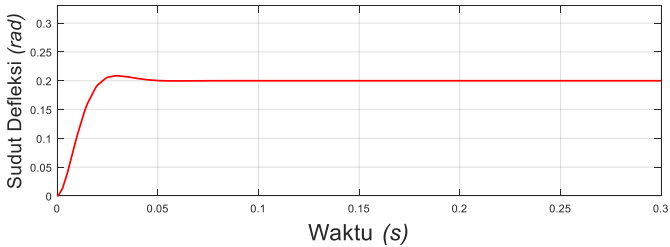
Dengan hasil pengujian diatas, kondisi awal yang akan digunakan pada pengujian-pengujian selanjutnya adalah kondisi awal skenario yang ketiga.

4.2. Pengujian *Nonlinear Feedback Linearization*

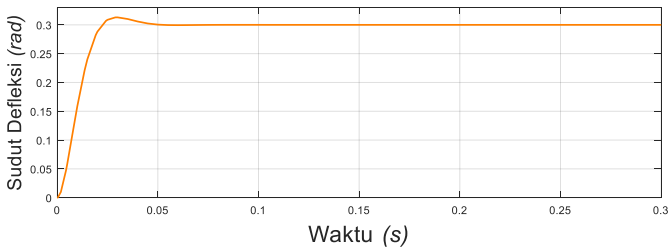
Tujuan dilakukannya pengujian ini adalah untuk membuktikan bahwa dengan metode *nonlinear feedback linearization*, dinamika translasi dan dinamika rotasi rudal yang non-linier dapat menjadi linier. Selain itu, menguji apakah masukkan yang diberikan pada setiap variabel tidak akan memengaruhi variabel yang lainnya untuk membuktikan bahwa sistem juga sudah *decoupled*.

Pengujian diawali dengan menguji respon aktuator sirip. Pengujian dilakukan untuk membuktikan bahwa sirip dapat bergerak sesuai dengan kriteria desain sistem dan dapat mengikuti referensi dengan stabil. Pengujian dilakukan dengan memasukkan masing-masing sudut defleksi $\delta_a, \delta_e, \delta_r$. Masukkan sudut defleksi sirip yang dimasukkan sebesar 0.1 rad untuk *aileron*, 0.2 rad untuk *elevator* dan 0.3 rad untuk *rudder*. Hasil simulasi dapat dilihat pada Gambar 4.3 dibawah ini.



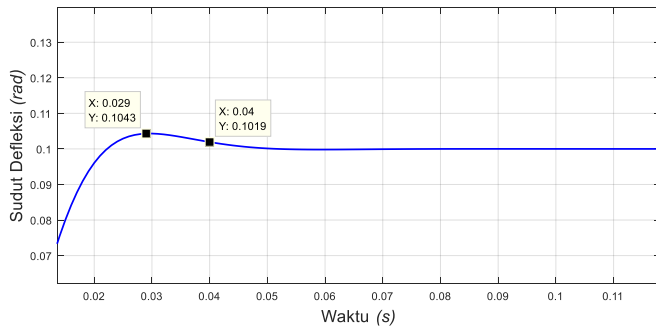


(b)



(c)

Gambar 4.3 Sudut Defleksi (a) *Aileron*, (b) *Elevator*, (c) *Rudder*



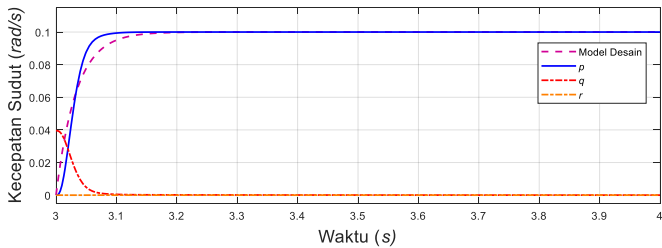
Gambar 4.4 Titik *Overshoot* Maksimum dan Waktu *Settling* dari Respon Sudut Defleksi *Aileron* (*Zoom In*)

Dari Gambar 4.3 yang dihasilkan diatas, dapat disimpulkan bahwa respon sistem yang dihasilkan dapat mengikuti referensi dengan stabil. Gambar 4.4 juga menunjukkan bahwa *overshoot* yang ditunjukkan dengan *datatip* sebelah kiri dan respon mencapai waktu *settling*-nya yang ditunjukkan dengan *datatip* sebelah kanan, mendekati dengan kriteria

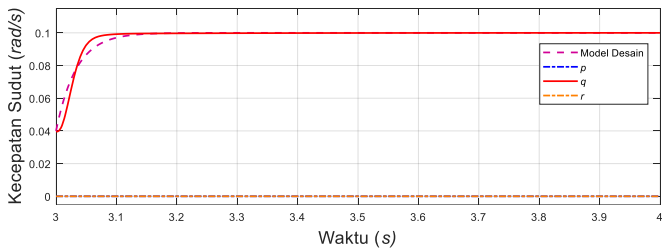
desain yang telah ditentukan pada perancangan yaitu maksimum *overshoot* 4.3% dan waktu *settling* 0.037 detik.

Selanjutnya adalah pengujian linierisasi dinamika rudal. Pengujian linierisasi dilakukan dua tahap yaitu pertama menguji hasil linierisasi dinamika kecepatan rotasi dan kedua yaitu hasil linierisasi dinamika kecepatan translasi. Pengamatan pengujian dilakukan setelah rudal mengalami fase *boost* dimana diasumsikan rudal tidak mengalami pengendalian saat fase tersebut dan rudal berhasil terbang ke udara.

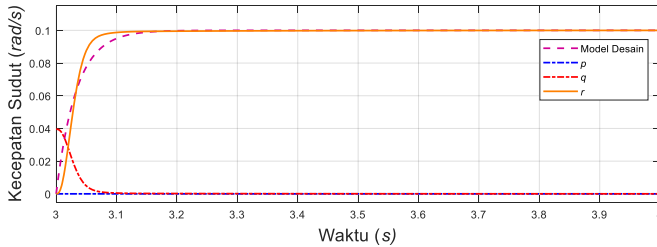
Pengujian rotasi dilakukan sebanyak tiga kali yaitu dengan memberikan masukan referensi kecepatan rotasi pada tiap sumbu koordinat badan. Masukkan yang diberikan hanya salah satu sumbu pada setiap pengujian untuk membuktikan sifat *coupled* sistem yang telah hilang. Selain itu, masing-masing sumbu juga diberikan masukan referensi sinyal masukan *step* dan dibandingkan dengan model desain untuk membandingkan *time constant*-nya. Pengujian dengan memberikan masukan sinyal *step* dengan amplitudo 0.1 rad/s yang diambil selama 4 detik.



(a)



(b)



(c)

Gambar 4.5 Pengujian Respon *Step* Hasil Linierisasi Kecepatan Rotasi (a) p , (b) q dan (c) r

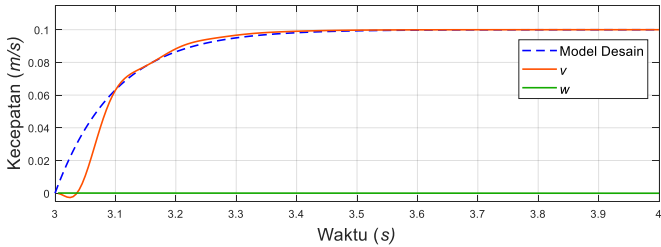
Dari hasil respon pengujian linierisasi kecepatan rotasi rudal, terlihat bahwa rudal dapat mengikuti referensi masukannya dan dinamika rotasi rudal sudah linier. Selain itu, masukkan masing-masing komponen kecepatan rotasi juga tidak memengaruhi komponen kecepatan rotasi lainnya sehingga linierisasi dan *decoupler* rudal bekerja sesuai dengan yang diharapkan. Waktu pengujian dilakukan setelah fase *boost* yaitu setelah 3 detik. Selanjutnya ialah mendapatkan nilai *time constant* dan *steady-state error* respon tersebut dengan menggunakan fitur *datatip cursor* dan *zoom* pada MATLAB untuk mengetahui nilainya. Nilai-nilai karakteristik kecepatan rotasi rudal tersebut dapat dilihat pada Tabel 4.2.

Tabel 4.2. Tabel Karakteristik Kecepatan Rotasi Rudal

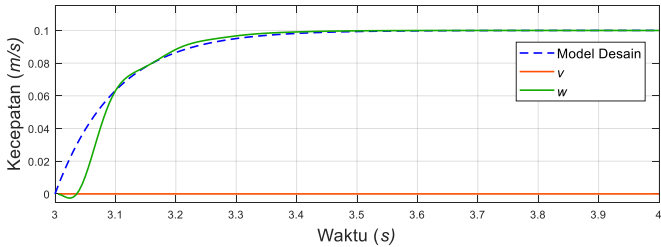
Karakteristik	Kecepatan Rotasi Rudal		
	p	q	r
<i>Time constant</i> (s)	0.0332	0.0333	0.0333
<i>Overshoot</i> (%)	0%	0%	0%
<i>Steady-state error</i>	1×10^{-6}	2×10^{-6}	1×10^{-6}

Setelah pengujian blok linierisasi kecepatan rotasi telah dilakukan, langkah selanjutnya adalah menggabungkan blok linierisasi kecepatan rotasi dengan linierisasi kecepatan translasi. Karena model rudal yang digunakan yaitu *skid-to-turn*, maka nilai *roll rate* p akan diberikan pengaturan proporsional untuk menjaga p tetap nol. Blok linierisasi kecepatan translasi akan dimasukkan dua referensi untuk masing-masing kecepatan translasi v dan w . Referensi yang diberikan berupa sinyal masukkan *step* dengan amplitudo 0.2 rad/s dan akan dibandingkan

dengan model desain yang diinginkan. Pengujian ini membuktikan bahwa kecepatan translasi telah linier dan *decoupled*.



(a)



(b)

Gambar 4.6 Respon Kecepatan Translasi Rudal (a) v dan (b) w

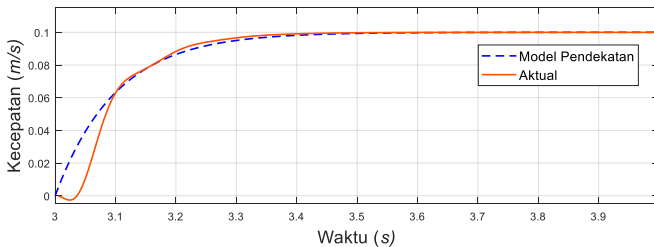
Dari hasil pada Gambar 4.6, terlihat bahwa dinamika rudal translasi yang telah digabungkan dengan dinamika rotasi yang dilinerisasi dapat menghasilkan kecepatan translasi yang linier dan telah *decoupled* serta mampu mengikuti referensi yang diberikan. Hasil tersebut juga menunjukkan adanya *gain* sebesar 0.5 dimana hasilnya menjadi 0.1 m/s dari masukkan 0.2 m/s , yang mana hal ini sesuai dengan perancangan linierisasi kecepatan translasi rudal. Selain itu, pada awal respon terdapat keunikan yaitu respon kecepatan translasi bergerak ke arah yang berlawanan di awal respon. Hal ini disebabkan oleh linierisasi kecepatan rotasi rudal masih mengalami transien hingga mencapai kondisi *steady-state*-nya. Ini yang menyebabkan adanya seperti *inverse response* untuk hasil linierisasi kecepatan translasi rudal. Karakteristik ini dapat dihilangkan apabila tidak adanya transien pada kecepatan rotasi rudal. Untuk karakteristik lebih lengkap dari kecepatan translasi linier dapat dilihat pada Tabel 4.3.

Tabel 4.3. Tabel Karakteristik Respon Kecepatan Translasi Rudal

Karakteristik	Kecepatan Translasi Rudal	
	Kecepatan v	Kecepatan w
<i>Time constant</i> (s)	0.1003	0.1003
<i>Overshoot</i> (%)	0	0
<i>GainOverall</i>	0.4998	0.4998

4.3. Pengujian *Discrete Model Predictive Control*

Pengujian DMPC diawali dengan identifikasi respon kecepatan rudal yang telah dilinierisasi. Identifikasi dilakukan untuk mendapatkan model yang lebih akurat dari hasil observasi respon dinamika yang telah dilinierisasi. Berdasarkan rancangan untuk mendapatkan model matematika diskrit dan hasil pengujian linierisasi kecepatan rudal, didapatkan informasi *time constant* dan *gain* yang dapat dilihat pada hasil pengamatan di Tabel 4.3. Perbandingan pendekatan respon orde satu yang akan digunakan dan respon kecepatan rudal dapat dilihat pada Gambar 4.7.



Gambar 4.7 Respon Kecepatan v dan Model Pendekatan Orde Satu

Dari berdasarkan Gambar 4.7, respon kecepatan rudal mendekati dengan respon sistem orde satu. Dengan demikian, kecepatan translasi linier rudal dapat dimodelkan dalam sistem orde satu dimana parameternya didapatkan untuk memperoleh model *state-space* kontinyu yang dituliskan dalam Persamaan (4.1) berikut.

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -9.97 & 0 \\ 0 & -9.97 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} + \begin{bmatrix} 4.998 & 0 \\ 0 & 4.998 \end{bmatrix} \begin{bmatrix} v_{ref} \\ w_{ref} \end{bmatrix} \quad (4.1)$$

Persamaan model *state-space* kontinyu tersebut harus didiskritisasi menjadi model *state-space* diskrit karena pengaturan DMPC adalah pengaturan diskrit. Oleh karena itu, dengan menggunakan fungsi *c2d*

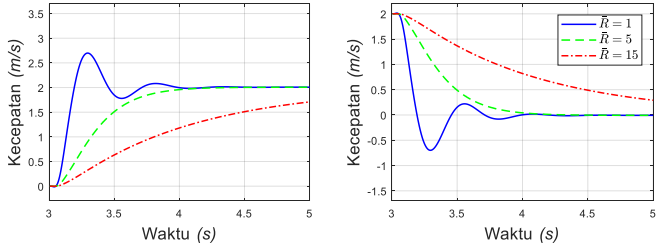
pada MATLAB dan *time sampling* sebesar 0.005 s didapatkan model *state-space* diskrit kecepatan rudal yaitu dituliskan pada Persamaan (4.2).

$$\begin{bmatrix} v(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} 0.9514 & 0 \\ 0 & -9.514 \end{bmatrix} \begin{bmatrix} v(k) \\ w(k) \end{bmatrix} + \begin{bmatrix} 0.0244 & 0 \\ 0 & 0.0244 \end{bmatrix} \begin{bmatrix} v_{ref}(k) \\ w_{ref}(k) \end{bmatrix} \quad (4.2)$$

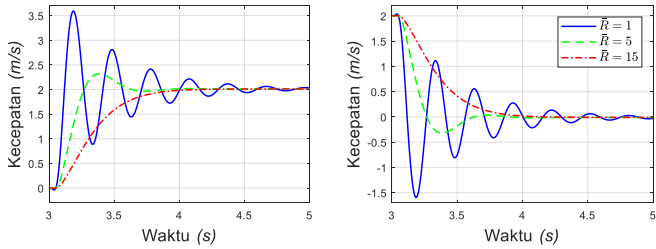
Tahap selanjutnya adalah menguji model yang dimasukkan dalam DMPC dengan menggabungkan prediksi masukkan referensi. Sinyal uji referensi prediktif yang akan diuji dalam DMPC terdapat dua jenis yaitu sinyal *step* dan trayektori kuadratik. Sinyal masukkan referensi *step* tersebut bernilai 2 dan 3 untuk referensi v dan w . Karena masukkannya berupa *step*, maka referensi prediksinya juga konstan. Parameter DMPC yang akan diuji yaitu bobot \bar{R} dan nilai panjang prediksi N_p dan N_c . Skenario yang akan diuji dapat dilihat pada Tabel 4.4. Pengujian dengan skenario pada tabel tersebut dilakukan untuk membandingkan dan menganalisis pengaruh parameter terhadap respon sistem untuk pengaturan DMPC. Hasil pengujian dengan menggunakan Tabel 4.4 dapat dilihat pada Gambar 4.8 dan Gambar 4.9.

Tabel 4.4. Tabel Pengujian Parameter DMPC

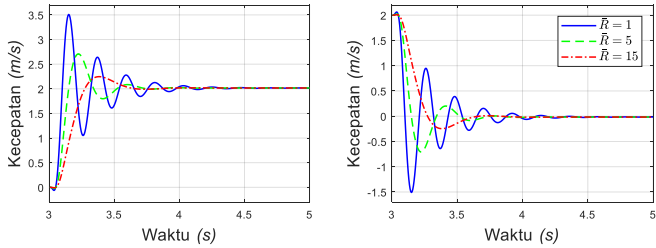
Skenario	N_p	N_c	\bar{R}
1	3	2	1
			5
			15
2	6	4	1
			5
			15
3	12	8	1
			5
			15
4	30	14	1
			5
			15



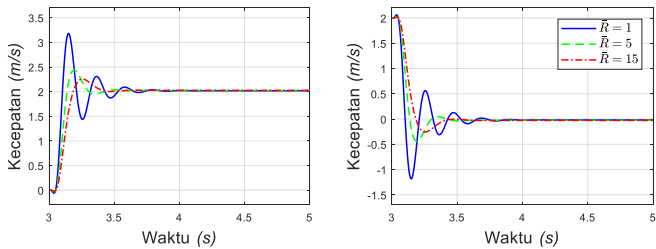
(a)



(b)

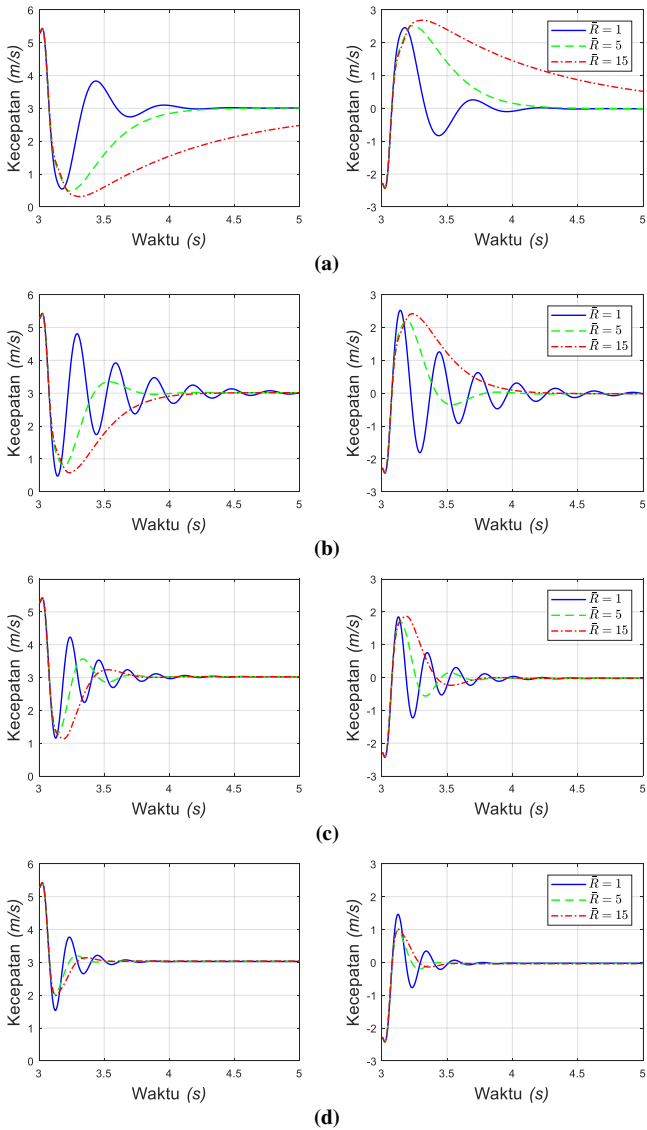


(c)



(d)

Gambar 4.8 (Kiri) Respon Kecepatan v dengan Masukkan *Step* Skenario (a) 1, (b) 2, (c) 3, (d) 4, (Kanan) Grafik *Error* Respon v



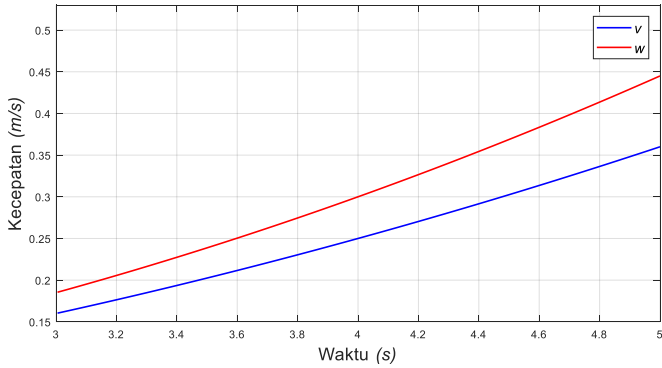
Gambar 4.9 (Kiri) Respon Kecepatan w dengan Masukkan *Step* Skenario (a) 1, (b) 2, (c) 3, (d) 4, (Kanan) Grafik *Error Respon* w

Berdasarkan hasil respon kecepatan rudal dengan pengaturan DMPC yang disajikan pada Gambar 4.8 dan Gambar 4.9, dapat dilihat bahwa pemilihan nilai N_p , N_c dan \bar{R} sangat memengaruhi bentuk respon keluaran kecepatan rudal yang dihasilkan.

Dari pengamatan gambar-gambar tersebut, semakin kecil nilai bobot \bar{R} , maka semakin besar *overshoot* respon yang dihasilkan. Ini disebabkan oleh besarnya bobot pada perubahan sinyal kendali pada fungsi tujuan memengaruhi perubahan sinyal kendali dalam meminimumkan fungsi harga yang menyebabkan perubahan sinyal kendali menjadi besar. Oleh karena itu, hasil respon dengan nilai bobot \bar{R} lebih kecil menghasilkan respon yang lebih agresif dibandingkan dengan nilai bobot \bar{R} yang besar.

Kemudian, nilai panjang horison prediksi N_p dan panjang horison sinyal prediksi N_c juga memengaruhi respon. Respon dengan nilai N_p dan N_c yang besar menghasilkan respon yang lebih cepat dengan nilai \bar{R} yang sama. Hal itu terjadi karena dengan mendapatkan informasi prediksi sistem dari model yang lebih jauh, pengaturan DMPC dapat menggunakan informasi tersebut untuk menghasilkan sinyal kendali yang lebih cepat dibandingkan dengan horison prediksi yang pendek. Akan tetapi, bila nilai horison prediksi N_p dan N_c yang terlalu besar, sinyal kendali yang dihasilkan pengaturan DMPC tidak akan memengaruhi perilaku sistem secara signifikan. Ilustrasi untuk penjelasan ini dapat dilihat pada Gambar 4.9 (c) dan (d) dimana respon sistem keduanya mulai tampak tidak berbeda jauh dibandingkan dengan respon Gambar 4.9 (a) dan (b). Selain itu, dengan menambah horison prediksi N_p dan N_c justru akan mengonsumsi waktu komputasi yang lebih lama dan akan memberikan hasil yang tidak optimal.

Dengan melihat pengujian untuk masukkan referensi konstan, pengaturan DMPC dapat memberikan sinyal kendali yang mampu mengendalikan kecepatan rudal konstan dengan *steady-state error* dari *gain* 0.5 dapat dihilangkan dengan pengaturan DMPC ini. Selanjutnya adalah menguji pengaturan DMPC dengan masukkan referensi yang dinamis yaitu berupa trayektori kuadrat. Trayektori yang digunakan adalah trayektori kuadrat karena prediksi yang digunakan adalah persamaan polinomial waktu kuadrat. Oleh sebab itu, diasumsikan prediksi masukkan referensi rudal selalu cocok dengan masukkan rudal yang sebenarnya. Sinyal referensi yang digunakan pada pengujian ini dapat dilihat pada Gambar 4.10.

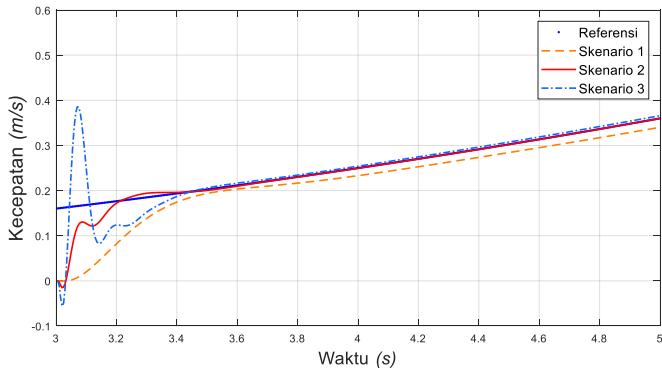


Gambar 4.10 Trayektori Referensi Kecepatan Rudal

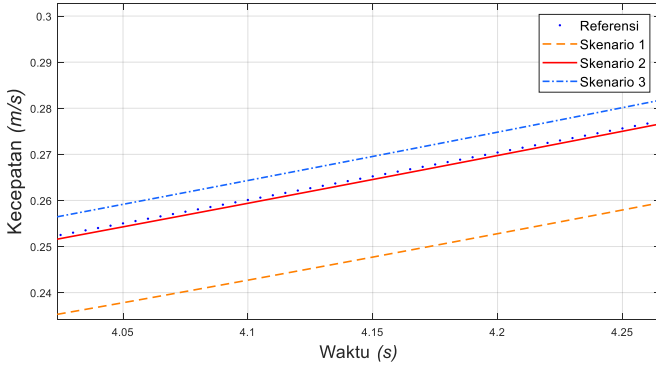
Kemudian dilakukan pengujian pengaturan kecepatan rudal dengan menggunakan pengaturan DMPC dan referensi pada Gambar 4.10. Beberapa skenario untuk parameter N_p, N_c dan \bar{R} dimasukkan untuk melihat responnya. Nilai parameter tersebut dapat dilihat pada Tabel 4.5. Hasilnya disajikan pada Gambar 4.12.

Tabel 4.5. Tabel Pengujian Parameter DMPC dengan Trayektori

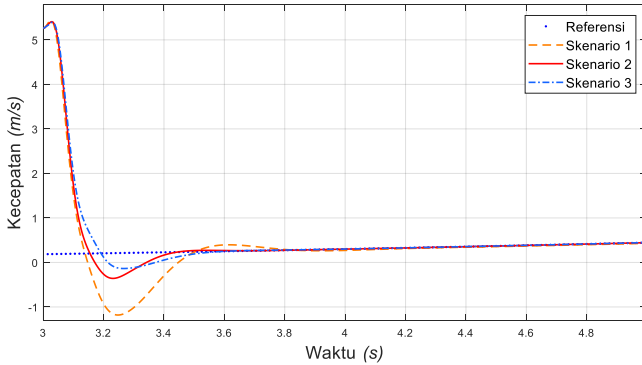
Skenario	N_p	N_c	\bar{R}
1	5	4	6
2	20	15	30
3	80	60	60



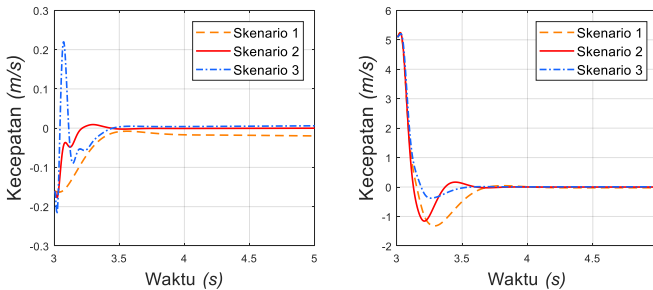
(a)



Gambar 4.11 (a) Respon Kecepatan v Rudal, **(b)** *Zoom In*



Gambar 4.12 Respon Kecepatan w Rudal dengan DMPC

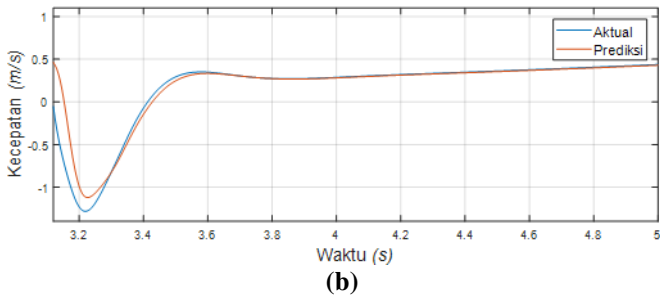
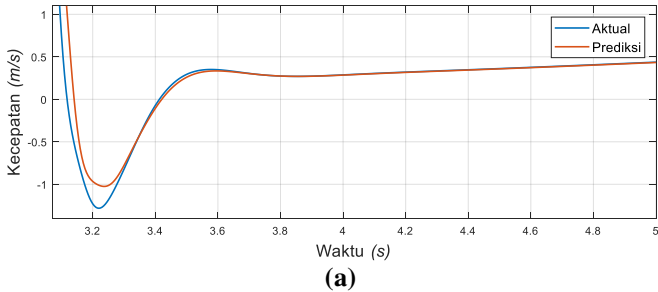


Gambar 4.13 Grafik Kesalahan Respon Kecepatan Translasi Terhadap Referensi (Kiri) v dan (Kanan) w

Tabel 4.6. Tabel RMSE Kesalahan Kecepatan Translasi v dan w

Skenario	Kecepatan Translasi Rudal (m/s)	
	Kecepatan v	Kecepatan w
1	0.0502	1.0259
2	0.0334	0.9701
3	0.0419	0.9823

Berdasarkan pengamatan gambar hasil pengujian dengan tabel di atas, dengan referensi prediktif yang diberikan dilihat bahwa *steady-state* kecepatan rudal dipengaruhi juga oleh besarnya bobot \bar{R} dan nilai panjang horison prediksi N_p . Dari ketiga respon yang diuji, garis warna oranye memiliki *overshoot* yang lebih besar dibandingkan lainnya. Selain itu, juga terlihat bahwa garis merah memiliki nilai *steady-state error* terkecil dibandingkan dengan nilai parameter lainnya pada Gambar 4.11. Dengan demikian, parameter yang akan digunakan pada pengaturan DMPC yaitu parameter pada skenario kedua.



Gambar 4.14 Perbandingan Kecepatan w Aktual dan Prediksi (a) 0.05 Detik (b) 0.1 Detik

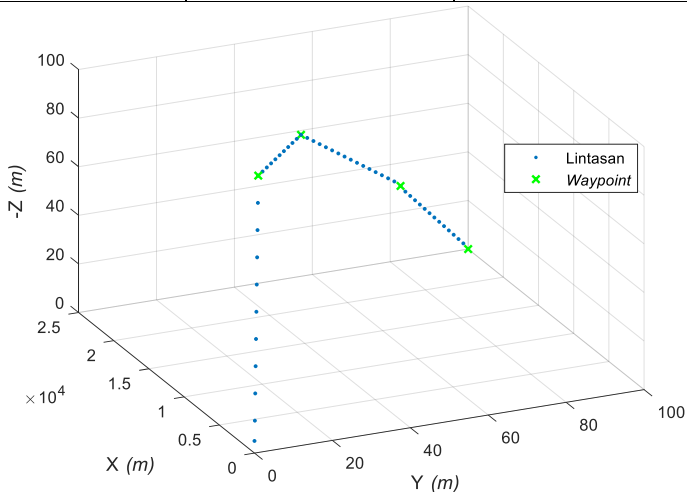
Gambar 4.14 diatas adalah perbandingan grafik kecepatan aktual dan prediksi sejauh 0.05 detik dan 0.1 detik dari model sistem. Kecepatan yang diprediksi terlihat mampu mengikuti bentuk kecepatan yang sebenarnya hingga pada detik ke 3.355 dan 3.445 detik dimana kecepatan aktual dan prediksi berhimpitan sekitar 0.2 detik setelah dari prediksinya. Namun, terdapat selisih prediksi dengan aktual pada kondisi *steady-state* sebesar 0.0026 m/s . Terbukti bahwa pengaturan DMPC dapat melakukan prediksi berdasarkan model sistem.

4.4. Pengujian Pemanduan NLGL

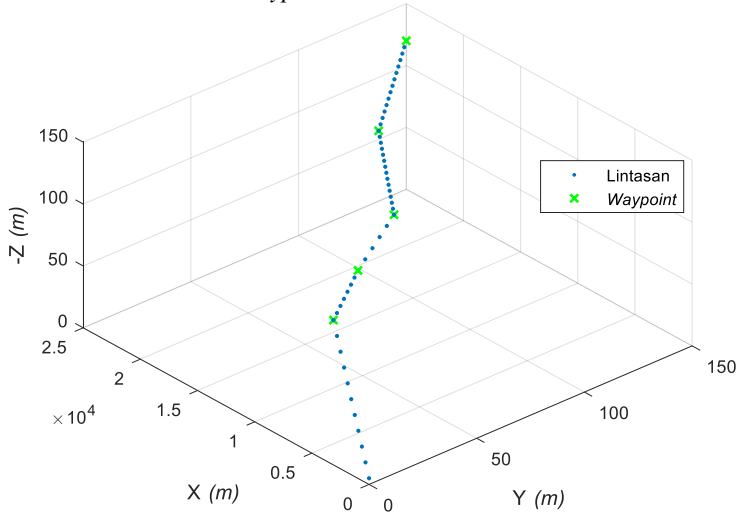
Pengujian ini diawali dengan memasukkan titik-titik *waypoint* akan dilewati rudal sebanyak tiga skenario pada Tabel 4.7 berikut.

Tabel 4.7. Tabel Daftar *Waypoint* untuk *Path Planning*

Waypoint (m)	Skenario	
	1	2
1	[5000 10 -100]	[5000 10 -100]
2	[10000 30 -100]	[8500 40 -100]
3	[18000 70 -50]	[11000 70 -110]
4	[25000 100 0]	[18000 100 -120]
5	–	[25000 150 -120]

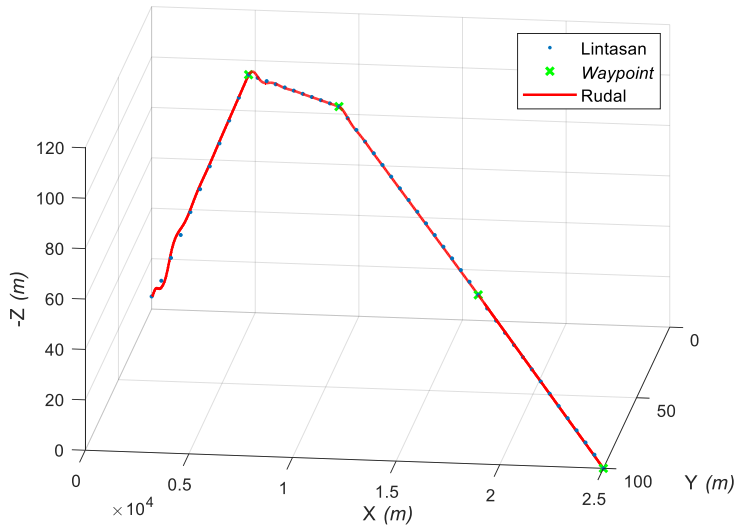


Gambar 4.15 Lintasan *Waypoint* Rudal Skenario 1

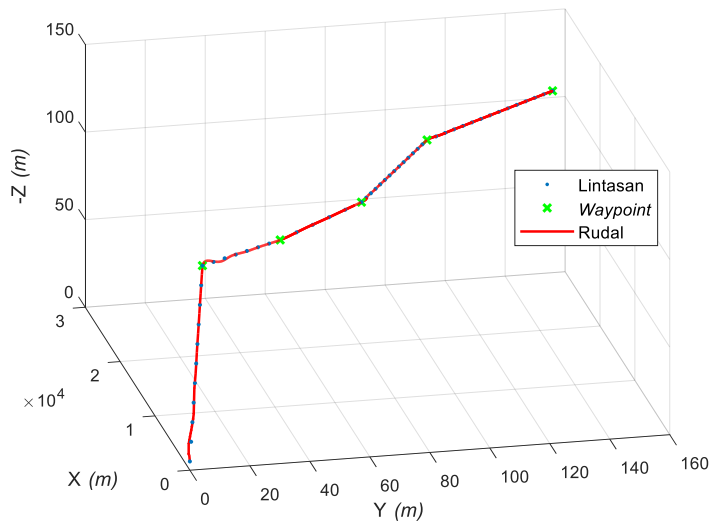


Gambar 4.16 Lintasan *Waypoint* Rudal Skenario 2

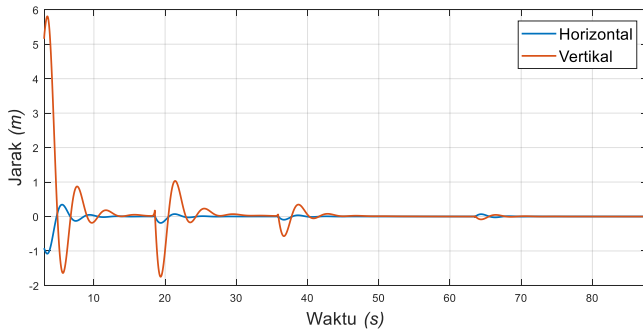
Berdasarkan pengujian *path planning* sebelumnya, rudal akan dipandu untuk mengikuti lintasan yang dihasilkan dengan menggunakan pemanduan NLGL. Algoritma pemandu akan diuji apakah dapat memandu rudal melewati seluruh *waypoint* yang telah ditentukan. Terdapat dua kriteria yang akan digunakan sebagai ukuran bahwa pemanduan rudal dapat memandu rudal melewati masing-masing lintasan. Pertama yaitu *miss distance* rudal terhadap masing-masing *waypoint* yang dilewati rudal. Terakhir yaitu *root mean squared error* (RMSE) dari *cross-track error* horizontal dan vertikal rudal terhadap garis lurus dimana menggambarkan *error* posisi rudal terhadap lintasannya. *Cross-track error* adalah perbedaan posisi aktual rudal terhadap lintasan terbangnya. Selain itu, *cross-track error* rudal terhadap lintasan harus tidak ada *error steady-state*. Pemandu diuji dengan dua skenario lintasan yang telah dibentuk dari *path planning*. Hasil gerak rudal melintasi lintasan antar *waypoint* dapat dilihat pada Gambar 4.17 dan Gambar 4.18. Sedangkan hasil *miss distance* masing-masing *waypoint* dan RMSE *cross-track error* lintasan dan rudal dapat dilihat pada Tabel 4.8 dan Tabel 4.9.



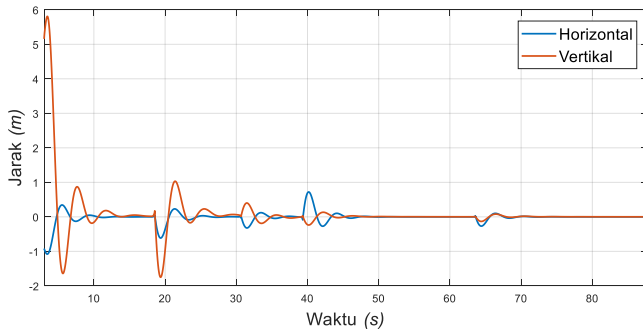
Gambar 4.17 Lintasan Rudal Skenario 1 dengan Pemandu NLGL



Gambar 4.18 Pengujian Pemanduan NLGL pada Skenario 2



Gambar 4.19 *Cross-Track Error* Pemanduan NLGL Skenario 1



Gambar 4.20 *Cross-Track Error* Pemanduan NLGL Skenario 2

Tabel 4.8. Tabel *Miss Distance* Masing-Masing *Waypoint*

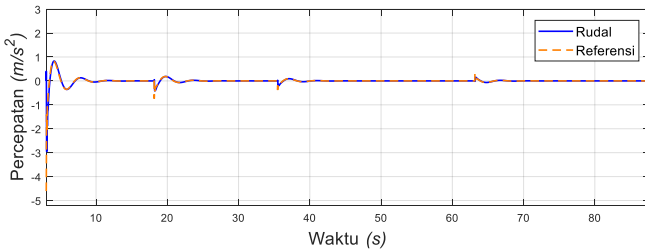
<i>Waypoint</i>	<i>Waypoint Miss Distance (m)</i>	
	Skenario 1	Skenario 2
1	0.1846	0.1907
2	0.1370	0.0333
3	0.0089	0.0874
4	0.1990	0.0456
5	—	0.0794
RMSE	0.1521	0.1034
Rata-Rata	0.1324	0.0872

Dari hasil yang disajikan pada Gambar 4.17 dan Gambar 4.18, terlihat bahwa rudal mampu mengikuti seluruh lintasan yang telah dibentuk dari hasil *path planning*. Dengan pengujian ini, maka pemandu NLGL dapat digunakan untuk memandu rudal untuk lintasan statis.

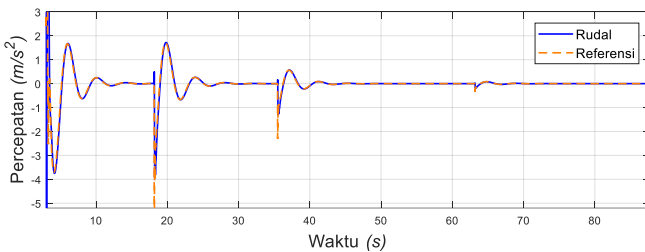
Tabel 4.9. Tabel RMSE *Cross-Track Error* Pemanduan NLGL

Skenario	RMSE <i>Cross-track Error</i> (m)	
	Horizontal	Vertikal
1	0.1273	0.7270
2	0.1721	0.7257
Rata-Rata	0.1497	0.7263

Dengan hasil *cross-track error* yang kurang dari 1 meter menunjukkan bahwa percepatan yang dihasilkan pemandu dapat diikuti dengan baik oleh rudal dengan pengaturan DMPC. Gambar 4.21 menunjukkan perbandingan trayektori percepatan pemandu dan percepatan rudal dalam sumbu y dan sumbu z. Kemudian Gambar 4.23 adalah hasil zoom in dari percepatan pada Gambar 4.22 (b).

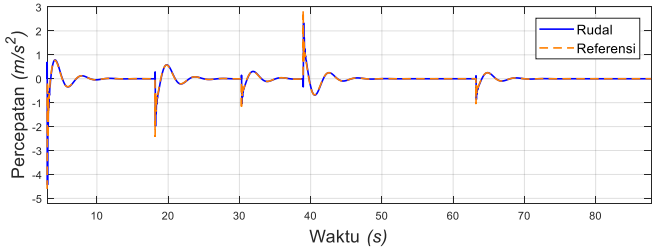


(a)

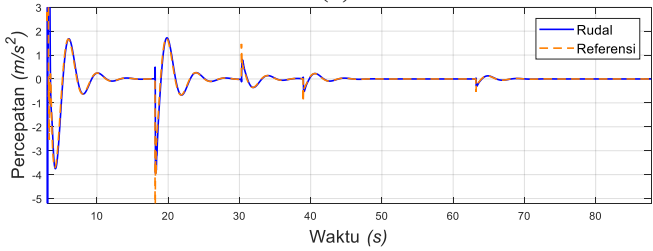


(b)

Gambar 4.21 Perbandingan Percepatan Translasi Skenario 1 (a) Horizontal (b) Vertikal

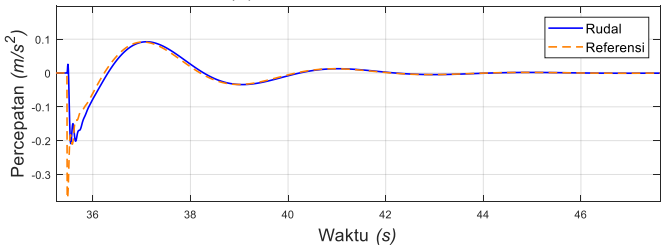


(a)

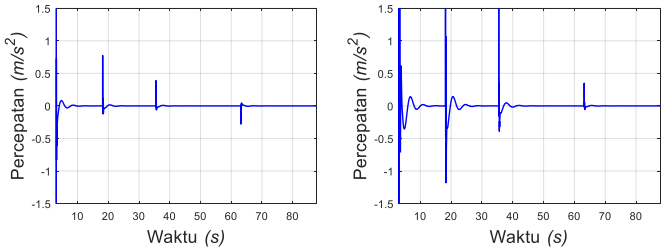


(b)

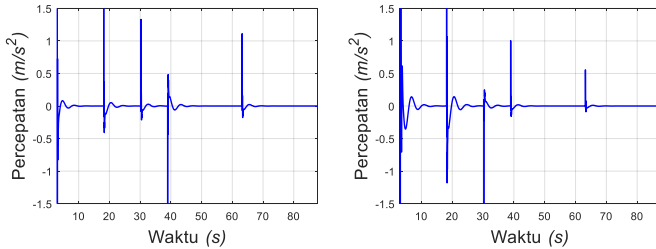
Gambar 4.22 Perbandingan Percepatan Translasi Skenario 2 (a) Horizontal (b) Vertikal



Gambar 4.23 Perbandingan Percepatan Translasi (*Zoom In*)



Gambar 4.24 *Error* Percepatan Rudal dan Pemandu (Kiri) Horizontal (Kanan) Vertikal Skenario 1



Gambar 4.25 Error Percepatan Rudal dan Pemandu (Kiri) Horizontal (Kanan) Vertikal Skenario 2

Tabel 4.10. Tabel Kesalahan Respon Percepatan Rudal dengan Referensi

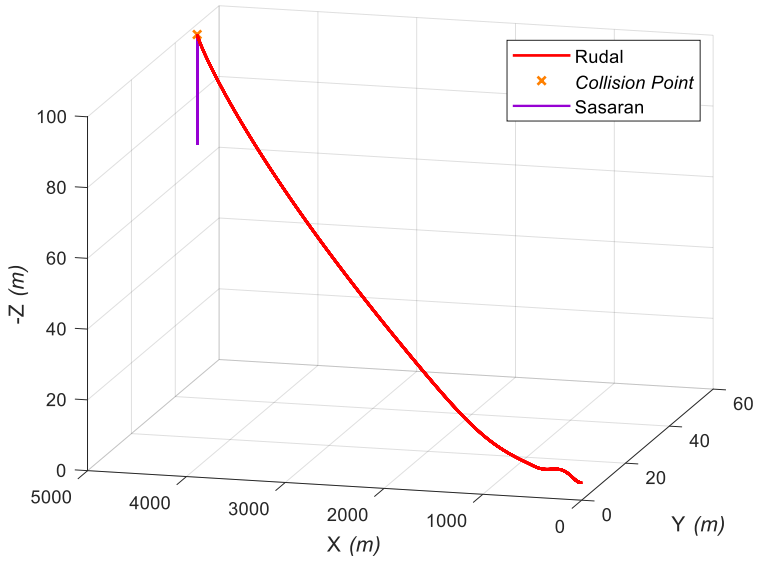
Skenario	RMSE Percepatan (m/s^2)	
	Horizontal	Vertikal
1	0.0931	2.6299
2	0.1289	2.6296
Rata-Rata	0.1110	2.6297

4.5. Pengujian Pemanduan *Proportional Navigation*

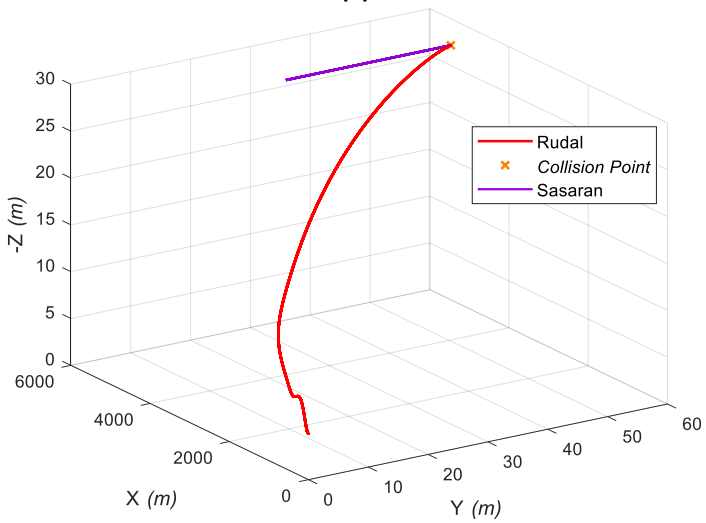
Untuk memastikan pengejaran rudal menuju sasaran berhasil, pemanduan *proportional navigation* diuji tanpa menggunakan *waypoint*. Pengujian ini diawali dengan memasukkan lokasi sasaran Bergeraknya. Setelah fase *boost*, rudal akan langsung bergerak menuju sasarannya dengan hanya menggunakan pemandu *proportional navigation*. Diasumsikan bahwa rudal selalu mengetahui kecepatan dan posisi sasarannya setiap saat. Rudal dikatakan mengenai sasarannya apabila *miss distance* rudal terhadap sasarannya kurang dari 1 meter. Beberapa skenario dijalankan untuk pengujian pemandu dimana skenario pertama rudal bergerak pada sumbu z saja, kemudian skenario kedua hanya pada sumbu x dan terakhir sasaran bergerak pada sumbu z yang disajikan dalam tabel yang dapat dilihat pada Tabel 4.11.

Tabel 4.11. Tabel Gerak Sasaran untuk Pengujian PN

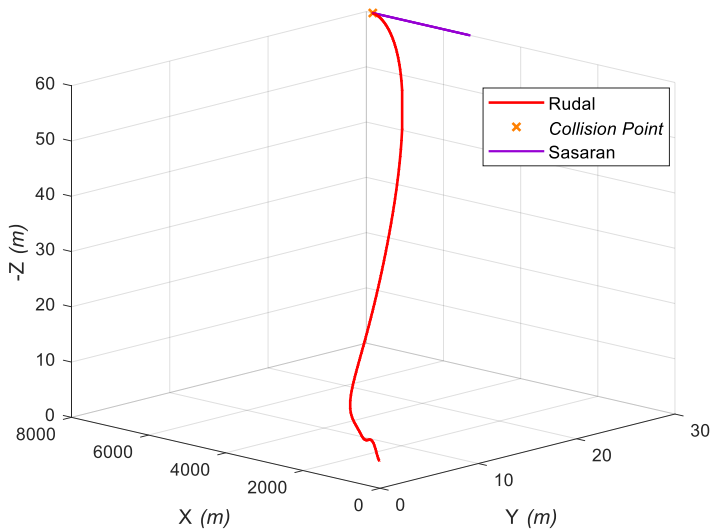
Gerak Sasaran	Skenario		
	1	2	3
Posisi (m)	[5000 50 -60]	[4500 20 -30]	[5000 30 -60]
Kecepatan (m/s)	[0 0 -2]	[0 2 0]	[100 0 0]



(a)



(b)



(c)

Gambar 4.26 Pemanduan PN Skenario (a) 1, (b) 2, (c) 3

Gambar 4.26 yang disajikan diatas dapat menunjukkan bahwa pemanduan PN dapat memberikan performansi pengejaran sasaran yang baik terlihat dengan *miss distance* yang relatif kecil sehingga bisa dikatakan rudal mampu mengenai sasarannya dengan tepat. Informasi waktu pengejaran dan *miss distance* hasil simulasi tersebut dapat dilihat pada Tabel 4.12.

Tabel 4.12. Tabel Waktu Tempuh dan *Miss distance*

Skenario	Durasi Perjalanan (s)	Jarak (m)
1	18.55	0.01652
2	16.79	0.07760
3	18.79	0.04009
Rata-Rata	18.04	0.04473

4.6. Pengujian Pengejaran Sasaran Bergerak

Pengujian pengejaran sasaran diawali dengan mengikuti lintasan *waypoint* kemudian dilanjutkan dengan mengejar sasaran. Skenario lintasan *waypoint* yang digunakan adalah lintasan *path planning* skenario kedua. Untuk skenario sasaran bergerak, terdapat dua skenario yang akan

diuji untuk mensimulasikan pengejaran sasaran bergerak ini. Masing-masing skenario menggambarkan sasaran bergerak dengan kecepatan tertentu pada dua atau tiga sumbu. Untuk lebih jelasnya, dapat dilihat pada Tabel 4.13.

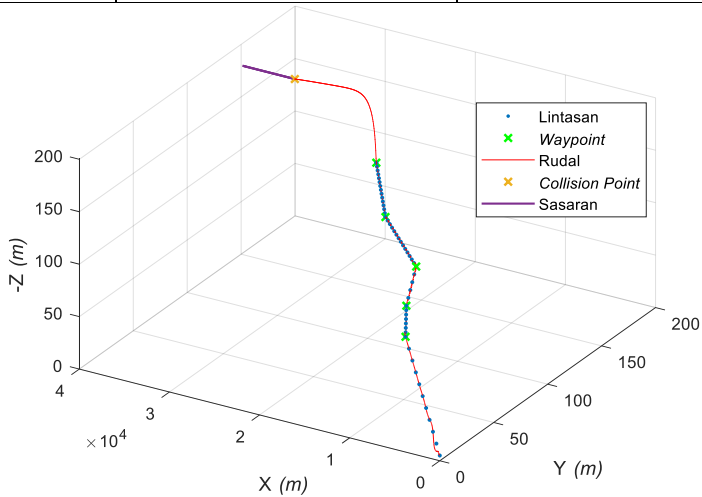
Tabel 4.13. Tabel Gerak Sasaran untuk Pengejaran Sasaran Bergerak

Skenario	Posisi Awal (m)	Kecepatan (m/s)
1	[40000 150 -180]	[-50 0 0]
2	[25000 -320 -10]	[40 5 -1.5]

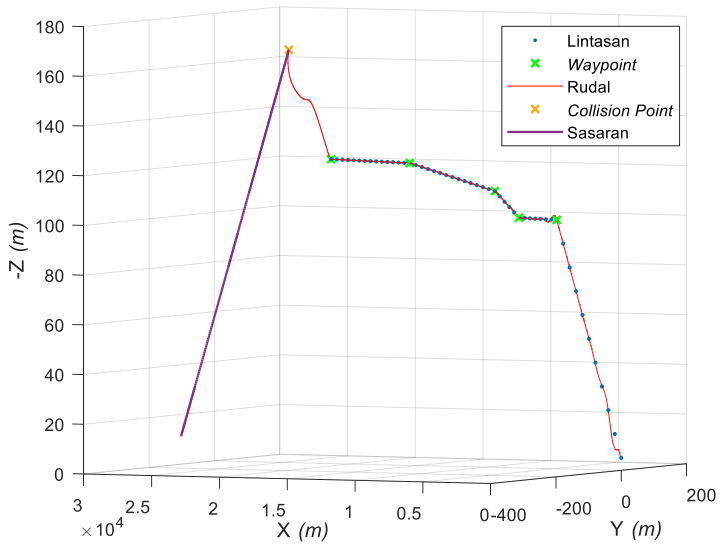
Hasil simulasi menunjukkan bahwa rudal berhasil mengenai sasarannya dengan melintasi lintasan *waypoint*. Hal yang cukup menarik yaitu pemanduan *proportional navigation* sangat sensitif terhadap perubahan LOS, sehingga terlihat grafik gerak rudal mengalami lengkungan hingga mengenai sasarannya.

Tabel 4.14. Tabel Waktu Tempuh dan *Miss distance* Pengejaran Sasaran

Skenario	Durasi Perjalanan (s)	<i>Miss Distance</i> (m)
1	119.145	0.4958
2	101.952	0.2107
Rata-Rata	110.549	0.3232

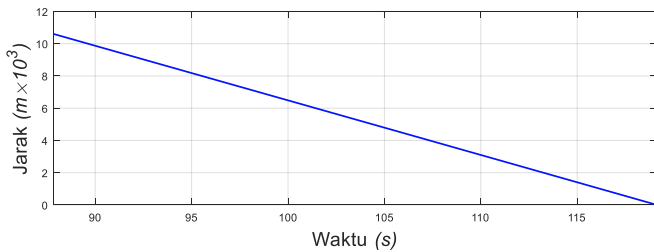


(a)

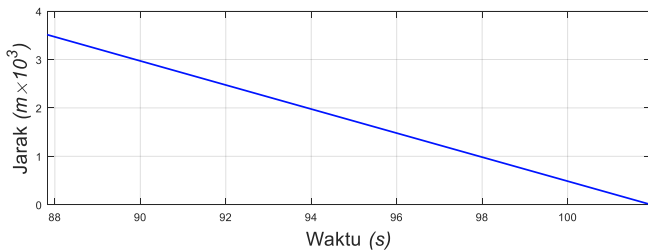


(b)

Gambar 4.27 Pengejaran Sasaran Skenario (a) 1, (b) 2



Gambar 4.28 Closing Distance Rudal dan Sasaran Skenario 1



Gambar 4.29 Closing Distance Rudal dan Sasaran Skenario 2

BAB 5

PENUTUP

Bab ini berisi kesimpulan yang dapat diambil dari penelitian yang telah dilakukan serta saran untuk pengembangan penelitian rudal lebih lanjut.

5.1. Kesimpulan

Dari simulasi rudal yang telah dijalankan, penelitian rudal ini dapat diambil kesimpulan sebagai berikut.

- a. Persamaan dinamika rotasi dan translasi rudal dapat dilinierisasi dengan menggunakan metode *nonlinear feedback linearization*. Terlihat dari hasil linierisasi, respon kecepatan rotasi dan translasi menjadi respon orde satu dengan *time constant* kecepatan rotasi 0.0333 detik dan *time constant* kecepatan translasi 0.102 detik.
- b. Pengaturan autopilot yang dirancang bekerja dengan baik yang dibuktikan dengan pengujian linierisasi dinamika rudal dan pengaturan DMPC mampu mengendalikan kecepatan translasi rudal mengikuti referensi yang diberikan.
- c. Pemanduan *nonlinear guidance logic* yang didesain bisa digunakan untuk pemanduan sasaran statis seperti lintasan *waypoint* rudal yang terlihat pada pengujian rudal berhasil mengikuti lintasan referensi hasil *path planning* dengan rata-rata *miss distance* terkecil terhadap *waypoint* yaitu 0.0872 meter dan RMSE rata-rata sebesar 0.1497 meter untuk bidang horizontal dan 0.7263 meter untuk bidang vertikal.
- d. Pemanduan *proportional navigation* bekerja untuk melakukan pengejaran sasaran dinamis atau bergerak dimana hasil pengujian menunjukkan *miss distance* yang relatif kecil dengan nilai terkecil 0.01652 meter
- e. Hasil pengujian autopilot dengan menggabungkan pemanduan menunjukkan implikasi bahwa kemampuan autopilot sangat memengaruhi perilaku rudal dalam mengikuti referensi pemanduan rudal dimana respon autopilot harus memiliki respon yang cepat.

5.2. Saran

Untuk memperbaiki dan mengembangkan penelitian rudal kedepannya, ada beberapa saran yang diberikan antara lain.

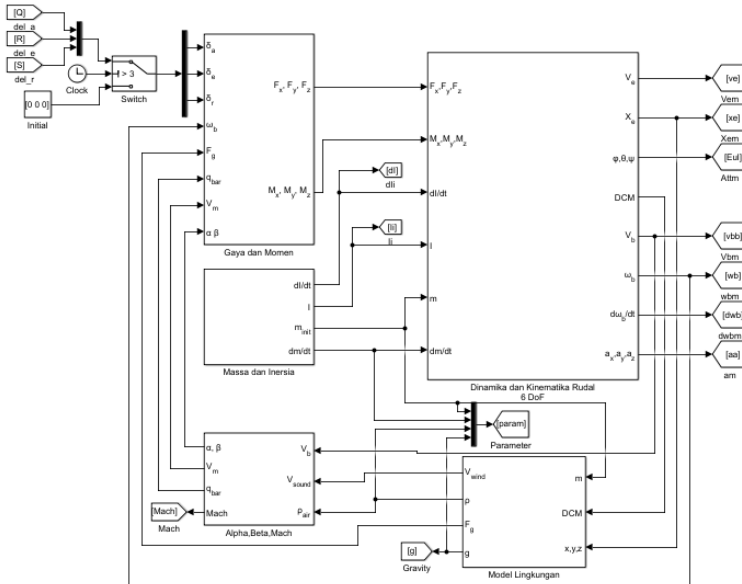
- a. Melakukan simulasi dengan menambahkan gangguan dan estimasi *state* dengan *observer*.
- b. Menggunakan koefisien aerodinamika yang sesuai dengan bentuk rudal untuk mendapatkan simulasi rudal yang lebih nyata dan akurat.
- c. Menggunakan metode pengaturan lainnya seperti pengaturan *altitude control* untuk pengaturan ketinggian rudal.
- d. Menggunakan metode pemanduan rudal berbeda khususnya untuk pengejaran sasaran yang lebih efisien.
- e. Menggunakan *path planning* yang lebih baik untuk respon rudal yang lebih *smooth* seperti menggunakan *dubins curve*.
- f. Menggunakan sistem koordinat navigasi dan transformasi koordinat *quaternion*.

DAFTAR PUSTAKA

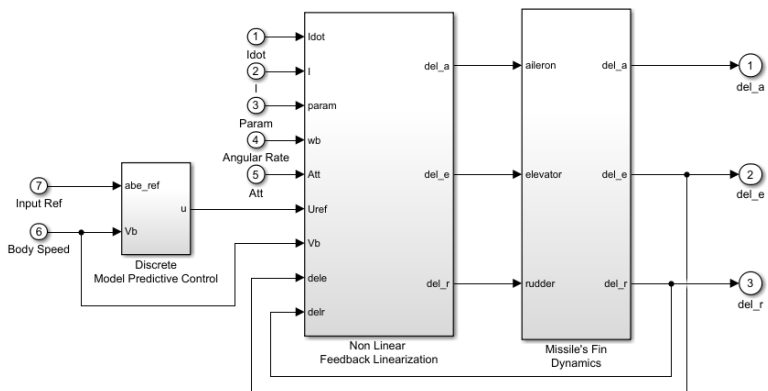
- [1] Siouris, George M. "Missile Guidance and Control Systems." *Springer Science & Business Media, 2004.*
- [2] Stevens, Brian L., Frank L. Lewis, and Eric N. Johnson. "Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems." *John Wiley & Sons, 2015.*
- [3] Özkan, Bülent. "Dynamic Modeling, Guidance, and Control of Homing Missiles." *The Graduate School of Natural and Applied Sciences of Middle East Technical University, 2005.*
- [4] Zipfel, Peter H. "Modeling and Simulation of Aerospace Vehicle Dynamics". *American Institute of Aeronautics and Astronautics, 2007.*
- [5] Slotine, Jean-Jacques E., and Weiping Li. "Applied nonlinear control". Vol. 199. No. 1. Englewood Cliffs, NJ: *Prentice hall, 1991.*
- [6] Wang, Liuping. "Model Predictive Control System Design and Implementation Using MATLAB". *Springer Science & Business Media, 2009.*
- [7] Debasish Ghose. "Guidance of Missiles", *NPTel Course, IISc, Bangalore, India, 2012*
- [8] Rana, M. S., H. R. Pota, and I. R. Petersen. "Performance of Sinusoidal Scanning with DMPC in AFM Imaging." *IEEE/ASME Transactions on Mechatronics* 20.1 (2015): 73-83.
- [9] Park, Sanghyuk, John Deyst, and Jonathan How. "A New Nonlinear Guidance Logic For Trajectory Tracking." *AIAA guidance, navigation, and control conference and exhibit. 2004.*
- [10] Stastny, Thomas. "L1 Guidance Logic Extension For Small UAVs: Handling High Winds and Small Loiter Radii." *arXiv preprint arXiv:1804.04209 (2018).*
- [11] Williams, Robert L., and Douglas A. Lawrence. "Linear State-Space Control Systems." *John Wiley & Sons, 2007.*
- [12] Ogata, Katsuhiko, and Yanjuan Yang. "Modern Control Engineering". Vol. 4. *Prentice-Hall, 2002.*
- [13] Yanushevsky, Rafael. "Modern Missile Guidance". *CRC Press, 2007.*
- [14] Padhi, Radhakant. "Linear Optimal Missile Guidance using LQR". *Diss. IISc-Bangalore.*

- [15] Hidayat, Zulkifli, Rusdhianto Effendie AK, Abdullah Alkaff, Mochammad Qomarudin, Rafif Prasetyo (2017). "Missile Guidance Permodelan dan Desain Sistem Otonom". *Laporan Teknis Laboratorium Sistem dan Sibernetika Departemen Teknik Elektro ITS*. 2017
- [16] Lin, C., & Cloutier, J. (1991). "High Performance, Adaptive, Robust Bank-to-Turn Missile Autopilot." *AIAA guidance, Navigation, Control Conference* (pp. 123–137)
- [17] Shamma, J., & Cloutier, J. (1993). Gain-Scheduled Missile Autopilot Design Using LPV Transformations. *Journal of Guidance, Control and Dynamics*, 16(2), 256–263.
- [18] AK, Rusdhianto Effendie, Muhamad Rafif Prasetyo, and Zulkifli Hidayat. "Designing an Optimal Control LQT for Controlling and Guidance of Missile." *JAREE (Journal on Advanced Research in Electrical Engineering) 2.1* (2018).

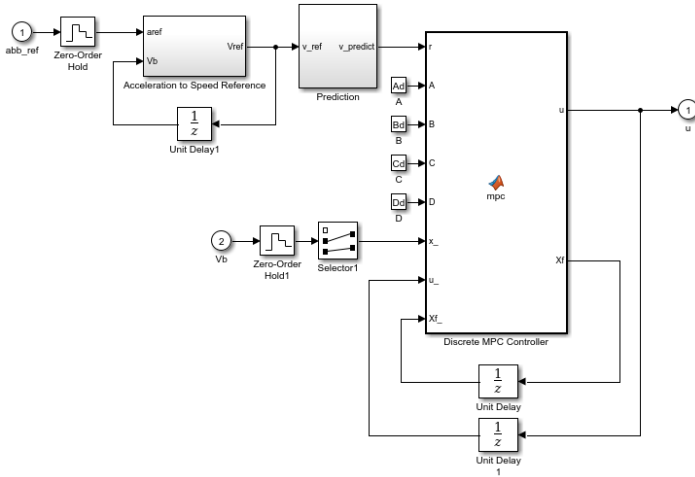
LAMPIRAN



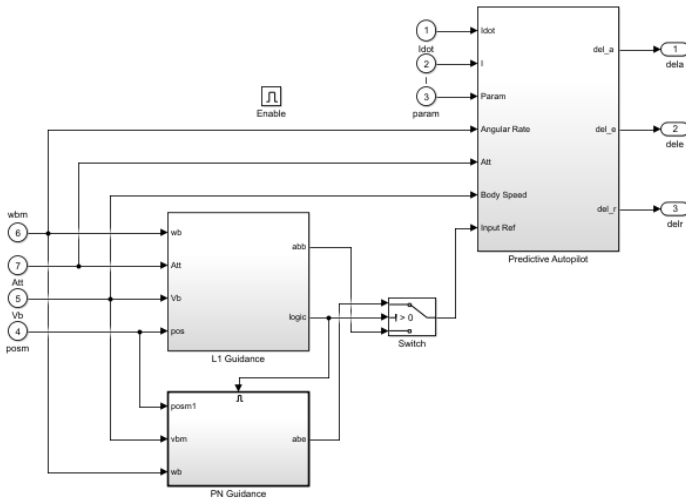
Lampiran 1. Blok Model Matematika Rudal dengan Simulink



Lampiran 2. Blok Autopilot Rudal dengan Simulink



Lampiran 3. Blok *Discrete Model Predictive Control* dengan Simulink



Lampiran 4. Blok Autopilot dan Pemanduan dengan Simulink

RIWAYAT PENULIS



Tommy Pratama lahir di Jambi pada tanggal 5 Oktober 1997 dari orang tua Ayah Zaini Djunaidi dan Ibu Yap Mui Cen. Penulis ialah anak pertama dari tiga bersaudara. Setelah penulis menempuh pendidikan menengah atas di SMA Xaverius 1 Jambi, penulis kemudian melanjutkan pendidikan tinggi di Departemen Teknik Elektro, Bidang Studi Teknik Sistem Pengaturan, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Surabaya pada tahun 2015. Selama di bangku kuliah, penulis aktif dalam kegiatan robotika dan tergabung dalam Tim ICHIRO ITS pada tahun 2016 hingga tahun 2019. Sejalan dengan keaktifan di tim robotika, penulis juga aktif sebagai asisten Laboratorium Teknik Sistem dan Sibernetika. Penulis sangat tertarik dalam riset dunia robotika terutama dalam algoritma cerdas dan teknik kontrol. Penulis dapat dihubungi di surat elektronik tommypra97@gmail.com.