

TUGAS AKHIR - EE184801

**TRACKING UNTUK MOBIL OTONOM DENGAN
MEMPERHATIKAN HALANGAN MENGGUNAKAN
*MODEL PREDICTIVE CONTROL***

YASMINA ALYA

NRP 07111840000107

Dosen Pembimbing

Ir. Ali Fatoni, MT

NIP 196206031989031002

Program Studi Sarjana Teknik Elektro

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

Tahun 2022



TUGAS AKHIR - EE184801

**TRACKING UNTUK MOBIL OTONOM DENGAN
MEMPERHATIKAN HALANGAN MENGGUNAKAN
MODEL PREDICTIVE CONTROL**

YASMINA ALYA

NRP 07111840000107

Dosen Pembimbing

Ir. Ali Fatoni, MT

NIP 196206031989031002

Program Studi Sarjana Teknik Elektro

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

Tahun 2022



FINAL PROJECT - EE184801

**TRACKING OF AUTONOMOUS CAR WITH ATTENTION
TO OBSTACLE USING *MODEL PREDICTIVE CONTROL***

Yasmina Alya

NRP 07111840000107

Advisor

Ir. Ali Fatoni, MT

NIP 196206031989031002

Undergraduate Study Program of Electrical Engineering

Department of Electrical Engineering

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

Year 2022

LEMBAR PENGESAHAN

TRACKING UNTUK MOBIL OTONOM DENGAN MEMPERHATIKAN HALANGAN MENGGUNAKAN MODEL PREDICTIVE CONTROLLER

TUGAS AKHIR

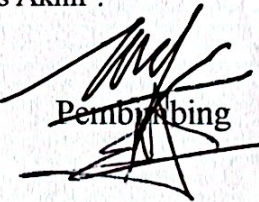
Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Teknik pada
Program Studi S-1 Teknik Elektro
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh : YASMINA ALYA

NRP. 07111840000107

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Ali Fatoni, MT.


Pembimbing

2. Eka Iskandar, ST., MT.

Ko-pembimbing

3. Yusuf Bilfaqih, ST., MT.


Penguji

4. Zulkifli Hidayat, ST., M.Sc.


Penguji

5. Mochammad Sahal, ST., M.Sc.


Penguji

SURABAYA

Juni, 2022

(Halaman ini sengaja dikosongkan)

APPROVAL SHEET

TRACKING CONTROL OF AUTONOMOUS CAR WITH ATTENTION TO OBSTACLE USING *MODEL PREDICTIVE CONTROL*

FINAL PROJECT

Submitted to fulfill one of the requirements
For obtaining a bachelor's degree at
Undergraduate Study Program of Electrical Engineering
Department of Electrical Engineering
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember

By : **YASMINA ALYA**
NRP. 07111840000107

Approved by Final Project Examiner Team :

1. Ir. Ali Fatoni, MT.



Advisor

2. Eka Iskandar, ST., MT.

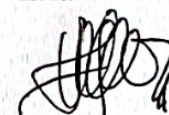
Co-Advisor



3. Yusuf Bilfaqih, ST., MT.

Examiner

4. Zulkifli Hidayat, ST., M.Sc.



Examiner

5. Mochammad Sahal, ST., M.Sc.



Examiner

SURABAYA

June, 2022

(Halaman ini sengaja dikosongkan)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Yasmina Alya/ 07111840000107
Departemen : Teknik Elektro
Dosen Pembimbing / NIP : Ir. Ali Fatoni, MT/ 196206031989031002

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Tracking untuk Mobil Otonom dengan Memperhatikan Halangan Menggunakan *Model Predictive Control*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 10 Juni 2022

Mengetahui
Dosen Pembimbing



Ir. Ali Fatoni, MT
196206031989031002.

Mahasiswa



Yasmina Alya
07111840000107.

(Halaman ini sengaja dikosongkan)

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Yasmina Alya / 07111840000107
Department : Electrical Engineering
Advisor / NIP : Ir. Ali Fatoni, MT/ 196206031989031002

hereby declare that the Final Project with the title of “Tracking Control of Autonomous Car with Attention to Obstacle Using *Model Predictive Control*” is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, 10 June 2022

Acknowledged
Advisor



Ir. Ali Fatoni, MT
196206031989031002.

Student



Yasmina Alya
07111840000107.

(Halaman ini sengaja dikosongkan)

ABSTRAK

TRACKING UNTUK MOBIL OTONOM DENGAN MEMPERHATIKAN HALANGAN MENGUNAKAN MODEL PREDICTIVE CONTROL

Nama Mahasiswa / NRP : Yasmina Alya / 0711184000107
Departemen : Teknik Elektro FTEIC - ITS
Dosen Pembimbing : Ir. Ali Fatoni, MT

Abstrak

Pada penelitian terdahulu mengenai MPC untuk path tracking dan obstacle avoidance, menunjukkan MPC menghasilkan gerakan berosilasi pada path tracking dan terkadang MPC tidak mampu menghindari halangan dengan manuver yang efektif. Penelitian dilakukan dengan memvariasikan bobot cost function dan hasil terbaik diperoleh ketika bobot eror lebih besar dibandingkan dengan bobot input. Penelitian tersebut dibatasi dengan kecepatan mobil konstan. Maka pada penelitian ini penulis bertujuan untuk menggunakan MPC untuk trajectory tracking dan obstacle avoidance. Pada penelitian ini, digunakan referensi yang berubah terhadap waktu (trajectory tracking). Sehingga MPC juga akan mengatur kecepatan mobil. Kombinasi parameter MPC divariasikan untuk mencari desain MPC dengan performa baik. Penelitian ini menggunakan Linear Time Variant MPC (LTV MPC). Pada sistem obstacle avoidance, deteksi halangan dilakukan dengan mengukur jarak antara mobil dengan halangan. Selama halangan terdeteksi, dilakukan perhitungan constraint posisi Y baru. Pengujian trajectory tracking dilakukan dengan 3 referensi track sinusoidal dan 1 track pergantian lajur. Pengujian obstacle avoidance menggunakan 1 dan 2 halangan. Evaluasi dilakukan berdasarkan nilai RMSE posisi, cost function, dan jarak terdekat dengan halangan saat manuver penghindaran. Hasil menunjukkan MPC berhasil menjalankan fungsi trajectory tracking dengan keterlambatan rata-rata 0.4 s, sehingga lebih cocok digunakan untuk kecepatan rendah. MPC tidak selalu berhasil melakukan obstacle avoidance karena terdapat variasi yang tidak memenuhi zona aman. Dapat diimplikasikan bahwa perumusan halangan sebagai constraint posisi Y kurang cocok digunakan untuk menjalankan fungsi obstacle avoidance pada MPC. Karena pada obstacle avoidance dihasilkan Gerakan osilasi yang belum cocok untuk implementasi dunia nyata.

Kata kunci: *Mobil otonom, Model Predictive Control, Path Tracking, Obstacle Avoidance.*

(Halaman ini sengaja dikosongkan)

ABSTRACT

TRACKING CONTROL OF AUTONOMOUS CAR WITH ATTENTION TO OBSTACLE USING *MODEL PREDICTIVE CONTROL*

Student Name / NRP : Yasmina Alya / 07111840000107
Department : Electrical Engineering ELECTICS - ITS
Advisor : Ir. Ali Fatoni, MT

Abstract

Previous research about MPC in for autonomous car shows that MPC generates an oscillating movement when doing path tracking and can't avoid obstacle effectively. The research is done by varying cost function weights where the best performance is produced when the error weight is greater than the input weight. Limitations of research includes the usage of constant velocity. Therefore, in this research the writer aims to use MPC for trajectory tracking and obstacle avoidance. So MPC will also controls the car's velocity. Time varying reference is used to represent trajectory tracking problem. Combinations of MPC parameters are varied to search for the best performing design. Research is done using Linear Time Variant MPC (LTV MPC). In obstacle avoidance system, obstacle detection is done by calculating the distance between the car and obstacle. While an obstacle is detected, the system recalculates a new lateral constraint to ensure the car move within a predefined safe zone. Trajectory tracking tests are done using 3 sinusoidal tracks and a lane change maneuver track. Obstacle avoidance tests are done using 1 and 2 obstacles. Evaluation is based on RMSE of position, cost function, and the closest distance between car and obstacle during avoidance maneuver. Results shows MPC can be used for trajectory tracking with average delay of 0.4 s. So it is better for a lower velocity. MPC can't always avoid obstacle, shown by one of the variation that can't fulfill the safe zone. It can be implicated that defining obstacle as lateral constraint is not suitable for obstacle avoidance function. The oscillating car movement shows that the system is not yet suitable for real life implementation.

Keywords: *Autonomous car, Model Predictive Control, Path Tracking, Obstacle Avoidance.*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Allah S.W.T. yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat membuat dan menyelesaikan tugas akhir ini dengan baik dan tepat waktu. Kegiatan tugas akhir ini merupakan bagian dari penyelesaian studi S-1 Departemen Teknik Elektro Institut Teknologi Sepuluh Nopember, dan laporan tugas akhir ini disusun untuk melengkapi hasil capaian dari tugas akhir yang telah dilaksanakan. Dalam pembuatan tugas akhir ini, penulis telah dibantu dan didukung oleh banyak pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Orang tua serta keluarga yang telah memberikan dukungan dan do'a kepada penulis.
2. Bapak Ir. Ali Fatoni, M.T. dan bapak Eka Iskandar, S.T, M.T. selaku dosen pembimbing yang telah memberi arahan dan bimbingan pada penulis selama proses pengerjaan tugas akhir ini.
3. Dosen dan tenaga pendidik Departemen Teknik Elektro ITS, khususnya bidang studi Teknik Sistem Pengaturan
4. Afina, Kayla, dan Athar sebagai anggota grup itu yang selalu menemani penulis dan memberikan dukungan dimanapun berada.
5. Enaw, Matrix, dan Dheo sebagai teman terdekat penulis selama masa perkuliahan di ITS.
6. Teman-teman penulis yang telah memberikan dukungan teknis maupun non teknik selama proses pengerjaan.

Penulis menyadari, bahwa masih terdapat kekurangan dari laporan tugas akhir ini dikarenakan keterbatasan dan kurangnya pengetahuan serta pengalaman penulis. Penulis berharap tugas akhir ini dapat bermanfaat bagi pembaca. Saran dan kritik juga penulis harapkan untuk pengembangan tugas akhir.

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN	i
APPROVAL SHEET	ii
PERNYATAAN ORISINALITAS	iv
STATEMENT OF ORIGINALITY	vi
ABSTRAK	viii
ABSTRACT	x
KATA PENGANTAR	xii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xviii
DAFTAR SIMBOL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	2
BAB 2 TINJAUAN PUSTAKA	3
2.1 Hasil Penelitian Terdahulu	3
2.2 Dasar Teori	3
2.2.1 Mobil Otonom	3
2.2.2 Trajectory Tracking	3
2.2.3 Model Mobil	4
2.2.4 Model Predictive Controller	5
2.2.5 Quadratic Programming	7
2.2.6 Algoritma Hildreth	7
2.2.7 Ekspansi Taylor Series	8
BAB 3 METODOLOGI	9
3.1 Metode yang Digunakan	9
3.2 Bahan dan Peralatan yang Digunakan	9
3.3 Urutan Pelaksanaan Penelitian	10
3.3.1 Pemodelan Sistem	10

3.3.2	Perancangan Kontroler	13
3.3.3	Perumusan Referensi	15
BAB 4	HASIL DAN PEMBAHASAN	19
4.1	Trajectory Tracking	19
4.1.1	Track Sinusoidal dengan Kecepatan 10 m/s	19
4.1.2	Track Sinusoidal dengan Kecepatan 20 m/s	21
4.1.3	Track Pergantian Lajur	22
4.1.4	Track Sinusoidal dengan Sudut Steering Melebihi Constraint	23
4.2	Obstacle Avoidance	26
4.2.1	1 Halangan dengan Kecepatan 10 m/s	26
4.2.2	1 Halangan dengan Kecepatan 20 m/s	28
4.2.3	2 Halangan dengan Kecepatan 10 m/s	29
BAB 5	KESIMPULAN DAN SARAN	33
5.1	Kesimpulan	33
5.2	Saran	34
	DAFTAR PUSTAKA	35
	LAMPIRAN	37
	BIODATA PENULIS	48

DAFTAR GAMBAR

Gambar 2.1 Bicycle Model	4
Gambar 2.2 Struktur Algoritma Model Predictive Controller	6
Gambar 2.3 Diagram LTV MPC.....	6
Gambar 3.1 Diagram Konfigurasi Simulasi.....	10
Gambar 3.2 Jalur Simulasi	11
Gambar 3.3 Zona Aman.....	12
Gambar 3.4 Constraint Posisi Y untuk Obstacle Avoidance	13
Gambar 3.5 Trayektori Referensi Sinusoidal untuk Simulasi Trajectory Tracking	16
Gambar 3.6 Trayektori Referensi Sinusoidal 2 untuk Simulasi Trajectory Tracking.....	16
Gambar 3.7 Trayektori Referensi Pergantian lajur	17
Gambar 3.8 Trayektori Referensi untuk Simulasi Obstacle Avoidance	17
Gambar 4.1 Trajectory Tracking 1 dengan kecepatan 10m/s	19
Gambar 4.2 Trajectory Tracking Pergantian Lajur.....	22
Gambar 4.3 Trajectory Tracking 2 dengan Kecepatan 10m/s	24
Gambar 4.4 Error Posisi Trajectory Tracking 2 dengan Kecepatan 10m/s.....	24
Gambar 4.5 Obstacle Avoidance dengan Kecepatan 10 m/s	26
Gambar 4.6 Obstacle Avoidance dengan kecepatan 20 m/s	28
Gambar 4.7 Obstacle Avoidance dengan 2 Halangan.....	29

DAFTAR TABEL

Tabel 3.1 Constraint Input.....	14
Tabel 4.1 Parameter Simulasi.....	19
Tabel 4.2 Trajectory Tracking 1 dengan Kecepatan 10m/s.....	20
Tabel 4.3 Trajectory Tracking dengan Kecepatan 20m/s.....	21
Tabel 4.4 Trajectory Tracking Pergantian Lajur	22
Tabel 4.5 Trajectory Tracking 2 dengan Kecepatan 10m/s.....	24
Tabel 4.6 Obstacle Avoidance dengan Kecepatan 10 m/s	26
Tabel 4.7 Obstacle Avoidance dengan Kecepatan 20 m/s	28
Tabel 4.8 Obstacle Avoidance dengan 2 Halangan.....	29

DAFTAR SIMBOL

ICR	: Pusat Rotasi
$\{O\}$: sumbu koordinat global
$\{B\}$: sumbu koordinat kendaraan
x_B & y_B	: sumbu x dan y koordinat kendaraan
R_B & R_F	: radius putar roda belakang dan roda depan
γ	: <i>steering angle</i>
θ	: <i>heading angle</i>
ω	: steering rate
L	: Panjang kendaraan (<i>wheel base</i>)
V	: kecepatan mobil
X	: vektor state
Y	: vektor output
U	: vektor input
A	: matriks state
B	: matriks input
C	: matriks output
x_r	: referensi posisi X
y_r	: referensi posisi Y
θ_r	: referensi heading angle
γ_r	: referensi steering angle
x_o	: posisi X obstacle
y_o	: posisi Y obstacle
J	: cost function
Q	: bobot input cost function
R	: bobot eror cost function
wu	: matriks bobot input
M	: matriks input constraint gabungan
N	: vektor constraint gabungan
λ	: lagrange multiplier
\vec{Y}	: prediksi output
\vec{U}	: prediksi input
F	: matriks output prediksi
Φ	: matriks input prediksi

(Halaman ini sengaja dikosongkan)

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, pengembangan kontrol untuk kendaraan otonom semakin pesat. Salah satu kendaraan otonom yang populer dikembangkan adalah mobil otonom. Mobil otonom mulai dikembangkan pada tahun 1926 dengan terciptanya “Linriccan Wonder”. Linriccan Wonder merupakan mobil pertama di dunia yang dapat dikendalikan melalui radio. Setelah itu, kedatangan mercedez-benz robotic van dengan vision guidance pada tahun 1980 meningkatkan perkembangan mobil otonom lebih pesat (Bimbraw, 2015). Pada abad ke-21 ini terdapat beberapa mobil yang memiliki fitur self driving. Beberapa fitur self driving adalah cruise control dan active lane assist. Cruise control adalah fitur rem dan akselerasi otomatis, sedangkan active lane assist adalah fitur steering otomatis (Charan & Srour, 2013). Maka dapat dikatakan bahwa fitur self driving hanya berperan sebagai bantuan mengemudi dan tidak mengizinkan mobil untuk berjalan tanpa pengemudi. Untuk merealisasikan mobil otonom yang dapat dijalankan tanpa pengemudi dibutuhkan sistem otonom yang handal.

Menurut SAE J306, tingkat otomasi mobil dapat dibagi menjadi 6 level dengan level 5 sebagai level tertinggi. Pada level 5, mobil dapat berjalan secara otomatis dalam kondisi apapun tanpa ada campur tangan pengemudi. Dalam kata lain, *Driver Assist System* (DAS) mengatasi seluruh *Dynamic Driving Task* (DDT) saat berkemudi. DDT dapat dibagi menjadi 2, yaitu pengaturan gerakan lateral dan longitudinal mobil serta deteksi dan pengaturan respon terhadap objek dan kejadian lain (Babak et al., 2017). Dalam pengembangan DAS, banyak penelitian telah dilakukan.

Pengaturan gerakan lateral dan longitudinal mobil berguna untuk menjaga mobil untuk berjalan pada jalur yang ditetapkan atau biasa disebut *Tracking*. Sementara deteksi dan pengaturan respon terhadap objek dan kejadian lain berguna untuk menghindari mobil bertabrakan. Hal ini biasa dikenal dengan *Obstacle Avoidance*. Pengaturan akan lebih baik dilakukan oleh 1 sistem utuh yang dapat mengatasi steering untuk *Tracking* serta *Obstacle Avoidance*. Namun ditemukan permasalahan untuk mewujudkan sistem ini. Lingkungan yang dijalani mobil selalu dinamis dan berubah terhadap waktu. Selain itu mobil otonom sendiri juga memiliki batasan-batasan mekanik tertentu yang perlu dijaga. Untuk itu dibutuhkan kontroler yang dapat memprediksi masa depan serta dapat menjaga mobil dalam batasannya.

Terhadap permasalahan ini, penulis mengusulkan metode pengaturan mobil otonom dengan menggunakan Model Predictive Controller (MPC). MPC diketahui mampu mengatasi *Path Tracking* serta *Obstacle Avoidance* (Hatem, 2018). Pengujian dilakukan dengan jalur lurus dengan 3 halangan statis yang diletakan pada jalur. Pada penelitian ini terbukti MPC dapat mengatur gerakan mobil menuju titik tujuan tanpa menabrak halangan. Namun, penelitian tersebut dibatasi dengan penggunaan kecepatan konstan dan pengujian dilakukan dengan memvariasikan bobot cost function. Sehingga pada tugas akhir ini akan dilakukan penelitian mengenai penggunaan MPC dalam mengatur posisi mobil untuk fungsi trajectory tracking dan obstacle avoidance.

1.2 Rumusan Masalah

Rumusan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana memodelkan mobil otonom?

2. Bagaimana perumusan permasalahan tracking?
3. Bagaimana desain *Model Predictive Controller* (MPC) untuk mengatur posisi mobil otonom?

1.3 Batasan Masalah

Pembatasan masalah digunakan untuk menghindari adanya penyimpangan maupun pelebaran pokok masalah dari tugas akhir ini. Berikut adalah beberapa batasan masalah yang digunakan dalam tugas akhir ini:

1. Fokus penelitian berada pada tracking dan obstacle avoidance, sehingga tidak mengimplementasikan sistem path planning.
2. Pengambilan data dilakukan menggunakan simulasi melalui aplikasi Matlab.
3. Penelitian dilakukan dengan mengabaikan gangguan eksternal.
4. Penelitian dilakukan dengan mengabaikan karakteristik dinamis dari mobil. Sehingga hanya digunakan model kinematik.
5. Halangan yang digunakan pada tugas akhir ini berupa halangan dinamis simpel. Yaitu halangan yang bersifat sementara seperti penutupan jalan atau mobil yang parkir pada trayektori mobil. Halangan tidak mencakup objek bergerak seperti mobil ataupun motor yang bergerak dengan kecepatan tertentu.
6. Penelitian dilakukan dengan menggunakan model deteksi halangan simpel. Yaitu dengan perbandingan jarak antara posisi halangan yang telah diketahui dengan posisi mobil instan.

1.4 Tujuan

Adapun tujuan dari Tugas Akhir adalah untuk mendesain *Model Predictive Controller* untuk mengatur gerakan mobil otonom untuk tracking dengan memperhatikan halangan.

1.5 Manfaat

Adapun manfaat dari tugas akhir ini adalah sebagai berikut:

1. Meningkatkan pengetahuan dan pemahaman mengenai trajectory tracking dan obstacle avoidance.
2. Sebagai referensi penelitian lain yang lebih kompleks mengenai mobil otonom.

BAB 2 TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Penelitian mengenai penggunaan *Model Predictive Controller* (MPC) telah banyak dilakukan. MPC banyak dimanfaatkan untuk mengatur sistem multi input-multi output (MIMO) serta sistem non-linear. Selain itu, salah satu keunggulan MPC adalah kemampuannya dalam memperhitungkan batasan fisik sistem. Salah satu sistem yang dapat diatur menggunakan MPC adalah mobil otonom. Terdapat berbagai macam aspek yang perlu diatur untuk menjalankan mobil otonom.

Salah satu aspek yang telah diteliti adalah pengaturan gerakan lateral mobil otonom untuk menghindari halangan statis yang dilakukan oleh Hatem, A (2018). Penelitian ini dilakukan dengan tujuan membangun kontroler linear dengan fungsi *path tracking* serta *obstacle avoidance* untuk mobil otonom. Pada penelitian ini mobil diasumsikan berjalan dengan kecepatan konstan. Halangan yang digunakan berupa halangan statis dan jalur yang digunakan berupa garis lurus. Pengujian fungsi *path tracking* dilakukan dengan memvariasikan bobot input kontrol. Hasil yang diperoleh menunjukkan bahwa pemilihan bobot yang tepat dapat menghasilkan gerakan mobil dengan osilasi kecil, diperoleh dengan bobot eror lebih besar dibandingkan bobot input. Selain itu kontroler juga dapat mengatasi gangguan dalam bentuk eror posisi inisial. Pengujian *obstacle avoidance* dilakukan dengan meletakkan 3 halangan dengan bentuk persegi Panjang di tengah jalur. Hasil pengujian menunjukkan bahwa mobil dapat menghindari halangan, meskipun dengan jarak lebih dari yang diperlukan. (Hatem, 2018)

Ada pula penelitian yang menggabungkan trayektori planning dengan control tracking menggunakan MPC. Pada penelitian ini referensi jalur diperoleh menggunakan algoritma A*. Kemudian MPC digunakan sebagai kontroler trajectory tracking. Tujuan penggunaan MPC adalah untuk meningkatkan akurasi tracking serta mempertahankan gerakan mobil yang mulus. Tujuan ini dituliskan dalam bentuk cost function dan constraint eror pada MPC. (Li et al., 2019)

2.2 Dasar Teori

2.2.1 Mobil Otonom

Mobil otonom merupakan mobil yang dapat beroperasi tanpa diatur oleh manusia dan tidak membutuhkan campur tangan dari manusia. Menurut Campbell et al. mobil otonom modern dapat mendeteksi lingkungan di sekitarnya, mengklasifikasi objek-objek yang dideteksi, serta dapat mengidentifikasi jalur navigasi yang tepat yang sesuai aturan lalu lintas berdasarkan informasi dari sensor. Untuk memperoleh navigasi mobil otonom yang baik, diperlukan berbagai macam teknologi dari berbagai macam bidang ilmu. Beberapa bidang antara lain adalah ilmu komputer, Teknik mesin, Teknik elektronika dan elektro, Teknik sistem pengaturan, dll. Hal ini penting karena mobil harus dapat memberikan respons yang tepat dalam menghadapi situasi yang tidak terduga pada dunia nyata. Respons yang diberikan mobil perlu dipastikan agar tidak memberikan *backlash* yang dapat merusak mobil itu sendiri (Bimbraw, 2015).

2.2.2 Trajectory Tracking

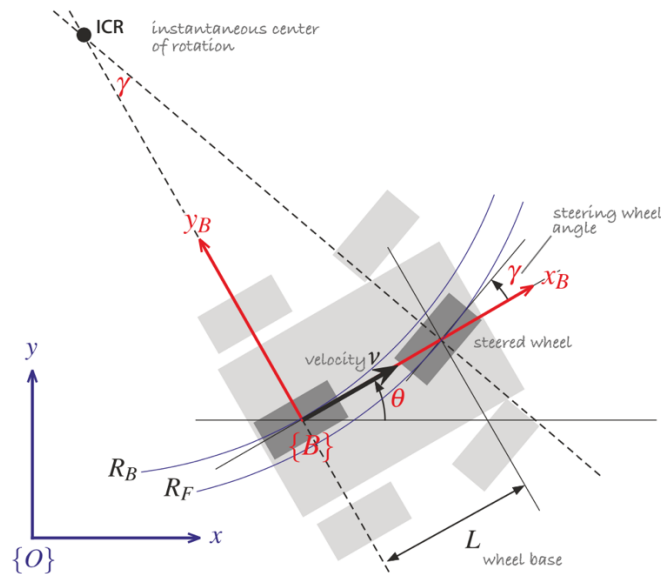
Permasalahan trajectory tracking adalah untuk mengatur agar sistem mengikuti referensi yang berubah terhadap waktu. Berbeda dengan path tracking dimana referensi tidak berubah

terhadap waktu. Artinya, pada permasalahan trajectory tracking sebuah sistem harus mencapai titik referensi tertentu pada waktu yang juga sudah ditentukan.(Faulwasser, 2012)

2.2.3 Model Mobil

2.2.3.1 Model kinematika mobil

Salah satu model yang sering digunakan untuk menggambarkan gerakan suatu kendaraan adalah bicycle model. Sesuai gambar (2.1), kendaraan memiliki 2 roda yaitu roda depan dan belakang. Roda belakang menempel pada bodi kendaraan dan roda depan dapat berputar terhadap sumbu vertikal kendaraan untuk mengarahkan kendaraan.



Gambar 2.1 Bicycle Model

Keterangan:

ICR : pusat rotasi

$\{O\}$: sumbu koordinat global

$\{B\}$: sumbu koordinat kendaraan

x_B & y_B : sumbu x dan y koordinat kendaraan

R_B & R_F : radius putar roda belakang dan roda depan

γ : *steering angle*

θ : *heading angle*

L : Panjang kendaraan (*wheel base*)

(Corke, 2017)

Kecepatan sudut (angular velocity) kendaraan dirumuskan sebagai

$$\dot{\theta} = \frac{v}{R_B} \quad (2.1)$$

kecepatan kendaraan dalam koordinat global adalah $(v \cos \theta, v \sin \theta)$. Jika digabungkan dengan persamaan (2.1), maka persamaan gerakan kendaraan dapat dirumuskan menjadi

$$\begin{aligned} \dot{x} &= V \cos(\theta) \\ \dot{y} &= V \sin(\theta) \\ \dot{\theta} &= \frac{V}{L} \tan(\gamma) \end{aligned} \quad (2.2)$$

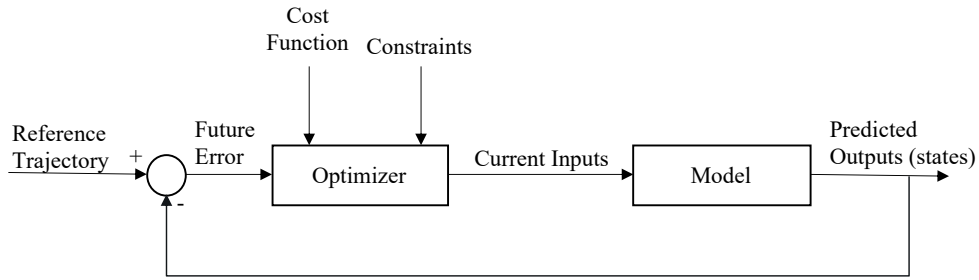
Model di atas disebut dengan model kinematika kendaraan. Model tersebut tidak menjelaskan mengenai gaya ataupun torsi yang menggerakkan kendaraan dengan kecepatan v . Perubahan dari $\dot{\theta}$ disebut sebagai turn rate, heading rate, atau yaw rate. Nilai dari yaw rate dapat diukur menggunakan gyroscope, namun juga dapat diperoleh dari informasi angular velocity ($\dot{\theta}$) (Corke, 2017).

2.2.4 Model Predictive Controller

Model Predictive Control (MPC) merupakan algoritme kontrol berbasis model dan aplikasi optimasi performa secara online. Dewasa ini pengaplikasian MPC untuk menyelesaikan masalah optimasi semakin luas. Bidang pengaplikasiannya meliputi Kesehatan, energi, aeronautika dan dirgantara, dan lain-lain. MPC memiliki 3 elemen utama yaitu model prediksi, fungsi tujuan (cost function) atau kendala, dan metode untuk mendapatkan sinyal kontrol (Santoso et al., 2021).

MPC memiliki kelebihan dan kekurangan. Kelebihan utama dari MPC adalah kemampuannya untuk mengatasi sistem non-linier dan sistem dengan Multi-Input-Multi-Output (MIMO). MPC dapat mengatasi sistem yang memiliki batasan fisik dengan menyertakannya ke dalam perumusan masalah sistem dan algoritma optimasi. Selain itu, MPC juga mudah untuk dikonfigurasi ulang dan dapat beradaptasi terhadap perubahan pada sistem yang dikontrol. Adapun kekurangan dari MPC adalah waktu komputasinya yang lama. Hal ini disebabkan oleh komputasi algoritma optimasi dilakukan pada cakrawala waktu (horizon time) yang ditetapkan. Lalu, beban komputasi pada komputer untuk menjalankan MPC juga cukup besar (Halim, 2021). Oleh karena itu, MPC sering kali diaplikasikan untuk sistem dengan dinamika yang lambat seperti plant kontrol proses (Hatem, 2018).

Konsep dari MPC adalah algoritma akan menghitung nilai yang akan digunakan sebagai masukan (input) sistem sebagai solusi dari permasalahan optimasi. MPC memprediksi keluaran sistem selanjutnya dalam rentang cakrawala waktu yang sudah ditentukan serta menggunakan masukan dan keluaran sebelumnya dari sistem. Setelah itu, perhitungan sinyal kontrol yang berada pada rentang cakrawala waktu dan optimasi dilakukan (Halim, 2021). MPC akan mengalkulasi sinyal kontrol optimal dari setiap input dengan memperhatikan batasan atau constraint. Respon sinyal selanjutnya akan dikomputasi sebagai permasalahan yang akan diselesaikan oleh horizon open loop yang ditentukan. pada elemen pertama dari respons sinyal kontrol optimal yang akan digunakan sebagai masukan pada sinyal input sistem selanjutnya, sementara itu elemen lainnya tidak akan digunakan. Gambar di bawah menunjukkan struktur dasar dari algoritma MPC (Santoso et al., 2021).



Gambar 2.2 Struktur Algoritma Model Predictive Controller

Fungsi tujuan (cost function) algoritma MPC dapat ditunjukkan pada persamaan (2.3) berikut

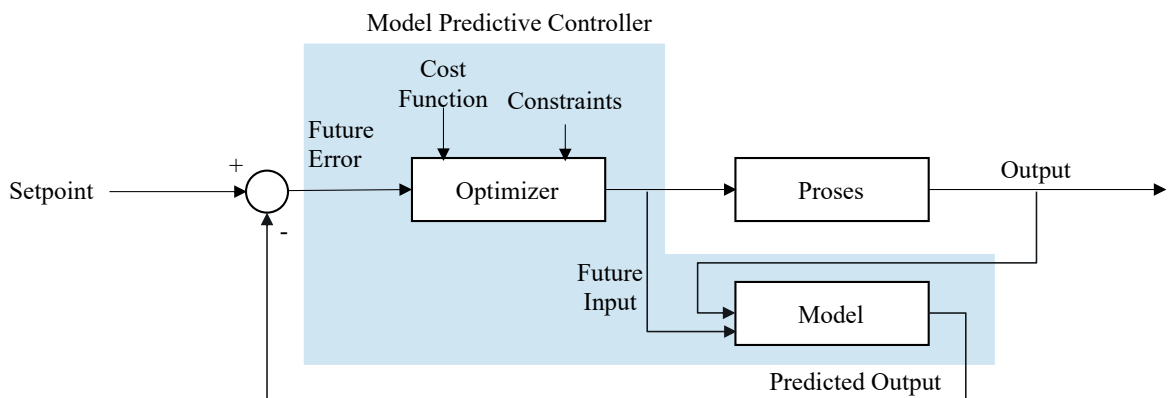
$$J_k = \sum_{i=0}^{Np} \mathbf{x}_{k+i+1}^T Q \mathbf{x}_{k+i+1} + \mathbf{u}_{k+i}^T R \mathbf{u}_{k+i} \quad (2.3)$$

dimana suku pertama dari cost function adalah untuk meminimalkan eror dan suku kedua adalah untuk meminimalkan energi keluaran input. Sehingga Q dan R adalah bobot untuk masing-masing eror dan input (Rossiter, 2018).

2.2.4.1 Adaptive MPC

Adaptive MPC merupakan salah satu metode MPC dimana model plant internal yang digunakan oleh kontroler tidak konstan, melainkan berubah terhadap waktu. Metode ini biasa digunakan untuk proses dengan dinamika yang bervariasi dengan kondisi operasi yang berbeda. Ketika kondisi operasi berubah dan dinamika berubah maka diperlukan model plant internal yang baru. Dengan begitu MPC dapat menghasilkan prediksi yang lebih akurat (Mathworks, 2018).

Salah satu jenis adaptive MPC adalah Linear time varying MPC (LTV MPC). LTV MPC digunakan untuk mengatasi sifat non-linear sistem. Pada metode ini, linearisasi model dilakukan tiap time step. Sehingga dihasilkan model linear baru yang lebih akurat sesuai operating point plant yang diatur (Abel & Katriniok, 2011).



Gambar 2.3 Diagram LTV MPC

(Bagyaveereswaran et al., 2016)

2.2.5 Quadratic Programming

Quadratic Programming atau Pemrograman kuadrat adalah algoritme yang dikembangkan untuk mencari solusi dari cost function persamaan kuadrat. Persamaan kuadrat dari cost function didefinisikan sebagai berikut

$$J = \frac{1}{2}x^T E x + x^T F \quad (2.4)$$

Dimana x merupakan variabel yang perlu dicari untuk meminimalkan cost function di atas. Adapun constraint didefinisikan sebagai

$$Mx \leq \gamma \quad (2.5)$$

untuk mendapatkan nilai optimum maka turunan parsial J terhadap x harus bernilai nol. Sehingga x optimum dapat dirumuskan sebagai

$$x = -E^{-1}(M^T \lambda) \quad (2.6)$$

λ disebut sebagai lagrange multiplier, yaitu variabel yang divariasikan untuk mendapatkan nilai x optimum (Wang, 2009).

2.2.6 Algoritma Hildreth

Nilai λ dapat diperhitungkan menggunakan beberapa macam algoritma pemograman kuadrat, salah satunya adalah Hildreth's Quadratic Programming. Algoritma ini memanfaatkan metode active set, yaitu memilih set batasan yang aktif pada tiap iterasi. pada tiap iterasi, batasan pertidaksamaan dianggap menjadi batasan persamaan dan dihitung solusi x yang meminimalkan cost function. nilai x diperoleh dengan mencari nilai λ . Suatu batasan dikatakan aktif jika seluruh λ_i bernilai positif. Jika diperoleh terdapat λ_i negatif maka dilanjutkan iterasi berikutnya hingga diperoleh seluruh λ_i positif (Wang, 2009).

Pada algoritma Hildreth, jika diperlukan λ_i negatif untuk meminimalkan cost function, maka nilai λ_i tersebut diganti menjadi 0 (Wang, 2009). Algoritma Hildreth dapat dituliskan sebagai berikut (Lam et al., 2020)

Algoritma 1. Algoritma Hildreth

Input: $E, F, M, \gamma, \bar{m}, \delta$

Output: x

Step 1: hitung solusi global $x = -E^{-1}F$

Step 2: cek apakah solusi memenuhi batasan $Mx \leq \gamma$

If memenuhi batasan then

Stop

Else

Step 3: hitung

$$H = ME^{-1}M^T$$

$$K = \gamma + ME^{-1}F$$

End

Set $\lambda^m = 0$

Step 4:

For $m = 1: \bar{m}$ **do**

$$\omega_i^m = -\frac{1}{h_{ii}} \left[k_i + \sum_{j=1}^{i-1} h_{ij} \lambda_j^{m+1} + \sum_{j=j+1}^n h_{ij} \lambda_j^m \right]$$

$\lambda_i^m = \max(0, \omega_i^m)$

If $\|\lambda^m - \lambda^{m+1}\| < \delta$ **then**
 Stop
Else
 Continue
End
End
Step 5: return $x = -E^{-1}F - E^{-1}M^T \lambda^{m+1}$

2.2.7 Ekspansi Taylor Series

Pendekatan suatu fungsi dapat dilakukan menggunakan ekspansi taylor series. Misalkan ingin dilakukan pendekatan fungsi variabel tunggal $f(x)$ di sekitar titik a , maka ekspansi taylor series dapat dituliskan sebagai

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots + \frac{f^n(a)(x - a)^n}{n!} \quad (2.7)$$

2.2.7.1 Linearisasi dengan Taylor Series

Pendekatan linear dapat diperoleh menggunakan ekspansi taylor series orde 1. Sehingga pendekatan linear untuk persamaan (2.7) dapat dituliskan sebagai

$$f(x) = f(a) + f'(a)(x - a) \quad (2.8)$$

(Ravi Tailor & Bhathawala, 2011).

sedangkan jika fungsi menggunakan variabel ganda $f(x_1, x_2)$, pendekatan linear di sekitar titik a, b dituliskan sebagai

$$f(x_1, x_2) = f(a, b) + [f'(a)(x - a) + f'(b)(x - b)] \quad (2.9)$$

(Ogata, 2010).

BAB 3 METODOLOGI

3.1 Metode yang Digunakan

Tahapan sistematis digunakan agar tujuan tugas akhir dapat tercapai. Tahapan yang diterapkan adalah sebagai berikut:

1. Studi Literatur
Pada tahap ini dilakukan studi literatur untuk mempelajari dan mengkaji teori menyangkut topik Tugas Akhir. Pada tahap ini juga dilakukan pengumpulan data penelitian terdahulu yang dianggap relevan dan terpercaya untuk keabsahan Tugas Akhir. Sumber studi literatur berupa paper, jurnal, buku, dan artikel yang berasal dari badan pemerintahan ataupun institusi akademik terpercaya.
2. Pemodelan Sistem
Pada tahap ini, dilakukan pemodelan matematis untuk badan mobil otonom dengan parameter yang telah diperoleh dari literatur pada tahap sebelumnya. Selain itu, dibuat juga sistem untuk fungsi *trajectory tracking* serta *obstacle avoidance*.
3. Perancangan Kontroler
Kontroler berupa *Model Predictive Controller* yang dapat mengatur gerakan mobil otonom agar mobil dapat mengikuti jalur pada lingkungan yang telah dirancang. Perancangan kontroler dilakukan menggunakan Matlab.
4. Tahap Simulasi
Tahap ini merupakan tahap pengujian sistem dan kontroler yang telah dibuat pada tahap-tahap sebelumnya. Pengujian dilakukan untuk pengambilan data sebagai bahan analisa performansi sistem. Pengujian dibagi menjadi beberapa tahap yaitu pengujian subsistem untuk memastikan masing-masing subsistem yang dibuat berjalan dengan baik. Kemudian dilanjutkan dengan keseluruhan sistem untuk menguji kemampuan kontroler dalam mengatur gerakan mobil otonom.
5. Analisis dan Evaluasi
Data yang diperoleh pada tahap simulasi kemudian dianalisis dan dievaluasi. Evaluasi dilakukan untuk melihat seberapa baik mobil dapat melakukan *trajectory tracking* serta menghindari halangan dengan menjaga batasan fisik mobil.
6. Kesimpulan
Kesimpulan diperoleh berdasarkan pengujian, analisis data, dan referensi terkait. Tahap ini menunjukkan hasil Tugas Akhir sesuai dengan rumusan masalah mengenai performa dan keandalan *model predictive controller* dalam mengendalikan gerakan mobil otonom untuk tracking lintasan.
7. Penyusunan Laporan
Tahap terakhir dalam Tugas Akhir ini adalah penyusunan laporan. Laporan mencakup keseluruhan kegiatan yang dikerjakan pada Tugas Akhir ini. Laporan meliputi pendahuluan, studi literatur, perancangan sistem, pengujian dan analisis, serta penutup.

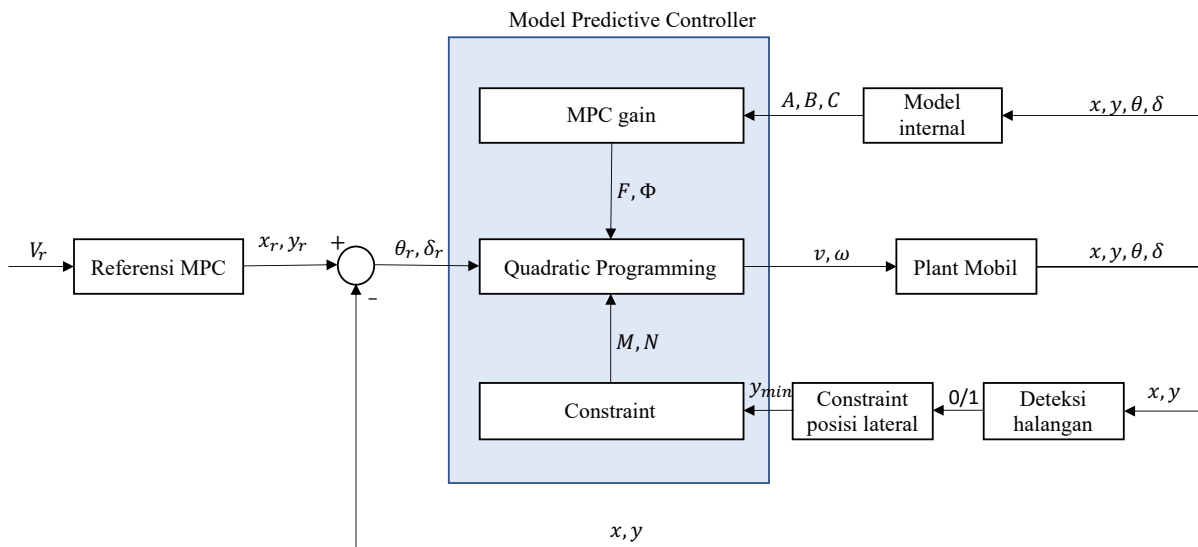
3.2 Bahan dan Peralatan yang Digunakan

Tugas akhir ini dilakukan dengan batasan pengujian yang dilakukan dengan simulasi pada aplikasi Matlab. Sehingga, peralatan yang digunakan berupa perangkat lunak, yaitu Matlab, serta perangkat keras berupa komputer pribadi.

3.3 Urutan Pelaksanaan Penelitian

3.3.1 Pemodelan Sistem

Studi literatur dilakukan dan telah dituliskan pada bab 2 laporan. Sesuai metode yang digunakan, langkah selanjutnya adalah melakukan pemodelan sistem. Secara garis besar, hubungan antar subsistem dapat digambarkan seperti gambar di bawah



Gambar 3.1 Diagram Konfigurasi Simulasi

Sesuai diagram di atas, pertama dimasukan referensi berupa kecepatan. Kemudian referensi diolah kembali menjadi referensi output berupa posisi x, posisi y, heading angle, dan steering angle mobil. referensi tersebut kemudian masuk ke dalam MPC dan MPC mulai menghitung nilai kecepatan dan steering rate untuk dimasukkan ke plant mobil. Kemudian diperoleh output baru yang digunakan untuk menghitung error, mengupdate model internal, dan mendeteksi halangan. Untuk mewujudkan sistem tersebut, pemodelan dilakukan dalam bentuk script Matlab secara bertahap sebagai berikut

3.3.1.1 Plant Mobil Otonom

Plant yang digunakan pada tugas akhir ini menggunakan fungsi Ackermann *Kinematics* yang telah tersedia pada Matlab. Fungsi ini, diketahui menggunakan persamaan non linear berikut

$$\begin{aligned} \dot{x} &= V \cos(\theta) \\ \dot{y} &= V \sin(\theta) \\ \dot{\theta} &= \frac{V}{L} \tan(\gamma) \\ \dot{\gamma} &= \omega \end{aligned} \quad (3.1)$$

Dengan input yaitu kecepatan mobil dan steering rate, $[V \ \omega]^T$.

3.3.1.2 Model Internal MPC

Model mobil yang digunakan sebagai model internal MPC adalah model kinematika. Penulis memilih menggunakan model kinematika agar sesuai dengan tujuan tugas akhir yang menitikberatkan pada fungsi mobil untuk tracking. Sehingga tidak diperlukan model dinamis yang mencantumkan gaya dan torsi mobil. Model internal yang digunakan MPC merupakan

model kinematika yang dilinearisasi menggunakan pendekatan taylor series orde 1, sehingga diperoleh

$$\begin{aligned}\dot{x} &= -V \sin(\theta) \cdot \dot{\theta} + \cos(\theta) \cdot V \\ \dot{y} &= V \cos(\theta) \cdot \dot{\theta} + \sin(\theta) \cdot V \\ \dot{\theta} &= \frac{\tan(\gamma)}{L} \cdot V + \frac{V}{L} (\tan^2(\gamma) + 1) \cdot \dot{\gamma} \\ \dot{\gamma} &= \omega\end{aligned}\quad (3.2)$$

Sesuai koordinat global, state yang digunakan adalah posisi x dan y, sudut heading mobil, dan sudut steering, $[x \ y \ \theta \ \gamma]^T$. Seluruh state ini digunakan sebagai output. Sedangkan input adalah kecepatan mobil dan steering rate, $[V \ \omega]^T$ Sehingga dapat dituliskan dalam bentuk state space seperti berikut

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} \quad (3.3)$$

Dengan

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -V \sin(\theta) & 0 \\ 0 & 0 & V \cos(\theta) & 0 \\ 0 & 0 & 0 & \frac{V}{L} (\tan^2(\gamma) + 1) \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ \frac{\tan(\gamma)}{L} & 0 \\ 0 & 1 \end{bmatrix} \quad (3.4)$$

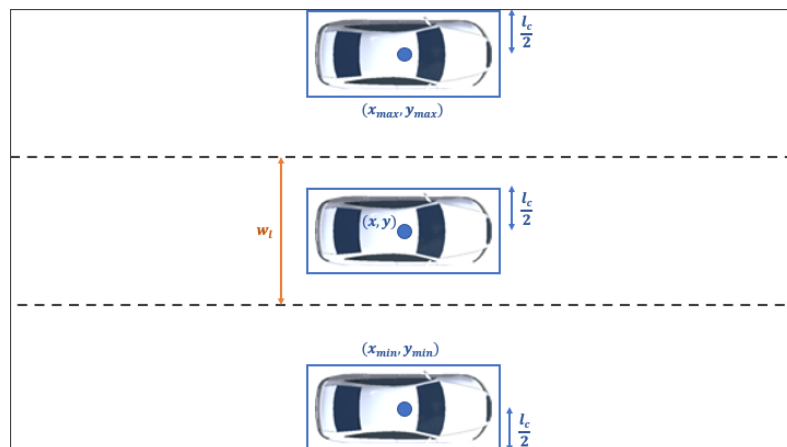
Lalu model state space didiskritkan dengan metode euler sesuai persamaan di bawah

$$\begin{aligned}\mathbf{X}(k+1) &= \mathbf{X}(k) + \dot{\mathbf{X}} \cdot T_s \\ &= \mathbf{X}(k) + [\mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{U}(k)] \cdot T_s \\ &= \mathbf{X}(k) + [\mathbf{A} \cdot T_s + \mathbf{I}] \cdot \mathbf{X}(k) + \mathbf{B} \cdot T_s \cdot \mathbf{U}(k)\end{aligned}\quad (3.5)$$

Dengan waktu sampling (T_s) 0.1 s.

3.3.1.3 Constraint Posisi Mobil

Pada tugas akhir ini diasumsikan bahwa mobil berjalan pada jalur dengan 3 lajur. Tiap lajur memiliki lebar sebesar 4 meter ($w_l=4$), seperti gambar di bawah



Gambar 3.2 Jalur Simulasi

jalur didefinisikan sebagai hard constraint pada posisi lateral mobil untuk model predictive controller seperti dibawah

$$\begin{aligned}
y_{min} &\leq y \leq y_{max} \\
\frac{l_c}{2} &\leq y \leq (3 \cdot w_l) - \frac{l_c}{2} \\
-6 &\leq y \leq 6
\end{aligned}
\tag{3.6}$$

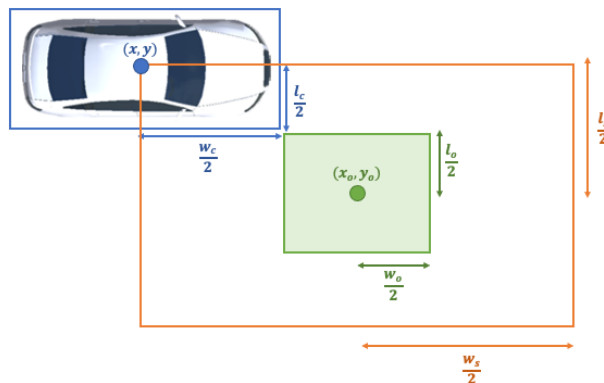
3.3.1.4 Obstacle Avoidance

Halangan berupa halangan statis dengan dimensi dan posisi yang telah diberikan. Halangan didefinisikan sebagai constraint yang membatasi output posisi lateral mobil. Constraint akan aktif jika pendeteksi halangan mendeteksi adanya objek di depan mobil pengujian. Karena pada penelitian ini tidak dilakukan path planning ataupun trajectory planning, keputusan mobil untuk menghindari halangan melalui lajur kiri atau kanan ditentukan sebelumnya oleh penulis.

Halangan statis yang digunakan memiliki dimensi $l_o \times w_o$, dimana l_o dan w_o adalah Panjang dan lebar halangan. Halangan selalu diletakkan di tengah lajur. Untuk memastikan bahwa mobil tidak menabrak halangan selagi berjalan maka dibuat “zona aman” di sekitar halangan. Titik tengah zona aman diletakkan pada titik tengah halangan. Maka, dihitung dari titik tengahnya, zona aman adalah sebagai berikut

- Panjang zona aman (l_s) = $\frac{1}{2}$ (panjang halangan (l_o) + Panjang mobil (l_c))
- Lebar zona aman (w_s) = $\frac{1}{2}$ (lebar halangan (w_o) + lebar mobil (w_c))

untuk lebih jelas dapat dilihat pada gambar di bawah



Gambar 3.3 Zona Aman

kemudian zona aman dapat dituliskan sebagai constraint posisi lateral (y) seperti berikut

$$\begin{aligned}
y_{min} &\leq y \\
y_o + \frac{1}{2}(l_o + l_c) &\leq y
\end{aligned}
\tag{3.7}$$

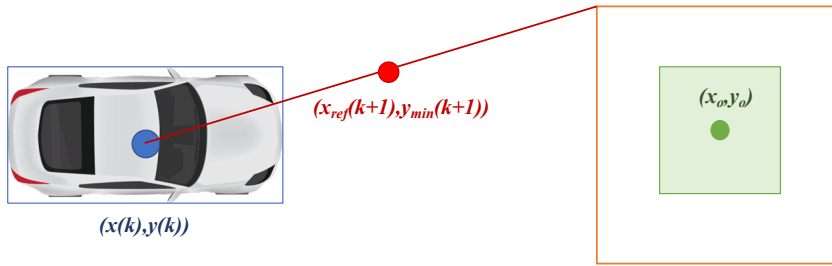
Diasumsikan bahwa mobil memiliki panjang 4 meter dan lebar 2 meter. Sedangkan halangan memiliki panjang 5 meter dan lebar 2 meter. Pada tugas akhir ini diasumsikan mobil dilengkapi dengan Lidar yang dapat mendeteksi adanya objek sejauh 50 meter di depan. Perangkat lidar dituliskan dalam bentuk fungsi Matlab. Fungsi akan memberikan sinyal 1 atau true jika pertidaksamaan berikut terpenuhi

$$|(x_o + x)^2 + (y_o + y)^2| < 50 \quad (3.8)$$

Sistem obstacle avoidance bekerja dengan cara mendefinisikan constraint posisi lateral atau posisi Y baru menyesuaikan dengan posisi instan mobil. Selama pendeteksi objek mendeteksi halangan maka akan dihitung kemiringan/ gradien (m) antara posisi mobil dengan zona aman terdekat. Kemudian nilai gradien digunakan untuk membentuk garis lurus untuk menentukan constraint posisi Y yang baru. Sehingga constraint posisi y baru dapat dirumuskan sebagai

$$y_{min} = m (x_{ref} - x) + y \quad (3.9)$$

Untuk lebih jelas dapat dilihat pada gambar di bawah



Gambar 3.4 Constraint Posisi Y untuk Obstacle Avoidance

3.3.2 Perancangan Kontroler

Selain mengatur parameter horizon kontrol (N_c) dan horizon prediksi (N_p). Diperlukan juga model internal plant, definisi cost function, dan constraint. Pada desain MPC tugas akhir ini, model sistem berupa state space diskrit sesuai yang didefinisikan sebelumnya pada persamaan (3.3).

3.3.2.1 Model prediksi MPC

Sesuai model internal yang digunakan, prediksi output MPC didefinisikan sebagai

$$\vec{Y} = Fx(k) + \Phi\vec{U}(k) \quad (3.10)$$

dengan

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} \quad \Phi = \begin{bmatrix} CB & 0 & 0 & 0 & \dots \\ CAB & CB & 0 & 0 & \dots \\ CA^2B & CAB & CB & 0 & \dots \\ CA^3B & CA^2B & CAB & CB & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \quad (3.11)$$

Dimana $X \in R^{4 \times 1}$, $A \in R^{4 \times 4}$, $B \in R^{4 \times 2}$, $C \in R^{4 \times 4}$, $\vec{Y} \in R^{4 \cdot N_p \times 1}$, $F \in R^{4 \cdot N_p \times 4}$, $\Phi \in R^{4 \cdot N_p \times 2 \cdot N_c}$ dan $\vec{U} \in R^{2 \cdot N_c \times 1}$. Serta N_p sebagai horizon prediksi, N_c horizon kontrol, matriks A dan B diperoleh dari model internal MPC pada persamaan (3.4). Adapun, karena seluruh state dipilih menjadi output maka matriks C merupakan matriks identitas.

3.3.2.2 Constraint

Seperti yang telah dijelaskan, pada sistem mobil terdapat beberapa batasan atau biasa disebut constraint. Constraint dapat dibagi menjadi 2 bagian utama dalam perumusan MPC, yaitu constraint input dan constraint output. Constraint input dirumuskan sesuai dengan batasan

keamanan dan batasan fisik yang dimiliki mobil. Pada penelitian ini digunakan constraint input sebagai berikut

Tabel 3.1 Constraint Input

Variable Input	Constraint
Kecepatan	$30 \text{ km/jam} \leq V \leq 100 \text{ km/jam}$
Steering Rate	$-60^\circ/\text{sec} \leq \omega \leq 60^\circ/\text{sec}$

Kecepatan paling rendah dipilih 30 km/jam sesuai dengan batas kecepatan untuk berkendara dalam kota, sedangkan dipilih kecepatan maksimal 100 km/jam sesuai dengan batas kecepatan untuk berkendara pada jalan tol. Batasan steering rate dipilih -10 hingga 10 derajat sesuai dengan batasan fisik mobil. Nilai ini diperoleh dari referensi penelitian yang telah dilakukan oleh Turri et al.

Constraint output pada penelitian ini ditentukan berdasarkan permasalahan penelitian, batasan penelitian, dan batasan fisik mobil. Sesuai yang dijelaskan sebelumnya, terdapat 4 output yaitu posisi X, posisi Y, heading angle, dan steering angle mobil. Output posisi X tidak dibatasi karena mobil harus bergerak menyesuaikan referensi yang ditentukan. Output posisi Y dibatasi yaitu agar tidak keluar dan harus berada sesuai zona aman untuk memastikan mobil tidak menabrak halangan. Output heading angle bernilai $-90^\circ \leq \theta \leq 90^\circ$ yang artinya mobil dibatasi untuk tidak dapat melakukan gerakan putar balik menuju x negatif. Terakhir adalah output steering angle senilai $-10^\circ \leq \gamma \leq 10^\circ$ sesuai dengan batasan fisik mobil yang mereferensi dari penelitian yang dilakukan oleh Turri et al..

3.3.2.3 Cost Function

Cost function MPC dibentuk sesuai dengan tujuan tugas akhir, yaitu untuk menjalankan fungsi trajectory tracking. Maka perlu didefinisikan cost function dengan tujuan untuk meminimalkan eror output. Selain itu, terdapat objektif lain yaitu pengeluaran energi yang optimal untuk input. Pada persamaan cost function (3.10) bobot dari input dinotasikan sebagai R dan bobot output posisi dinotasikan sebagai Q.

$$\begin{aligned} \min J &= \vec{U}^T \bar{R} \vec{U} + (R_s - \vec{Y})^T Q (R_s - \vec{Y}) \\ J &= \vec{U}(k)^T (\Phi^T \Phi + \bar{R}) \vec{U}(k) + 2 \vec{U}(k)^T \Phi^T (R_s - Fx(k)) \\ &\quad + (R_s - Fx(k))^T (R_s - Fx(k)) \end{aligned} \quad (3.12)$$

Jika dicocokkan dengan bentuk umum (2.4) diperoleh

$$\begin{aligned} J &= \frac{1}{2} \left(\vec{U}(k)^T \underbrace{2(\Phi^T \Phi + \bar{R})}_H \vec{U}(k) + \vec{U}(k)^T \Phi^T \underbrace{4(R_s - Fx(k))}_f \right. \\ &\quad \left. + 2(R_s - Fx(k))^T (R_s - Fx(k)) \right) \end{aligned} \quad (3.13)$$

Dengan constraint

$$M \cdot \vec{U}(k) \leq N \quad (3.14)$$

Keterangan:

$$\vec{U} = \begin{bmatrix} U(k+1|k) \\ U(k+2|k) \\ \vdots \\ U(k+N_c|k) \end{bmatrix} \quad (3.15)$$

N merupakan constraint gabungan antara constraint input ($\mathbf{U}(k)$) dan constraint output ($\mathbf{Y}(k)$). Constraint ketiga variable diperoleh dari definisi constraint lingkungan serta constraint fisik mobil. Perumusan constraint input dapat dilihat pada persamaan di bawah

$$\begin{aligned} &Umin \leq U \leq Umax \\ &\underbrace{\begin{bmatrix} I_{nu} \\ I_{nu} \\ \vdots \\ I_{nu} \end{bmatrix}}_{c2} \underbrace{\begin{bmatrix} U(1)min \\ U(2)min \\ \vdots \\ U(nu)min \end{bmatrix}}_{Umin} \leq \vec{U}(k) \leq \underbrace{\begin{bmatrix} I_{nu} \\ I_{nu} \\ \vdots \\ I_{nu} \end{bmatrix}}_{c2} \underbrace{\begin{bmatrix} U(1)max \\ U(2)max \\ \vdots \\ U(nu)max \end{bmatrix}}_{Umax} \\ &-\vec{U}(k) \leq -c2 \mathbf{Umin} \\ &\vec{U}(k) \leq c2 \mathbf{Umax} \end{aligned} \quad (3.16)$$

Sedangkan constraint input berdasarkan batasan output dirumuskan sebagai

$$\begin{aligned} &Ymin \leq Fx(k) + \Phi \vec{U}(k) \leq Ymax \\ &Ymin - Fx(k) \leq \Phi \vec{U}(k) \leq Ymax - Fx(k) \\ &-\Phi \vec{U}(k) \leq -Ymin + Fx(k) \\ &\Phi \vec{U}(k) \leq Ymax - Fx(k) \end{aligned} \quad (3.17)$$

Sehingga constraint (3.12) dapat didefinisikan lebih lanjut sebagai

$$\begin{aligned} &M = \begin{bmatrix} M1 \\ M2 \end{bmatrix}, N = \begin{bmatrix} N1 \\ N2 \end{bmatrix} \\ &M1 = \begin{bmatrix} -c2 \\ c2 \end{bmatrix}, M2 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \\ &N1 = \begin{bmatrix} -\mathbf{Umin} \\ \mathbf{Umax} \end{bmatrix}, N2 = \begin{bmatrix} -Ymin + Fx(k) \\ Ymin - Fx(k) \end{bmatrix} \end{aligned} \quad (3.18)$$

3.3.2.4 Hildreth Quadratic Programming

Kontroler MPC menyelesaikan permasalahan kuadrat menggunakan algoritma Hildreth. Perhitungan λ optimal (λ^*) akan berhenti jika telah mencapai iterasi maksimal atau jika perubahan nilai λ^* kurang dari 10^{-6} sehingga dapat dikatakan bahwa hasil perhitungan telah konvergen. Pembatasan iterasi maksimal akan digunakan sebagai variabel penelitian.

3.3.3 Perumusan Referensi

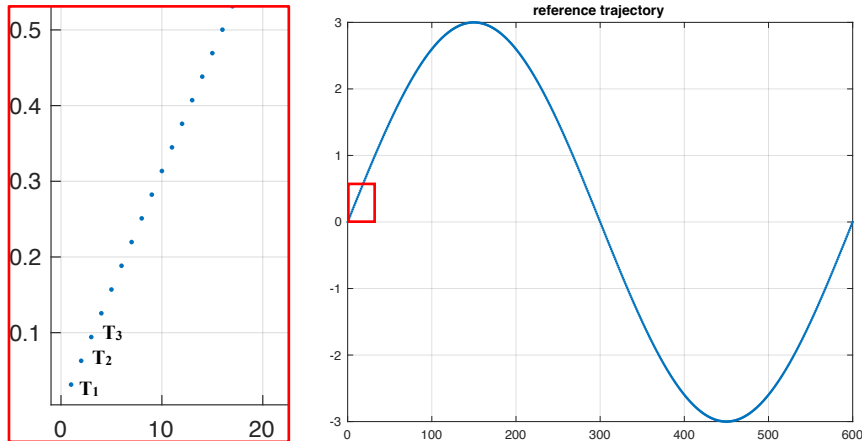
Sesuai dengan permasalahan trajectory tracking, referensi yang digunakan merupakan referensi yang berubah terhadap waktu. Pada penelitian ini perumusan referensi dilakukan dengan memasukkan kecepatan referensi terlebih dahulu. Kemudian dirumuskan referensi posisi X sebagai titik yang berubah terhadap waktu seperti berikut

$$x(k) = V_{ref} \cdot T_s \cdot k$$

Dimana T_s merupakan waktu sampling dan k adalah iterasi.

Kemudian dilakukan pembentukan referensi posisi y sebagai fungsi x. Pada pengujian trajectory tracking digunakan 3 track yaitu 2 track sinusoidal dan 1 track pergantian lajur. Track sinusoidal pertama yaitu dengan rumusan seperti di bawah

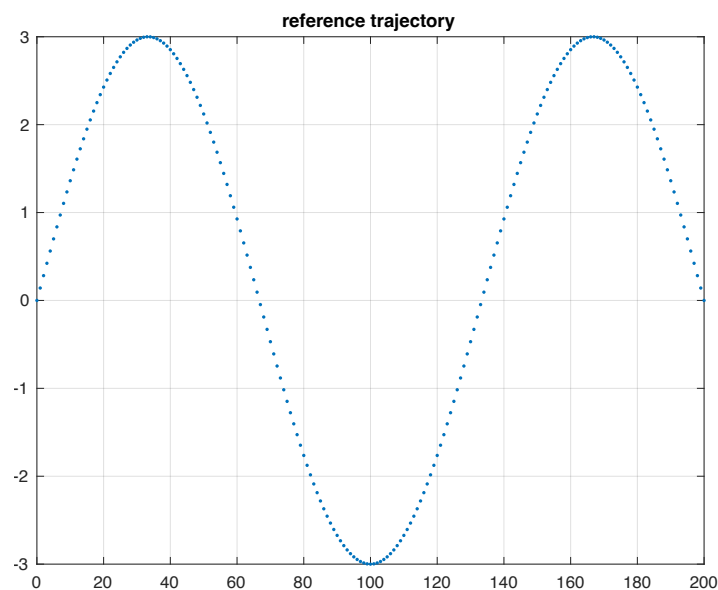
$$y_{ref}(k) = 3 \sin \left(x_{ref}(k) \left(\frac{2\pi}{60/T_s} \right) \right) \quad (3.19)$$



Gambar 3.5 Trayektori Referensi Sinusoidal untuk Simulasi Trajectory Tracking

Sementara track sinusoidal 2 menggunakan frekuensi dan amplitudo yang lebih tinggi untuk melihat hasil gerakan MPC jika dibutuhkan steering angle melebihi constraint yang dibutuhkan. Pada referensi ini dibutuhkan sudut steering lebih dari 10 derajat.

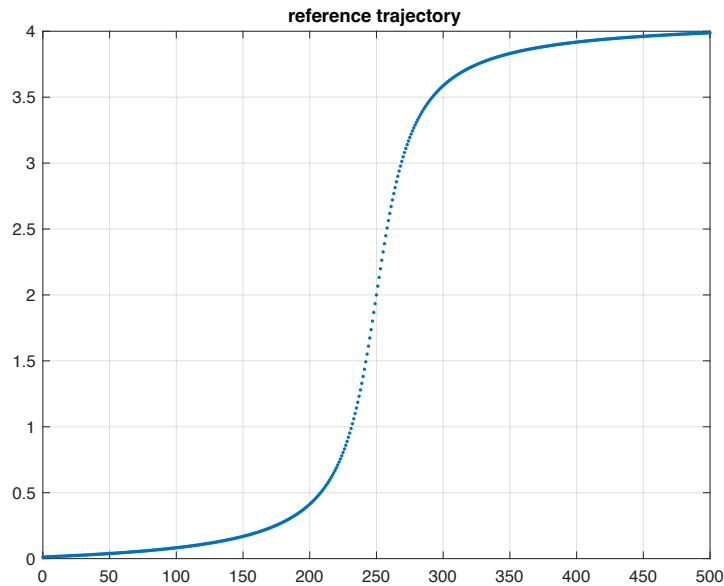
$$y_{ref}(k) = 3 \sin \left(x_{ref}(k) \left(\frac{9\pi}{60/T_s} \right) \right) \quad (3.20)$$



Gambar 3.6 Trayektori Referensi Sinusoidal 2 untuk Simulasi Trajectory Tracking

Track pengujian ketiga adalah track untuk pergantian lajur yang dirumuskan dan digambarkan seperti di bawah

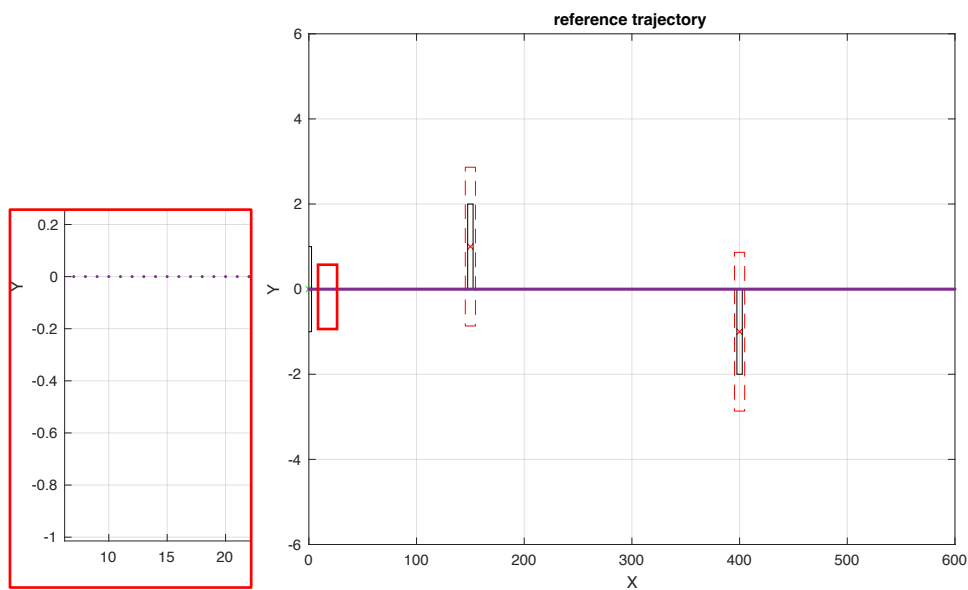
$$y(k) = \frac{1}{0.75} \tan^{-1} \left(\frac{x_{ref}(k) - 250}{20} \right) + 2 \quad (3.21)$$



Gambar 3.7 Trayektori Referensi Pergantian lajur

Sementara pada pengujian obstacle avoidance, digunakan referensi berupa garis lurus seperti berikut

$$y_{ref}(k) = 0 \quad (3.22)$$



Gambar 3.8 Trayektori Referensi untuk Simulasi Obstacle Avoidance

Perhitungan referensi MPC $[x_r \ y_r \ \theta_r \ \gamma_r]^T$ dilakukan dengan membandingkan posisi mobil pada step k dengan trayektori referensi pada step k .

(Halaman ini sengaja dikosongkan)

BAB 4 HASIL DAN PEMBAHASAN

Pengujian dibagi menjadi 2 bagian utama yaitu untuk mengevaluasi performa MPC dalam trajectory tracking dan yang ke-2 adalah untuk mengevaluasi performa MPC untuk fungsi obstacle avoidance. Berikut adalah parameter yang digunakan dalam simulasi

Tabel 4.1 Parameter Simulasi

Parameter	Nilai
Panjang Mobil	4 m
Lebar Mobil	2 m
Lebar Jalan	4 m
Bobot Error Output Cost Function (Q)	0.4
Bobot Input Cost Function (R)	0.6

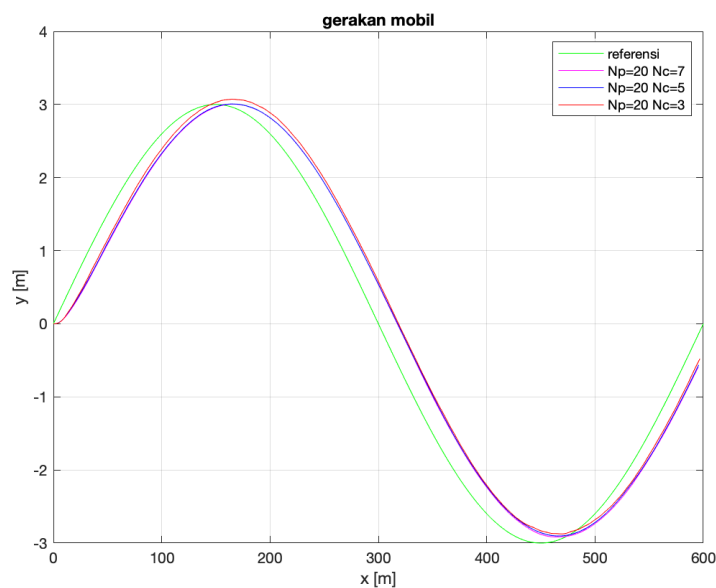
Pengujian sistem dilakukan dengan memvariasikan kombinasi horizon prediksi (N_p), horizon kontrol (N_c) dan iterasi maksimal quadratic programming. Evaluasi dilakukan dengan membandingkan gerakan mobil dan membandingkan nilai RMSE posisi X, Y, posisi absolut, cost function, dan waktu komputasi.

4.1 Trajectory Tracking

Pengujian trajectory tracking dibagi menjadi 4 bagian. Bagian pertama adalah dengan track referensi sinusoidal dengan kecepatan 10 m/s. Bagian ke-2 adalah pengujian dengan track referensi sinusoidal dengan kecepatan 20 m/s. bagian ke-3 adalah pengujian menggunakan track referensi untuk pergantian lajur. Bagian terakhir adalah pengujian dengan sinyal referensi sinusoidal yang membutuhkan sudut steering di luar constraint.

4.1.1 Track Sinusoidal dengan Kecepatan 10 m/s

Pengujian pertama dari trajectory tracking adalah dengan Berikut adalah hasil yang diperoleh ketika mobil dijalankan dengan referensi kecepatan sebesar 10m/s.



Gambar 4.1 Trajectory Tracking 1 dengan kecepatan 10m/s

dengan perbandingan RMSE, cost function, dan waktu komputasi dari masing-masing variasi sebagai berikut.

Tabel 4.2 Trajectory Tracking 1 dengan Kecepatan 10m/s

Iterasi Maksimal QP	Horizon Kontrol (Nc)	Horizon Prediksi (Np)	RMSE posisi X	RMSE posisi Y	RMSE posisi	Cost Function	Waktu komputasi
40	3	20	3.25	0.41	3.28	192.77	0.05
40	5	20	4.24	0.43	4.27	274.35	0.05
40	7	20	4.67	0.44	4.70	313.58	0.05

Terlihat dari hasil pengujian pertama dengan track sinusoidal pada kecepatan 10 m/s bahwa gerakan mobil yang dihasilkan oleh MPC menyerupai bentuk referensi yang diberikan. Namun terdapat eror posisi X sebesar 3.28- 4.70 m. Artinya pencapaian posisi mobil masih terlambat dibandingkan dengan referensi yang diberikan. Hal ini sesuai dengan perhitungan bahwa jika mobil berkendara dengan kecepatan 10 m/s, maka mobil memiliki perpindahan 1 m pada tiap time step. Sehingga jika mobil memiliki rata-rata eror yang dihitung pada time step sebesar 3 m, artinya posisi mobil terlambat selama 3 time step yaitu 0.3 s. Sehingga pada pengujian pertama ini, diperoleh bahwa mobil terlambat selama 0.32 – 0.47 s untuk mencapai posisi referensi. Keterlambatan ini bisa jadi dikarenakan mobil tidak langsung melaju dengan kecepatan 10 m/s. Dimana mobil memerlukan waktu transien untuk menyesuaikan kecepatan dari 0 m/s menuju 10 m/s.

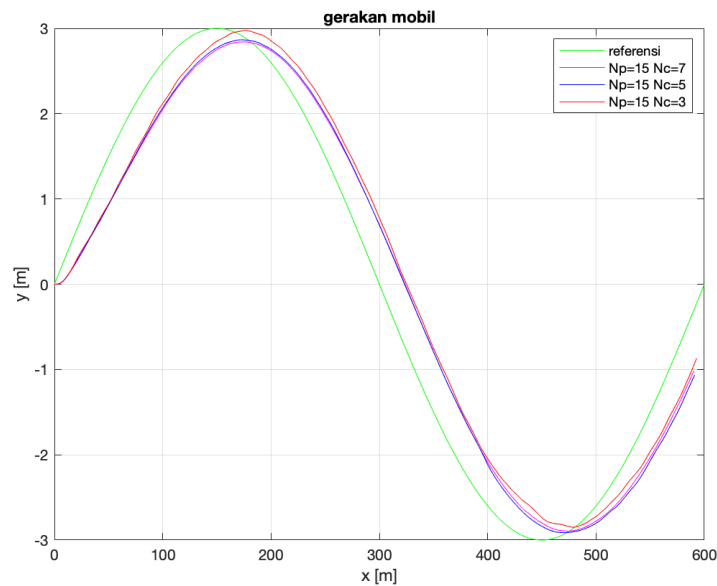
Pada pengujian pertama juga terlihat bahwa penggunaan iterasi maksimal sebanyak 40 iterasi cukup untuk mendapatkan nilai input optimal untuk menghasilkan gerakan yang menyerupai referensi. Hal ini ditunjukkan oleh eror posisi Y yang tidak terlalu besar yaitu 0.41 – 0.44 m. Ditinjau dari posisi Y mobil, juga terlihat bahwa posisi mobil selalu berada di dalam jalur yang ditentukan terlihat dari posisi yang tidak kurang dari -6 dan tidak lebih dari 6.

Pemilihan kombinasi parameter MPC mempengaruhi performa MPC. Terlihat bahwa semakin besar nilai horizon kontrol (Nc) maka semakin besar nilai eror dan nilai cost function. Hal ini sesuai dengan konsep MPC yang memanfaatkan input masa depan dalam perhitungan permasalahan optimisasi atau quadratic programming. Semakin jauh nilai horizon yang dipilih maka akan meningkatkan kompleksitas perhitungan optimisasi tersebut. Waktu komputasi terlihat tidak terpengaruh dengan perubahan nilai horizon kontrol yang dipilih. Artinya, perbedaan antara horizon kontrol 3, 5, dengan 7 tidak begitu signifikan sehingga waktu komputasi bisa tetap berada pada 0.05 s. Hal ini menunjukkan bahwa waktu perhitungan MPC untuk menentukan 1 input pada tiap time step memenuhi syarat harus kurang dari sampling yang ditentukan, pada penelitian ini yaitu 0.1 s.

Performa MPC untuk trajectory tracking dengan sinyal sinusoidal paling baik pada kecepatan 10 m/s adalah dengan variasi iterasi maksimal quadratic programming= 40, Nc= 3, dan Np=20 dengan nilai RMSE posisi 3.28 m dan keterlambatan pencapaian posisi selama 0.32 s.

4.1.2 Track Sinusoidal dengan Kecepatan 20 m/s

Pengujian selanjutnya menggunakan kecepatan tinggi yaitu 20 m/s dan diperoleh hasil seperti berikut



Gambar 4. 1 Trajectory Tracking dengan Kecepatan 20m/s

dengan perbandingan RMSE, cost function, dan waktu komputasi dari masing-masing variasi sebagai berikut.

Tabel 4.3 Trajectory Tracking dengan Kecepatan 20m/s

Iterasi Maksimal QP	Horizon Kontrol (Nc)	Horizon Prediksi (Np)	RMSE posisi X	RMSE posisi Y	RMSE posisi	Cost Function	Waktu Komputasi
80	3	15	7.03	0.63	7.06	698.11	0.05
80	5	15	8.77	0.66	8.80	930.13	0.05
80	7	15	9.46	0.67	9.48	1031.62	0.05

Pada pengujian ke-2 dengan track sinusoidal pada kecepatan 20 m/s, diperoleh hasil yang sama pada kecepatan 10 m/s. Yaitu bahwa gerakan yang dihasilkan MPC telah menyerupai referensi yang diberikan. Selain itu juga terdapat keterlambatan pencapaian posisi yang ditandai oleh eror posisi X sebesar 7.03 – 9.46 m. Dapat dilakukan perhitungan yang sama, yaitu dengan kecepatan 20 m/s perpindahan mobil harus sebesar 2 m pada tiap time step. Artinya pada pengujian ini, mobil terlambat 0.35– 0.47 s dalam pencapaian posisi disebabkan oleh waktu transien yang dibutuhkan mobil untuk mencapai kecepatan 20 m/s dari 0 m/s. Maka dapat disimpulkan bahwa, MPC mengatur kecepatan mobil agar sesuai dengan referensi dengan delay rata-rata 0.4 s. Sehingga, dengan eror waktu yang konstan, MPC lebih baik digunakan untuk kecepatan yang lebih kecil untuk menghasilkan eror posisi yang lebih kecil.

Pemilihan kombinasi parameter pada kecepatan 20 m/s berbeda dengan 10 m/s. Pada pengujian ini digunakan horizon prediksi (Np) sejauh 15 dengan iterasi maksimal quadratic programming 80 kali. Pemilihan ini didasarkan percobaan pribadi peneliti, dimana iterasi maksimal 40 kali tidak cukup untuk menghasilkan gerakan yang stabil. Ditemukan juga bahwa

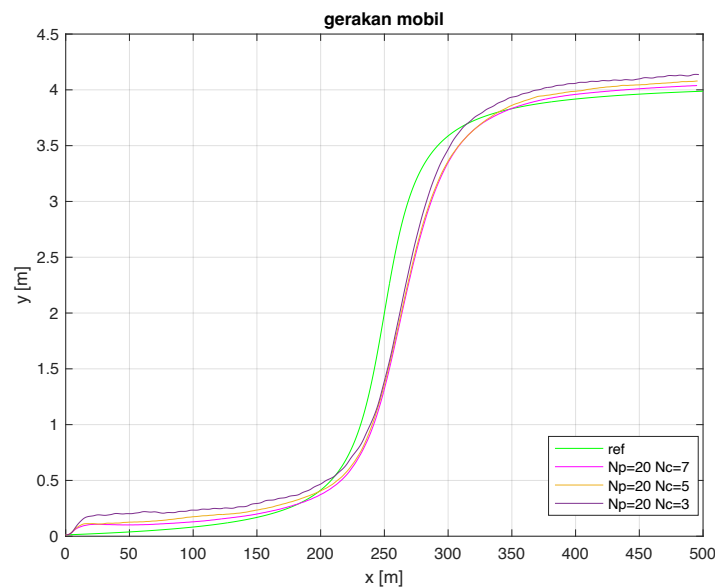
dengan iterasi 80 kali, pemilihan $N_p=15$ sudah dapat mengatur gerakan mobil yang baik. Hal ini ditunjukkan dari nilai eror posisi Y yaitu sebesar $0.63 - 0.67$ m. Artinya pemilihan antara kombinasi yang dipilih sangat penting dalam desain MPC. Masing-masing parameter perlu ditentukan dengan baik untuk menghasilkan desain yang seimbang.

Adapun efek dari peningkatan N_c yang sudah terlihat pada pengujian 1. Yaitu semakin besar nilai N_c maka semakin besar juga nilai eror serta cost function yang dihasilkan. Waktu komputasi juga tidak dipengaruhi oleh pemilihan N_c bernilai 3, 5, dan 7. Artinya penggunaan kecepatan lebih tinggi tidak mempengaruhi perhitungan MPC sama sekali.

Performa MPC untuk trajectory tracking dengan sinyal sinusoidal paling baik pada kecepatan 20 m/s performa MPC paling baik adalah dengan variasi iterasi maksimal quadratic programming= 80, $N_c=3$, dan $N_p= 15$ dengan nilai RMSE posisi 7.06 m dan keterlambatan pencapaian posisi selama 0.35 s.

4.1.3 Track Pergantian Lajur

Pengujian selanjutnya yaitu dengan track yang menggambarkan pergantian lajur. Pada pengujian ini, dilakukan pergantian dari lajur tengah ke lajur kiri. Hasil simulasi adalah sebagai berikut



Gambar 4.2 Trajectory Tracking Pergantian Lajur

dengan perbandingan RMSE, cost function, dan waktu komputasi dari masing-masing variasi sebagai berikut.

Tabel 4.4 Trajectory Tracking Pergantian Lajur

Iterasi Maksimal QP	Horizon Kontrol (N_c)	Horizon Prediksi (N_p)	RMSE Posisi X	RMSE Posisi Y	RMSE Posisi	Cost Function	Waktu Komputasi
40	3	20	3.258	0.259	3.268	192.170	0.05
40	5	20	4.246	0.281	4.256	273.164	0.05
40	7	20	4.669	0.295	4.678	312.121	0.05

Pada pengujian ke-3 dengan track pergantian lajur dan kecepatan 10 m/s diperoleh gerakan yang telah menyerupai track referensi. MPC berhasil mengatur gerakan mobil sehingga mobil dapat berpindah dari lajur tengah ke lajur kiri. Ditinjau dari eror posisi X sebesar 3.25 – 4.66 m, dapat dihitung juga keterlambatan pencapaian posisi pada pengujian ini adalah selama 0.32 – 0.46 s dengan rata-rata 0.4 s. Hal ini sesuai pada pengujian 1 dan 2 dengan track sinusoidal 1. Ditinjau dari eror posisi Y, pengujian ini menghasilkan nilai eror lebih kecil dibandingkan pengujian dengan track sinusoidal. Hal ini karena track sinusoidal lebih kompleks dibandingkan track pergantian lajur dalam hal pengaturan steering mobil.

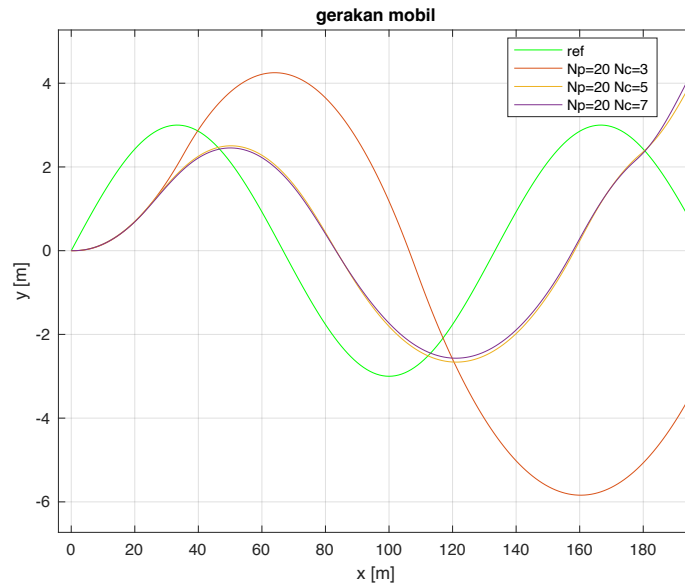
Variasi $N_c=7$ menghasilkan gerakan osilasi dengan amplitudo kecil pada waktu sebelum dan sesudah melakukan manuver pergantian lajur. Artinya MPC kesulitan untuk mengatur gerakan lurus dibandingkan gerakan dengan belokan. Artinya pemilihan N_c yang semakin jauh belum tentu menghasilkan gerakan yang lebih baik secara kualitatif. Sesuai dengan pengujian ke-2, dimana diperlukan kombinasi yang seimbang antar parameter MPC untuk menghasilkan gerakan yang bagus. Sementara pengaruh dari horizon kontrol terlihat sama seperti pengujian-pengujian sebelumnya, yaitu semakin besar nilai N_c maka semakin besar nilai eror dan cost function yang dihasilkan.

Performa MPC paling baik pada pengujian ke-3 dapat ditinjau dari 2 sisi yaitu kuantitatif dan kualitatif. Secara kuantitatif, performa terbaik dihasilkan oleh variasi iterasi maksimal quadratic programming= 40, $N_p=20$, dan $N_c=3$ dengan nilai RMSE posisi paling kecil yaitu 3.268 m. Secara kualitatif, performa terbaik dihasilkan oleh variasi iterasi maksimal quadratic programming= 40, $N_p=20$, dan $N_c=5$ karena menghasilkan gerakan menyerupai track referensi. Meskipun gerakan yang dihasilkan serupa dengan variasi $N_c=7$, dipilih $N_c=5$ merupakan yang terbaik karena secara kualitatif memiliki nilai eror yang lebih kecil yaitu RMSE posisi sebesar 4.256 m.

Keseluruhan pengujian trajectory tracking menunjukkan bahwa MPC dapat mengatur gerakan mobil menyerupai bentuk track referensi yang diberikan. Namun terdapat keterlambatan pencapaian posisi selama 0.4 s. Kompleksitas track referensi juga mempengaruhi performa MPC terlihat dari eror pada pengujian pergantian lajur memiliki eror posisi lebih kecil dibandingkan pengujian dengan track sinusoidal. Selain itu, MPC selalu menjaga agar sistem selalu sesuai dengan constraint yang diberikan. Namun, terdapat kekurangan yaitu dengan tidak pernah terpenuhinya demand, eror semakin besar sehingga sistem menjadi tidak stabil.

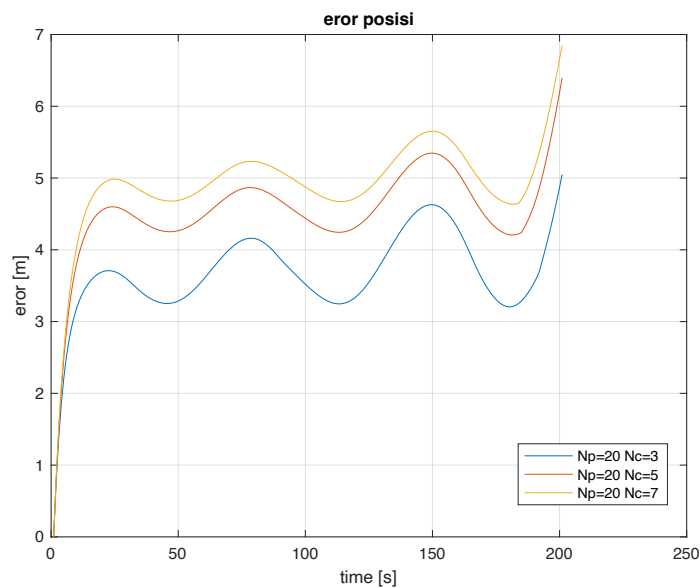
4.1.4 Track Sinusoidal dengan Sudut Steering Melebihi Constraint

Pengujian terakhir dari trajectory tracking yaitu track referensi sinusoidal yang membutuhkan sudut steering lebih dari batasan. Pada penelitian ini constraint sudut steering adalah $-10^\circ \leq \gamma \leq 10^\circ$. Pengujian ini dilakukan untuk mengevaluasi batasan dari MPC dan menitik beratkan pada pengaruh demand input yang melebihi constraint. Hasil simulasi adalah sebagai berikut



Gambar 4.3 Trajectory Tracking 2 dengan Kecepatan 10m/s

terlihat bahwa hasil gerakan mobil semakin menjauhi bentuk referensi. Oleh karena itu, pada pengujian ini diambil data error posisi sebagai bahan analisis tambahan. Grafik error posisi dapat dilihat seperti di bawah



Gambar 4.4 Error Posisi Trajectory Tracking 2 dengan Kecepatan 10m/s

dengan perbandingan RMSE, cost function, dan waktu komputasi dari masing-masing variasi sebagai berikut.

Tabel 4.5 Trajectory Tracking 2 dengan Kecepatan 10m/s

Iterasi Maksimal QP	Horizon Kontrol (Nc)	Horizon Prediksi (Np)	RMSE Posisi X	RMSE Posisi Y	RMSE Posisi	Cost Function	Waktu Komputasi
40	3	20	3.103	2.044	3.716	227.656	0.05
40	5	20	4.101	2.048	4.584	315.727	0.05
40	7	20	4.522	2.052	4.966	356.103	0.05

Pada pengujian ini, steering angle yang dibutuhkan melebihi $\pm 10^\circ$, artinya melebihi constraint yang telah ditentukan. Terlihat bahwa pada variasi dengan $N_c=5$ dan $N_c=7$ gerakan mobil masih bisa menyerupai bentuk referensi pada periode pertama atau dapat membentuk 1 gelombang sinusoidal. Setelah itu gerakan mobil terlihat menjauhi bentuk track referensi yang diberikan. Pada kedua variasi ini juga terlihat bahwa posisi Y mobil saat melakukan belokan lebih rendah dibandingkan posisi Y belokan referensi. Hal ini dikarenakan MPC menjaga agar steering angle tidak lebih dari $\pm 10^\circ$ sehingga mobil tidak bisa menghasilkan gerakan dengan belokan tajam.

Pada variasi dengan $N_c=3$, kompensasi MPC terhadap demand constraint berbeda dibandingkan dengan 2 variasi lainnya. Posisi Y gerakan mobil jauh melebihi Posisi Y track referensi saat melakukan belokan. Selain itu posisi X mobil juga jauh dibandingkan posisi X referensi. Artinya perhitungan MPC menghasilkan gerakan dengan steering rate rendah agar diperoleh sudut steering sesuai constraint, sehingga proses gerakan belok menjadi lebih lama dibandingkan 2 variasi lainnya.

Dilihat dari grafik eror, terlihat bahwa semakin besar nilai N_c maka semakin besar nilai eror. Pola ini sesuai dengan 2 pengujian sebelumnya dengan track sinusoidal 1. Meskipun nilai eror variasi $N_c=3$ secara kuantitatif merupakan nilai paling baik. Secara kualitatif, variasi $N_c=5$ dan $N_c=7$ menghasilkan gerakan yang lebih baik karena lebih menyerupai track referensi yang diberikan. Grafik ini menunjukkan bahwa nilai masing-masing terlihat semakin besar. Dengan demand yang tidak dapat dipenuhi, eror yang dihasilkan semakin tinggi. Sehingga sistem menjadi tidak stabil.

Pada pengujian ini, MPC tidak mengurangi kecepatan mobil. Meskipun secara nalar, dengan mengurangi kecepatan dapat dihasilkan gerakan yang lebih baik dan lebih sesuai dengan bentuk track. Namun karena permasalahan pada penelitian ini adalah trajectory tracking, MPC juga memperhitungkan pencapaian posisi yang harus sesuai dengan waktu. Berbeda dengan permasalahan path tracking dimana waktu pencapaian posisi tidak ditentukan.

Sehingga dapat dikatakan bahwa MPC memiliki suatu batasan yaitu referensi tidak boleh memiliki demand di luar constraint. Jika dihubungkan dengan implementasi pada dunia nyata, maka gerakan mobil ini sangat berbahaya karena mobil bisa keluar dari lintasan yang ada. Terlebih lagi jika menggunakan kecepatan yang lebih tinggi. Sehingga akan lebih cocok jika terdapat suatu kontroler tingkat tinggi untuk pembentukan referensi permasalahan trajectory tracking dengan memperhitungkan referensi kecepatan yang cocok agar gerakan mobil tidak keluar rintangan.

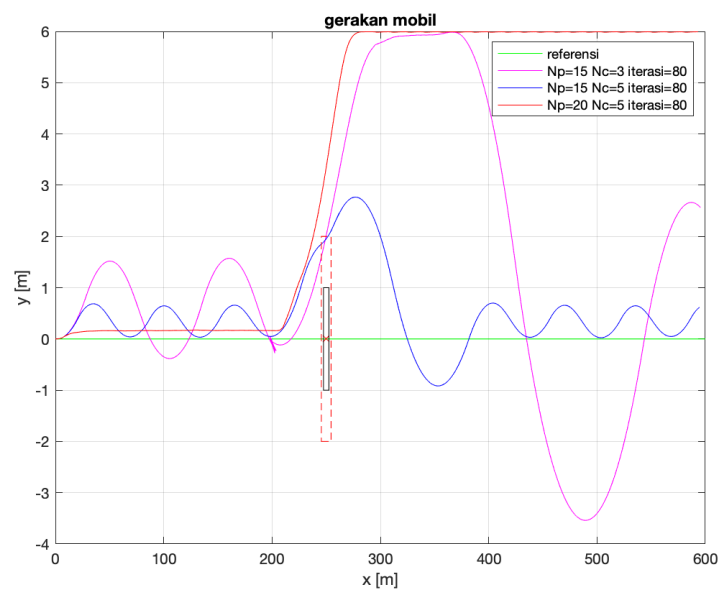
Performa MPC paling baik pada pengujian ke-4 dapat ditinjau dari 2 sisi yaitu kuantitatif dan kualitatif. Secara kuantitatif, performa terbaik dihasilkan oleh variasi iterasi maksimal quadratic programming= 40, $N_p=20$, dan $N_c=3$ dengan nilai RMSE posisi paling kecil yaitu 3.176 m. Secara kualitatif, performa terbaik dihasilkan oleh variasi iterasi maksimal quadratic programming= 40, $N_p=20$, dan $N_c=5$ karena menghasilkan gerakan menyerupai track referensi. Meskipun gerakan yang dihasilkan serupa dengan variasi $N_c=7$, dipilih $N_c=5$ merupakan yang terbaik karena secara kualitatif memiliki nilai eror yang lebih kecil yaitu RMSE posisi sebesar 4.584 m.

4.2 Obstacle Avoidance

Pengujian obstacle avoidance dibagi menjadi 3 bagian, yaitu menggunakan 1 halangan dengan kecepatan 10 m/s. bagian ke-2 yaitu menggunakan 1 halangan dengan kecepatan 20 m/s. pengujian bagian ke-3 menggunakan 2 halangan. Pada pengujian obstacle avoidance ditambahkan 1 variabel evaluasi yaitu jarak terdekat antara mobil dengan halangan. Pada pengujian menggunakan 1 halangan, ditentukan sebelumnya bahwa mobil akan menghindari menggunakan lajur kiri. Sementara pada pengujian menggunakan 2 halangan, ditentukan sebelumnya bahwa mobil menghindari dengan manuver yang paling dekat dengan track referensi.

4.2.1 1 Halangan dengan Kecepatan 10 m/s

Pengujian pertama adalah pengujian menggunakan 1 halangan dengan kecepatan 10 m/s. Sehingga diperoleh hasil seperti berikut



Gambar 4.5 Obstacle Avoidance dengan Kecepatan 10 m/s

dengan nilai eror, cost function, dan jarak terdekat antara mobil dengan halangan sebagai berikut.

Tabel 4.6 Obstacle Avoidance dengan Kecepatan 10 m/s

Iterasi Maksimal QP	Horizon Kontrol (Nc)	Horizon Prediksi (Np)	RMSE Posisi X	RMSE Posisi Y	RMSE Posisi	Cost Function	Waktu Komputasi	Jarak Terdekat
80	3	15	31.68	2.52	31.78	13825.08	0.05	2.33
80	5	15	4.45	0.96	4.55	252.96	0.05	1.99
80	5	20	4.62	4.51	6.46	739.03	0.05	3.34

Pada pengujian obstacle avoidance pertama, terlihat bahwa seluruh variasi menghasilkan gerakan osilasi. Hal ini mungkin dikarenakan kombinasi parameter yang belum cocok untuk track referensi garis lurus. Seperti yang dilihat pada pengujian trajectory tracking dengan track pergantian lajur, terdapat variasi yang menghasilkan gerakan osilasi saat sebelum dan sesudah

melakukan pergantian lajur. Maka sesuai analisa sebelumnya bahwa pemilihan kombinasi yang seimbang sangat penting.

Pada variasi dengan horizon prediksi (N_p)=15 dan horizon control (N_c)=3 terlihat bahwa mobil berhasil menghindari halangan. Hal ini ditunjukkan juga dengan nilai jarak terdekat antara mobil dengan halangan yaitu 2.33 m yang artinya sudah sesuai dengan zona aman. Namun, manuver penghindaran masih kurang efektif. Dapat dilihat bahwa setelah menghindari halangan, mobil membutuhkan waktu untuk kembali ke track referensi. Selain itu, pada variasi ini juga dihasilkan gerakan osilasi dengan amplitudo yang besar dan bertambah besar setelah mobil menghindari halangan. Sehingga diperoleh eror yang besar pada variasi ini. Hal ini bisa jadi dikarenakan jumlah iterasi yang belum cukup untuk mendapatkan hasil yang optimal.

Pada variasi dengan N_p =15 dan N_c =4 terlihat bahwa mobil tidak berhasil menghindari halangan. Hal ini ditunjukkan dengan jarak terdekat antara mobil dengan halangan sejauh 1.99 m yang kurang dari zona aman yaitu 2 m. Hal ini mungkin dikarenakan adanya osilasi saat mobil merubah sudut heading atau arah hadap mobil agar kembali paralel dengan jalur. Namun, dapat dikatakan bahwa variasi ini memiliki manuver yang paling efektif. Ditinjau dari jarak antara mobil dengan halangan yang tidak terlalu jauh ketika mobil bergerak kembali menuju track referensi.

Pada variasi dengan N_p =20 dan N_c =5 terlihat bahwa mobil berhasil menghindari halangan. Hal ini ditunjukkan dengan jarak terdekat sejauh 3.34 m. Selain itu, gerakan mobil juga mulus tanpa osilasi. Namun, setelah menghindari halangan mobil tidak kembali ke track referensi. Ada kemungkinan bahwa dengan variasi ini dibutuhkan waktu yang lebih lama dibandingkan waktu simulasi untuk kembali lagi ke track referensi. Meskipun begitu, terlihat bahwa posisi Y mobil tidak melebihi constraint yaitu $Y=6$. Artinya MPC dapat menjaga agar mobil tetap berada di dalam jalur yang ditentukan.

Pada $N_p=15$, terlihat bahwa nilai eror antara variasi dengan $N_c=3$ dan $N_c=5$ memiliki perbedaan yang signifikan. Dimana horizon kontrol (N_c) yang lebih besar menghasilkan cost function yang lebih kecil. Jika dilihat dari gerakan mobil yang dihasilkan, terlihat bahwa variasi $N_c=3$ menghasilkan gerakan osilasi dengan amplitudo yang jauh lebih besar saat kembali menuju track referensi. Gerakan osilasi ini, dapat menyebabkan perhitungan eror dan cost function menjadi semakin besar. Adapun pada $N_c=5$, terlihat bahwa peningkatan variasi horizon prediksi meningkatkan nilai eror dan nilai cost function. Hal ini sesuai dengan perumusan cost function yang menghitung eror antara referensi dengan prediksi output. Sehingga semakin besar horizon prediksi nilai cost function juga semakin besar.

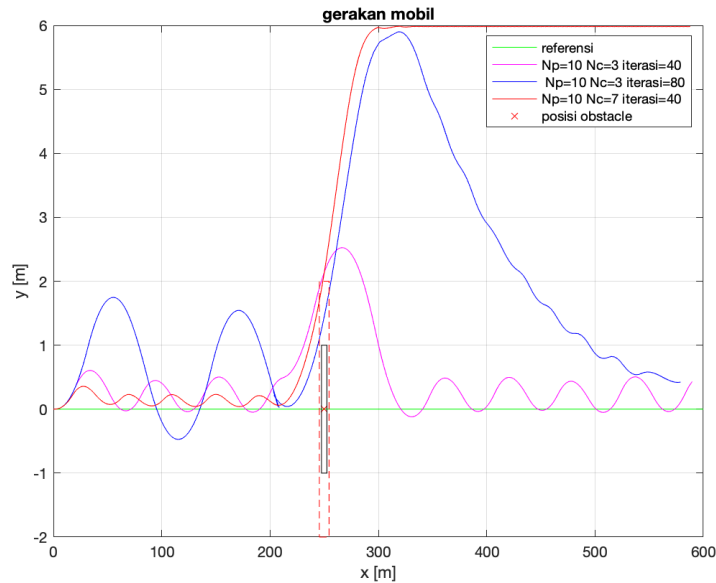
Dilihat dari waktu komputasi, sama seperti simulasi trajectory tracking, waktu komputasi MPC untuk memperhitungkan nilai input optimal tidak dipengaruhi dengan variasi yang dipilih. Dimana diperoleh waktu komputasi sebesar 0.05 s. Artinya kompleksitas antar variasi kombinasi parameter yang dipilih tidak begitu jauh untuk menghasilkan waktu komputasi yang berbeda.

Pada pengujian ini, performa MPC terbaik diperoleh pada variasi iterasi maksimal quadratic programming= 80, N_p =15, dan N_c =5 dengan nilai RMSE posisi 4.55 m dan jarak terdekat dengan halangan sejauh 1.99 m. Secara kualitatif, variasi ini merupakan yang paling baik karena memiliki titik putar balik yang tidak terlalu jauh dibandingkan dengan posisi halangan. Sehingga dapat dikatakan bahwa variasi ini memiliki manuver paling efektif

dibandingkan dengan variasi lainnya. Meskipun pada variasi ini zona aman tidak terpenuhi, dimana seharusnya jarak terdekat mobil seharusnya adalah 2 m.

4.2.2 1 Halangan dengan Kecepatan 20 m/s

Pengujian ke-2 yaitu menggunakan 1 halangan dengan kecepatan 20 m/s. Kemudian diperoleh hasil seperti berikut.



Gambar 4.6 Obstacle Avoidance dengan kecepatan 20 m/s

dengan nilai eror, cost function, dan jarak terdekat antara mobil dengan halangan sebagai berikut.

Tabel 4.7 Obstacle Avoidance dengan Kecepatan 20 m/s

Iterasi Maksimal QP	Horizon Kontrol (Nc)	Horizon Prediksi (Np)	RMSE Posisi X	RMSE Posisi Y	RMSE Posisi	Cost Function	Waktu Komputasi	Jarak Terdekat
40	3	10	7.96	0.77	8.00	641.81	0.05	2.16
40	7	10	10.90	4.37	11.74	1194.33	0.05	2.13
80	3	10	61.55	2.14	61.60	35527.79	0.05	1.99

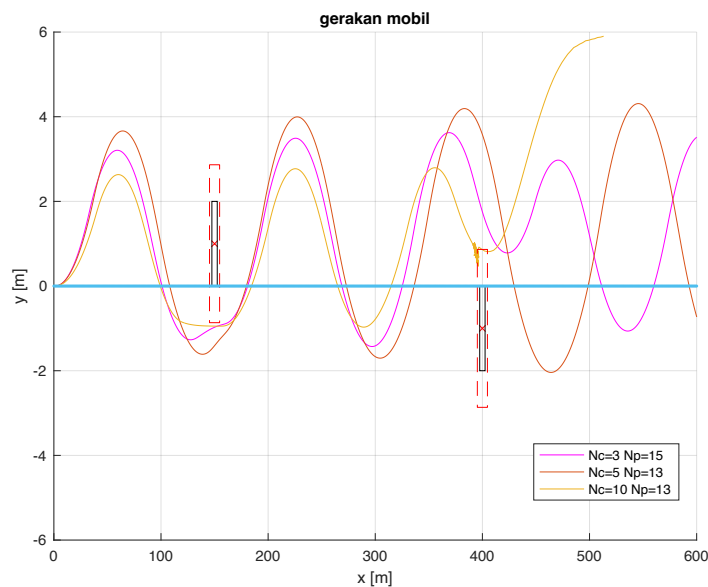
Sama seperti pengujian pertama. Seluruh variasi pada pengujian ini menghasilkan gerakan yang berhasil. Pada variasi Nc=7 dan iterasi maksimal sebanyak 40 kali, diperoleh bahwa MPC masih belum bisa menemukan nilai input yang dapat meminimalkan eror lateral dalam constraint yang ditentukan. Hal ini bisa jadi dikarenakan horizon kontrol (Nc) yang besar menghasilkan nilai cost function yang besar, sedangkan iterasi sebanyak 40 kali masih belum cukup untuk mendapatkan hasil yang optimal.

Pada kombinasi Np=10, Nc=7, dan iterasi maksimal quadratic programming= 40, terlihat bahwa mobil tidak bergerak kembali menuju trayektori referensi. Hal ini juga ditunjukkan oleh salah satu variasi pada pengujian sebelumnya dengan kecepatan 10 m/s. Hal ini bisa jadi dikarenakan horizon prediksi (Np) yang kurang jauh untuk mendapatkan hasil perhitungan yang baik ataupun bisa dikarenakan jumlah iterasi yang kurang banyak untuk mendapat hasil yang optimal.

Performa terbaik pada pengujian ini ditemukan pada variasi iterasi maksimal quadratic programming= 40, $N_c=3$, dan $N_p=10$ dengan nilai RMSE posisi 8.00 m dan jarak terdekat dengan obstacle sejauh 2.16 m. Variasi ini dipilih sebagai yang terbaik secara kualitatif dilihat dari manuver penghindaran yang efektif. Yaitu titik dimana mobil belok kembali menuju lajur tengah tidak jauh dari halangan.

4.2.3 2 Halangan dengan Kecepatan 10 m/s

Pengujian selanjutnya adalah pengujian dengan menggunakan 2 halangan. Pada pengujian ini diberikan jarak antar 2 halangan sejauh 250 m. Pemilihan manuver penghindaran sudah ditentukan sebelumnya yaitu ketika melewati halangan pertama mobil akan menghindari menggunakan lajur kanan dan ketika melewati halangan ke-2 mobil akan menghindari menggunakan lajur kiri. Sehingga diperoleh hasil seperti berikut



Gambar 4.7 Obstacle Avoidance dengan 2 Halangan

Adapun perbandingan nilai eror posisi, cost function, waktu komputasi dan jarak terdekat antara mobil dengan halangan adalah sebagai berikut

Tabel 4.8 Obstacle Avoidance dengan 2 Halangan

Iterasi Maksimal QP	Horizon Kontrol (N_c)	Horizon Prediksi (N_p)	RMSE Posisi X	RMSE Posisi Y	RMSE Posisi	Cost Function	Waktu Komputasi	Jarak Terdekat
80	3	15	3.537	1.972	4.050	244.148	0.05	2.018
80	5	13	4.527	2.345	5.098	303.758	0.05	2.442
80	10	13	266.401	1.408	266.405	874352.634	0.05	1.946

Pada pengujian obstacle avoidance dengan 2 halangan terbentuk Gerakan yang sama antar variasi yaitu beresilasi dengan amplitude yang besar. Artinya variasi ini memiliki kemampuan tracking garis lurus yang kurang baik.

Pada variasi dengan horizon prediksi (N_p)=15 dan horizon kontrol (N_c)=3, terlihat bahwa variasi ini berhasil menghindari kedua halangan. Manuver penghindaran halangan pertama terlihat cukup efektif, karena posisinya yang sangat dekat dengan zona aman. Pada halangan ke-2, mobil juga mampu menghindari namun manuvernya kurang efektif dibanding saat menghindari halangan pertama. Terlihat bahwa setelah menghindari halangan, mobil bergerak ke lajur kiri lagi meskipun tidak diperlukan. Hal ini mungkin dikarenakan horizon control yang kecil.

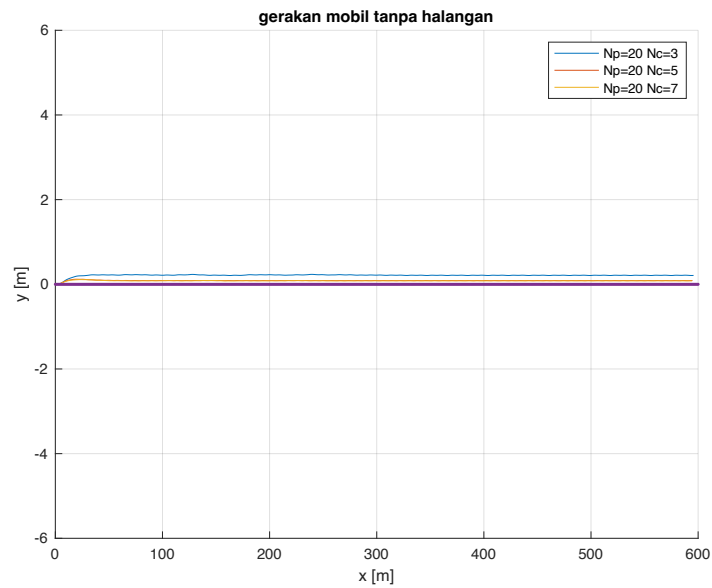
Pada variasi $N_p=13$ dan $N_c=5$, terlihat bahwa variasi ini berhasil menghindari kedua halangan. Jika dibandingkan dengan variasi pertama, gerakan yang dihasilkan cukup mirip namun dengan amplitudo osilasi yang lebih besar. Artinya, variasi 1 memiliki kemampuan tracking yang lebih baik. Gerakan manuver penghindaran halangan pertama juga cukup menyerupai namun dengan jarak mobil yang lebih jauh dengan halangan. Artinya variasi ini memiliki respon yang lebih lambat dibandingkan dengan variasi pertama. Respon lambat ini menyebabkan manuver penghindaran pada halangan ke-2 menjadi lebih efektif, yaitu tidak kembali ke lajur kiri setelah melewati halangan ke-2. Hal ini mungkin juga disebabkan jarak mobil yang lebih jauh dengan halangan ke-2.

Pada variasi $N_p=13$ dan $N_c=10$, terlihat bahwa variasi berhasil menghindari halangan pertama, namun pada halangan ke-2 mobil tidak berhasil menghindari. Hal ini dibuktikan juga dengan jarak terdekat yaitu 1.946 yang tidak sesuai dengan zona aman. Jika dilihat dari grafik, Gerakan mobil ini memiliki kemampuan tracking terbaik dengan amplitudo terkecil. Manuver penghindaran halangan pertama juga sangat efektif dimana jarak mobil selalu sesuai dengan zona aman. Mobil tidak memenuhi zona aman pada manuver penghindaran halangan ke-2 karena mobil berusaha melakukan tracking dan memiliki heading angle yang cukup curam, sehingga sulit untuk mengatur gerakan mobil agar sesuai dengan constraint secara tiba-tiba. Setelah menghindari halangan ke-2, mobil terlihat tidak kembali menuju track referensi, hal ini sama seperti beberapa variasi pada pengujian sebelumnya yang disebabkan oleh nilai N_c yang besar sehingga komputasi menjadi lebih kompleks.

Jika dibandingkan antara ketiga variasi dapat terlihat bahwa semakin baik kemampuan MPC dalam melakukan tracking maka manuver penghindaran semakin tidak efektif. Untuk itu dilakukan percobaan lebih lanjut menggunakan kombinasi parameter dengan kemampuan tracking yang lebih stabil. Yaitu dengan meningkatkan horizon prediksi (N_p)

4.2.4 2 Halangan dengan Horizon Prediksi Lebih Tinggi

Melanjutkan dari pengujian sebelumnya dengan 2 halangan, dilakukan pengujian dengan parameter horizon prediksi yang lebih tinggi untuk mendapatkan MPC dengan fungsi trajectory tracking yang lebih baik. Sehingga diperoleh hasil seperti berikut



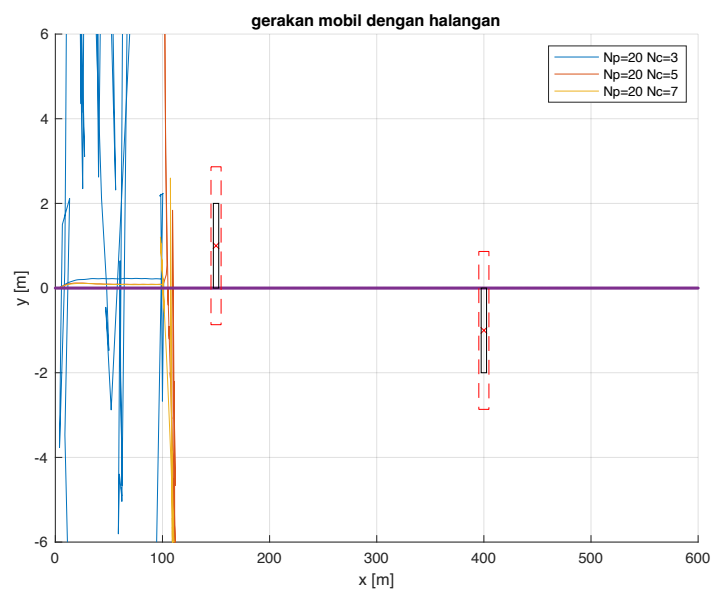
Gambar 4.8 Tracking Performance dengan Horizon Prediksi (N_p) 20

Dapat terlihat pada gambar 4.8 bahwa performa trajectory tracking menjadi lebih baik karena amplitudo osilasi lebih kecil dibandingkan dengan variasi pada pengujian sebelumnya. Performa tracking yang baik juga dibuktikan dengan nilai eror posisi pada tabel di bawah

Tabel 4.9 Tracking Performance dengan Horizon Prediksi (N_p) 20

Iterasi Maksimal QP	Horizon Kontrol (N_c)	Horizon Prediksi (N_p)	RMSE Posisi X	RMSE Posisi Y	RMSE Posisi	Cost Function	Waktu Komputasi
40	3	20	3.263	0.213	3.270	192.886	0.05
40	5	20	4.255	0.088	4.256	273.690	0.05
40	7	20	4.678	0.083	4.679	312.846	0.05

Kemudian dilakukan pengujian obstacle avoidance dengan 2 halangan menggunakan kombinasi parameter tersebut dan diperoleh hasil seperti berikut



Gambar 4.9 Obstacle Avoidance dengan 2 Halangan dan $N_p=20$

Terlihat bahwa seluruh variasi tidak berhasil menghindari halangan yang pertama. Pada jarak mobil sesuai dengan jarak deteksi, gerakan mobil menjadi tidak stabil. Hal ini dikarenakan gerakan mobil yang berosilasi dengan frekuensi tinggi meski dengan amplitudo yang lebih kecil. Sehingga perhitungan optimisasi menjadi lebih sulit.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Hasil pengujian dan analisa data simulasi mengenai trajectory tracking dan obstacle avoidance mobil otonomus dengan menggunakan Model Predictive Control diperoleh beberapa kesimpulan yaitu:

1. Pada pengujian trajectory tracking dengan track referensi sinusoidal diperoleh bahwa performa terbaik pada kecepatan 10 m/s diperoleh dengan variasi iterasi maksimal quadratic programming=40, $N_c=3$, dan $N_p=20$ dengan nilai RMSE posisi $X= 3.25$ m, RMSE posisi $Y= 0.41$ m, RMSE posisi= 3.28 m , dan rata-rata cost function= 192.77 sementara pada kecepatan 20 m/s diperoleh pada variasi iterasi maksimal quadratic programming=80, $N_c=3$, dan $N_p=15$ dengan nilai RMSE posisi $X= 7.03$ m, RMSE posisi $Y= 0.63$ m, RMSE posisi= 7.06 m , dan rata-rata cost function= 698.11.
2. Pada pengujian trajectory tracking dengan track pergantian lajur diperoleh bahwa gerakan mobil terbaik secara kualitatif terdapat pada variasi iterasi maksimal quadratic programming=40, $N_p=20$, dan $N_c=5$ dengan nilai RMSE posisi 4.256 m.
3. Pada pengujian obstacle avoidance dengan 1 halangan diperoleh bahwa performa terbaik yang menghasilkan manuver penghindaran paling efektif pada kecepatan 10 m/s diperoleh pada variasi iterasi 80, $N_c=5$, dan $N_p=15$ dengan nilai RMSE posisi 4.55 meter dan jarak terdekat dengan obstacle sejauh 1.99 meter. Sementara pada kecepatan 20 m/s diperoleh pada variasi iterasi 40, $N_c=3$, dan $N_p=10$ dengan nilai RMSE posisi 8.00 meter dan jarak terdekat dengan obstacle sejauh 2.16 meter.
4. Berdasarkan permasalahan trajectory tracking, MPC memiliki delay pencapaian posisi selama 0.4 s.
5. MPC memiliki batasan bahwa referensi yang digunakan tidak boleh memiliki demand di luar batasan atau di luar constraint yang telah ditentukan.
6. Pemilihan kombinasi parameter iterasi maksimal, horizon prediksi, dan horizon control dari MPC mempengaruhi respon dan performa MPC dalam melakukan trajectory tracking dan obstacle avoidance.
7. Diperlukan pemilihan kombinasi antar parameter MPC yang seimbang untuk mendapatkan performa MPC yang baik.
8. Perubahan parameter N_p mempengaruhi performa MPC dimana semakin besar nilai N_p maka semakin besar eror dan cost function yang dihasilkan. Begitu pula dengan perubahan parameter N_c .
9. Semakin bagus kemampuan tracking manuver penghindaran halangan semakin tidak efektif untuk halangan lebih dari 1.
10. LTV MPC dengan model linear pada penelitian ini dapat menghasilkan gerakan yang menyerupai trayektori referensi.
11. Dengan merumuskan obstacle sebagai constraint, MPC mampu menghindari obstacle meskipun dengan pengaturan gerakan yang berlebihan. Karena MPC sangat mengutamakan constraint dalam perhitungan nilai optimal, nilai hasil perhitungan terkadang tidak menghasilkan gerakan mobil yang dianggap bagus untuk implementasi dunia nyata.

5.2 Saran

Penelitian yang telah dilakukan penulis masih memiliki banyak kekurangan. Oleh karena itu untuk pengembangan penelitian selanjutnya dengan topik yang sama penulis ingin memberikan saran yaitu:

1. Penggunaan parameter dan constraint mobil sesuai referensi nyata, untuk mendapatkan hasil simulasi yang lebih akurat.
2. Pemodelan jalur dan halangan yang lebih kompleks, mereferensikan dari dunia nyata agar lebih menggambarkan kondisi nyata.
3. Penggunaan sensor untuk obstacle detection.
4. Menggunakan koordinat posisi relatif terhadap jalan (*frenet serret frame* atau *curve linear frame*) untuk meningkatkan akurasi posisi lateral.
5. Implementasi konfigurasi feedforward untuk mengatasi keterlambatan pencapaian posisi.
6. Diperlukannya pencarian kombinasi parameter MPC lain yang dapat menghasilkan performa lebih baik.
7. Percobaan dengan nilai δ yang lebih kecil pada algoritma Hildreth untuk memastikan nilai hasil optimisasi adalah nilai yang optimal.
8. Memperhitungkan adanya gangguan eksternal ataupun gangguan pengukuran dalam simulasi karena pada penelitian ini masih diasumsikan sistem tidak memiliki gangguan.
9. Melakukan pengujian sistem dengan implementasi secara nyata untuk mendapatkan hasil secara langsung mengenai sistem fisik.

DAFTAR PUSTAKA

- Abel, D., & Katriniok, A. (2011). *LTV-MPC Approach for Lateral Vehicle Guidance by Front Steering at the Limits of Vehicle Dynamics*.
- Babak, S. J., Hussain, S. A., Karakas, B., & Cetin, S. (2017). Control of autonomous ground vehicles: A brief technical review. *IOP Conference Series: Materials Science and Engineering*, 224(1). <https://doi.org/10.1088/1757-899X/224/1/012029>
- Bagyaveereswaran, V., Mathur, T. D., Gupta, S., & Arulmozhivarman, P. (2016). Performance comparison of next generation controller and MPC in real time for a SISO process with low cost DAQ unit. *Alexandria Engineering Journal*, 55(3), 2515–2524. <https://doi.org/10.1016/j.aej.2016.07.028>
- Bimbraw, K. (2015). Autonomous cars: Past, present and future: A review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings, 1*, 191–198. <https://doi.org/10.5220/0005540501910198>
- Charan, S., & Srour, A. (2013). *Self Driving Cars*. www.funginstitute.berkeley.edu
- Corke, P. (2017). *Robotics, Vision and Control* (1st ed., Vol. 118). Springer International Publishing. <https://doi.org/10.1007/978-3-319-54413-7>
- Faulwasser, T. (2012). *NMPC for Tracking and Path Following* [Dissertation, Guericke Universität]. <https://d-nb.info/1054135541/34>
- Halim, B. J. (2021). *Perancangan Sistem Autopilot dan Guidance Berbasis Model Predictive Control Sliding Curve pada Unmanned Surface Vehicle dengan Gangguan*.
- Hatem, A. (2018). *model predictive control for path tracking and obstacle avoidance of autonomous vehicle*.
- Lam, V. T. T., Sattar, A., Wang, L., & Lazar, M. (2020). Fast Hildreth-based Model Predictive Control of Roll Angle for a Fixed-Wing UAV. *IFAC-PapersOnLine*, 53(2), 5757–5763. <https://doi.org/10.1016/j.ifacol.2020.12.1608>
- Li, J., Ran, M., Wang, H., & Xie, L. (2019). MPC-based Unified Trajectory Planning and Tracking Control Approach for Automated Guided Vehicles. *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, 374–380. <https://doi.org/10.1109/ICCA.2019.8899955>
- Mathworks. (2018). *Adaptive MPC*. <https://www.mathworks.com/help/mpc/ug/adaptive-mpc.html>
- Ogata, Katsuhiko. (2010). *Modern control engineering*. Prentice-Hall.
- Ravi Tailor, M., & Bhathawala, P. H. (2011). Linearization of Nonlinear Differential Equation by Taylor's Series Expansion and Use of Jacobian Linearization Process. *International Journal of Theoretical and Applied Science*, 4(1), 36–38. https://researchtrend.net/ijtas/ijtas_2012/10%20KRUTI%20TAILOR.pdf

- Rossiter, J. A. (2018). *A First Course in Predictive Control Second Edition*.
- Santoso, A., Amirudin, B. P., & Kadir, R. E. A. (2021). Pengaturan Kemudi Kendaraan Otonom Four Wheel Steer dan Four Wheel Drive (4WS4WD) Menggunakan Model Predictive Control. *Jurnal Teknik ITS*, 10(1). <http://ejournal.its.ac.id/index.php/teknik/article/view/59574>
- Turri, V., Carvalho, A., Tseng, H. E., Johansson, K. H., & Borrelli, F. (2013). Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 378–383. <https://doi.org/10.1109/ITSC.2013.6728261>
- Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB®*. Springer London. <https://doi.org/10.1007/978-1-84882-331-0>

LAMPIRAN

1. Inisialisasi Parameter Trajectory Tracking dan Obstacle Avoidance

```
%% PARAMETER MPC
% horizon prediksi & kontrol
Np=20
Nc=3
% bobot cost function
R=0.6; % input
Q=0.4; % eror
% iterasi maksimal QP
iter=40;
% sampling time
Ts=0.1;
```

2. Pembentukan Lingkungan Simulasi Trajectory Tracking dan Obstacle Avoidance

```
%% INFO MOBIL DAN JALUR
% info jalur
lanes=3
laneWidth=4
% dimensi mobil
carLength=4;
carWidth=2;
```

3. Pembentukan Constraint Trajectory Tracking dan Obstacle Avoidance

```
%% CONSTRAINT INISIAL
% constraint input
vmin=kmph2mps(30); % 30 kmph
vmax=kmph2mps(100); % 100 kmph
omegamin=-60*pi/180; % rad/s
omegamax=60*pi/180;
% constraint output
xmin=0;
xmax=inf;
ymin=-laneWidth*lanes/2;
ymax=laneWidth*lanes/2;
thetamin=-90*pi/180; % rad
thetamax=90*pi/180;
psimin=-10*pi/180*Ts;
Umin=[vmin;psimin]
Umax=[vmax;psimax]
Ymin=[xmin;ymin;thetamin;psimin]
Ymax=[xmax;ymax;thetamax;psimax]

% menggabungkan constraint menjadi 1 matriks
Ucons=[Umin;Umax]; % matriks constraint input
Ycons=[Ymin;Ymax]; % matriks constraint output
```

4. Pembentukan Referensi Trajectory Tracking

```
%% PEMBENTUKAN REFERENSI TRAJECTORY TRACKING
vref=10; % m/s
simtime=60;
```

```

% referensi posisi x untuk sinyal sinusoidal
ref(:,1)=0:vref*Ts:vref*simtime;
% referensi posisi y untuk sinyal sinusoidal
ref(:,2)=3*sin(ref(:,1)*2*pi/60*Ts); % pengujian trajectory tracking 1
ref(:,2)=3*sin(ref(:,1)*9*pi/60*Ts); % pengujian trajectory tracking 2

% referensi posisi y untuk pergantian lajur
f = @(x) 1/0.75*atan((x-250)/20)+2;
for i=1:length(ref)
    ref(i,2)=f(ref(i,1));
End

```

5. Pembentukan Referensi Obstacle Avoidance

```

%% PEMBENTUKAN REFERENSI OBSTACLE AVOIDANCE
% referensi posisi x
ref(:,1)=0:vref*Ts:vref*simtime;
% referensi posisi y
ref(:,2)=0;

```

6. Kondisi Inisial Trajectory Tracking dan Obstacle Avoidance

```

%% KONDISI INISIAL MOBIL
% initial state
x0=[ref(1,1);ref(1,2);0;0];
x=x0;
x_old=x;
% initial control
u=[0;0];
u_old=u;
% initial discrete state space model
[Ap,Bp,Cp]=ackermann_modelDT_lin(Ts,x,u,carLength);
% initial MPC internal model
sizex=size(Ap,2);
sizeu=size(Bp,2);
sizey=size(Cp,1);
[Phi_Phi,Phi_F,Phi_R,F,Phi] =mpcgain_alya_real(Ap,Bp,Cp,Nc,Np);
% nilai inisial untuk simulasi
Xf=zeros(size(Bp,1),1); % initial state feedback
initPose = x; % initial car position

```

7. Perumusan Cost Function Trajectory Tracking dan Obstacle Avoidance

```

% PERUMUSAN COST FUNCTION
% bobot kecepatan= 0.3, bobot steering rate= 0.7
w=[0.3 0.7];
wu=[];
for i=1:Nc
    wu=[wu,w];
end
wu=diag(wu);
% pembentukan matriks H cost function
H=2*(Phi_Phi+R*wu);

```


8. Loop Simulasi Trajectory Tracking

```
for kk=1:length(ref)
    tic
%% PERHITUNGAN REFERENSI HEADING ANGLE DAN STEERING ANGLE
% membandingkan posisi x,y mobil dengan referensi x,y
sp=ref(kk,:);
n=1;
while x(1)>sp(1) && kk+n<=length(ref)
    sp=ref(kk+n,:);
    n=n+1;
    if n>length(ref)
        break
    end
end
ex=(x(1)-sp(1));
ey=(x(2)-sp(2));
% menghitung referensi heading angle
thetaref=atan(ey/ex);
if ey==0 || isnan(thetaref)
    thetaref=0;
end
ref(kk,3)=thetaref;
% menghitung referensi steering angle
ref(kk,4)=atan(thetaref*carLength/vref);

%% UPDATE INTERNAL MODEL MPC
[Ap,Bp,Cp] = ackermann_modelDT_lin(Ts,x,u,carLength);
[Phi_Phi,Phi_F,Phi_R,F,Phi,BarRs]
=mpcgain_alya_real(Ap,Bp,Cp,Nc,Np); % tanpa augmentasi

%% PEMBENTUKAN MATRIKS CONSTRAINT M DAN N
[M,N,sizeM,sizeN]=combined_constraint_real(Ucons,Ycons,ymin,ymax,Phi,F,
Xf,Nc,Np);

%% OPTIMIZER
% pembentukan matriks f cost function
wx=1; % bobot output
f=-4*Q*(Phi_R*ref(kk,:)'-Phi_F*Xf;
% perhitungan hildreth quadratic programming
eta=QPhild(H,f,M,N,iter,kk,[],1e-8);
deltaU=eta(1:sizeU,1);

%% WAKTU KOMPUTASI SIYAL KONTROL
t(kk)=toc;

%% UPDATE SINYAL KONTROL
u_old=u;
u=deltaU;

%% MEMASUKAN SINYAL KONTROL KE DALAM PLANT
mobil = ackermannKinematics("WheelBase",carLength);
```

```

    tspan = 0:Ts/2:Ts; % 1 step = (2)+1 baris
    inputs = [u(1) u(2)]; % ackermann: [velocity(v) steering
rate(psidot = omega)]
    [t,y] = ode45(@(t,y)derivative(mobil,y,inputs),tspan,initPose); %
plant ackermann

%% UPDATE POSISI MOBIL
initPose=y(size(y,1),:);

% UPDATE STATE DAN STATE FEEDBACK
x_old=x;
x=initPose';
Xf=x; % tanpa augmentasi

%% MENYIMPAN DATA UNTUK PLOT
u1(:,kk)=u; % sinyal input
x1(:,kk)=x; % state
time(kk)=Ts*kk; % waktu simulasi

%% MELANJUTKAN ITERASI SIMULASI
kk=kk+1;
end

```

9. Pembentukan Halangan Untuk Obstacle Avoidance

```

%% PEMBENTUKAN HALANGAN
% obstacle1 info
obstacle = struct;
obstacle.Length = 5;
obstacle.Width = 2;
obstacle.X = 150;
obstacle.Y = 1;
obstacle.safeDistanceX = 1/2*(obstacle.Length+carLength);
obstacle.safeDistanceY = 1/2*(obstacle.Width+carWidth);
obstacle.DetectionDistance = 50;
obstacle = GenerateObstacle(obstacle);

% obstacle2 info
obstacle2 = struct;
obstacle2.Length = 5;
obstacle2.Width = 2;
obstacle2.X = 400;
obstacle2.Y = -1;
obstacle2.safeDistanceX = 1/2*(obstacle.Length+carLength);
obstacle2.safeDistanceY = 1/2*(obstacle.Width+carWidth);
obstacle2.DetectionDistance = 50;
obstacle2 = GenerateObstacle(obstacle2);

```

Loop simulasi obstacle avoidance

```

for kk=1:length(ref)
    tic
%% PERHITUNGAN REFERENSI HEADING ANGLE DAN STEERING ANGLE
% membandingkan posisi x,y mobil dengan referensi x,y
sp=ref(kk,:);

```

```

n=1;
while x(1)>sp(1) && kk+n<=length(ref)
    sp=ref(kk+n,:);
    n=n+1;
    if n>length(ref)
        break
    end
end
ex=(x(1)-sp(1));
ey=(x(2)-sp(2));
% menghitung referensi heading angle
thetaref=atan(ey/ex);
if ey==0 || isnan(thetaref)
    thetaref=0;
end
ref(kk,3)=thetaref;
% menghitung referensi steering angle
ref(kk,4)=atan(thetaref*carLength/vref);

%% UPDATE INTERNAL MODEL MPC
[Ap,Bp,Cp] = ackermann_modelDT_lin(Ts,x,u,carLength);
[Phi_Phi,Phi_F,Phi_R,F,Phi,BarRs]
=mpcgain_alya_real(Ap,Bp,Cp,Nc,Np);

%% DETEKSI HALANGAN
% deteksi obstacle1
[detection, dist20bstacle(1,kk)] = obstacleDetect(x, laneWidth,
obstacle);
% deteksi obstacle2
[detection2, dist20bstacle(2,kk)] = obstacleDetect(x, laneWidth,
obstacle2);

%% MENGHITUNG CONSTRAINT POSISI Y BARU
% manuver menghindari obstacle1
[slopemax,ymax] = newyConstraintbwh(x,detection,laneWidth, lanes,
obstacle, ref(kk+1,:));
% manuver menghindari obstacle2
[slopemin,ymin] = newyConstraint(x,detection2,laneWidth, lanes,
obstacle2, ref(kk+1,:));

%% PEMBENTUKAN MATRIKS CONSTRAINT M DAN N
[M,N,sizeM,sizeN]=combined_constraint_real(Ucons,Ycons,ymin,ymax,Phi,F,
Xf,Nc,Np);

%% OPTIMIZER
% pembentukan matriks f cost function
wx=1; % bobot output
f=-4*Q*(Phi_R*ref(kk,:)'-Phi_F*Xf;
% perhitungan hildreth quadratic programming
eta=QPhild(H,f,M,N,iter,kk,[],1e-8);
deltaU=eta(1:sizeU,1);

%% WAKTU KOMPUTASI SINYAL KONTROL
t(kk)=toc;

```

```

%% UPDATE SINYAL KONTROL
u_old=u;
u=deltaU;

%% MEMASUKAN SINYAL KONTROL KE DALAM PLANT
mobil = ackermannKinematics("WheelBase",carLength);
tspan = 0:Ts/2:Ts; % 1 step = (2)+1 baris
inputs = [u(1) u(2)]; % ackermann: [velocity(v) steering
rate(psidot = omega)]
[t,y] = ode45(@(t,y)derivative(mobil,y,inputs),tspan,initPose); %
plant ackermann

%% UPDATE POSISI MOBIL
initPose=y(size(y,1),:);

% UPDATE STATE DAN STATE FEEDBACK
x_old=x;
x=initPose';
Xf=x; % tanpa augmentasi

%% MENYIMPAN DATA UNTUK PLOT
u1(:,kk)=u; % sinyal input
x1(:,kk)=x; % state
time(kk)=Ts*kk; % waktu simulasi

%% MELANJUTKAN ITERASI SIMULASI
kk=kk+1;
end

```

10. Matlab Function Pembentukan Obstacle

```

%% PEMBENTUKAN OBSTACLE
function obstacle = GenerateObstacle(obstacle)
% Generate obstacle and safe zone geometry.

%Obstacle:
% Front left
obstacle.flX = obstacle.X+obstacle.Length/2;
obstacle.flY = obstacle.Y+obstacle.Width/2;
% Front right
obstacle.frX = obstacle.X+obstacle.Length/2;
obstacle.frY = obstacle.Y-obstacle.Width/2;
% Rear left
obstacle.rlX = obstacle.X-obstacle.Length/2;
obstacle.rlY = obstacle.flY;
% Rear right
obstacle.rrX = obstacle.X-obstacle.Length/2;
obstacle.rrY = obstacle.frY;

%Safe zone:
% Front left

```

```

obstacle.flSafeX = obstacle.X+obstacle.safeDistanceX;
obstacle.flSafeY = obstacle.Y+obstacle.safeDistanceY;
% Front right
obstacle.frSafeX = obstacle.X+obstacle.safeDistanceX;
obstacle.frSafeY = obstacle.Y-obstacle.safeDistanceY;
% Rear left
obstacle.rlSafeX = obstacle.X-obstacle.safeDistanceX;
obstacle.rlSafeY = obstacle.flSafeY;
% Rear right
obstacle.rrSafeX = obstacle.X-obstacle.safeDistanceX;
obstacle.rrSafeY = obstacle.frSafeY;

```

11. Matlab Function Pembentukan Plant Diskrit

```

%% MODEL DISKRIT LINEAR PLANT MOBIL
function [Ad,Bd,Cd,Dd,U,Y,X,DX] =
ackermann_modelDT_lin(Ts,x,u,carLength)
theta=x(3);
psi=x(4);
v=u(1);
L=carLength;

Ac = [0 0 -v*sin(theta) 0
      0 0 v*cos(theta) 0
      0 0 0 v/L*(1+tan(psi)^2)
      0 0 0 0];

Bc = [cos(theta) 0
      sin(theta) 0
      tan(psi)/L 0
      0 1];

Cc = eye(4);
Dc = zeros(4,2);
% Generate discrete-time model using ZOH.
nx = size(Ac,1);
nu = size(Bc,2);
M = expm( [[Ac Bc]*Ts; zeros(nu,nx+nu)]);
Ad = M(1:nx,1:nx);
Bd = M(1:nx,nx+1:nx+nu);
Cd = Cc;
Dd = Dc;
% Nominal conditions for discrete-time plant
X = x;
U = u;
Y = x;
DX = Ad*x+Bd*u-x;

```

12. Matlab Function Model Internal MPC

```

%% MODEL INTERNAL MPC

```

```

function [Phi_Phi,Phi_F,Phi_R,F,Phi,BarRs]
=mpcgain_alya_real(Ap,Bp,Cp,Nc,Np)

[m1,n1]=size(Cp);
[n1,n_in]=size(Bp);
% m1= jumlah output; n1= jumlah state; n_in= jumlah input

P=Cp;
nx=size(Ap,1);
nB=size(Bp,2);
nC = size(Cp,1);
L=[];

for i=1:Np;
    Puterm = P*Bp;

    for j=1:Nc;
        vrow=(i-1)*nC+(j-1)*nC+1:(i-1)*nC+(j-1)*nC+1+(nC-1);
        if nB == 1
            vcol=j;
        else
            vcol=(j-1)*nB+1:(j-1)*nB+1+(nB-1);
        end
        Pu(vrow,vcol)=Puterm;
    end

    P=P*Ap;
    vrowF=(i-1)*nC+1:i*nC;
    Px(vrowF,1:nx) = P;
    L=[L;eye(nC)];
end

F=Px;           % CA...
Phi=Pu(1:Np*nC,:); % CAB..

I=eye(nC);
% data vector
BarRs=[I];
for i=2:Np
    BarRs=[BarRs,I];
end

Phi_Phi= Phi'*Phi;
Phi_F= Phi'*F;
Phi_R=Phi'*BarRs;

```

13. Matlab Function Deteksi Halangan

```
%% DETEKSI HALANGAN
```

```

function [detection, dist2Obstacle] = obstacleDetect(x, laneWidth,
obstacle)
% Detect when the vehicle sees an obstacle.
%#codegen
egoX = x(1);
egoY = x(2);
dist2Obstacle = sqrt( (obstacle.X - egoX)^2 + (obstacle.Y -
egoY)^2 );
flagCloseEnough = (dist2Obstacle < obstacle.DetectionDistance);
flagInLane = ( abs(obstacle.Y - egoY) < 2*laneWidth );
detection = ( flagCloseEnough && (egoX < obstacle.frSafeX) &&
flagInLane );

```

14. Matlab Function Perhitungan Constraint Posisi Y Baru untuk Penghindaran ke Kanan

```

%% PEMBENTUKAN CONSTRAINT POSISI Y BARU MANUEVER PENGHINDARAN KANAN
function [constraintSlope, ymax] =
newyConstraintbwh(x, detection, laneWidth, lanes, obstacle, ref)
egoX=x(1);
egoY=x(2);
refx=ref(1);
refy=ref(2);
if detection
    slope = ( (obstacle.rrSafeY - egoY)/(obstacle.rrSafeX - egoX) );
    fun = @(x) slope*(x-egoX)+egoY;
    if (egoX<=obstacle.rrSafeX) % sebelum melewati halangan
        if (egoY<obstacle.rrSafeY) % di bawah halangan
            constraintSlope = 0;
            ymax = obstacle.rrSafeY;
        else
            constraintSlope=slope;
            fun = @(x) slope*(x-egoX)+egoY;
            ymax = fun(refx);
            while ymax<obstacle.rrSafeY
                ymax = obstacle.rrSafeY;
            end
        end
    elseif ( (egoX>obstacle.rrSafeX) && (egoX<=obstacle.frSafeX) ) %
paralel di bawah halangan
        constraintSlope = 0;
        ymax = obstacle.rrSafeY;
    else % setelah melewati halangan
        constraintSlope = 0;
        ymax=laneWidth*lanes/2;
    end
else % tidak ada halangan
    constraintSlope = 0;
    ymax=laneWidth*lanes/2;
end

```

15. Matlab Function Perhitungan Constraint Posisi Y untuk Penghindaran ke Kiri

```

%% PEMBENTUKAN CONSTRAINT POSISI Y BARU MANUVER PENGHINDARAN KIRI
function [constraintSlope,ymin] = newyConstraint(x,detection, laneWidth,
lanes, obstacle, ref)
egoX=x(1);
egoY=x(2);
refx=ref(1);
refy=ref(2);
if detection
    slope = ( (obstacle.rlSafeY - egoY)/(obstacle.rlSafeX - egoX) );
    fun= @(x) slope*(x-egoX)+egoY;
    if (egoX<=obstacle.rlSafeX) % sebelum melewati halangan
        if (egoY>obstacle.rlSafeY) % di atas obstacle
            constraintSlope = 0;
            ymin = obstacle.rlSafeY;
        else
            constraintSlope=slope;
            fun= @(x) slope*(x-egoX)+egoY;
            ymin = fun(refx);
            while ymin>obstacle.rlSafeY
                ymin = obstacle.rlSafeY;
            end
        end
    elseif ( (egoX>obstacle.rlSafeX) && (egoX<=obstacle.flSafeX) ) %
paralel di atas halangan
        constraintSlope = 0;
        ymin = obstacle.rlSafeY;
    else % sudah melewati halangan
        constraintSlope = 0;
        ymin=-laneWidth*lanes/2;
    end
else % tidak ada halangan
    constraintSlope = 0;
    ymin=-laneWidth*lanes/2;
end
end

```

16. Matlab Function Quadratic Programming Hildreth

```

%% QUADRATIC PROGRAMMING HILDRETH
function eta=QPhild(H,f,A_cons,b,iter,kk,eta0,epsilon);
[n1,m1]=size(A_cons);
eta=eta0;
if isempty(eta0)
    eta=-H\f; % initial cost, global optimum
end
kh=0;

for i=1:n1
    if (A_cons(i,:)*eta>b(i)) % cek constraint
        kh=kh+1;
    else
        kh=kh+0;
    end
end
end

```



```

if (kh==0)
    return;
end

P=A_cons*(H\A_cons');
d=(A_cons*(H\f)+b);
[n,m]=size(d);
x_ini=zeros(n,m); % initial lambda
lambda=x_ini;
al=10;
for km=1:iter
% find the elements in the solution vector one by one
% km could be larger if the Lagranger multiplier has a slow
% convergence rate.
lambda_p=lambda;
for i=1:n
    w= P(i,:)*lambda-P(i,i)*lambda(i,1);
    w=w+d(i,1);
    la=-w/P(i,i);
    lambda(i,1)=max(0,la);
end
al=(lambda-lambda_p)'*(lambda-lambda_p);
if (al<epsilon);
    break;
end
end
end

eta=-H\f - H\A_cons'*lambda;

```

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis dilahirkan di Jakarta, 14 November 2000. Merupakan anak ke-2 dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu TK dan SD di Sekolah Islam Al-Jannah, SMP Al-Azhar Pusat, dan SMA Labschool Rawamangun. Setelah lulus dari SMA, penulis mengikuti SBMPTN dan diterima di Departemen Teknik Elektro FTEIC-IST pada tahun 2018 dan terdaftar dengan NRP 07111840000107.

Di Departemen Teknik Elektro penulis sempat aktif di suatu unit kegiatan mahasiswa VSNMC sebagai staff bendahara, menghadiri beberapa kegiatan Seminar yang diselenggarakan oleh Departemen, Himpunan Mahasiswa Teknik Elektro (HME) dan aktif sebagai Asisten laboratorium Sistem dan Sibernetika.