

**Kendali Prediksi Berbasis Model untuk Robot Beroda Tipe
Holonomic Tiga Roda**

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

Anas Iqbal Lutfika

16524001

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2020

LEMBAR PENGESAHAN

KENDALI PREDIKSI BERBASIS MODEL UNTUK ROBOT BERODA TIPE *HOLONOMIC* TIGA RODA

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Anas Iqbal Lutfika
16524001

Yogyakarta, 28 Juni 2021

Menyetujui,

Pembimbing


R M Sisdarmanto Adinandra, Ph.D.

025240101

LEMBAR PENGESAHAN

SKRIPSI

KENDALI PREDIKSI BERBASIS MODEL UNTUK ROBOT BERODA TIPE

HOLONOMIC TIGA RODA

Dipersiapkan dan disusun oleh:

Anas Iqbal Lutfika

16524001

Telah dipertahankan di depan dewan penguji

Pada tanggal: 9 Juli 2021

Susunan dewan penguji

Ketua Penguji : R M Sisdarmanto Adinandra, Ph.D.

Anggota Penguji 1: Almira Budiyo, S.Si., M.Eng.

Anggota Penguji 2: Elvira Sukma Wahyuni, S.Pdt., M.Eng.

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 9 Juli 2021

Ketua Program Studi Teknik Elektro



Yusuf Aziz Amrullah, Ph.D.

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 28 Juni 2021



Anas Iqbal Lutfika

KATA PENGANTAR

Alhamdulillah, Puji syukur penulis panjatkan hanya kepada Allah SWT yang telah memberikan segala rahmat-Nya, sehingga penulis dapat menyelesaikan tugas akhir skripsi berjudul “*KENDALI PREDIKSI BERBASIS MODEL UNTUK ROBOT BERODA TIPE HOLONOMIC TIGA RODA*” dengan baik. Tugas akhir ini dilakukan sebagai syarat wajib mahasiswa Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia untuk menyelesaikan jenjang pendidikan strata 1.

Selama melakukan penelitian dan penulisan laporan tugas akhir ini, penulis banyak menerima dukungan dan bantuan sehingga penelitian dan penulisan laporan dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terimakasih kepada:

1. Bapak Sisdamanto Adinandra, Ph.D. Selaku dosen pembimbing yang telah memberikan bimbingan dan arahan sehingga penelitian dan penulisan dapat diselesaikan dengan baik.
2. Bapak Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D. Selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Seluruh dosen serta karyawan Jurusan Teknik Elektro UII atas segala ilmu yang diajarkan kepada penulis sehingga penulis dapat menyelesaikan penelitian dan laporan tugas akhir.
4. Orang tua saya, Bapak Bakti Sulistyanto dan Ibu Isti Pamularsih. Yang selalu memberikan dukungan, semangat dan doa untuk kelancaran penelitian yang dilakukan.
5. Teman-teman mahasiswa Jurusan Teknik Elektro, Universitas Islam Indonesia Angkatan 2016.
6. Seluruh pihak yang terlibat yang telah memberikan masukan dan semangat dalam menyelesaikan Laporan Skripsi ini.

Semoga skripsi ini dapat bermanfaat bagi semua orang maupun bagi penulis sendiri, *aamiin ya rabbalalamiin*.

Wassalamu' alaikum Warahmatullahi Wabarakaatuh.

Yogyakarta, 28 Juni 2021

Anas Iqbal Lutfika

ABSTRAK

Penelitian yang dilakukan berfokus kepada penggunaan kendali prediksi berbasis model (MPC) untuk pengendalian pergerakan *point to point* robot tiga roda tipe *holonomic*. MPC merupakan salah satu metode pengendalian dan merupakan strategi pengendalian lanjutan. Motivasi utama menggunakan pengendalian prediksi berbasis model adalah kemampuannya untuk melakukan pertimbangan secara langsung, dapat meminimalisir nilai kesalahan, dan didesain untuk dapat memprediksi waktu di masa depan dan mengoptimalkan prediksi tersebut. Metode pengendalian yang dilakukan adalah mengendalikan nilai kecepatan untuk menghasilkan nilai posisi robot. Hasil dari pergerakan segi empat hanya dengan pengendalian PID mendapatkan nilai *error* rata-rata pada koordinat x dan y secara berurutan 9,17% dan 6,05%, pada pergerakan segi tiga hanya dengan pengendalian PID mendapatkan nilai *error* rata-rata pada koordinat x dan y secara berurutan 7,3% dan 3,3%. Hasil dari pergerakan segi empat dengan pengendalian MPC mendapatkan nilai rata-rata *error* pada koordinat x dan y berurutan 8,52% dan 5,96%, pada pergerakan segi tiga dengan pengendalian MPC mendapatkan nilai *error* rata-rata pada koordinat x dan y berurutan 7,55% dan 4,03%.

Kata kunci: MPC, robot *omni-directional* 3 roda.



DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	ix
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
BAB 2 TINJAUAN PUSTAKA.....	3
2.1 Studi Literatur.....	3
2.2 Tinjauan Teori.....	4
2.2.1 Omni Directional.....	4
2.2.2 Lokalisasi.....	5
2.2.3 Konfigurasi MPC.....	6
2.2.4 Indeks Performa <i>Model Predictive Control</i>	7
2.2.5 Mengatur nilai batasan.....	7
BAB 3 METODOLOGI.....	8
3.1 Mendesain MPC.....	8
3.1.1 Model Prediksi Waktu Diskrit.....	9
3.1.2 Indeks Performa <i>Model Predictive Control</i>	10

3.1.3 Mengatur nilai batasan	11
3.2 Mendesain komunikasi serial.....	12
3.3 Proses simulasi dan analisis	13
BAB 4 HASIL DAN PEMBAHASAN.....	14
4.1 Pergerakan robot membentuk sebuah segi empat.....	14
4.2 Pergerakan robot membentuk sebuah segi tiga.....	16
BAB 5 KESIMPULAN DAN SARAN.....	18
5.1 Kesimpulan	18
5.2 Saran	18
DAFTAR PUSTAKA	19
LAMPIRAN	1



DAFTAR GAMBAR

Gambar 2.1 Struktur kinematika robot beroda.....	4
Gambar 3.1 Diagram blok pengendalian MPC	8
Gambar 3.2 Membentuk model <i>state space</i> di MatLab	9
Gambar 3.3 Menentukan nilai N_p dan N_c di MatLab	10
Gambar 3.4 Menentukan nilai <i>constant weight</i> di MatLab	11
Gambar 3.5 Mengatur nilai batasan untuk setiap motor di MatLab.....	11
Gambar 3.6 Perangkat komunikasi serial HC-06.....	12
Gambar 3.7 Alur pengiriman dan penerimaan data	12
Gambar 3.8 Mendefinisikan komunikasi serial yang digunakan di MatLab	12
Gambar 3.9 Mengatur pengiriman dan penerimaan data di MatLab	12
Gambar 3.10 Pergerakan robot membentuk segi empat	13
Gambar 3.11 Pergerakan robot membentuk segi tiga	13
Gambar 4.1 Skema pergerakan robot	14
Gambar 4.2 Pergerakan riil robot berbentuk segi empat satu putaran dengan lima kondisi berbeda	14
Gambar 4.3 Pergerakan riil robot berbentuk segi empat satu putaran dengan lima kondisi berbeda	16

DAFTAR TABEL

Tabel 4.1 Data nilai <i>error</i> pergerakan segi empat.....	15
Tabel 4.2 Data nilai <i>error</i> pergerakan segi tiga	17



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi robot kini banyak berperan dalam kehidupan sehari-hari manusia. Seperti berperan dalam mempermudah kegiatan manusia saat bekerja, membantu manusia dalam melakukan komputasi sederhana hingga komputasi yang rumit, bahkan dapat meniru fungsi sosial manusia. Beragam jenis robot sudah semakin berkembang di era industri 4.0. Setiap robot memiliki jenis dan fungsi yang berbeda, salah satunya adalah robot beroda tipe *holonomic* tiga roda.

Robot beroda tipe *holonomic* adalah robot yang dapat bergerak ke segala arah dengan menggunakan roda *omni-directional* yang memungkinkan terjadinya gerakan translasi dan rotasi. Roda *omni-directional* adalah roda dengan piringan kecil di sekitar lingkaran tegak lurus terhadap arah putaran roda. Implementasi pada robot beroda tipe *holonomic* tiga roda adalah mengikuti jalur yang sudah ditentukan dan robot tahu kapan harus melakukan percepatan dan perlambatan. Ketika robot awal bergerak dibutuhkan percepatan, dan sebaliknya ketika robot mendekati target robot mengalami perlambatan. Maka dibutuhkan suatu aksi pengendalian untuk memaksimalkan gerak robot di jalur yang sudah ditentukan.

Dalam implementasi yang sesungguhnya sulit untuk mendapatkan kinerja yang baik, karena kendala masukan dan gangguan - gangguan yang secara alami muncul pada perangkat keras. Masalah ini dapat diatasi dengan menggunakan *Model Predictive Control*. MPC dapat memberikan keluaran yang optimal. Dengan mempertimbangkan kendala pada masukan, dapat menghasilkan aksi pengendalian yang mempertimbangkan batasan dari penggerak.

Model Predictive Control adalah strategi pengendalian optimal yang menggunakan model dari sistem untuk mendapatkan urutan pengendalian optimal dengan meminimalkan fungsi tujuan (*objective function*). Setiap jeda pada sampling, model digunakan untuk memprediksi tingkah laku dari sistem. Berdasarkan prediksi tersebut, dengan paksa fungsi objektif diminimalkan untuk urutan masukan selanjutnya, sehingga membutuhkan solusi dari masalah optimasi yang terbatas untuk setiap *sampling interval*[1].

Permasalahan lainnya ialah bagaimana kita dapat menerapkan pengendalian MPC ke robot beroda tipe *holonomic* tiga roda. Dengan mempertimbangkan berbagai faktor seperti batasan, kendala, nilai *error*, dan perangkat keras yang digunakan, robot harus mampu bergerak di jalur yang sudah ditentukan dan menjalankan pengendalian MPC untuk meminimalisir kesalahan jalur yang dilalui robot.

1.2 Rumusan Masalah

Bagaimana mendesain pengendalian MPC untuk menyelesaikan masalah *trajectory tracking* pada robot *holonomic* tiga roda.

1.3 Batasan Masalah

1. Penelitian ini tidak memperhitungkan masa robot.
2. Eksperimen dilakukan di lantai.
3. Sensor *rotary encoder* hanya terdapat pada motor DC PG28.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mendesain pengendalian MPC robot beroda untuk meminimalkan permasalahan yang terjadi pada robot *holonomic* tiga roda untuk pelacakan lintasan dengan meminimalkan nilai fungsi objektif dalam rentang cakrawala waktu tertentu.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah menghasilkan desain pengendalian robot *holonomic* tiga roda, untuk bergerak sesuai dengan koordinat yang diinginkan menggunakan pengendalian MPC.

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Pengendalian robot berbasis MPC telah sering dilakukan, seperti yang dilakukan oleh Aslam[2] membahas tentang implementasi *real-time* MPC menggunakan 16-bit *microcontroller* PIC untuk mengendalikan kecepatan motor DC di lingkungan terbatas. Dengan melakukan simulasi dalam empat mode, mode tanpa kendala, mode dengan kendala input, mode dengan kendala output, serta mode dengan kendala input dan output. Penelitian ini menyimpulkan dari keempat mode tersebut MPC dapat memberikan hasil yang lebih baik. MPC berhasil diimplementasikan pada *microcontroller* PIC 16-bit dengan metode *Single Input Single Output* (SISO)[2].

Pada tahun 2014 dalam penelitian Zometa[3] membahas tentang implementasi dari aspek yang berhubungan antara MPC dengan Embedded Platform. Penelitian dalam paper, menunjukkan bahwa kuantitas input yang terjadi dalam aktuator harus diperhitungkan secara berurutan untuk menentukan tingkat *sub* optimalisasi. Penelitian ini menggunakan robot *segway* yang dimodelkan dengan sistem LTI.

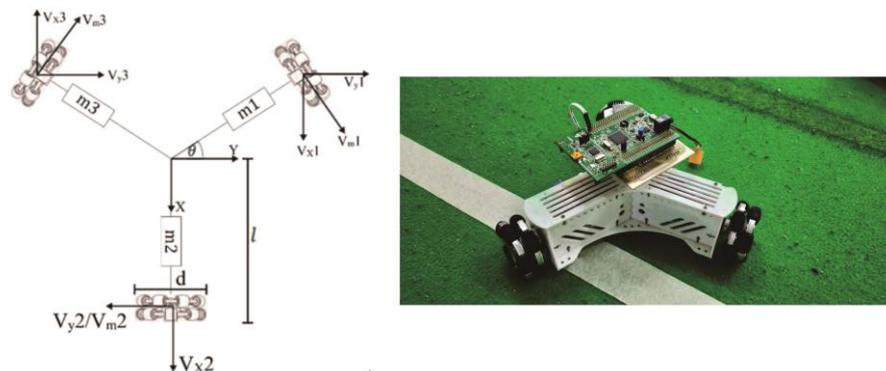
Penelitian lain oleh Filipe dkk[4] dengan menggunakan robot beroda *nonholonomic*, mereka berhasil mengaplikasikan MPC untuk mengatasi masalah *trajectory tracking*. Mereka menggunakan MPC termotivasi oleh masalah kemampuannya untuk mempertimbangkan dengan cara terbaik, pengendalian dan batasan tertentu yang biasanya muncul pada masalah tertentu. Pada penelitiannya untuk menyelesaikan masalah *trajectory tracking* menggunakan dua pendekatan. Pendekatan pertama menggunakan *nonlinear* MPC, pendekatan kedua menggunakan *linear* MPC. Hasil simulasi pada penelitian, dilakukan untuk melihat seberapa efektif kedua pendekatan tersebut. Alhasil pendekatan *linear* MPC memungkinkan untuk mengurangi upaya komputasi sejak masalah optimisasi telah diubah. Sebagai perbandingan dengan pendekatan *nonlinear*, telah dicatat bahwa strategi *linear* MPC mempertahankan kinerja yang baik, dengan upaya perhitungan yang lebih rendah, ini menunjukkan bahwa pendekatan *linear* MPC memiliki kerugian yang hanya berlaku untuk titik terdekat lintasan referensi[4].

Berdasarkan penelitian mengenai MPC, pengendalian menggunakan MPC melakukan tugas dengan baik sehingga meminimalkan kesalahan yang ada. MPC juga dirancang untuk sistem *nonlinear* dengan banyak ketidakpastian dan didesain untuk dapat memprediksi waktu di masa depan dan mengoptimalkan prediksi tersebut, sehingga dapat menghadapi kondisi yang cepat berubah.

2.2 Tinjauan Teori

2.2.1 Omni Directional

Kinematika robot adalah persamaan gerak robot yang dibuat berdasarkan arah kecepatan gerak robot tanpa memperhitungkan gaya dan masa robot. Model kinematika robot didapatkan dari hubungan antara kecepatan robot dengan kecepatan setiap roda *omni* sesuai Gambar 2.1. Kinematika robot pada penelitian ini dibuat tanpa memperhitungkan adanya slip pada setiap roda *omni* dan gaya gesek roda terhadap lantai diabaikan [7].



Gambar 2.1 Struktur kinematika robot beroda

Model kinematika yang dituliskan dari Gambar 2.1:

$$\begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\cos\theta & 0 \\ \sin\theta & \sin\theta & -1 \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} v_{m1} \\ v_{m3} \\ v_{m2} \end{bmatrix} \quad (2.1)$$

Dimana:

θ = sudut 30° ;

l = jarak antara roda dan titik pusat robot;

d = diameter roda robot;

v_i = kecepatan robot ke-i;

$v_{m i}$ = kecepatan roda omni ke-i;

$v_{x i}$ = kecepatan ke arah sumbu x roda *omni* ke-i;

$v_{y i}$ = kecepatan ke arah sumbu y roda *omni* ke-i;

v_θ = kecepatan sudut robot;

2.2.2 Lokalisasi

Lokalisasi adalah sebuah masalah dimana robot harus mengetahui posisinya pada suatu waktu tertentu. Lokalisasi digunakan untuk menggerakkan robot ke suatu titik secara otomatis [5]. Manfaat lain dari penyelesaian masalah lokalisasi ini adalah membantu robot untuk mengetahui penghalang yang berada di sekitar robot. Aplikasi untuk permasalahan lokalisasi dari teknik *dead-reckoning* menggunakan encoder adalah *odometry*.

Odometry adalah metode penggunaan data dari aktuator yang difungsikan untuk menghitung perubahan posisi pada robot. Untuk melakukan metode ini diperlukan sensor *rotary encoder* untuk mendapatkan pulsa sebagai data [8]. Data yang diperoleh nanti diolah menjadi sebuah informasi oleh *microcontroller* untuk pemetaan posisi dan navigasi.

Perhitungan *odometry* dilakukan dengan cara mengendalikan nilai kecepatan robot, sehingga mendapat nilai posisi. Persamaan dapat dituliskan sebagai berikut:

$$\theta(k) = \theta(k - 1) + v_{\theta}T_s; \quad (2.2)$$

$$x(k) = x(k - 1) + v_xT_s; \quad (2.3)$$

$$y(k) = y(k - 1) + v_yT_s; \quad (2.4)$$

Dimana:

$x(k)$ = nilai jarak robot pada koordinat x waktu ke- k

$y(k)$ = nilai jarak robot pada koordinat y waktu ke- k

$\theta(k)$ = nilai orientasi robot waktu ke- k

$x(k - 1)$ = nilai jarak robot pada koordinat y waktu ke- $(k - 1)$

$y(k - 1)$ = nilai jarak robot pada koordinat x waktu ke- $(k - 1)$

$\theta(k - 1)$ = nilai orientasi robot waktu ke- $(k - 1)$

v_{θ} = nilai kecepatan sudut robot

v_x = nilai kecepatan robot kearah sumbu x

v_y = nilai kecepatan robot kearah sumbu y

T_s = waktu sampling robot

2.2.3 Konfigurasi MPC

Model Predictive Control adalah strategi pengendalian optimal yang menggunakan model dari sistem untuk mendapatkan urutan pengendalian yang optimal dengan meminimalkan fungsi objektif. Setiap jeda pada sampling, model digunakan untuk memprediksi tingkah laku dari sistem. Berdasarkan prediksi tersebut, dengan paksa fungsi objektif diminimalkan untuk urutan masukan selanjutnya, sehingga membutuhkan solusi dari masalah optimasi yang terbatas untuk setiap *sampling interval*[1].

Variabel dasar yang digunakan pengendalian MPC untuk memprediksi tingkah laku sistem adalah k_i, N_p, N_c . Dimana estimasi pengendalian *trajectory* ditunjukkan oleh persamaan [9]:

$$\Delta u(k_i + N_c - 1) \quad (2.6)$$

Dimana:

k_i = nilai sampling;

Δu = nilai pengendalian *trajectory*;

N_c = nilai variabel cakrawala pengendali;

Nilai N_c digunakan untuk menangkap nilai pengendalian *trajectory* di waktu mendatang. Sedangkan nilai variabel kondisi di waktu mendatang diprediksi oleh N_p . N_p bisa disimpulkan panjang dari suatu optimasi. Nilai variabel kondisi di waktu mendatang ditunjukkan oleh persamaan [9]:

$$x(k_i + N_p | k_i) \quad (2.7)$$

Dimana:

$x(k_i)$ = nilai informasi saat ini;

N_p = nilai cakrawala prediksi;

Berdasarkan *state - space model* (A, B, C), nilai variabel kondisi di waktu mendatang dikalkulasikan secara berurutan menggunakan persamaan[1][2][9]:

$$x(k_i + N_p | k_i) = CA^{N_p}x(k_i) + \dots + CA^{N_p - N_c}B\Delta u(k_i + N_c - 1) \quad (2.8)$$

2.2.4 Indeks Performa Model Predictive Control

Cost function adalah indeks performa dalam MPC yang nilainya harus diminimalkan, persamaan yang dimiliki oleh *cost function* merupakan cerminan sebuah *error* terhadap nilai referensi dan aksi pengendalian. Dalam penelitian ini indeks performa yang digunakan adalah *Linear Quadratic Predictive Control*.

Demikian adalah persamaan *cost function* dalam penelitian ini [1][9]:

$$\min \sum_{k=1}^N (y[k] - r[k])^T Q_k (y[k] - r[k]) + \sum_{k=0}^{N-1} u^T[k] R_k u[k] + \Delta u^T[k] E_k \Delta u[k] \quad (2.9)$$

Dimana:

$r[k]$ = nilai referensi;

$y[k]$ = nilai perhitungan keluaran;

Q = Matriks bobot *state*;

R = Matriks bobot pengendali;

E = Nilai bobot dari masukan pengendali;

2.2.5 Mengatur nilai batasan

Nilai batasan diperlukan untuk mempercepat dan memastikan solusi yang sebenarnya dalam *real-time*. Sinyal dengan *range* yang tidak terbatas menghasilkan nilai yang tidak realistis, itulah mengapa dibutuhkan batasan dalam sistem. Memberikan batasan pada keluaran dan pada pengendalian maka pertidaksamaan yang terbentuk adalah:

$$y_{min} \leq y[k] \leq y_{max} \quad (2.10)$$

$$u_{min} \leq u[k] \leq u_{max} \quad (2.11)$$

Dimana:

$y_{min, i}$ = Batas bawah sinyal keluaran i ;

$y_{max, i}$ = Batas atas sinyal keluaran i ;

$u_{min, i}$ = Batas bawah sinyal kendali i ;

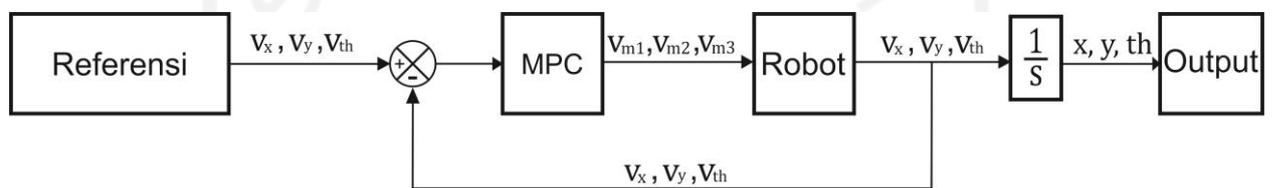
$u_{max, i}$ = Batas atas sinyal kendali i ;

BAB 3

METODOLOGI

3.1 Mendesain MPC

Pengendalian MPC digunakan untuk mengendalikan nilai posisi robot saat ini. Blok ini menggunakan algoritma MPC sebagai pengendaliannya. Algoritma prediksi dilakukan dengan mengubah nilai N_c , nilai N_p , dan nilai *weights* pada algoritma MPC. Rumusan algoritma prediksi dituliskan dalam persamaan 2.9. Dalam mendesain MPC penelitian ini menggunakan nilai N_c lebih kecil atau sama dengan N_p . Gambar 3.1 menunjukkan bagaimana blok diagram didesain.



Gambar 3.1 Diagram blok pengendalian MPC

Dimana:

V_{xr} = Kecepatan referensi koordinat x ;

V_{yr} = Kecepatan referensi koordinat y ;

V_{thr} = Kecepatan referensi koordinat $theta$;

V_x = Kecepatan koordinat x ;

V_y = Kecepatan koordinat y ;

V_{th} = Kecepatan orientasi robot;

V_{m1} = Kecepatan roda robot 1;

V_{m2} = Kecepatan roda robot 2;

V_{m3} = Kecepatan roda robot 3;

x = Nilai aktual robot koordinat x ;

y = Nilai aktual robot koordinat y ;

th = Nilai aktual orientasi robot;

3.1.1 Model Prediksi Waktu Diskrit

Model dasar yang dibutuhkan dalam penelitian ini adalah model persamaan diskrit[1][9]:

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k); \\y(k) &= Cx(k) + Du(k);\end{aligned}\tag{3.1}$$

Didapatkan model *state space* sebagai berikut:

$$\begin{aligned}\begin{bmatrix}vx \\vy \\v\theta\end{bmatrix} &= \begin{bmatrix}1 & 0 & 0 \\0 & 1 & 0 \\0 & 0 & 1\end{bmatrix} + \begin{bmatrix}\sin\theta & -\sin\theta & 0 \\-\cos\theta & -\cos\theta & 1 \\\frac{1}{l} & \frac{1}{l} & \frac{1}{l}\end{bmatrix} \begin{bmatrix}v1 \\v2 \\v3\end{bmatrix} \\ \begin{bmatrix}x \\y \\\theta\end{bmatrix} &= \begin{bmatrix}1 & 0 & 0 \\0 & 1 & 0 \\0 & 0 & 1\end{bmatrix} + [0]\end{aligned}\tag{3.2}$$

```
phi = 2*pi/3;
%L = 0.3;
L = 1/0.0714;
A = [ 1 0 0;...
      0 1 0;...
      0 0 1];

B = [ sin(phi) -sin(phi) 0;...
      -cos(phi) -cos(phi) -1;...
      1/L      1/L      1/L];

C = [ 1 0 0;...
      0 1 0;...
      0 0 1];

D = zeros(3);

Ts = 0.1; %sampling time
N = 7; %Prediction Horizon
Nu = 3; %Control Horizon

plantDisc = ss(A,B,C,D,Ts); %State Space Model
MPCobj = mpc(plantDisc, Ts, N, Nu); %MPC controller
```

Gambar 3.2 Membentuk model *state space* di MatLab

Dimana:

$x(k)$ = Kondisi awal;

$y(k)$ = Nilai perhitungan keluaran;

$u(k)$ = Masukan pengendalian;

A, B, C, D = Matriks *state-space*;

3.1.2 Indeks Performa Model Predictive Control

Cost function adalah indeks performa dalam MPC yang nilainya harus diminimalkan.

Persamaan 2.9 didapatkan nilai untuk mengatur objek MPC dengan memberikan nilai pada:

$$\min \sum_{k=1}^N (y[k] - r[k])^T Q_k (y[k] - r[k]) + \sum_{k=0}^{N-1} u^T[k] R_k u[k] + \Delta u^T[k] E_k \Delta u[k]$$

Dimana:

N_p = Cakrawala prediksi;

N_c = Cakrawala pengendalian ($1 \leq N_c \leq N_p$);

Penelitian ini menggunakan empat kondisi nilai cakrawala yang berbeda-beda yaitu:

1. $N_p = 3; N_c = 1;$
2. $N_p = 3; N_c = 3;$
3. $N_p = 7; N_c = 3;$
4. $N_p = 7; N_c = 5;$

```
Ts = 0.1; %sampling time
N = 7; %Prediction Horizon
Nu = 3; %Control Horizon
```

Gambar 3.3 Menentukan nilai N_p dan N_c di MatLab

Persamaan 2.9 didapatkan nilai untuk mengatur *weights* MPC dengan memberikan nilai pada:

$$\min \sum_{k=1}^N (y[k] - r[k])^T Q_k (y[k] - r[k]) + \sum_{k=0}^{N-1} u^T[k] R_k u[k] + \Delta u^T[k] E_k \Delta u[k]$$

Maka penggunaan nilai *constant weight* pada penelitian ini adalah:

$Q = [4.95303242439511 \ 4.95303242439511 \ 4.95303242439511];$

$R = [0.01 \ 0.01 \ 0.01];$

$E = [0.0201896517994655 \ 0.0201896517994655 \ 0.0201896517994655];$

```

q = [4.95303242439511 4.95303242439511 4.95303242439511];
MPCobj.W.Output = q;
c = [0.01 0.01 0.01];
MPCobj.W.Input=c;
d = [0.0201896517994655 0.0201896517994655 0.0201896517994655];
MPCobj.W.InputRate=d;

```

Gambar 3.4 Menentukan nilai *constant weight* di MatLab

3.1.3 Mengatur nilai batasan

Nilai batasan diperlukan untuk mempercepat dan memastikan solusi yang sebenarnya dalam *real-time*. Pertidaksamaan yang terbentuk adalah:

$$y_{min} \leq y[k] \leq y_{max} \quad (3.4)$$

$$u_{min} \leq u[k] \leq u_{max} \quad (3.5)$$

Dimana pada penelitian ini saya menggunakan nilai:

$$y_{min}, i = -1024;$$

$$y_{max}, i = 1024;$$

$$u_{min}, i = -1024;$$

$$u_{max}, i = 1024;$$

```

MPCobj.OV(1).Min = -1024;
MPCobj.OV(1).Max = 1024;
MPCobj.OV(2).Min = -1024;
MPCobj.OV(2).Max = 1024;
MPCobj.OV(3).Min = -1024;
MPCobj.OV(3).Max = 1024;

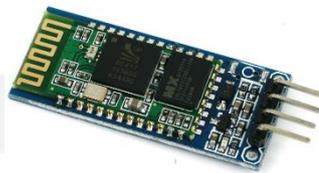
MPCobj.MV(1).Min = -1024;
MPCobj.MV(1).Max = 1024;
MPCobj.MV(2).Min = -1024;
MPCobj.MV(2).Max = 1024;
MPCobj.MV(3).Min = -1024;
MPCobj.MV(3).Max = 1024;

```

Gambar 3.5 Mengatur nilai batasan untuk setiap motor di MatLab

3.2 Mendesain komunikasi serial

Gambar 3.6 adalah gambar HC-06 yang digunakan dalam penelitian ini. HC-06 merupakan perangkat yang digunakan untuk mengirim dan menerima informasi data secara nirkabel dari PC/microcontroller atau menuju PC/microcontroller. Modul ini bekerja pada protokol komunikasi *Bluetooth 2.0* dan menggunakan teknik *frequency hopping spread spectrum (FHSS)* untuk menghindari interferensi dari perangkat lain. Dimana modul ini baik digunakan untuk komunikasi data secara *point to point*. Perangkat HC-06 digunakan untuk komunikasi satu perangkat saja, yaitu satu *transmitter* dan satu *receiver*. Dalam HC-06 terdapat enam pin, tetapi pada dasarnya pin yang digunakan hanya tiga, yaitu pin *ground*, pin TX, dan pin RX. Gambar 3.7 menunjukkan bagaimana alur data terkirim dari PC/microcontroller dan diterima oleh PC/microcontroller.



Gambar 3.6 Perangkat komunikasi serial HC-06



Gambar 3.7 Alur pengiriman dan penerimaan data

Transmisi data dilakukan secara *bit per bit*. Kecepatan *transfer* data harus sama antara pengiriman dan penerimaan agar tidak terjadi *overflow*. Kecepatan transmisi data disebut dengan *baudrate*. Dimana dalam penelitian ini saya menggunakan *baudrate 9600*.

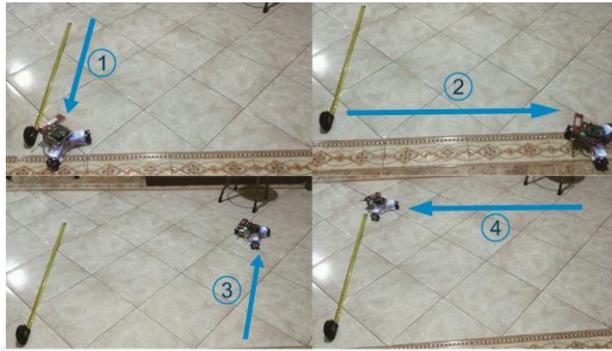
```
clear; clc; close all;
delete(instrfind);
s = serial('COM10', 'BaudRate', 9600);
fopen(s); |
```

Gambar 3.8 Mendefinisikan komunikasi serial yang digunakan di MatLab

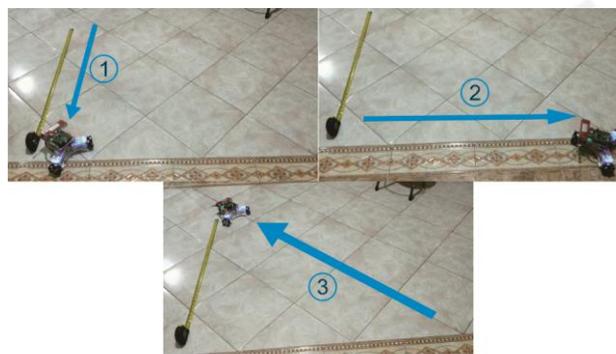
```
%-----Mengirim Data-----
fprintf(s,data6); %mengirim data ke robot
%-----Membaca Data-----
data = fscanf(s);
```

Gambar 3.9 Mengatur pengiriman dan penerimaan data di MatLab

3.3 Proses simulasi dan analisis



Gambar 3.10 Pergerakan robot membentuk segi empat



Gambar 3.11 Pergerakan robot membentuk segi tiga

Percobaan untuk pengendalian robot tipe *holonomic* 3 roda adalah mengendalikan robot dengan referensi lintasan berbentuk segi empat seperti pada Gambar 3.10 dan berbentuk segi tiga seperti pada Gambar 3.11, dengan beberapa percobaan yang dilakukan diantaranya:

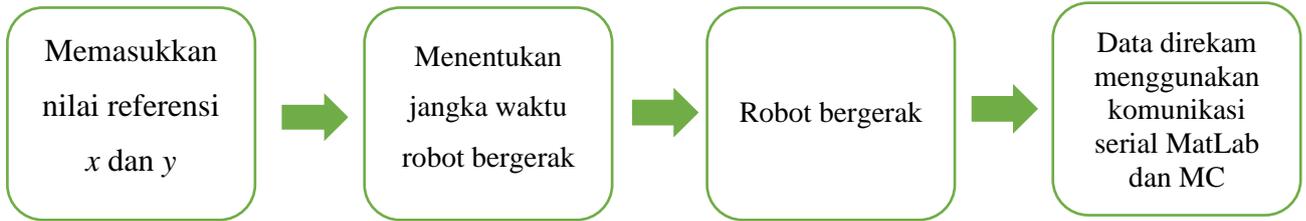
1. Percobaan robot membentuk sebuah segi empat dan segi tiga hanya dengan pengendalian PID pada setiap motor.
2. Percobaan robot membentuk sebuah segi empat dan segi tiga dengan pengendalian MPC.

Analisis dari percobaan ini dilakukan dengan membandingkan nilai rata-rata persen *error* dari dua percobaan. Nilai persen error didapatkan dari seberapa jauh nilai posisi aktual robot dengan nilai referensi lintasan yang diberikan.

$$\%error = |referensi lintasan - posisi aktual robot| \times 100\% \quad (3.6)$$

BAB 4

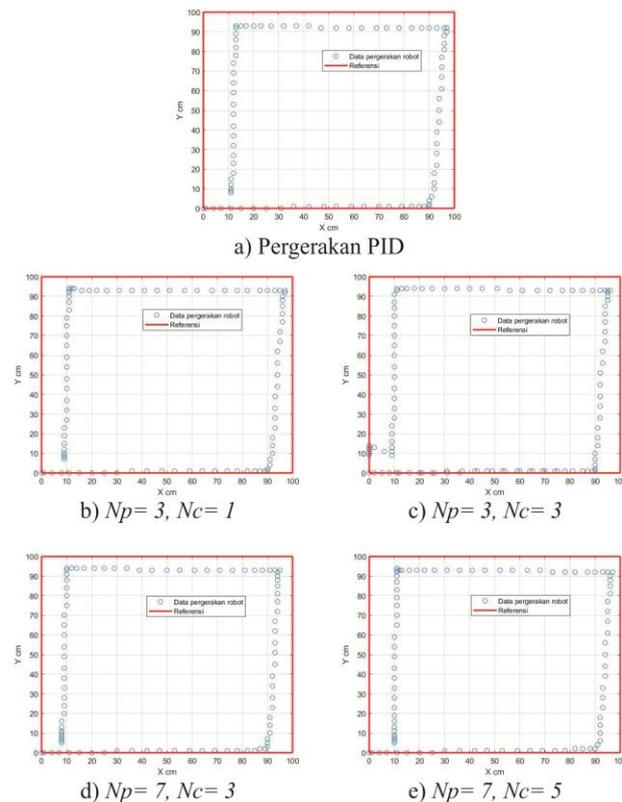
HASIL DAN PEMBAHASAN



Gambar 4.1 Skema pergerakan robot

Percobaan pada Gambar 4.1 dilakukan dengan memberikan nilai target pada robot dan data direkam di aplikasi MatLab sehingga akan memunculkan gambar atau grafik pergerakan robot. Percobaan yang akan dilakukan kali ini menggunakan nilai target x dan y sebesar 100cm dan robot bergerak membentuk segi empat dan segi tiga. Setelah dilakukan percobaan nilai error dirata-rata.

4.1 Pergerakan robot membentuk sebuah segi empat



Gambar 4.2 Pergerakan riil robot berbentuk segi empat satu putaran dengan lima kondisi berbeda

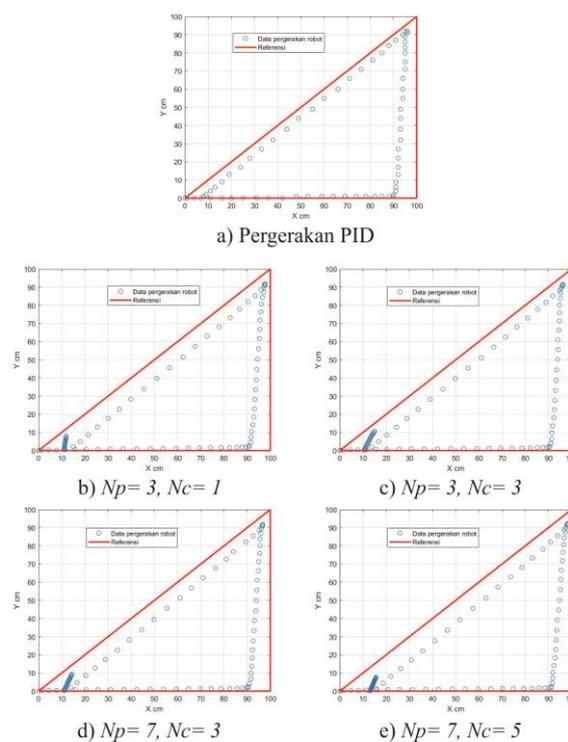
Gambar 4.2 menunjukkan robot mampu bergerak membentuk pola segi empat. Garis merah merupakan target dari pergerakan yang diinginkan. Data bulat berwarna biru merupakan data rekaman dari robot ketika bergerak. Dari lima pergerakan ini dapat dilihat pergerakan robot sudah membentuk sebuah segi empat dengan baik. Gambar 4.2 a) menunjukkan pergerakan segi empat robot dengan pengendalian PID, b) menunjukkan pergerakan segi empat robot dengan pengendalian MPC dengan nilai $N_p=3$ dan $N_c=1$, c) menunjukkan pergerakan segi empat robot dengan pengendalian MPC dengan nilai $N_p=3$ dan $N_c=3$, d) menunjukkan pergerakan segi empat robot dengan pengendalian MPC dengan nilai $N_p=7$ dan $N_c=3$, e) menunjukkan pergerakan segi empat robot dengan pengendalian MPC dengan nilai $N_p=7$ dan $N_c=5$. Pada percobaan ini juga tidak menghiraukan masa robot, sehingga tidak ada perhitungan kecepatan putar motor terhadap masa robot. Pergerakan robot tipe *holonomic* tiga roda memiliki nilai *error* seperti yang ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Data nilai *error* pergerakan segi empat

MPC								PID				
Nilai MPC		t (detik)	Target		Nilai Riil		Error (%)		Nilai Riil		Error (%)	
Prediction Horizon	Control Horizon		x	y	x	y	x	y	x	y	x	Y
3	1	0 – 2,5	100	0	90,3	2,41	9,72	2,41	90,7	1,64	9,31	1,64
		2,5 – 5	100	100	97,4	93,3	2,65	6,74	97,6	92,8	2,38	7,21
		5 – 7,5	0	100	11,7	94,2	11,7	5,8	13,5	86,5	13,5	6,51
		7,5 – 10	0	0	9,51	8,79	9,51	8,79	11,6	8,87	11,6	8,87
Rata-rata persen error							8,39	5,94			9,17	6,05
3	3	0 – 2,5	100	0	89,5	2,19	10,5	2,19	90,7	1,64	9,31	1,64
		2,5 – 5	100	100	95,6	93,8	4,42	6,19	97,6	92,8	2,38	7,21
		5 – 7,5	0	100	10,5	95,1	10,5	4,94	13,5	86,5	13,5	6,51
		7,5 – 10	0	0	9,11	9,81	9,11	9,81	11,6	8,87	11,6	8,87
Rata-rata persen error							8,63	5,78			9,17	6,05
7	3	0 – 2,5	100	0	89,7	2,35	10,3	2,35	90,7	1,64	9,31	1,64
		2,5 – 5	100	100	95,5	93,3	4,55	6,74	97,6	92,8	2,38	7,21
		5 – 7,5	0	100	10,5	94,8	10,5	5,25	13,5	86,5	13,5	6,51
		7,5 – 10	0	0	8,7	9,57	8,7	9,57	11,6	8,87	11,6	8,87
Rata-rata persen error							8,5	5,98			9,17	6,05
7	5	0 – 2,5	100	0	90,8	2,04	9,17	2,04	90,7	1,64	9,31	1,64
		2,5 – 5	100	100	97,4	92,6	2,65	7,37	97,6	92,8	2,38	7,21
		5 – 7,5	0	100	12,1	94	12,1	6,04	13,5	86,5	13,5	6,51
		7,5 – 10	0	0	10,3	9,26	10,3	9,26	11,6	8,87	11,6	8,87
Rata-rata persen error							8,56	6,18			9,17	6,05

Pada percobaan pengendalian robot membentuk pola segi empat, dari Tabel 4.1 pengaruh adanya nilai MPC pada pergerakan segi empat membuat nilai rata-rata *error* menjadi lebih rendah daripada nilai rata-rata *error* hanya dengan pengendalian PID. Pengaruh bertambahnya nilai N_p pada koordinat x membuat nilai rata-rata *error* menjadi lebih kecil, sedangkan pada koordinat y nilai rata-rata *error* menjadi lebih besar. Pengaruh bertambahnya nilai N_c pada koordinat x membuat nilai rata-rata *error* menjadi lebih besar, sedangkan pada koordinat y membuat nilai rata-rata *error* menjadi lebih kecil. Nilai riil pergerakan robot pada simulasi MPC masih memiliki selisih yang cukup besar dengan nilai referensi lintasan yang diberikan.

4.2 Pergerakan robot membentuk sebuah segi tiga



Gambar 4.3 Pergerakan riil robot berbentuk segi empat satu putaran dengan lima kondisi berbeda

Gambar 4.3 menunjukkan pergerakan robot mampu bergerak membentuk pola segi tiga dengan lima kondisi yang berbeda. Gambar 4.3 a) menunjukkan pergerakan segi tiga robot dengan pengendalian PID, b) menunjukkan pergerakan segi tiga robot dengan pengendalian MPC dengan nilai $N_p= 3$ dan $N_c= 1$, c) menunjukkan pergerakan segi tiga robot dengan pengendalian MPC dengan nilai $N_p= 3$ dan $N_c= 3$, d) menunjukkan pergerakan segi tiga robot dengan pengendalian MPC dengan nilai $N_p= 7$ dan $N_c= 3$, e) menunjukkan pergerakan segi tiga robot dengan pengendalian MPC dengan nilai $N_p= 7$ dan $N_c= 5$. Dari percobaan pergerakan membentuk pola segi tiga maka didapatkan nilai *error* ditunjukkan oleh Tabel 4.2.

Tabel 4.2 Data nilai *error* pergerakan segi tiga

MPC								PID				
Nilai MPC		t (detik)	Target		Nilai riil		Error (%)		Nilai riil		Error (%)	
Predictive Horizon	Control Horizon		x	y	x	y	x	y	x	y	x	y
3	1	0 – 2,5	100	0	90,69	1,96	9,31	1,96	90	1	10	1
		2,5 – 5	100	100	97,67	91,92	2,24	8,08	96	92	4	8
		5 – 7,5	0	0	10,47	1,96	10,47	1,96	8	1	8	1
Rata-rata persen error							7,34	4			7,33	3,33
3	3	0 – 2,5	100	0	90,28	1,72	9,72	1,72	90	1	10	1
		2,5 – 5	100	100	96,26	91,96	3,74	8,31	96	92	4	8
		5 – 7,5	0	0	9,73	1,47	9,73	1,47	8	1	8	1
Rata-rata persen error							7,73	3,83			7,33	3,33
7	3	0 – 2,5	100	0	90,42	1,96	9,58	1,96	90	1	10	1
		2,5 – 5	100	100	96,67	91,92	3,33	8,08	96	92	4	8
		5 – 7,5	0	0	10,14	2,05	10,14	2,05	8	1	8	1
Rata-rata persen error							7,68	4,03			7,33	3,33
7	5	0 – 2,5	100	0	91,23	1,96	8,77	1,96	90	1	10	1
		2,5 – 5	100	100	98,3	92,94	1,7	7,76	96	92	4	8
		5 – 7,5	0	0	11,91	3,07	11,91	3,07	8	1	8	1
Rata-rata persen error							7,46	4,26			7,33	3,33

Pada percobaan pengendalian robot membentuk pola segi tiga memiliki hasil yang berbeda dengan percobaan pengendalian robot membentuk pola segi empat. Pengaruh adanya pengendalian MPC membuat nilai *error* rata-rata lebih tinggi daripada nilai *error* rata-rata percobaan hanya dengan pengendalian PID. Pengaruh nilai N_p dan N_c pada pengendalian robot membentuk pola segi tiga sama dengan pengaruh nilai N_p dan N_c pada pengendalian robot membentuk pola segi empat. Dimana pengaruh bertambahnya nilai N_p pada koordinat x membuat nilai rata-rata *error* menjadi lebih kecil, sedangkan pada koordinat y nilai rata-rata *error* menjadi lebih besar. Pengaruh bertambahnya nilai N_c pada koordinat x membuat nilai rata-rata *error* menjadi lebih besar, sedangkan pada koordinat y membuat nilai rata-rata *error* menjadi lebih kecil.

Berdasarkan nilai rata-rata *error* parameter MPC terbaik dalam dua percobaan yang dilakukan adalah parameter dengan nilai $N_p = 3$ dan $N_c = 1$, dimana robot beroda tipe *holonomic* tiga roda mampu bergerak sesuai dengan nilai referensi lintasan dengan nilai *error* yang paling rendah. Parameter MPC terbaik kedua yaitu $N_p = 3$ dan $N_c = 3$, Parameter MPC terbaik ketiga yaitu $N_p = 7$ dan $N_c = 3$, Parameter MPC terbaik keempat yaitu $N_p = 7$ dan $N_c = 5$.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Pengendalian menggunakan MPC pada pergerakan robot membentuk pola segi empat memiliki nilai rata-rata *error* yang lebih rendah dibandingkan dengan nilai rata-rata *error* pergerakan robot pola segi empat dengan pengendalian PID, yaitu pada pengendalian MPC didapatkan nilai rata-rata *error* koordinat sumbu x dan y berturut-turut adalah 8,52% dan 5,97%, sedangkan pada pengendalian PID didapatkan nilai rata-rata *error* koordinat sumbu x dan y berturut-turut adalah 9,17% dan 6,05%.
2. Pengendalian menggunakan MPC pada pergerakan robot membentuk pola segi tiga memiliki nilai rata-rata *error* yang lebih tinggi dibandingkan dengan nilai rata-rata *error* pergerakan robot pola segi tiga dengan pengendalian PID, yaitu pada pengendalian MPC didapatkan nilai rata-rata *error* koordinat sumbu x dan y berturut-turut adalah 7,55% dan 4,03%, sedangkan pada pengendalian PID didapatkan nilai rata-rata *error* koordinat sumbu x dan y berturut-turut adalah 7,33% dan 3,33%.
3. Parameter MPC dengan nilai rata-rata *error* terendah adalah parameter dengan nilai $N_p = 3$ dan $N_c = 1$.
4. Pengendalian MPC hasil proses dipengaruhi oleh *cost function*, *constraint*, parameter N_p (*Prediction Horizon*) dan parameter N_c (*Control Horizon*).
5. Dari semua percobaan pengendalian MPC yang dilakukan robot berhasil bergerak sesuai dengan referensi lintasan yang diberikan yaitu membentuk pola segi empat dan segi tiga.

5.2 Saran

Saran untuk penelitian selanjutnya :

1. Perhitungan untuk lokalisasi pada robot menggunakan eksternal encoder untuk mengantisipasi terjadinya 'slip' pada motor.
2. Membuat jaringan khusus untuk melakukan komunikasi serial antara PC dengan robot untuk memaksimalkan penerimaan dan pengiriman data antara robot dengan PC.

DAFTAR PUSTAKA

- [1] F. Kühne, W. F. Lages, and J. M. Gomes da Silva Jr, “Model Predictive Control of a Mobile Robot Using Linearization,” *Proc. Mechatronics Robot. 2004, Aachen, Ger.*, pp. 525–530, 2004.
- [2] M. Jaffery, S. Aslam, M. Ghauri, M. S. Khuram, S. Rafiq, and S. Khan, “Real-Time Implementation of Model Predictive Control on A 16-bit Microcontroller for Speed Control of A DC Motor,” *J. Eng. Technol.*, vol. 6, no. 1, pp. 415–434, 2018.
- [3] P. Zometa, M. Kogel, T. Faulwasser, and R. Findeisen, “Implementation Aspects of Model Predictive Control for Embedded Systems,” *2012 Am. Control Conf.*, no. June, pp. 1205–1210, 2014, doi: 10.1109/acc.2012.6315076.
- [4] F. Kühne, J. Gomes, and W. Fetter, “Mobile Robot Trajectory Tracking Using Model Predictive Control,” *II IEEE Latin-American Robot. Symp.*, pp. 1–7, 2005.
- [5] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, “IMU-based localization and slip estimation for skid-steered mobile robots,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2845–2850, 2007, doi: 10.1109/IROS.2007.4399477.
- [6] A. Curran and K. J. Kyriakopolous, “Sensor-based self-localization for wheeled mobile robots,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 1, no. January 1993, pp. 8–13, 1993, doi: 10.1109/robot.1993.291954.
- [7] S. Kim, C. Hyun, Y. Cho, and S. Kim, “Tracking control of 3-wheels omni-directional mobile robot using fuzzy azimuth estimator,” *Proc. 10th WSEAS Int. Conf. Robot. Control ManufProceedings 10th WSEAS Int. Conf. Robot. Control Manuf. Technol. ROCOM '10*, vol. 11, no. 10, pp. 47–51, 2010.
- [8] R. W. Sudiby, A. S. Khalilullah, and H. Mahmudah, “Pemetaan Posisi dan Sistem Navigasi Mobile Robot Dalam Ruang Menggunakan Sensor Perpindahan Jenis Optical Laser,” *Rolay instiute Technol.*, no. January 2011, pp. 1–5, 2001.
- [9] I. Control, *Discrete-time MPC with constraints*, no. 9781848823303. 2009.



LAMPIRAN

