

**ALGORITMA *COSINUS* SEBAGAI SOLUSI KEADAAN AWAL PADA
KENDALI FORMASI BERDASARKAN JARAK**

TESIS

**TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL DAN
ELEKTRONIKA**

**Ditujukan untuk memenuhi persyaratan
memperoleh gelar Magister Teknik**



**ANGGORO DWI NUR ROHMAN
186060300111007**

**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
Juli 2021**



UNIVERSITAS BRAWIJAYA

**ALGORITMA *COSINUS* SEBAGAI SOLUSI KEADAAN AWAL PADA
KENDALI FORMASI BERDASARKAN JARAK**

TESIS

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Teknik**

ANGGORO DWI NUR ROHMAN

186060300111007

**FAKULTAS TEKNIK
TEKNIK ELEKTRO KONSENTRASI TEKNIK
KONTROL DAN ELEKTRONIKA**

MALANG

Juli 2021

TESIS

ALGORITMA *COSINUS* SEBAGAI SOLUSI KEADAAN AWAL PADA KENDALI FORMASI BERDASARKAN JARAK

ANGGORO DWI NUR ROHMAN

NIM. 186060300111007

telah dipertahankan di depan penguji
pada Tanggal **19 Juli 2021**
dinyatakan telah memenuhi syarat
untuk memperoleh gelar **Magister Teknik**

Komisi Pembimbing

Pembimbing I



M. Aziz Muslim, S.T., M.T., Ph.D.

Pembimbing II



Dr. Ir. Bambang Siswojo, M.T.

Malang,
Universitas Brawijaya
Fakultas Teknik, Jurusan Teknik Elektro
Plt. Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.

NIP. 197412032000121001

PERNYATAAN ORISINALITAS TESIS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan Saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Tesis ini adalah asli dari pemikiran Saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Tesis ini dapat dibuktikan terdapat unsur-unsur jiplakan, Saya bersedia Tesis dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 tahun 2003, Pasal 25 Ayat 2 dan Pasal 70).

Malang, 28 Juli 2021

Mahasiswa,

Nama : **ANGGORO DWI NUR ROHMAN**

NIM : **186060300111007**

PM : **TEKNIK ELEKTRO**

PROGRAM MAGISTER TEKNIK ELEKTRO

IDENTITAS TIM PENGUJI TESIS

JUDUL TESIS: Algoritma *Cosinus* Sebagai Solusi Keadaan Awal Pada Kendali Formasi Berdasarkan Jarak.

Nama Mahasiswa : Anggoro Dwi Nur Rohman
NIM : 186060300111007
Program Studi : Magister Teknik
Minat : Kontrol dan Elektronika

KOMISI PEMBIMBING :

Ketua : M. Aziz Muslim, ST., M.T., Ph.D.
Anggota : Dr. Ir. Bambang Siswojo, M.T.
SK Pembimbing : 1586 Tahun 2019
Tanggal SK : 26 Juli 2019

TIM DOSEN PENGUJI :

Dosen Penguji 1 : Dr. Ir. Ponco Siwindarto, M.Eng.Sc
Dosen Penguji 2 : Dr. Ir. Dipl.Ing. Mochammad Rusli
Tanggal Ujian : 19 Juli 2021
SK Penguji : 1238 Tahun 2021

RIWAYAT HIDUP

Anggoro Dwi Nur Rohman, terlahir di Kota Malang di tanggal 26 Januari 1993 dari ayah yang bernama Agus Hidayat dan Ibu yang bernama Suryani. Saya telah menempuh sarja di Universitas Negeri Malang angkatan 2011 dan lulus di tahun 2015. Selama menempuh sarja saya aktif di organisasi *Robotic Team* dibawah naungan Jurusan Teknik Elektro dan mengikuti berbagai perlombaan yang diadakan oleh DIKTI dibidang perlombaan KRAI dan KRSI. Saya aktif sebagai Asisten Pengajar di Jurusan Teknik Elektro dan mengajar praktikum di Laboratorium Elektronika Digital dan Laboratorium Mikroprosesor Lanjut. Setelah menempuh sarjana, saya bekerja sebagai *Embedded Software Developer* di IDEMIA DKI Jakarta selama dua setengah tahun.

Malang, Juli 2021

Anggoro Dwi Nur Rohman

RINGKASAN

Anggoro Dwi Nur Rohman, Magister Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juli 2021, *Algoritma Cosinus Sebagai Solusi Keadaan Awal Pada Kendali Formasi Berdasarkan Jarak*, Dosen Pembimbing: M. Aziz Muslim, ST., M.T., Ph.D. dan Dr. Ir. Bambang Siswojo, M.T..

Kendali formasi adalah topik penelitian kendali multi-robot, dimana sekelompok robot dapat mencapai formasi tertentu dan mempertahankannya ketika berpindah ke arah yang diinginkan. Salah satu pengembangan kendali formasi adalah kendali formasi berdasarkan jarak dimana setiap individu robot menggunakan informasi jarak antara sesamanya untuk mencapai tujuan formasi. Banyak pengembangan yang dilakukan pada kendali formasi berdasarkan jarak menggunakan model yang sederhana dan membutuhkan pengembangan lebih lanjut untuk penerapan kendali ke model yang lebih nyata. Ketika penerapan kendali formasi berdasarkan jarak, terdapat permasalahan kondisi awal yaitu robot tidak dapat menentukan koordinat tetangganya. Penelitian ini akan mengembangkan algoritma cosinus sebagai solusi untuk kondisi awal kendali formasi berdasarkan jarak. Algoritma cosinus terinspirasi dari rumus segitiga sederhana dan mengharuskan robot melakukan dua langkah saja untuk dapat menemukan koordinat tetangganya. Hasil percobaan simulasi, kendali formasi berdasarkan jarak menggunakan tiga model robot holonomic dan penerapan algoritma cosinus membutuhkan waktu rata-rata 6.5 detik untuk menemukan koordinat tetangganya.

Kata Kunci: Kendali Formasi, Multi-Robot, Algoritma Cosinus, Mobile Robot.

SUMMARY

Anggoro Dwi Nur Rohman, *Master Electrical Engineering, Engineering University of Brawijaya, Juli 2021, Formation Control Distance-Based Using Cosinus Algorithm For Multi Mobile-Robot, Academic Supervisor : M. Aziz Muslim, ST., M.T., Ph.D. and Dr. Ir. Bambang Siswojo, M.T..*

Formation control is a research topic of multi-robot control, where a group of robots can reach a certain formation and defend it when moving in the desired direction. One of the developments is distance-based where formation goals achieved using the distance between each other only. Many developments are using a simple model and need further development into a realistic model. When applying distance-based, there is a problem in the initial condition, namely that the robot cannot find the coordinates of its neighbors when using only distance. In this work, the cosine algorithm was developed as a solution to the initial conditions which are inspired by a simple triangle formula and need only two steps to find the coordinates. From simulation experiment results, distance-based formation control using three holonomic robot models and the application of the cosine algorithm takes an average of 6.5 seconds to find the coordinates of its neighbors.

Keywords:Formation Control, Multi-robot, Cosine Algorithm, Mobile Robot.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penulisan Tesis dengan judul “Algoritma *Cosinus* Sebagai Solusi Keadaan Awal Pada Kendali Formasi Berdasarkan Jarak”. Penulis mengucapkan terimakasih kepada pihak-pihak yang membantu proses penulisan Tesis ini hingga selesai.

1. Agus Hidayat, sebagai Ayah dan pemimpin keluarga saya.
2. Suryani, sebagai Ibu dan penopang semangat di keluarga saya.
3. M. Aziz Muslim, ST., M.T., Ph.D., selaku Ketua Dosen Pembimbing.
4. Dr. Ir. Bambang Siswojo, M.T., selaku Anggota Dosen Pembimbing.
5. Dr.Eng. Panca Mudjirahardjo, S.T., M.T, selaku Ketua Program Studi Magister Teknik Elektro Universitas Brawijaya
6. Teman-teman mahasiswa Magister Teknik Elektroyang telah bekerjasama dan memberikan informasi dalam penyusunan Tesis ini
7. Semua pihak yang membantu demi kelancaran penyelesaian Tesis ini.

Penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun sehingga dapat dijadikan koreksi untuk penyempurnaan Tesis ini. Semoga Tesis ini dapat bermanfaat sebagaimana mestinya.

Malang, Juli 2021

Anggoro Dwi Nur Rohman

DAFTAR ISI

Halaman Judul	i
Pengesahan	ii
Pernyataan Orisinalitas	iii
Identitas Tim Penguji	iv
Riwayat Hidup	v
Ringkasan	vi
<i>Summary</i>	vii
KATA PENGANTAR	viii
Daftar Isi	ix
Daftar Gambar	xi
Daftar Tabel	xii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi dan Perumusan Masalah	2
1.3 Tujuan dan Manfaat	3
2 KAJIAN PUSTAKA	4
2.1 Kendali Formasi Berdasarkan Jarak	4
2.1.1 Pendahuluan Formasi Multi Robot	4
2.1.1.1 Teori Graf	4
2.1.1.2 Teori Kekakuan Graf	4
2.1.2 Kendali Formasi Multi-Robot	5
2.2 Pemodelan Robot	7

3	KERANGKA KONSEP PENELITIAN	12
3.1	Definisi Permasalahan Kendali Formasi	13
3.2	Permasalahan dan Solusi	14
4	METODE PENELITIAN	16
4.1	Strategi Penentuan Koordinat Tetangga	16
4.2	Analisa Algoritama Dengan Tetangga Statis	18
4.3	Kendali Robot <i>Holonomic</i>	19
4.4	State Feedback	20
4.5	Desain Kendali	22
4.6	Kendali Formasi Multi Robot	25
5	HASIL DAN PEMBAHASAN	27
5.1	Pergerakan Robot Terhadap Algoritma <i>Cosinus</i>	27
5.2	Perbandingan Penerapan Algoritma <i>Cosinus</i>	28
5.3	Algoritma <i>Cosinus</i> Di Seluruh Kuadran	28
6	KESIMPULAN DAN SARAN	30
6.1	Kesimpulan	30
6.2	Saran	30
	DAFTAR REFERENSI	31

DAFTAR GAMBAR

2.1	Plot motion 3 robot dengan nilai $k_p=80$ dan $k_i=1$	7
2.2	(a) Geometri Robot (Correia, dkk,2012) (b) Grafik Gaya Robot . . .	8
3.1	Kerangka Penelitian	12
4.1	Strategi Penentuan Koordinat Langkah Pertama	17
4.2	Strategi Penentuan Koordinat Langkah Kedua	18
4.3	State-feedback Sistem	20
4.4	25
5.1	Grafik Pergerakan Robot Menggunakan Algoritma Cosinus danK- endali Formasi	27
5.2	Grafik Response Jarak Terhadap Waktu Pada Robot Dengan Kendali Formasi.	28

DAFTAR TABEL

4.1	Parameter model oleh Correia, dkk (2012)	22
5.1	Settling Time Dengan Konstanta Kp Yang Berbeda	29

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kendali formasi adalah topik penelitian kendali multi-robot untuk memecahkan permasalahan koordinasi pergerakan. Kendali formasi bertujuan untuk mengendalikan sekelompok robot dalam mencapai formasi tertentu dan dapat mempertahankan formasi tersebut ketika bermanuver menuju arah yang diinginkan. Penjabaran oleh Guanghai, dkk (2013), pengembangan kendali formasi dilakukan dari beberapa algoritma strategi. Seperti yang dikembangkan oleh Wang, dkk (2014) menggunakan strategi leader-follower, menggunakan Fuzzy-Logic sebagai tingkah laku robot oleh Ferik, dkk (2016) dan menggunakan struktur virtual dimana sekelompok robot memiliki titik referensi sebagai satu robot oleh Li, dkk (2015).

Penjabaran oleh Oh, dkk (2015), bahwa dari berbagai pengembangan kendali formasi dapat diambil garis besar menjadi tiga bagian, yaitu berdasarkan posisi, perpindahan, dan jarak. Ketiga bagian tersebut tertuju pada jawaban dari pertanyaan, "variabel apa yang digunakan sebagai sensor" dan "variabel apa yang aktif dikendalikan oleh sistem multi-robot untuk mencapai formasi yang diinginkan". Kendali formasi berdasarkan posisi adalah metode kendali formasi dimana robot diharuskan memiliki kemampuan untuk mengetahui kordinatnya sendiri berdasarkan koordinat global. Kerena itu kendali berdasarkan posisi membutuhkan sensor posisi seperti GPS. Kendali formasi berdasarkan perpindahan adalah kendali formasi dimana setiap robot tidak mengetahui koordinatnya berdasarkan koordinat global dikarenakan variabel yang digunakan sebagai sensor adalah kecepatan terhadap tetangganya. Metode berdasarkan posisi, robot membutuhkan kemampuan penyesuaian orientasi koordinat global sehingga setiap robot dibutuhkan sensor kompas untuk menyearhkannya. Formasi berdasarkan jarak adalah kendali formasi dimana variabel yang dikendalikan adalah variabel jarak antar robot yang terhubung sehingga koordinat yang digunakan tidak mengacu pada koordinat global. Penerapan formasi berdasarkan jarak menggunakan sensor yang lebih sedikit dibanding dengan posisi dan perpindahan. Namun pembahasan model yang lebih nyata untuk diterapkan kendali formasi berdasarkan jarak masih sedikit. Pengembangan formasi berdasarkan jarak telah dikembangkan menggunakan teori graph dengan model single dan double integrator sederhana oleh Oh, dkk (2014), menggunakan informasi

jarak untuk mengendalikan model sederhana double integrator oleh Cai, dkk (2014), menggunakan konsensus antara robot untuk mencapai bentuk formasi dengan model single integrator sederhana oleh Deghat, dkk (2016), mengendalikan bentuk formasi menggunakan adaptive control untuk mengestimasi kecepatan tetangga dari model robot yang sederhana oleh Kang, dkk (2014), menggabungkan kendali formasi berdasarkan jarak dan perpindahan untuk mengendalikan model robot yang sederhana oleh Park, dkk (2015), dan menggunakan kendali Proportional-Integral (PI) untuk mengendalikan jarak setiap model robot yang sederhana oleh Rozenheck, dkk (2015).

Kendali PI pada penelitian oleh Rozenheck, dkk (2015) tidak dapat langsung diterapkan menggunakan sensor jarak karena kendali tersebut mengambil informasi jarak menggunakan selisih dari koordinat kartesian global setiap robot. Sedangkan dalam praktiknya robot hanya bisa mengukur jarak dan tidak mengetahui koordinat dari robot tetangganya. Sudah umum penelitian dibidang lokalisasi menggunakan sensor jara seperti yang dibahas oleh Guo, dkk (2020) yang menggantikan sensor berbasis vision untuk mendapatkan koordinat dari beberapa robot, penelitian oleh Qiang, dkk (2017) membutuhkan beberapa sensor jarak yang terpasang statis digunakan untuk mengetahui koordinat sekelompok robot dan mengendalikannya secara terpusat, dan pengembangan Qiang, dkk (2018) dengan memasang dua sensor jarak di salah satu dari sekelompok robot, lalu mendistribusikan koordinat ke robot tetangganya. Lokalisasi menggunakan dua sensor jarak tersebut memanfaatkan rumus segitiga untuk mendapatkan koordinat robot tetangganya. Penelitian ini akan mengembangkan algoritma memanfaatkan rumus segitiga yang dikembangkan oleh Qiang, dkk (2018) dan menggantikan salah satu robot yang memiliki dua sensor dengan algoritma tersebut sehingga tidak mengharuskan salah satu robot memiliki dua sensor. Penelitian oleh Rozenheck, dkk (2015) mengembangkan kendali formasi menggunakan model holonomic sederhana dimana robot bergerak ke suatu arah tidak bergantung dari kondisi awal arah robot dan juga model sederhana tidak mempertimbangkan parameter fisik pada modelnya. Percobaan akan menggunakan tiga model robot holonomic dengan harapan menjadi langkah awal untuk mengembangkan kendali formasi berdasarkan jarak menggunakan model robot yang lebih nyata.

1.2 Identifikasi dan Perumusan Masalah

Pembahasan dari latar belakang dapat ditentukan pontensi permasalahan sebagai identifikasi masalah, yaitu kendali formasi berdasarkan jarak tidak dapat langsung

diterapkan menggunakan sensor jarak karena kendali formasi mengambil informasi jarak menggunakan selisih dari koordinat kartesian global setiap robot. Dalam penelitian ini akan digunakan batasan-batasan permasalahan sebagai berikut :

1. Kendali formasi berdasarkan jarak akan menggunakan tiga robot yang dijalankan secara simulasi.
2. Sensor jarak bekerja secara ideal.
3. Strategi yang akan dikembangkan hanya untuk mengetahui koordinat kondisi awal saja.

Dapat ditentukan permasalahan untuk penelitian ini, yaitu ”bagaimanakah strategi untuk mendapatkan koordinat kondisi awal kendali formasi apabila variabel yang dikendalikan adalah jarak antar robot?.

1.3 Tujuan dan Manfaat

Tujuan dari penelitian ini adalah mengetahui strategi untuk mendapatkan koordinat kondisi awal kendali formasi apabila variabel yang dikendalikan adalah jarak antar robot. Manfaat dari penelitian ini adalah memberikan referensi untuk permasalahan kendali multi-robot, khususnya pada permasalahan kendali formasi, terhadap model yang lebih nyata. dan membuka peluang penelitian dibidang kendali mengenai kendali formasi pada kendali multi-robot dilingkungan Fakultas Teknik Elektro, Universitas Brawijaya.

BAB 2

KAJIAN PUSTAKA

2.1 Kendali Formasi Berdasarkan Jarak

Pembahasan kendali formasi mutli-robot dikutip dari paper oleh Rozenheck, dkk (2015). Dimana peneliti membahas mengenai kendali formasi robot berdasarkan jaraknya lalu dikendalikan dengan kendali PI. Dari subbab ini akan dirangkum dari paper tersebut, yaitu mulai dari pendahuluan sampai kendalinya.

2.1.1 Pendahuluan Formasi Multi Robot

2.1.1.1 Teori Graf

Suatu graf \mathcal{G} , dinotasikan sebagai $\mathcal{G} = (\mathcal{V}, \varepsilon)$, merupakan pasangan \mathcal{V} dan ε , di mana \mathcal{V} merupakan himpunan tak kosong berisikan simpul pada graf tersebut dan ε merupakan himpunan sisi pada graf tersebut. Secara formal, himpunan ε dapat dinyatakan sebagai suatu koleksi subhimpunan berkardinalitas dua dari himpunan \mathcal{V} , atau dalam notasi matematika $\varepsilon \subseteq \mathcal{V} \times \mathcal{V}$. Sebuah \mathcal{G} dikatakan tak berarah (*undirected graph*), dimana himpunan sisi terdiri dari pasangan node (i, j) , maka sisi tersebut tidak memiliki urutan arah antara node i dengan j . Dinotasikan $n \triangleq |\mathcal{V}|$ sebagai jumlah dari node dan $m \triangleq |\varepsilon|$ sebagai jumlah dari sisinya. Apabila $(i, j) \in \varepsilon$ maka dapat disebut node i dan j berdekatan (*adjacent*). Himpunan dari node yang terhubung dari setiap simpul i dinotasikan dengan $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i, j) \in \varepsilon\}$, dan juga $i \sim j$. Matiks insidensi (*incidence*), $E \in \mathbb{R}^{n \times m}$, adalah matrik $\{0, \pm 1\}$ dimana baris matrik mengindikasikan simpulnya dan kolomnya sebagai sisinya. Matriks *laplacian* didefinisikan dengan $L(\mathcal{V}) = EE^T$

2.1.1.2 Teori Kekakuan Graf

Koordinat multi dimensi adalah konfigurasi matrik vector yang terdiri dari beberapa koordinat node, $p = [x_1^T \ \dots \ x_n^T]^T \in \mathbb{R}^{2n}$, dimana $x_i \in \mathbb{R}^2$ dan $x_i \neq x_j$ untuk semua $i \neq j$. Dfinisi sebuah kerangka (*framework*), dinotasikan dengan $\mathcal{G}(p)$, adalah graf tak berarah \mathcal{G} dengan konfigurasi x , dimana simpul i pada graf dipetakan kedalam koordinat x_i . Misalkan $(i, j) \in \varepsilon$ sama dengan sisi ke k dari graf langsung dan mendefinisikan vektor sisi dari kerangka, atau dapat disebut sebagai vektor posisi relatif, dengan $e_k \triangleq x_j - x_i$. Untuk semua vektor sisi dapat dinotasikan dengan

$$e = \begin{bmatrix} e_1^T & \dots & e_m^T \end{bmatrix} \in \mathbb{R}^{2m}.$$

Apabila kerangka $\mathcal{V}(p)$ dengan vektor sisi $\{e_k\}_{k=1}^m$, maka didefinisikan fungsi sisi (*edge function*), $F : \mathbb{R}^{2n} \times \mathcal{V} \rightarrow \mathbb{R}^m$ dengan

$$F(p, \varepsilon) \triangleq \left[\|e_1\|^2 \quad \dots \quad \|e_m\|^2 \right]^T \quad (2.1)$$

Matrik kekakuan $R(p)$ yang berhubungan erat dengan kerangka $\mathcal{G}(p)$ dapat didefinisikan dengan *Jacobian* dari fungsi sisi (Rozenheck, dkk (2015)),

$$\begin{aligned} R(p) &\triangleq \frac{\partial F(p, \mathcal{G})}{\partial x} \in \mathbb{R}^{m \times 2n} \\ &\triangleq \text{diag}(e_i^T)(E^T \otimes I_2) \end{aligned} \quad (2.2)$$

2.1.2 Kendali Formasi Multi-Robot

Dikarenakan pada penelitian sebelumnya menggunakan model sederhana, maka akan menggabungkan literatur oleh Oh, dkk (2014) dengan penyesuaian metode sebelumnya.

Pembahasan kendali dari formasi multi robot menggunakan gradient control. Apabila $n(n \geq 2)$ dimodelkan sebagai titik yang memiliki masa jenis bergerak diatas dimensi 2 (*Euclidean Space*), maka pergerakan dimodelkan dengan

$$\begin{cases} \dot{x}_i(t) = v_i(t) & i = 1, \dots, n. \\ \dot{v}_i(t) = u_i(t) \end{cases} \quad (2.3)$$

dimana $v_i(t) \in \mathbb{R}^2$ dan $u_i(t) \in \mathbb{R}^2$ adalah kecepatan dan input dari robot- i . Dinotasikan $d \in \mathbb{R}^m$ adalah vector jarak dimana isi dari matrik tersebut adalah d_k^2 yang mempresentasikan jarak yang diinginkan antara setiap robot i dan j untuk sisi $(i, j) \in \varepsilon$. Lalu didefinisi persamaan potensial yang memiliki hubungan antara jarak robot yang diinginkan dengan jarak yang sekarang

$$\Phi(e) = \frac{1}{2} \sum_{i \in \mathcal{V}} \|v_i\|^2 + \frac{1}{2} \sum_{k=1}^m (\|e_k\|^2 - d_k^2)^2 \quad (2.4)$$

Pengamatan dilakukan agar $\Phi(e) = 0$ jika dan hanya jika $\|e_k\|^2 = d_k^2, \forall k = 1, \dots, m$.

Kendali dari setiap robot menggunakan gradien negatif dari fungsi potensial

$$\begin{cases} \dot{p} &= \frac{\partial \Phi(e)}{\partial v} + Bv_{ref} \\ &= v(t) + Bv_{ref} \\ \dot{v} &= -C \left(\frac{\partial \Phi(e)}{\partial v} + \frac{\partial \Phi(e)}{\partial p} \right) \\ &= -C(v(t) + R(p)^T (R(p)x(t) - d)) \\ &= u(t) \end{cases} \quad (2.5)$$

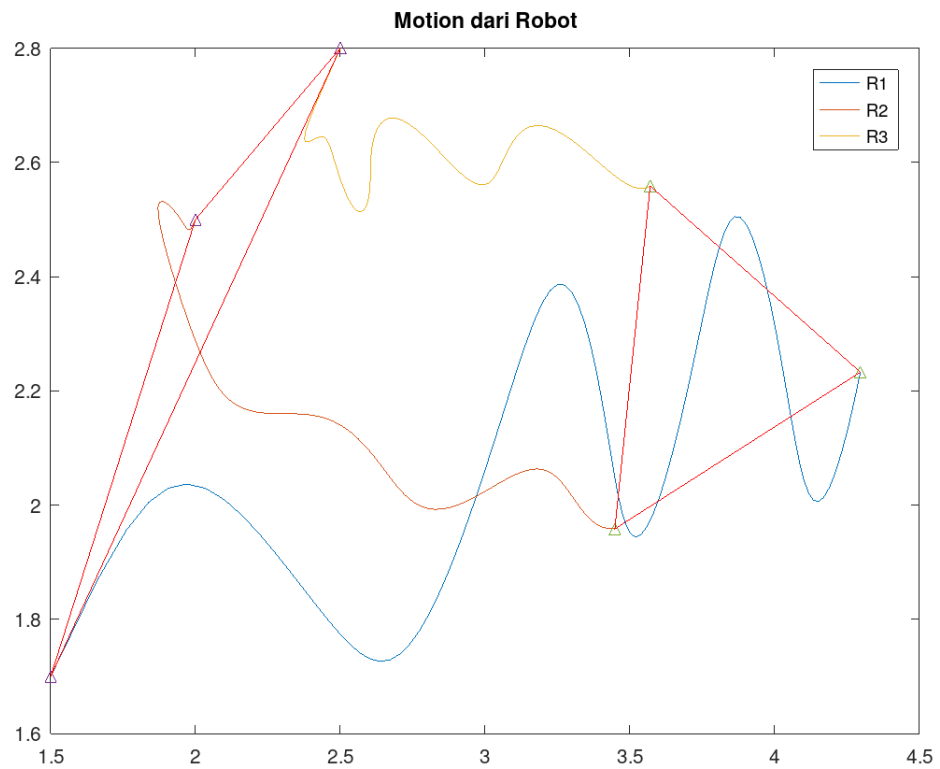
Dimana $v \in \mathbb{R}^{2n}$ adalah vector kecepatan dari seluruh node. Penambahan referensi kecepatan pada salah satu robot dapat menjadikan formasi bermanuver. Dimana $B \in \mathbb{R}^{2n \times 2}$ digunakan untuk indikasi robot ke i sebagai leader atau penerima kecepatan referensinya, $v_{ref} \in \mathbb{R}^2$ sebagai kecepatan referensi, dan C adalah konstanta pengendali yang akan digantikan dengan algoritma kendali. Dengan menerapkan kendali *Proportional*, konstanta C pada persamaan (2.6) dapat diubah dengan

$$\begin{cases} u_i(t) &= u_{k_{p1}}(t) + u_{k_{p2}}(t) \\ u_{k_{p1}}(t) &= -k_{p1}v_i(t) \\ u_{k_{p2}}(t) &= -R(p(t))^T k_{p2}(R(p(t))x_1(t) - d) \end{cases} \quad (2.6)$$

Dengan itu dapat digabungkan menjadi persamaan *state-space* menggunakan persamaan (2.5)

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_{p2}R(x)^T R(x) & -k_{p1} \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ k_{p2}R(x)^T \end{bmatrix} d + Bv_{ref}$$

$$\dot{x}_f(t) = A_f(x)x_f(t) + B_f(x)d + Bv_{ref} \quad (2.7)$$

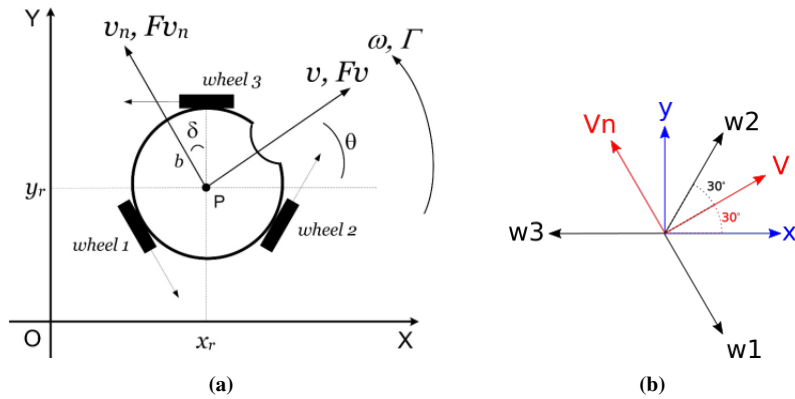


Gambar 2.1: Plot motion 3 robot dengan nilai $k_p=80$ dan $k_i=1$

2.2 Pemodelan Robot

Pembahasan pemodelan robot akan dirangkum dari jurnal oleh Correia, dkk (2012).

Robot menggunakan 3 aktuator penggerak dengan roda *omniwheel*, sehingga robot dapat bergerak kesegala arah. Pemasangan roda *omniwheel* memiliki sudut 120° terhadap roda lainnya. Sehingga setiap roda memiliki gaya dengan arah 90° dari sudut pemasangannya. Agar robot bergerak kesegala arah, ketiga aktuator harus dikendalikan untuk menghasilkan resultan gaya dengan arah yang diinginkan.



Gambar 2.2: (a) Geometri Robot (Correia, dkk,2012) (b) Grafik Gaya Robot

Kinematika robot dapat dirumuskan menjadi

$$\dot{\mathbf{x}}_p = R^T(\theta)\dot{\mathbf{x}}_r \quad (2.8)$$

dimana $R(\theta)$ adalah matrik rotasi ortogonal

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Koordinat robot dideskripsikan menggunakan vector $\mathbf{x}_p = [x_p \ y_p \ \theta]^T$, dimana x_p dan y_p adalah titik pusat, P , pada frame robot dan θ_p adalah selisih sudut antara *angular* frame global dengan robot. Vector $\dot{\mathbf{x}}_r = [\dot{x}_r \ \dot{y}_r \ \dot{\theta}_r]^T$ mendeskripsikan variable kecepatan terhadap titik pusat, P , dimana w sebagai kecepatan angular robot terhadap frame global. Karena robot memiliki aktuator, maka kecepatan roda memiliki hubungan terhadap kecepatan robot, dengan kata lain kecepatan pada titik pusat adalah sebuah fungsi dengan kecepatan roda sebagai parameternya. Untuk mendapatkan persamaan tersebut, maka dapat dianalisis dengan hukum *Power*. Apabila didefinisi hubungan antara gaya resultan robot dengan gaya yang dihasilkan roda

$$F_{\dot{x}_r} = F_{w1}(t) \cos(90^\circ) + F_{w2}(t) \cos(30^\circ) + -F_{w3}(t) \cos(30^\circ)$$

$$F_{\dot{y}_r} = -F_{w1}(t) + F_{w2}(t) \cos(60^\circ) + F_{w3}(t) \cos(60^\circ)$$

$$\Gamma = d.F_{w1}(t) + d.F_{w2}(t) + d.F_{w3}(t)$$

dimana d adalah jarak dari titik P ke lokasi roda, maka akan didapat matriks ge-

ometri antara $F_R = [F_{\dot{x}_r} \ F_{\dot{y}_r} \ \Gamma]^T$ dengan $F_w = [F_{w1} \ F_{w2} \ F_{w3}]^T$

$$F_R = \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ -1 & \frac{1}{2} & \frac{1}{2} \\ l & l & l \end{bmatrix} F_w$$

$$F_R = A F_w \quad (2.9)$$

Dalam kasus robot, *power* yang dihasilkan oleh setiap roda sama dengan *power* dari robot itu sendiri (Hacene, dkk,2019). Dengan menggunakan persamaan (2.9) akan menghasilkan persamaan kinematika robot menggunakan 3 roda *omniwheel*

$$P_w = P_R$$

$$F_w^T \dot{x}_w = F_R^T \dot{\mathbf{x}}_r$$

$$F_w^T \dot{x}_w = (A \cdot F_w)^T \dot{\mathbf{x}}_r$$

$$\dot{x}_w = A^T \dot{\mathbf{x}}_r$$

$$\dot{\mathbf{x}}_r = (A^T)^{-1} \dot{x}_w \quad (2.10)$$

dengan mensubstitusi persamaan (2.10) pada (2.8)

$$\dot{\mathbf{x}}_p = R^T(\theta)(A^T)^{-1} \dot{x}_w \quad (2.11)$$

Pergerakan robot juga dideskripsikan secara dinamika menggunakan hukum pergerakan dari *Newton*.

$$F_{\dot{x}_r}(t) - B_{\dot{x}_r} \dot{x}_r(t) - C_{\dot{x}_r} \text{sgn}(\dot{x}_r(t)) = M \ddot{x}_r(t) \quad (2.12)$$

$$F_{\dot{y}_r}(t) - B_{\dot{y}_r} \dot{y}_r(t) - C_{\dot{y}_r} \text{sgn}(\dot{y}_r(t)) = M \ddot{y}_r(t) \quad (2.13)$$

$$\Gamma(t) - B_{\dot{\theta}} \dot{\theta}(t) - C_{\dot{\theta}} \text{sgn}(\dot{\theta}(t)) = I \ddot{\theta}(t) \quad (2.14)$$

Dimana B_i adalah *viscous frictions* yang mempresentasikan perbandingan terbalik dari gaya yang bersifat linier terhadap gaya dorong dan kecepatan robot. $C_i \cdot \text{sgn}(i)$ adalah *coulumb frictions* yang mempresentasikan perbandingan terbalik terhadap perubahan kecepatan, dimana tanda bilangan berubah kebalikan dari kecepatannya.

$$\text{sgn}(\alpha) = \begin{cases} 1, & \alpha > 0 \\ 0, & \alpha = 0 \\ -1, & \alpha < 0 \end{cases}$$

seperti pada persamaan (2.9), resultan gaya robot berhubungan dengan gaya roda. Maka gaya roda dapat dideskripsikan dengan menghubungkan antara gaya yang dihasilkan oleh motor

$$F_{wi} = \frac{\tau_i(t)}{r_i} \quad (2.15)$$

dimana $\tau_i(t)$ adalah torsi dari motor

$$\tau_i(t) = l_i K_{ti} i_{ai}(t) \quad (2.16)$$

Untuk mendapatkan persamaan $i_{ai}(t)$, dapat digunakan deskripsi persamaan dinamika motor

$$u_i(t) = L_{ai} \frac{di_{ai}(t)}{dt} + R_{ai} i_{ai}(t) + K_{vi} \omega_{mi} \quad (2.17)$$

dimana L_{ai} dan R_{ai} adalah Induktansi dan resistansi armature motornya. K_{vi} adalah konstanta torsi motor dimana dalam satuan SI yang sama dengan K_v . Dalam praktiknya apabila motor dalam kecepatan *steady state* maka $\frac{di_{ai}}{dt}$ bernilai kecil, dan dalam persamaan (2.17) nilai induktansi dapat diabaikan.

Penjabaran dinamika robot bisa diubah dalam bentuk *state-space*

$$\dot{x}(t) = A_r x(t) + B_r u(t) + ksgn(x(t)) \quad (2.18)$$

$$y(t) = Cx(t) \quad (2.19)$$

dimana vektor *state* adalah $x(t) = [x_p \ y_p \ \theta \ \dot{x}_r \ \dot{y}_r \ \dot{\theta}_r]^T$, dan vektor output $y(t) = [x_p \ y_p \ \theta]^T$. Dimana $l = l_{1...3}$, $r = r_{1...3}$, $R_a = R_{a1...3}$ and $K_t = K_{t1...3}$, maka

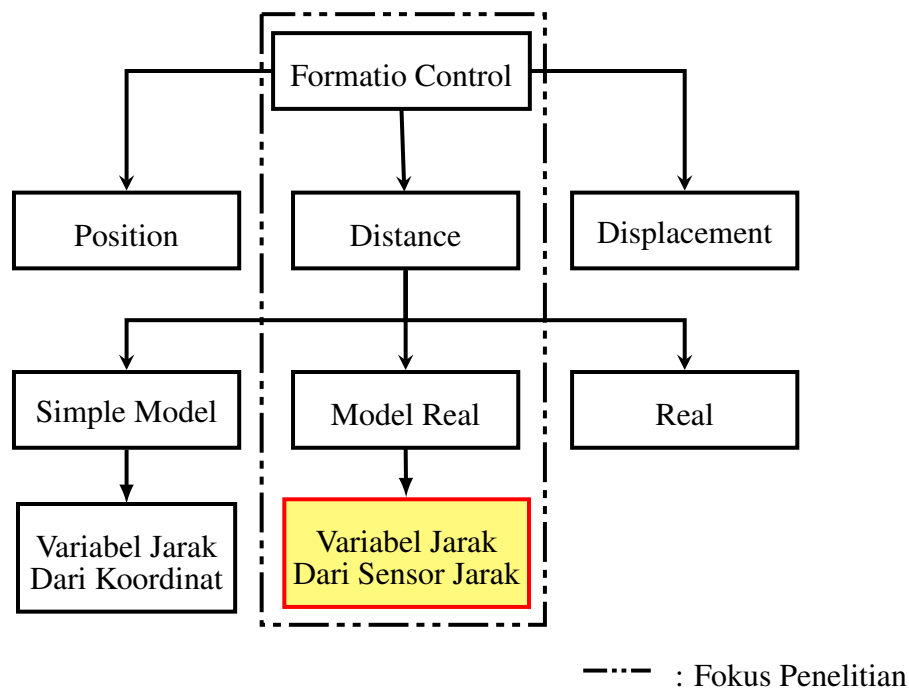
didapat matriks yang dapat mendeskripsikan sistem robot

$$\begin{aligned}
 A_r &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{3.l^2.K_t^2}{2.M.R_a.r^2} - \frac{B_{\dot{x}_r}}{M} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{3.l^2.K_t^2}{2.M.R_a.r^2} - \frac{B_{\dot{y}_r}}{M} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{3.l^2.K_t^2}{2.I.R_a.r^2} - \frac{B_{\dot{\theta}_r}}{I} \end{bmatrix}, \\
 B_r &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{l.K_t}{R_a.r} \frac{\cos(30^\circ)}{M} & -\frac{l.K_t}{R_a.r} \frac{\cos(30^\circ)}{M} \\ \frac{l.K_t}{R_a.r} \frac{-1}{M} & \frac{l.K_t}{R_a.r} \frac{\cos(60^\circ)}{M} & \frac{l.K_t}{R_a.r} \frac{\cos(60^\circ)}{M} \\ \frac{l.K_t}{R_a.r} \frac{b}{I} & \frac{l.K_t}{R_a.r} \frac{b}{I} & \frac{l.K_t}{R_a.r} \frac{b}{I} \end{bmatrix}, k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_{\dot{x}_r}}{M} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{C_{\dot{y}_r}}{M} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{C_{\dot{\theta}_r}}{M} \end{bmatrix}, \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

BAB 3

KERANGKA KONSEP PENELITIAN

Kerangka konsep penelitian akan dibahas mengenai potensi permasalahan yang timbul dalam topik kendali formasi. Kerangka penelitian ini berdasarkan literatur oleh Oh, dkk (2015), dimana didalam literatur tersebut, peneliti menguraikan berbagai metode yang digunakan dalam bidang kendali multi-robot, khususnya dalam kendali formasi. Kerangka penelitian akan digabungkan dengan permasalahan penerapan kendali formasi berdasarkan jarak oleh Rozenheck Rozenheck, dkk (2015), dimana kendali formasi tidak dapat langsung diterapkan menggunakan sensor jarak secara langsung.



Gambar 3.1: Kerangka Penelitian

Kendali formasi dikategorikan menjadi 3 bagian, yaitu berdasarkan posisi, perpindahan, dan jarak.

Pada formasi berdasarkan posisi, dimana agent diharuskan memiliki kemampuan untuk mengetahui koordinatnya sendiri berdasarkan koordinat global. Sehingga, koordinat tujuan didistribusikan kepada setiap agent dan agent bekerja untuk mencapai koordinat tersebut. Karena itu, kebutuhan individu untuk berinteraksi dengan individu lain sangat kecil. Metode formasi ini pada praktiknya, interaksi

antar individu dilakukan untuk menangani masalah disturbance, saturasi akselerasi, dan lain-lain. Karena metode ini membutuhkan kemampuan untuk mengetahui koordinat global, dibutuhkan biaya yang lebih dibanding metode lain dalam perangkat sensor yang *advance*, seperti sensor GPS;

Pada formasi kendali berdasarkan perpindahan, secara individu agent tidak mengetahui koordinatnya berdasarkan koordinat global. Akan tetapi, individu agent memiliki koordinatnya sendiri terhadap individu agent tetangganya dan harus dilakukan penyearahan terhadap koordinat setiap robot dengan koordinat global. Koordinat relatif itulah yang menjadi variable yang dikendalikan oleh agent. Oleh karena itu agent diharuskan memiliki kemampuan untuk mengetahui perpindahan dari individu lain berdasarkan koordinat agent itu sendiri, dan semua agent harus menyearahkan koordinatnya berdasarkan koordinat global, serta dibutuhkan interaksi antara individu lain untuk mencapai formasi yang diinginkan. Permasalahan pada metode ini ditujukan pada kendali formasi pada agent yang bersifat heterogent, pemeliharaan dalam komunikasi, dan kemampuan dalam menghindari rintangan;

Pada formasi berdasarkan jarak, dimana setiap individu agent memiliki koordinatnya masing-masing dan tidak perlu disearahkan dengan koordinat global. Variable yang dikendalikan pada metode ini adalah variabel jarak antar agent yang terhubung, sehingga dibutuhkan kemampuan untuk agent saling berkomunikasi antar agent lain. Permasalah pada metode ini ditujukan pada analisa stabilitas secara general; Pentingnya dilakukan investigasi pada penerapan model yang lebih nyata. Pemeliharaan komunikasi juga menyumbang dalam permasalahan secara praktik, dan kemampuan untuk menghindari rintangan juga dibutuhkan.

Penelitian oleh Rozenheck Rozenheck, dkk (2015) mengembangkan kendali formasi berdasarkan jarak dengan bagus. Kendali formasi yang dikembangkan menggunakan kendali PI untuk mengendalikan beberapa robot menggunakan variabel jarak. Akan tetapi pada kendali formasinya mengubah informasi koordinat menjadi jarak dimana pada penerapannya robot tidak dapat mengetahui koordinat tetangganya.

3.1 Definisi Permasalahan Kendali Formasi

Kendali formasi adalah kendali multi-agent untuk mencapai suatu formasi yang diinginkan. Dapat diperhatikan dalam Gambar 3.1, dari berbagai metode tersebut dapat disimpulkan secara umum dalam 3 kategori. Yaitu berbasis posisi, pergerakan, dan jarak. Pembagian tersebut berdasarkan kemampuan sensor yang digunakan dan penggunaan komunikasi dalam metodenya. Dari ketiga kategori tersebut,

kendali formasi berbasis jarak sangat dibutuhkan pembahasan mengenai penerapan metode tersebut pada model yang nyata. Pembagian antara model yang simple, model yang nyata, dan secara praktik berdasarkan tingkat analisisnya. Pada model yang simple, analisis dilakukan untuk mengembangkan strategi kendali formasi saja. Dalam kenyataannya model yang digunakan memiliki kekurangan, seperti batas kerja kecepatan, batasan sensor, komunikasi dan batasan lain-lain yang mempengaruhi kendali formasi tersebut.

Oh, dkk (2015) menyatakan bahwa mayoritas dari hasil penelitian yang menggunakan pendekatan ini (*distance-based*) berfokus pada model agent dengan integrator-tunggal di suatu bidang datar. Gagasan model yang nyata memiliki manfaat ketika menginvestigasi karakteristik kendali secara mendasar, model agent yang lebih realistis (model yang nyata) perlu untuk dipelajari lebih lanjut untuk menambah kepraktisan metode kendali multi-robot, khususnya pada kendali formasi berdasarkan jarak. Dengan bertambahnya kepraktisan diharapkan dapat diterapkan dalam model yang real. Pada penelitian oleh Rozenheck Rozenheck, dkk (2015), kendali formasi berdasarkan jarak dikendalikan menggunakan kendali PI dan menghasilkan pergerakan yang baik. Dapat diperhatikan pada persamaan (2.3) bahwa peneliti menggunakan model yang simpel untuk mengembangkan kendali multi-robotnya. Penelitian oleh Rozenheck Rozenheck, dkk (2015) memperoleh variabel jaraknya menggunakan koordinat tetangganya dimana pada penerapannya robot tidak mengetahui koordinat akan tetapi berbentuk jarak yang diperoleh dari sensor jarak.

3.2 Permasalahan dan Solusi

Dapat diperhatikan pada persamaan (2.7), state yang digunakan membutuhkan koordinat relatif dari tetangganya. Akan tetapi pada batasan penelitian ini, sensor yang digunakan hanya memberikan jarak terhadap tetangganya. Sedangkan koordinat relatif yang digunakan adalah kartesian. Apabila yang diketahui adalah jarak maka koordinat yang bisa digunakan adalah polar. Sedangkan koordinat polar membutuhkan sudut antara agent dan tetangganya. Oleh karena itu dibutuhkan algoritma khusus untuk mendapatkan sudut tersebut.

Untuk mengembangkan algoritma tersebut, dapat menggunakan hukum *cosinus* segitiga untuk menentukan sudutnya. Dengan memanfaatkan komunikasi antar robot, maka robot dapat mengirimkan informasi state kecepatan kepada tetangganya. Sehingga informasi tersebut dapat digunakan untuk memantau koordinat relatif terhadap tetangganya. Akan tetapi state kecepatan tersebut membutuhkan ni-

lai inisialisasi. Nilai inisialisasi ini akan diperoleh menggunakan algoritma *cosinus*. Sehingga, harapanya adalah kendali multi-robot menggunakan kendali PI dapat digunakan sesuai batasan penelitian.

BAB 4

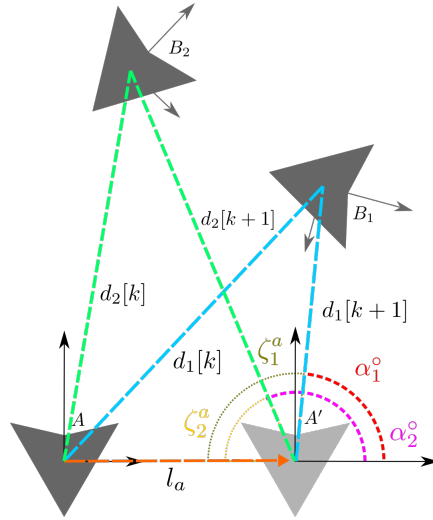
METODE PENELITIAN

Metode penelitian akan menerangkan tentang bagaimana strategi untuk menemukan koordinat tetangga menggunakan variabel jarak saja. Dapat diperhatikan pada persamaan (2.7) bahwa *state* yang dibutuhkan adalah kecepatan. Maka kendali robot diharuskan dapat mencapai kecepatan tertentu. Strategi pencarian koordinat mengharuskan juga robot untuk mencapai koordinat relatif tertentu. Pada bab ini akan menjelaskan bagaimana robot menemukan koordinat, mencapai kecepatan dan koordinat tertentu, dan membentuk formasi menggunakan model robot *holonomic*.

4.1 Strategi Penentuan Koordinat Tetangga

Tujuan dari strategi penentuan koordinat tetangga ini adalah menemukan koordinat menggunakan variabel jarak saja. Mengadopsi dari pengembangan oleh Qiang (Qiang, dkk (2018)) meletakkan dua sensor jarak pada salah satu robot, menggunakan kaidah rumus segitiga untuk mendapatkan sudut diantara robot tetangganya dan mengubahnya menjadi koordinat polar. Di penelitian ini menggunakan kaidah rumus segitiga tersebut akan tetapi robot hanya terpasang satu sensor jarak saja dan mengharuskan melakukan langkah tertentu untuk mendapatkan sudut diantara robot tetangganya.

Penentuan koordinat tetangga dapat ditemukan dengan mengubah koordinat polar menjadi koordinat kartesian. Koordinat polar membutuhkan panjang d_a , dan sudut α . Panjang d_a adalah variabel yang didapat dari sensor yang memberikan nilai jarak dari robot A ke robot B , akan tetapi untuk mendapatkan koordinat polar, pengukuran sudut α tidak tersedia. Algoritama yang ditawarkan memanfaatkan hukum *cosinus* pada segitiga untuk mendapatkan sudut tersebut.



Gambar 4.1: Strategi Penentuan Koordinat Langkah Pertama

Langkah pertama. Gambar 4.1 adalah Ilustrasi langkah pertama algoritma cosinus dimana robot A diharuskan berpindah sepanjang l_a atau ke koordinat $A' = (0, l_a)$ menggunakan kendali mode dua di Persamaan (4.9) akan tetapi robot harus menyimpan jarak di $[k]$ terlebih dahulu. Setelah berpindah robot A mendapatkan jarak di $[k+1]$ digunakan untuk menentukan sudut α_i^o menggunakan rumus segitiga cosinus. Berikut adalah Persamaan α_i^o .

$$\zeta_i^a = \cos^{-1} \left(\frac{l_a^2 + d_a[k+1]^2 - d_a[k]^2}{2d_a[k+1]l_a} \right) \quad (4.1)$$

$$\alpha_i^o = 180^\circ \pm \zeta_i^a \quad (4.2)$$

Variabel α_i^o dan $d_i[k+1]$ adalah nilai dari koordinat polar dari setiap robot tetangga A. Diubah menjadi koordinat kartesian untuk dapat dimasukkan dalam state kendali formasi.

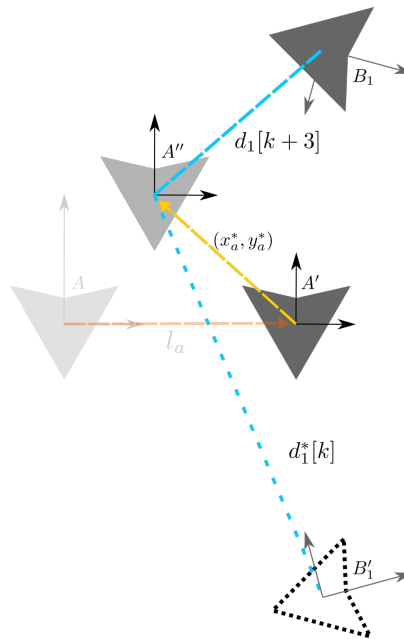
$$x_{B_i}^A = \begin{bmatrix} x_{B_i} = d_i[k] \cos \alpha_i^o \\ y_{B_i} = d_i[k] \sin \alpha_i^o \end{bmatrix} \quad (4.3)$$

Langkah kedua. Koordinat di Persamaan (4.3) akan menghasilkan bias dikarenakan Persamaan (4.2) tidak mengetahui letak kuadran sudutnya. Menggunakan ilustrasi di Gambar 4.1, langkah pertama menghasilkan dua kemungkinan koordinat robot B_1 dan B_1' . Apabila di Gambar 4.1, Sudut ζ_i^a adalah sudut segitiga $\angle AA'B_1$ atau $\angle AA'B_2$ sehingga dimungkinkan koordinat yang dihasilkan Persamaan (4.2) bisa berada pada kuadran 1 atau kuadran 4. Oleh karena itu di Persamaan 4.2 terdapat operasi \pm dimana operasi tersebut akan dilakukan berdasarkan letak kuadran

B_i .

$$\alpha_i^\circ = \begin{cases} 180^\circ - \zeta_i^a & , \text{Robot beradapadakuadran 1 dan 2} \\ 180^\circ + \zeta_i^a & , \text{Robot beradapadakuadran 3 dan 4} \end{cases} \quad (4.4)$$

Langkah kedua ini bertujuan untuk menentukan kejadian di Persamaan (4.4) dimana robot harus berpindah ke koordinat $A'' = (x^*_a, y^*_a)$ (Gambar 4.2). Sebelum robot berpindah, kondisi robot telah mendapatkan koordinat dari langkah pertama. Koordinat tersebut akan diubah menjadi jarak dan akan dibandingkan jarak tersebut dengan informasi jarak dari sensor setelah berpindah ke A'' . Apabila terdapat perbedaan maka kejadian di Persamaan (4.2) diubah kejadian selanjutnya dan mengkoreksi koordinat sebelumnya.



Gambar 4.2: Strategi Penentuan Koordinat Langkah Kedua

4.2 Analisa Algoritama Dengan Tetangga Statis

Telah dijelaskan pada Bab 4.1 bahwa robot bergerak ke arah yang random dengan jarak tertentu untuk mengetahui koordinat tetangga. Analisa akan dilakukan dengan membandingkan berbagai jarak dari tingkat rendah, sedang, dan tinggi untuk mengetahui respon algoritma yang sesuai dan optimal untuk mendapatkan koordinat tetangga.

Dari hasil analisa ini akan menghasilkan jarak terbaik untuk algoritma menentukan koordinat tetangga. Pembuktian akan dilakukan secara menampilkan grafik

respon.

4.3 Kendali Robot *Holonomic*

Kendali robot *holonomic* akan dibagi menjadi dua mode. Perbedaan kedua mode tersebut adalah *setpoint* kendalinya, dimana mode satu akan memiliki *setpoint* kecepatan robot sedangkan mode dua memiliki *setpoint* koordinat kerangka robot. Penjelasan lebih lengkap kegunaan dari kedua mode tersebut akan dijelaskan di pembahasan metode strategi penentuan koordinat.

Mode Satu. Kendali robot mode satu bertujuan untuk robot mencapai kecepatan yang diinginkan. Untuk mencapai tujuan tersebut akan menggunakan metode *state-feedback* seperti yang telah dijabarkan sebelumnya. Mengadopsi persamaan *state-space* dari robot.

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) + k_c \text{sgn}(x_c(t)), \quad (4.5)$$

$$y_c(t) = C_c x_c(t), \quad (4.6)$$

Dimana $A_c, B_c, k_c, C_c \in \mathbb{R}^{3 \times 3}$ adalah matriks parameter robot yang telah dijelaskan sebelumnya.

$$A_c = \begin{bmatrix} -\frac{3.l^2.K_t^2}{2.M.R_a.r^2} - \frac{B_{\dot{x}_r}}{M} & 0 & 0 \\ 0 & -\frac{3.l^2.K_t^2}{2.M.R_a.r^2} - \frac{B_{\dot{y}_r}}{M} & 0 \\ 0 & 0 & -\frac{3.l^2.K_t^2}{2.I.R_a.r^2} - \frac{B_{\dot{\theta}_r}}{I} \end{bmatrix},$$

$$B_c = \begin{bmatrix} 0 & \frac{l.K_t}{R_a.r} \cdot \frac{\cos(30^\circ)}{M} & -\frac{l.K_t}{R_a.r} \cdot \frac{\cos(30^\circ)}{M} \\ \frac{l.K_t}{R_a.r} \cdot \frac{-1}{M} & \frac{l.K_t}{R_a.r} \cdot \frac{\cos(60^\circ)}{M} & \frac{l.K_t}{R_a.r} \cdot \frac{\cos(60^\circ)}{M} \\ \frac{l.K_t}{R_a.r} \cdot \frac{b}{I} & \frac{l.K_t}{R_a.r} \cdot \frac{b}{I} & \frac{l.K_t}{R_a.r} \cdot \frac{b}{I} \end{bmatrix}, k_c = \begin{bmatrix} -\frac{C_{\dot{x}_r}}{M} & 0 & 0 \\ 0 & -\frac{C_{\dot{y}_r}}{M} & 0 \\ 0 & 0 & -\frac{C_{\dot{\theta}_r}}{M} \end{bmatrix},$$

$$C_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dimana $y_c(t) = x_c(t) = \begin{bmatrix} v & v_n & w \end{bmatrix}^T$ adalah vektor kecepatan. Lalu ditentukan fungsi kendali $u_c(t)$ menggunakan *state-feedback*.

$$u_c(t) = -K_c x_c(t) + N_c r_c \quad (4.7)$$

$$N_c = -[C(A_c - B_c K_c)^{-1} B_c]^{-1} \quad (4.8)$$

Dimana vektor $r_c = \begin{bmatrix} v^* & v_n^* & w^* \end{bmatrix}$ adalah *set-point* kendali mode satu dan matriks

K_c diperoleh dari solusi persamaan *Riccati*.

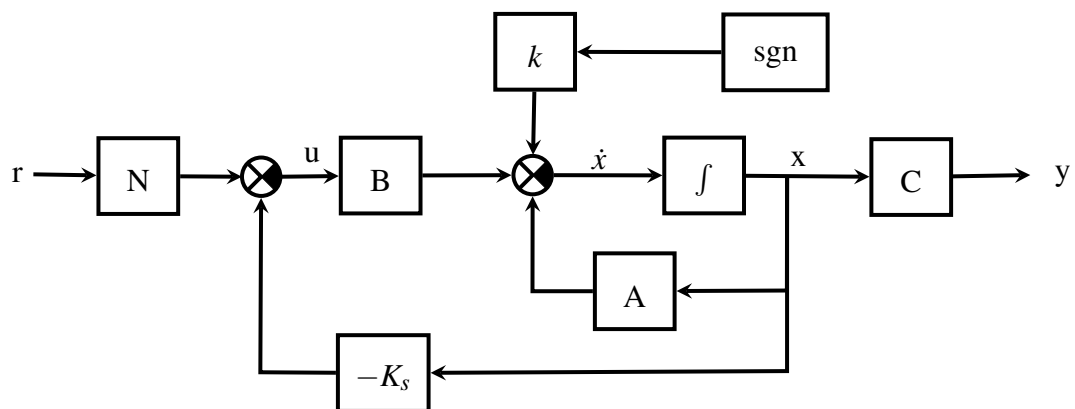
Mode dua. Kendali robot mode bertujuan untuk robot mencapai koordinat relatif yang diinginkan. Untuk mencapai tujuan tersebut akan menggunakan metode *state-feedback* seperti yang telah dijabarkan sebelumnya. Mengadopsi persamaan *state-space* dari robot Persamaan(2.18)-(2.19). Lalu ditentukan fungsi kendali $u(t)$ menggunakan *state-feedback*.

$$u(t) = -K_r x_c(t) + N_r r_r \quad (4.9)$$

$$N_r = -[C(A_r - B_r K_r)^{-1} B_r]^{-1} \quad (4.10)$$

Dimana vektor $r_c = [\dot{x}^*_r \quad y^*_r \quad \dot{\theta}^*_r \quad v^* \quad v_n^* \quad w^*]$ adalah *set-point* kendali mode dua dan matriks K_r diperoleh dari solusi persamaan *Riccati*.

4.4 State Feedback



Gambar 4.3: State-feedback Sistem

Pada persamaan (2.18) diketahui bahwa state memiliki dimensi 6×1 . Dimensi tersebut tidak menunjukkan sistem memiliki orde 6. Apabila diperhatikan orde dari sistem adalah orde 2. Dengan membaginya kedalam 3 persamaan *state-space* akan lebih mudah dalam analisis parameter kendalinya. Berikut adalah persamaannya.

$$\begin{bmatrix} \dot{x}_p \\ \ddot{x}_r \end{bmatrix} = \begin{bmatrix} 0 & A_{14} \\ 0 & A_{44} \end{bmatrix} \begin{bmatrix} x_p \\ \dot{x}_r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B_{11} & B_{12} & B_{13} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + k_{44} \text{sgn}(\dot{x}_r) \quad (4.11)$$

$$\begin{bmatrix} \dot{y}_p \\ \ddot{y}_r \end{bmatrix} = \begin{bmatrix} 0 & A_{25} \\ 0 & A_{55} \end{bmatrix} \begin{bmatrix} y_p \\ \dot{y}_r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B_{21} & B_{22} & B_{23} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + k_{55} \text{sgn}(\dot{y}_r) \quad (4.12)$$

$$\begin{bmatrix} \dot{\theta}_p \\ \ddot{\theta}_r \end{bmatrix} = \begin{bmatrix} 0 & A_{34} \\ 0 & A_{66} \end{bmatrix} \begin{bmatrix} \theta_p \\ \dot{\theta}_r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + k_{66} \text{sgn}(\dot{\theta}_r) \quad (4.13)$$

State feedback membutuhkan kembalian nilai state dari sistem dan mengkalikanya dengan besaran tertentu agar nilai karakteristik sistem tetap dalam keadaan stabil atau sesuai ketentuan. Secara umum, state tidak dapat diperoleh langsung dari sistem. Kemampuan untuk memperoleh state dari sistem langsung disebut dengan kemampuan Observability. Apabila sebuah sistem tidak Observable, maka dalam kendalinya dibutuhkan Observer. Dimana tugasnya adalah mengestimasi state pada sistem dengan membandingkan keluaran dan masukan. Syarat untuk dapat diterapkan state feedback, sistem harus observable dan controlable. Berikut adalah rumus untuk menguji apakah sistem bersifat controlable atau tidak (Dorf, dkk (2010)).

$$P_c = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$$

$$\text{rank}[P_c] = n \quad (4.14)$$

Apabila hasil dari $\text{rank}(P_c) \neq n$ maka sistem tidak *fully controlable*. Sedangkan untuk menguji observabilitas dapat menggunakan rumus berikut.

$$P_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

$$\text{rank}[P_o] = n \quad (4.15)$$

Apabila sistem observable, rank dari matriks Observability P_o sama dengan besar orde sistem. Menggunakan parameter robot pada Tabel 4.1 untuk diimplementasi pada persamaan (2.18)-(2.19) akan menghasilkan $rank[P_c] = 6$ dan $rank[Po] = 6$. Maka dari itu dapat disimpulkan sistem robot adalah controlable dan observable. Karena pengukuran pada setiap *state* dapat dilakukan, maka *observer* tidak dibutuhkan dalam desain kendali robot.

Tabel 4.1: Parameter model oleh Correia, dkk (2012)

Simbol	Deskripsi	Nilai
$B_v(N/m/s)$	viscous friction coefficient related to v	0.94
$B_{vn}(N/m/s)$	viscous friction coefficient related to vn	0.96
$B_\omega(N/rad/s)$	viscous friction coefficient related to ω	0.01
$C_v(N)$	coulomb friction coefficient related to v	2.2
$C_{vn}(N)$	coulomb friction coefficient related to vn	1.5
$C_\omega(N.m)$	coulomb friction coefficient related to ω	0.099
$b(m)$	radius of the robot	0.1
$M(kg)$	mass of the robot	1.5
$In(kg.m^2)$	inertia moment of the robot	0.025
δ	angle	30°
$r_1, r_2, r_3(m)$	radius of the wheels	0.035
l_1, l_2, l_3	reduction of the motors	19:1
$L_{a1...3}(H)$	motor's armature inductance	0.00011
$R_{a1...3}(\Omega)$	motor's armature resistance	1.69
$K_v(Volts/rad/s)$	motor's emf constant	0.0059
$K_{t1...3}(N.m/A)$	motor's torque constant	0.0059

4.5 Desain Kendali

Berdasarkan Dorf, dkk (2010), bahwa kendali optimal berdasarkan indeks kinerja sistem. Indeks tersebut adalah hasil dari meminimalisasi pada integral kuadrat error atau *integration square error* (ISE). Kendali optimal dilakukan oleh komputer untuk mengkalkulasi minimal indeks tersebut. Apabila sebuah sistem *state space*

$$\dot{x} = Ax + Bu$$

$$u = -K_s x$$

maka indeks kinerja

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (4.16)$$

dimana Q adalah matriks diagonal $n \times n$, R adalah matriks diagonal $m \times m$ dan keduanya adalah matriks pembobot terhadap state sistem dan input. Ketika indeks terminimalisasi, maka

$$K_s = R^{-1} B^T P \quad (4.17)$$

dengan matriks P $n \times n$ ditentukan dari solusi persamaan *Riccati*.

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (4.18)$$

Kalkulasi konstanta K_s akan dikalkulasi menggunakan persamaan-(4.11) (4.13). Sehingga konstanta K_s akan terbagi dalam sub matriks K_{sx} , K_{sy} , dan $K_{s\theta}$. Berikut adalah hasil kalkulasi.

$$K_s^x = \begin{bmatrix} 0.00000 & 0.00000 \\ 18.25742 & 3.25168 \\ -18.25742 & -3.25168 \end{bmatrix} = \begin{bmatrix} K_s^{x1} & K_s^{x2} \end{bmatrix}$$

$$K_s^y = \begin{bmatrix} -21.08185 & -3.74993 \\ 10.54093 & 1.87496 \\ 10.54093 & 1.87496 \end{bmatrix} = \begin{bmatrix} K_s^{y1} & K_s^{y2} \end{bmatrix}$$

$$K_s^\theta = \begin{bmatrix} 3.33333 & 0.99738 \\ 3.33333 & 0.99738 \\ 3.33333 & 0.99738 \end{bmatrix} = \begin{bmatrix} K_s^{\theta1} & K_s^{\theta2} \end{bmatrix}$$

Apabila diintegrasikan terhadap persamaan (2.18) terhadap diagram 4.3

$$K_s = \begin{bmatrix} K_s^{x1} & K_s^{y1} & K_s^{\theta1} & K_s^{x2} & K_s^{y2} & K_s^{\theta2} \end{bmatrix} \quad (4.19)$$

Setelah mendapatkan konstanta K_s , sistem sudah dalam keadaan stabil. Akan tetapi sistem tidak mencapai set poin yang diinginkan. Maka permasalahan tersebut dapat diselesaikan dengan *input refrence*. Dimana refrence tersebut akan dikalikan dengan konstanta N . Berikut adalah persamaan *input refrence* sebagai penambah dari *state feedback*.

$$u(t) = -Kx(t) + Nr \quad (4.20)$$

Sehingga persamaan *state space* menjadi berikut.

$$\begin{cases} \dot{x} &= (A - BK_s)x + BNr \\ y &= Cx \end{cases} \quad (4.21)$$

Untuk mendapatkan nilai N maka dapat diasumsikan bahwa sistem dalam keadaan *steady state*, yaitu $\dot{x} = 0$, sehingga persamaan *state space* menjadi berikut.

$$x = -(A - BK_s)^{-1}BNr \quad (4.22)$$

Dalam keadaan *steady state*, harapannya adalah nilai refrence sama dengan nilai keluaran, $y = r$. Sehingga dapat diperoleh persamaan N .

$$N = -[C(A - BK_s)^{-1}B]^{-1} \quad (4.23)$$

Berikut adalah hasil kalkulasi dari rumus N menggunakan matriks pada persamaan (4.11),(4.12), dan (4.13).

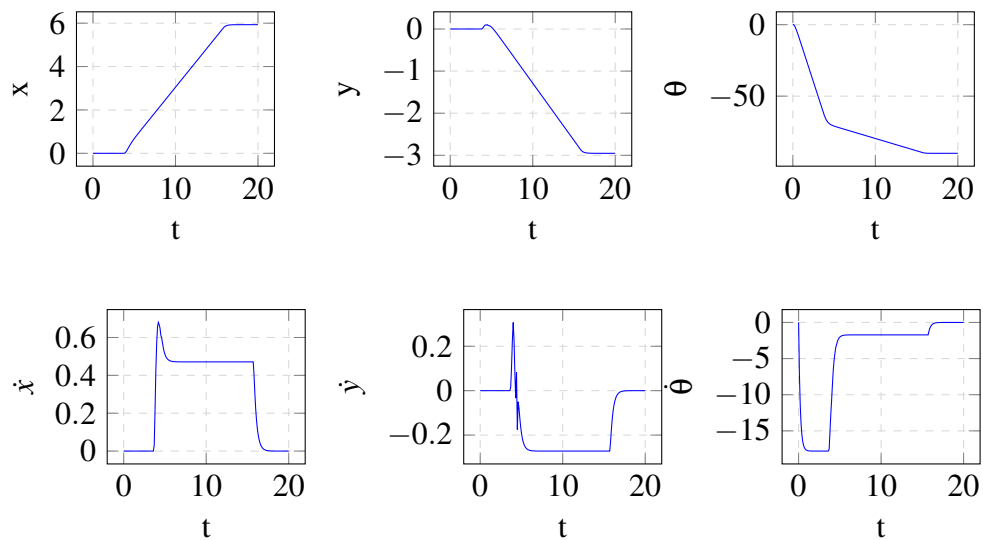
$$N^x = \begin{bmatrix} 0.00000 & 0.00000 \\ 18.25742 & 0.00000 \\ -18.25742 & 0.00000 \end{bmatrix} = \begin{bmatrix} N^{x1} & N^{x2} \end{bmatrix}$$

$$N^y = \begin{bmatrix} -21.08185 & 0.00000 \\ 10.54093 & 0.00000 \\ 10.54093 & 0.00000 \end{bmatrix} = \begin{bmatrix} N^{y1} & N^{y2} \end{bmatrix}$$

$$N^\theta = \begin{bmatrix} 3.33333 & 0.00000 \\ 3.33333 & 0.00000 \\ 3.33333 & 0.00000 \end{bmatrix} = \begin{bmatrix} N^{\theta1} & N^{\theta2} \end{bmatrix}$$

Apabila diintegrasikan terhadap persamaan (2.18) terhadap diagram 4.3

$$N = \begin{bmatrix} N^{x1} & N^{y1} & N^{\theta1} & N^{x2} & N^{y2} & N^{\theta2} \end{bmatrix} \quad (4.24)$$



Gambar 4.4: Grafik state terhadap t dengan state referensi $r = [6 \quad -3 \quad -90 \quad 0 \quad 0 \quad 0]$

4.6 Kendali Formasi Multi Robot

Pada sub bab 2.1.2 dijabarkan bagaimana kendali formasi menggunakan kendali-P dan menghasilkan persamaan (2.7). Persamaan tersebut adalah persamaan *state-space* kendali formasi. Apabila diperhatikan *state* yang digunakan adalah koordinat relatif dari robot. Akan tetapi dalam batasannya, robot hanya bisa mengetahui nilai jarak dari robot lain. Dengan kata lain, yang dibutuhkan dalam metode kendali formasi adalah jarak dalam bentuk koordinat, $x \in \mathbb{R}^2$. Sedangkan dalam kenyataanya yang diketahui adalah jarak, $r \in \mathbb{R}$. Apabila hanya variable jarak tersebut sebagai acuan kendali, maka robot tidak mengerti kearah mana harusnya robot itu bergerak untuk meminimalisasi error jaraknya.

Di Persamaan (2.7) adalah persamaan *state-space* yang menggunakan model sederhana di Persamaan (2.3). Apabila diperhatikan pada model tersebut masukan pada persamaan tersebut berbentuk kecepatan. Menggunakan Persamaan (4.5) dan (4.9) akan diperoleh persamaan *state-space* baru

$$\dot{x}_c(t) = \left((A_c - B_c K_c +) \right) x_c(t) K_{cf} \text{sgn}(x_c(t)) + (B_c N_c) r_c, \quad (4.25)$$

Lalu diterapkan di Persamaan (2.7) sebagai berikut.

$$\dot{p} = A_{cf}p(t) + B_{cf}v(t) + K_{cf}sgn(p(t)) \quad (4.26)$$

$$\dot{v} = -k_{p1}v_i(t) - R(p(t))^T k_{p2}(R(p(t))x_1(t) - d) \quad (4.27)$$

$$A_{cf} = \begin{bmatrix} A_c - B_c K_c & 0 & 0 \\ 0 & A_c - B_c K_c & 0 \\ 0 & 0 & A_c - B_c K_c \end{bmatrix}; B_{cf} = \begin{bmatrix} (B_c N_c) \\ (B_c N_c) \\ (B_c N_c) \end{bmatrix};$$

$$K_{cf} = \begin{bmatrix} k_c & 0 & 0 \\ 0 & k_c & 0 \\ 0 & 0 & k_c \end{bmatrix} \quad (4.28)$$

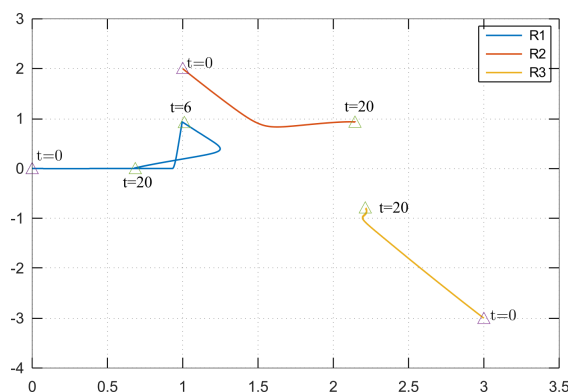
BAB 5

HASIL DAN PEMBAHASAN

Hasil pembahasan akan dilakukan dengan membahas hasil percobaan algoritma *cosinus*. Pembahasan hasil pertama akan membahas jalannya algoritma terhadap kendali formasi untuk menentukan koordinat kondisi awal dan mengetahui langkah-langkah nya. Pembahasa kedua akan membandingkan penerapan algoritma dengan tanpa algoritma sehingga mengetahui spesifikasi kecepatan algoritma untuk menemukan koordinat kondisi awal. Pembahasan terakhir akan dilakukan percobaan dengan kuadran yang berbeda sehingga algoritma dapat bekerja dengan baik untuk menemukan koordinat.

5.1 Pergerakan Robot Terhadap Algoritma *Cosinus*

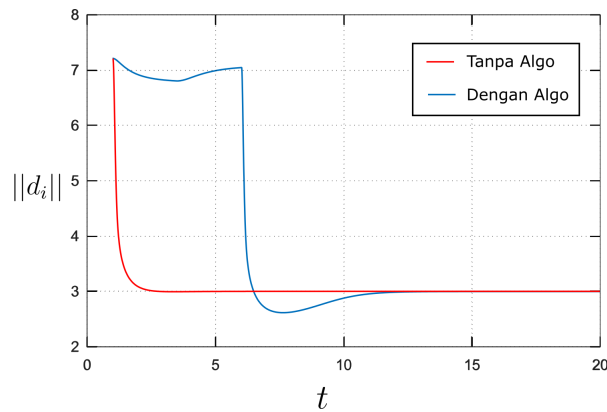
Kendali formasi berdasarkan jarak akan dijalankan secara simulasi menggunakan MATLAB/GNU Octave dan algoritma *cosinus* akan dijalankan pertama kali untuk mendapatkan state yang akan digunakan di Persamaan (16) dan (18). Simulasi akan menggunakan 3 robot dengan himpunan simpul $V = \{R_1, R_2, R_3\}$ dan himpunan sisi $E = \{(R_1, R_2), (R_3, R_2), (R_3, R1)\}$ sehingga variable jarak $d_1 = \|x_1 - x_2\|$, $d_2 = \|x_3 - x_2\|$ dan $d_3 = \|x_3 - x_1\|$. Gambar 4 adalah pergerakan robot terhadap koordinat global robot R_1 dengan kendali formasi algoritma *cosinus*. dapat diperhatikan robot R_1 menjalankan algoritma *cosinus* langkah pertama dimana robot berpindah sepanjang $l_a = 1$ sehingga setpoint kendali Persamaan (13) adalah $r_2^c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. Setelah setpoint tercapai, robot R_1 mendapatkan jarak yang dibutuhkan



Gambar 5.1: Grafik Pergerakan Robot Menggunakan Algoritma *Cosinus* dan Kendali Formasi

5.2 Perbandingan Penerapan Algoritma *Cosinus*

untuk menyelesaikan Persamaan (21) dan (22) lalu dilanjutkan ke langkah dua dengan berpindah ke koordinat (1,1) sehingga setpoint kendali Persamaan (13) adalah $r_2^c = [110000]^T$. Setelah setpoint tercapai maka dilakukan pengecekan kejadian Persamaan (23) dengan membandingkan jarak dari sensor dengan jarak dari koordinat yang dihasilkan dari Persamaan (22). Robot R_3 berada di kuadran 4 maka kejadian yang digunakan adalah dan robot R_2 berada di kuadran 1 maka kejadian yang digunakan adalah $\alpha_i^o = 180^\circ - \zeta_i^a$. Setelah koordinat ditemukan, maka kendali robot mulai berpindah menggunakan kendali formasi Persamaan (16) dan (18) dengan kondisi koordinat awal state berada di $t = 6$ dan akhir dari formasi di $t = 20$ pada Gambar 4.



Gambar 5.2: Grafik Response Jarak Terhadap Waktu Pada Robot Dengan Kendali Formasi.

5.3 Algoritma *Cosinus* Di Seluruh Kuadran

Seperti yang telah dijelaskan di pembahasan algoritma cosinus (subbab 2.4), bahwa kendali formasi di Persamaan (16) dan (18) mengharuskan state dalam bentuk koordinat dimana ketika kondisi awal robot tidak mengetahuinya, maka peran utama dari algoritma cosinus untuk mendapatkan state tersebut. Gambar 5 adalah grafik perbandingan kendali formasi Persamaan (16) dan (18) dengan algoritma cosinus dan tanpa algoritma cosinus dengan asumsi ketika dilakukan percobaan tanpa algoritma, kondisi awal state telah diketahui. Hasil jarak $\|d_i\|$ pada kendali formasi tanpa algoritma cosinus menunjukkan bahwa setiap robot mencapai jarak yang sama dalam waktu kurang lebih 4 detik. Sedangkan pada kendali formasi dengan algoritma cosinus terdapat tambahan waktu 6 detik. Tambahan waktu tersebut digunakan untuk menjalankan algoritmanya.

Tabel 5.1: Settling Time Dengan Konstanta Kp Yang Berbeda

Koordinat			Settling Time Tanpa Algoritma cosinus (detik)			Settling Time Dengan Algoritma cosinus (detik)			Selisih (detik)		
			A	B	C	A	B	C	A	B	C
R1	R2	R3									
(0,0)	(1,2)	(-2, 3)	6	5	3	14	11	9	8	6	6
(0,0)	(-2, -4)	(3, -2)	4	2	2	9	8	7	5	6	5
(0,0)	(1,2)	(3, -3)	7	4	2	16	13	11	9	9	9
(0,0)	(-2, -3)	(3,2)	10	7	4	16	11	9	6	4	5
Konstanta : (A) $K_{p1} = 50$; $K_{p2} = 3$; (B) $K_{p1} = 80$; $K_{p2} = 7$; (C) $K_{p1} = 100$; $K_{p2} = 15$;											

Percobaan di Gambar 5 adalah grafik settling time saat koordinat robot seperti di Gambar 4 yaitu R_2 berada di kuadran 1 dan R_3 berada di kuadran 4. Hasil percobaan Tabel 1 bertujuan untuk mengetahui kecepatan algoritma cosinus mencari koordinat tetangga dengan kombinasi kemungkinan dua robot dengan kejadian di Persamaan (23). Serta diberikan nilai K_{p1} dan K_{p2} di Persamaan (18) yang berbeda untuk mengetahui pengaruh parameter kendali formasi terhadap algoritma cosinus. Setelah dilakukan percobaan, selisih waktu ketika algoritma cosinus berjalan dengan perbedaan parameter menghasilkan waktu yang tidak jauh beda antara 0-2 detik. Maka, algoritma cosinus mendapatkan koordinat tetangga yang digunakan untuk nilai kondisi awal kendali hampir tidak berpengaruh terhadap kendali formasi Persamaan (18) dengan rata-rata waktu 6.5 detik, minimal waktu 4 detik dan maksimal 9 detik.

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kendali formasi berdasarkan jarak telah dikembangkan fokus dibagian metode kendali dan masih perlu dibahas penerapannya. Telah dikembangkan di penelitian sebelumnya dengan fokus di metode kendali formasi berdasarkan jarak dengan penerapan metode PI dan analisa menghasilkan respon yang baik, akan tetapi penerapan metode tersebut kenyataannya tidak berjalan dengan baik karena robot tidak mendapatkan kondisi awal yang digunakan untuk kendalinya. Penelitian ini telah mengusulkan algoritma *cosinus* sebagai solusi kondisi awal kendali. Dari hasil percobaan, algoritma *cosinus* dapat digunakan untuk mendapatkan koordinat kondisi awal kendali formasi berdasarkan jarak, algoritma dapat menemukan koordinat relatif tetangganya di semua kuadran, dan rata-rata waktu yang dibutuhkan adalah 6.5 detik.

6.2 Saran

Penelitian ini dimulai dengan pengembangan dibagian kendali formasi berdasarkan jarak dan penerapannya pada model yang nyata. Akan tetapi dalam proses penelitian terdapat permasalahan kondisi awal pada kendalinya sehingga pengembangan kendali formasi berdasarkan jarak masih jauh dianggap sempurna. Ditujuan kepada penelitian selanjutnya dibidang kendali formasi berdasarkan jarak terdapat peluang penelitian di bagian penerapan kendali tanpa menggunakan robot asli yaitu menggunakan metode HIL (Hardware In Loop). Saran ini disampaikan karena penelitian dibidang kendali formasi mengendalikan banyak robot dan juga pemodelan robot yang nyata telah banyak dikembangkan maka penggunaan metode HIL sangat bermanfaat untuk melakukan analisis kendali formasi. Peluang penelitian selanjutnya juga dapat dimulai di bagian analisis kendali menggunakan metode MPC (*Model Predictive Control*) karena metode MPC banyak digunakan untuk mengendalikan robot secara optimal dan *state feedback* adalah dasar dari metode MPC.

DAFTAR REFERENSI

- Cai, Xiaoyu, dan Marcio de Queiroz (2014). “Multi-agent formation maneuvering and target interception with double-integrator model”. In: *2014 American Control Conference*, pp. 287–292.
- Correia, Mariane Dourado, André Gustavo, dan Scolari Conceição (2012). “Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models”. In: *IFAC Proceedings Volumes 45.22*. 10th IFAC Symposium on Robot Control, pp. 7 –12.
- Deghat, Mohammad, Brian D. O. Anderson, dan Zhiyun Lin (2016). “Combined Flocking and Distance-Based Shape Control of Multi-Agent Formations”. In: *IEEE Transactions on Automatic Control* 61.7, pp. 1824–1837.
- Dorf, Richard, dan Robert Bishop (July 2010). *Modern Control Systems, 12th Edition*. ISBN: ISBN-10: 0136024580; ISBN-13: 978-0136024583.
- Ferik, Sami El, Mohammad Tariq Nasir, dan Uthman Baroudi (2016). “A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space”. In: *Applied Soft Computing* 44, pp. 117 –127.
- Guanghua, Wang, Li Deyi, dan Gan Wenyan (Jan. 2013). “Study on Formation Control of Multi-Robot Systems”. In: pp. 1335–1339. ISBN: 978-1-4673-4893-5.
- Guo, Kexin, Xiuxian Li, dan Lihua Xie (2020). “Ultra-Wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control”. In: *IEEE Transactions on Cybernetics* 50.6, pp. 2590–2603.
- Hacene, Nacer, dan Boubekeur Mendil (Apr. 2019). “Fuzzy Behavior-based Control of Three Wheeled Omnidirectional Mobile Robot”. In: *International Journal of Automation and Computing* 16.2, pp. 163–185.
- Kang, Sung-Mo, dkk (2014). “Distance-based formation control with a single moving leader”. In: *2014 American Control Conference*, pp. 305–310.
- Li, Xue-ren, dkk (2015). “Large-Scale Unmanned Aerial Vehicle Formation Maintenance Algorithm Based on Virtual Structure”. In: *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 1, pp. 417–421.
- Oh, Kwang-Kyo, dan Hyo-Sung Ahn (2014). “Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space”. In: *International Journal of Robust and Nonlinear Control*

- 24.12, pp. 1809–1820. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.2967>.
- Oh, Kwang-Kyo, Myoung-Chul Park, dan Hyo-Sung Ahn (2015). “A survey of multi-agent formation control”. In: *Automatica* 53, pp. 424–440.
- Park, Myoung-Chul, dan Hyo-Sung Ahn (2015). “Distance-based control of formations with orientation control”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 2199–2204.
- Qiang, Lv, dkk (2017). “Design and implementation of multi robot research platform based on UWB”. In: *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 7246–7251.
- Qiang, Lv, dkk (2018). “Formation control of multi robot based on UWB distance measurement”. In: *2018 Chinese Control And Decision Conference (CCDC)*, pp. 2404–2408.
- Rozenheck, O., S. Zhao, dan D. Zelazo (2015). “A proportional-integral controller for distance-based formation tracking”. In: *2015 European Control Conference (ECC)*, pp. 1693–1698.
- Wang, X., Z. Yan, dan J. Wang (2014). “Model predictive control of multi-robot formation based on the simplified dual neural network”. In: *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 3161–3166.