



TUGAS AKHIR - TE 141599

**PENGATURAN FORMASI ROBOT MOBIL
BERDASARKAN PENDEKATAN *VIRTUAL STRUCTURE*
DENGAN PENGHINDARAN RINTANGAN**

Anang Apriyanto
NRP 07111340000155

Dosen Pembimbing
Dr. Trihastuti Agustinah, ST.,MT.

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



FINAL PROJECT - TE 141599

***Formation Control of Mobile Robots Based on
Virtual Structure Approach with Obstacle
Avoidance***

Anang Apriyanto
NRP 07111340000155

Supervisor
Dr. Trihastuti Agustinah, ST.,MT.

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

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Pengaturan Formasi Robot Mobil berdasarkan Pendekatan *Virtual Structure* dengan Penghindaran Rintangan**” adalah merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2018

Anang Apriyanto
NRP 07111340000155

[Halaman ini sengaja dikosongkan]

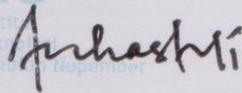
**PENGATURAN FORMASI ROBOT MOBIL BERDASARKAN
PENDEKATAN *VIRTUAL STRUCTURE* DENGAN
PENGHINDARAN RINTANGAN**

TUGAS AKHIR

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

Menyetujui:

Dosen Pembimbing,



Dr. Trihastuti Agustinah, ST., MT.
NIP. 196808121994032001



[Halaman ini sengaja dikosongkan]

PENGATURAN FORMASI ROBOT MOBIL BERDASARKAN PENDEKATAN *VIRTUAL STRUCTURE* DENGAN PENGHINDARAN RINTANGAN

Anang Apriyanto – 07111340000155

Pembimbing : Dr. Trihastuti Agustinah, ST.,MT.

ABSTRAK

Bekerja dalam formasi memiliki beberapa keuntungan diantaranya bisa mengurangi biaya, meningkatkan ketahanan dan meningkatkan efisiensi sistem. Pada tugas akhir ini dirancang suatu sistem pengaturan formasi menggunakan pendekatan *virtual structure* dengan penghindaran rintangan menggunakan metode *artificial potential field*. Robot ditugaskan untuk membentuk formasi (berupa bentuk V-terbalik atau belah ketupat), bergerak dalam formasi menuju titik tujuan, menghindari tabrakan antar robot, dan menghindari tabrakan dengan rintangan. Tabrakan antara robot dengan objek lain (robot lain atau rintangan) dapat dihindari apabila jarak antara keduanya berada pada jarak aman yaitu diatas 5 cm. Sistem hasil rancangan memanfaatkan posisi relatif robot dan jarak antara robot dengan rintangan di sekitarnya. Hasil simulasi menunjukkan robot berhasil membentuk formasi sesuai dengan yang diinginkan. Jarak terdekat antar robot dan jarak terdekat antara robot dengan rintangan berturut-turut adalah 5,228 cm dan 5,2349 cm.

Kata kunci: Pengaturan formasi, *tracking* formasi, *consensus*, *collision avoidance*, *artificial potential field*.

[Halaman ini sengaja dikosongkan]

FORMATION CONTROL OF MOBILE ROBOTS BASED ON VIRTUAL STRUCTURE APPROACH WITH OBSTACLE AVOIDANCE

Anang Apriyanto – 07111340000155

Supervisor : Dr. Trihastuti Agustinah, ST.,MT.

ABSTRACT

Moving in formation have so many benefits such as reducing the system cost, increasing system reliability and efficiency. In this final project, a formation control system is proposed. The formation control is based on virtual structure approach, while the obstacle avoidance is based on artificial potential field method. Robot is given task to form a formation (inverted V or rhombus shape), maintaining formation shape while moving to goal point, avoiding collision with other robots, and avoiding collision with obstacles. Collision with another object (another robot or obstacle) can be avoided if the distance between robot and robot and the obstacle is greater than 5 cm. The system proposed uses information such as robot relative positions and distance between robot and obstacles. Simulation results show that robots succeed at forming the desired formation. The closest interrobot distance and the closest robot-to-obstacle distance is 5.228 cm and 5.2349 cm, respectively.

Keywords: *Formation control, formation tracking, consensus, collision avoidance, artificial potential field.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “**Pengaturan Formasi Robot Mobil berdasarkan Pendekatan *Virtual Structure* dengan Penghindaran Rintangan**”. Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungannya yang telah diberikan selama proses pembuatan tugas akhir ini kepada :

1. Orang tua serta keluarga dan kerabat yang senantiasa memberikan doa serta dukungan.
2. Ibu Dr. Trihastuti Agustinah, ST.,MT. selaku dosen pembimbing yang selalu memberikan pengarahan, saran dan motivasi dalam kelancaran tugas akhir ini.
3. Bapak M. Latif yang telah membantu *me-review* tugas akhir ini dalam hal tata tulis.
4. Rekan-rekan: Rafif, Asa, Hendri, Matin, Nindya, dan Irfan yang telah memberikan bantuan dan dukungan baik secara langsung maupun tidak langsung dalam penulisan tugas akhir ini.
5. Tim dosen penguji yang telah memberikan saran perbaikan.
6. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari masih banyak kekurangan dalam tugas akhir ini. Kritik dan saran untuk perbaikan tugas akhir ini sangat diperlukan. Semoga tugas akhir ini dapat bermanfaat bagi semua pihak.

Surabaya, Januari 2018

Anang Apriyanto
NRP 0711134000155

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Metodologi	3
1.6 Sistematika Penulisan	3
1.7 Relevansi	4
BAB 2 TINJAUAN PUSTAKA	5
2.1 Teori Graf	5
2.1.1 Istilah-istilah	5
2.1.2 Matriks Representasi	7
2.1.3 <i>Consensus Protocol</i>	8
2.2 Robot Mobil	9
2.2.1 Model Kinematika Robot	11
2.3 Konsep Dasar Pengaturan Formasi	13
2.3.1 Pendekatan <i>Virtual Structure</i>	13
2.4 Metode <i>Artificial Potential Field</i>	15
BAB 3 PERANCANGAN SISTEM	19
3.1 Gambaran Umum Sistem	19
3.1.1 Spesifikasi Fisik Robot	19
3.1.2 Deskripsi Lingkungan Kerja Robot	19
3.1.3 Spesifikasi Formasi	20
3.2 Perancangan <i>Control Law</i>	21
3.2.1 Pembentukan Formasi	22
3.2.2 Pembentukan Formasi dan Penghindaran Tabrakan	25
3.2.3 Pembentukan Formasi dan Penghindaran Rintangan	27
3.2.4 <i>Tracking</i> Formasi dan Penghindaran Rintangan	29

3.3 Perancangan Simulasi dan Analisis	29
BAB 4 SIMULASI DAN ANALISIS SISTEM	31
4.1 Simulasi Pembentukan Formasi	31
4.1.1 Graf Tak-terhubung	32
4.1.2 Graf Terhubung	34
4.1.3 Graf Lengkap.....	36
4.2 Simulasi Pembentukan Formasi dengan Penghindaran	
Tabrakan Antar Robot	39
4.2.1 Graf Terhubung	39
4.2.2 Graf Lengkap.....	39
4.3 Simulasi Pembentukan Formasi dengan Penghindaran	
Rintangan.....	40
4.3.1 Metode APF Eksponensial	40
4.3.2 Metode APF Modifikasi	42
4.3.3 Analisis Penghindaran Tabrakan Antar Robot	43
4.4 Simulasi <i>Tracking</i> Formasi dengan Penghindaran Rintangan	44
4.4.1 Penghindaran Rintangan.....	44
4.4.2 Pelintasan Celah Sempit	47
BAB 5 PENUTUP	51
5.1 Kesimpulan	51
5.2 Saran.....	51
DAFTAR PUSTAKA	53
LAMPIRAN.....	55
RIWAYAT PENULIS	75

DAFTAR GAMBAR

Gambar 2.1 Graf tak-berarah dengan 5 simpul	5
Gambar 2.2 (a) Graf terhubung (b) Graf tak-terhubung	6
Gambar 2.3 Graf lengkap K_4	7
Gambar 2.4 Consensus protocol pada tiga agen.....	9
Gambar 2.5 Nilai state 5 agen dengan consensus protocol	10
Gambar 2.6 Permasalahan rendezvous.....	10
Gambar 2.7 Robot mobil differential drive	11
Gambar 2.8 Representasi visual dari masalah bergerak dalam formasi.....	14
Gambar 2.9 Langkah-langkah dalam algoritma untuk bergerak dalam formasi.....	14
Gambar 3.1 Spesifikasi bentuk formasi V terbalik	20
Gambar 3.2 Spesifikasi bentuk formasi belah ketupat	21
Gambar 3.3 Robot nyata dan robot virtualnya	22
Gambar 3.4 Posisi relatif robot virtual terhadap robot nyata	24
Gambar 3.5 Fungsi potensial U_{ij}	26
Gambar 4.1 (a) Graf tak-terhubung (b) Graf terhubung (c) Graf lengkap	31
Gambar 4.2 Posisi dan orientasi awal robot	32
Gambar 4.3 Trayektori robot graf tak-terhubung saat $t=20s$	33
Gambar 4.4 Indeks performansi formasi graf tak terhubung.....	33
Gambar 4.5 Trayektori robot graf terhubung saat $t=20s$	34
Gambar 4.6 Orientasi robot graf terhubung	34
Gambar 4.7 Kecepatan robot pada sumbu- x dan sumbu- y	35
Gambar 4.8 Jarak antara robot i dengan robot j	35
Gambar 4.9 Indeks performansi formasi graf terhubung	36
Gambar 4.10 Trayektori robot dengan graf lengkap	36
Gambar 4.11 Orientasi robot graf lengkap.....	37
Gambar 4.12 Indeks performansi formasi graf lengkap	37
Gambar 4.13 Orientasi robot graf lengkap dengan $k=0,1$	38
Gambar 4.14 IPF robot graf lengkap dengan $k=0,1$	38
Gambar 4.15 d_{ij} dengan graf terhubung.....	39
Gambar 4.16 d_{ij} dengan graf lengkap	40
Gambar 4.17 Trayektori robot metode APF eksponensial	41
Gambar 4.18 Orientasi robot metode APF.....	41
Gambar 4.19 Indeks performansi robot.....	42

Gambar 4.20 Trayektori robot metode APF modifikasi	42
Gambar 4.21 Orientasi robot menggunakan metode APF modifikasi ...	43
Gambar 4.22 Jarak antar robot dengan APF modifikasi	44
Gambar 4.23 Trayektori hasil <i>tracking</i> formasi dengan penghindaran rintangan	45
Gambar 4.24 IPF robot pada <i>tracking</i> formasi dengan penghindaran rintangan.....	46
Gambar 4.25 Orientasi robot pada <i>tracking</i> formasi dengan penghindaran rintangan	46
Gambar 4.26 Jarak antar robot pada tracking formasi dengan penghindaran rintangan	47
Gambar 4.27 Trayektori robot	48
Gambar 4.28 Jarak antar robot.....	49
Gambar 4.29 Orientasi robot	49
Gambar 4.30 IPF tracking formasi melewati celah.....	50

DAFTAR TABEL

Tabel 3.1 Spesifikasi robot.....	19
Tabel 3.2 Jarak antar robot pada formasi V terbalik	20
Tabel 3.3 Jarak antar robot pada formasi belah ketupat	21
Tabel 4.1 Posisi dan orientasi awal robot.....	32
Tabel 4.2 Posisi dan orientasi awal robot.....	45
Tabel 4.3 Posisi dan orientasi awal robot.....	47

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengaturan formasi adalah salah satu bidang yang paling diminati dalam pengaturan multi robot. Pengaturan formasi dapat dijadikan solusi pada berbagai permasalahan seperti survei di daerah yang berbahaya atau daerah bencana, pemetaan, serta formasi pesawat terbang atau satelit [1].

Berbagai pendekatan formasi robot telah diteliti dan dikembangkan. Tiga pendekatan untuk memecahkan permasalahan ini yaitu: *behavior based*, *virtual structure* dan *leader-following*. Pada pendekatan *behavior-based*, beberapa *behavior* yang diinginkan diberikan pada tiap robot. Aksi yang dilakukan oleh robot dihasilkan dari seberapa pentingnya *behavior* tersebut. *Behavior* dengan prioritas tinggi akan didahulukan daripada *behavior* dengan prioritas yang lebih rendah. Permasalahan utama dari pendekatan ini yaitu formalisasi matematis yang terlalu sulit dan akibatnya terkadang konfigurasi formasi yang diinginkan tidak tercapai. Permasalahan lainnya yaitu penentuan aksi robot seringkali membutuhkan komputasi yang kompleks sehingga dalam implementasinya membutuhkan perangkat dengan kemampuan komputasi yang baik. Waktu yang dibutuhkan untuk menentukan aksi robot juga menjadi lebih lama. Pada lingkungan kerja tertentu, robot diharuskan untuk beraksi dalam waktu yang singkat untuk menghindari kemungkinan adanya kecelakaan misalnya tabrakan dengan rintangan atau robot lain.

Pada pendekatan *leader-follower*, sebuah robot dipilih sebagai *leader*. Robot *leader* ini bergerak mengikuti jalur yang telah ditentukan dan selanjutnya robot-robot lain sebagai *follower* mengikuti *leader* atau dengan kata lain robot *leader* dijadikan acuan bagi robot *follower*. Kelemahan utama dari pendekatan ini adalah adanya hirarki dalam sistem kontrolnya dan ketergantungan yang sangat besar pada sebuah robot *leader*. Apabila robot *leader* menyimpang dari jalur yang diinginkan maka robot *follower* akan tetap mengikuti. Masalah lain yang timbul dari pendekatan ini yaitu tidak adanya umpan balik dari robot *follower* ke robot *leader*. Meskipun begitu, pendekatan ini adalah yang paling sederhana diantara ketiga pendekatan yang ada. Pada pendekatan *virtual structure*, robot-robot dalam formasi diperlakukan sebagai sebuah kesatuan *virtual structure*. Berbeda dengan pendekatan *leader-follower*,

pada pendekatan ini tidak terdapat hirarki. Untuk permasalahan *trajectory tracking*, *trajectory* tidak diberikan pada salah satu robot melainkan keseluruhan robot dalam formasi. Selain itu, pergerakan sebuah robot dalam formasi juga dipengaruhi oleh robot-robot lain sehingga apabila terdapat robot yang mengalami gangguan maka robot lain akan terpengaruh [2].

Pada kondisi dimana robot tidak hanya membentuk formasi tetapi juga menuju ke suatu titik tujuan tertentu, penghindaran rintangan perlu dipertimbangkan. Penghindaran rintangan pada sebuah sistem multi robot memerlukan strategi kontrol yang berbeda dengan metode penghindaran rintangan biasa. Jumlah robot yang lebih dari satu membuat kompleksitas permasalahan menjadi lebih tinggi. Penghindaran rintangan pada satu robot menugaskan robot untuk menghindari rintangan yang terdeteksi dan kembali mencari jalur menuju titik tujuan. Pada robot-robot yang bergerak dalam formasi, penghindaran rintangan harus mempertimbangkan penghindaran tabrakan dengan robot lain dan pembentukan kembali formasi setelah rintangan tidak terdeteksi. Kemudian robot kembali menuju titik tujuan yang diinginkan dengan tetap mempertahankan formasi

Pada penelitian ini akan disusun sebuah sistem pengaturan yang memungkinkan robot untuk membentuk formasi dan menghindari rintangan yang terdeteksi. Pembentukan formasi memanfaatkan topologi jaringan komunikasi antarrobot. Jaringan komunikasi ini juga digunakan untuk penghindaran tabrakan antarrobot. Penghindaran rintangan dalam penelitian ini memanfaatkan metode *Artificial Potential Field* (APF). Metode ini didasarkan pada data yang diperoleh dari sensor robot. Kelemahan-kelemahan pada metode yang digunakan diperbaiki dengan berbagai pendekatan yang ada.

1.2 Perumusan Masalah

Pengaturan formasi memiliki potensi aplikasi yang luas pada berbagai bidang seperti pemindahan objek, pengawasan, pemetaan, eksplorasi tempat berbahaya atau tempat bencana hingga *search and rescue*.

Penerapan pengaturan formasi pada tugas-tugas tersebut seringkali mengharuskan robot bekerja pada lingkungan yang tidak diketahui. Umumnya pada lingkungan tersebut terdapat objek-objek yang dapat menghalangi robot untuk bergerak sesuai tugas yang diberikan. Untuk itu pada tugas akhir ini dirancang strategi kontrol untuk mengatasi masalah tersebut.

Pada penelitian ini robot dengan posisi dan orientasi awal tertentu ditugaskan untuk membentuk dan mempertahankan formasi. Pada saat bersamaan robot juga ditugaskan menuju suatu titik tertentu atau mengikuti trayektori yang telah ditentukan. Saat rintangan terdeteksi, robot ditugaskan untuk menghindari rintangan.

Rumusan masalah pada penelitian ini adalah bagaimana merancang strategi kontrol bagi robot untuk memenuhi tugas-tugas tersebut.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini yaitu:

1. Robot mobil yang digunakan adalah robot mobil tipe *differential drive*
2. Rintangan bersifat statis yaitu diam dan tidak bergerak (berpindah tempat)
3. Bentuk formasi bersifat *translationaly invariant* yaitu tetap tanpa dipengaruhi oleh arah gerak robot

1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu merancang strategi kontrol memanfaatkan penerapan pendekatan *virtual structure* untuk membentuk dan mempertahankan formasi robot dan penerapan metode APF untuk penghindaran rintangan robot dalam formasi.

1.5 Metodologi

Tahapan-tahapan kegiatan dalam penelitian ini yaitu studi literatur, pemodelan kinematika robot, perancangan strategi kontrol, dan simulasi. Berdasarkan studi literatur, diperoleh permasalahan-permasalahan pada pengaturan formasi dan solusi yang ditawarkan. Model kinematika robot digunakan sebagai dasar perancangan sistem pengaturan. Simulasi dilakukan untuk mengamati hasil dari strategi kontrol yang telah dirancang.

1.6 Sistematika Penulisan

Penelitian ini dibagi menjadi lima bab dengan rincian sistematika sebagai berikut.

BAB 1 PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang penulis mengambil topik penelitian dan permasalahan-permasalahan terkait topik yang

diteliti penulis. Bab ini juga menjelaskan batasan masalah pada penelitian dan metodologi penelitian yang digunakan.

BAB 2 DASAR TEORI

Pada bab ini dijelaskan mengenai teori-teori yang digunakan dalam penelitian. Teori-teori tersebut berupa metode-metode kontrol yang digunakan untuk permasalahan yang berkaitan dengan pengaturan formasi dan penghindaran rintangan.

BAB 3 PERANCANGAN SISTEM

Bab ini menjelaskan spesifikasi fisik robot dan lingkungan kerja robot. Dijelaskan juga mengenai strategi kontrol yang dirancang penulis

BAB 4 SIMULASI DAN ANALISIS SISTEM

Sistem hasil rancangan pada Bab 3 disimulasikan. Hasil dari simulasi dianalisis dan dijelaskan pada bab ini.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi ringkasan singkat hasil analisis pada Bab 4. Selain itu, diberikan saran terkait penelitian lanjutan untuk mengembangkan strategi kontrol yang telah dirancang

1.7 Relevansi

Sistem hasil rancangan diharapkan bisa menjadi prototip untuk diterapkan pada permasalahan riil. Penelitian ini juga diharapkan mampu dikembangkan dan menjadi referensi bagi sistem pengaturan formasi.

BAB 2 TINJAUAN PUSTAKA

Teori diperlukan sebagai landasan dalam perancangan sistem pengaturan formasi robot mobil. Pada bab ini dibahas secara singkat mengenai teori graf, robot mobil nonholonomik, dan pendekatan *virtual structure* dalam sistem pengaturan formasi.

2.1 Teori Graf [3]

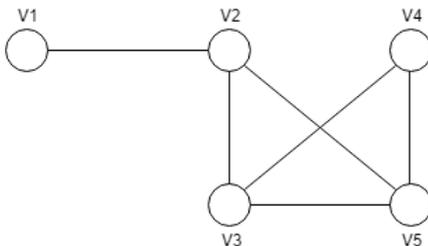
2.1.1 Istilah-istilah

Sebuah graf adalah sebuah himpunan yang terdiri atas himpunan simpul (*vertex*) dan himpunan busur (*edge*). Himpunan simpul dari sebuah graf \mathcal{G} dilambangkan dengan $V(\mathcal{G})$. Himpunan simpul dengan n anggota dapat dituliskan sebagai:

$$V(\mathcal{G}) = \{v_1, v_2, \dots, v_n\}.$$

Himpunan busur dilambangkan dengan $E(\mathcal{G})$. Busur adalah pasangan tak terurut dari simpul. Busur dapat dilambangkan dengan $\{v_i, v_j\}$ atau $v_i v_j$ dengan $i, j = 1, 2, \dots, n$ dan $i \neq j$. Kardinalitas (banyaknya anggota) dari himpunan simpul pada suatu graf \mathcal{G} disebut sebagai orde (*order*) graf \mathcal{G} sedangkan kardinalitas dari himpunan busur disebut sebagai ukuran (*size*) graf \mathcal{G} .

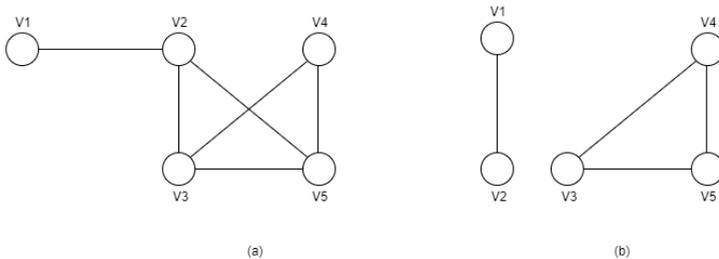
Graf dapat digambarkan dengan menggunakan titik untuk simpul dan garis untuk busur. Garis untuk sebuah busur $v_i v_j$ digambar antara simpul v_i dan v_j . Gambar 2.1 menunjukkan sebuah contoh graf tak-berarah, $\mathcal{G} = (V, E)$ dengan $V = \{v_1, v_2, \dots, v_5\}$ dan $E = \{v_1 v_2, v_2 v_3, v_3 v_4, v_3 v_5, v_2 v_5, v_4 v_5\}$.



Gambar 2.1 Graf tak-berarah dengan 5 simpul

Terdapat beberapa istilah yang sering digunakan dalam analisa graf diantaranya yaitu ketetanggaan (*adjacency*) dan persekitaran (*neighborhood*). Simpul v_i dan v_j disebut bertetanggaan (*adjacent*) apabila terdapat busur diantara keduanya, disimbolkan dengan $v_i \sim v_j$. Dalam hal ini, busur $v_i v_j$ dikatakan bersisian (*incident*) dengan simpul v_i dan v_j . Persekitaran $N(i) \subseteq V$ dari simpul v_i didefinisikan sebagai himpunan $\{v_j \in V \mid v_i v_j \in E\}$, dengan kata lain, himpunan semua simpul yang bertetanggaan dengan v_i . Dalam graf tak-berarah jika $v_j \in N(i)$, maka $v_i \in N(j)$.

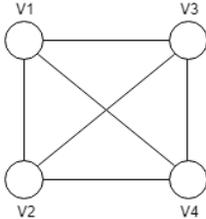
Jalur (*path*) dengan panjang n dari simpul v_x ke v_y adalah barisan dengan $n + 1$ simpul berbeda yang berawal dari v_x dan berakhir di v_y sedemikian sehingga setiap simpul yang berurutan bertetanggaan. Suatu graf \mathcal{G} dikatakan terhubung (*connected*) apabila selalu terdapat jalur yang menghubungkan dua simpul dalam graf tersebut. Graf dikatakan tak-terhubung (*disconnected*) jika keadaan tersebut tidak terpenuhi. Contoh graf terhubung dan graf tak-terhubung dapat dilihat pada Gambar 2.2.



Gambar 2.2 (a) Graf terhubung (b) Graf tak-terhubung

Dua buah graf terkadang memiliki struktur yang sama tetapi penamaan simpul dan busurnya ataupun penggambarannya berbeda. Dalam hal ini diperkenalkan konsep isomorfisme. Suatu graf \mathcal{G}_1 dikatakan isomorfik dengan graf \mathcal{G}_2 jika terdapat korespondensi satu satu f , dari $V(\mathcal{G}_1)$ ke $V(\mathcal{G}_2)$ sedemikian sehingga $v_i v_j \in E(\mathcal{G}_1)$ jika dan hanya jika $f(v_i) f(v_j) \in E(\mathcal{G}_2)$. Jika suatu graf \mathcal{G}_1 isomorfik dengan graf \mathcal{G}_2 maka dapat ditulis $\mathcal{G}_1 = \mathcal{G}_2$. Korespondensi satu satu $f: V(\mathcal{G}_1) \rightarrow V(\mathcal{G}_2)$ disebut isomorfisme.

Suatu graf \mathcal{G} dengan n simpul dikatakan sebagai graf lengkap (*complete*) apabila setiap pasang simpulnya bertetangga. Graf lengkap dengan n simpul diberi nama K_n .



Gambar 2.3 Graf lengkap K_4

2.1.2 Matriks Representasi

Graf dapat direpresentasikan dalam bentuk matriks. Matriks ini memiliki sifat-sifat yang merefleksikan sifat graf. Matriks-matriks tersebut yaitu matriks derajat (*degree*), matriks ketetangaan, matriks bersisian, dan matriks Laplacian. Untuk suatu graf tak-berarah \mathcal{G} , derajat (*degree*) dari suatu simpul v_i , disimbolkan $d(v_i)$, adalah jumlah anggota dari himpunan persekitaran $N(i)$. Dengan kata lain, derajat dari simpul v_i sama dengan banyaknya simpul yang bertetangga dengannya. Matriks derajat dari suatu graf \mathcal{G} adalah sebuah matriks diagonal dengan derajat tiap simpul pada diagonalnya, yaitu

$$\Delta(\mathcal{G}) = \begin{bmatrix} d(v_1) & 0 & \dots & 0 \\ 0 & d(v_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d(v_n) \end{bmatrix} \quad (2.1)$$

dengan n adalah jumlah simpul.

Matriks ketetangaan $A(\mathcal{G})$ adalah matriks simetri $n \times n$ yang menggambarkan hubungan ketetangaan antar simpul pada graf \mathcal{G} , yaitu

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & , \text{jika } v_i v_j \in E \\ 0 & , \text{lainnya} \end{cases} \quad (2.2)$$

Untuk suatu graf \mathcal{G} dengan orde n dan ukuran m , matriks bersisian (*incidence*) $B(\mathcal{G})$ adalah matriks $n \times m$ dimana

$$[B(\mathcal{G})]_{ij} = \begin{cases} 1 & , \text{jika } v_i \text{ dan } e_j \text{ berisian} \\ 0 & , \text{lainnya} \end{cases} \quad (2.3)$$

Matriks lain yang juga penting dalam teori graf adalah matriks Laplacian. Definisi sederhana dari matriks Laplacian dari suatu graf tak-berarah \mathcal{G} adalah:

$$L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G}) \quad (2.4)$$

Berikut ini adalah matriks derajat, matriks ketetanggaan, dan matriks Laplacian dari contoh graf pada Gambar 2.1:

$$\Delta(\mathcal{G}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

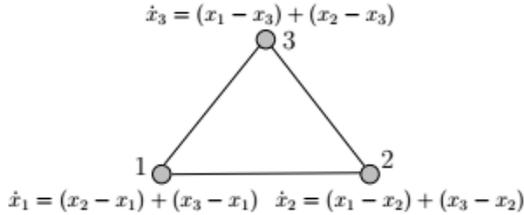
$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$L(\mathcal{G}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}$$

2.1.3 Consensus Protocol

Consensus merupakan salah satu permasalahan mendasar dalam sistem multiagen, dimana sekumpulan agen ditugaskan untuk menuju suatu nilai *state* yang sama. *Consensus protocol* berhubungan dengan berbagai permasalahan multiagen seperti *flocking*, *rendezvous*, *swarming*, *attitude alignment*, dan estimasi terdistribusi.

Consensus protocol melibatkan n agen yang saling terhubung melalui sebuah jalur pertukaran informasi. Laju perubahan *state* masing-masing agen ditentukan oleh jumlah total *state* relatifnya terhadap agen lain yang bertetanggaan (*adjacent*) dengannya. Gambar 2.4 menunjukkan *consensus protocol* pada tiga agen.



Gambar 2.4 Consensus protocol pada tiga agen

Persamaan pada Gambar 2.4 dapat ditulis ulang menjadi Persamaan 2.5

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j - x_i) \quad , i = 1, 2, 3, \dots, n \quad (2.5)$$

dimana $N(i)$ adalah persekitaran agen i .

Persamaan 2.5 diatas dapat ditulis ulang memanfaatkan matriks Laplacian sebagai berikut

$$\dot{x}(t) = -L(G)x(t) \quad (2.6)$$

dimana $L(G)$ adalah matriks Laplacian jaringan komunikasi antar-agen dan $x(t) = [x_{1(t)} \quad x_{2(t)} \quad \dots \quad x_{n(t)}]^T$.

Gambar 2.5 mengilustrasikan *consensus protocol* yang diaplikasikan pada 5 agen. Berdasarkan Gambar 2.5 terlihat bahwa agen-agen dengan nilai awal yang berbeda pada akhirnya menuju satu nilai yang sama.

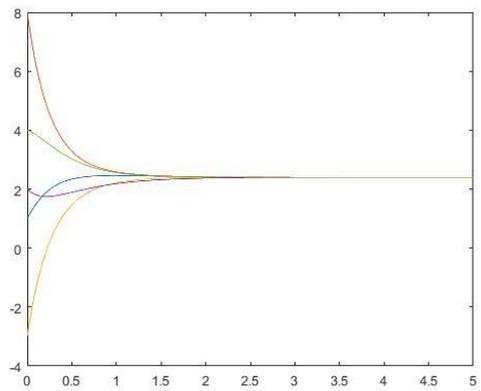
Salah satu penerapan *consensus protocol* adalah pada permasalahan *rendezvous*. Pada permasalahan ini, agen-agen yang posisi awalnya berbeda ditugaskan untuk berkumpul pada suatu titik. Penerapan *consensus protocol* untuk pemecahan permasalahan *rendezvous* diilustrasikan pada Gambar 2.6.

Gambar 2.6 menunjukkan perubahan posisi 5 agen dari posisi awal yang berbeda menuju satu posisi yang sama.

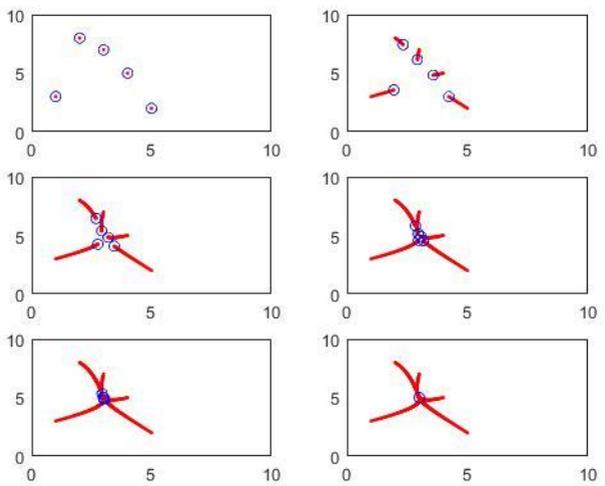
2.2 Robot Mobil

Pada berbagai literatur, istilah robot mobil bisa mengacu pada dua hal: *mobile robot* atau *car-like mobile robot*. Pada penelitian ini, istilah robot mobil mengacu pada terjemahan yang pertama. Dengan demikian robot mobil dapat diartikan sebagai suatu robot yang dapat berpindah-pindah posisi dalam lingkungan kerjanya. Lingkungan kerja robot dapat berupa daratan (*terrestrial*), air (*aquatic*), udara (*airborne*), maupun luar angkasa (*space*). Robot mobil dapat diaplikasikan pada berbagai bidang

diantaranya kesehatan, pertanian, kehutanan, militer, hingga eksplorasi luar angkasa.



Gambar 2.5 Nilai state 5 agen dengan *consensus protocol*



Gambar 2.6 Permasalahan *rendezvous*

Berdasarkan penggerakannya, robot mobil yang bekerja di daratan dibagi menjadi dua yaitu robot berkaki dan robot beroda. Robot berkaki

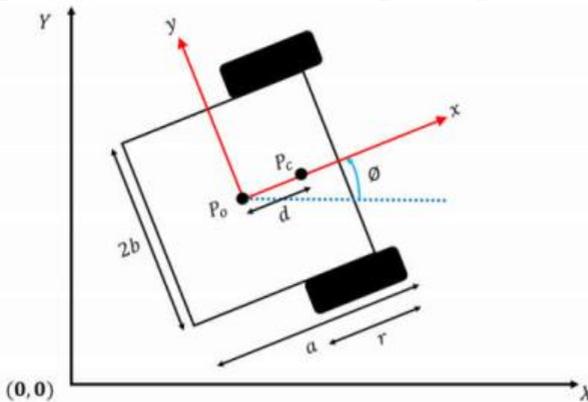
bergerak meniru gerakan pada hewan. Robot jenis ini umumnya digunakan untuk lingkungan kerja yang cenderung alamiah, kasar dan tak-terstruktur. Robot beroda lebih sering digunakan pada permukaan rata buatan manusia. Oleh karena itu, robot jenis ini sangat umum digunakan dalam dunia industri.

2.2.1 Model Kinematika Robot[4]

Model kinematik sebuah robot menyatakan sifat-sifat geometris pergerakan robot tanpa mempertimbangkan gaya-gaya yang mempengaruhinya. Pergerakan sebuah robot mobil beroda dipengaruhi oleh pergerakan tiap-tiap rodanya.

Robot mobil yang digunakan pada penelitian ini adalah robot mobil beroda jenis *differential drive*. Robot jenis ini memiliki dua roda yang dapat diatur kecepatannya secara terpisah. Gambar 2.7 mengilustrasikan sebuah robot mobil *differential drive* pada koordinat global $X - Y$.

P_o adalah titik asal (*origin*) koordinat lokal robot $x - y$, r adalah jari-jari roda robot, b merupakan lebar robot. Panjang robot dilambangkan dengan a , dan orientasi robot dilambangkan dengan θ .



Gambar 2.7 Robot mobil *differential drive*

Terdapat dua batasan nonholonomik yang membatasi gerak robot. Batasan pertama yaitu tidak adanya gerakan menyamping atau searah sumbu- y koordinat lokal robot. Secara matematis batasan ini dapat dituliskan sebagai:

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \quad (2.7)$$

dimana (x, y) adalah posisi robot dan θ adalah orientasi robot pada koordinat global. Batasan kedua yaitu tidak adanya slip pada roda robot yang secara matematis dituliskan sebagai berikut:

$$\dot{x} \cos \theta + \dot{y} \sin \theta + b\dot{\theta} = r\dot{\phi}_R \quad (2.8)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta - b\dot{\theta} = r\dot{\phi}_L \quad (2.9)$$

Batasan-batasan dalam persamaan di atas dapat ditulis ulang dalam bentuk matriks menjadi:

$$\Lambda(q)\dot{q} = 0 \quad (2.10)$$

dimana

$$\Lambda(q) = \begin{bmatrix} -\sin \theta & \cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & b & -r & 0 \\ \cos \theta & \sin \theta & -b & 0 & -r \end{bmatrix} \quad (2.11)$$

$$\dot{q} = [\dot{x} \quad \dot{y} \quad \dot{\theta} \quad \dot{\phi}_R \quad \dot{\phi}_L]^T \quad (2.12)$$

Kecepatan linear dari tiap roda yaitu

$$v = \frac{v_R + v_L}{2} = \frac{r\dot{\phi}_R + r\dot{\phi}_L}{2} \quad (2.13)$$

Kecepatan sudutnya yaitu

$$\omega = \frac{v_R - v_L}{2b} = \frac{r(\dot{\phi}_R - \dot{\phi}_L)}{2b} \quad (2.14)$$

Kecepatan robot pada koordinat local dapat dituliskan sebagai berikut

$$\dot{x}^r = \frac{r\dot{\phi}_R + r\dot{\phi}_L}{2} \quad (2.15)$$

$$\dot{y}^r = 0 \quad (2.16)$$

$$\dot{\theta} = \frac{r(\dot{\phi}_R - \dot{\phi}_L)}{2b} \quad (2.17)$$

Dalam bentuk matriks menjadi Persamaan 2.18

$$\begin{bmatrix} \dot{x}^r \\ \dot{y}^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (2.18)$$

Kecepatan robot pada koordinat global dapat dituliskan sebagai berikut:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (2.19)$$

Substitusi Persamaan 2.13 ke Persamaan 2.19 menghasilkan

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.20)$$

2.3 Konsep Dasar Pengaturan Formasi

Salah satu permasalahan mendasar dan penting dalam pengaturan multi-robot adalah pengaturan formasi. Secara garis besar terdapat tiga pendekatan dalam pengaturan formasi yaitu pendekatan *behavior-based*, *leader-follower*, dan *virtual structure*.

2.3.1 Pendekatan *Virtual Structure* [5]

Virtual structure didefinisikan sebagai sekumpulan elemen-elemen yang mempertahankan suatu hubungan geometris yang tetap terhadap satu sama lain dan terhadap satu kerangka acuan tertentu. Untuk mempermudah memahami definisi dari *virtual structure*, dapat memanfaatkan pengertian benda tegar. Benda tegar tersusun dari partikel-partikel yang posisi relatif antara satu sama lainnya tetap. Jika posisi partikel ke- i dilambangkan dengan r_i dan partikel ke- j dilambangkan dengan r_j , maka $|r_i - r_j| = d_{ij}$ konstan. Hal ini dikarenakan adanya gaya molekuler yang bekerja. Lebih lanjut, saat benda tegar bergerak, posisi partikel-partikel tersebut dapat dianggap statis terhadap suatu kerangka acuan lokal tertentu. Kerangka acuan ini ikut bergerak bersamaan dengan pergerakan benda tegar. *Virtual structure* pada dasarnya meniru struktur seperti pada benda tegar. Pada *virtual structure*, titik virtual dianggap seperti partikel-partikel tersebut dan suatu sistem pengaturan buatan digunakan untuk menggantikan peranan gaya molekuler.

Permasalahan bergerak dalam formasi memiliki dua tujuan utama yaitu bergerak ke arah yang ditentukan dan mempertahankan hubungan geometris yang tetap. Gambar 2.8 merepresentasikan permasalahan ini.

Pada Gambar 2.8, *virtual structure* pada kerangka acuan lokalnya dipetakan ke dalam kerangka acuan global dengan transformasi I_R^W .

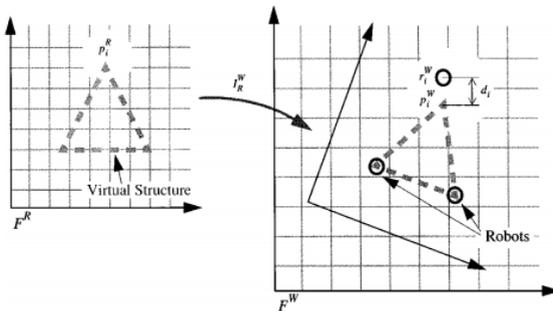
Virtual structure berbentuk segitiga ini kemudian menjadi referensi bagi robot.

Permasalahan bergerak dalam formasi dapat diperjelas sebagai berikut:

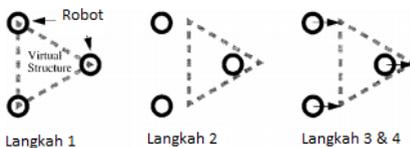
Terdapat n buah robot yaitu robot $1, 2, \dots, n$ dengan posisinya dalam kerangka koordinat global yaitu $r_1^W, r_2^W, \dots, r_n^W$. Tentukan sebuah *virtual structure* dengan n buah titik dengan posisinya dalam kerangka koordinat lokal yaitu $p_1^R, p_2^R, \dots, p_n^R$. Petakan $p_1^R, p_2^R, \dots, p_n^R$ ke $p_1^W, p_2^W, \dots, p_n^W$ menggunakan transformasi I_R^W . Robot akan bergerak dalam formasi apabila $r_i^W = p_i^W, i = 1, 2, \dots, n$ untuk semua waktu.

Solusi umum dari permasalahan tersebut disajikan dalam sebuah algoritma (juga pada Gambar 2.9) sebagai berikut:

1. Posisikan *virtual structure* (VS) pada posisi aktual robot
2. Pindahkan VS sejauh Δx dan $\Delta\theta$
3. Hitung trayektori tiap robot untuk memindahkan robot ke titik VS yang menjadi referensi
4. Sesuaikan kecepatan roda untuk mengikuti trayektori yang telah dihasilkan pada langkah 3
5. Kembali ke langkah 1



Gambar 2.8 Representasi visual dari masalah bergerak dalam formasi



Gambar 2.9 Langkah-langkah dalam algoritma untuk bergerak dalam formasi

2.4 Metode *Artificial Potential Field* [6]

Path planning dan penghindaran rintangan merupakan masalah utama pada pengaturan robot mobil. Masalah *path planning* diklasifikasikan menjadi dua, yaitu statis dan dinamis. Pada permasalahan statis, seluruh lingkungan kerja robot sudah diketahui *a priori* dan tidak mengalami perubahan. Pada permasalahan dinamis, lingkungan kerja robot hanya diketahui sebagian atau bahkan tidak diketahui sama sekali.

Terdapat beberapa pendekatan untuk memecahkan masalah *path planning* pada robot mobil, diantaranya: *Visibility Graph*(VG), *Voronoi Diagram*(VD), *Virtual Force Field*(VFF), *Virtual Force Histogram*(VFH), dan *Artificial Potential Field*(APF). Pada penelitian ini akan digunakan pendekatan APF. Fungsi potensial (*potential function*) merupakan jumlah dari potensial atraktif yang menarik robot ke arah tujuan (target) yang diinginkan dan total potensial repulsif yang mendorong robot menjauh dari rintangan.

Dalam metode APF, robot dianggap sebagai sebuah titik yang berada pada suatu medan gaya buatan (*artificial force field*). Medan buatan tersebut terdiri atas medan atraktif yang berasal dari target dan medan repulsif yang berasal dari rintangan.

Untuk robot dengan posisi $q = (x, y)^T$, APF didefinisikan sebagai:

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (2.21)$$

dimana $U(q)$ adalah APF, $U_{att}(q)$ adalah medan atraktif, dan $U_{rep}(q)$ adalah medan repulsif.

Medan atraktif yang dihasilkan oleh target adalah sebagai berikut:

$$U_{att}(q) = \frac{1}{2}k(q - q_g)^2 = \frac{1}{2}k\rho_{goal}^2(q) \quad (2.22)$$

dimana k adalah koefisien positif APF, $q_g = (x_g, y_g)^T$ adalah posisi target, dan $\rho_{goal}(q) = \|q - q_g\|$ adalah jarak antara posisi robot dengan posisi target.

Medan repulsif yang dihasilkan oleh satu rintangan adalah:

$$U_{rep}(q) = \begin{cases} 0, & \rho(q) \geq \rho_0 \\ \frac{1}{2}\eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right), & \rho(q) < \rho_0 \end{cases} \quad (2.23)$$

dimana η adalah faktor skala positif, $\rho(q)$ adalah jarak terdekat antara robot dengan rintangan, ρ_0 adalah jarak terjauh pengaruh rintangan.

Rintangan tidak mempengaruhi robot jika jarak antara keduanya lebih besar dari ρ_0 .

Negatif gradien dari APF disebut gaya artifisial (*artificial force*). Gaya atraktif adalah negatif gradien dari medan atraktif dan gaya repulsif adalah negatif gradien dari medan repulsif. Maka gaya artifisial adalah:

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = F_{att}(q) + F_{rep}(q) \quad (2.24)$$

dimana $F(q)$ adalah gaya artifisial, $F_{att}(q)$ adalah gaya atraktif, dan $F_{rep}(q)$ adalah gaya repulsif. Berdasarkan pengertian tersebut dan persamaan (2.22), maka didapatkan gaya atraktif:

$$F_{att}(q) = -\nabla U_{att}(q) = -\frac{1}{2}k\nabla\rho_{goal}^2(q) = -k(q - q_g) \quad (2.25)$$

Komponen $F_{att}(q)$ pada sumbu x dan sumbu y masing-masing adalah

$$\begin{aligned} F_{attx}(q) &= -k(x - x_g) \\ F_{atyy}(q) &= -k(y - y_g) \end{aligned} \quad (2.26)$$

Gaya repulsifnya adalah sebagai berikut:

$$F_{rep}(q) = \begin{cases} 0, & \rho(q) \geq \rho_0 \\ \eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left(\frac{1}{\rho^2(q)} \right) \nabla \rho(q), & \rho(q) < \rho_0 \end{cases} \quad (2.27)$$

$$F_{rep}(q) = \begin{cases} 0, & \rho(q) \geq \rho_0 \\ \eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left(\frac{1}{\rho^2(q)} \right) \left(\frac{q - q_c}{\|q - q_c\|} \right), & \rho(q) < \rho_0 \end{cases} \quad (2.28)$$

Komponen vektor gaya repulsif diatas pada sumbu- x dan sumbu- y masing-masing adalah:

$$F_{repx}(q) = \begin{cases} 0, & \rho(q) \geq \rho_0 \\ \eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left(\frac{1}{\rho^2(q)} \right) \left(\frac{x - x_c}{\|q - q_c\|} \right), & \rho(q) < \rho_0 \end{cases} \quad (2.29)$$

$$F_{repy}(q) = \begin{cases} 0, & \rho(q) \geq \rho_0 \\ \eta \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left(\frac{1}{\rho^2(q)} \right) \left(\frac{y - y_c}{\|q - q_c\|} \right), & \rho(q) < \rho_0 \end{cases} \quad (2.30)$$

Jika terdapat lebih dari satu rintangan, maka medan repulsif totalnya merupakan penjumlahan medan repulsif semua rintangan. Misal terdapat n rintangan maka:

$$U(q) = U_{att}(q) + \sum_{i=1}^n U_{rep_i}(q) \quad (2.31)$$

Sehingga gaya artifisialnya menjadi:

$$F(q) = F_{att}(q) + \sum_{i=1}^n F_{rep_i}(q) \quad (2.32)$$

[Halaman ini sengaja dikosongkan]

BAB 3 PERANCANGAN SISTEM

Dalam bab ini dipelajari perancangan *control law* yang dapat digunakan untuk pengaturan formasi robot mobil.

3.1 Gambaran Umum Sistem

Pada penelitian ini, robot ditugaskan untuk membentuk formasi, menuju suatu titik tujuan yang telah ditentukan dengan tetap mempertahankan formasi, dan menghindari rintangan yang terdeteksi.

Setiap robot memiliki sensor yang dapat mendeteksi objek disekitar robot. Sensor ini memberi data berupa posisi objek. Sebuah robot i dikatakan terhubung dengan robot j apabila terdapat sebuah jalur komunikasi yang memungkinkan robot i untuk bertukar informasi dengan robot j . Jalur komunikasi ini bersifat dua arah. Adapun informasi yang dikomunikasikan adalah berupa posisi dan orientasi robot. Pada penelitian ini, jalur komunikasi dianggap tetap dan tidak dipengaruhi oleh jarak antar robot. Robot dan jaringan komunikasi antar robot dapat direpresentasikan dalam bentuk graf. Robot direpresentasikan sebagai simpul dan jalur komunikasi yang menghubungkan dua robot direpresentasikan sebagai busur.

3.1.1 Spesifikasi Fisik Robot

Spesifikasi robot yang digunakan dalam simulasi pada penelitian ini adalah seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Spesifikasi robot

Deskripsi	Nilai	Satuan
Lebar Robot	3	cm
Panjang Robot	4	cm
Diameter roda	3	cm
Kecepatan maksimum, v_{max}	5	cm/s
Kecepatan angular maksimum, ω_{max}	3.14	rad/s

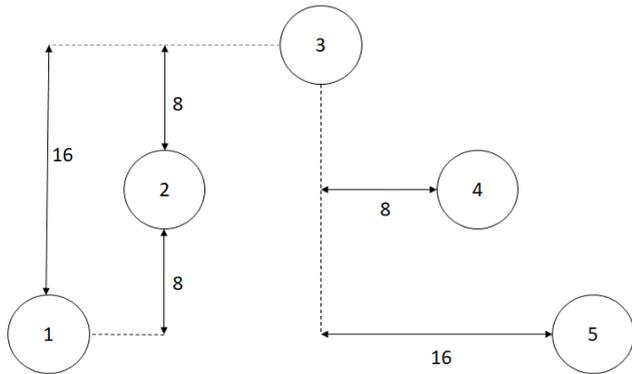
3.1.2 Deskripsi Lingkungan Kerja Robot

Lingkungan kerja robot berupa bidang dua dimensi dengan panjang dan lebar yang bervariasi pada tiap simulasi. Untuk simulasi yang

melibatkan rintangan, rintangan direpresentasikan sebagai sebuah lingkaran atau poligon (segi banyak).

3.1.3 Spesifikasi Formasi

Bentuk formasi yang akan dibentuk pada penelitian ini berupa bentuk V terbalik dan belah ketupat. Gambar 3.1 dan Tabel 3.2 mendeskripsikan bentuk V terbalik dengan lebih lengkap. Gambar 3.1 menunjukkan pola formasi yang diinginkan dengan jarak antar robot pada sumbu- x dan sumbu- y . Tabel 3.2 menunjukkan jarak antar robot secara keseluruhan.

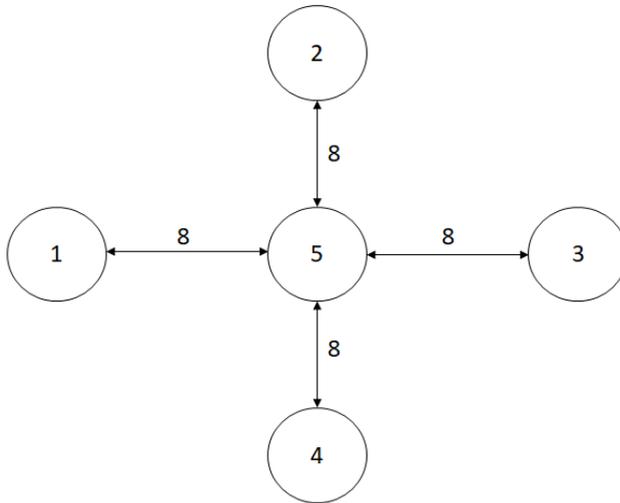


Gambar 3.1 Spesifikasi bentuk formasi V terbalik

Tabel 3.2 Jarak antar robot pada formasi V terbalik

Robot	1	2	3	4	5
1	0	11,3137	22,6274	25,2982	32
2	11,3137	0	11,3137	16	25,2982
3	22,6274	11,3137	0	11,3137	22,6274
4	25,2982	16	11,3137	0	11,3137
5	32	25,2982	22,6274	11,3137	0

Gambar 3.2 dan Tabel 3.3 menunjukkan bentuk formasi berupa belah ketupat lengkap dengan jarak antar robot yang diinginkan.



Gambar 3.2 Spesifikasi bentuk formasi belah ketupat

Tabel 3.3 Jarak antar robot pada formasi belah ketupat

Robot	1	2	3	4	5
1	0	11,3137	16	11,3137	8
2	11,3137	0	11,3137	16	8
3	16	11,3137	0	11,3137	8
4	11,3137	16	11,3137	0	8
5	8	8	8	8	0

Selanjutnya, bentuk formasi V terbalik akan disebut sebagai bentuk Formasi I (satu) dan bentuk formasi belah ketupat akan disebut sebagai bentuk Formasi II(dua).

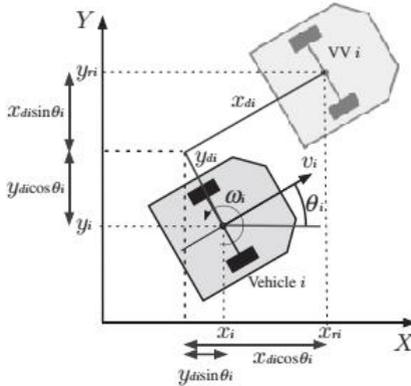
3.2 Perancangan *Control Law*

Perancangan *control law* dilakukan secara bertahap dimulai dengan pembentukan formasi. *Control law* kemudian dikembangkan dengan memasukkan penghindaran tabrakan antar-robot (*collision avoidance*). Selanjutnya dilakukan pengembangan berupa penghindaran rintangan (*obstacle avoidance*). Terakhir, dirancang *control law* yang

memungkinkan robot dalam formasi bergerak menuju suatu titik target yang ditentukan.

3.2.1 Pembentukan Formasi

Setiap robot mobil memiliki sebuah robot virtual seperti pada Gambar 3.3.



Gambar 3.3 Robot nyata dan robot virtualnya [7]

Hubungan geometris antara robot ke- i dengan robot virtualnya dijelaskan pada Persamaan 3.1 sebagai berikut:

$$\begin{bmatrix} x_{ri} \\ y_{ri} \\ \theta_{ri} \end{bmatrix} = \begin{bmatrix} x_i + x_{di} \cos \theta_i - y_{di} \sin \theta_i \\ y_i + x_{di} \sin \theta_i + y_{di} \cos \theta_i \\ \theta_i \end{bmatrix} \quad (3.1)$$

dimana $x_{ri}, y_{ri},$ dan θ_{ri} berturut-turut adalah posisi dan orientasi robot virtual ke- i dan $x_i, y_i,$ dan θ_i adalah posisi dan orientasi robot nyata ke- i . x_{di} dan y_{di} adalah posisi robot virtual ke- i pada koordinat lokal robot nyata ke- i .

Penurunan terhadap waktu Persamaan 3.1 menghasilkan

$$\begin{bmatrix} \dot{x}_{ri} \\ \dot{y}_{ri} \\ \dot{\theta}_{ri} \end{bmatrix} = \begin{bmatrix} \dot{x}_i + (-x_{di} \sin \theta_i - y_{di} \cos \theta_i) \dot{\theta}_i \\ \dot{y}_i + (x_{di} \cos \theta_i - y_{di} \sin \theta_i) \dot{\theta}_i \\ \dot{\theta}_i \end{bmatrix} \quad (3.2)$$

Seperti yang telah dijelaskan pada Subbab 2.2 bahwa

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos \theta_i \\ v_i \sin \theta_i \\ \omega_i \end{bmatrix} \quad (3.3)$$

Substitusi Persamaan 3.3 ke Persamaan 3.2 menghasilkan

$$\begin{bmatrix} \dot{x}_{ri} \\ \dot{y}_{ri} \\ \dot{\theta}_{ri} \end{bmatrix} = \begin{bmatrix} v_i \cos \theta_i + (-x_{di} \sin \theta_i - y_{di} \cos \theta_i) \omega_i \\ v_i \sin \theta_i + (x_{di} \cos \theta_i - y_{di} \sin \theta_i) \omega_i \\ \omega_i \end{bmatrix} \quad (3.4)$$

Persamaan 3.4 dapat ditulis ulang dalam bentuk matriks menjadi

$$\begin{bmatrix} \dot{x}_{ri} \\ \dot{y}_{ri} \\ \dot{\theta}_{ri} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -x_{di} \sin \theta_i - y_{di} \cos \theta_i \\ \sin \theta_i & x_{di} \cos \theta_i - y_{di} \sin \theta_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (3.5)$$

Dengan memperkenalkan matriks B_i dan B_θ sebagai

$$B_i = \begin{bmatrix} \cos \theta_i & -x_{di} \sin \theta_i - y_{di} \cos \theta_i \\ \sin \theta_i & x_{di} \cos \theta_i - y_{di} \sin \theta_i \end{bmatrix} \quad (3.6)$$

$$B_\theta = [0 \quad 1]$$

Substitusi 3.6 ke Persamaan 3.5 diperoleh

$$\begin{bmatrix} \dot{x}_{ri} \\ \dot{y}_{ri} \\ \dot{\theta}_{ri} \end{bmatrix} = \begin{bmatrix} B_i \\ B_\theta \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (3.7)$$

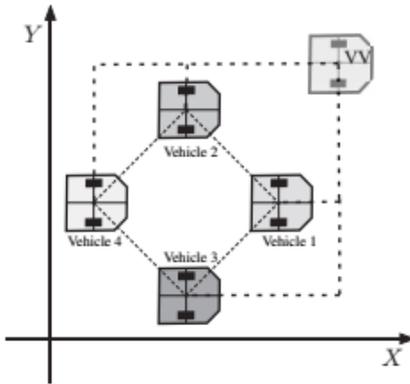
Dengan menggunakan *consensus protocol*, dapat dirancang sebuah *control law* untuk pembentukan formasi sebagai berikut

$$u_i = B_i^{-1} \left(-k \sum_{j \in N_i} (r_i - r_j) + \dot{r}_d \right) \quad (3.8)$$

dengan $u_i = [v_i \quad \omega_i]^T$ adalah sinyal kontrol yang akan menjadi masukan bagi robot ke- i . $r_i = [x_{ri} \quad y_{ri}]^T$ adalah posisi robot virtual ke- i pada koordinat global. \dot{r}_d adalah kecepatan referensi yang besarnya konstan.

Control law pada Persamaan 3.8 memungkinkan posisi seluruh robot virtual menuju satu titik menggunakan *consensus protocol*. *Consensus protocol* kemudian dijumlahkan dengan suatu kecepatan referensi guna

menyeragamkan orientasi robot. Perubahan posisi robot virtual pada tiap satu satuan waktu lalu dikonversikan menjadi kecepatan linear v_i dan kecepatan angular ω_i . Dengan demikian, robot virtual akan bergerak menuju satu titik yang sama dan secara simultan terus bergerak sesuai arah yang ditentukan oleh \dot{r}_d . Posisi robot virtual yang menuju satu titik ini bertindak seperti *virtual center* atau *virtual point* yang biasa digunakan pada pendekatan *virtual structure*. Robot dapat membentuk formasi dengan pemilihan x_{di} dan y_{di} yang tepat. Gambar 3.4 mengilustrasikan hal ini



Gambar 3.4 Posisi relatif robot virtual terhadap robot nyata

Control law pada Persamaan 3.8 dapat dikembangkan dengan menyamakan nilai x_{di} dan y_{di} untuk semua robot. Alih-alih berkumpul pada satu posisi, robot virtual ditugaskan untuk membentuk pola geometris sesuai bentuk formasi yang diinginkan. Pola geometris ini dibentuk dengan menggunakan sebuah titik virtual yang disebut *virtual center*. *Virtual center* digunakan sebagai acuan untuk pembentukan formasi dan secara geometris tidak harus berada di pusat atau tengah formasi. Pengembangan *control law* pada Persamaan 3.8 menjadi Persamaan 3.9.

$$u_i = B_i^{-1} \left(-k \sum_{j \in N_i} ((r_i - r_{ri}) - (r_j - r_{rj})) + \dot{r}_d \right) \quad (3.9)$$

dimana r_{ri} adalah posisi relatif referensi terhadap *virtual center*.

Control law pada Persamaan 3.9 bekerja dengan cara yang hampir sama dengan Persamaan 3.8. Perbedaan mendasar keduanya adalah pada Persamaan 3.8 robot virtual berkumpul pada satu titik dan bertindak seolah sebagai *virtual center*, sedangkan pada Persamaan 3.9 robot virtual menuju titik-titik referensi relatif terhadap *virtual center*.

Setelah membentuk formasi, robot ditugaskan untuk mempertahankannya. Dalam gerakan lurus (menuju satu arah), kecepatan referensi konstan dan sama bagi semua robot memungkinkan robot untuk mempertahankan formasi. Beberapa robot lebih cepat atau lebih lambat dari robot lain akan menyebabkan rusaknya formasi. Pada kedua *control law* yang telah dirancang, kecepatan referensi hanya sebagai *feedforward term*. Hal ini dapat memungkinkan kecepatan robot tidak menuju kecepatan referensi yang diinginkan atau dengan kata lain, error kecepatannya tidak menuju nol. Meskipun error kecepatan menuju nol, waktu yang dibutuhkan bisa cukup lama. Oleh karena itu, dirancang sebuah *control law* baru dengan mempertimbangkan error kecepatan sebagai berikut

$$\begin{aligned} \dot{v}_{ri} &= -k_{vr}(v_{ri} - v^*) \\ u_i &= B_i^{-1} \left(-k \sum_{j \in N_i} ((r_i - r_{ri}) - (r_j - r_{rj})) + v_{ri} \right) \end{aligned} \quad (3.10)$$

dimana v^* adalah kecepatan referensi. Dengan *control law* tersebut, maka robot bukan hanya membentuk formasi tetapi juga menuju nilai kecepatan referensi yang diinginkan.

3.2.2 Pembentukan Formasi dan Penghindaran Tabrakan

Sejauh ini *control law* yang dirancang hanya mempertimbangkan pembentukan formasi dan kontrol kecepatan untuk menuju kecepatan referensi yang diinginkan. Pada kondisi tertentu, robot bisa mengalami tabrakan (*collision*) dengan robot lain. Untuk itu perlu dirancang *control law* yang dapat mengatasi permasalahan ini.

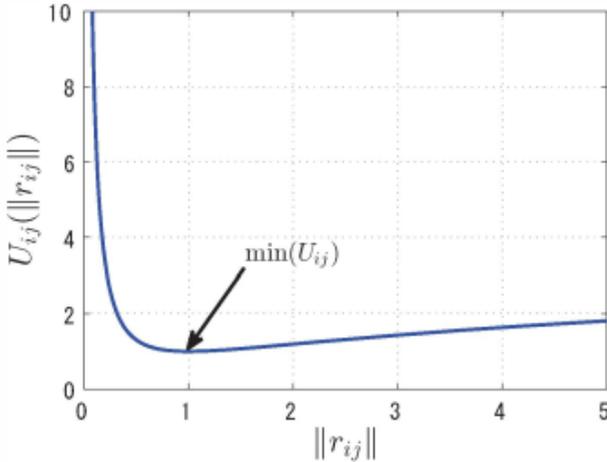
Salah satu upaya untuk mencegah tabrakan antar robot adalah dengan menggunakan model *flocking*. Dalam sistem multi-robot, *flocking* mempunyai tiga prinsip utama yaitu separasi, penjajaran, dan kohesi. Separasi mengarahkan robot untuk menjauhi robot lain yang posisinya terlalu dekat. Penjajaran membuat robot menuju satu arah yang sama. Kohesi mengarahkan robot untuk saling mendekat.

Penghindaran tabrakan antar-robot dapat memanfaatkan metode *artificial potential field*. Fungsi potensial yang dapat digunakan yaitu:

$$U_i = \sum_{j \in N_i} U_{ij}(\|r_{ij}\|) \quad (3.11)$$

$$U_{ij} = \frac{d}{\|r_{ij}\|} + \log(\|r_{ij}\|) \quad (3.12)$$

dimana $d > 0$ adalah parameter yang meminimumkan U_{ij} pada $\|r_{ij}\| = d, j \in N_i$. Fungsi potensial U_{ij} ditunjukkan pada Gambar 3.3



Gambar 3.5 Fungsi potensial U_{ij}

Berdasarkan Gambar 3.5, terlihat bahwa fungsi potensial U_{ij} nilainya semakin membesar saat jarak antar-robot yang terhubung, $\|r_{ij}\|$ semakin mengecil. Saat jarak antar-robot yang terhubung cukup jauh, nilai fungsi potensial akan mengecil. Nilai minimum didapat saat jarak antar-robot sebesar d .

Dengan mempertimbangkan penghindaran tabrakan antar-robot, dirancang sebuah *control law* baru sebagai berikut [7]:

$$\begin{aligned} \dot{v}_{ri} &= u_i^{co} + u_i^{ca} \\ u_i &= B_i^{-1} v_{ri} \end{aligned} \quad (3.13)$$

dimana

$$u_i^{co} = -k_{vr}(v_{ri} - v^*) - k_i \sum_{j \in N_i} \left((r_i - r_j) - (r_{ri} - r_{rj}) + k_v(v_{ri} - v_{rj}) \right) \quad (3.14)$$

$$u_i^{ca} = -\nabla_{r_i} U_i \quad (3.15)$$

Variabel u_i^{co} adalah sinyal kontrol terkait pembentukan dan pertahanan formasi. Persamaan 3.14 adalah pengembangan dari *control law* sebelumnya dengan penambahan penyeragaman kecepatan robot. Pada *control law* sebelumnya, kecepatan robot diupayakan menuju nilai yang ditentukan v^* , sedangkan pada Persamaan 3.14 kecepatan robot juga diseragamkan dengan robot-robot lain yang terhubung dengannya. Penyeragaman dilakukan dengan menggunakan *consensus protocol*. u_i^{ca} merupakan sinyal control terkait penghindaran tabrakan antar-robot. Penghindaran tabrakan antar-robot memanfaatkan gradien dari fungsi potensial U_{ij} .

3.2.3 Pembentukan Formasi dan Penghindaran Rintangan

Control law pada Persamaan 3.13 mengasumsikan lingkungan kerja robot tanpa rintangan. *Control law* tersebut dapat dikembangkan dengan penambahan penghindaran rintangan.

Pada penghindaran tabrakan antar-robot, fungsi potensial yang digunakan memanfaatkan informasi posisi robot yang saling terhubung. Penghindaran rintangan memanfaatkan informasi jarak antara robot dengan rintangan yang didapat dari sensor. Fungsi potensial untuk penghindaran rintangan adalah sebagai berikut:

$$U_i^o = \sum_{l \in N_i^o} k_o e^{-\frac{\|r_i^R - r_{il}^o\|}{l_o}} \quad (3.16)$$

dimana N_i^o adalah himpunan rintangan yang terdeteksi, $N_i^o = \emptyset$ jika tidak ada rintangan yang terdeteksi, r_i^R adalah posisi robot ke- i , r_{il}^o adalah posisi rintangan $l \in N_i^o$, k_o dan l_o adalah *gain* dan *range* dari gaya repulsif yang dihasilkan.

Control law untuk pembentukan formasi dan penghindaran rintangan adalah sebagai berikut [8]:

$$\dot{v}_{ri} = \begin{cases} u_i^{co} + u_i^{ca} & , N_i^o = \emptyset \\ \hat{u}_i^{co} + u_i^{ca} + u_i^{oa}, & \text{lainnya} \end{cases} \quad (3.17)$$

$$u_i = B_i^{-1} v_{ri}$$

dengan

$$\hat{u}_i^{co} = -k_{vr}(v_{ri} - v^*) - \hat{k}_i \sum_{j \in N_i} \left((r_i - r_j) - (r_{ri} - r_{rj}) + k_v(v_{ri} - v_{rj}) \right) \quad (3.18)$$

$$\hat{k}_i = \begin{cases} \frac{kI}{1 + D_a - d_{min,i}^o} & 0 < d_{min,i}^o < D_a \\ kI & (D_a < d_{min,i}^o) \end{cases} \quad (3.19)$$

$$u_i^{oa} = -\nabla_{r_i^R} U_i^o \quad (3.20)$$

dimana $d_{min,i}^o = \min l \in N_i^o (\|r_i^R - r_{il}^o\|)$ adalah jarak rintangan terdekat. Dengan D_o adalah radius sensor pada tiap robot, D_a ($0 < D_a \leq D_o$) adalah parameter desain.

Pada *control law* di atas (Persamaan 3.17), ketika rintangan terdeteksi cukup dekat, maka besarnya k_i yang terkait dengan *maintenance* formasi diperkecil sehingga untuk sementara robot dalam formasi dapat berpecah. Setelah rintangan tidak terdeteksi, nilai k_i akan kembali seperti semula sehingga robot akan membentuk formasi kembali.

Fungsi potensial pada Persamaan 3.16 disebut sebagai fungsi potensial eksponensial. Pada beberapa literatur, modifikasi dilakukan untuk menghasilkan gerakan robot yang lebih halus, salah satunya dengan menggunakan σ -norm. Fungsi potensial Khatib yang dimodifikasi dengan penggunaan σ -norm didefinisikan sebagai berikut [9]:

$$U_i^o = \sum_{l \in N_i^o} \left(\frac{1}{2} \right) k_o \left(\frac{1}{\|r_i^R - r_{il}^o\|_\sigma} - \frac{1}{l_o} \right)^2 \quad (3.21)$$

dengan

$$\|r_i^R - r_{il}^o\|_\sigma = \frac{1}{\epsilon} \left(\sqrt{1 + \epsilon \|r_i^R - r_{il}^o\|} - 1 \right) \quad (3.22)$$

dimana $\epsilon > 0$

3.2.4 Tracking Formasi dan Penghindaran Rintangan

Pada permasalahan *tracking* formasi, robot tidak hanya ditugaskan untuk membentuk dan mempertahankan formasi. Namun juga ditugaskan untuk bergerak menuju titik tujuan (target) atau mengikuti *trajectory* tertentu. Untuk menyelesaikan permasalahan ini, v^* perlu dimanipulasi. v^* pada *control law* yang telah dirancang bernilai konstan, dengan demikian robot dalam formasi hanya akan menuju satu arah yang sama. Untuk mengarahkan robot menuju titik tujuan tertentu, v^* diubah sebagai berikut [9]:

$$v^* = \frac{k_t(r_t - (r_i - r_{ri}))}{\|r_t - (r_i - r_{ri})\|} \quad (3.23)$$

dengan $r_t = [x_t \ y_t]^T$ adalah posisi titik tujuan.

3.3 Perancangan Simulasi dan Analisis

Simulasi dilakukan menggunakan perangkat lunak MATLAB dengan *robotic toolbox* [10]. Untuk analisis performansi formasi robot digunakan sebuah indeks performansi formasi modifikasi dari [11], yang didefinisikan sebagai berikut:

$$I(t) = \sum_{i=1}^n \sum_{(j=1)_{j \neq i}}^n |(\|r_i - r_j\| - \|r_{ri} - r_{rj}\|)|, \quad (3.24)$$

Indeks performansi diatas akan bernilai nol jika dan hanya jika jarak antar robot pada formasi yang terbentuk sama dengan jarak antar robot yang diinginkan seperti pada spesifikasi formasi.

[Halaman ini sengaja dikosongkan]

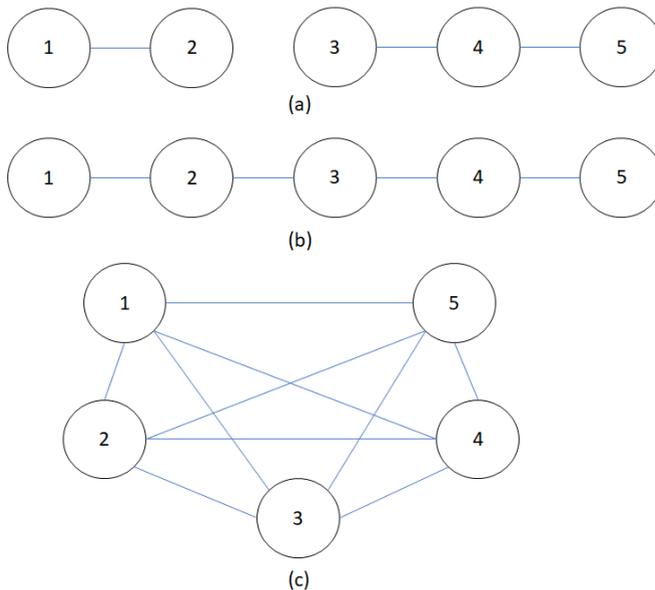
BAB 4

SIMULASI DAN ANALISIS SISTEM

Pada bab ini, sistem yang telah dirancang pada Bab 3 disimulasikan untuk mengetahui apakah objektif kontrol dari sistem telah terpenuhi. Hasil dari simulasi kemudian dianalisis untuk melihat performansi sistem.

4.1 Simulasi Pembentukan Formasi

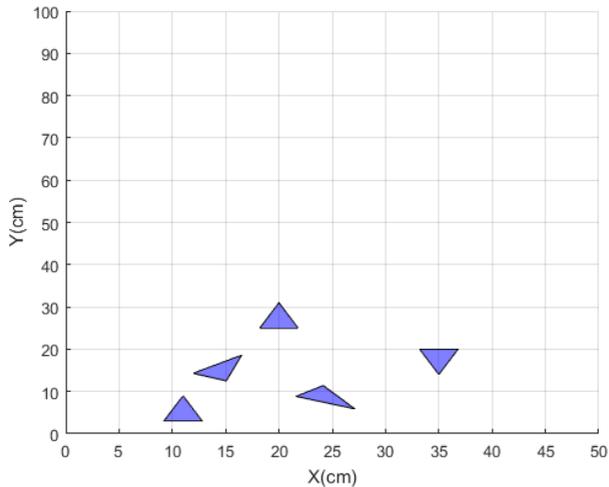
Pada simulasi pembentukan formasi, dianalisis keterkaitan antara konektivitas jaringan komunikasi antarrobot dengan formasi yang terbentuk. Jaringan komunikasi antarrobot direpresentasikan oleh sebuah graf. Jaringan tersebut berupa graf tak-terhubung, graf terhubung, dan graf lengkap. Gambar 4.1 mengilustrasikan konektivitas jaringan antar robot yang akan digunakan dalam simulasi. Posisi dan orientasi awal robot ditunjukkan pada Tabel 4.1 dan Gambar 4.2.



Gambar 4.1 (a) Graf tak-terhubung (b) Graf terhubung (c) Graf lengkap

Tabel 4.1 Posisi dan orientasi awal robot

Robot	$x(cm)$	$y(cm)$	$\theta(rad)$
1	35	17	$-\frac{\pi}{2}$
2	15	16	$\frac{\pi}{3}$
3	20	28	$-\frac{3\pi}{2}$
4	11	6	$\frac{\pi}{2}$
5	25	8	$-\frac{\pi}{4}$

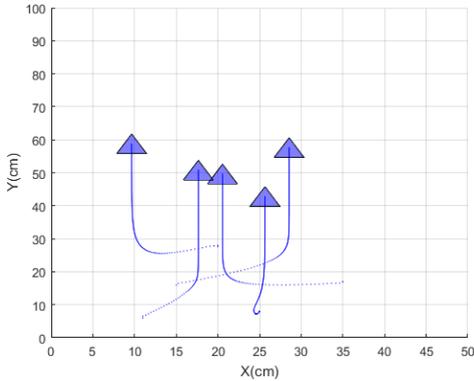


Gambar 4.2 Posisi dan orientasi awal robot

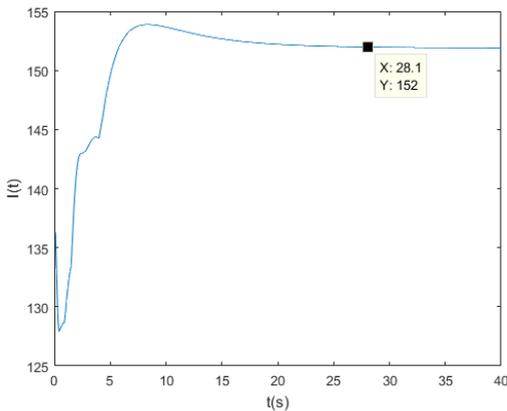
Parameter *gain* k sebesar 0,3 dan k_{vr} sebesar 1. Nilai parameter didapatkan berdasarkan *trial and error*. Kecepatan referensi robot v^* sebesar $[0 \ 2]^T$.

4.1.1 Graf Tak-terhubung

Simulasi dilakukan untuk sistem multi robot dengan jaringan seperti pada Gambar 4.1 (a) dan parameter serta kondisi awal seperti yang telah dijelaskan. Hasil simulasi ditampilkan pada Gambar 4.3.



Gambar 4.3 Trayektori robot graf tak-terhubung saat $t=20s$



Gambar 4.4 Indeks performansi formasi graf tak terhubung

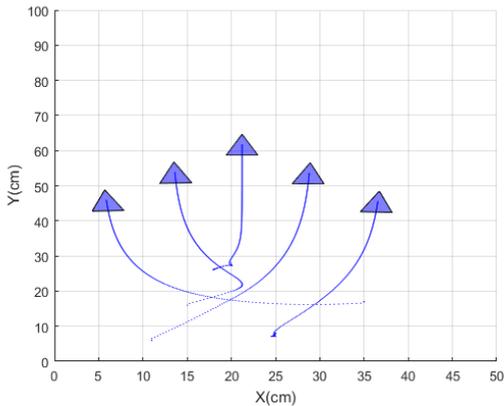
Gambar 4.3 menunjukkan posisi dan orientasi robot serta trayektori yang telah dilewati pada waktu $t=20s$. Simulasi dilanjutkan hingga waktu $t=50s$ namun robot tidak membentuk formasi yang diinginkan. Formasi yang terbentuk tetap berpola seperti saat $t=20s$.

Gambar 4.4 menunjukkan indeks performansi formasi dari robot. Indeks performansi menunjukkan nilai 152 saat sistem *steady state*. Besarnya nilai indeks performansi tersebut menunjukkan bahwa robot tidak membentuk formasi yang diinginkan.

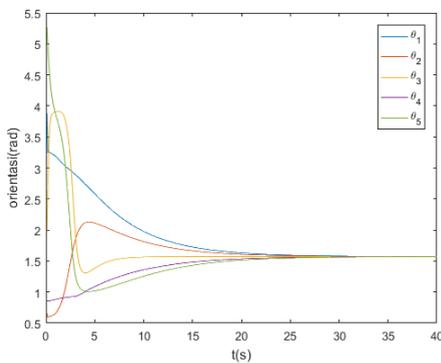
4.1.2 Graf Terhubung

Hasil simulasi dapat dilihat pada Gambar 4.5. Saat $t=20s$, robot sudah membentuk formasi yang diinginkan. Perubahan orientasi robot dari $t=0$ sampai $t=40s$ ditunjukkan pada Gambar 4.6.

Dari Gambar 4.6 terlihat bahwa orientasi robot yang awalnya beragam kemudian mulai menuju satu arah yang sama. Arah robot ditentukan oleh besarnya v^* . Kecepatan robot pada sumbu- x dan sumbu- y juga menuju ke satu nilai yang sama seperti pada Gambar 4.7.



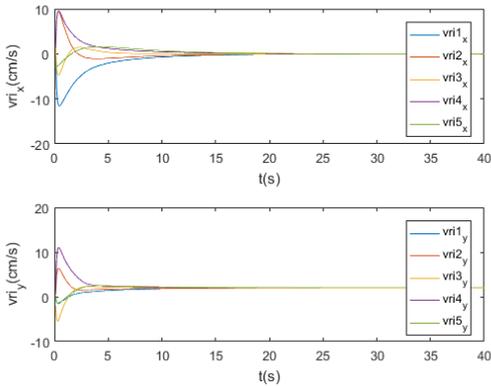
Gambar 4.5 Trayektori robot graf terhubung saat $t=20s$



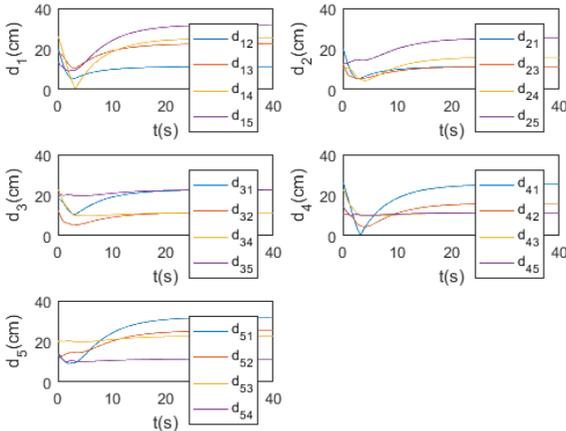
Gambar 4.6 Orientasi robot graf terhubung

Gambar 4.8 menunjukkan jarak antar robot ke- i dengan robot ke- j , d_{ij} . Dari gambar tersebut terlihat bahwa jarak antara robot 1 dengan robot 4 adalah sekitar 0 cm pada saat $t=4s$. Ini menunjukkan terjadinya tabrakan antara robot 1 dengan robot 4 saat detik ke 4.

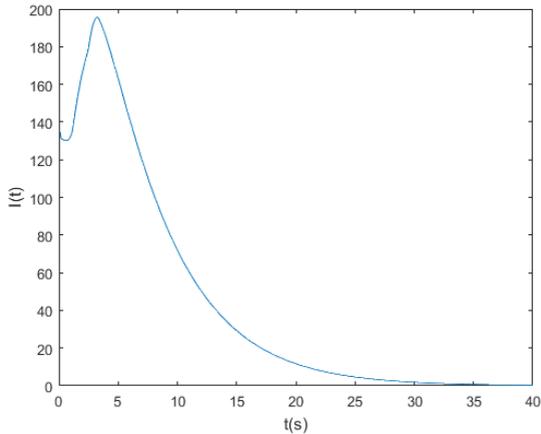
Indeks performansi formasi menuju nilai nol pada saat $t=20$ sekon seperti terlihat pada Gambar 4.9.



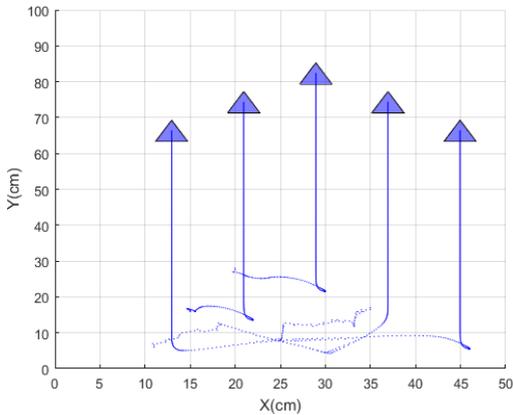
Gambar 4.7 Kecepatan robot pada sumbu- x dan sumbu- y



Gambar 4.8 Jarak antara robot i dengan robot j



Gambar 4.9 Indeks performansi formasi graf terhubung



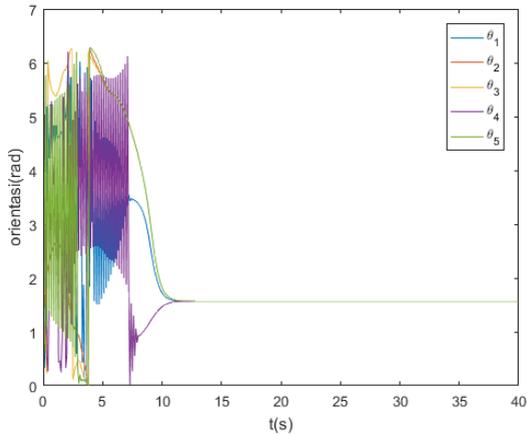
Gambar 4.10 Trayektori robot dengan graf lengkap

4.1.3 Graf Lengkap

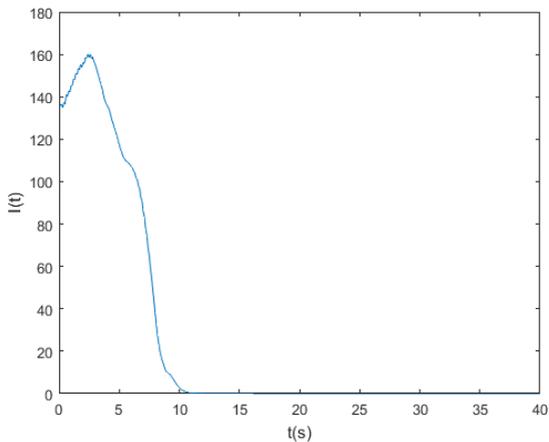
Hasil simulasi dapat dilihat pada Gambar 4.10. Gambar 4.10 menunjukkan robot berhasil membentuk formasi yang ditentukan. Adapun trayektori yang dihasilkan terlihat lebih kasar daripada trayektori robot dengan jaringan terhubung. Hal ini dikarenakan adanya osilasi yang

terjadi pada robot. Perubahan orientasi robot dan indeks performansi formasi berturut-turut ditunjukkan pada Gambar 4.11 dan Gambar 4.12.

Berdasarkan Gambar 4.11, terlihat bahwa robot berhasil pada rentang waktu antara detik ke 0 sampai detik ke 9. Dari Gambar 4.12 terlihat bahwa indeks performansi formasi menuju nilai nol yang berarti robot berhasil membentuk formasi.

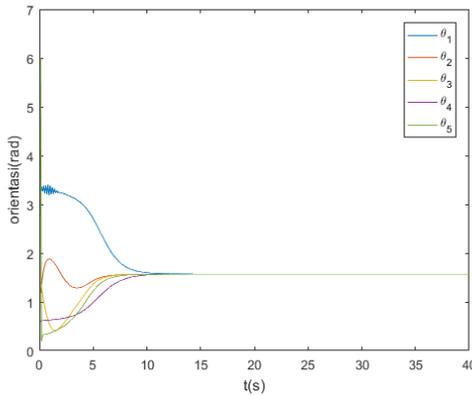


Gambar 4.11 Orientasi robot graf lengkap

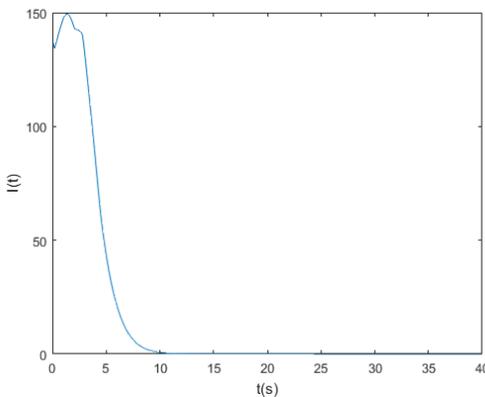


Gambar 4.12 Indeks performansi formasi graf lengkap

Osilasi terjadi akibat kombinasi antara graf lengkap dengan gain $k = 0,3$ menghasilkan masukan kontrol u_i yang terlalu besar. Osilasi dapat diredam dengan memperkecil nilai gain k . Gambar 4.13 dan 4.14 menunjukkan orientasi robot dan indeks performansi formasi saat $k = 0,1$. Gambar 4.13 menunjukkan bahwa osilasi dapat diredam dengan memperkecil nilai gain k . Dari gambar terlihat bahwa osilasi masih terjadi pada robot 1 tetapi dengan frekuensi dan amplitude yang lebih kecil dibandingkan dengan saat $k = 0,3$. Dari Gambar 4.14 terlihat bahwa formasi terbentuk lebih cepat.



Gambar 4.13 Orientasi robot graf lengkap dengan $k=0,1$

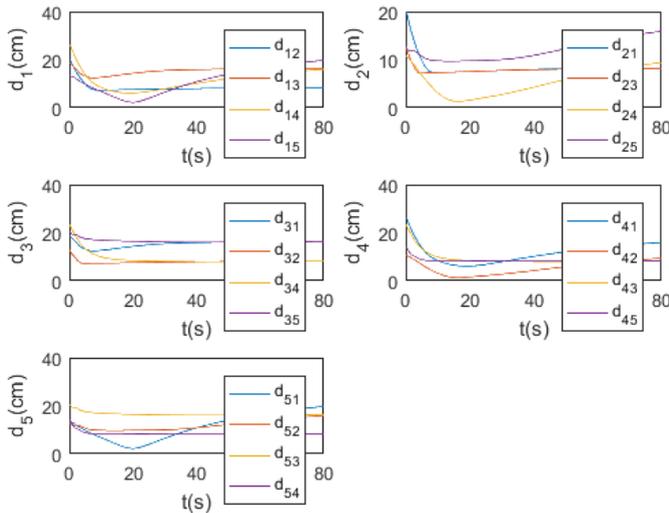


Gambar 4.14 IPF robot graf lengkap dengan $k=0,1$

4.2 Simulasi Pembentukan Formasi dengan Penghindaran Tabrakan Antar Robot

4.2.1 Graf Terhubung

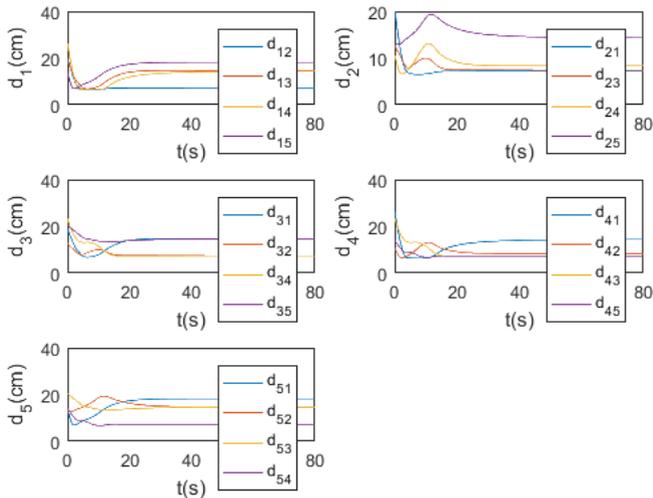
Jarak antar robot pada simulasi dengan jaringan berupa graf terhubung dapat dilihat pada Gambar 4.15. Pada gambar tersebut terlihat bahwa tabrakan antar robot masih tetap terjadi antara robot 1 dengan robot 5 dan robot 2 dengan robot 4. Alasan terjadinya tabrakan ini yaitu bahwa fungsi potensial yang digunakan untuk penghindaran tabrakan hanya menggunakan jarak antar robot yang terhubung. Pada simulasi ini, robot 1 dan robot 5 maupun robot 2 dengan robot 4 tidak memiliki jalur komunikasi antar robot atau dengan kata lain tidak terhubung sehingga tabrakan masih terjadi.



Gambar 4.15 d_{ij} dengan graf terhubung

4.2.2 Graf Lengkap

Hasil penerapan *control law* untuk pembentukan formasi dan penghindaran tabrakan antar robot dapat dilihat pada Gambar 4.16. Pada gambar tersebut, terlihat robot dapat menghindari tabrakan dengan robot lain. Meskipun begitu terdapat kelemahan dari sistem ini. Kelemahan



Gambar 4.16 d_{ij} dengan graf lengkap

sistem ini yaitu robot hanya dapat menghindari tabrakan dengan robot lain yang terhubung dengannya. Dengan demikian, untuk menjamin tidak adanya tabrakan sama sekali, jaringan komunikasi antar robot harus berupa graf lengkap. Perbaikan *control law* ini dengan memanfaatkan data sensor seperti pada *control law* pada Persamaan 3.17 dapat dilihat pada Subbab berikutnya.

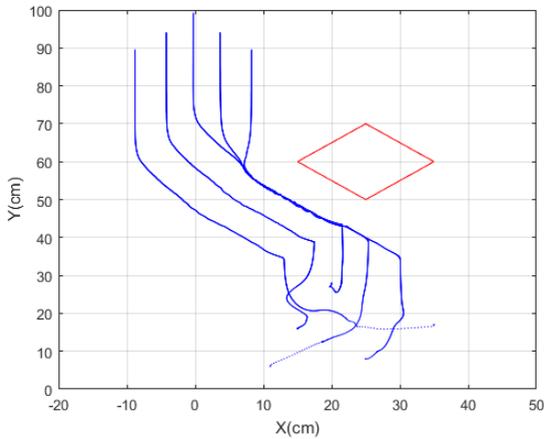
4.3 Simulasi Pembentukan Formasi dengan Penghindaran Rintangan

Pada subbab ini ditampilkan hasil simulasi penghindaran rintangan. Jaringan komunikasi robot merupakan graf lengkap. Analisis dilakukan dengan membandingkan penghindaran rintangan menggunakan APF eksponensial dengan penghindaran rintangan menggunakan APF modifikasi.

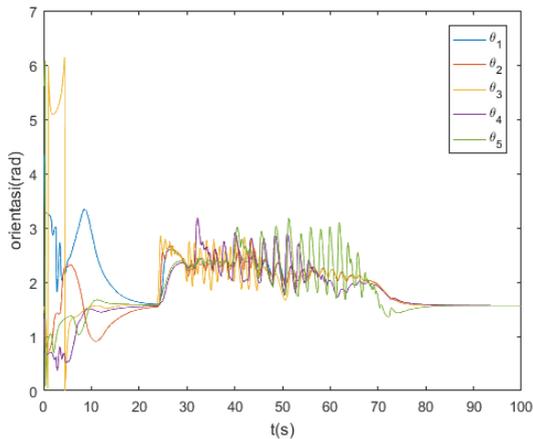
4.3.1 Metode APF Eksponensial

Trayektori hasil simulasi penghindaran rintangan dengan metode APF eksponensial ditunjukkan pada Gambar 4.17. Dari gambar tersebut dapat dilihat bahwa robot berhasil menghindari rintangan dengan tetap mempertahankan formasi.

Meskipun begitu, terjadi osilasi pada robot saat berada di dekat rintangan. Gambar 4.18 menunjukkan orientasi tiap robot. Dari gambar tersebut terlihat terjadi osilasi pada robot saat waktu antara $t=20s$ sampai $t=70s$. Gambar 4.19 menunjukkan indeks performansi formasi robot.



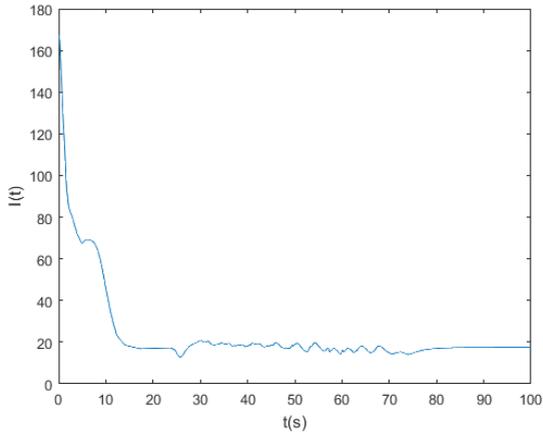
Gambar 4.17 Trayektori robot metode APF eksponensial



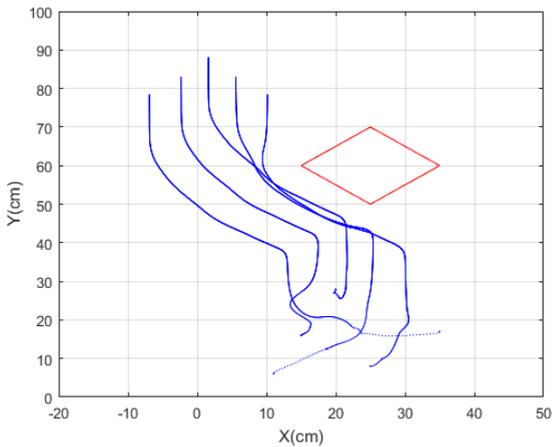
Gambar 4.18 Orientasi robot metode APF

4.3.2 Metode APF Modifikasi

Penggunaan fungsi potensial termodifikasi disimulasikan. Trayektori robot hasil simulasi menggunakan metode modifikasi ini ditunjukkan pada Gambar 4.20. Terlihat dari gambar tersebut bahwa robot berhasil membentuk formasi, menghindari rintangan, dan tetap mempertahankan bentuk formasi.



Gambar 4.19 Indeks performansi robot



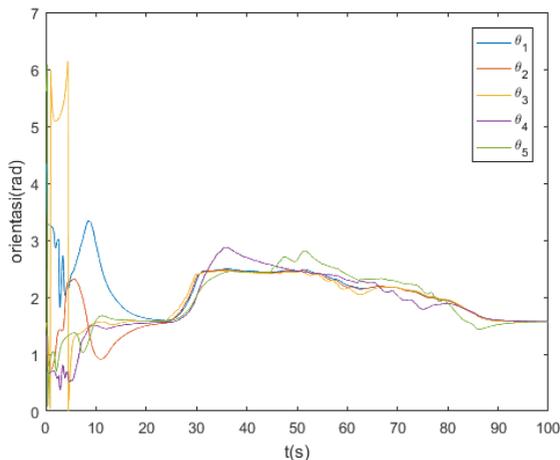
Gambar 4.20 Trayektori robot metode APF modifikasi

Secara visual tidak terlihat perbedaan trayektori yang dihasilkan oleh kedua metode. Orientasi robot selama simulasi ditunjukkan oleh Gambar 4.21. Gambar tersebut menunjukkan tidak adanya osilasi seperti pada Gambar 4.18.

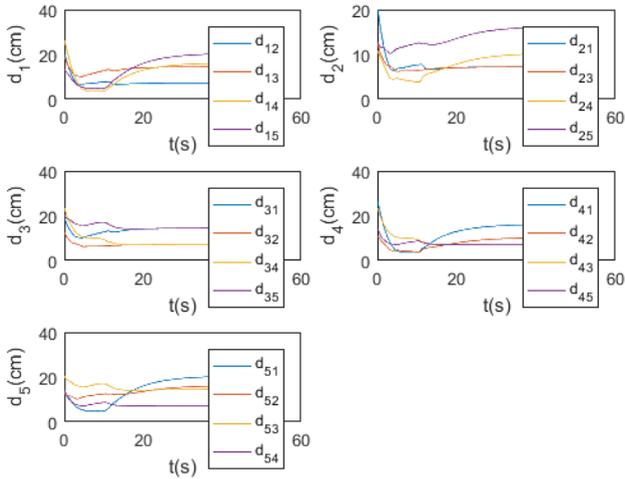
4.3.3 Analisis Penghindaran Tabrakan Antar Robot

Seperti yang telah dijelaskan pada Subbab 4.2, penghindaran tabrakan antar robot dengan hanya memanfaatkan fungsi potensial yang dihasilkan antar robot yang saling terhubung memiliki kelemahan. Untuk menjamin tidak adanya tabrakan, jaringan antar robot harus berupa graf lengkap. Dengan kata lain setiap robot harus mengetahui posisi robot lain. Penghindaran rintangan dapat mengatasi kelemahan tersebut. Robot yang berada dalam deteksi sensor robot lain akan menghasilkan gaya potensial layaknya rintangan, dengan demikian tabrakan dapat dihindari meskipun kedua robot tidak terhubung. Fungsi potensial yang digunakan adalah fungsi potensial hasil modifikasi.

Gambar 4.22 menunjukkan jarak antarrobot. Gambar tersebut dapat dibandingkan dengan Gambar 4.15 pada Subsubbab 4.2.1. Terlihat bahwa tabrakan antara Robot 1 dengan Robot 5 dan Robot 2 dengan Robot 4 dapat dihindari.



Gambar 4.21 Orientasi robot menggunakan metode APF modifikasi



Gambar 4.22 Jarak antar robot dengan APF modifikasi

4.4 Simulasi *Tracking* Formasi dengan Penghindaran Rintangan

Pada bagian ini *tracking* formasi disimulasikan. Bentuk formasi yang diinginkan adalah Formasi II seperti pada Subbab 3.1. Jaringan komunikasi antar robot berupa graf lengkap. Subbab ini dibagi menjadi dua bagian yaitu penghindaran sebuah rintangan dan pelintasan melalui celah sempit

4.4.1 Penghindaran Rintangan

Posisi dan orientasi awal robot ditunjukkan pada Tabel 4.2. Trayektori robot hasil dari simulasi dapat dilihat pada Gambar 4.23. Robot dengan posisi dan orientasi awal seperti ditunjukkan pada Tabel 4.2 ditugaskan untuk menuju titik $[120,100]^T$ ditunjukkan dengan tanda silang (x) warna merah. Rintangan berupa lingkaran dengan titik pusat berada pada $[50,50]^T$ dengan jari-jari sebesar 20cm.

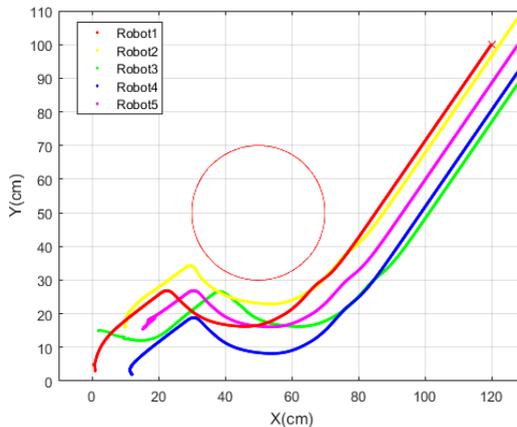
Robot membutuhkan waktu 80s untuk mencapai titik tujuan. Robot dalam formasi dapat menghindari rintangan dengan tetap mempertahankan formasi. Jarak terdekat robot dengan rintangan adalah

7,5926 , 5,2349 , 5,1977 masing-masing untuk robot 1, robot 2, dan robot 3. Rintangan tidak memasuki wilayah deteksi robot 4 dan robot 5.

Indeks performansi formasi ditunjukkan pada Gambar 4.24. Indeks performansi formasi tidak pernah mencapai nilai nol. Dengan kata lain formasi tidak sepenuhnya terbentuk. Meskipun begitu, pada waktu t=40 detik sampai t=80 detik, indeks menunjukkan nilai 1,001 yang berarti kesalahan formasi sangat kecil. Hal ini terjadi setelah tidak ada rintangan yang terdeteksi oleh robot. Nilai rata-rata IPF dari t=0 sampai t=80 adalah sebesar 8,2693.

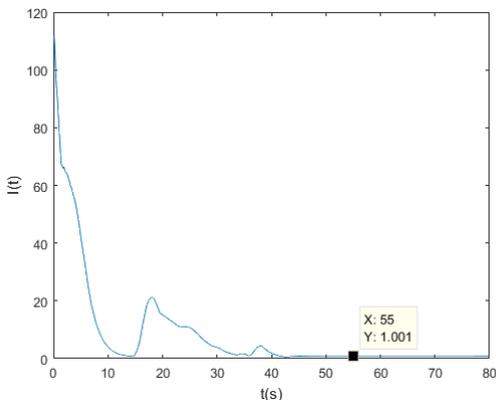
Tabel 4.2 Posisi dan orientasi awal robot

Robot	$x(cm)$	$y(cm)$	$\theta(rad)$
1	1	3	$-\frac{\pi}{2}$
2	10	16	$\frac{\pi}{3}$
3	2	15	$-\frac{3\pi}{2}$
4	12	2	$\frac{\pi}{4}$
5	19	19	$\frac{3\pi}{2}$

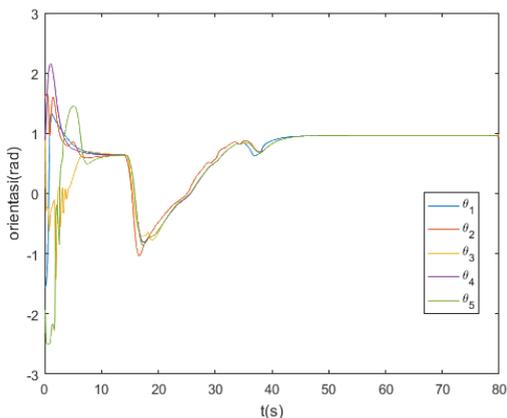


Gambar 4.23 Trayektori hasil *tracking* formasi dengan penghindaran rintangan

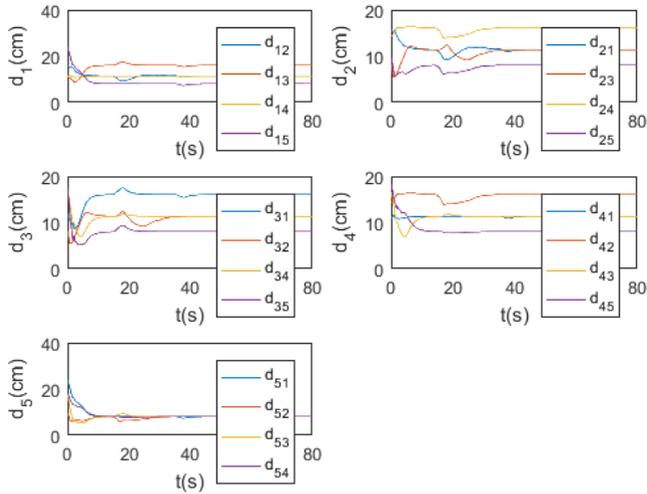
Gambar 4.25 menunjukkan orientasi robot dalam pergerakannya menuju titik tujuan. Terlihat bahwa robot cenderung memiliki orientasi yang sama. Pada Gambar 4.26 ditunjukkan jarak antarrobot. Gambar tersebut menunjukkan jarak antarrobot selalu berada diatas 5 cm (jarak aman). Jarak antar robot terdekat yang tercatat adalah sebesar 5,2288 cm yaitu antara robot 3 dan robot 5.



Gambar 4.24 IPF robot pada *tracking* formasi dengan penghindaran rintangan



Gambar 4.25 Orientasi robot pada *tracking* formasi dengan penghindaran rintangan



Gambar 4.26 Jarak antar robot pada *tracking* formasi dengan penghindaran rintangan

Tabel 4.3 Posisi dan orientasi awal robot

Robot	$x(cm)$	$y(cm)$	$\theta(rad)$
1	3	40	$-\frac{\pi}{2}$
2	10	35	$\frac{\pi}{2}$
3	15	30	$-\frac{3\pi}{2}$
4	10	25	$\frac{\pi}{4}$
5	1	20	$\frac{3\pi}{2}$

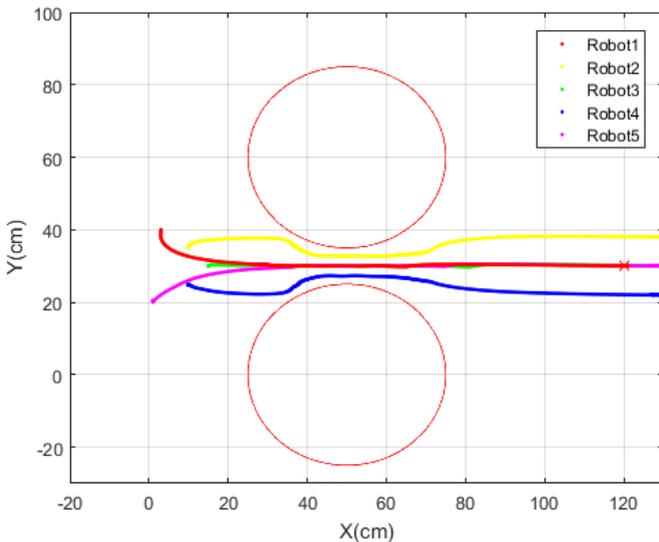
4.4.2 Pelintasan Celah Sempit

Pada bagian ini ditunjukkan hasil simulasi robot dalam formasi yang bergerak menuju titik tujuan dan dalam perjalanannya terdapat rintangan yang membentuk sebuah celah. Kondisi awal robot berupa posisi dan orientasi ditunjukkan pada Tabel 4.3. Jalur komunikasi antar robot berupa graf lengkap.

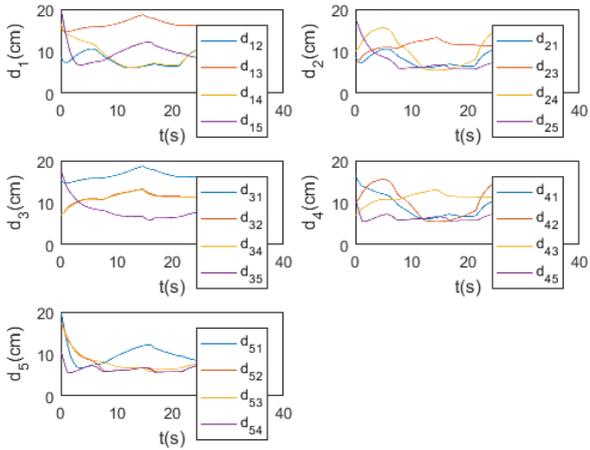
Titik tujuan yang harus dituju oleh robot adalah titik $[120,30]^T$ dan terdapat dua buah rintangan berbentuk lingkaran dengan jari-jari sebesar 25 cm masing-masing dengan titik pusat berada pada titik $[50,0]^T$ dan $[50,60]^T$.

Trayektori hasil simulasi ditunjukkan pada Gambar 4.27. Gambar tersebut menunjukkan robot membentuk formasi, berpencar (tidak mempertahankan formasi) saat melalui celah, kemudian membentuk kembali formasi setelah tidak terdeteksi adanya rintangan. Robot berhasil mencapai titik tujuan dalam waktu 35s.

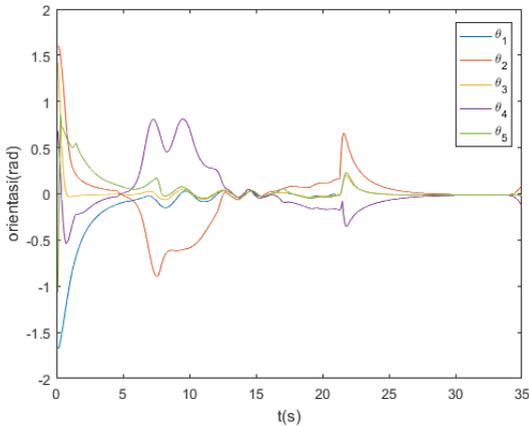
Meskipun melalui celah yang cukup sempit, tabrakan antar robot dapat dihindari seperti ditunjukkan pada Gambar 4.28. Gambar tersebut menunjukkan tidak terjadinya tabrakan antar robot. Jarak antar robot terdekat yang tercatat adalah antara robot 4 dan robot 5 yaitu sebesar 5,4053. Gambar 4.29 menunjukkan orientasi robot 2 dan robot 4 pada saat sekitar $t=5$ sampai $t=28$ saring berlawanan. Hal ini dikarenakan rintangan yang dideteksi menghasilkan gaya potensial yang besarnya sama dan arahnya berlawanan.



Gambar 4.27 Trayektori robot



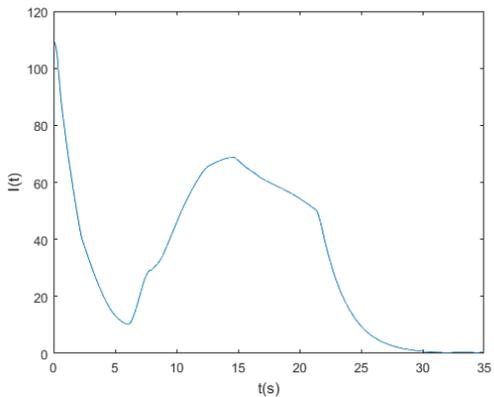
Gambar 4.28 Jarak antar robot



Gambar 4.29 Orientasi robot

Indeks performansi formasi (Gambar 4.30) menunjukkan nilai yang cukup besar saat robot-robot sedang melintasi celah. Namun setelah celah terlewati dan tidak ada rintangan yang terdeteksi, nilai IPF kembali

menuju nol. Robot dalam formasi kemudian bergerak menuju titik tujuan. Nilai rata-rata dari indeks performansi formasi adalah 32,9368.



Gambar 4.30 IPF *tracking* formasi melewati celah

BAB 5

PENUTUP

5.1 Kesimpulan

Pendekatan *virtual structure* dengan memanfaatkan informasi posisi robot dapat digunakan untuk pengaturan formasi. Hasil simulasi menunjukkan robot dalam formasi menuju titik tujuan dalam waktu 80 s. Nilai indeks performansi formasi (IPF) dalam keadaan *steady state* tanpa adanya rintangan terdeteksi adalah 1,001. Nilai rata-rata IPF adalah sebesar 8,2693. Formasi terbentuk sesuai dengan yang diinginkan apabila IPF bernilai nol.

Metode *artificial potential field* (APF) dapat digunakan untuk penghindaran tabrakan antar robot maupun penghindaran rintangan. Dalam simulasi, jarak terdekat antar robot adalah 5,228 cm sedangkan jarak terdekat antara robot dengan rintangan adalah 5,2349 cm. Keduanya masih berada dalam jarak aman bagi robot yaitu sebesar 5 cm.

5.2 Saran

Penelitian selanjutnya dapat memperbaiki kelemahan-kelemahan pada sistem rancangan dengan memilih nilai-nilai gain menggunakan metode *tuning* yang ada sehingga hasil lebih optimal. Penelitian juga dapat dikembangkan dengan mempertimbangkan rintangan bergerak.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] T. Chi, C. Zhang, Y. Song, dan J. Feng, “A strategy of multi-robot formation and obstacle avoidance in unknown environment,” dalam *IEEE International Conference on Information and Automation (ICIA)*, 2016, pp. 1455–1460.
- [2] Y. Q. Chen and Z. Wang, “Formation control: a review and a new consideration,” dalam *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3181–3186.
- [3] M. Mesbahi dan M. Egerstedt, “Graph Theoretic Methods in Multiagent Networks”, Princeton, NJ: Princeton University Press, 2010.
- [4] R. Dhaouadi dan A. A. Hatab, “Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework,” *Advances in Robotics & Automation*, vol. 2, no. 2, Sep. 2013.
- [5] M. A. Lewis and K.-H. Tan, “High Precision Formation Control of Mobile Robots Using Virtual Structures,” *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, Oct. 1997.
- [6] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” dalam *IEEE International Conference on Robotics and Automation Proceedings*, 1985, vol. 2, pp. 500–505.
- [7] C. Yoshioka dan T. Namerikawa, “Formation Control of Nonholonomic Multi-Vehicle Systems based on Virtual Structure,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5149–5154, Jan. 2008.
- [8] T. Miyazaki dan K. Takaba, “Formation control of mobile robots with obstacle avoidance,” dalam *14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014, pp. 121–126.
- [9] A. M. Burohman, A. Widyotriatmo, dan E. Joelianto, “Flocking for nonholonomic robots with obstacle avoidance,” dalam *International Electronics Symposium (IES)*, 2016, pp. 345–350.
- [10] P.I. Corke, Robotics, “Vision & Control: Fundamental Algorithms in MATLAB”. *Second edition*. Springer, 2017.
- [11] A. Sadowska, T. van den Broek, H. Huijberts, N. van de Wouw, D. Kostić, dan H. Nijmeijer, “A virtual structure approach to formation control of unicycle mobile robots using mutual

coupling,” *International Journal of Control*, vol. 84, no. 11, pp. 1886–1902, Nov. 2011.

LAMPIRAN

1. Script M-File Simulasi *Tracking* Formasi dan Penghindaran Rintangan

```
clc;clear;close all

%inisialisasi robot mobil
v1 = Unicycle( [diag([1 1]).^2], 'x0', [35 17 -
pi/2]', 'speedmax', 5);
v2 = Unicycle( [diag([1 1]).^2], 'x0', [15 16
pi/3]', 'speedmax', 5);
v3 = Unicycle( [diag([1 1]).^2], 'x0', [20 28 -
3*pi/2]', 'speedmax', 5);
v4 = Unicycle( [diag([1 1]).^2], 'x0', [11 6
pi/2]', 'speedmax', 5);
v5 = Unicycle( [diag([1 1]).^2], 'x0', [25 8 -
pi/4]', 'speedmax', 5);

v1.x=v1.x0;
v2.x=v2.x0;
v3.x=v3.x0;
v4.x=v4.x0;
v5.x=v5.x0;

%Matriks Laplacian Jaringan Antar Robot
%L=[3 -1 -1 0 -1;-1 3 -1 0 -1;-1 -1 3 -1 0;0 0 -
1 2 -1;-1 -1 0 -1 3];
L=[4 -1 -1 -1 -1;-1 4 -1 -1 -1;-1 -1 4 -1 -1;-1
-1 -1 4 -1;-1 -1 -1 -1 4];

%Jarak antara robot nyata dengan robot virtual
xdi=0.5*[1 1 1 1 1];
ydi=0*[1 1 1 1 1];

%Pola formasi yang diinginkan
uh =1*[-8 -4 0 4 8 4 8 12 8 4];

%Rintangan
```

```

xobs=25+2*[0 -5 0 5 0];
yobs=2*[25 30 35 30 25];
plot(xobs,yobs,'r')

%Posisi Robot Nyata
vx=[v1.x(1) v2.x(1) v3.x(1) v4.x(1) v5.x(1)
v1.x(2) v2.x(2) v3.x(2) v4.x(2) v5.x(2)];
vx3=[v1.x(3) v2.x(3) v3.x(3) v4.x(3) v5.x(3)];

%Posisi robot virtual
for j=1:5
    x(j)=vx(j)+xdi(j)*cos(vx3(j))-
ydi(j)*sin(vx3(j));

x(j+5)=vx(j+5)+xdi(j)*sin(vx3(j))+ydi(j)*cos(vx3
(j));
end
x=x';

%Parameter gain
kvr=1;
kI=0.2*ones(10,1);
kv=2;
co=5;
lo=8;
LL=-blkdiag(L,L);
vri=zeros(1,10)';

%Sensor
n=19;%Jumlah sensor
l=[8;0];%radius (range) sensor
sudut=linspace(-pi,pi,n);%range sudut sensor
sinar=zeros(n,2);
ujung=[repmat(vx(1:5)',1,n*2);repmat(vx(6:10)',1,
n*2)];

Freprii=zeros(n,2);

```

```

Frep_i=zeros(5,2);
jarter=inf*ones(5,1);

for ti=1:1:1000
    theta_i(1)=v1.x(3);
    theta_i(2)=v2.x(3);
    theta_i(3)=v3.x(3);
    theta_i(4)=v4.x(3);
    theta_i(5)=v5.x(3);

    B1=[cos(theta_i(1)) -xdi(1)*sin(theta_i(1))-
        ydi(1)*cos(theta_i(1));sin(theta_i(1))
        xdi(1)*cos(theta_i(1))-ydi(1)*sin(theta_i(1))];
    B2=[cos(theta_i(2)) -xdi(2)*sin(theta_i(2))-
        ydi(2)*cos(theta_i(2));sin(theta_i(2))
        xdi(2)*cos(theta_i(2))-ydi(2)*sin(theta_i(2))];
    B3=[cos(theta_i(3)) -xdi(3)*sin(theta_i(3))-
        ydi(3)*cos(theta_i(3));sin(theta_i(3))
        xdi(3)*cos(theta_i(3))-ydi(3)*sin(theta_i(3))];
    B4=[cos(theta_i(4)) -xdi(4)*sin(theta_i(4))-
        ydi(4)*cos(theta_i(4));sin(theta_i(4))
        xdi(4)*cos(theta_i(4))-ydi(4)*sin(theta_i(4))];
    B5=[cos(theta_i(5)) -xdi(5)*sin(theta_i(5))-
        ydi(5)*cos(theta_i(5));sin(theta_i(5))
        xdi(5)*cos(theta_i(5))-ydi(5)*sin(theta_i(5))];
    for j=1:5
        for i=1:n
            sudutrel=theta_i(j)+sudut(i);
            sinar(i,:)=[cos(sudutrel) -
            sin(sudutrel);sin(sudutrel) cos(sudutrel)]*1;
            ujung(j,(i*2)-1)=vx(j);
            ujung(j+5,(i*2)-1)=vx(j+5);
            ujung(j,i*2)=sinar(i,1)+vx(j);
            ujung(j+5,i*2)=sinar(i,2)+vx(j+5);
        end
    end
end
for j=1:5

```

```

        for i=1:n
            xs1=[ujung(j, (i*2)-1) ujung(j, i*2)];
            ys1=[ujung(j+5, (i*2)-1)
ujung(j+5, i*2)];
            [xi
yi]=polyxpoly(xs1, ys1, xobs, yobs);
            rintc=[xi yi]';
            if numel(rintc)~=0
                rintc1=[rintc(1,1) rintc(2,1)]';
                %jarobs=norm(rintc1-
[vx(j);vx(j+5)]);

jarobs=(1/0.7)*(sqrt(1+0.7*norm(rintc1-
[vx(j);vx(j+5)]))-1);
                sigm=(1/sqrt(1+0.7*norm(rintc1-
[vx(j);vx(j+5)]^2)));
                Frepii(i,:)=1*co*(1/jarobs)-
(1/lo)*(1/jarobs)^2*sigm*-(rintc1-
[vx(j);vx(j+5)]);

%Frepii(i,:)=(20*(co/lo^2))*exp(-
(jarobs)^2/lo^2)*([vx(j);vx(j+5)]-rintc1);
                %Frepii(i,:)=(1/sqrt((jarobs-
lo)^2+1))*(jarobs-lo)*(1/sqrt(1+0.7*jarobs^2))*-
([vx(j);vx(j+5)]-rintc1);
                if jarobs<jarter(j)
                    jarter(j)=jarobs;
                end

            else
                Frepii(i,:)= [0;0];

            end

        end
        Frepj(j,:)=sum(Frepii);
    end
    Frep=[Frepj(:,1)' Frepj(:,2)'];
    for j=1:5
        if norm(Frepj(j,:))~=0

```

```

        ki(j)=(kI(j)/(1+2-jarter(j)));
        ki(j+5)=(kI(j+5)/(1+2-jarter(j)));
    else
        ki(j)=kI(j);
        ki(j+5)=kI(j+5);
    end
end
for i=1:length(L)
    for j=1:length(L)
        if i~=j
            if L(i,j)~=0
                rij=[x(i)-x(j) x(i+5)-
x(j+5)];
                Uijx(i,j)=((norm(rij)-
10)*((norm(rij)-5)^-3))*rij(1);
                Uijy(i,j)=((norm(rij)-
10)*((norm(rij)-5)^-3))*rij(2);
            end
        end
    end
end
for i=1:length(L)
    Uix(i)=sum(Uijx(i,:));
    Uiy(i)=sum(Uijy(i,:));
end
Ui=[Uix Uiy];
dvri=@(t,vri) -kvr*(vri-1*[0 0 0 0 0 1 1 1 1
1]')+ki'.*LL*(x-uh'+kv*vri)-0.1*[Ui]'+Frep';
[t,vri]=ode45(dvri,[0 0.1],vri);

vri=vri(end,:);
u1=B1\([vri(1);vri(6)]);
u2=B2\([vri(2);vri(7)]);
u3=B3\([vri(3);vri(8)]);
u4=B4\([vri(4);vri(9)]);
u5=B5\([vri(5);vri(10)]);
u=[u1 u2 u3 u4 u5];

```

```

odo=v1.update ([u(1,1) u(2,1)]);
odo=v2.update ([u(1,2) u(2,2)]);
odo=v3.update ([u(1,3) u(2,3)]);
odo=v4.update ([u(1,4) u(2,4)]);
odo=v5.update ([u(1,5) u(2,5)]);
[v1.x(1) v1.x(2) v1.x(3);v2.x(1) v2.x(2)
v2.x(3);v3.x(1) v3.x(2) v3.x(3);v4.x(1) v4.x(2)
v4.x(3);v5.x(1) v5.x(2) v5.x(3)];

asli1=v1.plotv ([v1.x(1),v1.x(2),v1.x(3)], 'size',
1.5);

asli2=v2.plotv ([v2.x(1),v2.x(2),v2.x(3)], 'size',
1.5);

asli3=v3.plotv ([v3.x(1),v3.x(2),v3.x(3)], 'size',
1.5);

asli4=v4.plotv ([v4.x(1),v4.x(2),v4.x(3)], 'size',
1.5);

asli5=v5.plotv ([v5.x(1),v5.x(2),v5.x(3)], 'size',
1.5);

hold on
%vir=plot(x(1:5),x(6:10),'ob');

senspl=plot (ujung(1,:),ujung(6:10),'b',ujung(2:
),ujung(7:10),'b',ujung(3:5),ujung(8:10),'b',uju
ng(4:5),ujung(9:10),'b',ujung(5:5),ujung(10:10),'
.b');
vir=plot (vx(1:5),vx(6:10),'b');
xlabel('X(cm)')
ylabel('Y(cm)')
vir(1).MarkerSize=1;
axis([-10 50 0 100])

grid on

```

```

        drawnow()
        vx=[v1.x(1) v2.x(1) v3.x(1) v4.x(1) v5.x(1)
v1.x(2) v2.x(2) v3.x(2) v4.x(2) v5.x(2)];
        vx3=[v1.x(3) v2.x(3) v3.x(3) v4.x(3)
v5.x(3)];
        vx3=mod(vx3,2*pi);
        for j=1:5
            x(j)=vx(j)+xdi(j)*cos(vx3(j))-
ydi(j)*sin(vx3(j));

x(j+5)=vx(j+5)+xdi(j)*sin(vx3(j))+ydi(j)*cos(vx3
(j));
        end
        vri=[(B1*[u(1,1);u(2,1)])
(B2*[u(1,2);u(2,2)]) (B3*[u(1,3);u(2,3)])
(B4*[u(1,4);u(2,4)]) (B5*[u(1,5);u(2,5)])];
        vri=[vri(1,:) vri(2,:)]';
        cobs(ti)=getframe(gcf);
        %x=x+vri;
        if t~=100
            %delete(asli1)
            delete([asli1 asli2 asli3 asli4 asli5])
            delete(senspl)

        end
end
end

```

[Halaman ini sengaja dikosongkan]

2. M-File Unicycle Class [10]

```
a%Unicycle vehicle class
%
% This concrete class models the kinematics of
a differential steer vehicle
% (unicycle model) on a plane. For given
steering and velocity inputs it
% updates the true vehicle state and returns
noise-corrupted odometry
% readings.
%
% Methods::
%   init           initialize vehicle state
%   f              predict next state based on
odometry
%   step           move one time step and return
noisy odometry
%   control        generate the control inputs
for the vehicle
%   update         update the vehicle state
%   run            run for multiple time steps
%   Fx             Jacobian of f wrt x
%   Fv             Jacobian of f wrt odometry
noise
%   gstep          like step() but displays vehicle
%   plot           plot/animate vehicle on current
figure
%   plot_xy        plot the true path of the
vehicle
%   add_driver     attach a driver object to this
vehicle
%   display        display state/parameters in
human readable form
%   char           convert to string
%
% Class methods::
%   plotv          plot/animate a pose on current
figure
%
```

```

    % Properties (read/write)::
    %   x                true vehicle state: x, y,
theta (3x1)
    %   V                odometry covariance (2x2)
    %   odometry        distance moved in the last
interval (2x1)
    %   rdim            dimension of the robot (for
drawing)
    %   L                length of the vehicle
(wheelbase)
    %   alphasim        steering wheel limit
    %   maxspeed        maximum vehicle speed
    %   T                sample interval
    %   verbose         verbosity
    %   x_hist          history of true vehicle
state (Nx3)
    %   driver          reference to the driver
object
    %   x0              initial state, restored on
init()
    %
    % Examples::
    %
    % Odometry covariance (per timestep) is
    %   V = diag([0.02, 0.5*pi/180].^2);
    % Create a vehicle with this noisy odometry
    %   v = Bicycle( 'covar', diag([0.1 0.01].^2
);
    % and display its initial state
    %   v
    % now apply a speed (0.2m/s) and steer angle
(0.1rad) for 1 time step
    %   odo = v.step(0.2, 0.1)
    % where odo is the noisy odometry estimate, and
the new true vehicle state
    %   v
    %
    % We can add a driver object
    %   v.add_driver( RandomPath(10) )

```

```

    % which will move the vehicle within the region
    -10<x<10, -10<y<10 which we
    % can see by
    %     v.run(1000)
    % which shows an animation of the vehicle moving
    for 1000 time steps
    % between randomly selected waypoints.
    %
    % Notes::
    % - Subclasses the MATLAB handle class which
    means that pass by reference semantics
    %     apply.
    %
    % Reference::
    %
    %     Robotics, Vision & Control, Chap 6
    %     Peter Corke,
    %     Springer 2011
    %
    % See also RandomPath, EKF.

    % Copyright (C) 1993-2017, by Peter I. Corke
    %
    % This file is part of The Robotics Toolbox for
    MATLAB (RTB).
    %
    % RTB is free software: you can redistribute it
    and/or modify
    % it under the terms of the GNU Lesser General
    Public License as published by
    % the Free Software Foundation, either version
    3 of the License, or
    % (at your option) any later version.
    %
    % RTB is distributed in the hope that it will
    be useful,

```

```

    % but WITHOUT ANY WARRANTY; without even the
    implied warranty of
    % MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the
    % GNU Lesser General Public License for more
    details.
    %
    % You should have received a copy of the GNU
    Lesser General Public License
    % along with RTB. If not, see
    <http://www.gnu.org/licenses/>.
    %
    % http://www.petercorke.com

```

```

classdef Unicycle < Vehicle

```

```

    properties

```

```

        % state

```

```

        W                % wheel separation

```

```

        accelmax

```

```

        vprev

```

```

        steerprev

```

```

    end

```

```

    methods

```

```

        function veh = Unicycle(varargin)

```

```

        %Unicycle.Unicycle Unicycle object

```

```

constructor

```

```

    %

```

```

    % V = Unicycle(VA, OPTIONS) creates a
    Unicycle object with actual odometry

```

```

    % covariance VA (2x2) matrix
    corresponding to the odometry vector [dx dtheta].

```

```

    %

```

```

    % Options::

```

```

    % 'W',W                Wheel separation [m]

```

```

    (default 1)

```

```

5m/s)          % 'vmax',S          Maximum speed (default
               % 'x0',x0          Initial state (default
(0,0,0) )      % 'dt',T           Time interval
               % 'rdim',R        Robot size as fraction
of plot window (default 0.2)
               % 'verbose'       Be verbose
               %
               % Notes::
               % - Subclasses the MATLAB handle class
which means that pass by reference semantics
               % apply.

               veh = veh@Vehicle(varargin{:});

               veh.x = zeros(3,1);

               opt.W = 1;
               opt.accelmax = Inf;

               veh = tb_optparse(opt, veh.options,
veh);
               veh.vprev = 0;
end

function xnext = f(veh, x, odo, w)
    %Unicycle.f Predict next state
based on odometry
    %
    % XN = V.f(X, ODO) is the predicted
next state XN (1x3) based on current
    % state X (1x3) and odometry ODO
(1x2) = [distance, heading_change].
    %
    % XN = V.f(X, ODO, W) as above but
with odometry noise W.

```

```

%
% Notes::
% - Supports vectorized operation
where X and XN (Nx3).
    if nargin < 4
        w = [0 0];
    end

    dd = odo(1) + w(1); dth = odo(2) +
w(2);

    % straightforward code:
    % thp = x(3) + dth;
    % xnext = zeros(1,3);
    % xnext(1) = x(1) + (dd +
w(1))*cos(thp);
    % xnext(2) = x(2) + (dd +
w(1))*sin(thp);
    % xnext(3) = x(3) + dth + w(2);
    %
    % vectorized code:

    thp = x(:,3) + dth;
    xnext = x + [dd*cos(thp)
dd*sin(thp) ones(size(x,1),1)*dth];
end

function dx = deriv(veh, t, x, u)
% to be called from a continuous
time integrator such as ode45 or Simulink

% implement acceleration limit if
required
if ~isinf(veh.accelmax)
    if (u(1) - veh.vprev)/veh.dt >
veh.accelmax
        u(1) = veh.vprev +
veh.accelmax * veh.dt;
    end
end

```

```

elseif (u(1) > veh.vprev)/veh.dt < -veh.accelmax -
    u(1) = veh.vprev -
veh.accelmax * veh.dt;
end
veh.vprev = u(1);
end

% implement speed and steer angle
limits
u(1) = min(veh.speedmax, max(u(1),
-veh.speedmax));

% compute the derivative
dx = zeros(3,1);
dx(1) = u(1)*cos(x(3));
dx(2) = u(1)*sin(x(3));
dx(3) = u(2)/veh.W;
end

function odo = update(veh, u)
%Unicycle.update Update the vehicle
state
%
% ODO = V.update(U) is the true
odometry value for
% motion with U=[speed,steer].
%
% Notes::
% - Appends new state to state
history property x_hist.
% - Odometry is also saved as
property odometry.

% update the state
dx = veh.dt * veh.deriv([], veh.x,
u);
veh.x = veh.x + dx;

```

```

        % compute and save the odometry
        odo = [ norm(dx(1:2)) dx(3) ];
        veh.odometry = odo;

        veh.x_hist = [veh.x_hist; veh.x'];
% maintain history
end

function J = Fx(veh, x, odo)
%Unicycle.Fx Jacobian df/dx
%
% J = V.Fx(X, ODO) is the Jacobian df/dx
(3x3) at the state X, for
% odometry input ODO (1x2) = [distance,
heading_change].
%
% See also Unicycle.f, Vehicle.Fv.
dd = odo(1); dth = odo(2);
thp = x(3) + dth;

J = [
    1    0   -dd*sin(thp)
    0    1    dd*cos(thp)
    0    0    1
];
end

function J = Fv(veh, x, odo)
%Unicycle.Fv Jacobian df/dv
%
% J = V.Fv(X, ODO) is the Jacobian
df/dv (3x2) at the state X, for
% odometry input ODO (1x2) =
[distance, heading_change].
%
% See also Unicycle.F, Vehicle.Fx.
dd = odo(1); dth = odo(2);
thp = x(3);

```

```

        J = [
            cos(thp)    0 %-dd*sin(thp)
            sin(thp)    0 %dd*cos(thp)
            0            1
        ];
    end

    function s = char(veh)
    %Unicycle.char Convert to a string
    %
    % s = V.char() is a string showing
vehicle parameters and state in
    % a compact human readable format.
    %
    % See also Unicycle.display.

        ss = char@Vehicle(veh);

        s = 'Unicycle object';
        s = char(s, sprintf('      W=%g',
veh.W));
        s = char(s, ss);
    end
end % method

end % classdef

```

[Halaman ini sengaja dikosongkan]

3. M-File Indeks Performansi Formasi

```
fref=zeros(5,5);
fact=zeros(5,5);
posisi=[v1.x_hist(:,1:2)      v2.x_hist(:,1:2)
v3.x_hist(:,1:2)             v4.x_hist(:,1:2)
v5.x_hist(:,1:2)];
banyak=size(v1.x_hist,1);
indeks=zeros(banyak,1);

for k=1:banyak
    for i=1:5
        for j=1:5
            if i~=j
                fref(i,j)=norm([uh(i)
uh(i+5)]-[uh(j) uh(j+5)]);
                fact(i,j)=norm([posisi(k,2*i-
1)           posisi(k,2*i)]-[posisi(k,2*j-1)
posisi(k,2*j)]);
                err(i,j)=abs(fact(i,j)-
fref(i,j));
            end
        end
    end
    indeks(k)=sum(sum(err));
end
t=1:banyak;
plot(t/10,indeks)
xlabel('t (s)')
ylabel('I (t)')
```

[Halaman ini sengaja dikosongkan]

RIWAYAT PENULIS



Anang Apriyanto, lahir di Wonosobo pada 23 April 1996. Putra terakhir dari pasangan Bapak Sudiono dan Ibu Turaseh. Setelah menempuh pendidikan formal di SD N 3 Banjarnegoro, SMP N 2 Wonosobo dan SMA N 1 Gadingrejo, penulis melanjutkan studi S1 di Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan mengambil Jurusan Teknik Elektro, Bidang Studi Teknik Sistem Pengaturan. Penulis merupakan anggota Laboratorium Sistem dan Sibernetika. Pada tanggal 8 Januari 2018, penulis mengikuti ujian Tugas Akhir untuk memperoleh gelar Sarjana Teknik. Penulis dapat dihubungi melalui email: anangapriyanto0@gmail.com