

Kendali Formasi Mobile Robot Berdasarkan Jarak Menggunakan Algoritma *Cosinus*

ANGGORO DWI NUR ROHMAN¹, M. AZIZ MUSLIM², BAMBANG SISWOJO³

¹²³Program Studi Magister, Teknik Elektro, Universitas Brawijaya, Indonesia
Email : anggoro_dwi@student.ub.ac.id

Received 30 November 201x | *Revised* 30 Desember 201x | *Accepted* 30 Januari 201x

ABSTRAK

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

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

ABSTRACT

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

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

1. PENDAHULUAN

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

Penjabaran oleh Kwang-Kyo (**Oh et al., 2015**) bahwa dari berbagai pengembangan kendali formasi dapat diambil garis besar menjadi tiga bagian, yaitu berdasarkan posisi, perpindahan, dan jarak. Ketiga bagian tersebut tertuju pada jawaban dari pertanyaan, "variabel apa yang digunakan sebagai sensor" dan "variabel apa yang aktif dikendalikan oleh sistem multi-robot untuk mencapai formasi yang diinginkan". Kendali formasi berdasarkan posisi adalah metode kendali formasi dimana robot diharuskan memiliki kemampuan untuk mengetahui kordinatnya sendiri berdasarkan koordinat global. Kerena itu kendali berdasarkan posisi membutuhkan sensor posisi seperti GPS. Kendali formasi berdasarkan perpindahan adalah kendali formasi dimana setiap robot tidak mengetahui koordinatnya berdasarkan koordinat global dikarenakan variabel yang digunakan sebagai sensor adalah kecepatan terhadap tetangganya. Metode berdasarkan posisi, robot membutuhkan kemampuan penyesuaian orientasi koordinat pada koordinat global sehingga setiap robot dibutuhkan sensor kompas untuk menyearahkannya. Formasi berdasarkan jarak adalah kendali formasi dimana variabel yang dikendalikan adalah variabel jarak antar robot yang terhubung sehingga koordinat yang digunakan tidak mengacu pada koordinat global. Sehingga pada penerapannya, formasi berdasarkan jarak menggunakan sensor yang lebih sedikit dibanding dengan posisi dan perpindahan. Namun salah satu permasalahan pada kendali formasi berdasarkan jarak adalah penerapan model yang lebih nyata. Pengembangan formasi berdasarkan jarak telah dikembangkan menggunakan teori graph dengan model sederhana pada *single* dan *double integrator* oleh Kwang-Kyo (**Oh & Ahn, 2014**), menggunakan informasi jarak untuk mengendalikan model sederhana *double integrator* oleh Xiaoyu (**Cai & de Queiroz, 2014**), menggunakan konsensur antara robot untuk mencapai bentuk formasi pada model *single integrator* sederhana oleh Deghat (**Deghat et al., 2016**), mengendalikan bentuk formasi menggunakan *adaptive control* untuk mengestimasi kecepatan tetangga dari model robot yang sederhana oleh Sung-Mo (**Kang et al., 2014**), menggabungkan kendali formasi berdasarkan jarak dan perpindahan untuk mengendalikan model robot yang sederhana oleh Myoung-Chul (**Park & Ahn, 2015**), dan menggunakan kendali *Proportional-Integral* (PI) untuk mengendalikan jarak setiap model robot yang sederhana oleh Rozenheck (**Rozenheck et al., 2015**).

Kendali PI pada penelitian oleh Rozenheck (**Rozenheck et al., 2015**) tidak dapat langsung diterapkan menggunakan sensor jarak karena kendali tersebut mengambil informasi jarak menggunakan selisih dari koordinat global kartesian setiap robot. Sedangkan dalam praktiknya robot hanya bisa mengukur jarak dan tidak mengetahui koordinat dari robot tetangganya. Sensor jarak sudah umum untuk digunakan pada penelitian dibidang lokalisasi seperti sensor *Ultra-wide band* (UWB) penelitian oleh Kexing (**Guo et al., 2020**) yang menggantikan sensor berbasis *vision* untuk mendapatkan koordinat dari beberapa robot, penelitian oleh Qiang (**Qiang et al., 2017b**) yang membutuhkan dua sensor UWB yang terpasang statis yang digunakan untuk mengetahui koordinat sekelompok robot untuk dikendalikan secara terpusat dan pengembangan oleh Qiang (**Qiang et al., 2017a**) dengan memasang dua sensor UWB

yang terpasang statis tersebut dipasang pada salah satu dari sekelompok robot lalu mendistribusikan koordinat pada robot tetangganya. Lokalisasi menggunakan dua sensor jarak tersebut memanfaatkan rumus segitiga untuk mendapatkan koordinat suatu robot tetangga. Penelitian ini akan mengadopsi rumus segitiga tersebut dan menggantikan salah satu dari dua sensor dengan algoritma *cosinus* sehingga tidak mengharuskan salah satu robot memiliki dua sensor. Penelitian oleh Rozenheck (**Rozenheck et al., 2015**) mengembangkan kendali formasi menggunakan model *holonomic* sederhana dimana robot bergerak ke suatu arah tidak bergantung dari arah robot pertamakali dan tidak mempertimbangkan parameter fisik pada modelnya. Percobaan akan menggunakan tiga model robot *holonomic* dengan harapan menjadi langkah awal untuk mengembangkan kendali formasi berdasarkan jarak menggunakan model robot yang lebih nyata..

2. METODE

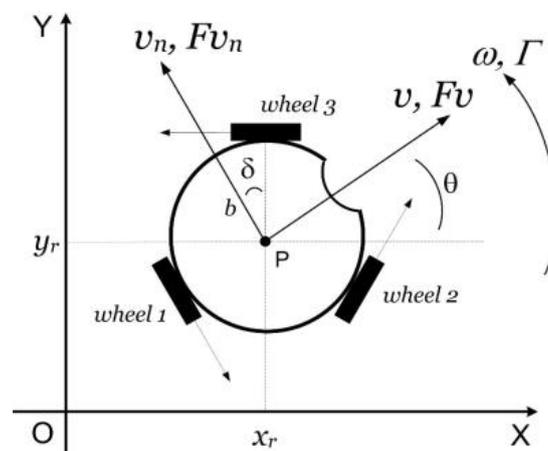
2.1 Model Robot

Robot menggunakan tiga buah motor yang dihubungkan pada *omniwheel* sehingga robot dapat bergerak kesegala arah (Gambar 1). Berikut adalah model dari robot *holonomic* dalam bentuk *state-space*.

$$\dot{x}(t) = A_r x(t) + B_r u(t) + K_r \text{sgn}(x(t)) \quad (1)$$

$$y(t) = Cx(t) \quad (2)$$

Dimana vector $u(t) = [u_1(t) \ u_2(t) \ u_3(t)]^T$ adalah vector tegangan motor untuk memutar roda $\vartheta = \{1,2,3\}$ yang memiliki batasan tegangan $-6 \leq u_\vartheta \leq 6$ volt. Vektor $y(t) = x(t) = [v(t) \ v_n(t) \ w(t)]^T$ adalah kecepatan robot yang akan diperoleh dari sensor percepatan, dimana v adalah kecepatan robot bergerak maju/mundur atau koordinat x berdasarkan kerangka robot, v_n adalah kecepatan robot untuk menyamping kiri/kanan atau koordinat sumbu y berdasarkan kerangka robot, dan $w(t)$ adalah kecepatan rotasi dari kerangka robot. Maktriks $A_r \in R^{3 \times 3}$ dan $B_r \in R^{3 \times 3}$ adalah parameter fisik robot yang diperoleh dari persamaan fisika. Maktriks $K_r \in R^{3 \times 3}$ adalah parameter *friction* dari robot yang diestimasi dari hasil identifikasi sistem dengan cara percobaan. Penelitian ini menggunakan parameter fisik dan hasil estimasi *friction* yang dibahas secara jelas didalam paper penelitian oleh Mariane (**Correia et al., 2012**).



Gambar 1. Geometri Rangka Robot (Correia et al., 2012)

2.2 Kendali Robot Holonomic

Kendali robot holonomic akan dibagi menjadi dua mode. Perbedaan kedua mode tersebut adalah *setpoint* kendalinya, dimana mode satu akan memiliki *setpoint* kecepatan robot sedangkan mode dua memiliki *setpoint* koordinat kerangka robot. Penjelasan lebih lengkap kegunaan dari kedua mode tersebut akan dijelaskan pada bab metode algoritma *cosinus*.

Mode Satu. Kendali robot mode satu bertujuan untuk robot mencapai kecepatan yang diinginkan. Untuk mencapai tujuan tersebut akan menggunakan metode *state-feedback* pada persamaan kendalinya.

$$u_{c1}(t) = -K_1^c x(t) + N_1^c r_1^c \quad (3)$$

$$N_1^c = -[C(A_r - B_r K_1^c)^{-1} B_r]^{-1} \quad (4)$$

Dimana vector $r_1^c = [v^* \ v_n^* \ w^*]$ adalah *setpoint* kendali mode satu, $K_1^c \in R^{3 \times 3}$ adalah konstanta yang diperoleh dari optimasi persamaan *Riccati* terhadap matriks A_r dan B_r pada persamaan (1), dan $N_1^c \in R^{3 \times 3}$ adalah konstanta yang diperoleh dari *inverse state-space* pada keadaan *steady state* yang menghasilkan persamaan (4). Dengan menggabungkan persamaan (1) dan persamaan (3) akan mendapatkan persamaan *state-space* robot yang baru.

$$\dot{x}(t) = A_c x(t) + B_c r_1^c + K_r \text{sgn}(x(t)) \quad (5)$$

$$A_c = (A_r - B_r K_1^c) \quad (6)$$

$$B_c = B_r N_1^c \quad (7)$$

Mode Dua. Kendali robot mode dua bertujuan untuk robot mencapai titik koordinat tertentu berdasarkan koordinat kerangkanya. Untuk mencapai tujuan tersebut akan memodifikasi persamaan (1) dengan menambah koordinat *frame* robot pada *state*-nya. *State* robot menjadi $x_{c2}(t) = [x_r(t) \ y_r(t) \ \theta_r(t) \ v(t) \ v_n(t) \ w(t)]^T$. Berikut adalah *state-space* robot mode dua.

$$\dot{x}_{c2}(t) = A_{c2} x_{c2}(t) + B_{c2} u_{c2}(t) + K_{c2}(x_{c2}(t)) \quad (8)$$

$$y_{c2}(t) = C_{c2} x_{c2}(t) \quad (9)$$

$$A_{c2} = \begin{bmatrix} 0 & I \\ 0 & A_r \end{bmatrix} \in R^{6 \times 6} \quad (10)$$

$$B_{c2} = \begin{bmatrix} 0 & B_r \end{bmatrix} \in R^{6 \times 3} \quad (11)$$

$$K_{c2}(x) = \begin{bmatrix} 0 & 0 \\ 0 & K_r \text{sgn}(x) \end{bmatrix} \in R^{6 \times 3} \quad (12)$$

Untuk menentukan mendapatkan $u_{c2}(t)$ sebagai persamaan kendali pada persamaan (8), akan dilakukan metode yang sama seperti mode satu.

$$u_{c2}(t) = -K_2^c x_{c2}(t) + N_2^c r_2^c \quad (13)$$

Dimana $r_2^c = [x_r^* \ y_r^* \ \theta_r^* \ v^* \ v_n^* \ w^*]^T$ adalah *setpoint* kendali mode dua, $K_2^c \in R^{3 \times 6}$ adalah konstanta yang diperoleh dari optimasi persamaan *Riccati* terhadap matriks A_{c2} dan B_{c2}

menggunakan persamaan (8) dan $N_2^c \in R^{3 \times 6}$ adalah konstanta yang diperoleh dari *inverse state-space* pada keadaan *steady state*.

2.3 Kendali Formasi Berdasarkan Jarak

Didefinisi sebuah *undirect graph* $G = (\mathcal{V}, \mathcal{E})$ terdiri dari himpunan simpul \mathcal{V} dan himpunan sisi $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Didefinisi $n \triangleq |\mathcal{V}|$ sebagai jumlah dari node dan $m \triangleq |\mathcal{E}|$ sebagai jumlah dari sisinya. Didefinisi $p = [x_1^T \dots x_n^T]^T \in R^{3n}$, dimana $x_i \in R^3$ koordinat robot i dan $x_i \neq x_j$ untuk semua $i \neq j$. Dinotasikan vektor posisi relatif $e_k \triangleq x_j - x_i$ dan semua vector sisi $e = [e_1^T \dots e_m^T] \in R^{3m}$. Didefinisikan fungsi *Jacobian* sisi dari penelitian oleh Rozenheck (**Rozenheck et al., 2015**)

$$R \triangleq \text{diag}(e_i^T)(E^T \otimes I_2) \in R^{m \times 3n} \quad (14)$$

Dimana $\text{diag}(A_i)$ adalah $\text{diag}(A_i) \triangleq \text{blkdiag}\{A_1, \dots, A_n\} \in R^{np \times nq}$, I_2 adalah matrik identitas 2×2 dan $E \in R^{m \times 3n}$ adalah matrik *incidence* $\{0, \pm 1\}$ yang baris dan kolom matrik nya mengindikasikan simpul dan sisinya. Kendali formasi mengadopsi dari penelitian sebelumnya (**Oh & Ahn, 2014**) dimana persamaan potensial didefinisikan sebagai berikut.

$$\Phi(e) = \frac{1}{2} \sum_{i \in \mathcal{V}} \|v_i\|^2 + \frac{1}{2} \sum_{k=1}^m \left(\|e_k\|^2 - d_k^2 \right)^2 \quad (15)$$

Lalu, diperoleh *Gradient negative* dari persamaan (15) untuk digunakan sebagai masukan dari persamaan (5) dan diberikan konstanta *proportional*.

Pada peneliatn ini mengusulkan untuk mengubah model sederhana dari penelitian sebelumnya oleh Rozenheck (**Rozenheck et al., 2015**) menjadi model robot *holonomic pada* persamaan (5). Berikut adalah hasil persamaannya yang akan dijalankan secara simulasi.

$$\dot{p} = A_f p(t) + B_f \frac{\partial \Phi(e)}{\partial v} \quad (16)$$

$$= A_f p(t) + B_f v(t)$$

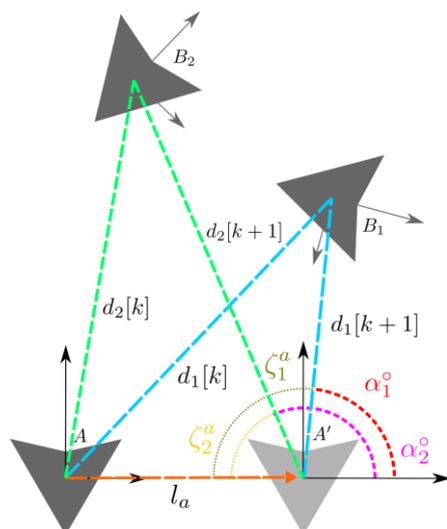
$$\dot{v} = -C \left(\frac{\partial \Phi(e)}{\partial v} + \frac{\partial \Phi(e)}{\partial p} \right) \quad (17)$$

$$= -k_{p1} v(t) + R^T(p(t)) k_{p2} (R(p(t))p(t) - d)$$

Dimana $A_f \in R^{3n \times 3n}$ dan $B_f \in R^{3n \times 3n}$ adalah matrix diagonal dari A_c dan B_c .

2.4 Algoritma Cosinus

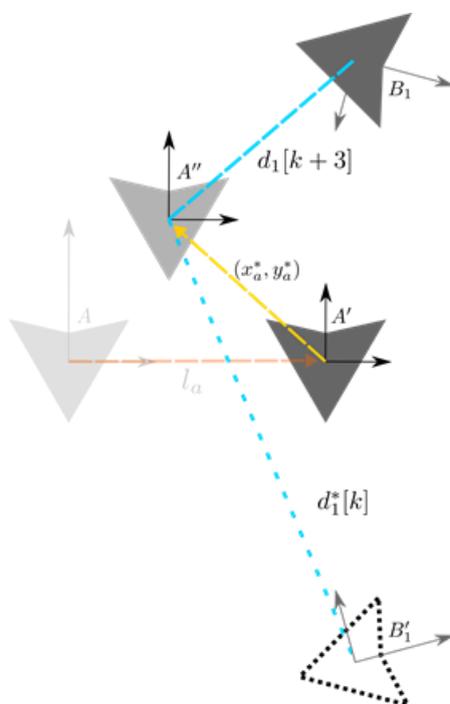
Telah didefinisi pada pendahuluan bahwa menggunakan informasi jarak saja robot tidak mengetahui koordinat tetangganya, sedangkan pada persamaan (16) dan (17), sebagai kendali formasi, membutuhkan *state* koordinat pertamakali. Pada penelitian ini diusulkan sebuah algoritma *cosinus* untuk menemukan *state* ketika kondisi awal. Algoritma *cosinus* membutuhkan robot untuk mengerjakan dua Langkah yang akan dijelaskan sebagai berikut.



Gambar 2. Strategi Algoritma *Cosinus* Pencarian Koordinat Tetangga Langkah Pertama

Dinotasikan \mathcal{N}_A adalah himpunan tetangga dari robot A . Dinotasikan $B_i \in \mathcal{N}_A$ adalah robot i sebagai tetangga dari robot A . Dinotasikan $d_i[k]$ adalah informasi jarak antara robot A ke robot B_i .

Langkah pertama. Dapat diperhatikan pada Gambar 2 di mana robot A menyimpan jarak $d_i[k]$ dan menggunakan kendali mode dua pada persamaan (8) untuk berpindah sepanjang l_a atau ke koordinat $A' = (0, l_a)$ sehingga akan mendapatkan jarak $d_i[k+1]$. Dari perbedaan d_i sesudah $(k+1)$ dan sebelum (k) berpindah tersebut akan digunakan untuk menentukan sudut α_i^o menggunakan rumus segitiga *cosinus*. Berikut adalah persamaan α_i^o .



Gambar 3. Strategi Algoritma *Cosinus* Pencarian Koordinat Tetangga Langkah Kedua

$$\zeta_i^a = \cos^{-1} \left(\frac{l_a^2 + d_i[k+1]^2 - d_i[k]^2}{2d_i[k+1]l_a} \right) \quad (18)$$

$$\alpha_i^\circ = 180^\circ \pm \zeta_i^a \quad (19)$$

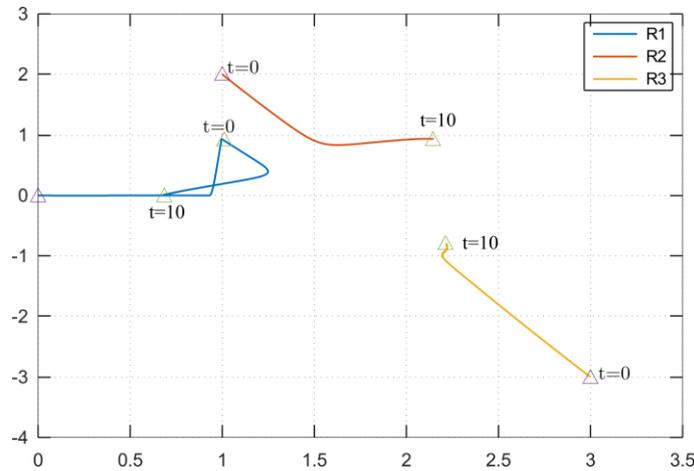
Variable α_i° dan $d_i[k+1]$ adalah nilai dari koordinat polar dari setiap robot tetangga A . Diubah menjadi koordinat kartesian untuk dapat dimasukkan dalam *state* kendali formasi.

$$x_{B_i}^A = \begin{bmatrix} x_{B_i} = d_i[k+1] \cos \alpha_i^\circ \\ y_{B_i} = d_i[k+1] \sin \alpha_i^\circ \end{bmatrix} \quad (20)$$

Langkah kedua. Hasil koordinat yang dihasilkan pada persamaan (20) akan menghasilkan bias dikarenakan persamaan (19) tidak mengetahui letak kuadran sudut nya. Sudut ζ_i^a adalah sudut segitiga (Gambar 3) $\angle AA'B_1$ atau $\angle AA'B_2$ sehingga dimungkinkan koordinat yang dihasilkan persamaan (20) bisa berada pada kuadran 1 atau kuadran 4 (perhatikan robot B_1 pada (Gambar 3)). Oleh karena itu pada persamaan (19) terdapat operasi \pm dimana operasi tersebut akan dilakukan berdasarkan letak kuadran B_i .

$$\alpha_i^\circ = \begin{cases} 180^\circ - \zeta_i^a, & \text{Robot berada pada kuadran 1 dan 2} \\ 180^\circ + \zeta_i^a, & \text{Robot berada pada kuadran 3 dan 4} \end{cases} \quad (21)$$

Langkah kedua ini bertujuan untuk menentukan kejadian pada persamaan (21) dimana robot diharuskan untuk berpindah ke koordinat $A'' = (x_a^*, y_a^*)$ (Gambar 3/Gambar 4). Sebelum robot berpindah, kondisi robot telah mendapatkan koordinat dari langkah pertama. Koordinat tersebut akan diubah menjadi jarak dan akan dibandingkan jarak tersebut dengan informasi jarak dari sensor setelah berpindah ke A'' . Apabila terdapat perbedaan maka kejadian pada persamaan (20) diubah ke kejadian selanjutnya dan mengkoreksi koordinat sebelumnya.

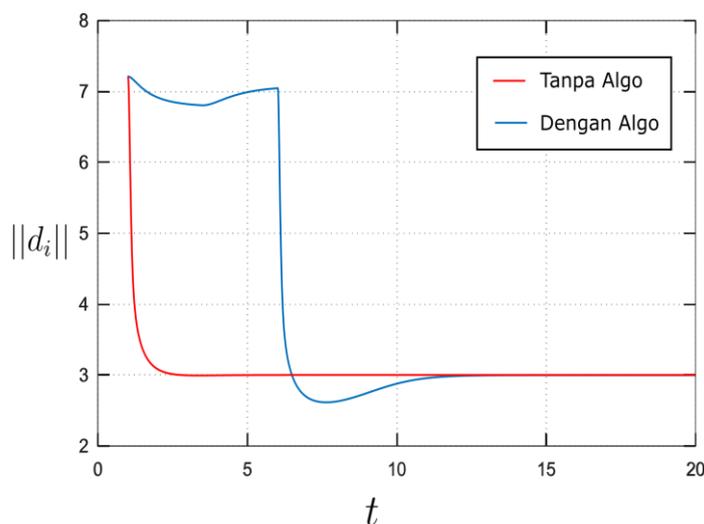


Gambar 4. Grafik Pergerakan Robot Menggunakan Algoritma *Cosinus* Pada Kendali Formasi

3. HASIL DAN PEMBAHASAN

Kendali formasi berdasarkan jarak akan dijalankan secara simulasi menggunakan *MATLAB / GNU Octave* dan algoritma *cosinus* akan dijalankan pertama kali untuk mendapatkan *state* yang akan digunakan di persamaan (16) dan (17). Simulasi akan menggunakan 3 robot dengan himpunan simpul $\mathcal{V} = \{R_1, R_2, R_3\}$ dan himpunan sisi $\mathcal{E} = \{(R_1, R_2), (R_3, R_2), (R_3, R_1)\}$ sehingga variable jarak $d_1 = \|x_1 - x_2\|$, $d_2 = \|x_3 - x_2\|$ dan $d_3 = \|x_3 - x_1\|$. Gambar 4 adalah pergerakan robot terhadap koordinat global robot R_1 dengan kendali formasi algoritma

cosinus. dapat diperhatikan robot R_1 menjalankan algoritma *cosinus* langkah pertama dimana robot berpindah sepanjang $l_a = 1$ sehingga *setpoint* kendali persamaan (13) adalah $r_2^c = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$. Setelah *setpoint* tercapai, robot R_1 mendapatkan jarak yang dibutuhkan untuk menyelesaikan persamaan (19) dan (20) lalu dilanjutkan ke langkah ke dua dengan berpindah ke koordinat (1,1) sehingga *setpoint* kendali persamaan (13) adalah $r_2^c = [1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$. Setelah *setpoint* tercapai maka dilakukan pengecekan kejadian persamaan (21) dengan membandingkan jarak dari sensor dengan jarak dari koordinat yang dihasilkan dari persamaan (20). Robot R_3 berada pada kuadran 4 maka kejadian yang digunakan adalah $\alpha_i^o = 180^\circ + \zeta_i^a$ dan robot R_2 berada pada kuadran 1 maka kejadian yang digunakan adalah $\alpha_i^o = 180^\circ - \zeta_i^a$. Setelah koordinat ditemukan, maka kendali robot mulai berpindah menggunakan kendali formasi persamaan (16) dan (17) dengan kondisi koordinat awal *state* berada pada $t = 0$ dan akhir dari formasi pada $t=10$ pada Gambar 4.



Gambar 5. Grafik *Response* Jarak Terhadap Waktu Pada Robot Dengan Kendali Formasi.

Seperti yang telah dijelaskan pada sub-bab 2.4 Algoritma *Cosinus* bahwa kendali formasi pada persamaan (16) dan (17) mengharuskan *state* dalam bentuk koordinat dimana ketika kondisi awal robot tidak mengetahuinya maka perlu diketahui seberapa lama robot melakukan algoritmanya. Gambar 5 adalah grafik perbandingan kendali formasi persamaan (16) dan (17) dengan algoritma *cosinus* dan tanpa algoritma *cosinus* dengan asumsi *state* pertamakali telah diketahui. Hasil jarak $||d_i||$ pada kendali formasi tanpa algoritma *cosinus* menunjukkan bahwa setiap robot mencapai jarak yang sama dalam waktu kurang lebih 4 detik. Sedangkan pada kendali formasi dengan algoritma *cosinus* terdapat tambahan waktu 6 detik. Tambahan waktu tersebut digunakan untuk menjalankan algoritmanya.

Tabel 1. *Settling Time* Dengan Konstanta K_p Yang Berbeda

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

Dilakukan percobaan *settling time* dengan beberapa koordinat yang mewakili kombinasi antara dua robot dengan kejadian pada persamaan (21). Serta diberikan nilai K_{p1} dan K_{p2} pada persamaan (17) yang berbeda juga. Dapat diperhatikan pada Tabel 1 selisih dari penerapan algoritma dan tidak memiliki rata-rata 6.5 detik. Artinya bahwa algoritma membutuhkan waktu 4-9 detik untuk menemukan koordinat tetangganya.

4. KESIMPULAN

Pada paper ini diusulkan penggunaan algoritma *cosinus* untuk kendali formasi berdasarkan jarak pada tiga model robot *holonomic*. Dari hasil percobaan penerapan algoritma *cosinus* pada kendali formasi berdasarkan jarak dengan *controller proportional* diperoleh hasil yang cukup baik, dimana algoritma *cosinus* dapat menemukan koordinat tetangganya dalam rata-rata waktu 6.5 detik.

DAFTAR RUJUKAN

- Cai, X., & de Queiroz, M. (2014). Multi-agent formation maneuvering and target interception with double-integrator model. *2014 American Control Conference*, 287–292. <https://doi.org/10.1109/ACC.2014.6858603>
- Correia, M. D., Gustavo, A., & Conceição, S. (2012). Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models. *IFAC Proceedings Volumes*, 45(22), 7–12. <https://doi.org/https://doi.org/10.3182/20120905-3-HR-2030.00002>
- Deghat, M., Anderson, B. D. O., & Lin, Z. (2016). Combined Flocking and Distance-Based Shape Control of Multi-Agent Formations. *IEEE Transactions on Automatic Control*, 61(7), 1824–1837. <https://doi.org/10.1109/TAC.2015.2480217>
- Ferik, S. el, Nasir, M. T., & Baroudi, U. (2016). A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space. *Applied Soft Computing*, 44, 117–127. <https://doi.org/https://doi.org/10.1016/j.asoc.2016.03.018>
- Guanghua, W., Deyi, L., Wenyan, G., & Peng, J. (2013). *Study on Formation Control of Multi-Robot Systems*. 1335–1339. <https://doi.org/10.1109/ISDEA.2012.316>
- Guo, K., Li, X., & Xie, L. (2020). Ultra-Wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control. *IEEE Transactions on Cybernetics*, 50(6), 2590–2603. <https://doi.org/10.1109/TCYB.2019.2905570>
- Kang, S.-M., Park, M.-C., Lee, B.-H., & Ahn, H.-S. (2014). Distance-based formation control with a single moving leader. *2014 American Control Conference*, 305–310. <https://doi.org/10.1109/ACC.2014.6858587>
- Li, X., Li, B., Zhang, P., Zhang, J., & Zhang, S. (2015). Large-Scale Unmanned Aerial Vehicle Formation Maintenance Algorithm Based on Virtual Structure. *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, 1, 417–421. <https://doi.org/10.1109/ISCID.2015.5>
- Oh, K.-K., & Ahn, H.-S. (2014). Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space. *International Journal of Robust and Nonlinear Control*, 24(12), 1809–1820. <https://doi.org/10.1002/rnc.2967>
- Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, 53, 424–440. <https://doi.org/https://doi.org/10.1016/j.automatica.2014.10.022>
- Park, M.-C., & Ahn, H.-S. (2015). Distance-based control of formations with orientation control. *2015 54th IEEE Conference on Decision and Control (CDC)*, 2199–2204. <https://doi.org/10.1109/CDC.2015.7402533>

- Qiang, L., Heng, W., Huican, L., & Ying, Z. (2017a). Design and implementation of multi robot research platform based on UWB. *2017 29th Chinese Control And Decision Conference (CCDC)*, 7246–7251. <https://doi.org/10.1109/CCDC.2017.7978492>
- Qiang, L., Heng, W., Huican, L., & Ying, Z. (2017b). Design and implementation of multi robot research platform based on UWB. *2017 29th Chinese Control And Decision Conference (CCDC)*, 7246–7251. <https://doi.org/10.1109/CCDC.2017.7978492>
- Rozenheck, O., Zhao, S., & Zelazo, D. (2015). A proportional-integral controller for distance-based formation tracking. *2015 European Control Conference (ECC)*, 1693–1698. <https://doi.org/10.1109/ECC.2015.7330781>
- Wang, X., Yan, Z., & Wang, J. (2014). Model predictive control of multi-robot formation based on the simplified dual neural network. *2014 International Joint Conference on Neural Networks (IJCNN)*, 3161–3166. <https://doi.org/10.1109/IJCNN.2014.6889491>