

Kendali Formasi Mobile Robot Berdasarkan Jarak Menggunakan Algoritma *Cosinus*

Anggoro Dwi Nur Rohman¹, M. Aziz Muslim², dan Bambang Siswojo³

Program Studi Magister Teknik Elektro, Universitas Brawijaya
Jl. MT Haryono 167, Malang, 65145, Indonesia

anggoro_dwi@student.ub.ac.id¹, muh_aziz@ub.ac.id², bsiswoyo@ub.ac.id³

Abstract— Controlling a group of robots poses challenges in terms of movement coordination. One of the research topics to control a group of robots is formation control. Formation control has the aim of controlling a group of robots in reaching a certain formation and can maintain that formation when moving towards the desired direction. One method of formation control uses a proximity sensor to reach and maintain a formation. The development of distance-based formation control has been mostly carried out using a simple model so that further development is needed in its application to a more real model. There is a problem when distance-based formation control is applied to mobile robots, namely the robot cannot know the coordinates of its neighbors because the distance has no direction. So the cosine algorithm is developed to find the coordinates of the neighbors as a solution to the initial conditions for distance-based formation control. Algorithm development as well as using a holonomic robot model as a first step in the development of using a more real robot. From the results of the experiment with the application of the cosine algorithm and distance-based formation control using the holonomic mobile robot model, the robot can find the coordinates of its neighbors in a fairly short time and the settling time achieved is quite good. The average time it takes for the cosine algorithm to get the coordinates of its neighbors is 6.5 seconds. The minimum settling time that can be obtained from distance-based formation control with a mobile holonomic robot model is up to 2 seconds.

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

Abstrak— Mengendalikan sekelompok robot memiliki tantangan dalam permasalahan koordinasi pergerakan. Salah satu topik penelitian untuk mengendalikan sekelompok robot adalah kendali formasi. Kendali formasi memiliki tujuan untuk mengendalikan sekelompok robot dalam mencapai formasi tertentu dan dapat mempertahankan formasi tersebut ketika berpindah menuju arah yang diinginkan. Salah satu metode kendali formasi menggunakan sensor jarak untuk mencapai dan mempertahankan formasi. Pengembangan Kendali formasi berdasarkan jarak telah banyak dilakukan menggunakan model yang sederhana maka dibutuhkan pengembangan lebih lanjut penerapannya ke model yang lebih nyata. Terdapat permasalahan ketika kendali formasi berdasarkan jarak diterapkan pada mobile robot, yaitu robot tidak dapat mengetahui koordinat tetangganya dikarenakan besaran jarak tidak memiliki arah. Maka dikembangkan algoritma *cosinus* untuk menemukan koordinat tetangga sebagai solusi kondisi awal pada kendali formasi berdasarkan jarak. Pengembangan algoritma sekaligus menggunakan model robot *holonomic* sebagai langkah awal pengembangan menggunakan robot yang lebih nyata. Dari hasil percobaan dengan penerapan algoritma *cosinus* dan kendali formasi berdasarkan jarak menggunakan model mobile robot *holonomic*, robot dapat menemukan koordinat tetangganya dengan waktu yang cukup singkat dan *settling time* yang dicapai cukup baik. Rata-rata waktu yang dibutuhkan untuk algoritma *cosinus* mendapatkan koordinat tetangganya adalah 6.5 detik. Waktu *settling time* minimal yang dapat diperoleh dari kendali formasi berdasarkan jarak dengan model robot mobile *holonomic* mencapai 2 detik.

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

I. PENDAHULUAN

Kendali formasi adalah topik penelitian kendali *multi-robot* untuk memecahkan permasalahan koordinasi pergerakan [1]. Kendali formasi bertujuan untuk mengendalikan sekelompok robot dalam mencapai formasi tertentu dan dapat mempertahankan formasi tersebut ketika bermanuver menuju arah yang diinginkan. Pengembangan kendali formasi dilakukan dari berbagai strategi [2], yaitu *leader-follower* [3],

berdasarkan tingkah laku dengan *Fuzzy-Logic* [4], struktur virtual [5].

Dari berbagai pengembangan tersebut dapat diambil garis besar menjadi 3 bagian [6], 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". Dikhususkan pada kendali formasi berdasarkan jarak, Variabel yang dikendalikan pada metode ini adalah variabel jarak antar

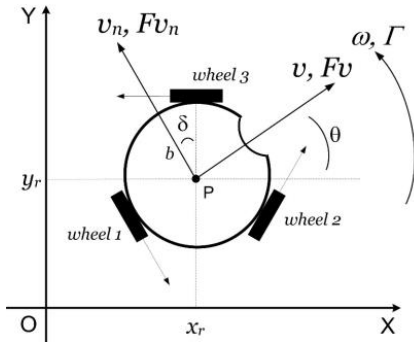
agent yang terhubung, Koordinat yang digunakan tidak mengacu pada koordinat global. Sehingga pada penerapannya, formasi berdasarkan jarak menggunakan sensor yang lebih sedikit. Namun salah satu permasalahan pada metode tersebut adalah penerapan model yang lebih nyata. Pengembangan formasi berdasarkan jarak telah dikembangkan menggunakan teori *graph* pada single dan double integrator [7] dan telah diterapkannya pada simpel model dengan kendali *Proportional-Integral* (PI) [8].

Kendali PI pada penelitian sebelumnya 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. Selain itu, informasi jarak adalah besaran yang tidak memiliki arah sehingga robot diharuskan untuk mengetahui koordinat tetangganya untuk mencapai jarak yang diinginkan.

Pada penelitian ini akan dikembangkan sebuah algoritma untuk menemukan koordinat tetangga menggunakan informasi jarak dan digunakan untuk mengetahui koordinat tetangga pertama kali sebelum menjalankan kendali formasi berdasarkan jarak. nilai kondisi awal pada kendali formasi berdasarkan jarak. Percobaan akan menggunakan model robot holonomic dengan harapan menjadi langkah awal untuk mengembangkan kendali formasi berdasarkan jarak menggunakan model robot yang lebih nyata.

II. METODE

A. Model Robot



Gambar 1 Geometri Rangka Robot [9]

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* [9].

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

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

Dimana vector $u(t) = [u_1(t) \ u_2(t) \ u_3(t)]^T$ adalah masukan model bersatuan *volt* yang memiliki tegangan motor $-6 \leq u_i \leq 6$ *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. Matrix $A_r \in R^{3 \times 3}$ dan $B_r \in R^{3 \times 3}$ adalah parameter fisik robot yang diperoleh dari persamaan fisika. Matrix $K_r \in R^{3 \times 3}$ adalah parameter *friction* dari robot yang diestimasi dari hasil identifikasi sistem dengan cara percobaan.

B. Kendali Robot *Holonomic*

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

1) Mode Satu

Kendali robot mode satu bertujuan untuk mencapai kecepatan robot yang diinginkan. Untuk mencapai tujuan tersebut akan menggunakan persamaan kendali sebagai berikut

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

Keterangan:

$$r_1^c = [v^* \ v_n^* \ w^*]$$

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

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

$$\dot{x}(t) = A_c x(t) + B_c r_1^c + K_r \text{sgn}(x(t)) \quad (4)$$

Keterangan:

$$A_c = (A_r - B_r K_1^c)$$

$$B_c = B_r N_1^c$$

2) Mode Dua

Kendali robot mode dua bertujuan untuk mencapai titik koordinat tertentu berdasarkan koordinat frame robot. Untuk mencapai tujuan tersebut akan dimodifikasi persamaan (1) dengan menambah koordinat *frame* robot pada *state* nya

$$\dot{x}_{c2}(t) = A_{c2} x_{c2}(t) + B_{c2} u_{c2}(t) + K_{c2}(x_{c2}(t)) \quad (5)$$

$$y_{c2}(t) = C_{c2} x_{c2}(t)$$

Keterangan:

$$x_{c2}(t) = [x_r(t) \ y_r(t) \ \theta_r(t) \ v(t) \ v_n(t) \ w(t)]^T$$

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

$$B_{c2} = [0 B_r] \in R^{6 \times 3}$$

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

Dimana $u_{c2}(t)$ adalah persamaan kendali mode dua.

$$u_{c2}(t) = -K_2^c x_{c2}(t) + N_2^c r_2^c \quad (6)$$

Keterangan:

$$r_2^c = [x_r^* \quad y_r^* \quad \theta_r^* \quad v^* \quad v_n^* \quad w^*]^T$$

Dimana $r_2^c, K_2^c \in R^{3 \times 6}$ dan $N_2^c \in R^{3 \times 6}$ adalah setpoint kendali mode dua dan kostanta yang diperoleh dari cara yang sama pada mode satu menggunakan matrix model yang dimodifikasi.

C. Kendali Formasi Berdasarkan Jarak

Didefinisi sebuah *undirect graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 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 $x_i \in R^3$ 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 [8]

$$R \triangleq \text{diag}(e_i^T)(E^T \otimes I_2) \in R^{m \times 3n} \quad (7)$$

Keterangan:

$$\text{diag}(A_i) \triangleq \text{blkdiag}\{A_1, \dots, A_n\} \in R^{np \times np}$$

$$I_2 = \text{Matrik identitas } 2 \times 2$$

Dimana $E \in R^{n \times m}$, adalah matrik *incidence* $\{0, \pm 1\}$ yang baris dan kolom matrik nya mengindikasikan simpulnya dan sisinya. Kendali formasi mengadopsi dari penelitian sebelumnya [7] dimana persamaan potensial didefinisikan sebagai berikut.

$$\Phi(e) = \frac{1}{2} \sum_{i \in \mathcal{V}} \|v_i\|^2 + \frac{1}{2} \sum_{k=1}^m (\|e_k\|^2 - d_k^2)^2 \quad (8)$$

Lalu, *Gradient negative* dari persamaan (8) akan digunakan untuk masukan dari persamaan (4) dan diberikan konstanta *proportional* [8]. Berikut adalah persamaan model yang dijalankan secara simulasi.

$$\dot{p} = A_f p(t) + B_f \frac{\partial \Phi(e)}{\partial v} = A_f p(t) + B_f v(t) \quad (9)$$

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

$$= -k_{p1} v(t) + R^T(p(t)) k_{p2} (R(p(t))p(t) - d) \quad (10)$$

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 .

D. Algoritma Cosinus

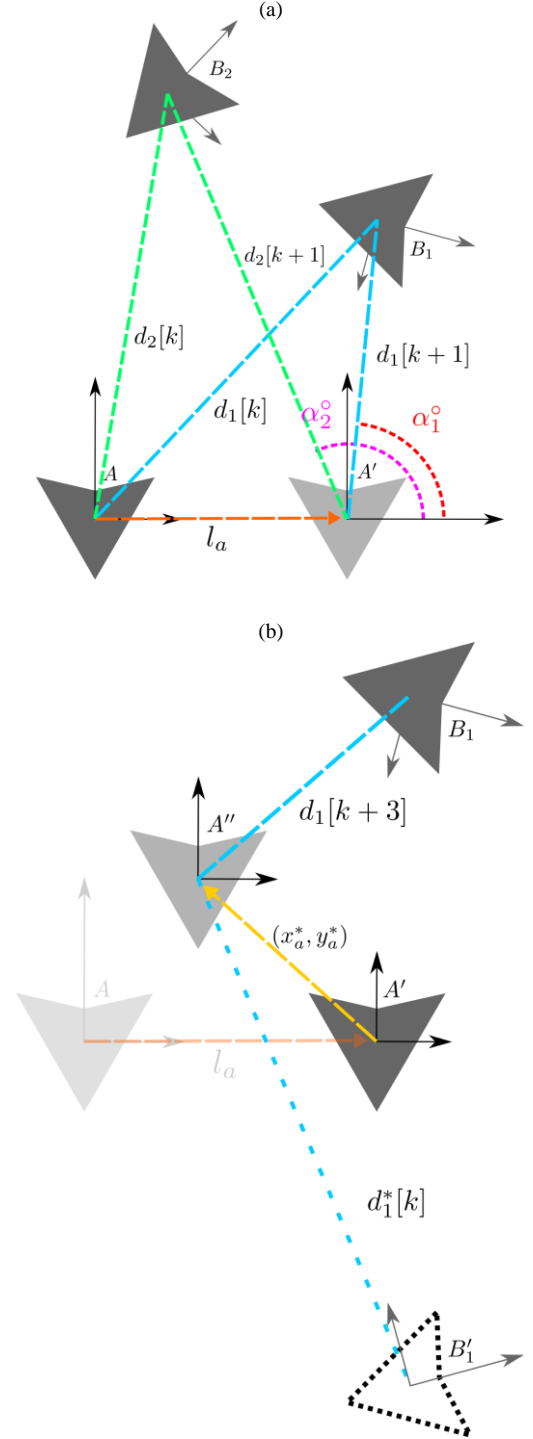
Dinotasikan \mathcal{N}_A adalah himpunan tetangga dari robot A . Dinotasikan $B_i \in \mathcal{N}_A$ adalah robot i sebagai tetangga dari robot A . Dinotasikan $d_i[k]$ adalah informasi jarak antara robot A ke robot B_i . Langkah pertama (Gambar 2a) yang dilakukan oleh robot A adalah menyimpan jarak $d_i[k]$ dan menggunakan kendali mode dua pada persamaan (6) 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 dan sebelum berpindah tersebut akan digunakan untuk menentukan

sudut α_i^o menggunakan rumus segitiga *cosinus* pada persamaan (11).

$$\zeta_i^a = \cos^{-1} \left(\frac{l_a^2 + d_i[k+1]^2 - d_i[k]^2}{2d_i[k+1]l_a} \right) \quad (11)$$

$$\alpha_i^a = 180^\circ \pm \zeta_i^a \quad (12)$$

Dari persamaan (12) akan didapat koordinat tetangga menggunakan vector berikut.



Gambar 2. Ilustrasi Strategi Algoritma *cosinus*
(a) Pencarian Koordinat Tetangga; (b) Pencarian Koordinat Tetangga,

$$x_{B_i}^A = \begin{bmatrix} x_{B_i} = d_i[k+1] \cos \alpha_i^\circ \\ y_{B_i} = d_i[k+1] \sin \alpha_i^\circ \end{bmatrix} \quad (13)$$

Hasil koordinat yang dihasilkan pada persamaan (13) akan menghasilkan bias dikarenakan letak kuadran sudut yang dihasilkan pada persamaan (12). Sudut ζ_i^a adalah sudut segitiga (Gambar 2a) $\angle AA'B_1$ atau $\angle AA'B_2$. Oleh karena itu pada persamaan (12) terdapat operasi \pm dimana operasi tersebut akan dilakukan berdasarkan letak kuadran B_i .

$$\alpha_i^\circ = \begin{cases} 180^\circ - \zeta_i^a, & \text{Kuadran 1 dan 2} \\ 180^\circ + \zeta_i^a, & \text{Kuadran 3 dan 4} \end{cases} \quad (14)$$

Maka perlu satu Langkah lagi untuk menentukan kejadian pada persamaan (14). Langkah tersebut adalah memindah robot lagi ke koordinat $A'' = (x_a^*, y_a^*)$ (Gambar 2b). Setelah dilakukan perpindahan maka akan dibandingkan jarak yang diperoleh dari sensor dengan jarak yang dihasilkan dari koordinat persamaan (13). Apabila terdapat perbedaan maka kejadian pada persamaan (14) diubah ke kejadian selanjutnya.

III. HASIL

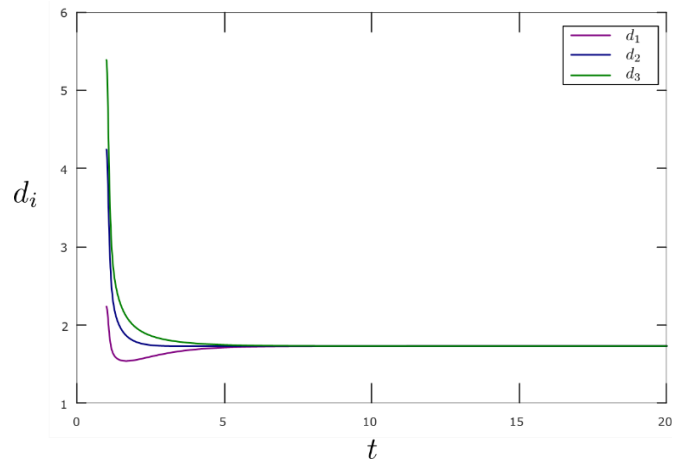
Kendali formasi berdasarkan jarak akan dijalankan secara simulasi menggunakan persamaan (9) dan (10). Simulasi akan menggunakan 3 robot sehingga himpunan simpul $\mathcal{V} = R_1, R_2, R_3$. Menggunakan himpunan sisi $\mathcal{E} = \{(R_1, R_2), (R_3, R_2), (R_3, R_1)\}$ sebagai konfigurasi jaringan antar robot sehingga variable jarak $d_1 = \|x_1 - x_2\|$, $d_2 = \|x_3 - x_2\|$ dan $d_3 = \|x_3 - x_1\|$. Dapat diperhatikan pada Gambar 3 adalah grafik jarak d_i terhadap waktu dan grafik pergerakan robot terhadap koordinat global robot R_1 . Gambar grafik tersebut adalah hasil dari kendali formasi yang tidak diimplementasi algoritma *cosinus* dengan asumsi nilai state persamaan (9) dan (10) ditentukan secara langsung. Gambar 4 adalah grafik jarak d_i dan pergerakan robot terhadap koordinat global robot R_1 yang diimplementasi dengan algoritma *cosinus*. Dapat diperhatikan pada Gambar 3d, Robot R_1 menjalankan algoritma *cosinus* langkah 1 dimana robot berpindah sepanjang $l_a = 1$ sehingga *setpoint* kendali persamaan (6) adalah $r_2^c = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$. Setelah *setpoint* tercapai, robot R_1 mendapatkan jarak yang dibutuhkan untuk menyelesaikan persamaan (12) dan (13) lalu dilanjutkan ke langkah ke 2 dengan berpindah ke koordinat (1,1) sehingga *setpoint* kendali persamaan (6) adalah $r_2^c = [1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$. Setelah *setpoint* tercapai maka dilakukan pengecekan kejadian persamaan (14) dengan membandingkan jarak dari sensor dengan jarak dari koordinat estimasi. Robot R_3 berada pada kuadran 4 maka case yang digunakan adalah $\alpha_i^\circ = 180^\circ + \zeta_i^a$ dan robot R_2 berada pada kuadran 4 maka case yang digunakan adalah $\alpha_i^\circ = 180^\circ - \zeta_i^a$. Setelah koordinat ditemukan, maka robot menggunakan kendali formasi berdasarkan jarak pada persamaan (9) dan (10).

Dilakukan percobaan *settling time* dengan beberapa koordinat yang mewakili kombinasi dua robot dan kejadian pada persamaan (14). Serta diberikan nilai K_{p1} dan K_{p2} pada

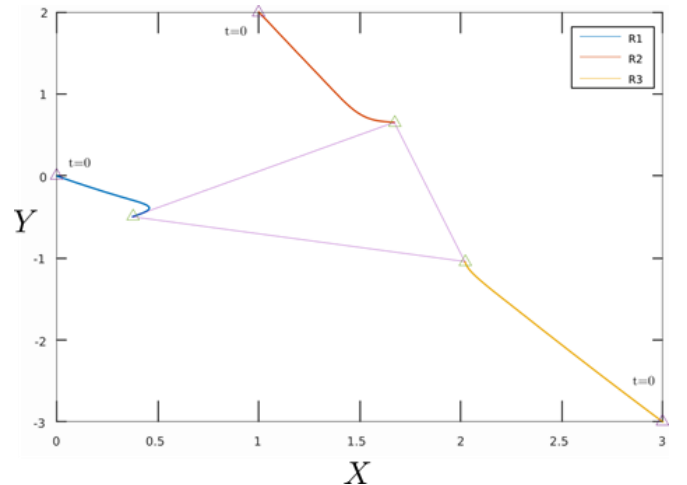
persamaan (10) yang berbeda juga. Dapat diperhatikan pada Tabel 1 bahwa dari selisih penggunaan algoritma dan tidak, memiliki rata-rata 6.5 detik. Artinya bahwa algoritma membutuhkan waktu 5-7 detik untuk menemukan koordinat tetangganya. Konstanta K_p mempengaruhi *settling time* robot untuk menuju jarak yang sama dan apabila dilihat dari baris kedua mengalami saturasi pada nilai 2 detik dengan nilai K_p yang lebih besar.

IV. PENUTUP

Berdasarkan hasil percobaan, rata-rata waktu yang dibutuhkan untuk algoritma *cosinus* menemukan koordinat tetangganya adalah 6.5 detik. *Settling time* minimal yang bisa dicapai oleh kendali formasi berdasarkan jarak menggunakan model robot *holonomic* adalah 2 detik.



(a) Grafik jarak setiap tetangga robot



(b) Grafik jarak setiap tetangga robot

Gambar 3. Grafik Jarak dan Pergerakan Robot Tanpa Menggunakan Algoritma *cosinus*

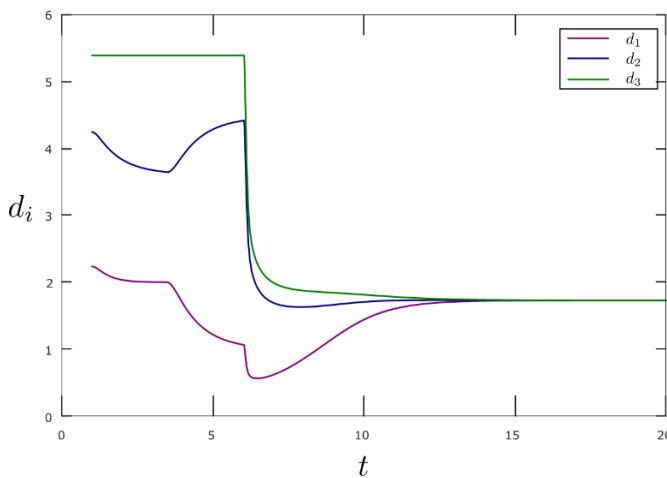
TABEL 1. *SETTLING TIME* DENGAN KONSTANTA KP YANG BERBEDA

Koordinat			Settling Time Tanpa Algoritma <i>cosinus</i> (detik)			Settling Time 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

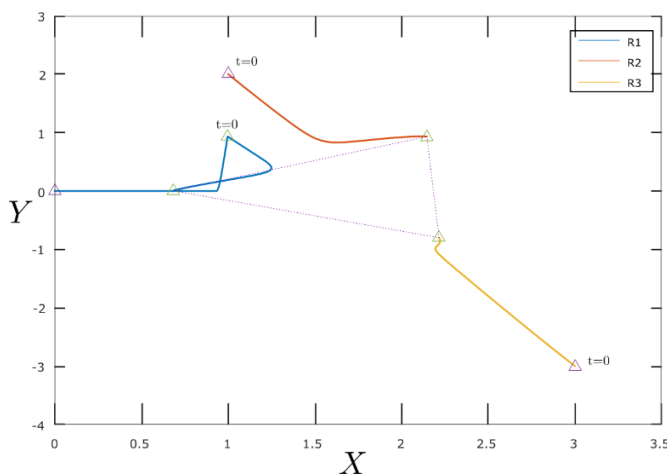
Konstanta:
 (A) $K_{p1} = 50$; $K_{p2} = 3$; (B) $K_{p1} = 80$; $K_{p2} = 7$; (C) $K_{p1} = 100$; $K_{p2} = 15$

REFERENSI

- [1] L. Parker, "Current research in multirobot systems," *Artificial Life and Robotics*, vol. 7, pp. 1–5, 2003, doi: 10.1007/BF02480877.
- [2] W. Guanghua, L. Deyi, G. Wenyan, and J. Peng, "Study on Formation Control of Multi-Robot Systems," 2013, pp. 1335–1339, doi: 10.1109/ISDEA.2012.316.
- [3] X. Wang, Z. Yan, and J. Wang, "Model predictive control of multi-robot formation based on the simplified dual neural network," in *2014 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2014, pp. 3161–3166, doi: 10.1109/IJCNN.2014.6889491.
- [4] S. el Ferik, M. T. Nasir, and U. Baroudi, "A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space," *Applied Soft Computing*, vol. 44, pp. 117–127, 2016, doi: <https://doi.org/10.1016/j.asoc.2016.03.018>.
- [5] C. Yoshioka and T. Namerikawa, "Formation Control of Nonholonomic Multi-Vehicle Systems based on Virtual Structure," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5149–5154, 2008, doi: <https://doi.org/10.3182/20080706-5-KR-1001.00865>.
- [6] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015, doi: <https://doi.org/10.1016/j.automatica.2014.10.022>.
- [7] K.-K. Oh and H.-S. Ahn, "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1809–1820, 2014, doi: 10.1002/rnc.2967.
- [8] O. Rozenheck, S. Zhao, and D. Zelazo, "A proportional-integral controller for distance-based formation tracking," in *2015 European Control Conference (ECC)*, Jul. 2015, pp. 1693–1698, doi: 10.1109/ECC.2015.7330781.
- [9] M. D. Correia, A. Gustavo, and S. Conceição, "Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 7–12, 2012, doi: <https://doi.org/10.3182/20120905-3-HR-2030.00002>.



(a) Grafik jarak setiap tetangga robot



(b) Grafik jarak setiap tetangga robot

Gambar 4. Grafik Jarak dan Pergerakan Robot Menggunakan Algoritma *cosinus*