

KENDALI FORMASI BERDASARKAN JARAK MENGGUNAKAN ALGORITMA COSINUS PADA MOBILE ROBOT

ANGGORO DWI NUR ROHMAN

UNIVERSITAS BRAWIJAYA

March 17, 2021

Abstract

Penelitian ini ditujukan untuk mengembangkan algoritma kendali formasi berdasarkan jarak pada multi mobile robot dimana setiap robot hanya bisa mendeteksi tetangganya saja. Kendali formasi berdasarkan jarak diterapkan pada model holonomic mobile robot menggunakan omniwheel. Algoritma cosinus digunakan untuk menemukan koordinat tetangga pada kondisi awal. Hasil percobaan dibuktikan secara grafik dari perbandingan menggunakan algoritma dan tidak, bahwa penerapan algoritma dapat mendeteksi koordinat tetangga pada kondisi awal dan tidak mempengaruhi kendali formasi.

I Pendahuluan

Kendali formasi adalah topik penelitian kendali multi-robot untuk memecahkan permasalahan koordinasi pergerakan [6]. 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 [3], yaitu *leader-follower* [8], berdasarkan tingkah laku dengan *Fuzzy-Logic* [2], struktur virtual [9].

Dari berbagai pengembangan tersebut dapat diambil garis besar menjadi 3 bagian [5], yaitu berdasarkan posisi, perpindahan, dan jarak. Ketiga bagian tersebut tertuju pada jawaban dari pertanyaan, "variable apa yang digunakan sebagai sensor" dan "variable apa yang aktif dikendalikan oleh sistem multi-robot untuk mencapai formasi yang diinginkan". Dikususkan pada kendali formasi berdasarkan jarak, Variable yang dikendalikan pada metode ini adalah variabel jarak antar agent yang terhubung, Koordinat yang digunakan tidak mengacu pada koordinat global. Sehingga pada penerapannya, for-

masi 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 [4] dan menerapkannya pada simpel model dengan kendali *Proportional-Integral* [7]. Akan tetapi pada penerapannya, pengukuran jarak antar tetangga diperoleh dari selisih koordinat global robot dan tetangganya. Sedangkan dalam praktiknya robot hanya bisa mengukur jarak dan tidak mengetahui koordinat dari robot tetangga.

Pada penelitian ini akan dikembangkan sebuah algoritma untuk mengetahui koordinat tetangga berdasarkan informasi sensor jarak sehingga hasil pencarian koordinat tersebut dapat digunakan pada kendali formasi berdasarkan jarak. Percobaan akan menggunakan model robot holonomic dengan harapan menjadi langkah awal mengembangkan kendali formasi berdasarkan jarak menggunakan model robot yang lebih nyata.

II Metode

i. Kendali Robot Holonomic

Berikut adalah model dari robot holonomic dalam bentuk *state-space* [1]. Dimana robot menggunakan tiga buah motor yang dihubungkan pada *omniwheel* sehingga robot dapat bergerak kesegala arah.

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

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

Vector $u(t) = [u_1(t) \ u_1(t) \ u_2(t)]^T$ adalah masukan model bersatuan *volt* dengan batasan $-6 \leq u_i(t) \leq 6$ pada tegangan motor robot. Vector $y(t) = x(t) = [v(t) \ v_n(t) \ w(t)]^T$ adalah kecepatan robot yang akan diperoleh dari sensor percepatan, dimana $v(t)$ adalah kecepatan pada sumbu x , $v_n(t)$ adalah kecepatan pada sumbu y , dan $w(t)$ adalah kecepatan rotasi dari frame robot. Matrix $A_r \in \mathbb{R}^{3 \times 3}$ dan $B_r \in \mathbb{R}^{3 \times 3}$ adalah parameter fisik robot berdasarkan yang diperoleh dari identifikasi secara persamaan fisika. Matrix $K_r \in \mathbb{R}^{3 \times 3}$ adalah parameter *friction* dari robot yang diestimasi dari hasil percobaan.

Kendali dari robot akan menggunakan dua mode *state-feedback*. **Mode satu**, bertujuan untuk mencapai kecepatan robot yang diinginkan. Untuk mencapai tujuan tersebut akan menggunakan persamaan kendali sebagai berikut

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

$$r_1^c = [v^*(t) \ v_n^*(t) \ w^*(t)]^T$$

Dimana $r_1^c, K_1^c \in \mathbb{R}^{3 \times 3}$ dan $N_1^c \in \mathbb{R}^{3 \times 3}$ adalah set-point 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*.

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

Dengan menggabungkan persamaan (1) dengan persamaan (3) akan mendapatkan persamaan *state-space* robot yang baru

$$\begin{aligned} \dot{x}(t) &= (A_r - B_r K_1^c) x(t) + B_r N_1^c r_1^c + K_r \text{sgn}(x(t)) \\ &= A_c x(t) + B_c r_1^c + K_r \text{sgn}(x(t)) \end{aligned} \quad (4)$$

Mode dua, bertujuan untuk mencapai titik koordinat tertentu berdasarkan koordinat frame robot. Untuk mencapai tujuan tersebut akan dimodifikasi persamaan (1) dengan menambah state koordinat frame robot menjadi $y_{c2}(t) = x_{c2}(t) = [x_r(t) \ y_r(t) \ \theta_r(t) \ v(t) \ v_n(t) \ w(t)]^T$, $A_{c2} = \begin{bmatrix} 0 & I \\ 0 & A_r \end{bmatrix} \in \mathbb{R}^{6 \times 6}$, $B_{c2} = \begin{bmatrix} 0 \\ B_r \end{bmatrix} \in \mathbb{R}^{6 \times 3}$, $K_{c2}(x) = \begin{bmatrix} 0 & 0 \\ 0 & K_r \text{sgn}(x) \end{bmatrix} \in \mathbb{R}^{6 \times 3}$. Berikut adalah state space model untuk kendali mode satu.

$$\begin{aligned} \dot{x}_{c2}(t) &= A_{c2} x_{c2}(t) + B_{c2} u_{c2}(t) + K_{c2}(x_{c2}(t)) \\ y_{c2}(t) &= C_{c2} x_{c2}(t) \end{aligned}$$

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

$$u_{c2}(t) = -K_2^c x_{c2}(t) + N_2^c r_2^c \quad (5)$$

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

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

ii. Kendali Formasi Berdasarkan Jarak

Dinotasikan $n \triangleq |\mathcal{V}|$ sebagai jumlah dari node dan $m \triangleq |\mathcal{E}|$ sebagai jumlah dari sisinya. Dinotasikan $p = [x_1^T \ \dots \ x_n^T]^T \in \mathbb{R}^{3n}$, dimana $x_i \in \mathbb{R}^3$ 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 $e = [e_1^t \dots e_m^t] \in \mathbb{R}^{3m}$. didefinisikan fungsi *Jacobian* sisi [7]),

$$R(p) \triangleq \text{diag}(e_i^T)(E^T \otimes I_2) \in \mathbb{R}^{m \times 3n} \quad (6)$$

Dimana $E \in \mathbb{R}^{n \times m}$, adalah matrik *incidence* $\{0, \pm 1\}$ dimana baris matrik mengindikasikan simpulnya dan kolomnya sebagai sisinya dan $\text{diag}(A_i) \triangleq \text{blkdiag}\{A_1, \dots, A_n\} \in \mathbb{R}^{n \times n}$ Orde kendali formasi yang digunakan adalah orde dua Mengadopsi persamaan potensial, didefinisi persamaan [4].

$$\Phi(e) = \frac{1}{2} \sum_{i \in \mathcal{V}} \|v_i\|^2 + \frac{1}{2} \sum_{k=1}^m (\|e_k\|^2 - d_k^2)^2 \quad (7)$$

Lalu kendali formasi menggunakan persamaan (4) sebagai modelnya diberikan *negative gradient* dari fungsi potensial dan konstanta *proportional* [7]

$$\begin{aligned} \dot{p} &= A_f p(t) + B_f \frac{\partial \Phi(e)}{\partial v} \\ &= A_f p(t) + B_f v(t) \\ \dot{v} &= -C \left(\frac{\partial \Phi(e)}{\partial v} + \frac{\partial \Phi(e)}{\partial p} \right) \\ &= -k_{p1} v(t) + R(p(t))^T k_{p2} (R(p(t))p(t) - d) \end{aligned} \quad (8)$$

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

iii. Algoritma Cosinus

Robot $B_i \in \mathcal{N}_A$, adalah tetangga dari robot A . $d_i[k]$ adalah jarak yang diperoleh dari sensor. Dalam strategi (Gambar 1a) ini diperlukan perpindahan robot A ke $A' = (0, l_a)$. Perpindahan tersebut akan menghasilkan jarak $d_i[k+1]$. Dari perbedaan terse-

but akan didapatkan sudut α_i° .

$$\begin{aligned} \alpha_i^\circ &= 180^\circ \pm \cos^{-1} \left(\frac{l_a^2 + d_i[k+1]^2 - d_i[k]^2}{2d_i[k+1]l_a} \right) \\ &= 180^\circ \pm \zeta_i^a \end{aligned} \quad (9)$$

Dari persamaan (9) akan didapat koordinat tetangga.

$$x_{B_i}^A = \begin{cases} x_{B_i} = d_i[k+1] \cos \alpha_i^\circ \\ y_{B_i} = d_i[k+1] \sin \alpha_i^\circ \end{cases} \quad (10)$$

Untuk memvalidasi apakah koordinat telah sesuai dapat menggunakan nilai jarak pada sensor dibandingkan dengan jarak dari hasil koordinat persamaan (10). Akan tetapi hasil validasi tersebut akan mengalami bias dikarenakan sudut ζ_i^a adalah sudut segitiga $\angle AA'B_1$ atau $\angle AA'B_2$. Oleh karena itu pada persamaan (9) terdapat operasi \pm dimana operasi tersebut akan dilakukan berdasarkan letak kuadran B_i .

$$\alpha_i^\circ = \begin{cases} 180^\circ - \zeta_i^a, & \text{Kuadran I,II} \\ 180^\circ + \zeta_i^a, & \text{Kuadran III,IV} \end{cases} \quad (11)$$

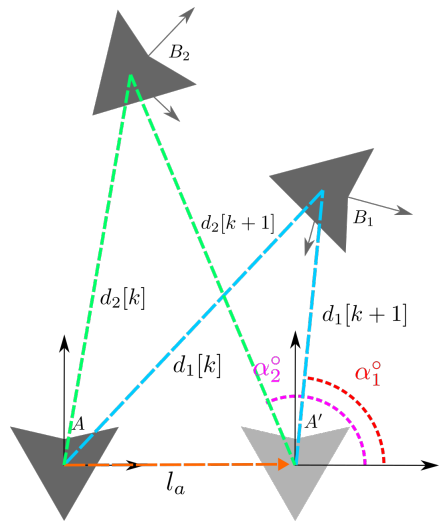
Diperlukan satu langkah lagi untuk menentukan kejadian pada persamaan (11)(Gambar 1b). Langkah 2 adalah langkah pengujian dari hasil koordinat yang telah dikalkulasi dan membandingkannya dengan nilai yang didapat dari sensor. Apabila terdapat perbedaan maka kejadian pada persamaan (11) diubah ke kejadian selanjutnya.

III Hasil

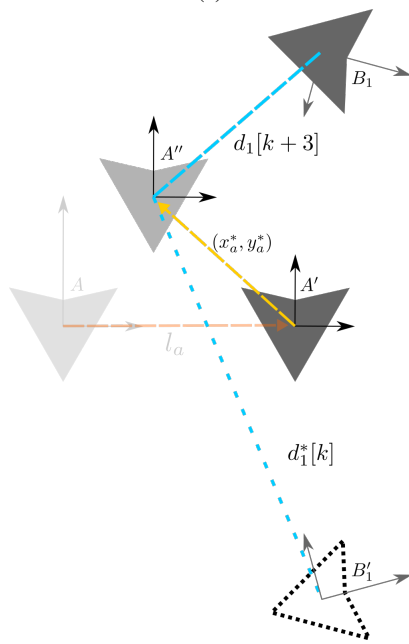
Berikut pada gambar 2 adalah hasil dari algoritma cosinus. Parameter $l_a = 1$ pada persamaan (9) dan $(x_a^*, y_a^*) = (1, 1)$. Pada gambar (2c,2d) dan gambar (2a,2b) menunjukkan bahwa algoritma tidak mempengaruhi kendali formasi.

IV DAFTAR REFERENSI

- [1] Mariane Dourado Correia, André Gustavo, dan Scolari Conceição. "Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models". In: *IFAC Proceedings Volumes* 45.22 (2012). 10th IFAC Symposium on Robot Control, pp. 7 –12. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20120905-3-HR-2030.00002>. URL: <http://www.sciencedirect.com/science/article/pii/S1474667016335807>.
- [2] Sami El Ferik, Mohammad Tariq Nasir, dan Uthman Baroudi. "A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space". In: *Applied Soft Computing* 44 (2016), pp. 117 –127. ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2016.03.018>. URL: <http://www.sciencedirect.com/science/article/pii/S1568494616301272>.
- [3] Wang Guanghua, dkk. "Study on Formation Control of Multi-Robot Systems". In: Jan. 2013, pp. 1335–1339. ISBN: 978-1-4673-4893-5. DOI: 10.1109/ISDEA.2012.316.
- [4] Kwang-Kyo Oh, dan Hyo-Sung Ahn. "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space". In: *International Journal of Robust and Nonlinear Control* 24.12 (2014), pp. 1809–1820. DOI: 10.1002/rnc.2967. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.2967>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.2967>.
- [5] Kwang-Kyo Oh, Myoung-Chul Park, dan Hyo-Sung Ahn. "A survey of multi-agent formation control". In: *Automatica* 53 (2015), pp. 424 –440. ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2014.10.022>. URL: <http://www.sciencedirect.com/science/article/pii/S0005109814004038>.
- [6] Lynne Parker. "Current research in multi-robot systems". In: *Artificial Life and Robotics* 7 (Mar. 2003), pp. 1–5. DOI: 10.1007/BF02480877.
- [7] O. Rozenheck, S. Zhao, dan D. Zelazo. "A proportional-integral controller for distance-based formation tracking". In: *2015 European Control Conference (ECC)*. 2015, pp. 1693–1698. DOI: 10.1109/ECC.2015.7330781.
- [8] X. Wang, Z. Yan, dan 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)*. 2014, pp. 3161–3166. DOI: 10.1109/IJCNN.2014.6889491.
- [9] Chika Yoshioka, dan Toru Namerikawa. "Formation Control of Nonholonomic Multi-Vehicle Systems based on Virtual Structure". In: *IFAC Proceedings Volumes* 41.2 (2008). 17th IFAC World Congress, pp. 5149 –5154. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20080706-5-KR-1001.00865>. URL: <http://www.sciencedirect.com/science/article/pii/S1474667016397609>.

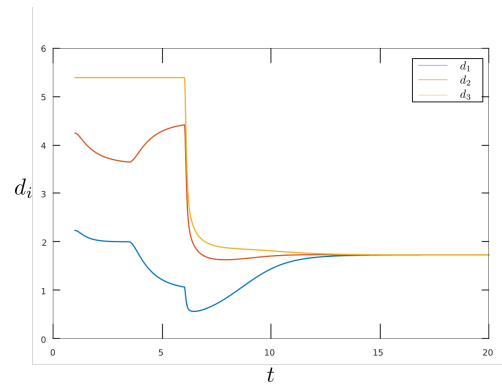


(a)

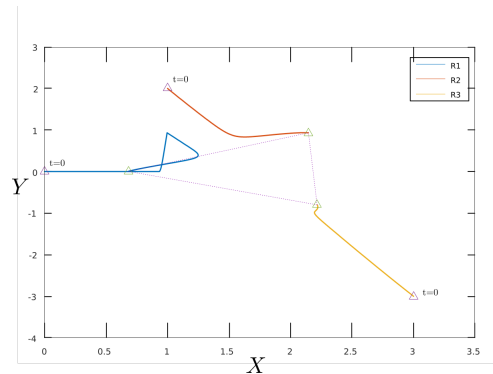


(b)

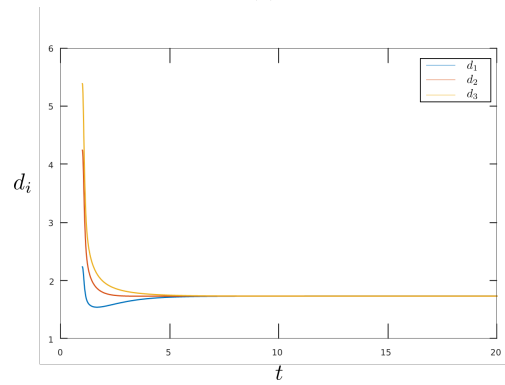
Gambar 1: Strategi penentuan koordinat (a) langkah 1 dan (b) Langkah 2



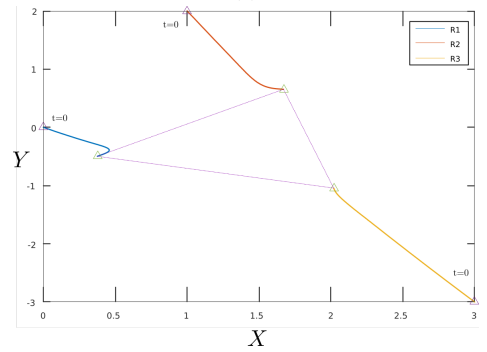
(a)



(b)



(c)



(d)

Gambar 2: Plot menggunakan algoritma (a. Sensor Jarak dan b. Pergerakan robot) dan tidak menggunakan algoritma (c. Sensor Jarak dan d. Pergerakan Robot)