BAB 2
TINJAUAN PUSTAKA

2.1 Studi Pustaka

Pengendali berbasis konsensus cukup banyak digunakan pada pengendalian sistem majemuk. Salah satunya yang dilakukan pada penelitian yang dilakukan Liu dkk[3], melakukan pengendalian *The Tethered Space Net Robot (*TSNR*)* secara simulasi dengan menggunakan *robust distributed coordination control* berdasarkan *adaptive law* dengan memanfaatkan pendekatan Lyapunov dan teori *graph.* Hasil pada penelitian ini menunjukan bahwa *Maneuverable Units* (MUs) serempak dapat mengikuti lintasan yang diinginkan dengan baik dan effektif, serta TSNR dapat menjaga bentuk *capture-available* dengan konfigurasi simetris dan asimetris.

Untuk pengendalian sistem robot beroda majemuk berbasis konsensus, telah ada penelitian yang dilakukan Zhaoxia [4], [5], dengan menggunakan *Nonholonomic mobile robots* yang menghasilkan dua penelitian dengan judul yang berbeda. Pada penelitiannya, mereka menginvestigasi *distributed consensus-based robust adaptive formation control* untuk *nonholonomic mobile robot.*

Dan juga penelitian yang menggunakan *Nonholonomic Robots* juga digunakan pada penelitian Maghenem dkk[6], penelitian ini menggunakan *decentralized consensus-based formation controller* dengan mempertimbangkan model kinematis dan model dinamis untuk menggerakkan secara seragam dan asimtotik kumpulan *nonholonomic mobile-robot* yang terhubung sesuai dengan formasi orientasi yang diinginkan.

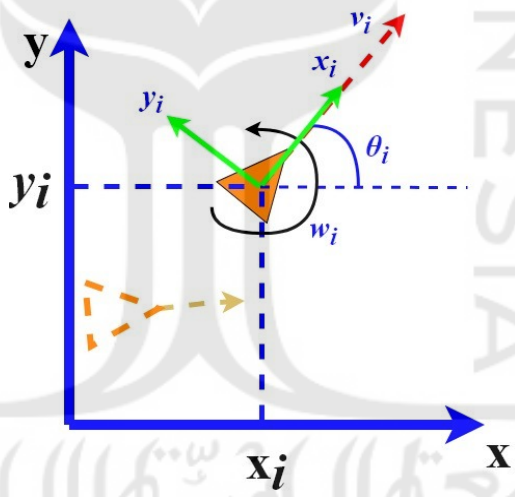
Pengendalian formasi robot beroda majemuk bisa berdasarkan *vector fields,* seperti penelitian yang dilakukan oleh Ji-Wook dkk[7]. Penelitiannya melakukan pengendalian *wheeled Mobile Robots* berbasis metode *vector fields.* Pengendalian formasi ini bersifat hierarki, yang berarti salah satu robot akan menjadi *leader* dari segala kelompok, beberapa robot menjadi *sub-leader* dari kelompoknya, dan ada yang menjadi anggota dari kelompok yang disebut pada *paper* ini *hierarchical formation control.* Penelitian yang sama juga dilakukan oleh Sruthi[6], penelitiannya dimana pengendali formasi dengan adanya *obstacle*.

2.2 Tinjauan Teori

Penelitian yang dilakukan memiliki tema *Swarm Search & Rescue Robotic System* yang berarti menggunakan lebih dari sebuah robot, dan menggunakan 2 buah pengendalian robot terdistribusi untuk mengendalikan robot *unicycle* majemuk yang bersifat *nonholonomic* sebagai objek. Pengendalian formasi tersebut yakni berbasis *vector field* dan berbasis konsensus terdistribusi. Teori tentang robot model *unicycle* dan 2 jenis pengendalian di atas akan dijelaskan di bawah ini.

2.2.1 Robot Model *Unicycle*

Robot model *unicycle* merupakan salah satu jenis robot *nonholonomic mobile.* Model robot tersebut sangat sederhana karena hanya memiliki dua buah masukan yakni kecepatan linear dan kecepatan putar orientasi. Dan keluaran dari model tersebut jika diilustrasikan pada sebuah bidang (*x,y*) pada Gambar 2.1 di yakni kecepatan posisi robot terhadap sumbu x , kecepatan posisi posisi robot terhadap sumbu y, dan kecepatan putar orientasi robot terhadap bidang. Model matematis dari robot *unicycle* jika dituliskan adalah berikut[8]:



Gambar 2.1: Robot Model Unicycle pada bidang

$$q_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & 0 \\ \sin(\theta_i) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \tag{2.1}$$

Keterangan:

$\dot{x}_i$: Kecepatan robot di sumbu X

$\dot{y}_i$: Kecepatan robot di sumbu Y

$\dot{\theta}_i$: Kecepatan angular robot

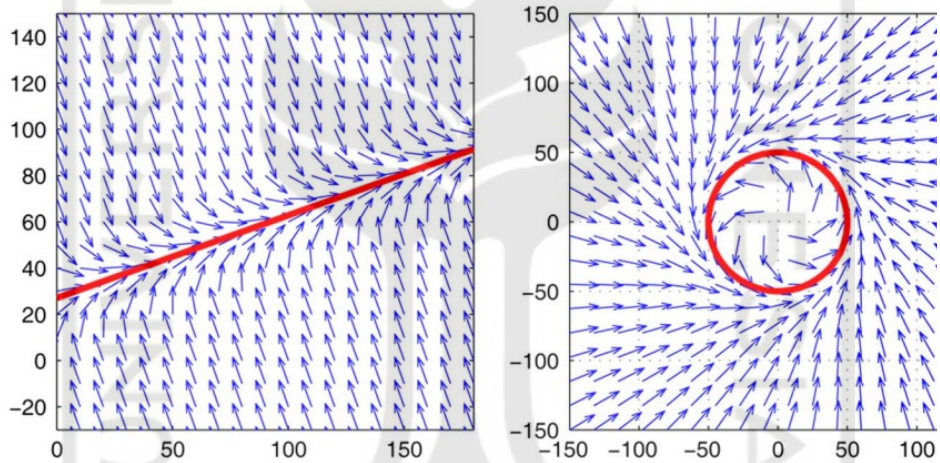
θ_i : Orientasi robot

v_i : Kecepatan linear robot

ω_i :Kecepatan angular robot

2.2.2 Pengendalian Berbasis *Vector Field*

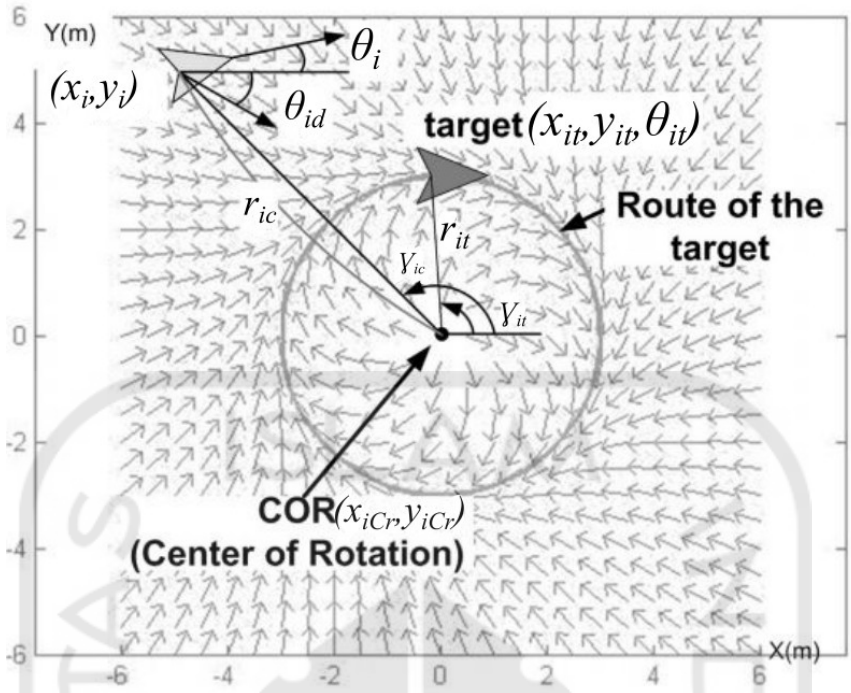
Pengendalian berbasis *vector field* merupakan pengendalian dengan memanfaatkan medan vektor yang merupakan penetapan vektor yang telah ditetapkan pada setiap titik pada sebuah bidang[9]. *Vector field* dibuat berdasarkan jalur yang telah ditentukan, berdasarkan pada tulisan Nelson, Derek[10], terdapat dua buah ide ilustrasi *vector field* yakni, *straight-line path following* dan *circular path following* yang digambarkan pada Gambar 2.2.



Gambar 2.2: Ide *vector field, straight-line* (kiri) dan *circular* (kanan)

Pada pengendalian berbasis *vector field,* peneliti menggunakan referensi yang ditulis oleh Kwon[7] yang melakukan pengendalian formasi secara hierarki dengan metode *vector field.*

Pengendalian formasi robot dengan menggunakan metode *vector field* dimulai dengan menentukan posisi target $[x_{it}\ y_{it}]^T$ yang akan ditentukan pada bab selanjutnya dan posisi aktual $[x_i\ y_i]^T$ pada robot. Dengan kedua referensi tersebut, maka akan didapatkan titik *Center of Rotation* (COR) yang ditunjukan pada Gambar 2.3. Titik COR dirumuskan pada persamaan (2.2), dan titik ini digunakan sebagai acuan untuk menghitung sinyal kendali.



Gambar 2.3: Center of Rotation

$$COR(x_{iCr}, y_{iCr}) = \begin{bmatrix} x_{it} + r_{it}\cos\left\{\theta_{it} + sgn(v_{it})\,sgn(\omega_{it})\frac{\pi}{2}\right\} \\ y_{it} + r_{it}\sin\left\{\theta_{it} + sgn(v_{it})\,sgn(\omega_{it})\frac{\pi}{2}\right\} \end{bmatrix} \tag{2.2}$$

dimana v_{it} dan ω_{it} adalah kecepatan target linear dan kecepatan target angular pada robot.

Selanjutnya pada metode *vector field*, terdapat orientasi robot diinginkan (θ_{id}) yang berdasarkan rumus dasar dari *vector field.* Jika $v_t > 0$, dan $\omega_t > 0$ maka orientasi robot diinginkan (θ_{id}) berdasarkan *vector field* ditulis pada persamaan 2.3 dengan $k_{id} > 0$ dan didukung oleh persamaan 2.4 – 2.5 .

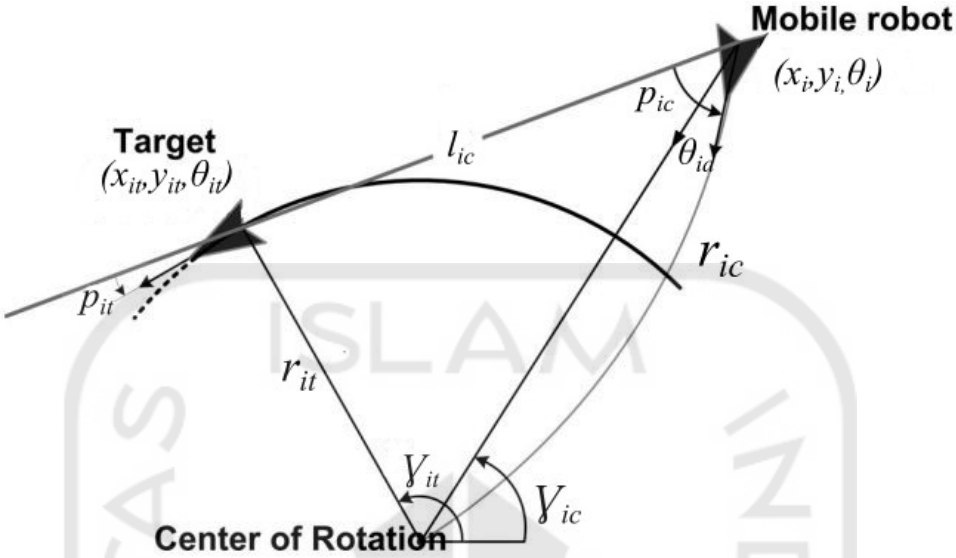
$$\theta_{id}(e_r) = \begin{cases} \gamma_{ic} + sgn(\omega_{it})\left\{\frac{\pi}{2} + \tan^{-1}(k_{id}e_{ir})\right\}, v_{it} \geq 0 \\ \gamma_{ic} + sgn(\omega_{it})\left\{\frac{\pi}{2} + \tan^{-1}(k_{id}e_{ir})\right\} - \pi, v_{it} < 0 \end{cases} \tag{2.3}$$

dimana,

$$\gamma_{ic} = atan2\left(\frac{y_i - y_{iCr}}{x_i - x_{iCr}}\right) \tag{2.4}$$

$$e_{ir} = r_{ic} - r_{it} \tag{2.5}$$

Setelah merumuskan *(θ_{id})* maka selanjutnya merumuskan hubungan posisi target dengan posisi aktual dari agen model robot *unicycle.* Gambar 2.4 menunjukan hubungan antara posisi aktual dan posisi target



Gambar 2.4: Hubungan antara posisi target dan posisi aktual robot

Dengan berdasarkan Gambar 2.4 maka diuraikan menjadi beberapa persamaan mulai dari Persamaan (2.6) – (2.10) yang merumuskan hubungan antara posisi target dan posisi aktual robot.

$$r_{ic} = \sqrt{(y_{ic} - y_{iCr})^2 + (x_{ic} - x_{iCr})^2} \tag{2.6}$$

$$r_{it} = \frac{v_{it}}{\omega_{it}} \tag{2.7}$$

Persamaan (2.6) merupakan rumus dari r_{ic} yang merupakan jarak antara COR dan posisi aktual robot. Selain itu terdapat r_{it} yang merumuskan jarak terhadap posisi target robot dengan COR yang bergantung terhadap v_{it} dan ω_{it} yang dituliskan pada Persamaan (2.7).

$$\theta_{it} = \begin{cases} \gamma_{ic} + sgn(\omega_{it})\frac{\pi}{2}, v_{it} \geq 0 \\ \gamma_{ic} + sgn(\omega_{it})\frac{\pi}{2} - \pi, v_{it} < 0 \end{cases} \tag{2.8}$$

Selanjutnya untuk mencari orientasi posisi target, maka dirumuskan pada Persamaan (2.8) yang telah dirumuskan oleh Kwon[7]. Kemudian untuk q_{ic} atau q_{it} merupakan sudut antara posisi aktual atau posisi target dengan garis lurus yang dibentuk antara posisi aktual dan posisi target

robot yang berinisial l_{jc} yang kemudian dirumuskan pada persamaan (2.9) – (2.10)

$$p_{ic}=\theta_i-atan2\left(\frac{y_{it}-y_i}{x_{it}-x_i}\right) \tag{2.9}$$

$$p_{it}=\theta_{it}-atan2\left(\frac{y_{it}-y_{ic}}{x_{it}-x_{ic}}\right) \tag{2.10}$$

Kemudian sinyal kendali dari metode *vector field* untuk setiap robot-i dirumuskan pada Persamaan (2.11) – (3.22) yang juga telah dirumuskan oleh Kwon[7].

$$\omega_i=\frac{v_i}{r_{ic}}\sin(\theta_i-\gamma_{ic})-sgn(\omega_{it})\frac{k_{id}v_i\cos(\theta_i-\gamma_c)}{1+(k_{id}e_{ir})^2}-k_{i\theta}e_{i\theta} \tag{2.11}$$

$$v_i=\mu(p_{ic})\frac{1}{\cos(p_{ic})}(v_{it}\cos(p_{it})+k_{il}e_{il}) \tag{2.12}$$

dimana,

$$e_{il}=l_{ic}=\sqrt{(y_{it}-y_i)^2+(x_{it}-x_i)^2} \tag{2.13}$$

$$e_{i\theta}=\theta_i-\theta_{id} \tag{2.14}$$

$$\mu(p_{ic})=1-\exp\left\{-\left(|p_{ic}|-\frac{\pi}{2}\right)\Big/\sigma^2\right\} \tag{2.15}$$

Persamaan-Persamaan (2.2) – (2.15) di atas diterapkan pada setiap robot-*i.* Pada metode *vector field* terdapat beberapa kontrol variabel seperti $k_{i\theta}$, ki_d pada Persamaan (2.11) kemudian k_{il} pada Persamaan (2.12) dan σ pada Persamaan (2.15).

2.2.3 Pengendalian Berbasis Konsensus Terdistribusi

Pengendali terdistribusi berbasis konsensus merupakan pengendalian yang menggunakan algoritma konsensus yang digunakan oleh sistem yang memiliki beberapa agen yang akan melakukan negosiasi untuk mencapai suatu nilai konsensus yang dapat diterima oleh semua agen. Untuk mencapai nilai konsensus, maka setiap agen perlu memperoleh informasi dari agen lainnya (tetangga-tetangga agen) yang digunakan sebagai pertimbangan untuk mencapai nilai konsensus. Secara sederhana, algoritma konsensus dituliskan pada persamaan (2.16)[11]:

$$z_i[k+1]=\sum_{j=1}^{n}a_{ij}[k]z_j[k] \tag{2.16}$$

$$a_{ij}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1j} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2j} \\ a_{31} & a_{33} & a_{33} & \dots & a_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{ij} \end{bmatrix} \tag{2.17}$$

Keterangan:

n : Jumlah agen yang digunakan

i, j : merupakan identitas agen, i = 1, 2, ... n

z_i : Nilai variabel pada agen i

a_{ij} : Nilai *Weighted* terhadap *Agent* dalam bentuk matriks,

Pada persamaan (2.3) memiliki variabel nilai *weighted* yang sangat penting dan berpengaruh, karena variabel tersebut menentukan bentuk *topology* pertukaran informasi antar agen. Dan selain itu, nilai tersebut juga memberikan pengaruh berupa *gain* dari informasi antar agen.

Selanjutnya pada pengendalian formasi dengan pengendali distribusi berbasis konsensus, peneliti menggunakan referensi yang ditulis oleh Peng[8]. Pada referensi tersebut, Pengendali dimulai dengan persamaan transformasi yang disimbolkan dengan z_{1i}, z_{2i}, dan z_{3i}. Namun selain itu pada referensi ini menggunakan *virtual leader* untuk membentuk formasi, dan posisi setiap robot berdasarkan formasi yang dituliskan dengan px_i dan py_i yang merupakan jarak setiap robot-i terhadap *virtual leader*.

$$z_{1i}(t)=\theta_i(t),$$

$$z_{2i}(t)=(x_i(t)-px_i)\cos\theta(t)+(y_i(t)-py_i)\sin\theta_i(t)+k_0\,sign(u_{1i}(t))z_{3i}(t), \tag{2.18}$$

$$z_{3i}(t)=(x_i(t)-px_i)\sin\theta(t)+(y_i(t)-py_i)\cos\theta_i(t),$$

$$u_{1i}(t)=\omega_i(t), \tag{2.19}$$

$$u_{2i}(t)=v_i(t)-(1+k_0{}^2)u_{1i}(t)z_{3i}(t) \tag{2.20}$$

Dengan Persamaan (3.6) sampai (3.7), maka Persamaan sinyal kendali adalah sebagai berikut

$$\begin{aligned}\omega_i=u_{1i}=u_{10}-\alpha\sum_{j\in\mathbb{N}_i}a_{ij}(z_{1i}-z_{1j})-\alpha b_i(z_{1i}-z_{10})\\ -\beta\,sign(a_{ij}(z_{1i}-z_{1j})+b_i(z_{1i}-z_{10}))\end{aligned} \tag{2.21}$$

$$
\begin{aligned}
v_i = u_{2i} = -\alpha \sum_{j \in \mathbb{N}_i} a_{ij}(z_{2i} - z_{2j}) - \alpha b_i(z_{2j} - z_{20}) \\
-\beta\, sign(a_{ij}(z_{2i} - z_{2j}) + b_i(z_{2i} - z_{20})) - k_0 |u_{1i}| z_{2i}
\end{aligned}
\tag{2.22}
$$

Pengendali terdistribusi memiliki beberapa variabel kontrol berupa k_0, α, dan β. Selain itu terdapat b_j selain a_{ij} yang merupakan *weighted value* robot-i terhadap *virtual leader.*