

## DESAIN DAN PEMODELAN KONTROL KINEMATIK PERGERAKAN ROBOT BERODA DENGAN MENGUNAKAN 6 RODA OMNI-WHEELS

Indrazno Siradjuddin<sup>1</sup>, Mukhamad Luqman Muttaqin Faizin<sup>2</sup>, Gillang Al Azhar<sup>3</sup>, Anggit Murdani<sup>4</sup>

<sup>1-3</sup>Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

<sup>4</sup>Jurusan Teknik Mesin, Politeknik Negeri Malang, Indonesia

### Informasi Artikel

#### Riwayat Artikel

Diterima 5 maret 2020

Direvisi 18 April 2020

Diterbitkan 28 April 2020

#### Kata kunci:

Robot omnidirectional  
Kontrol robot beroda  
Kontrol kinematik  
Kontrol kinematik kecepatan

#### Keywords:

Omnidirectional Robot  
Mobile Robot Control  
Kinematics Control  
Velocity Kinematics Control

### ABSTRAK

Pengembangan desain kontrol kinematik sangatlah penting dalam pengembangan kontrol untuk robot beroda. Hal ini sangat dibutuhkan mengingat bahwa robot beroda memiliki banyak parameter yang mampu merubah persamaan kontrol kinematiknya terutama pada kontrol kinematik untuk robot beroda dengan jenis roda omnidirectional, baik dari segi jenis roda yang digunakan hingga jumlah roda penggerak yang digunakan. Dengan berbagai macam hal yang dapat merubah persamaan kontrol kinematiknya, maka pada makalah ini dibuat sebuah persamaan kontrol kinematik yang general, yang dapat diaplikasikan untuk berbagai macam roda omnidirectional, serta berbagai jumlah roda yang digunakan. Selain itu persamaan general yang telah dibuat, diaplikasikan untuk menguji respon robot beroda dengan menggunakan 6 buah *omni-wheels* untuk menguji hasil respon dari persamaan general kontrol yang telah dibuat. Pengujian dilakukan dengan menggunakan simulasi program dengan menggunakan pemrograman dengan menggunakan Bahasa pemrograman python. Hasil yang didapatkan menunjukkan robot mampu bergerak sesuai dengan arah gerak target yang ditentukan, yaitu membentuk pola jalur yang linier serta mampu bergerak membentuk pola lingkaran dan pola setengah gelombang sinus. Hal ini menunjukkan bahwa kontrol kinematik yang dirancang mampu membuat robot bergerak sesuai dengan yang direncanakan. Hasil dari respon robot berupa sinyal kontrol, pola yang dibentuk serta nilai perubahan error disajikan dalam bentuk grafik.

### ABSTRACT

*Development of the kinematics control is very important for the development of kinematics control for mobile robots. This is very necessary because mobile robots have a lot of factors that can manipulate the equation of its kinematic control, such as the type of wheels, the number of wheels, etc. With this kind of problem, it necessary to generate a general equation for the robot's kinematic control, which in this journal we purpose the general equation for the mobile robot control, and we evaluate the outcome by applying the general equation into the 6 omnidirectional robot control. To make a valid statement, we simulate the control to understanding the control outcome by using a python program. The results of the simulation show us that the robot can move as planned, that the robot produce a linear trajectory, circular trajectory, and half sine wave trajectory. Depennds on the results, it can be concluded that the proposed kinematics control equation can make the robot moves well as we planned. The results of the respon, the trajectory, and the changes in error values are presented in graphical form.*

### Penulis Korespondensi:

Indrazno Siradjuddin,  
Jurusan Teknik Elektro,  
Politeknik Negeri Malang,  
Jl. Sukarno Hatta No. 9, Malang, Jawa Timur, Indonesia.  
Email: [indrazno@polinema.ac.id](mailto:indrazno@polinema.ac.id)

### 1. PENDAHULUAN

Penelitian yang berhubungan dengan robot semakin banyak mengalami kemajuan, serta semakin menarik untuk dikaji lebih dalam. Hal ini dikarenakan pengaplikasian robot dapat dilakukan diberbagai bidang [1], seperti pada bidang kesehatan [2], industri [3], eksplorasi lingkungan [4], misi penyelamatan [5], pendidikan [6], kebutuhan militer [7], dan masih banyak lagi. Salah satunya adalah penelitian mengenai robot beroda. Penelitian pada robot beroda dilakukan untuk meneliti penggunaan sistem roda *holonomic* [8] serta sistem roda *non-holonomic* [9]. Dimana dari beberapa penelitian tersebut didapatkan beberapa informasi bahwa robot beroda dengan sistem roda penggerak *holonomic* memiliki sifat pergerakan yang lebih bebas dan lebih fleksibel, hal ini dikarenakan konstruksi dari roda penggerak tersebut mendukung agar robot mampu bergerak kesegala arah. Sehingga robot beroda dengan menggunakan tipe roda *omnidirectional* memiliki berbagai macam hal yang dapat diteliti dan dikembangkan.

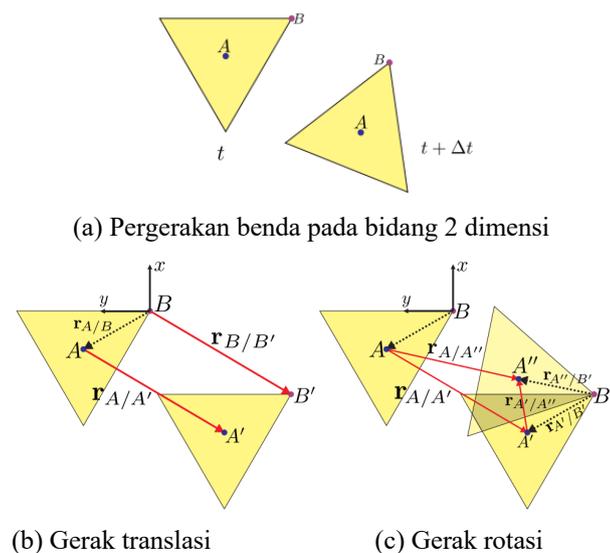
Penelitian – penelitian yang banyak dilakukan untuk robot beroda dengan roda *omnidirectional*, mayoritas menggunakan roda *omni-wheels* dan roda *mecanum-wheels*, dimana telah banyak diaplikasikan untuk kebutuhan industri, kesehatan, maupun membantu kegiatan manusia sehari – hari [11]. Penelitian pada robot beroda *omnidirectional* ini dilakukan dengan cara meneliti dan mengembangkan pemodelan kontrol kinematiknya, hal ini didukung karena pemodelan kontrol kinematik dari robot bergantung pada jumlah dan jenis dari roda penggeraknya (*omni-wheels* atau *mecanum-wheels*). Selain itu pengembangan – pengembangan kontrol yang dilakukan pada kontrol kinematiknya juga beragam, seperti memberikan kontrol PI [12] dan PID [13][14], ada juga yang menggunakan logika fuzzy terhadap kinematiknya [15]. Selain itu penelitian juga dilakukan dengan merubah desain robot, seperti jumlah roda penggeraknya [16][17], serta konstruksi tubuh robot [18]. Dari berbagai literatur yang ada, belum ada yang membahas tentang pemodelan kontrol kinematik untuk 6 buah roda *omni-wheels*.

Tujuan dari penelitian yang dilakukan ini adalah untuk mengembangkan pemodelan sistem kontrol kinematik untuk robot beroda dengan menggunakan 6 buah roda *omni-wheels*. Dimana kontrol akan diuji dengan menggunakan simulasi untuk mengetahui bagaimana respon sistemnya ketika robot tersebut bergerak baik terhadap target statik maupun target dinamik. Dimana pada makalah ini telah disusun secara urut pembahasannya adalah: dasar algoritma perpindahan dan pergerakan benda pada bidang 2 dimensi yang dibahas pada bab 2, pemodelan dari *invers* kinematic untuk kontrol robot serta pengaplikasian kontroler integral yang ada pada pada bab 3, Hasil simulasi pengujian respon sistem yang terbagi menjadi 2, yaitu respon terhadap target posisi tetap (statik), dan target target posisi yang berubah (dinamik) yang dibahas pada bab 4. Serta kesimpulan dan pandangan terhadap pengembangan penelitian lebih lanjut yang ada pada bab 5.

### 2. KAJIAN PUSTAKA

#### 2.1. Algoritma Perpindahan dan Pergerakan Benda Pada Bidang 2 Dimensi

Pemodelan dari kontrol kinematik robot sebenarnya menerapkan konsep dari algoritma gerak dan perpindahan benda pada bidang 2 dimensi, dimana pergerakan tersebut lebih kompleks dibandingkan dengan gerak sebuah titik atau partikel, karena pergerakan ini melibatkan tidak hanya translasi, namun juga gerak rotasi.



Gambar 1. Jenis pergerakan benda pada bidang 2 dimensi: translasi dan rotasi

Pergerakan benda yang dimaksudkan diilustrasikan seperti pada gambar 1(a), dimana perpindahan benda berbentuk segitiga tersebut melibatkan gerak translasi dan rotasi pada interval waktu ke-  $t$  hingga waktu ke-  $t + \Delta t$ . Pada bidang tersebut diibaratkan ada 2 titik yang ditempatkan posisinya tetap, yaitu titik A dan titik B. Jarak titik A dan titik B tidak pernah berubah dimana jarak tersebut adalah  $\mathbf{r}_{A/B}$ . Pada gambar 1(b) benda mengalami perpindahan, sehingga menyebabkan kedua titik yang ada pada benda tersebut berpindah, titik A berpindah ke titik A', dan titik B berpindah ke titik B'. Ketika benda tersebut bertranslasi, yang perlu digaris bawahi adalah jarak antara titik A dan titik B tidak berubah, sehingga hal ini berlaku juga ketika kedua titik itu berpindah sejauh  $\mathbf{r}_{A/A'}$  dan  $\mathbf{r}_{B/B'}$ , jarak antara A' dan B' tetap sama, sehingga didapatkan,

$$\mathbf{r}_{A/B} = \mathbf{r}_{A'/B'} \tag{1}$$

$$\mathbf{r}_{A/A'} = \mathbf{r}_{B/B'} \tag{2}$$

Selanjutnya pada gambar 1(c), bidang segitiga bergerak berputar (berotasi) terhadap salah satu titik yang menjadi acuan rotasi terhadap sumbu z, yaitu mengacu pada titik B' dengan kecepatan angular sebesar  $\boldsymbol{\omega}$ . Pada kondisi ini titik A' berpindah dari posisinya menuju ke posisi A'' sejauh  $\mathbf{r}_{A'/A''}$ . Sehingga berdasarkan gambar 1(c) didapatkan persamaan

$$\mathbf{r}_{A/A''} = \mathbf{r}_{A/A'} + \mathbf{r}_{A'/A''} \tag{3}$$

Bila persamaan (3) diturunkan terhadap waktu maka akan didapatkan

$$\frac{d\mathbf{r}_{A/A''}}{dt} = \frac{d\mathbf{r}_{A/A'}}{dt} + \frac{d\mathbf{r}_{A'/A''}}{dt} \tag{4}$$

$$\mathbf{v}_{A/A''} = \mathbf{v}_{A/A'} + \mathbf{v}_{A'/A''} \tag{5}$$

Dimana persamaan (5) merupakan persamaan kecepatan perubahan titik manapun yang ada pada bidang segitiga tersebut.  $\mathbf{v}_{A/A''}$  dan  $\mathbf{v}_{A/A'}$  mewakili kecepatan perubahan titik A, baik berpindah ke posisi A' maupun ke posisi A''. Sedangkan  $\mathbf{v}_{A'/A''}$  menunjukkan kecepatan relatif perpindahan titik A' ke titik A'' dengan referensi sumbu putar ada pada titik B', sehingga persamaan (5) dapat dipecah menjadi

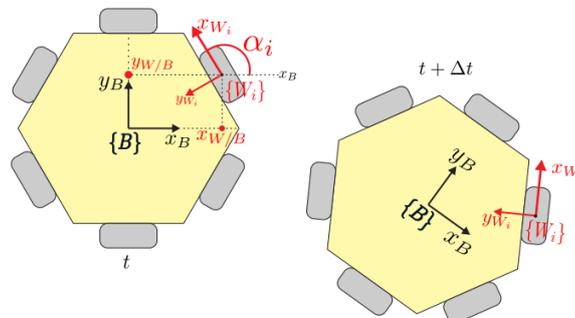
$$\mathbf{v}_{A/A''} = \mathbf{v}_{A/A'} + \boldsymbol{\omega} \times \mathbf{r}_{A'/B'} \tag{6}$$

Serta berdasarkan sifat pada persamaan (1) maka akan didapatkan nilai  $\mathbf{r}_{A''/B'} = \mathbf{r}_{A'/B'}$ .

## 2.2 Pemodelan Kinematik Robot Berdasarkan pada Algoritma Perpindahan benda terhadap bidang 2 dimensi

Sebelum membahas inti dari sub bab ini, berikut beberapa keterangan terkait variabel – variabel yang digunakan untuk mewakili beberapa parameter input yang dibutuhkan,

- $\mathbf{r} = (x, y, z)^T$  - mewakili vektor posisi koordinat terhadap sumbu koordinat  $x, y, z$ .
- $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)^T$  - mewakili vektor sudut angular terhadap sumbu koordinat  $x, y, z$ .
- $\mathbf{x} = (\mathbf{r}, \boldsymbol{\theta})^T$  - menunjukkan vektor posisi dan arah hadap robot
- $\mathbf{v} = (v_x, v_y, v_z)^T$  - menunjukkan vektor kecepatan translasi terhadap sumbu  $x, y$ , dan  $z$ .
- $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$  - menunjukkan vektor kecepatan angular terhadap sumbu  $x, y$ , dan  $z$ .
- $\boldsymbol{\xi} = (\mathbf{v}, \boldsymbol{\omega})^T$  - mewakili vektor kecepatan (translasi dan angular).



Gambar 2. Perpindahan posisi sebuah robot secara utuh

Gambar 2 menunjukkan perpindahan tubuh robot secara keseluruhan. Dimana titik tengah robot digunakan sebagai acuan, yang disimbolkan dengan  $\{B\}$ . Untuk roda ke- $i$  yang menempel pada robot disimbolkan sebagai  $\{W_i\}$  dimana titik tengah dari roda tersebut memiliki posisi  $(x_{W/B}, y_{W/B})^T$  terhadap titik tengah robot. Berdasarkan sifat dan persamaan yang telah dibahas pada sub bab 2.1, kecepatan roda ke- $i$  terhadap badan robot yaitu  $\mathbf{v}_{W/B} = (\dot{x}_{W/B}, \dot{y}_{W/B}, \dot{z}_{W/B})^T$  dapat dituliskan sebagai,

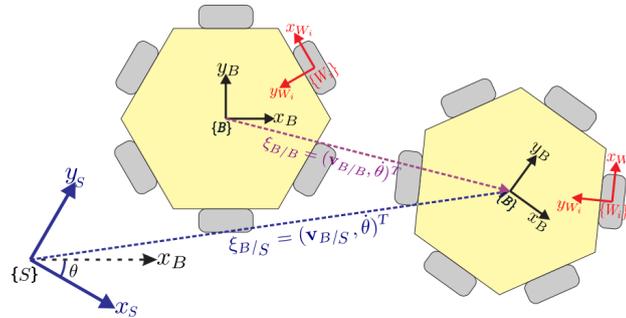
$$\mathbf{v}_{W/B} = \mathbf{v}_{B/B} + \boldsymbol{\omega} \times \mathbf{r}_{W/B} \tag{7}$$

dengan  $\mathbf{v}_{B/B} = (\dot{x}_{B/B}, \dot{y}_{B/B}, \dot{z}_{B/B})^T$  yang menunjukkan kecepatan badan robot terhadap frame robot itu sendiri ( $\{B\}$ ), sedangkan hasil perkalian matriks ruas sebelah kanan menghasilkan

$$\boldsymbol{\omega} \times \mathbf{r}_{W/B} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \omega_x & \omega_y & \omega_z \\ x_i & y_i & z_i \end{vmatrix} \tag{8}$$

Sementara itu pada kasus ini, yaitu robot yang hanya bergerak pada 2 dimensi saja hanya memiliki kecepatan angular terhadap sumbu  $z$  saja, sehingga  $\boldsymbol{\omega} = (0, 0, \omega_z)^T$ . Serta diasumsikan bahwa robot tidak mampu bertranslasi terhadap sumbu  $z$ , maka persamaan (7) dapat disederhanakan menjadi

$$\begin{pmatrix} \dot{x}_{W/B} \\ \dot{y}_{W/B} \end{pmatrix} = \begin{pmatrix} \dot{x}_{B/B} \\ \dot{y}_{B/B} \end{pmatrix} + \begin{pmatrix} -y_{W/B} \omega_z \\ x_{W/B} \omega_z \end{pmatrix} \tag{9}$$



Gambar 3. Body twist velocity dan spatial twist velocity

Selanjutnya mari kita lihat pergerakan keseluruhan dari tubuh robot terhadap lingkungan atau titik acuan globalnya, yang dinotasikan sebagai  $\{S\}$  seperti pada gambar 3, sehingga untuk dapat mendapatkan persamaan kecepatan robot terhadap frame global ( $\{S\}$ ), kecepatan robot terhadap tubuh robot harus dikalikan dengan matriks rotasi  $\mathbf{H}_{B/S}$

$$\boldsymbol{\xi}_{B/S} = \mathbf{H}_{B/S} \boldsymbol{\xi}_{B/B} \tag{10}$$

$$= \begin{pmatrix} \mathbf{R}_{B/S} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{B/B} \\ \omega_z \end{pmatrix} \tag{11}$$

$$= \underbrace{\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{spatial twist velocity}} \begin{pmatrix} \dot{x}_{B/B} \\ \dot{y}_{B/B} \\ \omega_z \end{pmatrix} \tag{12}$$

dengan  $\mathbf{R}_{B/S}$  merupakan matriks rotasi 2 dimensi,  $\theta$  adalah sudut yang terbentuk antara sumbu  $x$  frame global ( $\{S\}$ ) dan sumbu  $x$  frame robot ( $\{B\}$ ), dan  $\omega_z$  adalah kecepatan angular terhadap sumbu  $z$ . Perlu diketahui bahwa semua titik dan bagian yang ada pada tubuh robot memiliki kecepatan angular yang sama ketika robot berputar yaitu  $\dot{\theta}$ . Banyak literatur untuk bidang robotik yang menyebut  $\boldsymbol{\xi}_{B/S}$  adalah *spatial twist velocity*. Tujuan dari pemodelan algoritma kontrol kinematik robot adalah untuk mendapatkan persamaan invers kinematiknya, dimana dari persamaan invers kinematiknya akan didapatkan kecepatan putar yang dibutuhkan untuk masing – masing roda penggerak robot. Sehingga untuk mendapatkan invers kinematiknya persamaan (12) ditulis sebagai berikut

$$\boldsymbol{\xi}_{B/B} = \mathbf{H}_{B/S}^{-1} \boldsymbol{\xi}_{B/S} \tag{13}$$

$$= \begin{pmatrix} \mathbf{R}_{B/S}^{-1} & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{B/S} \\ \dot{\theta} \end{pmatrix} \quad (14)$$

$$= \underbrace{\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}}_{\text{body twist velocity}} \quad (15)$$

Persamaan (18) juga dikenal sebagai persamaan *body twist velocity*, atau merupakan kecepatan absolut yang telah dibahas pada sub-bab sebelumnya. Selain itu,  $\mathbf{R}_{B/S}^{-1} \mathbf{v}_{B/S}$  merupakan kecepatan absolut dari robot terhadap frame tubuhnya sendiri yaitu  $\mathbf{v}_{B/B}$ . Sehingga dapat disubstitusikan persamaan tersebut pada persamaan (7), maka akan didapatkan

$$\mathbf{v}_{W/B} = \mathbf{R}_{B/S}^{-1} \mathbf{v}_{B/S} + \boldsymbol{\omega} \times \mathbf{r}_{W/B} \quad (16)$$

$$= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} + \begin{pmatrix} -y_{W_i/B} \omega_z \\ x_{W_i/B} \omega_z \end{pmatrix} \quad (17)$$

Persamaan (17) bisa dituliskan juga sebagai berikut

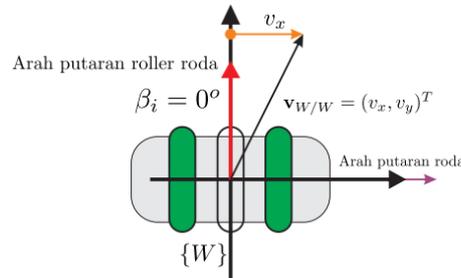
$$\mathbf{v}_{W/B} = \begin{pmatrix} 1 & 0 & -y_i \\ 0 & 1 & x_i \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \quad (18)$$

Mengambil ide seperti pada persamaan (13), kecepatan absolut roda robot dapat dihitung dengan cara

$$\mathbf{v}_{W/W} = \mathbf{R}_{W/B}^{-1} \mathbf{v}_{W/B} \quad (19)$$

$$= \begin{pmatrix} \cos(\alpha_i) & \sin(\alpha_i) \\ -\sin(\alpha_i) & \cos(\alpha_i) \end{pmatrix} \begin{pmatrix} 1 & 0 & -y_i \\ 0 & 1 & x_i \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \quad (20)$$

Pada persamaan yang telah didapatkan, vektor *spatial twist velocity* yaitu  $(\dot{x}, \dot{y}, \dot{\theta})^T$ , dapat digunakan untuk mencari nilai kecepatan roda  $\mathbf{v}_{W/W}$ , yang memiliki elemen  $(v_x, v_y)$ .



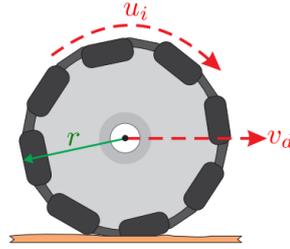
Gambar 4. Roda omni-wheels

Selanjutnya yang dilakukan dalam membuat pemodelan kontrol kinematik robot ini, adalah mencari persamaan antara kecepatan roda  $\mathbf{v}_{W/W}$  dengan sinyal kontrol kecepatan angular  $u_i$  untuk tiap rodanya. Jenis roda *omnidirectional* yang digunakan adalah *omni-wheels*. Dimana roda jenis ini memiliki kemampuan untuk bergerak *sliding* kesamping, karena mekanisme roda yang dilengkapi roler yang terpasang dengan sudut  $\beta$ . Sudut  $\beta$  merupakan sudut yang terbentuk antara arah putar roler dengan sumbu  $y$  dari frame roda. Kecepatan perpindahan roda dihasilkan oleh kecepatan putar robot roda itu sendiri dapat dituliskan dengan

$$v_d = v_x + v_y \tan \beta_i \quad (21)$$

$$= (1 \tan \beta_i) \begin{pmatrix} v_x \\ v_y \end{pmatrix} \quad (22)$$

Seperti yang diketahui, pada umumnya kecepatan translasi  $v_d$  roda adalah hasil perhitungan dari kecepatan angular roda  $u_i$  dengan jari – jari roda  $r$ , yaitu



Gambar 5. Kecepatan angular dan tranlasional dari putaran roda

$$u_i = \frac{v_d}{r} \tag{23}$$

Kecepatan translasi dan rotasi dari roda dapat dilihat seperti pada gambar 5, dimana semua roda yang digunakan pada robot jenisnya sama. Dalam bentuk matriksnya  $u_i$  dapat dituliskan dengan

$$u_i = \begin{pmatrix} \frac{1}{r} \frac{\tan(\beta_i)}{r} \\ \frac{v_x}{v_y} \end{pmatrix} \tag{24}$$

$$= \begin{pmatrix} \frac{1}{r} \frac{\tan(\beta_i)}{r} \end{pmatrix} \mathbf{v}_{W/W} \tag{25}$$

Kemudian substitusikan persamaan (25) ke persamaan (20), dimana persamaan ini merupakan persamaan kontrol kecepatan angular dari roda ke-  $i$  dari robot yang bisa dituliskan sebagai fungsi  $\mathbf{g}_i(\theta)$ , yaitu

$$u_i = \underbrace{\begin{pmatrix} \frac{1}{r} \frac{\tan(\beta_i)}{r} \end{pmatrix} \begin{pmatrix} \cos(\alpha_i) & \sin(\alpha_i) \\ -\sin(\alpha_i) & \cos(\alpha_i) \end{pmatrix} \begin{pmatrix} 1 & 0 & -y_i \\ 0 & 1 & x_i \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{g}_i(\theta)} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \tag{26}$$

Jika nilai  $\mathbf{g}_i(\theta)$  disederhanakan, maka akan didapatkan

$$\mathbf{g}_i(\theta) = \frac{1}{r} \begin{pmatrix} -\sin(\alpha_i + \theta) \tan(\beta_i) + \cos(\alpha_i + \theta) \\ \sin(\alpha_i + \theta) + \cos(\alpha_i + \theta) \tan(\beta_i) \\ x_i(\sin(\alpha_i) + \cos(\alpha_i) \tan(\beta_i)) + y_i(\sin(\alpha_i) \tan(\beta_i) - \cos(\alpha_i)) \end{pmatrix}^T \tag{27}$$

dan seperti yang sudah diketahui pada gambar 4 yaitu nilai dari sudut  $\beta_i$  untuk roda ke-  $i$  dengan jenis omni-wheels nilainya adalah  $0^\circ$ , sehingga didapatkan nilai  $\tan(0) = 0$ , maka persamaan (27) dapat disederhanakan menjadi

$$\mathbf{g}_i(\theta) = \frac{1}{r} \begin{pmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ x_i \sin(\alpha_i) - y_i \cos(\alpha_i) \end{pmatrix}^T \tag{28}$$

dimana kontrol  $\mathbf{g}_i(\theta)$  merupakan persamaan untuk mengontrol roda ke-  $i$  saja, sehingga untuk persamaan yang digunakan untuk mengontrol semua roda didapatkan sebuah matriks invers jacobian yang disimbolkan dengan  $\mathbf{G}(\theta) \in \mathbb{R}^{n \times 3}$ .

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} \mathbf{g}_1(\theta) \\ \mathbf{g}_2(\theta) \\ \vdots \\ \mathbf{g}_n(\theta) \end{pmatrix} \tag{29}$$

$$\mathbf{u} = \mathbf{G}(\theta) \boldsymbol{\xi}_{B/S} \tag{30}$$

Persamaan (30) dikenal sebagai *invers velocity kinematic* atau kinematik balik kecepatan yang menunjukkan kontrol pada robot dengan cara mengontrol tiap roda robot yaitu dengan menggunakan sinyal kontrol vektor  $\mathbf{u}$ . Untuk tujuan simulasi dan analisa respon kinematik dengan pemodelan state-space, persamaan invers dari persamaan (30) sangat dibutuhkan, yaitu

$$\boldsymbol{\xi}_{B/S} = \mathbf{G}^\dagger(\theta) \mathbf{u} \tag{31}$$

dimana  $\mathbf{G}^\dagger = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1}$  merupakan hasil dari pseudo invers dari  $\mathbf{G}^\dagger(\theta)$ . Sehingga sekarang bisa kita verifikasi hasil pemodelan kontrol kinematik pada persamaan (30). Diketahui bahwa persamaan umum *state-space* dapat dituliskan sebagai

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{32}$$

### 2.3 Algoritma kontrol robot terhadap posisi tujuan robot

Permasalahan yang dihadapi dalam mengembangkan algoritma kontrol adalah mencari data feedback yang dapat memberikan nilai error.

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}^*(t) \tag{33}$$

serta kontrol yang didesain untuk diaplikasikan pada robot adalah untuk dapat mengurangi nilai error yang dihasilkan dengan bentuk atau sifat eksponensial, maka dapat dituliskan sebagai berikut

$$\frac{d\mathbf{e}(t)}{dt} = -\lambda\mathbf{e}(t) \tag{34}$$

dimana  $\lambda$  merupakan nilai konstanta bernilai positif yang digunakan untuk mengendalikan kecepatan penurunan eksponensial errornya. Dengan menurunkan persamaan (33) terhadap waktu ( $t$ ) maka didapatkan

$$\frac{d\mathbf{e}(t)}{dt} = \frac{d\mathbf{x}(t)}{dt} - \frac{d\mathbf{x}^*(t)}{dt} \tag{35}$$

dapat disubstitusikan persamaan (33) dan (34) ke persamaan (35), menjadi

$$\frac{d\mathbf{x}(t)}{dt} = -\frac{d\mathbf{e}(t)}{dt} + \frac{d\mathbf{x}^*(t)}{dt} \tag{36}$$

$$\dot{\mathbf{x}} = \lambda(\mathbf{x}^*(t) - \mathbf{x}(t)) + \dot{\mathbf{x}}^*(t) \tag{37}$$

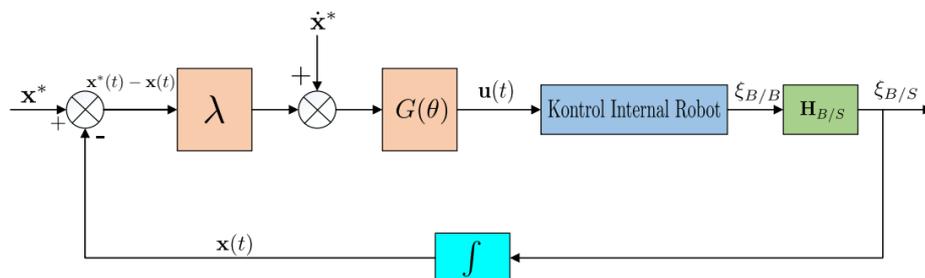
dapat diketahui dari sebelumnya bahwa  $\dot{\mathbf{x}}$  menunjukkan nilai dari kecepatan spasial robot  $\xi_{B/S}$ . Dari persamaan – persamaan yang telah didapat sebelumnya didapatkan bentuk terakhir dari persamaan untuk mengontrol robot dengan menggunakan roda omnidirectional dengan mensubstitusikan persamaan (37) ke persamaan (30), menjadi

$$\mathbf{u} = \mathbf{G}(\theta)(\lambda(\mathbf{x}^*(t) - \mathbf{x}(t)) + \dot{\mathbf{x}}^*(t)) \tag{38}$$

Hasil respon dari control yang telah dimodelkan dapat dilihat pada bab 4 dari pembahasan makalah ini, serta pengaplikasian realnya adalah untuk menjaga kecepatan dari masing – masing motor penggerak robot agar robot dapat bergerak secara linier ke posisi tujuan.

## 3. METODE PENELITIAN

### 3.1 Desain Kontrol yang diaplikasikan

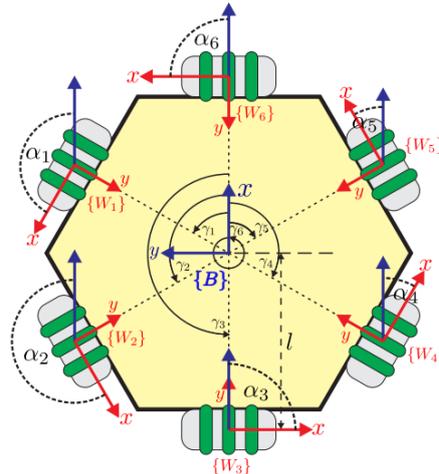


Gambar 6. Desain sistem kontrol

Sistem kontrol yang diaplikasikan akan disimulasikan dengan menggunakan program yang menggunakan bahasa pemrograman python, dimana digunakan untuk menguji kontroler yang bersifat reaktif terhadap besarnya error yang diperoleh oleh kontroler seperti yang ditunjukkan pada gambar 6. Selain mendapatkan input berupa posisi tujuan, respon sistem juga akan diuji pada target yang bergerak (memiliki kecepatan perpindahan), sehingga akan teruji bagaimana respon sistem kontrol yang sudah didesain untuk tetap menjaga kestabilannya dalam bergerak menuju titik koordinat yang juga bergerak.

### 3.2 Desain Mekanis robot yang diteliti

Mekanisme robot yang diteliti adalah robot menggunakan 6 buah roda omnidirectional omni-wheels, dimana dipasangkan pada robot sesuai pada gambar 7.



Gambar 7. Desain mekanis robot yang diteliti

Arah hadap dari robot sendiri adalah menghadap ke arah roda robot nomor 6. Sehingga dari spesifikasi yang telah digambarkan pada gambar 7, didapatkan beberapa nilai – nilai yang dapat diinputkan pada parameter persamaan/pemodelan kontrol dari robot, yaitu:

- Sudut yang dibentuk antara sumbu  $x$  robot dengan sumbu  $x$  roda ( $\alpha_i$ )
- Sudut yang dibentuk antara sumbu  $x$  robot dengan titik berpotongannya antara sumbu  $y$  roda dengan titik tengah robot ( $\gamma_i$ )

Berikut nilai nilai yang didapatkan dari desain mekanis robot:

Roda Ke -	Sudut $\alpha_i$	Sudut $\gamma_i$
1	$5/6 \pi$	$1/3 \pi$
2	$-5/6 \pi$	$2/3 \pi$
3	$-1/2 \pi$	$\Pi$
4	$-1/6 \pi$	$-2/3 \pi$
5	$1/6 \pi$	$-1/3 \pi$
6	$1/2 \pi$	$2\pi$

Tabel 1. Spesifikasi sudut pada mekanis robot

Serta dikarenakan jenis roda yang digunakan adalah roda omniwheels, maka sudut  $\beta$  yang dimiliki pada masing – masing ban adalah  $0^\circ$ .

### 3.3 Pemodelan matematis sesuai dengan mekanisme robot

Untuk posisi koordinat roda robot terhadap body robot ( $x_i$  dan  $y_i$ ) dapat dicari dengan memasukkan nilai  $l$  yang diketahui melalui persamaan berikut

$$x_i = l \cos(\gamma_i) \tag{39}$$

$$y_i = l \sin(\gamma_i) \tag{40}$$

Sehingga akan didapatkan persamaannya

$$\mathbf{g}_i(\theta) = \frac{1}{r} \begin{pmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ l (\cos(\gamma_i) \sin(\alpha_i) - (\sin(\gamma_i) \cos(\alpha_i))) \end{pmatrix}^T \tag{41}$$

dengan memasukkan sudut – sudut yang ada pada tabel 1, maka akan didapatkan persamaan matematis untuk kontrolnya adalah sebagai berikut

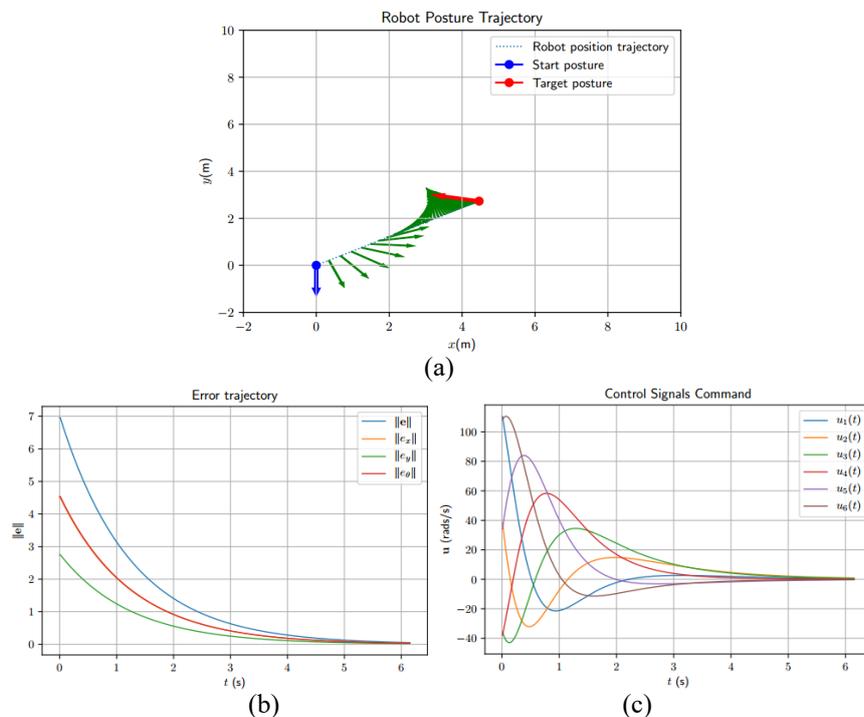
$$\mathbf{G}(\theta) = \frac{1}{r} \begin{pmatrix} \cos(\frac{5}{6}\pi) & \sin(\frac{5}{6}\pi) & l \\ \cos(-\frac{5}{6}\pi) & \sin(-\frac{5}{6}\pi) & l \\ \cos(-\frac{1}{2}\pi) & \sin(-\frac{1}{2}\pi) & l \\ \cos(-\frac{1}{6}\pi) & \sin(-\frac{1}{6}\pi) & l \\ \cos(\frac{1}{6}\pi) & \sin(\frac{1}{6}\pi) & l \\ \cos(\frac{1}{2}\pi) & \sin(\frac{1}{2}\pi) & l \end{pmatrix} \quad (42)$$

#### 4. HASIL DAN PEMBAHASAN

Pada bab ini akan ditunjukkan hasil simulasi respon control yang disajikan datanya dalam bentuk grafik, melalui simulasi program dengan menggunakan bahasa pemrograman python. Pengujian respon dilakukan dengan 2 cara, yaitu menguji respon control terhadap tujuan yang tetap (statis) serta menguji respon control terhadap tujuan yang bergerak (dinamik).

##### 4.1 Hasil dan Pembahasan simulasi dengan target statis (tidak bergerak)

Simulasi yang dilakukan dalam selang waktu 0 sampai 12 detik atau  $0 \leq t < 12$ , dengan sampling waktu  $\Delta t = 0.01s$ . Pada pengujian ini robot diletakkan pada posisi koordinat awal  $\mathbf{x}(0.0, 0.0, -90^\circ)$ , dan koordinat yang dituju adalah  $\mathbf{x}^*(4.5, 2.75, 170^\circ)$ .



Gambar 8. Hasil simulasi respon control dengan target statis

Berdasarkan hasil yang didapatkan seperti pada gambar 8, dapat dilihat pada bagian (a) robot bergerak dari sumbu koordinat awal 0.0, 0.0, dengan arah hadap  $-90^\circ$  (diwakili oleh arah panah biru), kearah posisi target tujuan yaitu 4.5, 2.75, dengan arah hadap akhir adalah  $170^\circ$  (diwakili arah panah merah), dengan proses perubahan posisi dan arah hadapnya diwakili dengan arah panah warna hijau. Serta hasil respon yang didapatkan dengan nilai penguatan  $\lambda = 0.8$ , menunjukkan dalam kurun waktu 6 detik robot dapat bergerak sehingga mampu menurunkan nilai error posisinya yang dapat dilihat pada gambar bagian (b) serta dapat dilihat pula dari respon kecepatan putar tiap motornya pada gambar bagian (c), dalam kurun waktu 6 detik dapat dilihat motor yang ada pada robot telah berhenti bergerak, yang dapat diasumsikan robot telah berhasil mencapai posisi tujuan. Jika dilihat dari nilai error akhir seperti pada gambar 8(c), hal ini menunjukkan kontrol kinematic telah berjalan sesuai dengan yang diinginkan dan bisa disimpulkan bahwa kontrol terhadap target statis memiliki hasil yang stabil dan bagus, hal ini terlihat dari nilai error akhir pada grafik, yaitu mendekati nilai 0, yang berarti robot mampu berhenti dititik tujuan.

### 4.2 Hasil dan Pembahasan simulasi dengan target bergerak

Pada pengujian ini, dikarenakan posisi target juga bergerak maka perlu diketahui  $\dot{\mathbf{x}}^*$ . Dimana dalam simulasi nilai tersebut mewakili kecepatan perpindahan sumbu x, sumbu y, dan arah hadap dari robot. Untuk nilai perpindahan x dan y tidak menemui masalah, tinggal menurunkan persamaan perpindahannya saja, namun yang menjadi masalah adalah perubahan kecepatan pada arah hadapnya ( $\dot{\theta}^*$ ), yaitu dengan persamaan

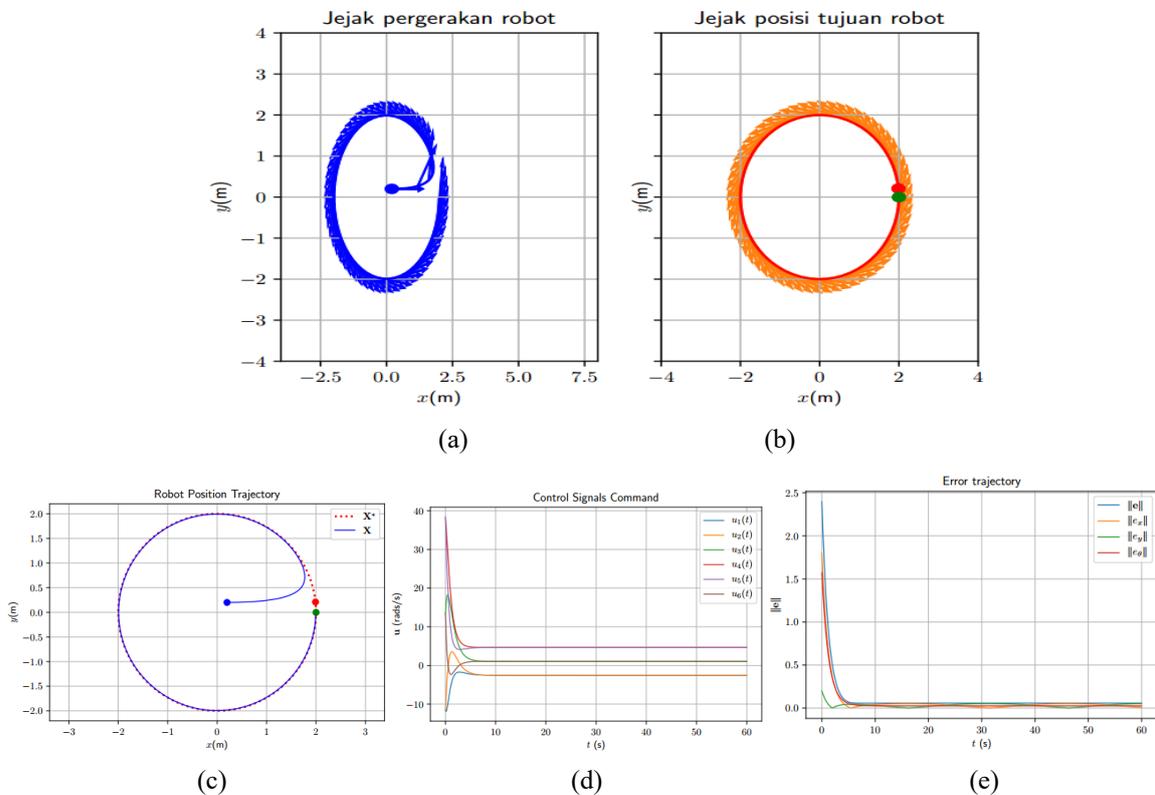
$$\theta^* = \arctan(\dot{y}^*(t), \dot{x}^*(t)) \tag{43}$$

Dimana untuk perubahan kecepatan arah hadapnya bisa didapatkan dengan cara menurunkan persamaan (43), sehingga didapatkan

$$\dot{\theta}^* = \frac{\partial \theta^*}{\partial \dot{x}^*} \frac{d\dot{x}^*}{dt} + \frac{\partial \theta^*}{\partial \dot{y}^*} \frac{d\dot{y}^*}{dt} \tag{44}$$

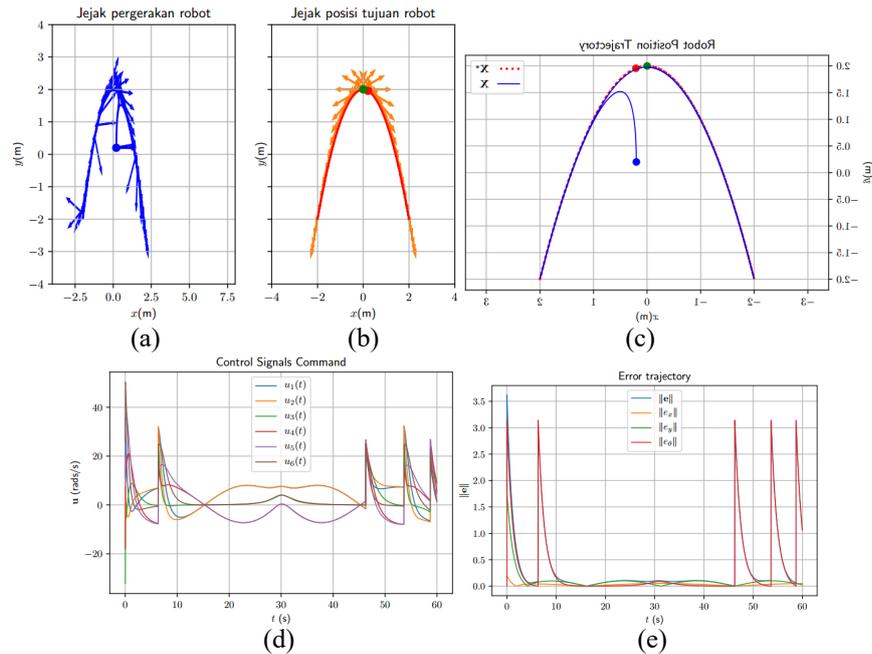
$$= \frac{\dot{x}^* \dot{y}^* - \dot{y}^* \dot{x}^*}{\dot{x}^{*2} + \dot{y}^{*2}} \tag{45}$$

selanjutnya adalah mensimulasikan respon dengan memberikan nilai  $\dot{\mathbf{x}}^*$ -nya. Berikut hasil pengujiannya dengan membentuk pola melingkar serta membentuk pola gelombang sinus.



Gambar 9 hasil simulasi respon dengan target bergerak

Berdasarkan hasil yang ditunjukkan pada gambar 9, pada bagian (a) dapat diketahui pola yang dibentuk berdasarkan gerakan dari robot, pada (b) adalah pola yang dibentuk oleh target yang bergerak, dimulai dari titik merah dan diakhiri pada titik hijau, pada bagian (c) dapat dilihat perbandingan posisi target yang bergerak dengan posisi dari pergerakan robot, dapat terlihat bahwa respon robot mampu mengikuti pola yang dibuat oleh posisi target yang bergerak, pada (d) dapat dilihat respon sinyal kontrol untuk masing – masing motor penggerak robot dan pada (e) dapat dilihat perubahan nilai error yang terjadi, berdasarkan hasil yang didapat robot mampu mempertahankan nilai error mendekati nilai 0. Dari gambar 9, dapat disimpulkan pada pengujian ini kontrol robot yang dimodelkan mampu bekerja dengan baik, karena berdasarkan gambar 9 bagian (a) terlihat pola atau jejak pergerakan robot mampu bergerak membentuk pola lingkaran, sesuai seperti pola gerak yang diinginkan yang ada pada gambar 9 (b). Selain itu dapat dilihat bahwa selama simulasi pergerakan robot, nilai error posisi yang ditunjukkan pada gambar 9 (e), nilai error selalu bernilai mendekati 0, hal ini menunjukkan bahwa respon robot untuk bergerak sesuai pola posisi yang diinginkan bekerja dengan optimal.



Gambar 10. Hasil simulasi respon dengan target bergerak membentuk gelombang setengah sinus

Hasil yang juga masih bagus didapatkan dengan merubah pola dari pergerakan posisi tujuan yang akan dituju oleh robot, pada pengujian ketiga diuji target digerakkan dengan membentuk pola setengah gelombang sinus, dapat dilihat pada gambar 10, hasil respon dari robot menggambarkan bahwa robot masih dapat mengikuti pola yang diinginkan, terlihat sinyal respon error dapat turun mendekati 0, meskipun beberapa kali sinyal error arah hadap robot naik nilainya, namun robot kembali dapat menurunkan nilai errornya untuk dapat mendekati nilai 0. Dari gambar 10, dapat disimpulkan pada pengujian ini kontrol robot yang dimodelkan mampu bekerja dengan baik, karena berdasarkan gambar 10 bagian (a) terlihat pola atau jejak pergerakan robot mampu bergerak membentuk pola setengah gelombang sinus, sesuai seperti pola gerak yang diinginkan yang ada pada gambar 10 (b). Selain itu dapat dilihat bahwa selama simulasi pergerakan robot, nilai error posisi yang ditunjukkan pada gambar 10 (e), nilai error selalu bernilai mendekati 0, hal ini menunjukkan bahwa respon robot untuk bergerak sesuai pola posisi yang diinginkan bekerja dengan optimal. Sehingga berdasarkan kedua percobaan simulasi pergerakan robot terhadap pola target yang bergerak, memiliki hasil akhir yang bagus, karena nilai error yang dihasilkan baik pengujian pola lingkaran ataupun pola setengah gelombang sinus mampu bertahan mendekati nilai 0, yang berarti kontrol robot mampu membuat robot bergerak mengejar target yang bergerak untuk membuat pola yang diinginkan.

## 5. KESIMPULAN

Dapat disimpulkan bahwa persamaan yang telah dimodelkan dapat menghasilkan respon sinyal yang bagus, baik untuk target pergerakan robot yang statis maupun target yang bergerak. Serta dapat dibuktikan secara simulasi bahwa pemodelan kontrol yang telah dibuat dapat menghasilkan sinyal respon yang bagus. Hal ini dapat dilihat dari hasil simulasi yang disajikan dalam bentuk gambar grafik, dimana untuk target statis, respon robot berjalan dengan bagus, terlihat dari nilai error yang semakin menurun setiap waktunya, dan pada hasil akhirnya, dalam waktu 6 detik, robot telah sampai atau berhenti pada posisi target yang diinginkan, terlihat pada saat  $t = 6s$ , nilai error bernilai 0. Kemudian untuk hasil simulasi target bergerak membentuk pola lingkaran dan setengah gelombang sinus, dalam kurun waktu pengujian selama 1 menit, robot mampu bergerak sesuai dengan pola yang diinginkan, terlihat dari perubahan nilai error selama pengujian yaitu pada  $t = 15s$  hingga  $t = 60s$ , nilai error dapat terjaga mendekati nilai 0. Hal tersebut telah menunjukkan bahwa pemodelan kontrol kinematik yang dibuat mampu membuat robot beroda 6 *omniwheels* bergerak secara optimal sesuai dengan yang diinginkan.

## 6. DAFTAR PUSTAKA

- [1] Y. Yi, F. Mengyin, S. Changsheng, W. Meiling and Z. Cheng, "Control Law Design of Mobile Robot Trajectory Tracking and Development of Simulation Platform," *2007 Chinese Control Conference*, Hunan, 2007, pp. 198-202

- [2] J. Wang, B. Wei, Y. Zhang, and H. Chen, "Design of an autonomous mobile robot for hospital," in *2009 IEEE International Symposium on IT in Medicine Education*, vol. 1, 2009, pp. 1181–1186.
- [3] T. Shu, S. Gharaaty, W. Xie, A. Joubair and I. A. Bonev, "Dynamic Path Tracking of Industrial Robots With High Accuracy Using Photogrammetry Sensor," in *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1159-1170, June 2018.
- [4] S. Chen and J. Yu, "Underwater cave search and entry using a robotic fish with embedded vision," *Proceedings of the 33rd Chinese Control Conference*, Nanjing, 2014, pp. 8335-8340.
- [5] F. Niroui, K. Zhang, Z. Kashino and G. Nejat, "Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610-617, April 2019.
- [6] P. Devid, P. Bambang Eka, W. Indah Uly, "Implementasi Model Robot Edukasi Menggunakan Mikrokontroler Atmega8 Untuk Robot Pemadam Api," in *IPI Indonesian Journal on Networking and Security (IJNS)*, Surakarta, 2013.
- [7] C. Chen, S. Wang, L. Li, S. Ke, C. Wang and X. Bu, "Intelligent Covert Satellite Communication for Military Robot Swarm," in *IEEE Access*, vol. 8, pp. 5363-5382, 2020.
- [8] Chaidir, Ali Rizal; Anam, Khairul; Rahardi, Gamma Aditya. "Lane Tracking pada Robot Beroda Holonomic menggunakan Pengolahan Citra." *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, [S.l.], v. 8, n. 1, p. 69, jan. 2020. ISSN 2459-9638.
- [9] K. Shojaei and A. M. Shahri, "Adaptive robust time-varying control of uncertain non-holonomic robotic systems," in *IET Control Theory & Applications*, vol. 6, no. 1, pp. 90-102, 5 January 2012.
- [10] Xueshan Gao, Kejie Li, Yan Wang, Guangliang Men, Dawei Zhou and Koki Kikuchi, "A floor cleaning robot using Swedish wheels," *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, 2007, pp. 2069-2073.
- [11] T. Kumbasar and H. Hagra, "A Self-Tuning zSlices-Based General Type-2 Fuzzy PI Controller," in *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 991-1013, Aug. 2015.
- [12] H. Talebi Abatari and A. Dehghani Tafti, "Using a fuzzy PID controller for the path following of a car-like mobile robot," *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, Tehran, 2013, pp. 189-193.
- [13] C. H. KAZANCI and A. F. KOCAMAZ, "PID Optimization on Differential Drive Mobile Robot," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018, pp. 1-6.
- [14] H. Mehrjerdi, M. Saad and J. Ghommam, "Hierarchical Fuzzy Cooperative Control and Path Following for a Team of Mobile Robots," in *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 907-917, Oct. 2011.
- [15] T. Sun, Y. Zhang, H. Zhang, P. Wang, Y. Zhao and G. Liu, "Three-wheel Driven Omnidirectional Reconfigurable Conveyor Belt Design," *2019 Chinese Automation Congress (CAC)*, Hangzhou, China, 2019, pp. 101-105.
- [16] Suwanda, Ivan. "Rancang Bangun Robot Omni Wheel Penyedot Debu Menggunakan Sensor Accelerometer Berbasis Mikrokontroler ATmega16." *Jurnal Teknik Elektro Universitas Tanjungpura*, vol. 2, no. 1, 2014..
- [17] Jaishree, M. & Keerthana, P. & Lakshmisree, P.S. & S J, Jebasingh. (2018). Design and implementation of omni-wheel robotic system under automatic height control and adaptation. *International Journal of Pure and Applied Mathematics*. 119. 1625-1634.
- [18] E. Kaneko and N. Umez, "Rapid construction of coarse indoor map for mobile robots," *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, Nagoya, 2017, pp. 1-3, doi: 10.1109/GCCE.2017.8229367.