A General Inverse Kinematic Formulation and Control Schemes for Omnidirectional Robots

Indrazno Siradjuddin, Gilang Al Azhar, Sapto Wibowo, Ferdian Ronilaya, Cahya Rahmad, and Erfan Rohadi

***Abstract*—An omnidirectional mobile robot configuration is a promising mobile robot technology in the future, since it has holonomic properties, where the motion constraints is only limited in the robot actuator space. In the two dimensional task space, the omnidirectional robot is able to move in any configuration. This omnidirectional motion ability is primarily caused by the mechanism of the robot wheels. Two most commonly used wheel technologies are omni and mecanum wheels. Indeed, many literatures and studies are available in developing the kinematic model, however the presented kinematic models were developed based on the specific cases. For control engineers, this may be the issue where the kinematic development has to be formulated from scratch to meet their own robot specification. Therefore, it is important to have a general framework for developing the omnidirectional kinematic model. This work presents a generic kinematic formulation to model omnidirectional mobile robot using omni and mecanum wheel types. The formulation can be used for any number of robot wheel configuration. In this paper, three omnidirectional robot platform configurations: three omni-wheels robot, four mecanum-wheels robot and six omni-wheels robot have been chosen for discussion to demonstrate the developed generic kinematic formulation. Additionally, this work also proposes two control schemes for controlling the robot motion: an exponential decreased error tracking algorithm (so called as the model based control scheme) and a Proportional-Integral (P-I) control scheme. The state space formulation has also been exposed to validate the controllability of the kinematic control system. The performances of both control schemes have been simulated and analysed for two cases of robot tracking application: a static and a moving target. The performances have been measured in terms of robot posture in two dimensional space, robot control signals and the error signals trajectories. The controller gains have been manually chosen by trial and error. The simulation results have demonstrated satisfactory performances of the developed generic kinematic formulation and control schemes. The derived control schemes guarantees an exponential decrease of the error.**

***Index Terms*—Omnidirectional Mobile Robots, Kinematic Control, Mecanum Wheeled Robot, Omni Wheeled Robot.**

I. Introduction

MOBILE robotics has received more attention in scientific research, due their broad application in many fields. Aplication ranges from education [1], healthcare [2], industries [3], military [4], and space exploration [5]. The robot motion mechanism and its motion constraints are subject to holonomic (omnidirectional) and nonholonomic systems studies [6]. Omnidirectional do not have constraint in the robot velocity domain while nonholonomic constraints impede certain robot velocities, or in another word the robot velocity can only be directed in the direction of the wheel rotation. For instance, a differential drive robot [7], [8], [9], tricycle robot [10] or a car-like robot [11] cannot be moved sideways. Therefore, omnidirectional robot mechanism is a promising mobile robot technology in the future due to its motion agility in task space and its simplicity to control the robot motion. Recently, in [12], a multiple terrains robot utilising omni wheel platform has been proposed. The motion agility of the omnidirectional mobile robot system can be very useful to any grid-based (discrete) path planning where sideways and diagonal motions are required for the robot to travel from one grid to another [13].

Most common techniques to design an omnidirectional robot is to utilise omnidirectional wheel or mecanum wheel which is also known as the Swedish wheel [14], [15], which have been patented in [16], [17]. A variety of omnidirectionality has been used in [18]. The fundamental work which has to be done at first before the robot controller can be designed is the robot kinematic formulation [19]. Many literatures can be studied to know the state-of-art of omnidirectional robot navigation and motion control. In [20], the kinematics and dynamics of four omni-wheels robot have been discussed, where the fuzzy control algorithm was proposed to control the robot motion. Using the same idea of utilising fuzzy algorithm, fuzzy controllers for an omnidirectional mobile robot have been presented in [21] and [22]. Another omnidirectional kinematics control laws have been proposed and demonstrated where optimal control and PID were used [23], [24], [25], [26], [27]. Despite, rich literature studies of the modelling and control of omnidirectional robots from the research cited, an explicit generic omnidirectional kinematic formulation has not been presented according to the best of authors knowledge.

The main objective of our presentation and contribution is to develop a generic inverse kinematic control algorithm for omnidirectional mobile robots where most used wheel types including omni and mecanum are considered. The intention of creating the proposed inverse kinematic control algorithm

Manuscript received June 04, 2020; revised June 29, 2021. This work is supported by the Ministry of Education and Culture of Indonesia through fundamental research project scheme, contract number: 145/SP2H/LT/DRPM/2020.

I. Siradjuddin is an Associate Professor in the Department of Electrical Engineering, State Polytechnic of Malang, Indonesia, 65141 (Phone: +62-341-404424, Fax: +62-341-404425, e-mail: indrazno@polinema.ac.id).

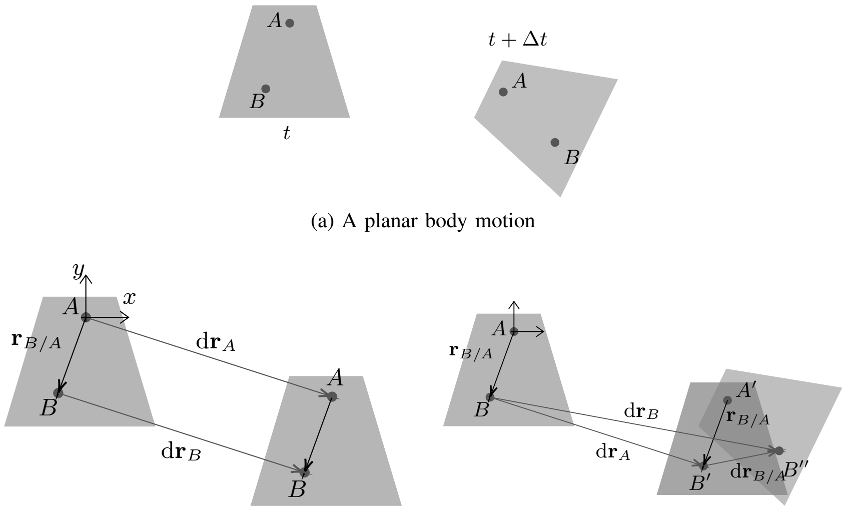
G.A. Azhar is a Lecturer in the Department of Electrical Engineering, State Polytechnic of Malang, Indonesia, 65141 (Phone: +62-341-404424, Fax: +62-341-404425, e-mail: gillang_al_azhar@polinema.ac.id).

S. Wibowo is a Senior Lecturer in the Department of Electrical Engineering, State Polytechnic of Malang, Indonesia, 65141 (Phone: +62-341-404424, Fax: +62-341-404425, e-mail: sapto.wibowo@polinema.ac.id).

F. Ronilaya is a Senior Lecturer in the Department of Electrical Engineering, State Polytechnic of Malang, Indonesia, 65141 (Phone: +62-341-404424, Fax: +62-341-404425, e-mail: ferdian@polinema.ac.id)

C. Rahmad is an Associate Professor in the Department of Information Technology, State Polytechnic of Malang, Indonesia, 65141 (Phone: +62-341-404424, Fax: +62-341-404425, e-mail: cahya.rahmad@polinema.ac.id)

E. Rohadi is a Senior Lecturer in the Department of Information Technology, State Polytechnic of Malang, Indonesia, 65141 (Phone: +62-341-404424, Fax: +62-341-404425, e-mail: erfanr@polinema.ac.id)



(a) A planar body motion

(b) Translation

(c) Rotation

Fig. 1: Steps of a planar body motion: translation and rotation

is to further improve the reference study in the field of omnidirectional robot control. A simple, yet, computationally effective, an exponential decreased error control scheme which later is called a model based approach is proposed. Furthermore, a proportional-integral control scheme is discussed and compared.

The rest discussion of this work is organised as follows: Section II presents the fundamental theory of a rigid body motion, followed by the development of the generic inverse kinematic omnidirectional robot model in Section III in detail. The model based and the P-I control scheme formulations are described in Section IV Simulation results and analysis of robot control performances for both cases: static and moving targets are discussed in Section V. Concluding remarks and future work are addressed in Section VI.

II. Rigid Body Motion

The development of the robot kinematics is derived from the concept of a rigid body motion in which its geometric aspect is focussed. Describing a rigid body motion is more complicated than a particle motion since it involves not only translation but also rotation motion. A planar body motion within a time interval between t and $t + \Delta t$, and vectors describing its motion are illustrated in Figure 1(a). The planar motion means that during the motion, any particle in the rigid body is limited in a plane which always parallel to each other and equidistant to a fixed plane. The path of any particle in a rigid body during translation motion can be a straight line or a curve, where the first is also called rectilinear translation, and the latter is known as a curvilinear translation. However, in both translations, the path of any given points in rigid body are identical. Let's analyse two arbitrary points in a rigid body, point A and point B. Selecting a point A as a reference point, where the virtual frame is reference is located, the location of point B relative to point A is described as $\mathbf{r}_{B/A}$ as shown in Figure 1(b). In any motion, $\mathbf{r}_{B/A}$ remains the same. At any given time of motion during translation, vector $\mathrm{d}\mathbf{r}_A$ and vector $\mathrm{d}\mathbf{r}_B$ represent the displacement of point A and point B, where in this step of motion, $\mathrm{d}\mathbf{r}_A = \mathrm{d}\mathbf{r}_B$. In the second step of motion, a rigid body rotates about a fixed axis of rotation, z axis of reference frame at point A, with angular velocity $\boldsymbol{\omega}$. In this step, point B is displaced at a new location where the new displacement vector is expressed as $\mathrm{d}\mathbf{r}_B$, and the displacement vector of point B before the rotation is expressed as $\mathrm{d}\mathbf{r}_A$, as depicted in Figure 1(c). Vector $\mathrm{d}\mathbf{r}_{B/A}$ describes the change of relative position of point B with respect to point A. Thus, the vectors relation can be expressed as

$$\mathrm{d}\mathbf{r}_B = \mathrm{d}\mathbf{r}_A + \mathrm{d}\mathbf{r}_{B/A} \tag{1}$$

By taking the time derivative of Eq.(1), we have

$$\begin{aligned} \frac{\mathrm{d}\mathbf{r}_B}{\mathrm{d}t} &= \frac{\mathrm{d}\mathbf{r}_A}{dt} + \frac{\mathrm{d}\mathbf{r}_{B/A}}{dt} \qquad (2) \\ \mathbf{v}_B &= \mathbf{v}_A + \mathbf{v}_{B/A} \qquad (3) \end{aligned}$$

where Eq.(3) describes the equation of the linear velocity of any point in rigid body undergoing general plane motion. $\mathbf{v}_A$ and $\mathbf{v}_B$ denote the absolute velocity of point A and point B, respectively. $\mathbf{v}_{B/A}$ expresses the relative velocity of point B with respect to point A, where it can be seen as the result of cross product between the angular velocity $\boldsymbol{\omega}$ and the displacement vector $\mathbf{r}_{B/A}$.

$$\mathbf{v}_B = \mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}_{B/A} \tag{4}$$

The same body motion process can be exercised by taking point B as the location of the frame of reference to obtain the linear velocity relation in Eq.(5)

$$\mathbf{v}_A = \mathbf{v}_B + \boldsymbol{\omega} \times \mathbf{r}_{A/B} \tag{5}$$

III. The Kinematics Development

To begin this section, it is necessary to introduce in advance, the notations or symbols which are used in the kinematics development process as follows

$\mathbf{r} = (x, y, z)^\mathsf{T}$ denotes the position vector in x, y, z axes

$\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)^\mathsf{T}$ denotes the angular position vector

$\mathbf{x} = (\mathbf{r}, \boldsymbol{\theta})^\mathsf{T}$ denotes the posture vector

$\mathbf{v} = (v_x, v_y, v_z)^\mathsf{T}$ denotes the translational velocity vector

$\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^\mathsf{T}$ denotes the rotational velocity vector

$\boldsymbol{\xi} = (\mathbf{v}, \boldsymbol{\omega})^\mathsf{T}$ denotes the velocity vector

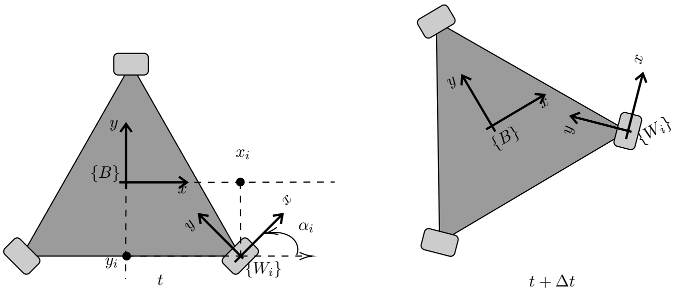


Fig. 2: Absolute robot body motion

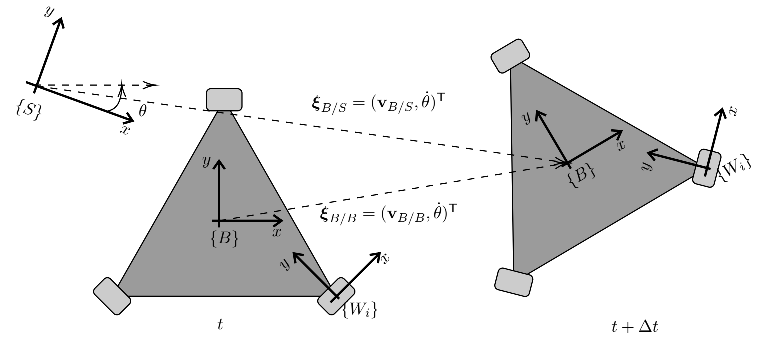

y
$\{S\}$
x
θ
$\boldsymbol{\xi}_{B/S} = (\mathbf{v}_{B/S}, \dot{\theta})^\mathsf{T}$
y
x
$\{B\}$
$\boldsymbol{\xi}_{B/B} = (\mathbf{v}_{B/B}, \dot{\theta})^\mathsf{T}$
y
x
$\{W_i\}$
t
y
x
$\{B\}$
y
x
$\{W_i\}$
$t + \Delta t$

Fig. 3: Body and spatial twist

Figure 2 depicts robot general body motion. The geometric centre of robot body is selected as the reference frame of any point in the robot body denoted as frame $\{B\}$. The body frame of the i-th wheel is denoted as W_i where its origin (x_i, y_i) is expressed in the robot body frame. As discussed earlier in Section II about general motion of rigid body, it can be explicitly processed that the absolute velocity of the i-th wheel with respect to the robot body frame $\mathbf{v}_{W/B} = (\dot{x}_{W/B}, \dot{y}_{W/B}, \dot{z}_{W/B})^\mathsf{T}$ can be expressed as

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

where $\mathbf{v}_{B/B} = (\dot{x}_{B/B}, \dot{y}_{B/B}, \dot{z}_{B/B})^\mathsf{T}$ represents the absolute velocity of the robot body reference frame, and the cross product is computed using formal determinant as follows

$$\boldsymbol{\omega} \times \mathbf{r}_{W/B} \quad = \quad \begin{vmatrix} \hat{\boldsymbol{\imath}} & \hat{\boldsymbol{\jmath}} & \hat{\boldsymbol{k}} \\ \omega_x & \omega_y & \omega_z \\ x_i & y_i & z_i \end{vmatrix} \tag{7}$$

and in this case, the only angular velocity of the robot body is measured about z axis, therefore $\boldsymbol{\omega} = (0, 0, \omega_z)^\mathsf{T}$. It is assumed that either the robot body and the robot wheels do not move in the z axis direction. Without loss of generality, in this step of the kinematics development, moving out the z axis component from the equation, we have

$$\begin{pmatrix} \dot{x}_{W/B} \\ \dot{y}_{W/B} \end{pmatrix} \quad = \quad \begin{pmatrix} \dot{x}_{B/B} \\ \dot{y}_{B/B} \end{pmatrix} + \begin{pmatrix} -y_i\omega_z \\ x_i\omega_z \end{pmatrix} \tag{8}$$

Now, let's see the motion of the robot in more general by placing the world frame of reference $\{S\}$ in any arbitrary location in the robot working space (see Figure 3). The absolute robot body velocity $\boldsymbol{\xi}_{B/B}$ can be represented in the world frame $\{S\}$ using transformation matrix $\mathbf{H}_{B/S}$ which is computed as

$$\begin{aligned} \boldsymbol{\xi}_{B/S} &= \mathbf{H}_{B/S}\boldsymbol{\xi}_{B/B} && (9) \\ &= \begin{pmatrix} \mathbf{R}_{B/S} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{B/B} \\ \omega_z \end{pmatrix} && (10) \\ &= \underbrace{\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_{B/B} \\ \dot{y}_{B/B} \\ \omega_z \end{pmatrix}}_{\text{spatial twist velocity}} && (11) \end{aligned}$$

where $\mathbf{R}_{B/S}$ is two dimensional rotation matrix, θ is the angular position of the robot body frame $\{B\}$ with respect to the world frame $\{S\}$ and ω_z is the angular velocity about z axis. Note that any vector sitting on the plane of the robot body experiences the same angular velocity which is also the same as the angular velocity of the robot body frame with respect to the world frame $\dot{\theta}$. In many robotics literature, $\boldsymbol{\xi}_{B/S}$ is also called the spatial twist velocity. The idea of developing the robot kinematics control algorithm is to have the inverse kinematics, which represent the mapping between the spatial twist velocity and the control signals which will be given to actuate each motor to drive the wheel. Therefore, the final representation of the kinematics should be expressed in the body frame of each wheel. Thus Eq.(11) should be rearranged to get its inverse, which is derived as follows

$$\begin{aligned} \boldsymbol{\xi}_{B/B} &= \mathbf{H}_{B/S}^{-1}\boldsymbol{\xi}_{B/S} && (12) \\ &= \begin{pmatrix} \mathbf{R}_{B/S}^{-1} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{B/S} \\ \dot{\theta} \end{pmatrix} && (13) \\ &= \underbrace{\begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}}_{\text{body twist velocity}} && (14) \end{aligned}$$

Eq.(14) is also known as body twist velocity formula, or the absolute velocity as discussed in the previous section.

Indeed, the $\mathbf{R}_{B/S}^{-1}\mathbf{v}_{B/S}$ denotes the absolute velocity of the robot body $\mathbf{v}_{B/B}$. Let's substitute it into Eq.(8), thus we have

$$\begin{aligned}\mathbf{v}_{W/B} &= \mathbf{R}_{B/S}^{-1}\mathbf{v}_{B/S} + \boldsymbol{\omega} \times \mathbf{r}_{W/B} &(15)\\ &= \begin{pmatrix}\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{pmatrix}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} + \begin{pmatrix}-y_i\omega_z\\ x_i\omega_z\end{pmatrix} &(16)\end{aligned}$$

Working on Eq.(16) to express with full spatial velocity parameters, it can be a new expression of $\mathbf{v}_{W/B}$ as

$$\mathbf{v}_{W/B} = \begin{pmatrix}1 & 0 & -y_i\\ 0 & 1 & x_i\end{pmatrix}\begin{pmatrix}\cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} \quad (17)$$

Taking the same idea of Eq.(14), the absolute velocity of the wheel body or the wheel body twist velocity can be computed as

$$\mathbf{v}_{W/W} = \begin{pmatrix}\cos(\alpha_i) & \sin(\alpha_i)\\ -\sin(\alpha_i) & \cos(\alpha_i)\end{pmatrix}\mathbf{v}_{W/B} \quad (18)$$

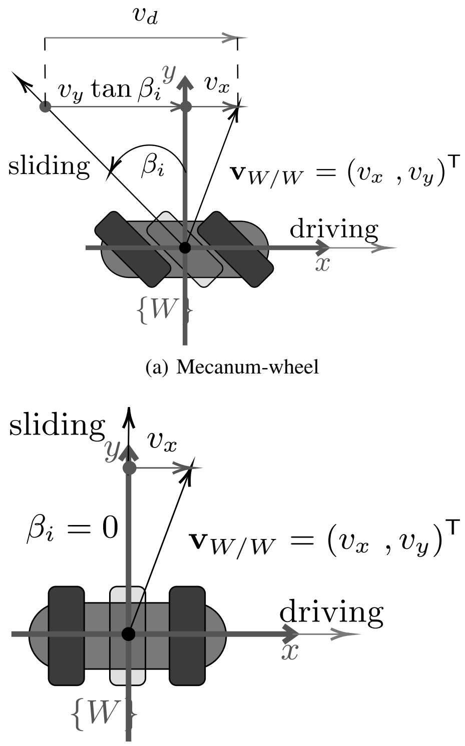
In the last expression, the spatial twist velocity vector $(\dot{x}, \dot{y}, \dot{\theta})^\mathsf{T}$ has now been propagated to have the velocity vector of the wheel body frame $\mathbf{v}_{W/W}$ where the velocity vector elements are denoted as $(v_x, v_y)^\mathsf{T}$. Next step of the kinematic control algorithm development is to find the mapping function between the wheel velocity $\mathbf{v}_{W/W}$ and the angular velocity control command u_i of the i-th wheel. To further the control algorithm development process, the kinematics of the common types of robot wheel for omnidirectional motion: mecanum-wheel and omni-wheel, should be known. Figure 4 illustrates the kinematic structure and the wheel velocity vector. Both omnidirectional wheel types have similar element of the component structure, rollers, which make the wheel able to freely slide in any direction. The difference of those rollers on mecanum-wheel and omni-wheel is the direction of the free sliding vector which characterised by angle β_i measured from the y axis of the wheel body frame as depicted in Figure 4(a) and Figure 4(b). Indeed, the wheel driving direction is parallel with the x axis of the wheel body frame which is denoted as v_d. Composing the driving velocity from the element of the wheel body twist velocity, we have

$$\begin{aligned}v_d &= v_x + v_y\tan(\beta_i) &(19)\\ &= \begin{pmatrix}1 & \tan(\beta_i)\end{pmatrix}\begin{pmatrix}v_x\\ v_y\end{pmatrix} &(20)\end{aligned}$$

Denoting the translational driving velocity v_d as the product of the angular velocity of the wheel u_i and the radius of wheel circumference r (Note that every wheel radius is assumed to be identical), thus we have

$$\begin{aligned}u_i &= \frac{v_d}{r} &(21)\\ u_i &= \begin{pmatrix}\frac{1}{r} & \frac{\tan(\beta_i)}{r}\end{pmatrix}\begin{pmatrix}v_x\\ v_y\end{pmatrix} &(22)\\ &= \begin{pmatrix}\frac{1}{r} & \frac{\tan(\beta_i)}{r}\end{pmatrix}\mathbf{v}_{W/W} &(23)\end{aligned}$$

Substituting Eq.(18) into Eq.(23), we finally obtain the robot spatial velocity, Eq.(24), and the i-th wheel angular velocity control command mapping function $\mathbf{g}_i(\theta)$, Eq.(25). For the case of n wheels robot, the robot inverse Jacobian matrix denoted as $\mathbf{G}(\theta) \in \mathbb{R}^{n\times 3}$ can be compossed by stacking all $\mathbf{g}_i(\theta)$ in the rows direction,



(a) Mecanum-wheel

(b) Omni-wheel

Fig. 4: The driving velocity vector of mecanum and omni wheels

$$\mathbf{G}(\theta) = \begin{pmatrix}\mathbf{g}_1(\theta)\\ \mathbf{g}_2(\theta)\\ \vdots\\ \mathbf{g}_n(\theta)\end{pmatrix} \quad (26)$$

Therefore, the vector of the robot control signals $\mathbf{u} = (u_1, u_2, \cdots, u_n)^\mathsf{T}$ is computed as

$$\mathbf{u} = \mathbf{G}(\theta)\boldsymbol{\xi}_{B/S} \quad (27)$$

Eq.(27) is known as the inverse velocity robot kinematics. For simulation purpose and kinematic model analysis using state-space model, the robot forward velocity kinematic computation is required which simply computed as

$$\boldsymbol{\xi}_{B/S} = \mathbf{G}^\dagger(\theta)\mathbf{u} \quad (28)$$

where $\mathbf{G}^\dagger = \mathbf{G}^\mathsf{T}\left(\mathbf{G}\mathbf{G}^\mathsf{T}\right)^{-1}$ is the pseudo-inverse of $\mathbf{G}(\theta)$. Now we can verify the controllability of the velocity kinematic model in Eq.(28). Let's recall the general form of the state-space model, expressed as follows

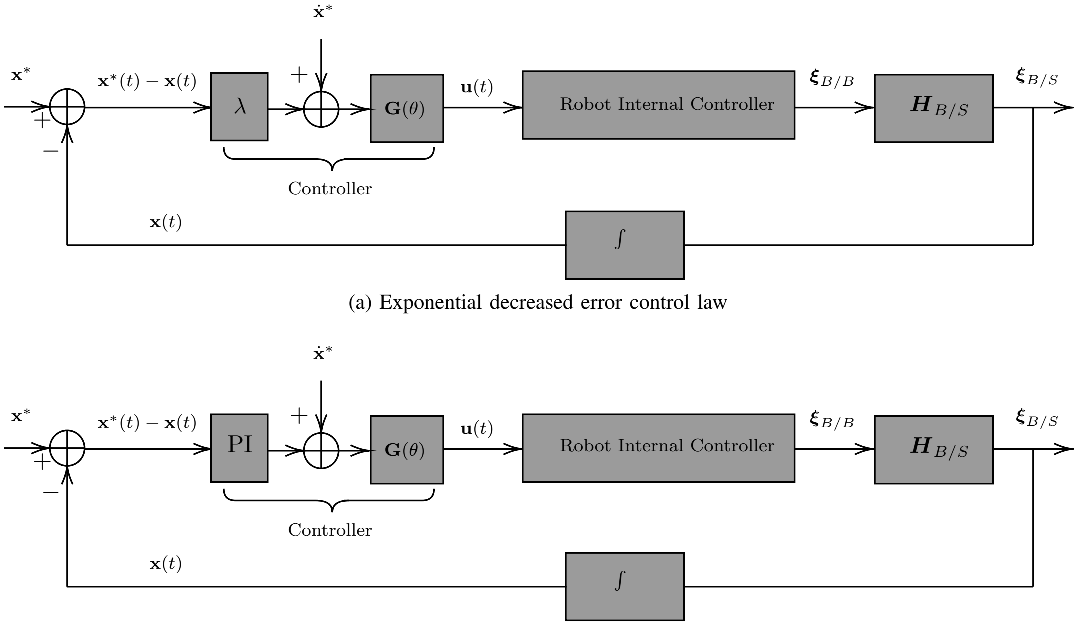
$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\mathbf{u} \quad (29)$$

where $\boldsymbol{x}$ is the robot state vector which, in this case, $\boldsymbol{x} = (x, y, \theta)^\mathsf{T}$ represents the robot posture state in the task space. Therefore, it is trivial from Eq.(28), we can deduce that $\boldsymbol{A} \in \mathbb{R}^{m\times m} = 0$ and $\boldsymbol{B} = \mathbf{G}^\dagger$. The state model in Eq.(29) is known to be controllable if rank($\boldsymbol{\mathcal{C}}$) = dim($\mathbf{x}$) = m is satisfied:

$$\text{rank}\left(\boldsymbol{\mathcal{C}}\right) = \text{rank}\left(\boldsymbol{B}\ \boldsymbol{A}\boldsymbol{B}\ \boldsymbol{A}^2\boldsymbol{B}\boldsymbol{A}^{n-1}\boldsymbol{B}\right) = n \quad (30)$$

$$u_i = \underbrace{\begin{pmatrix}\frac{1}{r} & \frac{\tan(\beta_i)}{r}\end{pmatrix}\begin{pmatrix}\cos(\alpha_i) & \sin(\alpha_i)\\ -\sin(\alpha_i) & \cos(\alpha_i)\end{pmatrix}\begin{pmatrix}1 & 0 & -y_i\\ 0 & 1 & x_i\end{pmatrix}\begin{pmatrix}\cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{pmatrix}}_{\mathbf{g}_i(\theta)}\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} \tag{24}$$

$$\mathbf{g}_i(\theta) = \frac{1}{r}\begin{pmatrix}-\sin(\alpha_i+\theta)\tan(\beta_i)+\cos(\alpha_i+\theta)\\ \sin(\alpha_i+\theta)+\cos(\alpha_i+\theta)\tan(\beta_i)\\ x_i\left(\sin(\alpha_i)+\cos(\alpha_i)\tan(\beta_i)\right)+y_i\left(\sin(\alpha_i)\tan(\beta_i)-\cos(\alpha_i)\right)\end{pmatrix}^{\mathsf{T}} \tag{25}$$



(a) Exponential decreased error control law

(b) PI control law

Fig. 5: Feedback control diagrams

Subsituting Eq.(27) into Eq.(29), it is trivial that rank($\boldsymbol{\mathcal{C}}$) $= n$ since $\boldsymbol{B}$ is the identity matrix $\mathbb{I}$ which rank($\mathbb{I}$) $= n$. Therefore it implies that there exists such controller law to stabilise the robot at the origin.

IV. Control Law Algorithm

The problem for developing the control law is to find a state feedback control algorithm that can achieve error, $\mathbf{e}$, tracking such that $\lim\limits_{t\to\infty} \mathbf{e} = 0$ and the control law vector $\mathbf{u}$ is bounded for $0 \leq t < \infty$. Let's define the error function as follows

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}^*(t) \tag{31}$$

and design the control law which can reduce the error exponentially, thus the exponential decreased error can be expressed as follows

$$\frac{\mathrm{d}\mathbf{e}(t)}{\mathrm{d}t} = -\lambda\mathbf{e}(t) \tag{32}$$

where λ is a positive constant which expresses the rate of exponentially decreased error. Taking the derivative of Eq.(31) with respect to time t, we get

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

and substituting Eq.(31) and Eq.(32) into Eq.(33), we have

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = -\frac{\mathrm{d}\mathbf{e}(t)}{\mathrm{d}t} + \frac{\mathrm{d}\mathbf{x}^*(t)}{\mathrm{d}t} \tag{34}$$

$$\dot{\mathbf{x}} = \lambda(\mathbf{x}^*(t) - \mathbf{x}(t)) + \dot{\mathbf{x}}^*(t) \tag{35}$$

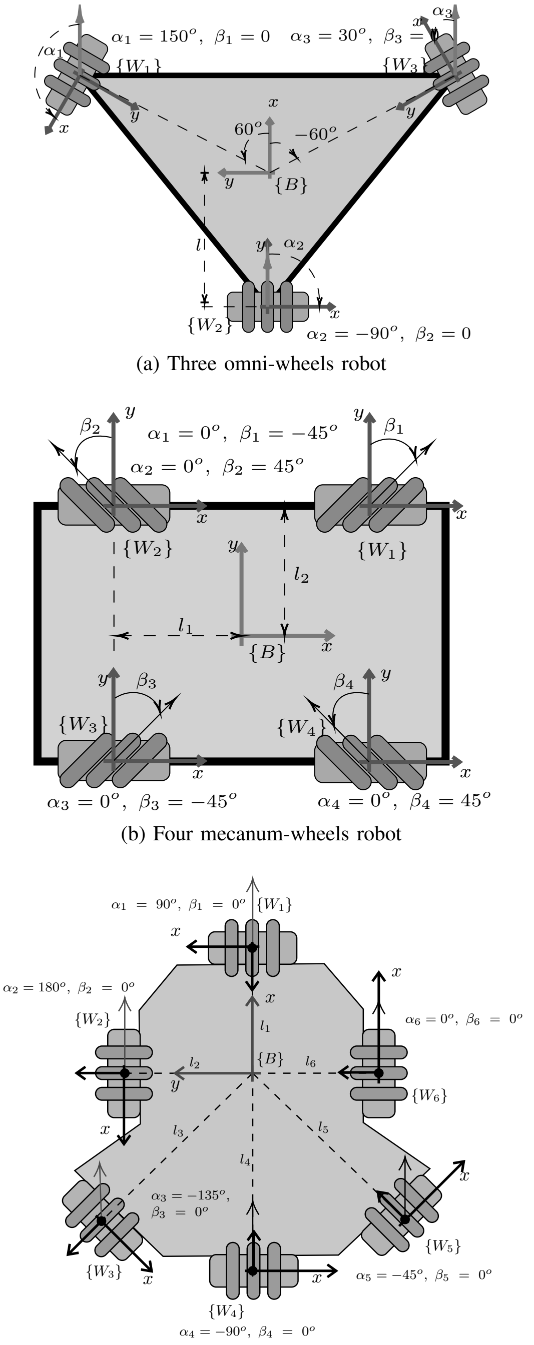
It is known from previous discussion that $\dot{\mathbf{x}}$ is another representation of robot spatial velocity $\boldsymbol{\xi}_{B/S}$. Therefore, we finally obtain an exponential decresed error control law for omnidirectional robot by substituting Eq.(35) into Eq.(27), computed as

$$\mathbf{u} = \mathbf{G}(\theta)\left(\lambda(\mathbf{x}^*(t) - \mathbf{x}(t)) + \dot{\mathbf{x}}^*(t)\right) \tag{36}$$

In this work, a P-I (proportional-integral) controller scheme is compared. The P-I controller is described as follows:

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}^*(t) + k_p(\mathbf{x}^*(t) - \mathbf{x}(t)) + k_i\int_0^{\tau}(\mathbf{x}(\tau)^* - \mathbf{x}(\tau))\mathrm{d}\tau \tag{37}$$

where k_p and k_i are the proportional and the derivative controller gains. Note that k_p and k_i are positive constants. τ is the integration time limit or integration cycle limit in digital controller. Therefore, the control signal using PI


$\alpha_1 = 150^o,\ \beta_1 = 0$
$\alpha_3 = 30^o,\ \beta_3 = 0$
$\{W_1\}$
$\{W_3\}$
$\{W_2\}$
$\{B\}$
60^o
-60^o
l
$\alpha_2 = -90^o,\ \beta_2 = 0$
x
y
$\alpha_1 = 0^o,\ \beta_1 = -45^o$
$\alpha_2 = 0^o,\ \beta_2 = 45^o$
β_1
β_2
β_3
β_4
$\{W_4\}$
l_1
l_2
$\alpha_3 = 0^o,\ \beta_3 = -45^o$
$\alpha_4 = 0^o,\ \beta_4 = 45^o$
$\alpha_1 = 90^o,\ \beta_1 = 0^o$
$\alpha_2 = 180^o,\ \beta_2 = 0^o$
$\alpha_6 = 0^o,\ \beta_6 = 0^o$
$\{W_5\}$
$\{W_6\}$
l_3
l_4
l_5
l_6
$\alpha_3 = -135^o,\ \beta_3 = 0^o$
$\alpha_5 = -45^o,\ \beta_5 = 0^o$
$\alpha_4 = -90^o,\ \beta_4 = 0^o$

(a) Three omni-wheels robot

(b) Four mecanum-wheels robot

(c) Six omni-wheels robot

Fig. 6: Omnidirectional robot kinematics structures

control law is computed as:

$$\mathbf{u} = \mathbf{G}(\theta)(\dot{\mathbf{x}}^*(t) + k_p(\mathbf{x}^*(t) - \mathbf{x}(t)) + k_i \int_0^\tau (\mathbf{x}(\tau)^* - \mathbf{x}(\tau))\mathrm{d}\tau) \tag{38}$$

Both, the exponential decreased error and the PI control schemes can be illustrated in state feedback control diagram as depicted in Figure 5. In real-time application, the function of the internal robot controller is to stabilise the motors of robot wheels at the desired angular velocity which is given by the computation of the control law. In simulation, such ideal motor model is assumed, the internal robot controller is simply the implementation of the forward velocity kinematics as described in Eq.(28).

V. Simulation results and discussion

In this work, simulation programs were written using Python programming language. The kinematics modelling and control system performance are provided. Case studies using three different robot kinematic configurations are discussed. The first two robot kinematic configurations are the most commonly used in robotic applications and research: three omni-wheels and four mecanum-wheels robots. The last case is to study a unique turtle-like mobile robot kinematic configuration where 6 omni-wheels are used. The 6 omni-wheels robot configuration is a complex kinematic structure, therefore, it is important to the reader to a better understanding to formulate the kinematic control algorithm using the proposed generic kinematic formulation method. The robot kinematic configurations in this discussion are depicted in Figure 6.

Table I and Table II show the kinematic parameters of three omni-wheels, four mecanum-wheels and six omni-wheels robots, respectively

TABLE I: Specification of 3 omni-wheels and 4 mecanum-wheels robot

Wheels Number	Three omni wheels		Four mecanum wheels	
	α_i	β_i	α_i	β_i
W_1	150^o	0^o	0^o	-45^o
W_2	-90^o	0^o	0^o	45^o
W_3	30^o	0^o	0^o	-45^o
W_4	-	-	0^o	45^o

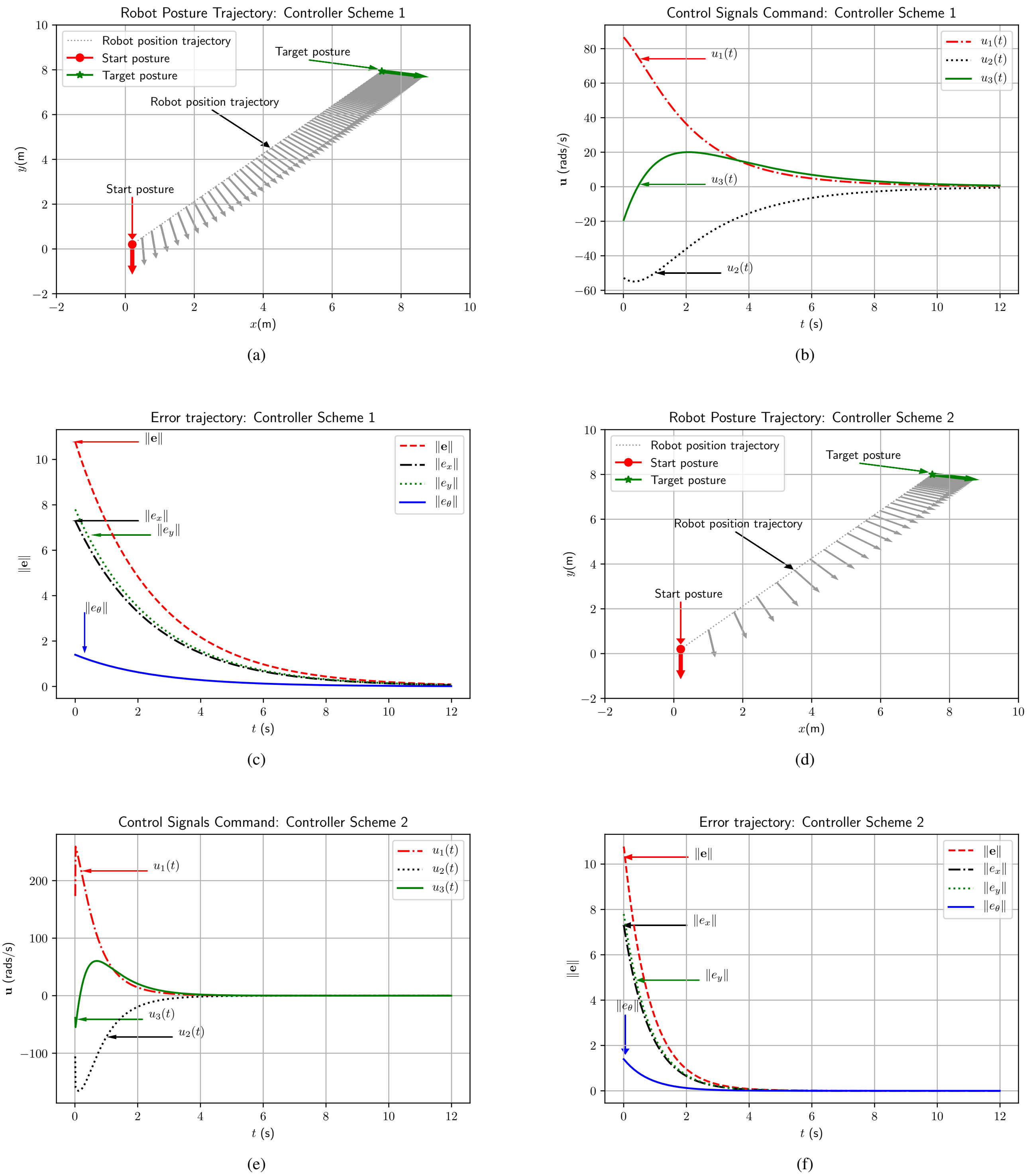
TABLE II: specification of 6 omni-wheels robot

Wheels Number	Six omni wheels	
	α_i	β_i
W_1	90^o	0^o
W_2	180^o	0^o
W_3	-135^o	0^o
W_4	-90^o	0^o
W_5	-45^o	0^o
W_6	0^o	0^o

By referring Figure 6, the position each wheel frame (x_i, y_i) with respect to the robot body frame $\{B\}$ can be obtained. The positions of each wheel frame on three omni-wheels robot are $(x_1, y_1) = \left(l\cos\left(\frac{\pi}{3}\right), l\sin\left(\frac{\pi}{3}\right)\right)$, $(x_2, y_2) = (l\cos(\pi), l\sin(\pi))$, and $(x_3, y_3) = \left(l\cos\left(-\frac{\pi}{3}\right), l\sin\left(-\frac{\pi}{3}\right)\right)$ where l is a shared distance between the robot frame and the wheel frame. The rest of the kinematics parameters can be used (see Table I) to obtain the inverse robot Jacobian described in Eq.25 which resulted as

$$\mathbf{G}(\theta) = \frac{1}{r}\begin{pmatrix} \cos\left(\theta + \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) & l \\ \cos\left(\theta - \frac{\pi}{2}\right) & \sin\left(\theta - \frac{\pi}{2}\right) & l \\ \cos\left(\theta + \frac{\pi}{6}\right) & \sin\left(\theta + \frac{\pi}{6}\right) & l \end{pmatrix} \tag{39}$$

For a four mecanum-wheels robot, the positions of each wheel frame is more trivial to obtain, which are $(x_1, y_1) = (l_1, l_2)$, $(x_2, y_2) = (-l_1, l_2)$, $(x_3, y_3) = (-l_1, -l_2)$, and $(x_4, y_4) = (l_1, -l_2)$. Using the same procedure, the inverse



(a) (b) (c) (d) (e) (f)

Fig. 7: Static target tracking control performance for three omni-wheels robot

robot Jacobian for four mecanum-wheels robot can be verified as

$$\mathbf{G}(\theta) = \frac{1}{r}\begin{pmatrix} \cos(\theta)+\sin(\theta) & \sin(\theta)-\cos(\theta) & -l_1-l_2 \\ \cos(\theta)-\sin(\theta) & \sin(\theta)+\cos(\theta) & -l_1-l_2 \\ \cos(\theta)+\sin(\theta) & \sin(\theta)-\cos(\theta) & l_1+l_2 \\ \cos(\theta)-\sin(\theta) & \sin(\theta)+\cos(\theta) & l_1+l_2 \end{pmatrix} \quad (40)$$

For a Six Omniwheels, the position each wheel frame (x_i, y_i) with respect to the robot body frame $\{B\}$ can be obtained. The positions of each wheel frame on six wheeled robot are

$$(x_1, y_1) = (l\cos(0^o), l\sin(0^o)) \quad (41)$$

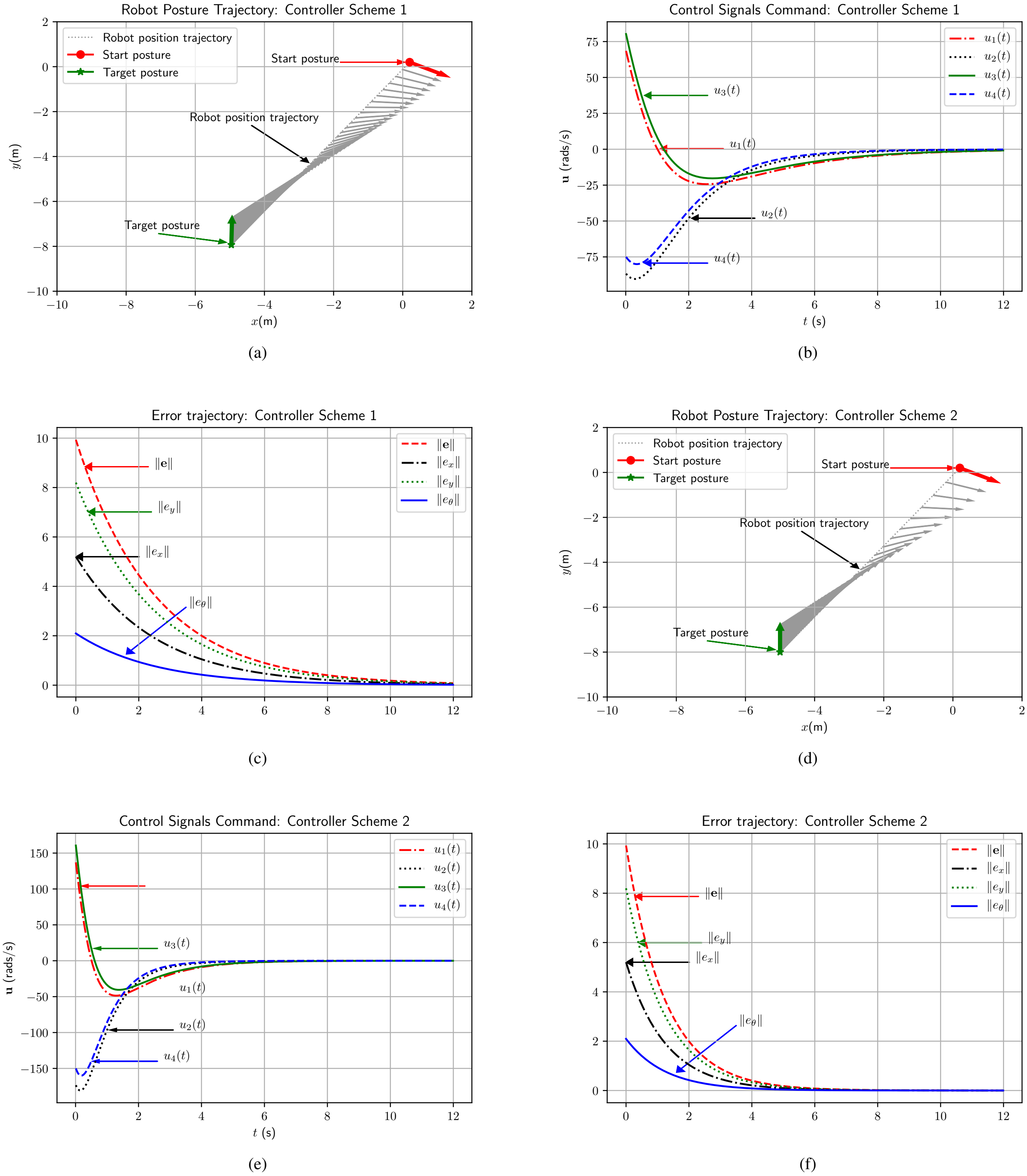
$$(x_2, y_2) = (l\cos(\frac{\pi}{2}), l\sin(\frac{\pi}{2})) \quad (42)$$

$$(x_3, y_3) = (l\cos(\frac{3\pi}{4}), l\sin(\frac{3\pi}{4})) \quad (43)$$

$$(x_4, y_4) = (l\cos(\pi), l\sin(\pi)) \quad (44)$$

$$(x_5, y_5) = (l\cos(-\frac{3\pi}{4}), l\sin(-\frac{3\pi}{4})) \quad (45)$$

$$(x_6, y_6) = (l\cos(-\frac{\pi}{2}), l\sin(-\frac{\pi}{2})) \quad (46)$$



(a)

(b)

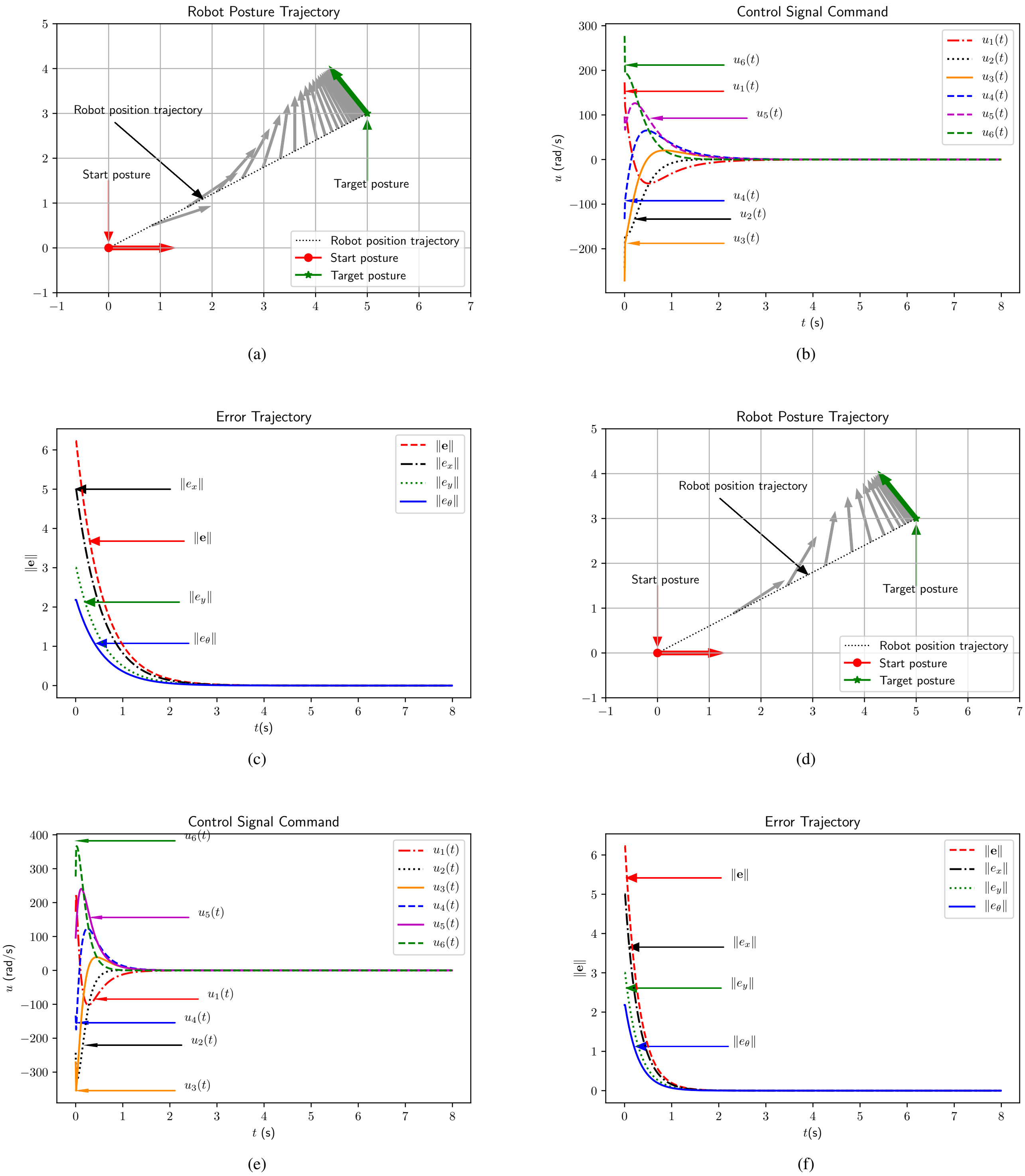
(c)

(d)

(e)

(f)

Fig. 8: Static target tracking control performance for four mecanum-wheels robot



(a)

(b)

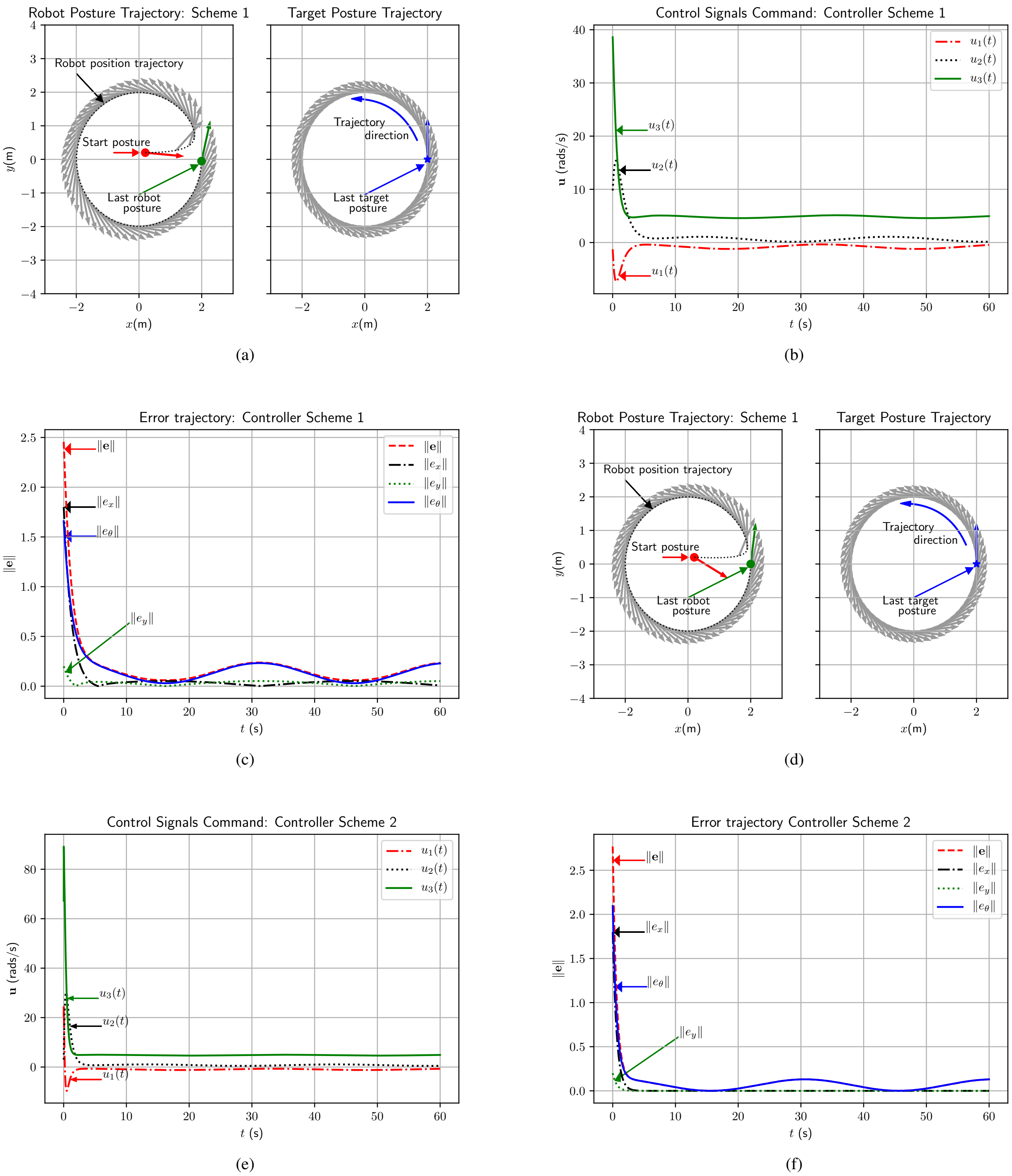
(c)

(d)

(e)

(f)

Fig. 9: Static target tracking control performance for six omni-wheels robot



(a)

(b)

(c)

(d)

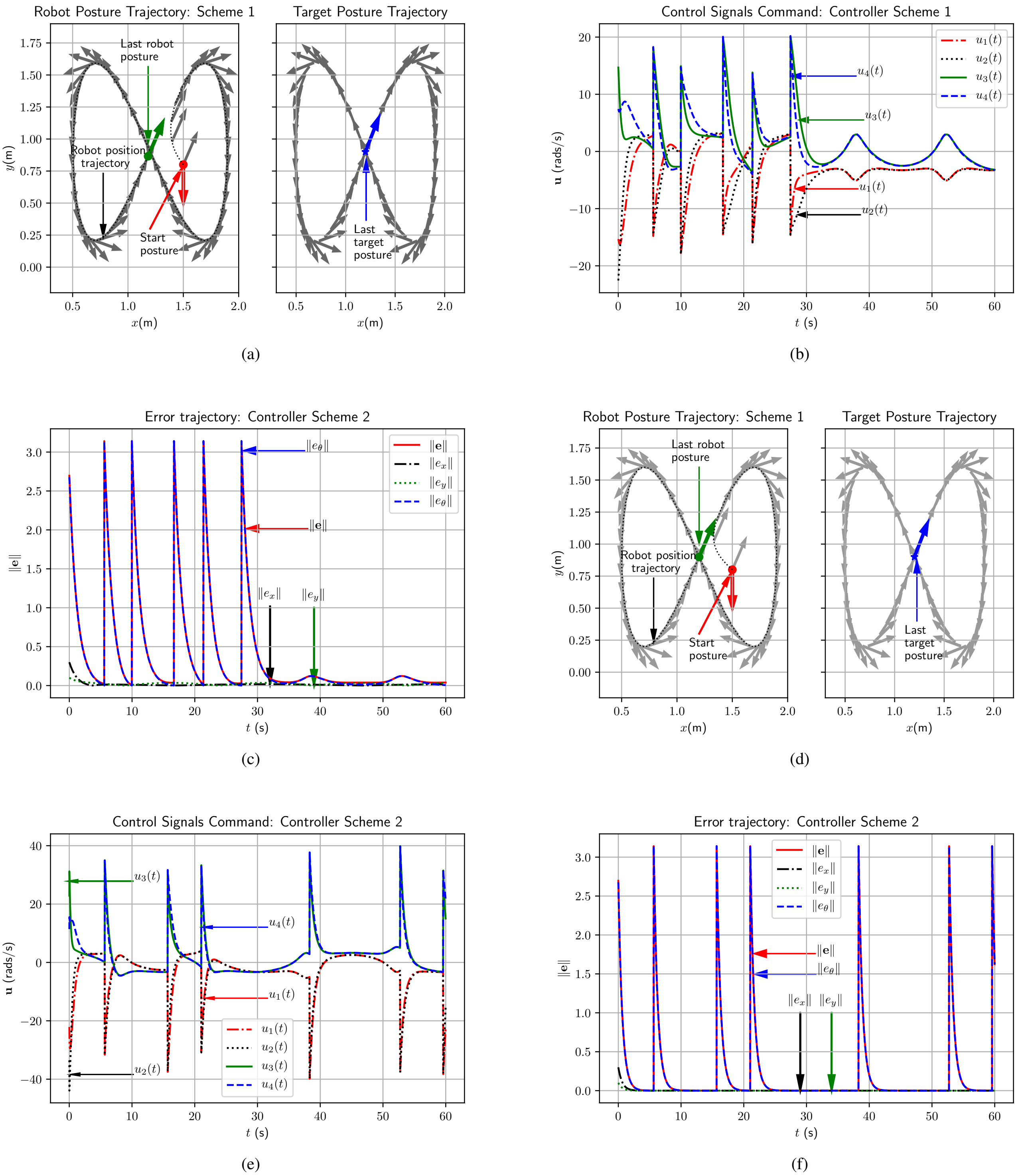
(e)

(f)

Fig. 10: Three omni-wheels robot control performance results for tracking a moving target

The inverse robot Jacobian for six omni-wheels robot can be formulated as

$$
\mathbf{G}(\theta) = \frac{1}{r}\begin{pmatrix} \cos\left(\theta + \frac{\pi}{2}\right) & \sin\left(\theta + \frac{\pi}{2}\right) & l \\ \cos\left(\theta + \pi\right) & \sin\left(\theta + \pi\right) & l \\ \cos\left(\theta - \frac{3\pi}{4}\right) & \sin\left(\theta - \frac{3\pi}{4}\right) & l \\ \cos\left(\theta - \frac{\pi}{2}\right) & \sin\left(\theta - \frac{\pi}{2}\right) & l \\ \cos\left(\theta - \frac{\pi}{4}\right) & \sin\left(\theta - \frac{\pi}{4}\right) & l \\ \cos\left(\theta\right) & \sin\left(\theta\right) & l \end{pmatrix} \tag{47}
$$



(a)

(b)

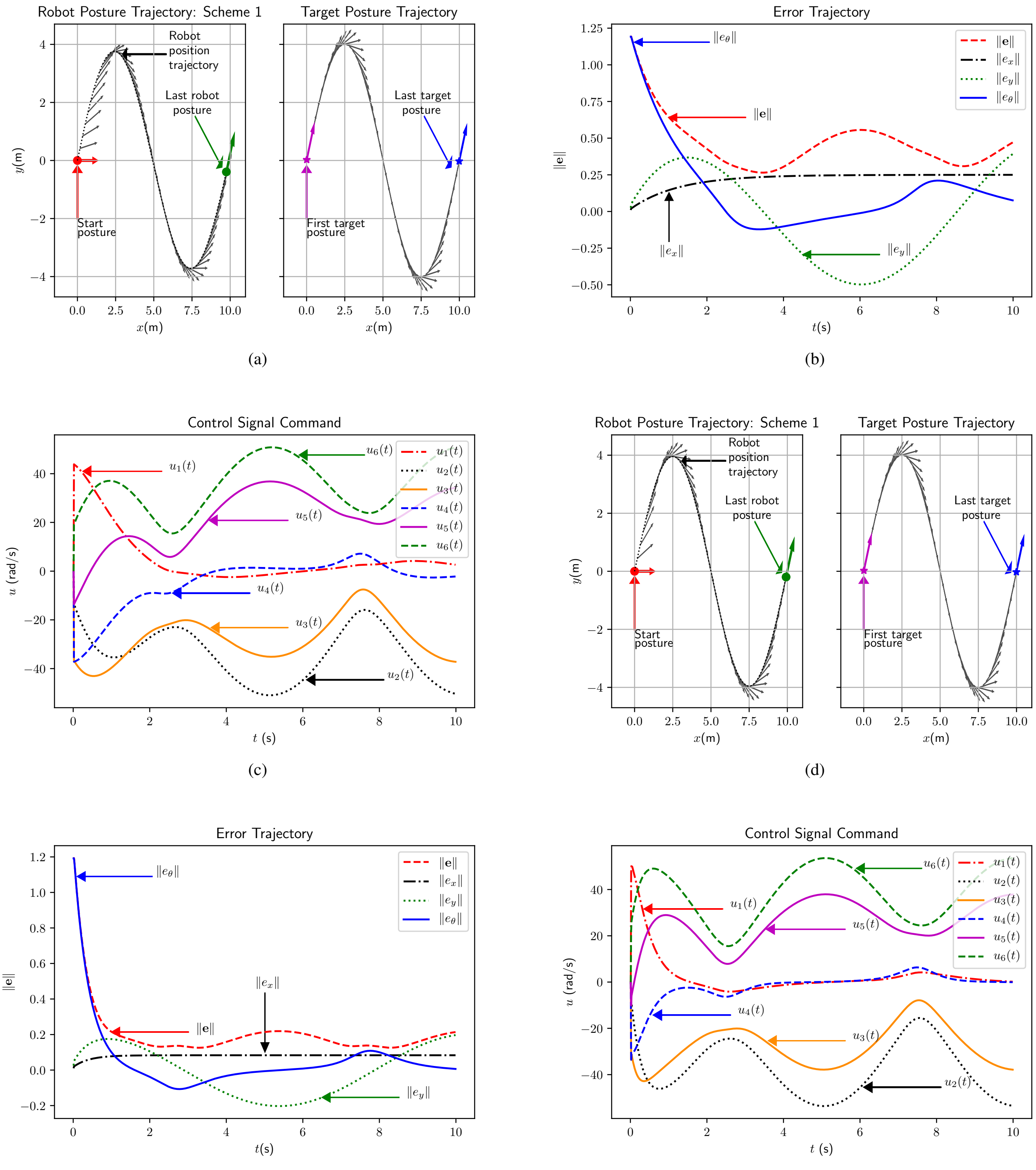
(c)

(d)

(e)

(f)

Fig. 11: Four mecanum-wheels robot control performance results for tracking a moving target



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 12: Six omni-wheels robot control performance results for tracking a moving target

The controller performances which were analysed includes: trajectories for position tracking, control signals, and error signals. Simple exponential decreased error and P-I control schemes have been used in simulation to verify the control performances for posture tracking of a static target and a moving target. Different starting robot and target postures configuration have been considered for each simulation of the three cases of kinematic configuration to enrich the understanding of the systems.

A. Control Performance for A Static Target

The simulation time was configured as $0 \leq< t < 12s$, with sampling time of $\Delta t = 0.01s$. The kinematic structure parameters of the omni-wheeled robot and the mecanum-wheeled robot were ($l = 0.5\text{m}$) and ($l_1 = 0.25\text{m}, l_2 = 0.1\text{m}$), respectively. Figure 7 depicts the comparison of the control performance between the model based control scheme (Scheme 1) and the P-I control scheme (Scheme 2) for tracking a static target application. The controller gains were set as ($\alpha = 0.8$) and ($k_p = 0.8, k_i = 0.4$) for the model based controller and the P-I controller, respectively. The controller gains were obtained by trial-error to get good results to be presented. Computation to find the controller gains based on the expected controller output performances is not part of this current work. In this simulation, the initial robot and the target postures were defined as $\mathbf{x}_0 = (0.2\text{m}, 0.2\text{m}, -90^o)^\mathsf{T}$ and $\mathbf{x}^* = (7.5\text{m}, 8.\text{m}, 170^o)^\mathsf{T}$, respectively. Figure 7(a), 7(b) and 7(c) shows the robot posture trajectory $\mathbf{x}(t)$, the wheel motor control signals $\mathbf{u}(t)$ and the error signals $\mathbf{e}(t)$, respectively, for the case of the model based controller. Figure 7(d), 7(e) and 7(f) are for the case of P-I controller. It can be seen from the robot posture trajectories that the holonomic or omnidirectionality of robot movements are obvious, where translational and rotational motions were executed in the same instant of time, to contrast with the nonholonomic robots such as: differential drive and car-like robots, where the translation and rotation motion, in general, have to be executed separately due to nonholonomic motion constraints [6]. The error signals $\mathbf{u}(t)$ reduces exponentially in both control schemes as shown in Figure 7(c) and Figure 7(f). It can be observed that the P-I controller converged faster at $t = 6s$ than the model based controller which converged at $t = 12s$. It can be observed that the P-I controller results faster tracking than the model based controller.

The same simulation control performance measurements have been observed for the case of the four mecanum-wheels robot as illustrated in Figure 8. The initial and the target robot postures were configured as $\mathbf{x}_0 = (0.2\text{m}, 0.2\text{m}, -30^o)^\mathsf{T}$ and $\mathbf{x}_0 = (-5.0\text{m}, -8.0\text{m}, 90^o)^\mathsf{T}$, respectively. Other than the robot and the target postures, the same simulation parameters used in the omni wheels were configured. The comparison of the robot posture trajectories between the model based and the P-I controllers can be seen in Figure 8(a) and Figure 8(d). Again, the omnidirectionality motions have been demontrated by the four mecanum wheels robot. At the same instant as the robot translates, the control signals (see Figure 8(b) and Figure 8(e)) correct the robot orientation in the positive direction of rotation (counter clock wise direction). The exponential decreased errors trajectories can be achieved as shown in Figure 8(c) and Figure 8(f).

Figure 9 depicts the control performance of the robot positioning task for the case of six omni-wheels robot. The initial state of $\mathbf{x}_0 = (0\text{m}, 0\text{m}, -0^o)^\mathsf{T}$ and the target state of $\mathbf{x}^* = (5.0\text{m}, 3.0\text{m}, 135.0^o)^\mathsf{T}$ were used to validate the performance of model based and the P-I control schemes. The state error of both schemes decreased exponentially and the angular velocities of the six omni-wheels converged to zero.

B. Control Performance for A Moving Target

For a moving target robot tracking application, the velocity vector $\dot{\mathbf{x}}^*(t)$ should be known. In simulation, the desired posture of a moving target trajectory vector $(x^*(t), y^*(t))$ is defined in time interval $t \in [0, T]$. A smooth target posture can be defined by setting the orientation in the tangent direction of a moving target velocity, thus

$$\theta^*(t) = \arctan2(\dot{y}^*(t), \dot{x}^*(t)) \tag{48}$$

The $\arctan2$ is the four-quadrant inverse tangent function. The desired angular velocity $\dot{\theta}^*(t)$ can be obtained by taking the first derivative of Eq.(48) which derived as follows

$$\begin{aligned} \dot{\theta}^*(t) &= \frac{\partial \theta^*}{\partial \dot{x}^*}\frac{\mathrm{d}\dot{x}^*}{\mathrm{d}t} + \frac{\partial \theta^*}{\partial \dot{y}^*}\frac{\mathrm{d}\dot{y}^*}{\mathrm{d}t} && (49)\\ &= \frac{\dot{x}^*\ddot{y}^* - \dot{y}^*\ddot{x}^*}{\dot{x}^* + \dot{y}^*} && (50) \end{aligned}$$

Therefore, in the next discussion of robot control simulations, the desired moving trajectory $\dot{\mathbf{x}}^* = (\dot{x}^*, \dot{y}^*, \dot{\theta}^*)$ is defined by selecting the desired position function at first, and followed by generating the desired angular velocity vector Eq.(50) within time interval $t \in [0, T]$. Many other techniques to generate the desired robot path can be found in many literatures within the topic of robot path planning. A significant contribution of robot planning algorithms is written in [28]. Recently in interesting idea of robot path planning algorithm for chasing a moving target is proposed in [29]. Indeed, in real-time application such as ”predator-prey” robot applications, the solution is not trivial. Robot sensors for observing a target or multiple targets are required, for instance: cameras and laser sensors. A complex computation for extracting target features from robot environments are required. In this work, such aforementioned real-time robot application is not in the scope of discussion, a moving target trajectories are generated numerically, instead. The moving target position trajectories for three robot kinematic configurations were defined as:

$$x^*(t) = \begin{cases} 2\cos(2\pi t) & \text{Three omni-wheels} \\ 1.2 + 0.7\sin(2\pi t) & \text{Four mecanum-wheels} \\ t & \text{Six omni-wheels} \end{cases} \tag{51}$$

$$y^*(t) = \begin{cases} 2\sin(2\pi t) & \text{Three omni-wheels} \\ 0.9 + 0.7\sin(4\pi t) & \text{Four mecanum-wheels} \\ 4\sin(2\pi t) & \text{6 omni-wheels} \end{cases} \tag{52}$$

Taking the first derivative, one can obtain

$$\dot{x}^*(t) = \begin{cases} -4\pi\sin(2\pi t) & \text{Three omni-wheels} \\ 1.4\pi\cos(2\pi t) & \text{Four mecanum-wheels} \\ 1 & \text{Six omni-wheeled} \end{cases} \tag{53}$$

$$\dot{y}^*(t) = \begin{cases} 4\pi\cos(2\pi t) & \text{Three omni-wheels} \\ 2.8\cos(4\pi t) & \text{Four mecanum-wheels} \\ 8\pi\sin(2\pi t) & \text{Six omni-wheels} \end{cases} \quad (54)$$

and taking second derivative, we have

$$\ddot{x}^*(t) = \begin{cases} 8\pi\cos(2\pi t) & \text{Three omni-wheels} \\ -2.8\pi\sin(2\pi t) & \text{Four mecanum-wheels} \\ 0 & \text{Six omni-wheels} \end{cases} \quad (55)$$

$$\ddot{y}^*(t) = \begin{cases} -8\pi\sin(2\pi t) & \text{Three omni-wheeled} \\ -4.6\sin(4\pi t) & \text{Four mecanum-wheels} \\ -16\pi\sin(2\pi t) & \text{Six omni-wheels} \end{cases} \quad (56)$$

and finally the desired angular velocity $\dot{\theta}^*(t)$ for each robot types can be computed by substituting Eq.(53) - Eq.(56) into Eq.(50). Therefore $\dot{\mathbf{x}}^*(t) = (\dot{x}^*(t), \dot{y}^*(t), \dot{\theta}^*(t))^\mathsf{T}$ can be defined. Figure 10 and Figure 11 depicts the observed control performances of omni-wheeled robot and mecanum-wheeled robot, respectively. In Figure 10(a) and Figure 10(d), the observed omni-wheeled robot posture trajectories in the two dimensional space between the model based and the P-I control laws can be compared. A smooth circular posture moving target tracking was achieved by both control schemes. From initial robot posture of $\mathbf{x}_0 = (0.2\text{m}, 0.2\text{m}, -30^o)^\mathsf{T}$, the controller signals (see Figure 10(b) and Figure 10(e)) successfully moved the robot to approach the moving target with small errors (see Figure 10(c) and Figure 10(f)) at $t = 10s$. Indeed, the control signal vector $\mathbf{u}(t)$ and the error signal vector $\mathbf{u}(t)$ during time period of $0 \leq t < 60s$ never converged to zero, since the target always move in that time period.

In contrast, the desired moving target posture trajectory for the mecanum wheeled robot was made to be more challenging where the trajectory was not a simple circle, but curved like the shape of ”8”, as seen in Figure 11(a) and Figure 11(d). Observing the tracking of the robot orientation of both schemes, the model based and the P-I controllers, there are big sudden control signals at every corner of the curve (see also Figure 11(b) and Figure 11(e)). This phenomenon occurs due to big error between the current robot orientation and the target orientation at every corner of the curve as depicted in Figure 11(c) and Figure 11(f).

Figure 12 shows the performance of the six omni-wheels robot. The moving target trajectory was made to be a sinusoid-like. By comparing robot posture trajectory and target posture trajectory, it can be verified that both kinematic control schemes: model based and P-I, have demonstrated the successful tracking performance.

VI. Conclusion and Future Works

A general inverse velocity kinematics framework for omnidirectional mobile robot has been presented. The general inverse velocity kinematics framework can be used as the basis for control law formulations. From the derived inverse velocity framework, the forward velocity kinematics formulation was obtained, thus the representation of the state-space model could be modelled. Furthermore, the controllability of the model has been verified by observing the rank of the input matrix of the state-space model. Instances of the control schemes have been proposed and verified, for two most common cases of wheel types: omnidirectional and mecanum wheels. From the simulation results, it can be concluded that both control schemes: the model based and the P-I, have demonstrated satisfactory performances either for a static and a moving target robot tracking application. It performed an exponential decreased error, where it was clearly seen in the simulation for a static target tracking. The controller gains λ, k_p and k_i were configured by trial and error during simulation attempts for good presentations. It should be noted, that the controller gains used may not be the best or the most optimal configuration. The algorithm to find the optimal control gains remained in our future work, where control law (eg. Linear Quadratic Regulator, Linear Quadratic Gaussian) from optimal control theory can be considered. The ”spiking” control commands as presented in the moving target case may be reduced using decoupled control for translation and orientation, which also remained in our future work.

Acknowledgement

The authors would like to grateful thank to the State Polytechnic of Malang for providing lab facilities. The authors also thank to Sarah Rizky Ardhani from The Wesleyan University, USA, for her useful discussion on paper proof reading.

References

[1] F. Arvin, J. Espinosa Mendoza, B. Bird, A. West, S. Watson, and B. Lennox, “Mona: an affordable open-source mobile robot for education and research,” *Journal of Intelligent and Robotic Systems*, 2018.

[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] M. Ben Ari and F. Mondada, *Robots and Their Applications*. Cham: Springer International Publishing, 2018, pp. 1–20. [Online]. Available: https://doi.org/10.1007/978-3-319-62533-1-1

[4] E. Amareswar, G. S. S. K. Goud, K. R. Maheshwari, E. Akhil, S. Aashraya, and T. Naveen, “Multi purpose military service robot,” in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 2, 2017, pp. 684–686.

[5] A. Ellery, “Space robotics: Part 3: Robotic rovers for planetary exploration,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 4, p. 31, 2004. [Online]. Available: https://doi.org/10.5772/5809

[6] Xiaoping Yun and N. Sarkar, “Unified formulation of robotic systems with holonomic and nonholonomic constraints,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 640–650, 1998.

[7] G. Klancar, D. Matko, and S. Blazic, “Mobile robot control on a reference path,” in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, 2005, pp. 1343–1348.

[8] N. Uddin, “A two-wheeled robot trajectory tracking control system design based on poles domination approach,” *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp. 154–161, 2020.

[9] M. Mendes, A. P. Coimbra, and M. M. Crisostomo, “Assis - cicerone robot with visual obstacle avoidance using a stack of odometric data,” *IAENG International Journal of Computer Science*, vol. 45, no. 1, pp. 219–227, 2018.

[10] T. Taniguchi and M. Sugeno, “Trajectory tracking controller design for a tricycle robot using piecewise multi-linear models,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2017.

[11] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019. [Online]. Available: https://doi.org/10.1177/1729881419839596

[12] P. Nilas and T. Ngo, “A multi-terrain spherical amphibious robot for on-land, in-water, and underwater operation,” in *Proceedings of The World Congress on Engineering and Computer Science 2019*, International Association of Engineers. Newswood Limited, 2019.

[13] Y.-H. Cheng, P.-J. Chao, and C.-N. Kuo, "Mobile robot path planning using a teaching-learning-interactive learning-based optimization," *IAENG International Journal of Computer Science*, vol. 46, no. 2, pp. 199–207, 2019.

[14] G. Indiveri, "Swedish wheeled omnidirectional mobile robots: Kinematics analysis and control," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 164–171, 2009.

[15] C. Rohrig and D. HeB, "Omniman: An omnidirectional mobile manipulator for human-robot collaboration," in *Proceedings of The International MultiConference of Engineers and Computer Scientists 2019*, International Association of Engineers. Newswood Limited, 2019.

[16] J. Grabowiecky, "Vehicle wheel," US1305535A·1919-06-03.

[17] J. F. Blumrich, "Omnidirectional wheel," US3789947A·1974-02-05.

[18] Z. Lu, M. Lin, S. Wang, Y. Zhang, and Y. Yu, "Research on a new-style under-actuated omnidirectional mobile robot based on special coupling drive system," *IEEE Access*, vol. 7, pp. 152 138–152 148, 2019.

[19] J. Alexander and J. Maddocks, "On the kinematics of wheeled mobile robots," *The International Journal of Robotics Research*, vol. 8, no. 5, pp. 15–27, 1989. [Online]. Available: https://doi.org/10.1177/027836498900800502

[20] R. H. Abiyev, N. Akkaya, and I. Gunsel, "Control of omnidirectional robot using z-number-based fuzzy system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 238–252, 2019.

[21] E. Malayjerdi, H. Kalani, and M. Malayjerdi, "Self-tuning fuzzy pid control of a four-mecanum wheel omni-directional mobile platform," in *Electrical Engineering (ICEE), Iranian Conference on*, 2018, pp. 816–820.

[22] C. Cai, J. Lu, and Z. Li, "Kinematic analysis and control algorithm for the ballbot," *IEEE Access*, vol. 7, pp. 38 314–38 321, 2019.

[23] D. Zhao, X. Deng, and J. Yi, "Motion and internal force control for omnidirectional wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 3, pp. 382–387, 2009.

[24] G. Klančar, G. Mušič, H. Chen, and M. Seder, "Wheeled robot navigation based on a unimodal potential function," in *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2019, pp. 1–5.

[25] J. Z. Woodruff, S. Ren, and K. M. Lynch, "Motion planning and feedback control of rolling bodies," *IEEE Access*, vol. 8, pp. 31 780–31 791, 2020.

[26] K. Kawamura, T. Emaru, Y. Kobayashi, and A. A. Ravankar, "Adaptive control for omnidirectional wheeled robot," in *2016 IEEE/SICE International Symposium on System Integration (SII)*, 2016, pp. 367–372.

[27] W. Zheng and T. Ito, "Dynamic surface control–based adaptive neural tracking for full-state constrained omnidirectional mobile robots," *Advances in Mechanical Engineering*, vol. 11, no. 4, pp. 1–14, 2019. [Online]. Available: https://doi.org/10.1177/1687814019846750

[28] S. M. LaValle, *Planning Algorithms*. USA: Cambridge University Press, 2006.

[29] D. Drake, S. Koziol, and E. Chabot, "Mobile robot path planning with a moving goal," *IEEE Access*, vol. 6, pp. 12 800–12 814, 2018.