International Journal on Advanced Science Engineering Information Technology

Vol.15 (2025) No. 2
ISSN: 2088-5334

Optimization of Robotic Movement: Applying Lie Algebra to Improve the Performance of mBot2

Puguh Wahyu Prasetyo [a,*], Adhy Kurnia Triatmaja [b], Fariz Setyawan [a], Angger Hafid Prabawa [a], Hafizh Naufal Azmi [a], Muhammad Ardiyansyah [c], Samsul Arifin [d]

[a] *Mathematics Education Study Program, Faculty of Teacher Training and Education, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[b] *Electronics and Informatics Education Study Program, Faculty of Teacher Training and Education, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[c] *Faculty of Mathematics and Science, University of Jyväskylä, Jyväskylän Yliopisto, Seminaarinkatu, Finland*
[d] *Data Science, Faculty of Engineering and Design, Institut Teknologi Sains Bandung, Indonesia*
Corresponding author: [*]puguh.prasetyo@pmat.uad.ac.id

Abstract*—Robot is expected to play a transformative role across various sectors, particularly in education, where the model contributes to the development of essential skills such as creativity, problem-solving and computational thinking in the Society 5.0 era. Educational robots such as mBot2 of Makeblock have achieved popularity for individual ability to improve learning experiences through hands-on engagement. Therefore, this research focused on the movement model of mBot2, specifically in exploring how the arm functionality of the model could be optimized using mathematical concepts such as Lie algebra. The research identified Gröbner basis for the hand system of robot. Through the implementation of Lie algebra on Python-based platform available on mBot2, the model significantly improved the movement accuracy and efficiency of robots. The analysis demonstrates that the implementation of Lie Algebra significantly reduces mBot2's orientation error from 17.25 degrees to 2.25 degrees, resulting in an 86.96% improvement. This substantial accuracy enhancement underscores Lie Algebra's efficacy in managing intricate transformations and ensuring precise coordinate transformations, making it an ideal solution for robust robotic control systems. Furthermore, the average time reduction of approximately 11.7% across both activities. These developments showed the versatility of robots as an instructional tool by increasing its value as a teaching aid and extending the application. In a specific Lie algebra, the research proposed additional advancements outside its present uses, such as building robots with switches circuit systems to improve stability. These showed how educational robots are becoming essential in current educational frameworks, supporting both fundamental and advanced learning.

***Keywords*—Lie algebra; nonholonomic car; radical of lie algebra; robotic; special euclidean.**

Manuscript received 28 Sep. 2024; revised 19 Oct. 2024; accepted 2 Mar. 2025. Date of publication 30 Apr. 2025.

I. INTRODUCTION

The role of robots in assisting human activities is predicted to increase significantly in the era of Society 5.0. This trend is evidenced by data from the past five years, showing a consistent rise in the adoption of robotic technologies across various sectors, including industry, healthcare, and education [1]. In the industrial sector, the deployment of robots is indispensable, particularly in manufacturing processes such as car assembly, painting, and other related activities. Robotic systems in healthcare have enabled surgeons to perform remote surgeries using robotic arms controlled from a distance. Additionally, robots in the field of education have become instrumental in supporting STEM (Science, Technology, Engineering, and Mathematics) education. Several companies have developed educational robot, including LEGO with its LEGO Mindstorms series and Makeblock having mBot as well as mBot2. This robot can also be assembled using control boards such as Arduino or Raspberry Pi.

The implementation of LEGO Mindstorms and mBot in STEM-based learning has been shown in various research [2]. According to a comparative analysis [2], mBot is particularly suitable for beginners due to its affordability, allowing the tool to be a viable instrument for educational research, particularly in supporting experiential learning data. Moreover, recent advancements in robotic programming have seen the successful development of software packages based on Lie Group and Algebra concepts, such as the manif packages [3] as well as navlie packages [4]. The development raises important questions about how Lie Algebra concepts can be effectively applied to control systems, especially in mBot2. This inquiry is particularly relevant in the context of maximizing the usefulness of educational robots in STEM-based learning environments, from elementary education to higher education, especially in teaching algebraic structures and Lie algebra. The objective is to analyze control strategy driving the robot car to move toward the target location (x_d, y_d, z_d), catch an object, adjust arm positions, which allow the end-effector to securely grab the object (x_p, y_p, z_p) and pick it up. The control strategy should ensure the following conditions include positioning, where the robot reaches the targeted position. Other strategies include catching an object, and arm-control, as after the robot position is close to the target, the arm should be placed and managed to reach the object by using inverse kinematics. The incorporation of Lie algebras into robotic systems presents an innovative method to address various challenges in robot dynamics and control. Lie algebras, such as $\mathfrak{se}(2)$, $\mathfrak{so}(2)$, $\mathfrak{so}(3)$, and $\mathfrak{se}(3)$, provide a mathematical framework that facilitates the analysis, implement complex transformations and controls in robotic. Research has shown that using these algebras can lead to significant advancements in robotic perception, control, and trajectory optimization.

To sustain stability and performance, robot functioning in dynamic situations need to adjust effectively. In this process, Special Euclidean Algebra in two-dimensional ($\mathfrak{se}(2)$) holds significant relevance for regulating planar motion through the management of both linear and angular velocities. Accuracy and responsiveness of robotic systems are greatly increased when Lie algebra is used to develop controllers, such as PID controllers, which improves task efficiency. Moreover, strong body motions and robot kinematics are the main topics of foundational works relating to the work of Selig in [5], which offer a comprehensive introduction to the usage of Lie groups and algebras in robotics. Armanini et al. [6] investigated the dynamics modeling of soft manipulators using a discrete Cosserat approach. Recent developments have increased the use of Lie algebra in robotics even more. Research by Raj et al [7] explained these algebraic criteria for stability of switched system and Gallo et al [8] used specific Lie algebras to describe rigid body rotation and motion.

Extending the theoretical groundwork in this area, Kousar et al. [9] investigated the building of nilpotent and solvable Lie algebras in image fuzzy environments. Applications in practice include intelligent controllers for wheeled mobile robot [10], formation tracking of multi-robot systems [11], and differential invariants for $SE(2)$ [12]. Numerous research, including those on mobile manipulators with $n-$link prismatic arms [13], Lie algebra applications in mobile robot control [14], and the stability of switched systems [15], [16], have addressed the difficulties of managing car-like mobile manipulators as well as non-holonomic systems.

The incorporation of Lie algebra with advanced control and estimation methods has been a focal point in recent research. For example, analyzing higher-order kinematics in multibody systems using nilpotent algebra [17]. State estimation methods, such as range-only pose estimation [18] and motion estimation on Lie groups [3], also show the importance of Lie algebra in improving robotic perception and control. Additionally, the exploration of symmetry-preserving robot perception and control through geometry and learning [19] points to the convergence of Lie algebraic methods with machine learning methods. The incorporation of Lie algebra with sensor fusion and perception systems is a prominent theme in developing robotic advances. Research in multi-sensor-based control shows the increasing significance of these methods in creating more user-friendly human-robot interfaces. Following this discussion, motion, and control of a dual-arm car-like robot using an embedded FPGA-based system [20] along with place recognition through LiDAR scans [21], [22]. Novel uses of Lie algebra include low-cost landmine defusal robot [23], robotic arm movements and body frame for social robot [24], which signifies the promise of the field for solving practical problems. These developments point to a wider trend of Lie algebra being combined with other mathematical methods. The process is shown by research on magnetic soft robot for medical purposes and continuum robot [9], as well as the investigation of robotic dynamics on Lie groups [25], [26]. In addition, the research investigates how the control systems of educational robots such as mBot2 can be improved to work better in STEM-based learning environments by incorporating Lie algebra ideas, namely $\mathfrak{se}(3)$. The model looks at the challenge fixes for incorporating Lie algebra-based inverse kinematics into robotic arms, mBot2, for object manipulation. The benefits of these controllers over conventional methods for improving robot effectiveness in dynamic contexts are also examined in this research. The finding also considers how these incorporations can help with the teaching of Lie algebra and algebraic structures in contextual perspective.

II. Materials and Method

The theoretical foundations of this research originated from the application of computational algebra methods, such as Groebner basis, and Lie algebra to robotic systems. These mathematical frameworks offered excellent tools for characterizing robot dynamics, kinematics, and control, enabling more accurate motion planning as well as troubleshooting in complex form. A non-empty set $\mathfrak{g}$ completed with binary operation $[.,.]: \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ formed a Lie algebra when satisfied the following properties [27], [28], [29]:

1. Linearity: For every x, y, z in $\mathfrak{g}$ and for every scalar a, b, Lie bracket satisfied:
$$[ax + by, z] = a[x, z] + b[y, z]$$
$$[z, ax + by] = a[z, x] + b[z, y]$$
2. Alternative: For every x in $\mathfrak{g}$, $[x, x] = 0$.
3. Jacobi identity: For every $x, y, z \in \mathfrak{g}$ satisfy
$$[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$$

The analysis and synthesis referred to [30], [31], [32], [33], [34], [35], [36], [37], [38] to explain some Lie algebra and related concepts. In this research, the analysis implemented Lie algebras $\mathfrak{se}(2)$ and $\mathfrak{se}(3)$. Lie algebra $\mathfrak{se}(2)$ is associated with Lie group $SE(2)$, which represented the group of rigid body transformations in two-dimensional space (i.e., rotations and translations). Moreover, an element of $\mathfrak{se}(2)$ is represented as a 3×3 matrix of the form:

$$\begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix},$$

where ω represented the angular velocity, v_x is the linear velocity in the x direction and v_y represented the linear velocity in the y direction [39]. As previously mentioned, every Lie algebra has basis, and this basis of $\mathfrak{se}(2)$ consisted of the following three matrices:

$$E_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; E_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} E_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Lie Algebra $\mathfrak{se}(3)$ is associated with Lie group $SE(3)$, which represented the group of rigid body transformations in three-dimensional space (i.e., rotations and translations). An element of $\mathfrak{se}(3)$ is represented as 4×4 of the form:

$$\begin{bmatrix} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & \omega_x & v_y \\ -\omega_z & \omega_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where $\omega_x, \omega_y, \omega_z$ (v_x, v_y, v_z) represented the angular velocity vector (respectively, the linear velocity vector) [40]. Following the discussion, the basis of $\mathfrak{se}(3)$ consisted of the following six matrices:

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; G_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; G_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Lie Algebra $\mathfrak{se}(2)$ and $\mathfrak{se}(3)$ have been used widely in the robotic areas, where $\mathfrak{se}(2)$ is perfectly used to analyze robot car motion that worked on the ground. However, $\mathfrak{se}(3)$ is used to analyze rigid motion in three-dimensional in the global system coordinate. Some previous research used the concept of Lie algebra to optimize robot motion, namely [3], [4]. Following this process, certain open-source packages were also provided. For example, manif and navlie which were used by following the provided instruction link https://github.com/artivis/manif [3] and https://github.com/decargroup/navlie [4], respectively. Both packages were provided in two types of language programming, that were for C++ robot car Arduino based and python robot car Raspberry pi. Moreover, some educational robot producers, had currently developed another control board, namely, cyberpi.

During the analysis, the control board is regularly implemented on mBot2, a robot produced by makeblock. This motivated the research question, how to implement a concept of Lie algebra especially $\mathfrak{se}(3)$ in robot car with arm motion planning research case, mBot2. Theoretical, computational and practical differences between Lie Algebra $\mathfrak{se}(2)$ $\mathfrak{se}(3)$ are shown in Table I. Consequently, the concept of radicals in a ring had also developed in the context of Lie algebra. The research referred to [41], [42] for a class of rings where γ is defined as a radical class, as described in the sense of Kurosh and Amitsur. The ideal $\gamma(R)$ of a ring R is the largest ideal of R which belongs to γ [41]. In addition, ring R is called a γ −semisimple when $\gamma(R) = 0$ [42]. In the context of Lie algebra, the radical of a $\mathfrak{g}$ is defined as the largest solvable ideal. Analogous to the concept in ring theory, Lie algebra $\mathfrak{g}$ is termed semi simple when and only when the radical of $\mathfrak{g}$ is 0 [43]. Additionally, the concept of nilpotency in Lie algebra is also found, such as the concept of nilpotency in radical ring theory. The significance concepts of nilpotency and radicals in radical ring theory, which is certainly generalized in the framework of Lie theory, is applied as a mechanism for rigid motion represented by a Lie algebra, ensuring greater stability as shown by the results in the investigations [44], [45], [46], [47]. A major contribution and importance of semi simple rings in theoretical algebra is described in [48]. Furthermore, an in-depth analysis provided sufficient condition for a system to be stable as shown in [44]. The analysis implemented switched Differential Algebraic Equations (DAEs), where this sufficient condition is met under certain conditions including the radical and compact of Lie algebra, as shown in Table I.

TABLE I
THEORETICAL AND PRACTICAL DIFFERENCES BETWEEN $\mathfrak{se}(2)$ AND $\mathfrak{se}(3)$

Aspect	$\mathfrak{se}(2)$	$\mathfrak{se}(3)$
Dimensionality of	3 (2 −Dimensions + translation)	6 (3 −Dimensions + translation)
Solvability of	Yes	No
Radical of	$\mathfrak{se}(2)$	$\mathbb{R}^3$
Nilpotent elements of	Translation in 2D	Translation in 3D
Implementation of	2D rigid motion	3D rigid motion

During the analysis, a computational method is also used to analyze mBot2 with arm motion. This research generated a mathematical model which perfectly showed mBot2 with arm motion. The most important factor in generating this model is the position of end-effectors of mBot2. Moreover, mBot2 which is an example of robot car with arm produced by makeblock, is shown in Figure 1. The analysis researched mBot2 mathematical as well as computational property motion and implemented the concept of Lie Algebra $\mathfrak{se}(3)$ in its python-based language platform. Following the discussion, Table II showed robot car specification of Makeblock mBot2.

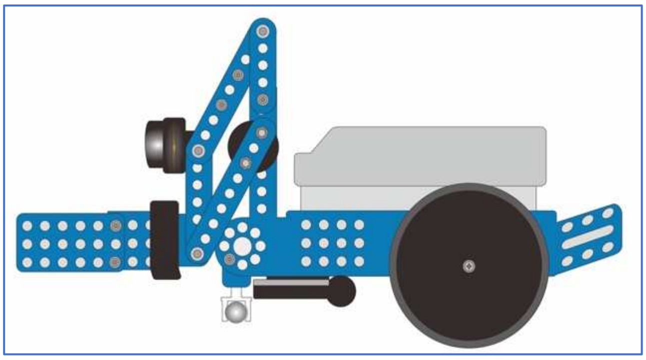

Fig. 1 Differential Wheel Robot Car with Arm

TABLE II
MAKEBLOCK MBOT2 SPECIFICATION

Specification	mBot2
Dimension $\boldsymbol{l} \times \boldsymbol{w} \times \boldsymbol{h}$ in cm (robot only) $\boldsymbol{l} \times \boldsymbol{w} \times \boldsymbol{h}$	$28 \times 13 \times 8$
Number of the arm segments	Two
Control Board	CyberPi
Processor	ESP32-WROVER-B
Programming Language	Block-Programming (Scratch Based), Python-Based
Number of Joints	One
Arm Rotation	No
Sensor	Ultrasonic and RGB Color
Wireless communication	Bluetooth, WIFI, WIFI LAN

As supporting data, this paper presents several novel theoretical results. These include mathematical models of the movement of armed robots, particularly mBot2. Additionally, experimental data compares the movement of mBot2 with two distinct types of programs: one that utilizes Lie Algebra and another that does not. However, this research does not compare mBot2 with programming algorithms other than these two.

III. RESULTS AND DISCUSSION

As motivated by the research of [49], [50], [51], [52], the positional variables of robot car with arm described in Figure 1 are represented as $G_{pos} = SE(2) \times SO(2)_{1st} \times SO(2)_{2nd}$. Where $SO(2)_{1st}$ is a special orthogonal in two-dimension explaining the rotation performed by the joint between anchored arm and the arm containing end-effector of mBot2. Additionally, $SO(2)_{2nd}$ represented a special orthogonal in two-dimension explaining the rotation performed by end-effector (hand of mBot2) which moved asymmetrically. The position of rigid body of grounded robot car is represented as Lie group $SE(2)$. The process followed from the arm of mBot2 structure and theoretical concept explained in [49], having two joints, namely X_1, X_2. The detail of the system is described in the following equations:

$$X_1 = blockdg\{\theta_1, \omega_1, \alpha_1\}$$

$$X_2 = blockdg\{\theta_2, \omega_2, \alpha_2\}$$

Simplified to

$$X_k = blockgd\{\theta_k, \omega_k, \alpha_k\} \in G \text{ where } k \in \{1,2\}$$

$$\theta_k = blockdg\{\theta_k^1, \theta_k^2\} \in G_{pos}$$

$$\omega_k = blockdg\{\omega_k^1, \omega_k^2\} \in \mathbb{R}^{p_1} \times \mathbb{R}^{p_2}$$

$$\alpha_k = blockdg\{\alpha_k^1, \alpha_k^2\} \in \mathbb{R}^{p_1} \times \mathbb{R}^{p_2}$$

where θ_k^i is the position of the i-th joint, ω_{ki} represents the velocity of the i-th joint, and $\alpha_{ki} \in \mathbb{R}^{p_i}$ is the acceleration of the i-th joint. Following this process, the measurement update step is calculated as:

$$\mu_{k+1} = \mu_{k+1|k}\, e_G^{\nu_{k+1}}$$

$$P_{k+1} = \mathcal{G}(\nu_{k+1})(I - K_{k+1}H_{k+1})P_{k+1|k}\mathcal{G}(\nu_{k+1})^T$$

Previous research derived a constant acceleration motion model and the necessary Jacobian H_{k+1} to perform motion estimation using optical markers, gyroscopes, as well as accelerometers as measurements [15]. In this analysis, $\alpha_{ki} \in \mathbb{R}^{p_i}$ is the acceleration of the i-th joint, $n = 2$ represented the number of joints the body had, while p_i is the number of degrees of freedom included in the i-th joint. Euclidean vector is viewed as a pure translation and is an element of a subgroup of $SE(n)$ [16]. To include velocities and accelerations in the state $X_k \in G$, the analysis used individual matrix representation obtained by simple matrix embedding. Robot car with arm is represented as a rigid body movement in $\mathbb{R}^3$, with P being the set of all robot positions. Subsequently, the set of all end-effector robot positions is a subset of $\mathbb{R}^3$, signifying $P \subset \mathbb{R}^3$ with the fixed of height.

Lemma 1

The joint X position is represented as (x, y, z) where x is the coordinate of end-effector of mBot2 on the x −axes and y represented the coordinate of robot on the y −axes. Additionally, z is the distance between ground and end-effector of mBot2.

Proof. It is obvious.

The analysis explained the theoretical background including the kinematics explanations and an example of theoretical implementation. This section started with theoretical background including robot car should move on the ground. Some previous research analyzed robot car motion which worked on the plane. The investigations used the concept of Lie Algebra $\mathfrak{se}(2)$ (special Euclidean in two-dimension) in individual work [12], [53]. However, this research used a robot car completed with an arm consisting of more than one arm segment. End-effector of robot car coordinate is represented as a position in three-dimensional global coordinate. Consequently, the analysis assumed that the coordinate of end-effector of robot car completed with arm is in the three-dimensional global coordinate. Where P is the set of all the robot coordinates, $P \subset \mathbb{R}^3$, and $\mathbb{R}^3 = \{(x, y, z) | x, y, z \in \mathbb{R}\}$. Since robot car worked on the ground with the fixed height of the anchored arm segment. The height of robot car is measured from an anchored joint to the ground.

Based on this condition, the pose of end-effector is represented by a homogeneous transformation matrix $T \in SE(3)$, where SE(3) is Lie group in the three-dimensional coordinate system, which combined rotation matrix $R \in SO(3)$ and translation vector $t \in \mathbb{R}^3$. The process followed from [5] that T is represented as $T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$, where R is a 3×3 rotation matrix representing the orientation and t signified a 3×1 vector representing the position. Moreover, the kinematics of robot car with arm included, 1. Forward, 2. Velocity, and 3. Inverse kinematics that are later explained. The pose T of end-effector is viewed as a sequence which terminated on finite number of poses. Therefore, the pose T is shown as $T = \prod_{\lambda \in \mathbb{N}} T_\lambda$, where each $T_\lambda, \lambda \in \mathbb{N}$ is a transformation matrix corresponding to λ −th joint and it is expressed as $T_\lambda = \begin{bmatrix} R_\lambda & t_\lambda \\ 0 & 1 \end{bmatrix}$, as R_λ represented the rotation due to λ −th joint, and t_λ is translation due to λ −th joint. The twist $\xi \in \mathfrak{se}(3)$ representing the velocity of end-effector had the form $\xi = \begin{bmatrix} v \\ \omega \end{bmatrix}$, where v is the linear velocity and ω is the angular velocity. Following this discussion, the twist is mapped to a 4×4 matrix $\hat{\xi} = \begin{bmatrix} \Omega & v \\ 0 & 0 \end{bmatrix}$. End-effector velocity is given by $\dot{T} = \hat{\xi}T$, where $\hat{\xi}$ is the matrix representation of the twist. The joint angles satisfied $T(\theta_1, \theta_2, \dots, \theta_n) = T_d$ in moving end-effector to a desired pose T_d, This process is typically done using numerical methods or iterative algorithms, where the concept of twists and exponentials of $se(3)$ matrices (using the matrix exponential) played a crucial role. To move end-effector of robot arm to a desired pose T_d, joint angles $\theta_1, \theta_2, \dots, \theta_n$ is located, satisfying the following equation $T(\theta_1, \theta_2, \dots, \theta_n) = T_d$, where, $T(\theta_1, \theta_2, \dots, \theta_n)$ represented the series of forward kinematics of robot, which is the transformation matrix that defined the pose of end-effector given the joint angles. The objective of this process is to find the joint angles that result in end-effector reaching the desired pose T_d.

Example 2.

The analysis implemented the method used for specific cases, especially where robot car had arm with two segments. The car moved on a two-dimensional plane with an SE(3) transformation describing its pose of end-effector. Following the process, the arm with two segments had one in a fixed position which could not be moved. The general transformation of end-effector is the product of transformation of the car and two arm segments. $T(\theta, \psi)$ is the product of base transformation (car) and exponentials of the twists corresponding to the joints. Given a desired end-effector pose T_d, the objective is to determine θ_1 and θ_2 by using numerical methods. The process started with an initial guess for the angles and iteratively updated the angles using Jacobian or optimization methods until $T(\theta_1, \theta_2) = T_d$. During this process, the kinematic model of the mobile robot is signified by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

where $\dot{x} = \frac{dx}{dt}, \dot{y} = \frac{dy}{dt}, \dot{z} = \frac{dz}{dt}, \dot{\theta} = \frac{d\theta}{dt}$, θ is the robot car orientation, v represented the linear velocity of the car, and ω is the angular velocity of the car. The outcome showed that $\dot{z} = \frac{dz}{dt} = 0$ since the distance between the anchored joints to the ground is a fixed constant as previously mentioned. In real-life implementation, robot is designed uniquely based on it purposes. Robot cars with arms are commonly designed for solving some tasks including moving a specific object and then putting the object in a specific or designed place. Additionally, robot movement is managed manually by using human control system remotely or autonomous system control. Robot cars detect the object in an autonomous system by using a specific sensor and measure the distance between its coordinate and object coordinate. In the following lemma, the analysis recalled the distance between two points in three-dimensional cartesian coordinate. As inspired by the analysis of [54], [55], [56], this research presented the following property.

Lemma 3 [54]

$C = [x\ y\ z]^T$ is the coordinate of robot and $C_p = [x_p\ y_p\ z_p]^T$ represented the purpose coordinate of robot. Additionally, the distance between C and C_p is given by $d_{rp} = \sqrt{[C - C_p][C - C_p]^T}$.

Proof.

$C = [x\ y\ z]^T$ is the coordinate of robot and $C_p = [x_p\ y_p\ z_p]^T$ represented the purpose coordinate of robot. The process followed from the basic geometric property that:

$$d_{rp} = \sqrt{(x - x_p)^2 + (y - y_p)^2 + (z - z_p)^2}$$

$$d_{rp} = \sqrt{(x - x_p)(x - x_p) + (y - y_p)(y - y_p) + (z - z_p)(z - z_p)}$$

$$d_{rp} = \sqrt{\begin{bmatrix} x - x_p & y - y_p & z - z_p \end{bmatrix} \begin{bmatrix} x - x_p \\ y - y_p \\ z - z_p \end{bmatrix}}$$

$$d_{rp} = \sqrt{[x - x_p \quad y - y_p \quad z - z_p][x - x_p \quad y - y_p \quad z - z_p]^T}$$

$$d_{rp} = \sqrt{[C - C_p][C - C_p]^T}$$

During the working operation, robot placed end-effector built in its arm to the accurate position or coordinate. In the next lemma, the analysis scrutinizes end-effector coordinate, and we assumed that robot car had two segments of its arm with specific segment-lengths, namely l_1 and l_2.

Theorem 4.

Robot car used with arm consisted of two arm segments, namely l_1 and l_2. When the arm segment with the length l_1 is anchored on the body of robot and assumed that the joint only moved two-dimensional rotation, then the end-effector coordinate (x_c, y_c, z_c) are given by

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} l_2 \cdot \sin\psi \cdot \cos\theta \\ l_2 \sin\psi . \sin\theta \\ l_1 - l_2 \cos\psi \end{bmatrix},$$

where θ is the robot orientation, and ψ represented the degree of the first and second arm segment.

Proof.

Figures 2 and 3 show the image of mBot2 orientation with arm movement to prove the theorem. Since robot car moved on the ground and end-effector coordinate in two-dimensional perspective, the coordinate of end-effector (x, y, z) is the projection on (x, y).

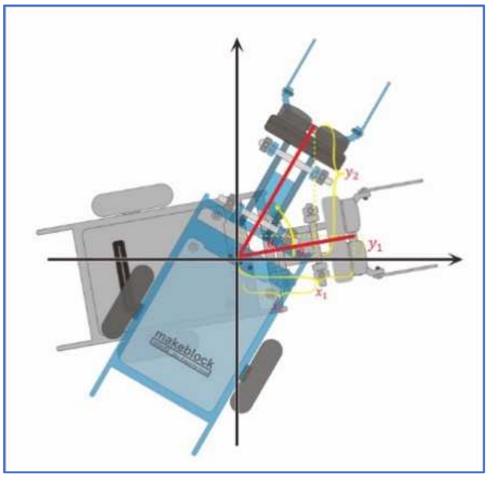

Fig. 2 mBot2 Orientation with Arm Movement

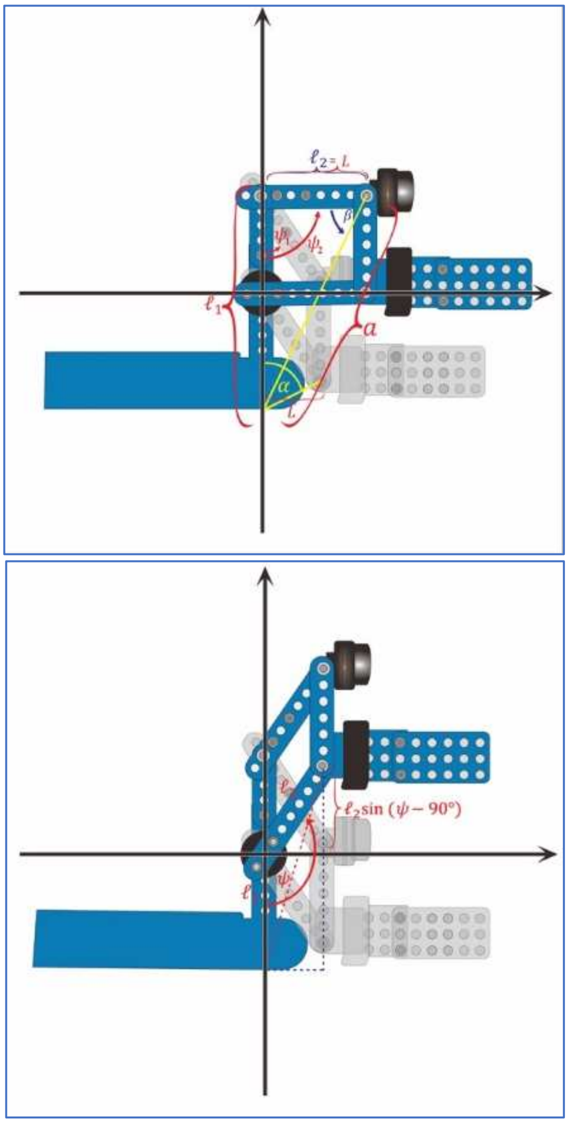

Fig. 3 Arm Movement Image of mBot2

The process is understandable by using trigonometry property that:

$$x = L\cos\theta$$
$$x = L\sin\theta$$

where L is the length of projection of second arm segment of l_2. When the angle between first and second arm segment is $\psi = 90^0$, then length of L reached the maximum, which is $L = l_2$. Moreover, the analysis had

$$L = l_2 \cos\psi$$

The z −position of end-effector is determined by

$$\begin{aligned} z \quad &= l_1 + l_2\sin(\psi - 90^0) \\ &= l_1 + l_2(\sin\psi\,.\cos 90^0 - \cos\psi\,.\sin 90^0) \\ &= l_1 + l_2\sin\psi\,.\cos 90^0 - l_2\cos\psi\,.\sin 90^0 \\ &= l_1 + l_2\sin\psi\,.(0) - \cos\psi\,.(1) \\ z \quad &= l_1 - l_2\cos\psi \end{aligned}$$

The research therefore has

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} l_2 \cdot \sin\psi \cdot \cos\theta \\ l_2.\sin\psi\,.\sin\theta \\ l_1 - l_2\cos\psi \end{bmatrix},$$

where θ is the robot orientation, and ψ represents the degree of first and second arm segment. As a direct consequence of Theorem 2, the analysis has the following corollary.

Corollary 5.

During the process, (x, y, z) is the coordinate of the end-effector. Moreover, the hand system of robot car with arm is given by:

$$\begin{gathered} x = l_2.sin\,\psi \cdot cos\,\theta \\ y = l_2\,sin\,\psi\,.sin\,\theta \\ z = l_1 - l_2\,cos\,\psi \\ 0 = c^2 + s^2 - 1 \\ 0 = c_1^2 + s_1^2 - 1 \end{gathered}$$

where $c = cos\,\theta\,, c_1 = cos\,\psi\,, s = sin\,\theta\,, s_1 = sin\,\psi.$

Proof.

The process followed from Lemma 2 that the coordinate of end-effector is given by:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} l_2 \cdot \sin\psi \cdot \cos\theta \\ l_2.\sin\psi\,.\sin\theta \\ l_1 - l_2\cos\psi \end{bmatrix}$$

Since the arm grabbed the object with the coordinate (a, b, c), therefore the analysis had

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_2.\sin\psi \cdot \cos\theta \\ l_2.\sin\psi\,.\sin\theta \\ l_1 - l_2\cos\psi \end{bmatrix}$$

Following this process

$$\begin{gathered} x = l_2.s_1 \cdot c \\ y = l_2.s_1.s \\ z = l_1 - l_2 c_1. \end{gathered}$$

Since

$$\begin{gathered} \sin^2\psi + \cos^2\psi = 1 \\ \sin^2\theta + \cos^2\theta = 1, \end{gathered}$$

the analysis therefore had the hand system of end-effector for the given coordinate (a, b, c)

$$\begin{gathered} x = l_2.\sin\psi \cdot \cos\theta \\ y = l_2\sin\psi\,.\sin\theta \\ z = l_1 - l_2\cos\psi \\ 0 = c^2 + s^2 - 1 \\ 0 = c_1^2 + s_1^2 - 1 \end{gathered}$$

where $c = \cos\theta\,, c_1 = \cos\psi\,, s = \sin\theta\,, s_1 = \sin\psi$.

A. Computational Exploration

The process followed from Theorem 3 in the previous section, as the hand system is shown as

$$\begin{gathered} x = l_2.\sin\psi \cdot \cos\theta \\ y = l_2\sin\psi\,.\sin\theta \\ z = l_1 - l_2\cos\psi \\ 0 = c^2 + s^2 - 1 \\ 0 = c_1^2 + s_1^2 - 1 \end{gathered}$$

where $c = \cos\theta\,, c_1 = \cos\psi\,, s = \sin\theta\,, s_1 = \sin\psi$. The research needed the concept of Gröbner basis to solve the hand system. During the process, the analysis referred to [57] for researching the concepts of the basis. In this section, the research explained how this base is deployed to solve the problem computationally step by step as shown in Table III.

TABLE III

GRÖBNER BASIS FOR FINDING HAND SYSTEM SOLUTION

START ALGORITHM	
Inputs	$x, y, z, \psi, \theta, l_1, l_2, s, c, s_1, c_1$ (Initial values and variables)
Steps	Identify and Set Up Polynomial Variables: Identify all relevant variables: $x, y, z, \psi, l_1, l_2, s, c, s_1, c_1$ → **Output:** A list of polynomial variables Substitute Trigonometric Functions: a. Replace trigonometric function sin and cos with polynomial variables b. $\sin\theta = s, \cos\theta = c$ with constraint $s^2 + c^2 = 1$ c. $\sin\psi = s_1, \cos\psi = c_1$ with constraint $s_1^2 + c_1^2 = 1$ → **Output:** A transformed system of polynomial equations Construct the Polynomial Ideal Generated by the polynomial equations → **Output:** The polynomial ideal associated with the system Compute Gröbner Basis: → **Output:** The Gröbner Basis of the Ideal Analyze the Gröbner Basis: a. Analyze the Gröbner basis to identify simplified forms of the equations. b. Use the basis to solve the system step-by-step → **Output:** Solutions Interpret the Results: Understand the implications of the Gröbner basis on the system of equations, such as finding specific solutions or determining the nature of the solution space. → **Output:** Final solutions
END ALGORITHM	

Lemma 6

The constructed ideal from the hand system

$$x = l_2 . \sin\psi \cdot \cos\theta$$
$$y = l_2 \sin\psi . \sin\theta$$
$$z = l_1 - l_2 \cos\psi$$
$$0 = c^2 + s^2 - 1$$
$$0 = c_1^2 + s_1^2 - 1$$

Is $I = \langle x - l_2 s c_1, y - l_2 s s_1, z - l_1 + l_2 c, c_2 + s_2 - 1, c_1^2 + s_1^2 - 1 \rangle$

The analysis had the following property in researching for the shape of mBot2.

Lemma 7

During the process, (x, y, z) is the ultrasonic coordinate. Subsequently, the coordinate of the hand is $(x + 5\cos\theta, y + 5\sin\theta, z - 5)$, where θ is mBot2 orientation.

Proof.

Let (x_1, y_1, z_1) be the coordinate of the hand during the analysis of the research. Therefore,

$$\cos\theta = \frac{x_1 - x}{5} \Rightarrow x_1 = x + 5\cos\theta$$
$$\sin\theta = \frac{y_1 - y}{5} \Rightarrow y_1 = y + 5\sin\theta$$
$$z_1 = z - 5$$

Hence, $(x_1, y_1, z_1) = (x + 5\cos\theta, y + 5\sin\theta, z - 5)$, where θ is mBot2 orientation.

Using mBot2 mechanical property, the application of property explained in Lemma 4 and 5, as well as computational processes by using Google Colab, is shown in Table IV which included computational results.

TABLE IV

COMPUTATIONAL RESULTS ON MBOT2 END-EFFECTOR

End-Effector Coordinate		Angle ψ (in degree)	Robot Orientation θ (in degree)
Ultrasonic	**Hand**		
(10,8,10)	(11,8,5)	78.9538025854616	0
(10,8,11)	(11,8,6)	83.31936762404638	0
(10,8,12)	(11,8,7)	87.76411130411034	0
(10,8,12.5)	(11,8,7.5)	90	0

B. Experimental Exploration

During this section, the analysis explained the experimental implementation of Lie Algebra $\mathfrak{se}(3)$ by using python-based program in mBot2, which is robot car produced by Makeblock. Additionally, the research proposed the following algorithm to deploy the program in mBot2, as shown in Table V.

TABLE V

LINE TRACKING, GRAB AND MOVE ALGORITHM FOR MBOT2

Activity	Instruction
Initialization	Set base_power, kp, left_power, right_power to initial values. Define functions: 1. skew_symmetric(v): Generates skew-symmetric matrix. 2. apply_se3_transform(omega, v): Applies SE(3) transformation to velocity vectors. 3. apply_se3_servo_transform(omega, v, servo_angle): Applies SE(3) transformation to servo angles.
Startup	Show instructions for stopping, starting line-following, and checking color recognition.
Button A Pressed	Stop all processes and halt the robot.
Button B Pressed	1. Initialize base_power and kp. 2. Set servos to 90°. 3. Begin line-following loop: a. Adjust left_power and right_power based on track sensor readings. b. Detect intersections and randomly turn left or right. c. Recognize colors and execute corresponding actions: 1. Blue: Stop, search for an object, and pick it up. 2. Green: Stop, turn left, and load an item. 3. Other colors: Continue following the line.
Joystick Pressed (Check Color):	Stop all processes and print the detected color to the console.
Additional Functions:	1. Load Item: Open gripper, adjust servos. 2. Search for Object: Rotate and stop when an object is in 15 cm. 3. Pick Up Item (Left): Open and close gripper.

Activity	Instruction
	4. Turn Left/Right: Apply SE(3) transformation for turning.
	5. Open/Close Gripper: Apply SE(3) transformation to adjust servo angles.

mBot2 is managed practically in a specific working area as shown in Figure 4.

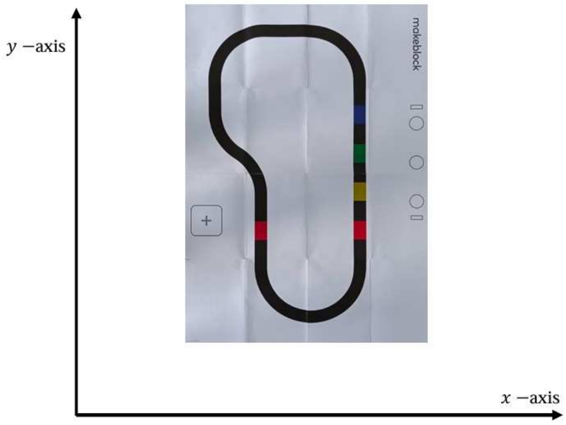

Fig. 4 Default Package Path Area for mBot2 provided by Makeblock

The mBot2 is designed to traverse the Path Area for a duration of up to 39 seconds. The optimal position and orientation of the mBot2 were defined accordingly (please see Table VI). The orientation angle is the angle formed by the orientation of mBot2, illustrated previously in Figure 2, corresponds to the x −axis and counterclockwise on the default package path area.

TABLE VI
THE IDEAL POSITION OR ORIENTATION OF MBOT2

Time	0	1	2	3	…	37	38	39
Degree	90°	120°	110°	90°	…	120°	100°	90°

After experimental observation, we finally obtained the following result, which elucidates the extent of the mBot2's capabilities. This is achieved by employing a program based on Lie algebra concepts in contrast to a program devoid of such concepts.

TABLE VII
THE POSITION OR ORIENTATION OF MBOT2

Time	0	1	2	3	…	37	38	39
With Lie	90°	120°	110°	95°	…	125°	100°	95°
Without Lie	90°	120°	115°	100°	…	100°	85°	80°

Based on the data presented in Tables VI and VII, the following table shows the degree variations between the ideal position or orientation of the mBot2 and the experimentally programmed trajectory achieved using Lie Algebra and without Lie Algebra.

Table VIII presents data indicating that the disparities between the ideal position or orientation of mBot2 and the mBot2 programmed using the Lie algebra concept are equal or smaller when employing the Lie algebra concept. We will provide a more detailed explanation of the results by illustrating the conditions depicted in Figures 5, 6, 7, and 8.

TABLE VIII
THE POSITION OR ORIENTATION OF MBOT2

Time	0	1	2	3	…	37	38	39
The difference With Lie	0°	0°	0°	5°	…	5°	0°	5°
The difference Without Lie	0°	0°	5°	10°	…	20°	15°	10°

Table VIII presents data indicating that the disparities between the ideal position or orientation of mBot2 and the mBot2 programmed using the Lie algebra concept are equal or smaller when employing the Lie algebra concept. We will provide a more detailed explanation of the results by illustrating the conditions depicted in Figures 5, 6, 7, and 8.

Both Lie Algebra program and the program without Lie Algebra concept were run three times in each experiment under the following parameters. The factors included mBot2 followed the path, stopping at the blue color, picking up the object in front of it without turning first, stopping again at the yellow color to place the object, and then following the path until the model completed a full lap. Following this discussion, Time Comparison with Lie Algebra $\mathfrak{se}(3)$ using the program without Lie algebra concept language platform is shown in Table IX.

TABLE IX
TIME COMPARISON

Activity	with Lie Algebra $\mathfrak{se}(3)$			using the program without Lie algebra concept language		
	I	II	III	I	II	III
Take the object	00:08:80	00:08:45	00:08:25	00:09:45	00:09:41	00:09:26
Put the object	00:14:35	00:12:45	00:12:28	00:14:51	00:15:31	00:15:70

The utilization of Lie Algebra in robotic control systems, as demonstrated in Table IX, significantly enhances task efficiency. Notably, it results in an average time reduction of approximately 11.7% across both activities, underscoring its superiority in executing precise and coordinated movements. As mBot2 with Lie algebra reached the measured points faster in Table V showed that programming with the concepts of Lie algebra allowed mBot2 to move more efficiently. The next experiment, which is motivated by the previous outcome, observed how orientation changes of mBot2 moved through the default track from Makeblock as shown in Figure 5. The model crossed the track, stopped at the blue color, rotated 360 degrees, caught the object, and stopped again at the yellow color. Moreover, mBot2 moved right to place the object and kept going until it completed a full lap. Figure 5 showed mBot2 orientation comparison during the movement of the model. During the process, mBot2 is set up with base power 20 RPM and the length of the track is 173,5 cm. The ideal orientation is defined beforehand by considering the shape of turns on the track including precise and ideal direction of mBot2.

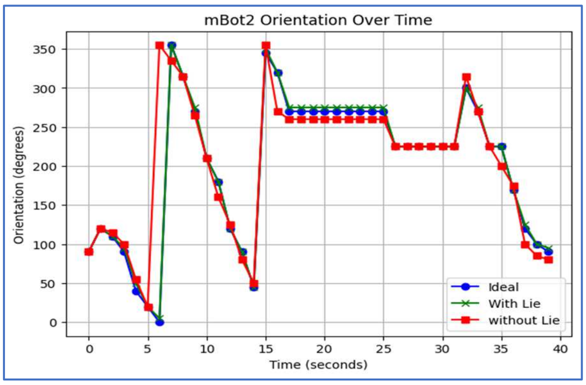

Fig. 5 mBot2 Orientation Over Time

The difference between ideal orientation and orientation of mBot2 programmed with the concept of Lie algebra is compared. This programmed pattern is also compared to the model programmed without the concept of Lie algebra, as shown in Figure 6.

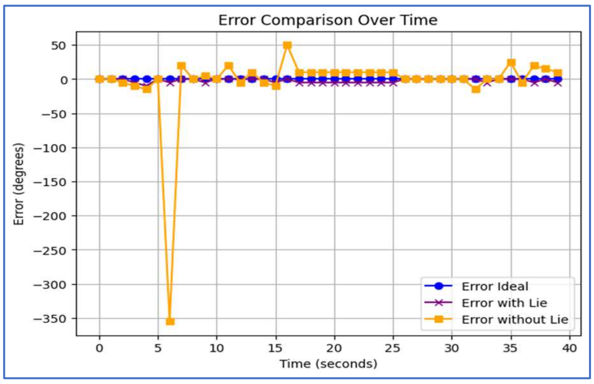

Fig. 6 Error Comparison

The ideal orientation of robot is 90 degrees in final pose at 39 seconds. Meanwhile, mBot2 with Lie algebra concept and the program without Lie algebra concept, the results were 95 and 80 degrees. These outcomes showed that mBot2 with Lie algebra had a smaller error margin. The condition is presented in the following graph, with a visual representation of mBot2 shown in Figure 7.

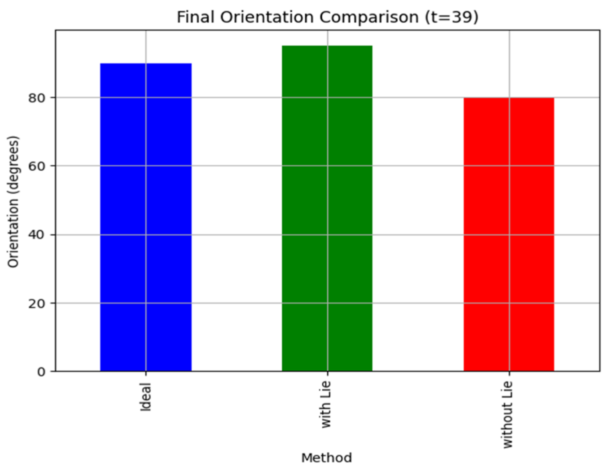

Fig. 7 Final Orientation of mBot2 ($t = 39$)

Despite the larger degree shown in Figure 7 (95^0), which deviates from the ideal degree position of 90^0 and the programming approach without utilizing the Lie algebra concept (80^0), the difference between the ideal degree and the degree achieved using the Lie algebra concept is significantly smaller, only 5^0compared to the difference without Lie algebra (10^0). Furthermore, considering the route direction depicted in Figure 8, mBot2 should be oriented with a greater angle than 90^0.

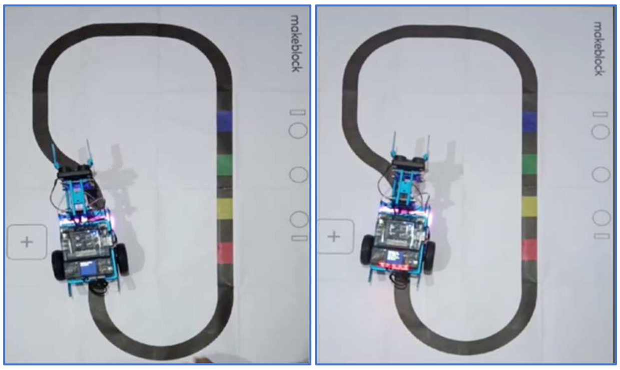

Fig. 8 The Final Pose Comparison

The research ensured the stability of the implementation of Lie Algebra $\mathfrak{se}(3)$, which represented rigid movements, specifically separating rotation and translation. The process is inspired by the reference [44] when it is implemented in switched differential-algebraic equations (more details is discussed about switched DAE in [44]). Moreover, the outcome is expected to motivate the development of robot hardware that supported theoretical concept in future investigation and construction, as well as the application of Lie algebra concepts in programming its automatic control systems. In addressing the issue, research should present several important definitions as described in this part. Several important definitions could be found in reference [44] and [7], which are explained in this section, including a linear Differential Algebraic Equation (DAE) given by $B\frac{dx}{dt} = Ax$, where $A, B \in \mathbb{R}^{n \times n}$ [7], [44]. Other definitions included Linear DAE as regular when $\det(sB - A)$ is a non-zero polynomial and linear DAE is stable when $\lim_{t\to\infty} \|x(t)\| = 0$ for all $x(0^-) \in \mathbb{R}^n$ [7], [44]. Consistency space (respectively, inconsistency space) is signified by $\mathcal{V}$ (resp, $\mathcal{W}$) and it is defined as the set of all $x(0^-) \in \mathbb{R}^n$. Moreover, the resulting solution is smooth, which is defined as the set of all $x(0^-) \in \mathbb{R}^n$ where the solution $x(t) = 0$ for all $t > 0$. A switched DAE is given by $B_\sigma \frac{dx}{dt} = A_\sigma x$ where σ is assumed to be piece-wise constant, right continuous function, and had finitely many discontinuities for any finite duration [7], [44]. The class of all non-zero infimum dwell time switching signals is defined as $S^* := \cup_{\tau \in \mathbb{R}^+} S^\tau \coloneqq \{\sigma \in S | \inf\{t_k - t_{k-1} | k \in \mathbb{N}\} > 0\}$, where S is the set of all such arbitrary switching signals [7], [44]. Following the discussion, implementing Theorem 1 in [44], the analysis therefore had the following property.

Proposition 8.

Consider switched DAE with a stable and regular DEA subsystem and suppose $E_i(I - \Pi_i)\Pi_j = 0$ for every i, j $\mathcal{P}$. Suppose $\{A_i^{diff} | i \in \mathcal{P}\}_{LA} = \mathfrak{so}(3) \oplus \mathbb{R}^3$, then switched DAE is globally stable for S^*, where $S^* := \cup_{\tau \in \mathbb{R}^+} S^\tau \coloneqq$

$\{\sigma \in S | inf\{t_k - t_{k-1} | k \in \mathbb{N}\} > 0\}$, and S is the set of all such arbitrary switching signals.

Proof. Consider Lie algebra $\mathfrak{se}(3)$, the outcome shows that $\mathfrak{se}(3) = \mathfrak{so}(3) \ltimes \mathbb{R}^3$ where $\mathfrak{so}(3)$ is a compact Lie algebra and $\mathbb{R}^3$ represented its radical. Let $\{A_i^{diff} | i \in \mathcal{P}\}_{LA} = \mathfrak{so}(3) \oplus \mathbb{R}^3$ where $\{A_i^{diff} | i \in \mathcal{P}\}_{LA}$ is shown as

$$A_1^{diff} = \left\{ \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} | a, b, c \in \mathbb{R} \right\} = \mathfrak{so}(3)$$

$$A_2^{diff} = \left\{ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} | x_1, x_2, x_3 \in \mathbb{R} \right\} = \mathbb{R}^3$$

where

$$\mathfrak{so}(3) \oplus \mathbb{R}^3 = \left\{ \begin{pmatrix} 0 & -a & b & x_1 \\ a & 0 & -c & x_2 \\ -b & c & 0 & x_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} | a, b, c, x_1, x_2, x_3 \in \mathbb{R} \right\}$$

$\{A_i^{diff} | i \in \mathcal{P}\}_{LA}$ satisfied conditions 1 and 2 in Theorem 1 [44]. Hence, switched DAE is globally stable for S^*, where $S^* := \bigcup_{\tau \in \mathbb{R}^+} S^\tau \coloneqq \{\sigma \in S | \inf\{t_k - t_{k-1} | k \in \mathbb{N}\} > 0\}$, and S is the set of all such arbitrary switching signals.

IV. Conclusion

In conclusion, the research applied the concept of Lie algebra to mBot2 mobile robot, developed by Makeblock, using its Python-based programming platform on CyberPi control board. During the analysis, movement model and Groebner basis have been obtained from computation of the hand system of mBot2 with its add-on arm. The experimental results also showed that programming with Lie algebra, particularly $\mathfrak{se}(2)$ and $\mathfrak{se}(3)$, led to more precise robot movement in orientation, allowing its movements to be more efficient when compared to the program without Lie algebra. The analysis shows that the average absolute error in the orientation of mBot2 is 2.25 degrees with Lie Algebra and 17.25 degrees without Lie Algebra. This indicates that the use of Lie Algebra reduces the error by 86.96% compared to methods without it. The significant improvement in accuracy and efficiency can be scientifically justified by Lie Algebra's ability to handle complex transformations and maintain consistency in coordinate mapping, making it highly suitable for precise robotic control systems like mBot2. Moreover, the existence of electrical circuit switches implemented in the control system of the robot signified its stability, especially for robot movements that are represented as $\mathfrak{so}(3) \oplus \mathbb{R}^3$.

Acknowledgment

This research is fully funded by DRTPM Number 107/E5/PG.02.00.PL/2024; 0609.12/LL5-INT/AL.04/2024; 100/PFR/LPPM-UAD/VI/2024. The authors are grateful to Directorate of Research, Technology and Community Service of Indonesia, the Institute of Research and Community Service of Universitas Ahmad Dahlan (LPPM UAD), Prof. Andrew Douglas, Ph.D (CUNY, USA); Prof. Sugiyarto, Ph.D and Alfian Ma'arif, M.T (UAD, Indonesia), and all parties who helped to manage this research.

References

[1] C. Muller, "World Robotics 2023," Frankfurt am Main, 2023. Accessed: Sep. 03, 2024. [Online]. Available: https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2023.pdf

[2] D. Georgieva and T. Georgieva-Trifonova, "Developing Mathematical Competencies Through Makeblock mBot Programming in Computer Modelling Education," *TEM Journal*, vol. 12, no. 4, pp. 2437–2447, Nov. 2023, doi: 10.18421/TEM124-56.

[3] J. Deray and J. Solà, "Manif: A micro Lie theory library for state estimation in robotics applications," *J Open Source Softw*, vol. 5, no. 46, p. 1371, Feb. 2020, doi: 10.21105/joss.01371.

[4] C. C. Cossette, M. Cohen, V. Korotkine, A. Del Castillo Bernal, M. A. Shalaby, and J. R. Forbes, "Navlie: A Python Package for State Estimation on Lie Groups," in *IEEE International Conference on Intelligent Robots and Systems*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 5282–5287. doi:10.1109/IROS55552.2023.10342362.

[5] J. M. Selig, "Lie Groups and Lie Algebras in Robotics," in *Computational Noncommutative Algebra and Applications*, J. Byrnes, Ed., 2004. [Online]. Available: http://www.wtv-books.de/nato-pco.htm

[6] C. Armanini, I. Hussain, M. Z. Iqbal, D. Gan, D. Prattichizzo, and F. Renda, "Discrete Cosserat Approach for Closed-Chain Soft Robots: Application to the Fin-Ray Finger," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2083–2098, Dec. 2021, doi:10.1109/TRO.2021.3075643.

[7] P. Raj and D. Pal, "Lie Algebraic Criteria for Stability of Switched Systems of Differential Algebraic Equations (DAEs)," *IEEE Control Syst Lett*, vol. 5, no. 4, pp. 1333–1338, Oct. 2021, doi:10.1109/LCSYS.2020.3036577.

[8] E. Gallo, "The SO(3) and SE(3) Lie Algebras of Rigid Body Rotations and Motions and their Application to Discrete Integration, Gradient Descent Optimization, and State Estimation," Universidad Politécnica de Madrid, Madrid, 2022. doi: 10.48550/arXiv.2205.12572.

[9] S. Kousar, S. Arshad, N. Kausar, and T. P. Hong, "Construction of Nilpotent and Solvable Lie Algebra in Picture Fuzzy Environment," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Dec. 2023, doi: 10.1007/s44196-023-00213-w.

[10] S. Mondal, R. Ray, S. R. N., and S. Nandy, "Intelligent controller for nonholonomic wheeled mobile robot: A fuzzy path following combination," *Math Comput Simul*, vol. 193, pp. 533–555, Mar. 2022, doi: 10.1016/j.matcom.2021.10.028.

[11] C. Wang, J. Ji, Z. Miao, and J. Zhou, "Formation tracking of multi-robot systems with switching directed topologies based on Udwadia-Kalaba approach," *Appl Math Model*, vol. 126, pp. 147–158, Feb. 2024, doi: 10.1016/j.apm.2023.10.035.

[12] M. Sangalli, S. Blusseau, S. Velasco-Forero, and J. Angulo, "Differential Invariants for SE(2)-Equivariant Networks," in *Proceedings - International Conference on Image Processing, ICIP*, IEEE Computer Society, 2022, pp. 2216–2220. doi:10.1109/ICIP46576.2022.9897301.

[13] R. Chand, S. A. Kumar, R. P. Chand, and S. Reddy, "A Car-like Mobile Manipulator with an n-link Prismatic Arm," in *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/CSDE53843.2021.9718408.

[14] P. Coelho and U. Nunes, "Lie algebra application to mobile robot control: A tutorial," *Robotica*, vol. 21, no. 5, pp. 483–493, Sep. 2003, doi: 10.1017/S0263574703005149.

[15] P. Xu, F. Long, Q. Wang, J. Tian, X. Yang, and L. Mo, "Finite-Time Stability for Caputo Nabla Fractional-Order Switched Linear Systems," *Fractal and Fractional*, vol. 6, no. 11, Nov. 2022, doi:10.3390/fractalfract6110621.

[16] C. M. Zagabe and A. Mauroy, "Switched nonlinear systems in the Koopman operator framework: Toward a Lie-algebraic condition for uniform stability," in *2021 European Control Conference (ECC)*, Delft: IEEE, 2021, pp. 281–288. doi:10.23919/ECC54610.2021.9655200.

[17] M. Ceccarelli, S. K. Agrawal, V. Glazunov, A. Hernández, J. Carlos, and J. Correa, *Advances in Service and Industrial Robotics*. Springer, 2023.

[18] A. Goudar, T. D. Barfoot, and A. P. Schoellig, "Continuous-Time Range-Only Pose Estimation," in *Proceedings - 2023 20th Conference on Robots and Vision, CRV 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 29–36. doi:10.1109/CRV60082.2023.00012.

[19] Z. Fan *et al.*, "A Cartesian-Based Trajectory Optimization with Jerk Constraints for a Robot," *Entropy*, vol. 25, no. 4, Apr. 2023, doi:10.3390/e25040610.

[20] R. Chand, R. P. Chand, M. Assaf, P. R. Naicker, S. V. Narayan, and A. F. Hussain, "Embedded FPGA-based Motion Planning and Control of a Dual-arm Car-like Robot," in *2022 IEEE 7th Southern Power Electronics Conference, SPEC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi:10.1109/SPEC55080.2022.10058252.

[21] L. Gui, C. Zeng, J. Luo, and X. Yang, "Place Recognition through Multiple LiDAR Scans Based on the Hidden Markov Model," *Sensors*, vol. 24, no. 11, Jun. 2024, doi: 10.3390/s24113611.

[22] H. Wang, Y. Mao, and J. Du, "Continuum Robots and Magnetic Soft Robots: From Models to Interdisciplinary Challenges for Medical Applications," Mar. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/mi15030313.

[23] H. Naidu and P. Ramtekkar, "An Innovative Affordable Robot to Defuse Landmines using IoT and Wireless Communication Technique to save precious life," in *International Conference on Emerging Trends in Engineering and Technology, ICETET*, IEEE Computer Society, 2023. doi: 10.1109/ICETET-SIP58143.2023.10151542.

[24] J. D. D. Ducut, I. C. Valenzuela, E. P. Dadios, and R. K. C. Billones, "Design and Development of Robotic Arm Movements and Body Frame for a Social Robot for Graduation Rites," in *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/HNICEM54116.2021.9731930.

[25] F. C. Park, J. E. Bobrow, and S. R. Ploen, "A Lie Group Formulation of Robot Dynamics," *Int J Rob Res*, vol. 14, no. 6, pp. 609–618, 1995, doi: 10.1177/027836499501400606.

[26] S. R. Ploen and F. C. Park, "A Lie group formulation of the dynamics of cooperating robot systems," *Rob Auton Syst*, vol. 21, no. 3, pp. 279–287, 1997, doi: 10.1016/S0921-8890(96)00802-0.

[27] B. C. Stacey, *Springer Briefs in Mathematical Physics Volume 41: A First Course in the Sporadic SICs*. 2021. [Online]. Available: http://www.springer.com/series/11953

[28] M. Majumdar and A. Bhattacharyya, *An introduction to smooth manifolds*. Springer Nature, 2023. doi: 10.1007/978-981-99-0565-2.

[29] K. H. Hofmann and S. A. Morris, *The Structure of Compact Groups*. Boston: Walter de Gruyter GmbH, Berlin/Boston, 2023. doi:10.1515/9783111172606.

[30] M. Ceccarelli, *Fundamentals of Mechanics of Robotic Manipulation*, vol. 112. in Mechanisms and Machine Science, vol. 112. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-90848-5.

[31] Y. Ren, "The BGG Category for Generalized Reductive Lie Algebras," *Frontiers of Mathematics*, vol. 19, no. 1, pp. 127–142, 2024, doi: 10.1007/s11464-021-0352-8.

[32] D. Gourevitch, E. Sayag, and I. Karshon, "Annihilator varieties of distinguished modules of reductive Lie algebras," *Forum of Mathematics, Sigma*, vol. 9, 2021, doi: 10.1017/fms.2021.42.

[33] Z. Chen and Y. Yu, "Biderivations and strong commutativity-preserving maps on parabolic subalgebras of simple Lie algebras," *Linear and Multilinear Algebra*, vol. 70, no. 14, pp. 2659–2671, 2022, doi: 10.1080/03081087.2020.1809621.

[34] M. Sahai and B. Sharan, "On the lower Lie nilpotency index of a group algebra," *Quaestiones Mathematicae*, vol. 44, no. 10, pp. 1347–1354, 2021, doi: 10.2989/16073606.2020.1795740.

[35] N. Torres Alberto, L. Joseph, V. Padois, and D. Daney, "A linearization method based on Lie algebra for pose estimation in a time horizon A linearization method based on Lie algebra for pose estimation in a time horizon A linearization method based on Lie algebra for pose estimation in a time horizon," 2022. [Online]. Available: https://hal.science/hal-03621688v1

[36] C. J. Bordin, C. G. De Figueredo, and M. G. S. Bruno, "Diffusion Particle Filtering on the Special Orthogonal Group Using Lie Algebra Statistics," *IEEE Signal Process Lett*, vol. 29, pp. 2058–2062, 2022, doi: 10.1109/LSP.2022.3210870.

[37] S. Goodarzi and A. Khotanloo, "Lie Centralizers and generalized Lie derivations on prime rings by local actions," *Commun Algebra*, vol. 51, no. 12, pp. 5277–5286, 2023, doi:10.1080/00927872.2023.2228429.

[38] M. Abdaoui, "2-Local superderivations on infinite-dimensional Lie superalgebras," *Commun Algebra*, vol. 52, no. 3, pp. 1124–1141, 2024, doi: 10.1080/00927872.2023.2257797.

[39] J. Zhai, Z. Geng, and J. Yang, "A Specified Time Obstacle Avoidance Control Strategy for Wheeled Mobile Robots," in *Proceedings of the 33rd Chinese Control and Decision Conference, CCDC 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 4692–4696. doi:10.1109/CCDC52312.2021.9602322.

[40] H. Zhang, D. Ye, Y. Xiao, and Z. Sun, "Adaptive Control on SE(3) for Spacecraft Pose Tracking with Harmonic Disturbance and Input Saturation," *IEEE Trans Aerosp Electron Syst*, vol. 58, no. 5, pp. 4578–4594, Oct. 2022, doi: 10.1109/TAES.2022.3165524.

[41] P. W. Prasetyo, H. Marubayashi, and I. E. Wijayanti, "On the restricted graded Jacobson radical of rings of Morita context," *Turkish Journal of Mathematics*, vol. 46, no. Special Issue 2, pp. 1985–1993, 2022, doi:10.55730/1300-0098.3246.

[42] G. J. Cooke and R. G. McDougall, "A generalization of the lower radical type construction for strict and base radical classes," *Commun Algebra*, 2024, doi: 10.1080/00927872.2024.2337279.

[43] H. V. Khánh, "Leavitt Path Algebras in Which Every Lie Ideal is an Ideal and Applications," *Transformation Groups*, 2024, doi:10.1007/s00031-024-09848-1.

[44] P. Raj and D. Pal, "On Stability of Switched Differential Algebraic Equations: A Decomposition-Based Approach," *IEEE Control Syst Lett*, vol. 8, pp. 358–363, 2024, doi: 10.1109/LCSYS.2024.3382612.

[45] P. J. McCarthy and C. Nielsen, "Global Sampled-Data Regulation of a Class of Fully Actuated Invariant Systems on Simply Connected Nilpotent Matrix Lie Groups," *IEEE Trans Automat Contr*, vol. 67, no. 1, pp. 436–442, Jan. 2022, doi: 10.1109/TAC.2021.3058053.

[46] Y. Chen and S. Trenn, "Stability analysis of switched nonlinear differential-algebraic equations via nonlinear Weierstrass form," 2022 *European Control Conference (ECC)*, pp. 1091–1096, Jul. 2022, doi:10.23919/ecc55457.2022.9838148.

[47] S. Lin and X. Liu, "Passivity and Control for Multiweighted and Directed Fractional-Order Network Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 4, pp. 1733–1746, Apr. 2023, doi: 10.1109/TCSI.2023.3239907.

[48] I. E. Wijayanti, M. Ardiyansyah, and P. W. Prasetyo, "On a Class of λ-Modules," *Ukrainian Mathematical Journal*, vol. 73, no. 3, 2021, doi: 10.1007/s11253-021-01931-0.

[49] V. Joukov, J. Cesic, K. Westermann, I. Markovic, I. Petrovic, and D. Kulic, "Estimation and Observability Analysis of Human Motion on Lie Groups," *IEEE Trans Cybern*, vol. 50, no. 3, pp. 1321–1332, Mar. 2020, doi: 10.1109/TCYB.2019.2933390.

[50] G. Li, Z. Li, C. Y. Su, and T. Xu, "Active Human-Following Control of an Exoskeleton Robot with Body Weight Support," *IEEE Trans Cybern*, vol. 53, no. 11, pp. 7367–7379, Nov. 2023, doi:10.1109/TCYB.2023.3253181.

[51] L. Chen *et al.*, "Research Ideas Discovery via Hierarchical Negative Correlation," *IEEE Trans Neural Netw Learn Syst*, vol. 35, no. 2, pp. 1639–1650, Feb. 2024, doi: 10.1109/TNNLS.2022.3184498.

[52] Q. Li, Y. Wang, and F. Lv, "Semantic Correlation Attention-Based Multiorder Multiscale Feature Fusion Network for Human Motion Prediction," *IEEE Trans Cybern*, vol. 54, no. 2, pp. 825–838, Feb. 2024, doi: 10.1109/TCYB.2022.3184977.

[53] X. Peng, Z. Sun, M. Chen, and Z. Geng, "Arbitrary Configuration Stabilization Control for Nonholonomic Vehicle With Input Saturation: A c-Nonholonomic Trajectory Approach," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1663–1672, Feb. 2022, doi: 10.1109/TIE.2021.3060674.

[54] J. Jin, Y. G. Kim, S. G. Wee, D. H. Lee, and N. Gans, "A Stable Switched-System Approach to Collision-Free Wheeled Mobile Robot Navigation," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 86, no. 3–4, pp. 599–616, Jun. 2017, doi:10.1007/s10846-017-0467-z.

[55] X. Li, Z. Xu, Z. Su, H. Wang, and S. Li, "Distance- and Velocity-Based Simultaneous Obstacle Avoidance and Target Tracking for Multiple Wheeled Mobile Robots," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 1736–1748, Feb. 2024, doi:10.1109/TITS.2023.3312373.

[56] A. Ait Ladel, A. Benzaouia, R. Outbib, M. Ouladsine, and E. M. El Adel, "Robust fault tolerant control of continuous-time switched systems: An LMI approach," *Nonlinear Analysis: Hybrid Systems*, vol. 39, Feb. 2021, doi: 10.1016/j.nahs.2020.100950.

[57] J. L. Awange, B. Paláncz, R. H. Lewis, and L. Völgyesi, *Mathematical Geosciences Hybrid Symbolic-Numeric Methods Second Edition*, 2nd ed. Springer, 2023. doi: 10.1007/978-3-030-92495-9.