Design Digital Twins Systems and Control Mobile Robot with Mecanum Wheels for Trajectory Tracking

Huu-Thanh Nguyen, Cong-Tien Dinh, Tan-Luy Nguyen
Faculty of Electrical & Electronics Engineering
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University, Ho Chi Minh City, VietNam
Email: {thanh.nguyenhuu8802, tien.dinhbachkhoa19, ntanluy}@hcmut.edu.vn

***Abstract*—This paper involves controlling four-mechanism-wheeled robot systems to follow a predetermined trajectory using a back-stepping technique and designing a Digital Twins (DT) system to read robot data and observe the real robot through a virtual counterpart in the digital world. DT is executed in simulated and real environments to minimize errors and optimize processing speed. The kinematic and dynamic models are designed, and the back-stepping technique is used to build the virtual and actual controllers. The simulation and experiment are conducted to verify the effectiveness of the proposed method.**

Keywords—Mecanum, back-stepping control, mobile robot, trajectory tracking, digital twins

I. INTRODUCTION

In need of flexibility, mecanum wheels allow mobile robots to move in any direction without the need to turn. This capability is highly beneficial in applications such as autonomous delivery services, indoor logistics, construction, manufacturing, healthcare, and environments with limited workspace. The mecanum robots can navigate obstacles and execute complex maneuvers effectively, making them versatile solutions for various industries [1-5]. The mecanum robots are usually used in cases requiring flexible movement, they can be controlled by various methods, such as remote control, computer programming, or applications programmed on mobile devices. In addition, the mecanum robot can be equipped with embedded computer systems, wireless communication systems, and automated control systems. These support the robots in performing complex tasks and interacting with its environment effectively [1-5].

In [1], the authors designed a controller for the mecanum robot and the digital twin system. However, the controller is based on the kinematic function of mecanum robot. To improve the controller, we designed the backstepping controller based on kinematics and dynamics [2]. Besides, we will clarify the method of locating mobile robot by ultra-wideband to send position data back to the backstepping controller [3]. There are various control techniques, such as PID control, Fuzzy control [4], and sliding mode control [5]. However, the backstepping control is completely suitable for this project. A mecanum mobile robot is a nonlinear dynamical model and has subsystems; therefore, a backstepping algorithm can be established from it.

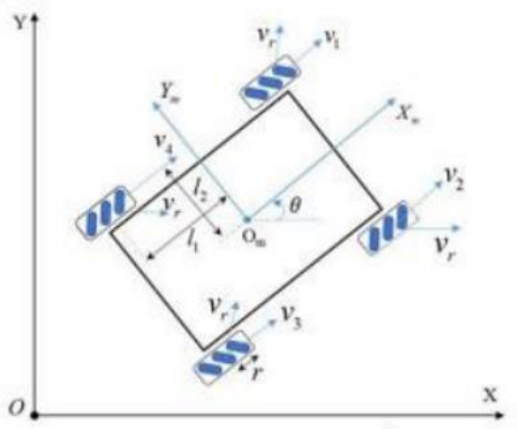

Fig.1. Global frame and robot frame

Observing the status of the robot is an essential task. In addition to reading data from sensors attached to the robot, we can see the real-time appearance of the robot via a virtual model on the Webots platform. In the digital environment, we used a mecanum model and synchronized it with the physical robot using a wireless protocol. If a trajectory for the virtual robot in Webots is set, the physical robot will follow the signal from the virtual one in real-time. If a robot in the real world stops or moves, the virtual robot will imitate its twin [1].

In this paper, we design a Digital Twins system for mecanum robots in virtual and actual environments. Then, we propose a backstepping control for robots to track trajectories as well as develop a user interface.

II. MATHEMATICAL THEORY

A. Kinematic of Mecanum Robot

Consider a model of a mecanum robot depicted in Fig. 1, the kinematics can be written as

$$\omega_i = J \cdot \dot{Q}_p \tag{1}$$

with the function matrix

$$J = \frac{1}{r}\begin{bmatrix} 1 & -1 & -(l_1+l_2) \\ 1 & 1 & (l_1+l_2) \\ 1 & -1 & (l_1+l_2) \\ 1 & 1 & -(l_1+l_2) \end{bmatrix}$$

TABLE I. PARAMETER TABLE

Symbol	Definition	Current
x	x in Global frame	m
y	y in Global frame	m
θ	The angle of the robot from the origin coordinates	rad
Q	Vector of robot position and angle in global frame	
x_p	Xp in robot frame	m
y_p	Yp in robot frame	m
Q_p	Vector of robot position and angle in robot frame	
ω_i	Angular velocity of each wheel	rad/s
r	Radius of wheel	m
$l1$	Distance from O_pY_p axis to wheel	m
$l2$	Distance from O_pX_p axis to wheel	m
μ	Coefficient of friction	
g	Gravity	m/s^2
I	Mass moment of inertia of robot	Kg.m^2
Ib	Mass moment of inertia of wheel	Kg.m^2
N_i	Pressure force	N

where ω_i $(i = 1, \dots, 4)$ is angular velocity of each mecanum wheel and $\dot{Q}_p = [\dot{x}_p\ \dot{y}_p\ \dot{\theta}]^T$. The expression describing the correlation between the velocity of the robot in the global coordinate system and the coordinate system attached to the robot can be presented as

$$\dot{Q} = T(\theta)\dot{Q}_p \tag{2}$$

with the function matrix

$$T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\dot{Q} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$ and $\dot{Q}_p = [\dot{x}_p\ \dot{y}_p\ \dot{\theta}]^T$. From (2), the kinematic function for robot is written as:

$$\dot{Q}_p = T(\theta)^{-1}\dot{Q} \tag{3}$$

B. Dynamic of Mecanum Robot

The total kinetic energy of the system can be defined as [2]

$$E = \frac{1}{2}m\left(\dot{x}_p{}^2 + \dot{y}_p{}^2\right) + \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}I_b(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2) \tag{4}$$

The total kinetic and potential energy of the system is written as $L = E + T = E$ (Potential energy = 0), where

$$L = \frac{1}{2}\left(m\frac{r^2}{8} + I\frac{r^2}{16(l_1 + l_2)} + I_b\right)(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2) + \left(m\frac{r^2}{8} - I\frac{r^2}{16(l_1 + l_2)}\right)(\omega_1\omega_3 + \omega_2\omega_4) - I\frac{r^2}{16(l_1+l_2)}(\omega_1\omega_2 + \omega_1\omega_4 + \omega_2\omega_3 + \omega_3\omega_4) \tag{5}$$

Following the Euler-Lagrange equation [2], one obtains

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}_i}\right) - \frac{\partial L}{\partial \varphi_i} = F_i \tag{6}$$

where $\dot{\varphi}_i = \omega_i$ and F_i are values of the generalized force

$$\left(m + \frac{4I_b}{r^2}\right)(\ddot{x} + \dot{y}\dot{\theta}) = \frac{1}{r}[(F_1 + F_2 + F_3 + F_4)\cos\theta - (-F_1 + F_2 - F_3 + F_4)\sin\theta]$$

$$\left(m + \frac{4I_b}{r^2}\right)(\ddot{y} + \dot{x}\dot{\theta}) = \frac{1}{r}[(F_1 + F_2 + F_3 + F_4)\sin\theta + (-F_1 + F_2 - F_3 + F_4)\cos\theta]$$

$$\left(I + \frac{4I_b(l_1 + l_2)^2}{r^2}\right)\ddot{\theta} = \frac{l_1 + l_2}{r}[(-F_1 + F_2 + F_3 - F_4)] \tag{7}$$

From (7), we have the matrices as follows:

$$\ddot{Q} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix}$$

$$M = \begin{bmatrix} m + \frac{4I_b}{r^2} & 0 & 0 \\ 0 & m + \frac{4I_b}{r^2} & 0 \\ 0 & 0 & I + \frac{4I_b(l_1 + l_2)^2}{r^2} \end{bmatrix}$$

$$C(Q, \dot{Q}) = \begin{bmatrix} \left(m + \frac{4I_b}{r^2}\right)\dot{y}\dot{\theta} \\ -\left(m + \frac{4I_b}{r^2}\right)\dot{x}\dot{\theta} \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \cos\theta + \sin\theta & \cos\theta - \sin\theta & \cos\theta + \sin\theta & \cos\theta - \sin\theta \\ -\cos\theta + \sin\theta & \cos\theta + \sin\theta & -\cos\theta + \sin\theta & \cos\theta + \sin\theta \\ -(l_1 + l_2) & (l_1 + l_2) & (l_1 + l_2) & -(l_1 + l_2) \end{bmatrix}$$

$$F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} \tau_1 - \mu r N_1 sgn(\omega_1) \\ \tau_2 - \mu r N_2 sgn(\omega_2) \\ \tau_3 - \mu r N_3 sgn(\omega_3) \\ \tau_4 - \mu r N_4 sgn(\omega_4) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} - \begin{bmatrix} \mu r N_1 sgn(\omega_1) \\ \mu r N_2 sgn(\omega_2) \\ \mu r N_3 sgn(\omega_3) \\ \mu r N_4 sgn(\omega_4) \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}; \xi = \begin{bmatrix} \mu r N_1 sgn(\omega_1) \\ \mu r N_2 sgn(\omega_2) \\ \mu r N_3 sgn(\omega_3) \\ \mu r N_4 sgn(\omega_4) \end{bmatrix}$$

The differential equation of the system is written as

$$M\ddot{Q} + C(Q, \dot{Q}) + B\xi = B\tau \tag{8}$$

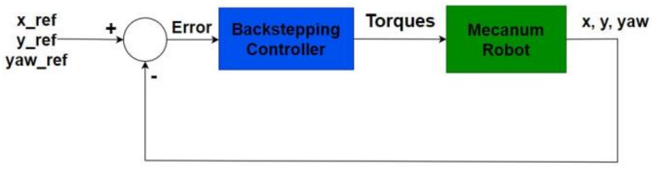

Fig. 2. Block diagram of the control system

where τ_i (i =1,..,4) is the torques provided by the control signal.

C. The Back-stepping Controller

Definite of the reference trajectory matrix $Q = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix}$, the differential equation of the system is rewritten as

$$\ddot{Q} = -M^{-1}[C(Q,\dot{Q}) + B\xi] + M^{-1}B\tau$$

Given $x_1 = Q; x_2 = \dot{Q}$, we have

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 = x_2 \\ \dot{x}_2 = f_2(x_1,x_2) + g_2(x_1,x_2)u \end{cases}$$

where

$$f_1(x_1) = 0\,;\; g_1(x_1) = 1$$

$$f_2(x_1,x_2) = -M^{-1}[C(Q,\dot{Q}) + B\xi];\;\; g_2(x_1,x_2) = M^{-1}B$$

The error between the measured value and the reference value is presented as $e_1 = x_{1r} - x_1$ with x_{1r} that is set trajectory and $e_2 = x_{2r} - x_2$ with x_{2r} that is virtual signal control.

The block diagram of the control system is presented in Fig. 2, where the back-stepping technique is used for designing the controllers that is performed by the following procedure:

Step 1 [3]: Lyapunov V_1

$$V_1(x) = \frac{1}{2}{e_1}^2 \Rightarrow \dot{V}_1(x) = e_1(\dot{x}_{1r} - \dot{x}_1) = e_1[\dot{x}_{1r} - x_2]$$

For $V_1(x) \leq 0, \forall x$ we choose:

$$\dot{x}_{1r} - x_2 = -k_1e_1 \Rightarrow x_2 = [\dot{x}_{1r} + k_1e_1] \qquad (9)$$

Then $\dot{V}_1(x) = -k_1{e_1}^2 \leq 0, \forall x$ with $k_1 > 0$

Step 2 [3]: Lyapunov V_2

$$V_2(x) = V_1(x) + \frac{1}{2}{e_2}^2$$

$$\Rightarrow \dot{V}_2(x) = -k_1{x_1}^2 + e_2\dot{e}_2$$

$$= -k_1{e_1}^2 + e_2[\dot{x}_{2r} - \dot{x}_2]$$

$$= -k_1{e_1}^2 + e_2[\dot{x}_{2r} - f_2(x_1,x_2) - g_2(x_1,x_2)u]$$

For $V_2(x) \leq 0, \forall x$ we choose:

$$\dot{x}_{2r} - f_2(x_1,x_2) - g_2(x_1,x_2)u = -k_2e_2$$

$$\Rightarrow u = g_2(x_1,x_2)^{-1}[\dot{x}_{2r} - f_2(x_1,x_2) + k_2e_2]$$

Then $\dot{V}_2(x) = -k_1{e_1}^2 - k_2{e_2}^2 \leq 0, \forall x$ with $k_1, k_2 > 0$.

The control law is designed as

$$u = g_2(x_1,x_2)^{-1}[\dot{x}_{2r} - f_2(x_1,x_2) + k_2e_2] \qquad (10)$$

From (9) we have $x_2 = [\dot{x}_{1r} + k_1e_1]$, then

$$\begin{cases} x_{2r} = [\dot{x}_{1r} + k_1e_1] \\ \dot{x}_{2r} = [\ddot{x}_{1r} + k_1\dot{e}_1] \end{cases} \qquad (11)$$

Putting (11) into (10) we have the control law

$$u = g_2(x_1,x_2)^{-1}[(\ddot{x}_{1r} + k_1\dot{e}_1) - f_2(x_1,x_2) + k_2e_2]$$

and

$$\tau = (M^{-1}B)^{-1}[(\ddot{Q}_r + k_1\dot{e}_1) + M^{-1}[C(Q,\dot{Q}) + B\xi] + k_2e_2] \qquad (12)$$

Eq. (12) can be transformed in to the following form $\tau = M.B^{-1}.B^T.(B^T)^{-1}.[(\ddot{Q}_r + k_1\dot{e}_1) + M^{-1}[C(Q,\dot{Q}) + B\xi] + k_2e_2]$. Then

$$\tau = B^T.(B.B^T)^{-1}M[(\ddot{Q}_r + k_1\dot{e}_1) + M^{-1}[C(Q,\dot{Q}) + B\xi] + k_2e_2] \qquad (13)$$

III. SIMULATION AND EXPERIMENTAL RESULTS

A. MATLAB Simulation

In this section, we will conduct simulations to evaluate the performance of the backstepping controller. The simulations will be carried out using MATLAB Simulink. The block diagram of our control system is illustrated in Fig.3.

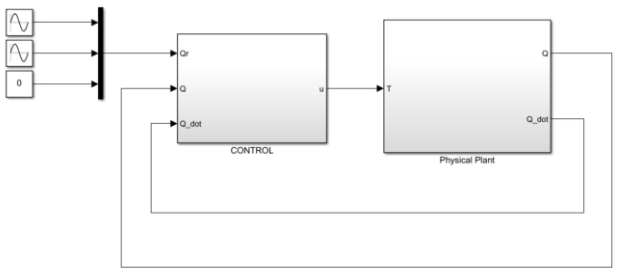

Fig.3. The block diagram of the control system in Simulink

To evaluate the performance of the controller, we will choose a trajectory that poses several technical challenges for the robot, from smooth direction changes to maintaining accurate speed and position. This helps comprehensively assess the technical capabilities of both the robot and the controller. Therefore, an eight-shaped trajectory will be used as the reference trajectory for the robot to follow. The equation for the eight-shaped trajectory is as follows, with the initial angle θ is set to 0 radian:

$$x = 2\,sin\left(\frac{2\pi t}{30}\right) + 2.5$$

$$y = 2\,sin\left(\frac{4\pi t}{30}\right) + 2.5$$

and the parameters are set as follows: m=5, I=5, $I_b = 0.1$, $l_1 = 0.3$, $l_2 = 0.3$, r=0.048, g=9.81, μ =0.01

The result in Fig. 4 shows the output of the back-stepping controller, indicating that the robot has followed the reference trajectory of a figure-eight shape, with the initial position at (2.5, 2.5) (m) and the sample time is 0.05 seconds. The graphs in Fig. 5 show that the robot's trajectories in x and y directions have closely followed the reference trajectory.

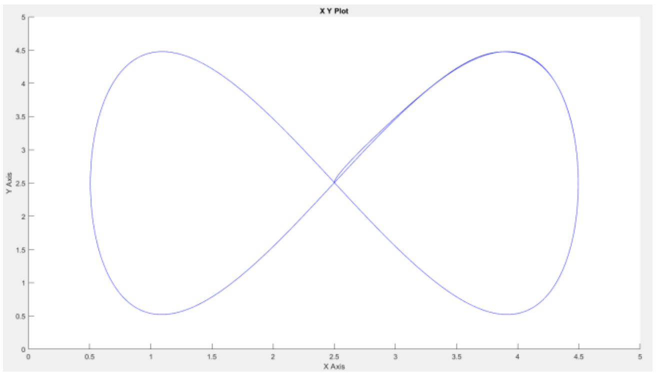

Fig.4. The simulation result of robot's trajectory

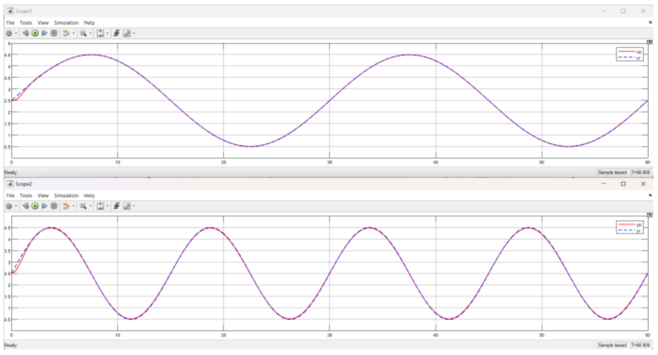

Fig.5. The robot's trajectories in x and y directions over time

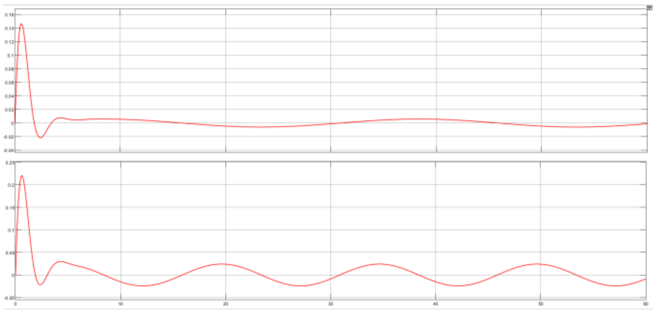

Fig.6. The tracking errors in the x and y directions over time

The dashed blue lines represent the reference trajectory, and the red lines represent the robot's trajectory. Furthermore, it can be seen that it takes about 3 seconds for the output response to reach the steady-state.

The tracking errors in the x and y directions over time are shown in Fig.6, where it can be seen that the errors are relatively small. From the time the robot reaches the steady-state, the maximum error in the x direction is less than 1 cm, and in the y direction, it is about 3 cm. This indicates that the Backstepping controller has performed quite well.

B. *Robot in Virtual Environment*

When the controller is performed well, we try to implement it in a virtual robot model created in Webots. It is worth emphasizing that Webot is a powerful software where we can create a virtual world with controllable objects such as mobile robots, robotic arms, etc. In this virtual environment, it is easy to set up physical properties such as floor slipperiness, friction forces, and even sensor noise. This makes the virtual environment more realistic and helps us predict what will happen when applied to a model in the real world [1].

After setting up appropriate parameters for the virtual world, we will use available sensors in Webots such as

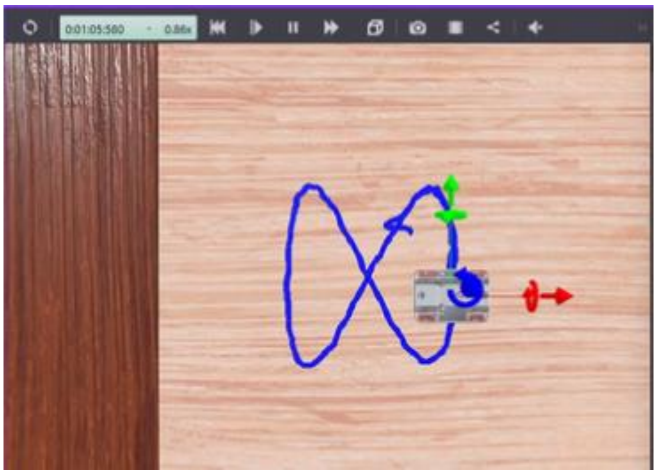

Fig.7. The robot's trajectory in Webots

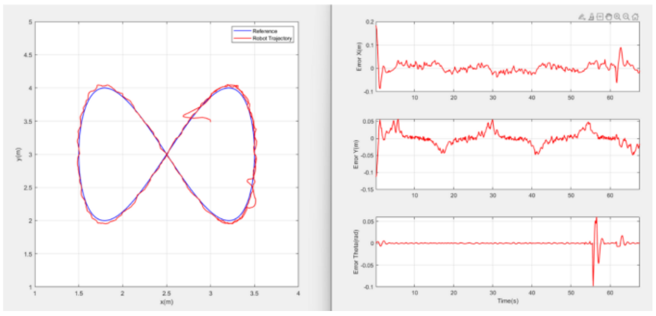

Fig.8. Virtual robot's trajectory and tracking errors over

encoders, IMU, and GPS mounted on the robot [1]. These sensors will be simulated with noise to resemble real-world conditions. Subsequently, the controller will be implemented on the virtual mobile robot to follow the eight-shaped trajectory. As the robot moves, data from these sensors will be read and fed back to the controller to compute and execute the next steps. The figure above shows that the virtual robot has followed an eight-shaped trajectory, where a pen has been attached to the robot to illustrate its path.

The results in Fig. 7 and Fig. 8 provide a clearer evaluation of the error when using the Backstepping controller on the virtual robot. The graph on the right compares the trajectory of the virtual robot with the reference trajectory, where the blue line represents the reference trajectory and the red line represents the trajectory of the virtual robot. It can be seen that the two trajectories almost overlap, indicating that the trajectory errors in the x and y directions over time are quite small. Looking at the three graphs on the right side of the figure, we see that after stabilization, the maximum error in the x direction is 10 cm and in the y direction is 5 cm, with the angle error being only 0.05 radians. Their RMS errors are shown in Table II.

TABLE II. RMS ERRORS OF VIRTUAL ROBOT

$e_{xv(RMS)}$ (m)	0.0227
$e_{yv(RMS)}$ (m)	0.0209
$e_{\theta v(RMS)}$ (rad)	0.0083

Although the errors are larger than those in the simulation, these errors are still small, indicating the controller still performs very well. However, despite the sensor noise settings, this is still conducted in a virtual environment, which is nearly ideal. Therefore, we next implement this controller on a real model to verify its performance.

C. *Robot in Real-World Environment*

Next, we will test the accuracy of the Ultra Wideband (UWB) sensor. We will draw a specific trajectory on the floor and control the robot to follow that path. In Fig.9, when the robot is made to move in a square pattern, the coordinates of the robot will be read by the UWB Tag to collect data and evaluate accuracy. From this, we can see that the read results are nearly accurate to the reference square. However, in the lower half of the square, larger errors have appeared, with the largest error in the x-direction being about 13 cm and in the y-direction about 20 cm. Therefore, it can be seen that UWB is very sensitive to noise. The average error in the x-direction is 0.0385 m and in the y-direction is 0.0664 m.

Since the sensor is quite sensitive to noise, we can anticipate that the trajectory tracking results may not be as expected. Here, an eight-shaped trajectory will continue to be selected for the robot to follow, with the center of the reference figure-eight trajectory set at coordinates [2.5, 2.5] (m) and the initial angle θ is 0 radian.

The results in Fig. 10 show that, despite recalibrating the UWB sensor and running multiple tests, the robot still tends to follow the preset eight-shaped trajectory but with considerable errors. Data collection and analysis indicate that the maximum error in both the x and y directions is approximately 0.1 m. During movement, there are times when the front of the robot deviates, but the IMU BNO055 sensor mounted on the robot continues to function by reading the yaw angle value and sending it to the controller to ensure the front of the robot maintains an angle of 0 rad, as specified in the trajectory. The RMS errors of the physical robot are shown in Table III:

TABLE III. RMS ERRORS OF PHYSICAL ROBOT

$e_{xv(RMS)}$ (m)	0.0529
$e_{yv(RMS)}$ (m)	0.0370
$e_{\theta v(RMS)}$ (rad)	0.0389

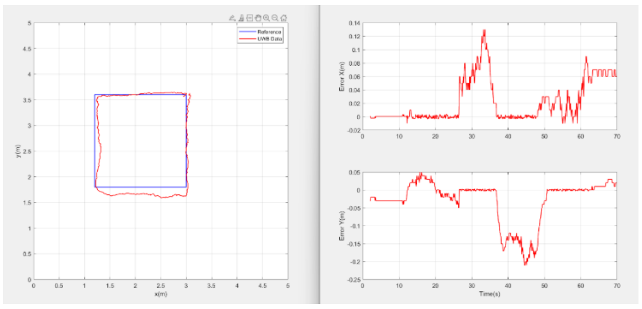

Fig.9. The results of robot moving in a square path

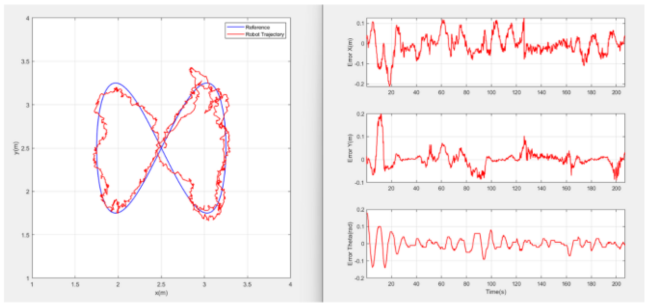

Fig.10. Physical robot's trajectory and tracking errors

It can be seen that there is a certain discrepancy between the real robot and the virtual robot in terms of trajectory tracking. The reason is that the virtual robot always operates in an ideal environment, where the coordinate data received from the Webots localization function is always accurate. In contrast, in the real-world model, the coordinate data read from the UWB is easily affected by noise, resulting in less satisfactory outcomes.

Overall, the simulation results and virtual environment tests have demonstrated that the backstepping controller functions effectively in guiding the robot to follow the given trajectory, even for complex paths requiring quick direction changes. However, to perform better in real-world scenarios, improvements in localization accuracy are necessary to ensure the most precise coordinate readings possible.

IV. CONCLUSION

This article designs and implements a back-stepping controller for position and heading angle control of a four-mecanum-wheeled mobile robot. Through simulation, the robot has been able to closely follow any given trajectory, achieving smooth responses, and high accuracy with very small errors and fast response times. Similarly, the virtual robot on Webots can also follow the reference trajectory precisely. These results confirm that the back-stepping controller has performed well, even with the difficult trajectories that require quick direction changes. Additionally, a Digital Twins system has been designed to easily monitor the movements of the real robot. In this system, a virtual robot model is created in Webots with parameters such as structure, kinematics, dynamics, and actuators similar to the real robot model. The development direction of this project is to optimize the coordinate determination to help the robot model follow the trajectory better, and interact better with the virtual robot in the Webot environment to improve the Digital Twins system.

ACKNOWLEDGMENT

This research is funded by the Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number B2023-20-10. We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] H. Yang, F. Cheng, H. Li, and Z. Zuo, "Design and Control of Digital Twin Systems Based on a Unit Level Wheeled Mobile Robot", IEEE Transactions on Vehicular Technology, vol. 73, no. 1, pp. 323-332, January 2024.

[2] Z. HENDZEL and Ł. RYKAŁA, "Modelling of Dynamics of a Wheeled Mobile Robot with Mecanum Wheels with the use of Lagrange Equations of the Second Kind", Int. J. of Applied Mechanics and Engineering, vol.22, no.1, pp.81-99, January 2023.

[3] U. Kumar and N. Sukavanam, "Backstepping Based Trajectory Tracking Control of a Four Wheeled Mobile Robot," International Journal of Advanced Robotic Systems, vol. 5, no. 4, November 2008.

[4] P. T. H. Sen, "Design controller for underactuated wheel mobile robots based on fuzzy hierarchical sliding mode and backstepping techniques", EPU Journal of Science and Technology for Energy, vol. 31, no. 11, July 2023.

[5] N. M. Dong, N. M. Tien, D. Q. Hiep, B. V. Bac, C. V. Vuong, and N. D. Thang, "Research dynamic surface control for four-mecanum wheeled mobile robot," No. FEE (2022), December 2022.