



Design of Embedded Control System with Fuzzy Controller and Nonlinear Controller for the Line Follower Robot

Nguyen Xuan Chiem^{1,*} , Nguyen Cong Binh Nguyen² 

^{1,2}Department of Automation and Computing Techniques, Le Quy Don Technical University, Vietnam

Email: ¹chiemnx@mta.edu.vn, ²nguyentutu@gmail.com

*Corresponding Author

Abstract—In this paper, an embedded system for motion control of a Line Follower Robot (LFR) is presented. Line Follower Robot can change direction by changing the relative rotational speed of the wheels, and thus does not require additional steering motion. A robot designed at the Control Systems Laboratory, Le Quy Don Technical University, is chosen as the research platform in this paper. A fuzzy logic controller has been used to ensure the smallest position and angle deviation. The rules of the fuzzy logic controller are built based on the successor's experience when considering the Lyapunov function. The output of this controller is linear velocity and angular velocity, which are the achieved values for the robot dynamics control loop. A nonlinear controller with blocked signals is synthesized based on the synergetic control theory (STC). The combination of control laws ensures the system is stable enough to measure noise and uncertainties in the robot's model and parameters. In addition, we realize that the control system is embedded. Simulation and experimental results with different scenarios demonstrate the effectiveness of the proposed control law.

Keywords—Line Follower Robot; Fuzzy Logic Controller; Lyapunov Function; Embedded System; Synergetic Control Theory

I. INTRODUCTION

LFR has been widely used in medicine, education, industry, and entertainment in recent years. In research [1], [2], an LFR was designed to serve patients in hospitals. This robot delivers medicine to patients by following lines drawn on the floor whenever requested. LFRs are widely used in education at all levels [3]. Many competitions and games based on this robot have been developed. Another application of LFRs is the automatic transportation of equipment in industrial facilities [4]. Line detection variables are implemented based on infrared sensor modules [5]-[7]. Line following modules are widely used for most line tracking robots due to their simplicity in programming and communication and low cost. Nowadays, with the development of computer vision technology, many robots use cameras to detect lines and calculate output status values, as feedback signals in the control system [8], [9].

Control methods and controllers are commonly used on robots to enhance tracking performance. The controller is designed for this robot to follow a predetermined path by color or boundary. The control system for LFR is often built in the form of centralized control or hierarchical control. Hierarchical control includes two controllers: a kinematic

controller and a robot dynamic controller. In the study [10], [11], using a PID controller to adjust the robot's motion, the results show that the proportional link value has a great influence on the tracking stability of the robot. The LQR centralized controller based on the local linearization model is presented in the study [12]. The control laws are proposed to improve the quality of line tracking when the mathematical model has nonlinear components. In the study [13], the backstepping control law is designed to overcome the uncertainty and load disturbance of the mathematical model. Based on the robot error model and using the Lyapunov method, a variable structure controller designed to ensure stable tracking performance is presented in research [14]. In research [15], [16], a hierarchical control method is used with controllers designed using the Lyapunov method and the ensemble control theory. In research [17], [18], a robust, vibration-free sliding mode controller (SMC) is designed and applied to a path-following robot. The results of this study show that the tracking quality of complex paths is accurate, and the movement speed is fast and stable.

Many research papers have used soft computing algorithms for mobile robot control in academic as well as engineering fields. Fuzzy logic is used in designing feasible solutions to perform local navigation, global navigation, path planning, steering control, and speed control of mobile robots [19], [20]. In [21]-[25], a method for tracking the path of a smart vehicle using two fuzzy controllers combined with vehicle direction control has been developed. In the study [21], a fuzzy logic controller and a regression method to find the optimal design parameters for a robot following a path with the shortest path completion time have been presented. In the study [22], a detailed design of a preference-based fuzzy behavioral system for robot vehicle navigation using a multi-valued logic framework has been presented. A fuzzy PD control law with parameter optimization using Nature-Inspired Optimization Algorithms has been presented in the study [23]. The neural network control method to overcome measurement noise is presented in research [26]. The research presented above has good simulation results, but when designing embedded systems, the control signal is limited by saturation functions, leading to limited control quality in some situations.

This study aims to overcome measurement noise factors and limit control signals on embedded systems. The kinematic control law is designed based on fuzzy logic and Lyapunov functions to overcome noise in image processing.

The dynamic control law is designed based on the composite theory to limit the control signal according to the physical value of the real system. This paper is organized as follows: Section 2 presents the experimental model and mathematical model of the LFR. The synthesis of kinematic and dynamic controllers based on fuzzy theory and the STC is presented in Section 3. Section 4 presents simulation results on MATLAB software and experimental results. Conclusions and further research directions are described in Section 5.

II. PROBLEM FORMULATION

A. Platform Introduction and Operating Principle

In this paper, LFR at the Control Systems Laboratory, Le Quy Don Technical University. Is taken as the research object. The mechanical 3D structure is shown in Fig. 1.

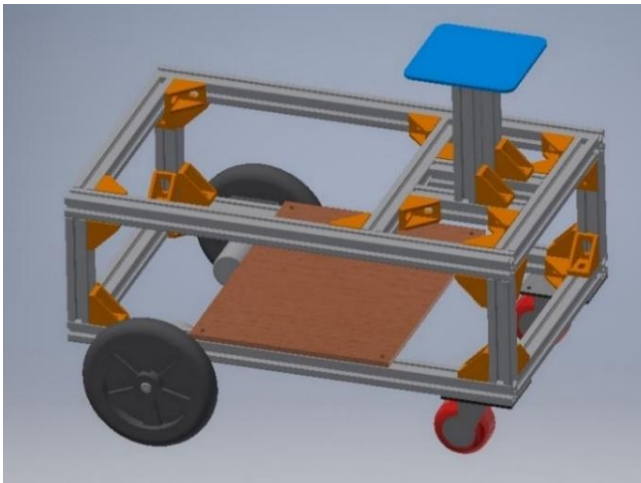


Fig. 1. Mechanical 3D structure of LFR

LFR is a differential two-wheel motor, two DC motors control the speed and direction of the robot by changing the speed of the two wheels. The speed of the two wheels is measured by a relative sensor mounted on the motor shaft with a resolution of 300 pulses. The Kinect Camera is connected to the Jetson Nano embedded computer, the captured images are processed, and information about the deviation distance and deviation angle of the robot is calculated. The Arduino MEGA 2560 embedded board executes the embedded control program and outputs signals to the DC Driver to control the two motors of the robot. At the same time, it receives signals from the MPU 6050 sensor modules, the encoder mounted on the robot. The robot's schematic is shown in Fig. 2.

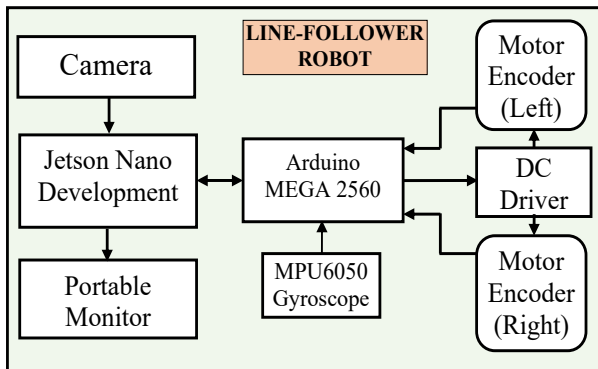


Fig. 2. LFR's schematic

B. Mathematical Model of LFR

An approximate diagram of a real-world two-wheeled robot is shown in Fig. 3 and lists the important components and distances for the model, as well as the forces considered: one vertical force and one horizontal force at each drive wheel (the forces at the rolling wheels are considered negligible). The curve $\sigma(q)$ represents the path the robot needs to follow.

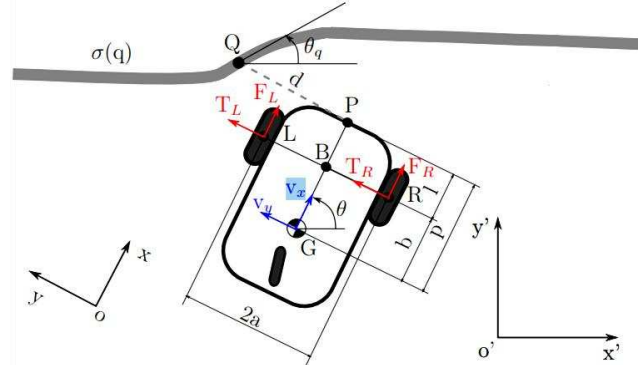


Fig. 3. Scheme of the robot and the line it follows

Considering the robot as a rigid body, moving on a plane and without slipping, its dynamics are constructed by applying the Law of Conservation of Momentum and the Law of Conservation of Angular Momentum [7]. The velocity and acceleration have been calculated in a common reference frame fixed on the floor (x', y') , but the vectors are expressed in terms of the x-y motion basis, which is fixed on the robot and has its own angular velocity. Both basic coordinate systems are shown in Fig. 3. The dynamics model presented in the studies [9], [27] is as follows:

$$\begin{cases} \dot{d} = -l\omega - \theta_e(v_x - \omega d) \\ \dot{\theta}_e = \omega - c(v_x - \omega d) \\ \dot{v}_x = \frac{1}{mr}u_v - 2\frac{B_f}{mr^2}v_x - b\omega^2 \\ \dot{\omega} = \frac{1}{I_z + mb^2}\left(\frac{a}{r}u_\omega - 2\frac{B_f a^2}{r^2}\omega + mb\omega v_x\right) \end{cases} \quad (1)$$

where d is the distance from the sensor axis to the line (m); θ_e is defined as follows $\theta_e = \theta - \theta_q$ (rad.); I_z is the moment of inertia of the robot along the vertical axis ($kg.m^2$), B_f is the total coefficient of friction between the rotor and the wheel (kg/ms), ω is the angular velocity of the robot ($rad./s$), $uv = ur + ul$, $u\omega = ur - ul$ with ur, ul being the voltage applied to the motors respectively (V), r is the radius of the wheel (m), v_x is the linear velocity of the robot (m/s), c is the curvature of the line, m is the mass of the robot (kg).

III. EMBEDDED CONTROL SYSTEM DESIGN FOR LFR

A. Algorithm to Determine d and θ_e

In the study to determine d and θ_e , the camera was used to determine the line. The technique to determine the blue line is based on feature analysis. The purpose of this approach is to try to highlight the difference between the observed line image and the image without a line. This technique is based on the characteristics of the image, from the captured images through color filters, analysis, and giving the characteristics

of the lane. The OpenCV-Python library in the Ubuntu operating system is used to implement this technique. After finding the line in the image, the values of the deviation angle (θ_e) and the distance (d) of the robot from the line are calculated. The value of the deviation angle (θ_e) is the angle between the main contour in the image and the vertical axis of the image. Use the `cv2.minAreaRect()` function, which will return information about the rectangle surrounding the border, including the angle between the rectangle and the horizontal axis of the image. After some simple calculations, we can find the deviation angle (θ_e) easily. About the distance (d), we can calculate it by counting the number of pixels from the vertical axis in the middle of the image to the center of the rectangle surrounding the border. From that number of pixels, we can calculate the distance in meters by experimentally measuring how many meters one pixel on the image is because the distance and viewing angle from the camera are unchanged. Fig. 4 illustrates how to calculate d and θ_e on the image and the result after calculation and image processing.

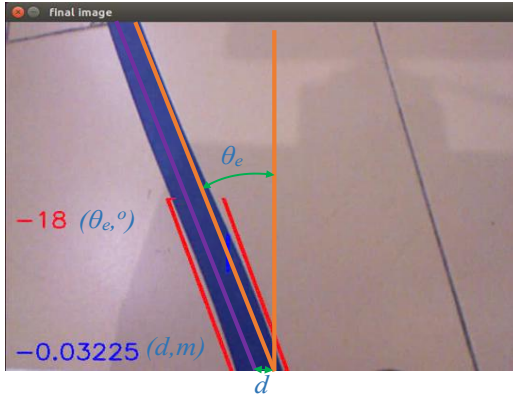


Fig. 4. Results d , θ_e after image processing

B. Algorithm to Determine v_x and ω

The linear velocity v_x is determined by two relative encoder sensors attached to the right and left motor shafts. Two external interrupts are used to count the number of motor pulses in a period of 100 (ms). The motor's translational velocity is the average of the total translational velocity of the two wheels. The program to calculate the velocity v_x has the following form:

```
void countEncoder() {
    encoderCountR++;
}
void countEncoderL() {
    encoderCountL++;
}
void loop() {
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        speed_R=2*pi*(encoderCountR/300)*(1000.0/interval);
        speed_L=2*pi*(encoderCountL/300)*(1000.0/interval);
        v_x=p_wheel*(speed_R+speed_L)/2;
        encoderCountR = 0;
        encoderCountL = 0;
    }
}
```

The robot's angular velocity ω is determined through the MPU6050 sensor by determining the angular velocity around the vertical z-axis. The angular velocity nebula program ω (GyroZ) has the following form:

```
Wire.beginTransmission(MPU);
Wire.write(0x47);
Wire.endTransmission(false);
Wire.requestFrom(MPU, 2, true);
GyroZ = (Wire.read() << 8 | Wire.read()) / 131.0;
GyroZ = GyroZ - 3.90;
```

Where 3.90 is the velocity calibration at the sensor's zero point.

C. Controller Synthesis for Embedded Control Systems

The robot controller consists of two components: the kinematic controller and the dynamic controller. First, the Kinematic controller is synthesized based on the kinematic equation. From equation (1), we use the first two equations to synthesize the control law:

$$\begin{cases} \dot{d} = -l\omega - \theta_e(v_x - \omega d) \\ \dot{\theta}_e = \omega - c(v_x - \omega d) \end{cases} \quad (2)$$

Select the candidate Lyapunov function as follows:

$$V = \frac{1}{2}d^2 + \frac{1}{2}\theta_e^2 \quad (3)$$

From the derivative of the Lyapunov function, we get:

$$\begin{aligned} \dot{V} &= d\dot{d} + \theta_e\dot{\theta}_e = \\ &= d(-l\omega - \theta_e(v_x - \omega d)) + \theta_e(\omega - c(v_x - \omega d)) = \\ &= \omega(\theta_e - ld) - (v_x - \omega d)(\theta_e d - c\theta_e) \end{aligned} \quad (4)$$

To ensure stability according to Lyapunov with the values of l, c within the given limits. We synthesize a structured fuzzy controller with 2 input variables d, θ_e , and 2 output variables v_r, ω_r . These two values Select the linguistic variables as follows: the distance deviation d (Big negative, Negative, Zero, Positive, Big positive); the angle deviation θ_e (Big negative, Negative, Zero, Positive, Big positive) is defined as in Fig. 5 and Fig. 6. Based on the analysis experience, the control law of the controller is designed to include 25 IF-THEN rules listed in Table I, and Table II. The defuzzification method of the fuzzy controller is calculated according to the defuzzification principle of the Tagaki-Sugeno fuzzy model.

Table 1. The fuzzy rule bases for the angle controller

ω_r	θ_e					
	NB	N	Z	P	BP	
d	BN	0.0732	0.0319	-0.0094	-0.0506	-0.0919
	N	0.0778	0.0366	-0.0047	-0.0459	-0.0872
	Z	0.0825	0.0413	0	-0.0413	-0.0825
	P	0.0872	0.0459	0.0047	-0.0366	-0.0778
	BP	0.0919	0.0506	0.0094	-0.0319	-0.0732

Table 2. The fuzzy rule bases for the distance controller

v_r	θ_e					
	NB	N	Z	P	BP	
d	BN	0.7149	0.3598	0.0047	-0.3504	-0.7056
	N	0.3570	0.1791	0.0012	-0.1768	-0.3547
	Z	0.0015	0.00075	0	0.00075	-0.0015
	P	-0.3517	-0.1753	0.0012	0.1776	0.3540
	BP	-0.7026	-0.3489	0.0047	0.3583	0.7119

Synthesizing the dynamic controller when the input is limited, from equation (1) we use the last two equations

combined with expanding the phase space of the system to calculate the additional input condition [16] as follows.

$$\begin{cases} \dot{v}_x = \frac{1}{mr} \frac{2}{\pi} U_0 \arctg(b_v x_5) - 2 \frac{B_f}{mr^2} v_x - b\omega^2 \\ \dot{x}_5 = u_1 \\ \dot{\omega} = \frac{1}{I_z + mb^2} \left(\frac{a}{r} \frac{2}{\pi} U_0 \arctg(b_\omega x_6) - 2 \frac{B_f a^2}{r^2} \omega + mb\omega v_x \right) \\ \dot{x}_6 = u_2 \end{cases} \quad (5)$$

In which, U_0 is the maximum value of the voltage signal supplied to two motors at the same time on the real system, or $\max(ur + ul) \leq U_0$, b_v and b_ω are positive constants.

According to the requirements of the LFR problem, the motor moves with a given translational velocity $v_0 + vr$ and an angular velocity of ωr . We choose the technology invariant corresponding to the control objective as follows:

$$\begin{cases} v_x = v_0 + v_r \\ \omega = \omega_r \end{cases} \quad (6)$$

The control law synthesis manifolds are selected as follows:

$$\begin{cases} \psi_v = \frac{2}{\pi} U_0 \arctg(b_v x_5) + k_v (v_x - v_0 - v_r) \\ \psi_\omega = \frac{2}{\pi} U_0 \arctg(b_\omega x_6) + k_\omega (\omega - \omega_r) \end{cases} \quad (7)$$

According to the ADAR method, the macro variables ψ_v and ψ_ω must satisfy the solution of the system of basis function equations:

$$\begin{cases} T_v \dot{\psi}_v + \psi_v = 0 \\ T_\omega \dot{\psi}_\omega + \psi_\omega = 0 \end{cases} \quad (8)$$

From equations (7) and (8), we have:

$$\begin{cases} \frac{2}{\pi} \frac{U_0 b_v u_1}{(b_v x_5)^2 + 1} + k_v \left(\frac{1}{mr} \frac{2}{\pi} U_0 \arctg(b_v x_5) - 2 \frac{B_f}{mr^2} v_x - b\omega^2 - \dot{v}_r \right) = -\frac{\psi_v}{T_v} \\ \frac{2}{\pi} \frac{U_0 b_\omega u_2}{(b_\omega x_6)^2 + 1} + \frac{k_\omega}{I_z + mb^2} \left(\frac{a}{r} \frac{2}{\pi} U_0 \arctg(b_\omega x_6) - 2 \frac{B_f a^2}{r^2} \omega + mb\omega v_x - \dot{\omega}_r \right) = -\frac{\psi_\omega}{T_\omega} \end{cases} \quad (9)$$

Solving the system of equations (9), we get the virtual control signals u_1 , and u_2 as follows:

$$\begin{cases} u_1 = \frac{\pi (b_v x_5)^2 + 1}{2 U_0 b_v} \left(k_v \left(\frac{1}{mr} \frac{2}{\pi} U_0 \arctg(b_v x_5) + 2 \frac{B_f}{mr^2} v_x + b\omega^2 + \dot{v}_r \right) - \frac{\psi_v}{T_v} \right) \\ u_2 = \frac{\pi (b_\omega x_6)^2 + 1}{2 U_0 b_\omega} \left(-\frac{k_\omega}{I_z + mb^2} \left(\frac{a}{r} \frac{2}{\pi} U_0 \arctg(b_\omega x_6) - 2 \frac{B_f a^2}{r^2} \omega + mb\omega v_x - \dot{\omega}_r \right) - \frac{\psi_\omega}{T_\omega} \right) \end{cases} \quad (10)$$

The real control signal supplied to the motor is calculated according to the formula:

$$\begin{cases} u_r = \frac{U_0}{\pi} (\arctg(b_v x_5) + \arctg(b_\omega x_6)) \\ u_l = \frac{U_0}{\pi} (\arctg(b_v x_5) - \arctg(b_\omega x_6)) \end{cases} \quad (11)$$

IV. SIMULATION AND EXPERIMENTAL RESULTS

A. Simulation Results

The robot model is designed with hardware with model parameters determined relatively as follows: $m =$

10 (kg) ; $a = 0.18 \text{ (m)}$, $b = 0.3 \text{ (m)}$, $l = 0.068 \text{ (m)}$, $r = 0.035 \text{ (m)}$, $I_z = 9.292 \cdot 10^{-3} \text{ (kg/m}^2\text{)}$, $B_f = 1.543 \cdot 10^{-4}$. The simulation scenario is conducted with the condition that the robot will follow a path with a fixed small curvature $c = 0.01$, and the robot's translational velocity is stable at 0.4 (m/s) . The initial values of the robot are as follows: $d_0 = -0.2 \text{ (m)}$, $\theta_{e0} = 0.3 \text{ (rad.)}$, $v_{x0} = 0 \text{ (m/s)}$, $\omega_0 = 0 \text{ (rad./s)}$. Parameters of the control law: $b_v = 10$, $b_\omega = 10$, $T_v = 0.1$, $T_\omega = 0.1$, $k_v = 10$, $k_\omega = 10$, $U_0 = 6 \text{ (V)}$. The simulation results performed with the proposed control law are shown in Fig. 7, Fig. 8, and Fig. 9. From the results, we can see that the proposed control law ensures that the robot follows the desired path. Fig. 7 shows the error in distance and angle approaching 0 after 7 (s). The translational velocity and angular velocity of the robot approaching the desired value after 4 (s) are shown in Fig. 8. Fig. 9 shows that the voltage value supplied to the motor is less than U_0 .

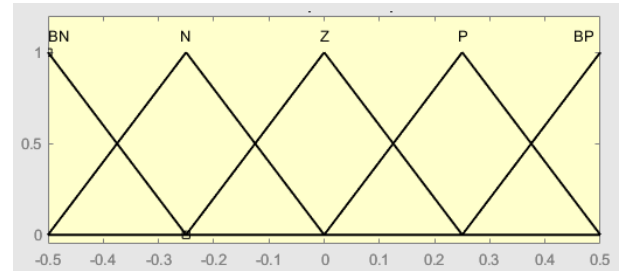


Fig. 5. Input membership function for distance d

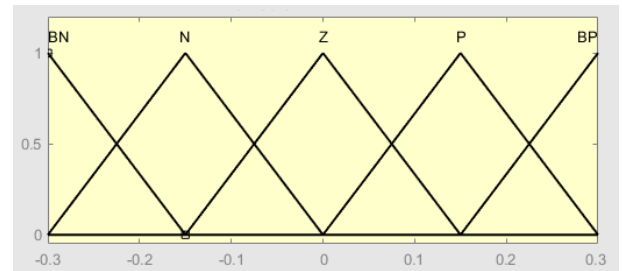


Fig. 6. Input membership function for angle θ_e

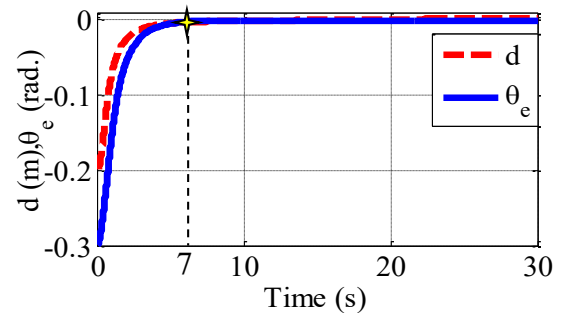


Fig. 7. Response of distance error d and angle error θ_e of the robot

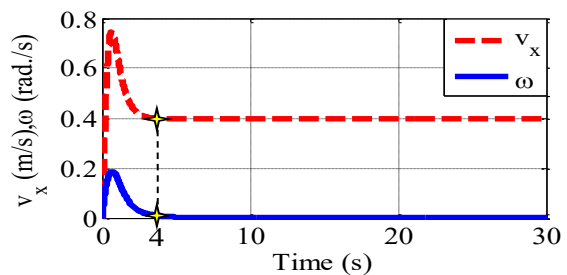


Fig. 8. Response of linear velocity v_x and angular velocity ω of the robot

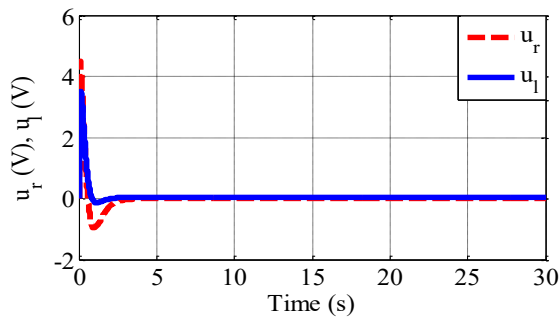


Fig. 9. Control voltage signals for robot motors

B. Experimental Results

Experimental research on LFR has been carried out to study the effectiveness of the proposed control algorithm. The trajectory tracking experiment for the experimental LFR presented in Section 2 was carried out based on the Arduino embedded board. The experimental scenario for the robot to follow a curved line. The experimental results are shown in Fig. 10, Fig. 11, and Fig. 12. From the results in Fig. 10, it can be seen that the LFR has followed the line with acceptable error. Obviously, at the bend points, the robot moves with the largest distance error of 0.05 (m) at 6.6 (s), the angle error θ_e changes rapidly due to the bend of the line. The linear velocity of the robot reaches the set value of 0.4 (m/s) after more than 6 (s) and the angular velocity changes rapidly when entering a bend (Fig. 11). The visual test results of the tracking performance of the two controllers are shown in Fig. 12. In which the blue line is the line to be tracked and the red line is the robot's center of gravity displacement line. From the figures, it can be seen that the robot's line tracking result is quite good, but there is still some deviation.

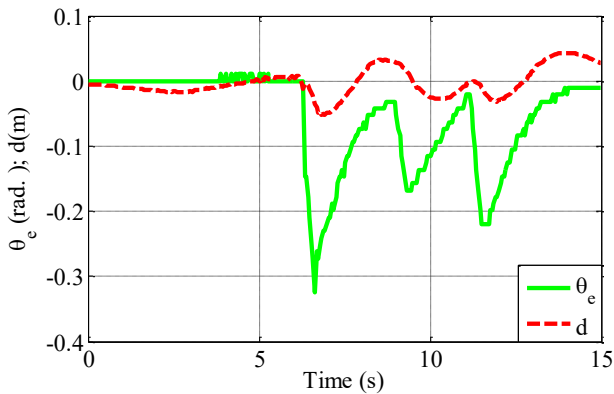


Fig. 10. Experimental trajectory tracking errors d and θ_e of the proposed control law

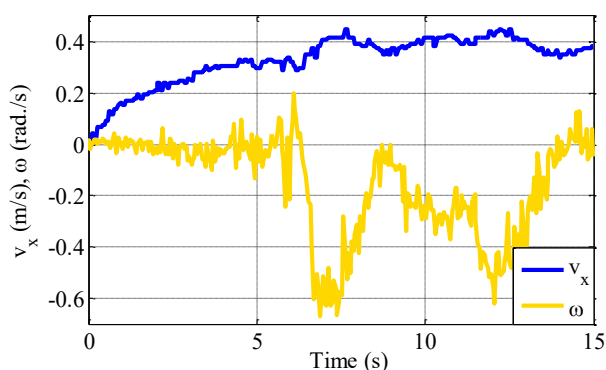


Fig. 11. Experimental velocity v_x and ω of the proposed control law



Fig. 12. Experimental results of the tracking performance

V. CONCLUSION

The purpose of this study is to design a successful embedded controller for line-following robots. The LFR experimental model is designed in a computer vision laboratory to detect lines. The control law for the robot is designed to include two serial controllers, and the dynamic control law is built based on fuzzy theory through Lyapunov functions. The dynamic control law with bounded inputs is designed based on the ADAR method and the expansion of the dynamic equation. The proposed control law ensures the robot is robust to input disturbances and performs efficiently on embedded systems. Simulation and experimental results have verified the effectiveness of the proposed control law. In addition, we also see that the choice of control law parameters greatly affects the control quality. In the following studies, the authors will present controllers when the system has noise and uncertainty factors to overcome these disadvantages.

REFERENCES

- [1] N. Gu, D. Wang, Z. Peng, J. Wang and Q. -L. Han, "Advances in Line-of-Sight Guidance for Path Following of Autonomous Marine Vehicles: An Overview," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 12-28, 2023, <https://doi.org/10.1109/TSMC.2022.3162862>.
- [2] B. Ilias, R. Nagarajan, M. Murugappan, K. Helmy, A. S. A. Omar, and M. A. A. Rahman, "Hospital nurse following robot: hardware development and sensor integration," *International Journal of Medical Engineering and Informatics*, vol. 6, no. 1, pp. 1-13, 2014, <https://doi.org/10.1504/IJMEI.2014.058521>.
- [3] M. Makrodimitris, A. Nikolakakis, and E. Papadopoulos, "Semi-autonomous color line-following educational robots: Design and implementation," *Proceedings of the 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1052-1057, 2011, <https://doi.org/10.1109/AIM.2011.6027098>.
- [4] F. Kaiser, S. Islam, W. Imran, K. H. Khan, and K. M. A. Islam, "Line follower robot: Fabrication and accuracy measurement by data acquisition," *Proceedings of the 2014 International Conference on Electrical Engineering and Information & Communication Technology*, pp. 1-6, 2014, <https://doi.org/10.1109/ICEEICT.2014.6919137>.
- [5] V. Balaji, M. Balaji, M. Chandrasekaran, and I. Elamvazuthi, "Optimization of PID control for high speed line tracking robots," *Procedia Computer Science*, vol. 76, pp. 147-154, 2015, <https://doi.org/10.1016/j.procs.2015.12.329>.
- [6] A. Latif, H. A. Widodo, R. Rahim, and K. Kunal, "Implementation of line follower robot based microcontroller ATmega32A," *Journal of Robotics and Control (JRC)*, vol. 1, no. 3, pp. 70-74, 2020, <https://doi.org/10.18196/jrc.1316>.
- [7] A. Ubaidillah and H. Sukri, "Application of odometry and Dijkstra algorithm as navigation and shortest path determination system of warehouse mobile robot," *Journal of Robotics and Control (JRC)*, vol. 4, no. 3, pp. 413-423, 2023, <https://doi.org/10.18196/jrc.v4i3.18489>.

- [8] M. S. Prakash, K. A. Vignesh, J. Shyamsunthar, K. Raman, J. S. Raju, and N. Raju, "Computer vision assisted line following robot," *Procedia Engineering*, vol. 38, pp. 1764-1772, 2012, <https://doi.org/10.1016/j.proeng.2012.06.215>.
- [9] N. X. Chiem and N. H. Nam, "Synthesis of active disturbance rejection controller via extended state observer combined with LQR controller for two-wheeled line tracking robot," *Journal of Robotics and Control (JRC)*, vol. 6, no. 2, pp. 677-685, 2025, <https://doi.org/10.18196/jrc.v6i2.24754>.
- [10] Y. Oktarina, M. Nawawi, and W. G. Tulak, "Analysis of the sensor line on line follower robot as an alternative transport the tub trash in the shopping center," *VOLT: Jurnal Ilmiah Pendidikan Teknik Elektro*, vol. 2, no. 2, pp. 101-108, 2017, <https://dx.doi.org/10.30870/volt.v2i2.1859>.
- [11] M. Alfian and R. D. Puriyanto, "Mecanum 4 omni wheel directional robot design system using PID method," *Journal of Fuzzy Systems and Control*, vol. 1, no. 1, pp. 6-13, 2023, <https://doi.org/10.59247/jfsc.v1i1.27>.
- [12] F. Lin, Z. Lin, and X. Qiu, "LQR controller for car-like robot," in *Proceedings of the 2016 35th Chinese Control Conference (CCC)*, pp. 2515-2518, 2016, <https://doi.org/10.1109/ChiCC.2016.7553742>.
- [13] I. Hassani and C. Rekik, "Backstepping controller for mobile robot in presence of disturbances and uncertainties," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 934-954, 2023, <https://doi.org/10.31763/ijrcs.v3i4.1110>.
- [14] A. D. Asham, "Mathematical analysis of a line-follower robot, a stable controller design using Lyapunov approach, and experimental tests," *International Journal of Dynamics and Control*, vol. 11, no. 1, pp. 385-395, 2023, <https://doi.org/10.1007/s40435-022-00973-x>.
- [15] N. X. Chiem, "Cascade control for trajectory-tracking mobile robots based on synergetic control theory and Lyapunov functions," *Control Systems and Optimization Letters*, vol. 3, no. 1, pp. 14-19, 2025, <https://doi.org/10.59247/csol.v3i1.169>.
- [16] A. A. Kolesnikov, "Introduction of synergetic control," *2014 American Control Conference*, pp. 3013-3016, 2014, <https://doi.org/10.1109/ACC.2014.6859397>.
- [17] H. Yildiz, N. K. Can, O. C. Ozguney, and N. Yagiz, "Sliding mode control of a line following robot," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 11, p. 561, 2020, <https://doi.org/10.1007/s40430-020-02645-3>.
- [18] S. T. Dang, X. M. Dinh, T. D. Kim, H. L. Xuan, and M. H. Ha, "Adaptive backstepping hierarchical sliding mode control for 3-wheeled mobile robots based on RBF neural networks," *Electronics*, vol. 12, no. 11, p. 2345, 2023, <https://doi.org/10.3390/electronics12112345>.
- [19] M. D. Akmal, A. R. Al Tahtawi, K. Wijayanto, and F. Wahab, "Trajectory tracking control design for mobile robot using interval type-2 fuzzy logic," *Journal of Fuzzy Systems and Control*, vol. 2, no. 2, pp. 67-73, 2024, <https://doi.org/10.59247/jfsc.v2i2.200>.
- [20] B. Xiong and Q. U. Shiru, "Intelligent vehicle's path tracking based on fuzzy control," *Journal of Transportation Systems Engineering and Information Technology*, vol. 10, no. 2, pp. 70-75, 2010, [https://doi.org/10.1016/S1570-6672\(09\)60034-9](https://doi.org/10.1016/S1570-6672(09)60034-9).
- [21] M. Basso, E. P. de Freitas, "A UAV guidance system using crop row detection and line follower algorithms," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 3, pp. 605-621, 2020, <https://doi.org/10.1007/s10846-019-01006-0>.
- [22] M. F. Selekwia, D. D. Dunlap, D. Shi, and E. G. Collins Jr, "Robot navigation in very cluttered environments by preference-based fuzzy behaviors," *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 231-246, 2008, <https://doi.org/10.1016/j.robot.2007.07.006>.
- [23] N. X. Chiem, N. D. Anh, A. D. Lukianov, P. D. Tung, H. D. Long, and N. D. Linh, "Design real-time embedded optimal PD fuzzy controller by PSO algorithm for autonomous vehicle mounted camera," *AIP Conference Proceedings*, vol. 2188, no. 1, p. 030008, 2019, <https://doi.org/10.1063/1.5138401>.
- [24] M. Głowicki and B. S. Butkiewicz, "Autonomous line-follower with fuzzy control," in *2013 Signal Processing Symposium (SPS)*, pp. 1-6, 2013, <https://doi.org/10.1109/SPS.2013.6623591>.
- [25] H. Omrane, M. S. Masmoudi, and M. Masmoudi, "Fuzzy logic based control for autonomous mobile robot navigation," *Computational Intelligence and Neuroscience*, vol. 2016, no. 1, p. 9548482, 2016, <https://doi.org/10.1155/2016/9548482>.
- [26] H. M. Leal, R. S. Barbosa, and I. S. Jesus, "Control of a mobile line-following robot using neural networks," *Algorithms*, vol. 18, no. 1, p. 51, 2025, <https://doi.org/10.3390/a18010051>.
- [27] P. Morin, C. Samson, "Motion control of wheeled mobile robots," *Springer handbook of robotics*, vol. 1, pp. 799-826, 2008, https://www-sop.inria.fr/members/Pascal.Morin/PapersDR2/chap_08.pdf.