Proposal of a Fault-tolerant controller for wheeled mobile robots with faulty actuators

Ho Sy Phuong
School of Engineering and Technology
Vinh University
Nghe An, Vietnam
hophuong@vinhuni.edu.vn

Ngo Manh Tien
Institute of Physics
Vietnam Academy of Science and Technology
Ha Noi, Vietnam
nmtien@iop.vast.vn

Ngo Duy Tan
Space Technology Institute
Vietnam Academy of Science and Technology
Ha Noi, Vietnam
ndtan@sti.vast.vn

Mai The Anh
School of Engineering and Technology
Vinh University
Nghe An, Vietnam
theanh@vinhuni.edu.vn

Duong Dinh Tu
School of Engineering and Technology
Vinh University
Nghe An, Vietnam
duongdinhtu@vinhuni.edu.vn

***Abstract*— This article introduces a fault-tolerant control scheme for a two-wheeled mobile robot. The focus of this model is to address the impact of faults within the actuators, a critical aspect that significantly impacts the robot's performance and operational efficiency. An observer is designed to monitor the dynamic state of the robot's system, allowing it to promptly identify and assess the actuator's faults that may arise during its operation. Based on this data, the extent of the fault's influence on the overall system is estimated, providing essential information for subsequent control decisions. Analyzing and synthesizing control laws are built on mathematics and Lyapunov stability theory. The simulation is done by MATLAB-Simulink to validate the efficiency of the proposed control law, contributing valuable insights to the domain of robotics and control engineering.**

Keywords: WMR, Faults observer, Lyapunov stabilizer, mobile robot, PID.

I. INTRODUCTION

Ensuring the reliability and stability of a controlled system has become increasingly crucial due to the potential negative impacts of faults occurring on sensors, actuators, and other components. Such faults would lead to undesirable performance and even instability in the system. Fault-tolerant control (FTC) enables robots to maintain reliable operation across diverse working environments, effectively preventing work termination resulting from tolerable faults [1]. Fault-tolerant control strategies are approached for linear systems in [2,3,4]. However, dealing with various strong nonlinear properties of actual systems in practice, the study of fault-tolerant control for nonlinear systems is of practical significance.

According to [5], wheeled mobile robots exhibit a nonlinearity multivariable nature and are characterized by strong interactions and time-varying parameters. Within the mobile robot system, the actuator stands as a component that is relatively susceptible to failure. Actuator faults can be classified into some types: failure actuator [7], stuck fault, partial loss of control effectiveness fault [8], and bias-actuator faults [9].

In addressing the issues of system performance degradation and instability resulting from the mentioned faults, several noteworthy fault-tolerant control methods targeting actuator faults have been introduced in works [10,11]. Among these methods, Adaptive Fault-Tolerant Control (AFTC) has been well-established as a practical approach to handling actuator faults and system uncertainties [12,13].

Numerous researchers have integrated intelligent methods such as neural networks and fuzzy logic into FTC schemes for nonlinear systems [14, 15], as these approaches can identify unknown nonlinear characteristics [16]. The adaptive technique is a viable method for designing controllers to compensate for actuator faults [17], as it enables the estimation of unknown parameters at each instant and facilitates rapid adjustments of control gains in response to parameter changes [18].

Utilizing an estimator to acquire system state/fault information is a highly effective approach to ensure the control performance of a system in the presence of faults. In this article, the heart of our FTC system lies in the design faults observer and implementation of the controller. From the Lyapunov stability theory and the fault observer's inputs, we have proposed a control law to ensure that the robot can effectively react against the adverse effects of actuator faults.

This research is expected to contribute valuable insights and solutions to specifically fault-tolerant and wheeled robot control.

II. PROBLEM FORMULATION

A. WMR Model

The design of a nonholonomic mobile robot with two driven wheels could be described as the structure in Figure 1.

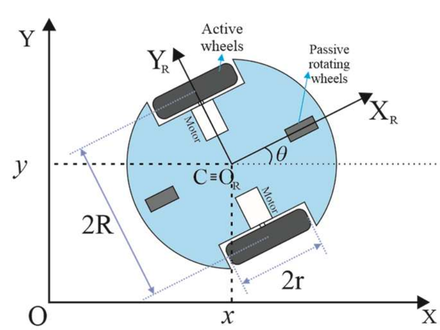

Figure 1. Differential-drive mobile robot

In Figure 1, C is the center of mass of the platform, $2R$ is the distance between two active wheels, and r is the radius of the active wheel. Two coordinate systems are used for mobile robot modeling and control: Inertial Frame $\{OXY\}$, and Robot Frame $\{O_R X_R Y_R\}$. The θ is the orientation angle of the robot in the inertial reference system, the robot position in the inertial and robot frame are $q=[x \quad y \quad \theta]^T$ and $q_R=[x_r \quad y_r \quad \theta_r]^T$.

A transformation matrix $Rot(\theta)$ was used to convert between fixed and robot coordinate systems [19]:

$$\dot{q}_R = Rot(\theta)\dot{q} \Rightarrow \dot{q} = Rot(\theta)^{-1}\dot{q}_R \quad (1)$$

where $Rot(\theta)^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

and $\dot{\phi}_R, \dot{\phi}_L$ are the rotational velocities of the right and left wheels, respectively. The translational and angular speeds of the robot in robot frame obtained as:

$$v = \frac{r(\dot{\phi}_R + \dot{\phi}_L)}{2},\ \omega = \frac{r(\dot{\phi}_R - \dot{\phi}_L)}{2R} \quad (2)$$

Combining (1) and (2) results in the velocity relationship between two coordinates which is presented as follows:

$$\Leftrightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r(\dot{\phi}_R + \dot{\phi}_L)}{2}\cos\theta \\ \frac{r(\dot{\phi}_R + \dot{\phi}_L)}{2}\sin\theta \\ \frac{r(\dot{\phi}_R - \dot{\phi}_L)}{2R} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

The forward kinematic model of the robot can be described as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S(q)V(t) \quad (4)$$

According to [20] and [21], the dynamical form of the nonholonomic mobile robot with two driven wheels, represented by a nonlinear dynamic model, can be expressed using the Euler-Lagrange formula as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_d = B(q)\tau + J^T(q)\lambda \quad (5)$$

With the implication that the WMR satisfies the conditions of non-slipping and pure rolling. The nonholonomic constraint is:

$$J(q)\dot{q} = 0 \quad (6)$$

Where q is an n-dimensional vector, τ is the r-dimensional input vector, τ_d is the vector of impact noise and model bias noise, bounded: $\tau_d = [d_x \ d_y \ d_\theta]^T$, λ is Lagrange constraint force product., $M(q)$ is a positively definite symmetric matrix of size *n* x *n*;, $C(q,\dot{q})$ is the centripetal and Coriolis matrix, $G(q)$ is the gravitation vector, $B(q)$ is input transformation matrix size *n* x *r* (*r*<*n*), $J(q)$ is the matrix associated with nonholonomic constraints.

In robot's structure here, "C" coincides with the midpoint on the axis between the wheels, the gravitational force is neglected, wheel frictions are given to the system with the fault defined in the kinematic model, the disturbance value is added to the control signal of the actuator, *m* is the mass and *I* represent the moment inertia of the robot, we have:

$$\begin{cases} M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, C(q,\dot{q}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, G(q) = 0, \\ B(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \\ J^T(q)\lambda = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \end{cases} \quad (7)$$

The dynamic equation of the robot can be shown as:

$$M(q)\ddot{q} = B(q)\tau + J^T(q)\lambda \quad (8)$$

The Lagrange multipliers λ are not known, so we can be eliminated $J^T(q)\lambda$ via the $S(q)$ matrix.

Differential formula (4), we have :

$$\ddot{q} = \dot{S}(q)V(t) + S(q)\dot{V}(t) \quad (9)$$

With (8), we can be rewrite (7) as:

$$M(q)S(q)\dot{V}(t) + M(q)\dot{S}(q)V(t) = B(q)\tau + J^T(q)\lambda \quad (10)$$

Multiplying of both sides of Equation (10) with the transformation matrix $S^T(q)$, we have:

$$\begin{aligned} S^T(q)M(q)S(q)\dot{V}(t) + S^T(q)M(q)\dot{S}(q)V(t) \\ = S^T(q)B(q)\tau + S^T(q)J^T(q)\lambda \end{aligned} \quad (11)$$

Herein $\quad S^T(q)J^T(q) = 0 \quad (12)$

So, (11) becomes:

$$M_1\dot{V}(t) + C_1V(t) = B_1\tau \quad (13)$$

$$\text{where } \begin{cases} M_1 = S^T(q)M(q)S(q) = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \\ C_1 = S^T(q)M(q)\dot{S}(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ B_1 = S^T(q)B(q) = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ R & -R \end{bmatrix} \end{cases} \quad (14)$$

Set: $u_1 = \tau_1 + \tau_2$, $u_2 = \tau_1 - \tau_2$.

The dynamics equation of the robot can be represented as follows:

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{rm} & \frac{1}{rm} \\ \frac{R}{rI} & \frac{-R}{rI} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{rm} & 0 \\ 0 & \frac{R}{rI} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{15}$$

B. Faults model in Orbital tracking control

With $q = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ and $q_R = \begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}^T$ are position in the inertial and robot frame, the robots satisfy the corresponding non-holonomic constraints. We define the tracking error model of the system as:

$$e_q = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ \mathrm{y_r} - y \\ \theta_r - \theta \end{bmatrix} \tag{16}$$

Differential formula (16):

$$\dot{e}_q = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - v + v_r \cos\theta_e \\ -\omega x_e + v_r \sin\theta_e \\ \omega_r - \omega \end{bmatrix} \tag{17}$$

The current velocities control input used for robot tracking is given as:

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e + k_x x_e \\ \omega_r + k_y v_r y_e + k_\theta \sin\theta_e \end{bmatrix} \tag{18}$$

k_1, k_2, k_3 are positive gain values.

In the scope of this article, we consider the actuator fault of the robot. In fact, the actuator's output can be expressed with the unknown constant actuator effectiveness factors.

$$\begin{cases} v_f = v - \Delta v \\ \omega_f = \omega - \Delta\omega \end{cases} \tag{19}$$

Where: $\Delta v = \mu_1 v,\ \Delta\omega = \mu_2 \omega\ \ (0 < \underline{\mu}_i < \mu_i < \overline{\mu}_i < 1)$

In this case, (17) will be changed. The current velocities control input in (18) will no longer be relevant.

The error model of the system, when considering the faults, is presented as:

$$\dot{e}_q = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - v + v_r \cos\theta_e + \Delta\omega y_e + \Delta v \\ -\omega x_e + v_r \sin\theta_e + \Delta\omega x_e \\ \omega_r - \omega + \Delta\omega \end{bmatrix} \tag{20}$$

C. Fault Observer

Consider a nonlinear system of the form

$$\begin{cases} \dot{x}(t) = f(x(t)) + g_1(x(t))u + g_2(x(t))f_a(t) \\ y(t) = h(x(t)) \end{cases} \tag{21}$$

Where $x \in R^n$ (state vector), $u \in R^m$ (input), $f_a \in R^p$ (faults).

In [22], an observer was introduced to estimate the disturbance in equation (21) under the assumption that the system state *x* and input *u* are known, and $\dot{f}_a = 0$. With $p(x(t))$ and $L(x(t))$ are the observer gains to be chosen, satisfying:

$$L(x(t)) = \frac{\partial p(x(t))}{\partial x(t)} \tag{22}$$

The observer equations can be represented as follows:

$$\begin{cases} \dot{z} = -L(x(t)) \begin{pmatrix} g_2(x(t))z + g_2(x(t))p(x(t) \\ + f(x(t)) + g_1(x(t))u \end{pmatrix} \\ \hat{f}_a(t) = z + p(x(t)) \end{cases} \tag{23}$$

Where $z \in R^p$ is the observer state, $\hat{f}_a(t)$ is the estimation of the faults $f_a(t)$.

Rewrite (20) to form (21) by setting the control variable and defining corresponding state variables, we have:

$$\dot{q}_e(t) = f(q_e(t)) + g_1(q_e(t))u(t) + g_2(q_e(t))f_a(t) \tag{24}$$

Here:

$$f(q_e(t)) = \begin{bmatrix} y_e\omega_r \\ v_r \sin\theta_e - x_e\omega_r \\ 0 \end{bmatrix}; u(t) = \begin{bmatrix} v_r \cos\theta_e - v \\ \omega_r - \omega \end{bmatrix};$$

$$g_1(q_e(t)) = g_2(q_e(t)) = \begin{bmatrix} 1 & -y_e \\ 0 & x_e \\ 0 & 1 \end{bmatrix}; f_a(t) = \begin{bmatrix} \Delta v \\ \Delta\omega \end{bmatrix}$$

According to [22], with system state $q_e(t)$, input $u(t)$ are known, and $\dot{f}_a = 0$. By choosing $p(q_e(t))$ and $L(q_e(t))$ satisfying (22). An observer model (23) can be applying for (24).

III. The Proposed Fault-Tolerant Controller

Based on the estimated value of errors affecting the actuator from the observer (23), the FTC is proposed to minimize the impact of faults on robot performance and ensure that the trajectory tracking problem is correct. There are two control loops in the controller: kinematic and dynamic control. The schematic structure of the controller is proposed, shown in Figure 2.

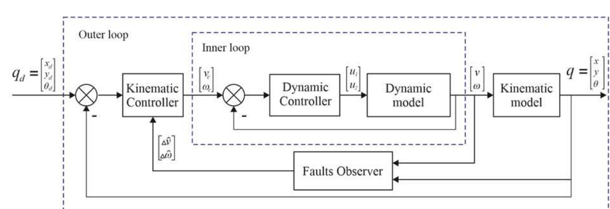

Figure 2. Structure of FTC

A. Kinematic controller

To track the desired trajectory for WMR under the impact of faults to the actuator, (20) will be considered.

The Lyapunov equation is selected as follows:

$$V = \frac{x_e^2 + y_e^2}{2} + \frac{1-\cos\theta_e}{k_y} \tag{25}$$

Derivation of (25) and from (20), we obtained:

$$\dot{V} = x_e\left(-v + v_r\cos\theta_e + 2\Delta\omega y_e + \Delta v\right) + ... + \frac{\sin\theta_e\left(\omega_r - \omega + k_y y_e v_r + \Delta\omega\right)}{k_y} \tag{26}$$

According to the Lyapunov stability theorem [23], for the system to be stable, $\dot{V} < 0$, it happens if and only if the condition (27) is met:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r\cos\theta_e + k_x x_e + 2y_e\Delta\omega + \Delta v \\ \omega_r + k_y v_r y_e + k_\theta \sin\theta_e + \Delta\omega \end{bmatrix} \tag{27}$$

By the observer model (23), $\hat{f}_a(t) = [\Delta\hat{v} \quad \Delta\hat{\omega}]^T$ is the estimation of actuator faults. So, the kinematic controller is selected by (28):

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_r\cos\theta_e + k_x x_e + 2y_e\Delta\hat{\omega} + \Delta\hat{v} \\ \omega_r + k_y v_r y_e + k_\theta \sin\theta_e + \Delta\hat{\omega} \end{bmatrix} \tag{28}$$

B. Dynamic controller

The controller aims to design $u_1(t)$ and $u_2(t)$, which create the forward speed v and angular speeds ω to asymptotically follow reference trajectories v_c and ω_c in (27). An SMC controller with PI-type sliding surface (SMC-PI) is proposed for the inner control loop.

1) PID controller:

The PID controller is a feedback control mechanism extensively applied across diverse engineering domains to regulate and stabilize systems effectively. Combining three constituents, the PID controller aims to provide a well-balanced response that achieves fast error correction and stable control. It helps dampen oscillations and overshooting by adjusting the control output based on the error change rate [24-26].

The PID controller is used:

$$u_{PID} = \begin{bmatrix} k_{Pv}v_e + k_{Iv}\int_0^t v_e dt + k_{Dv}\dfrac{d\omega_e}{dt} \\ k_{P\omega}\omega_e + k_{I\omega}\int_0^t \omega_e dt + k_{D\omega}\dfrac{d\omega_e}{dt} \end{bmatrix} \tag{29}$$

2) SMC-PI

According to [27], PI-type of sliding surface is proposed for SMC, which describe follow as:

$$\sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} v_e + k_\sigma \int v_e dt \\ \omega_e + k_\sigma \int \omega_e dt \end{bmatrix}, \; k_\sigma > 0 \tag{30}$$

Derivation of (30):

$$\dot{\sigma} = \begin{bmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \end{bmatrix} = \begin{bmatrix} \dot{v}_r - \dot{v} + k_\sigma v_e \\ \dot{\omega}_r - \dot{\omega} + k_\sigma \omega_e \end{bmatrix} \tag{31}$$

When $\dot{\sigma} = 0$:

$$\begin{cases} \dot{v} = \dot{v}_r + k_\sigma v_e \\ \dot{\omega} = \dot{\omega}_r + k_\sigma \omega_e \end{cases} \tag{32}$$

From (15) we have:

$$\dot{v} = \frac{1}{rm}u_1, \; \dot{\omega} = \frac{R}{rI}u_2 \tag{33}$$

Combine (32) with (33), the equivalent control law can be obtained:

$$u_{eq} = \begin{bmatrix} rm(\dot{v}_r + k_\sigma v_e) \\ \dfrac{rI}{R}(\dot{\omega}_r + k_\sigma \omega_e) \end{bmatrix} \tag{34}$$

The switching control law is formulated as follows:

$$u_{sw} = \begin{bmatrix} k_v sign(\sigma_1) \\ k_\omega sign(\sigma_2) \end{bmatrix}, \; k_v, k_\omega > 0 \tag{35}$$

In this case, to mitigate chattering, the conventional sign function has been substituted with the continuous tanh function, which approximates the sign function, and the ability to approximate the sign function dependent on the steepness of the tanh function [28]. The switching control law can be edited:

$$u_{sw} = \begin{bmatrix} k_v \tanh(\sigma_1) \\ k_\omega \tanh(\sigma_2) \end{bmatrix} \tag{36}$$

From (34), (36), the SMC control law with PI type sliding surface is showed as:

$$u_{SMC} = u_{eq} + u_{sw} = \begin{bmatrix} rm(\dot{v}_r + k_\sigma v_e) + k_v \tanh(\sigma_1) \\ \dfrac{rI}{R}(\dot{\omega}_r + k_\sigma \omega_e) + k_\omega \tanh(\sigma_2) \end{bmatrix} \tag{37}$$

C. Stability analysis

Based on (37), the positive Lyapunov equation is selected as:

$$V = \frac{1}{2}\begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \end{bmatrix} \tag{38}$$

Derivation of (38):

$$\dot{V} = \begin{bmatrix} \sigma_1\dot{\sigma}_1 \\ \sigma_2\dot{\sigma}_2 \end{bmatrix} \tag{39}$$

Combine (33) (31) (37), we have:

$$\begin{bmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \end{bmatrix} = \begin{bmatrix} \dot{v}_r + k_\sigma v_e - \dfrac{1}{rm}u_1 \\ \dot{\omega}_r + k_\sigma \omega_e - \dfrac{R}{rI}u_2 \end{bmatrix} \tag{40}$$

With control law was proposed in (39), (42) become:

$$\begin{bmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \end{bmatrix} = = \begin{bmatrix} -\dfrac{k_v \tanh(\sigma_1)}{rm} \\ -\dfrac{Rk_\omega \tanh(\sigma_2)}{rI} \end{bmatrix} \tag{41}$$

Hence,

$$\dot{V} = \begin{bmatrix} -\dfrac{k_v \sigma_1 \tanh(\sigma_1)}{rm} \\ -\dfrac{Rk_\omega \sigma_2 \tanh(\sigma_2)}{rI} \end{bmatrix} \leq \begin{bmatrix} -\dfrac{k_v}{rm}|\sigma_1| \\ -\dfrac{Rk_\omega}{rI}|\sigma_2| \end{bmatrix} \leq 0 \tag{42}$$

System (15) with SMC (35) to be stable according to the Lyapunov stability theorem [23].

IV. NUMERICAL SIMULATION

The wheel mobile robot with the structure in Figure 1 controlled by the control algorithms proposed in section III has been tested in MATLAB-Simulink. The parameters used in the simulation showed in Table 1, Table 2.

TABLE I. WMR PARAMETERS

Parameters	Notation	Values	Unit
The mass of robot	m	15	kg
Moment of inertia	I	2.5	kgm^2
Radius of active wheel.	r	0.1	m
½ distance between two active wheels	R	0.5	m

TABLE II. THE GAINS OF CONTROLLER

Notation	k_x	k_y	k_θ	k_v	k_ω	k_σ
Values	50	2	5	100	100	10
Notation	k_{Pv}	k_{Iv}	k_{Dv}	$k_{P\omega}$	$k_{I\omega}$	$k_{D\omega}$
Values	Used PID tuner application in the MATLAB/Simulink					

The reference trajectory was given as:

$$q_r = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} 5\cos(0.3t) \\ 5\cos(0.3t) \\ 0.3t + \dfrac{\pi}{2} \end{bmatrix} \tag{43}$$

And initial state: $x_0 = 0,\ y_0 = 0,\ \theta_0 = 0$.

The faults imposed on the actuator of the mobile robot is set at t>10s as: $\Delta v = 0.45v;\ \Delta\omega = 0.35\omega$.

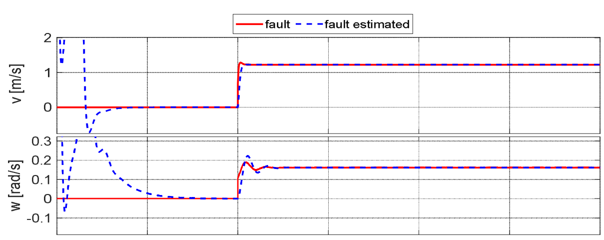

Figure 3. Faults and Estimate.

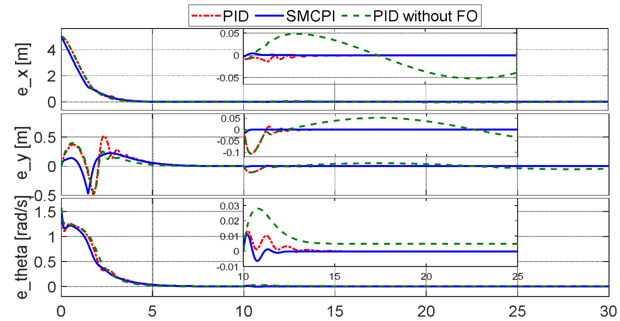

Figure 4. Error tracking of controllers.

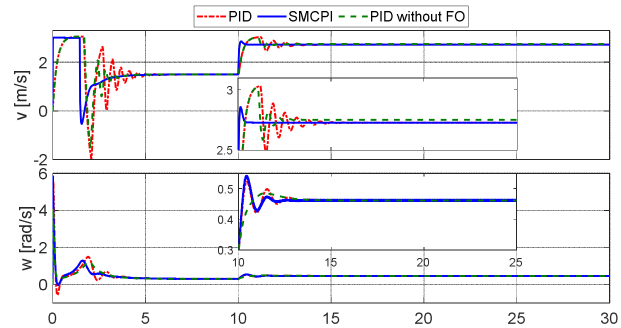

Figure 5. Linear and Angular velocitys.

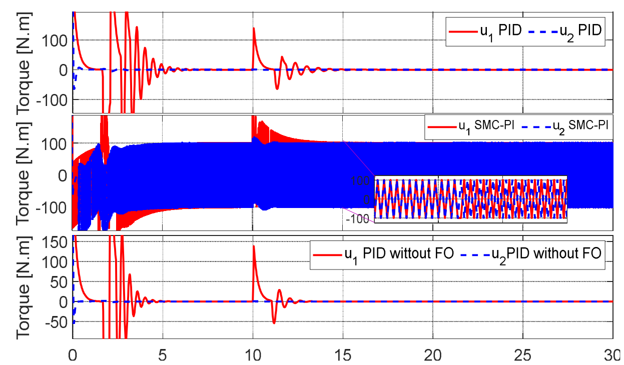

Figure 6. Controller's signal.

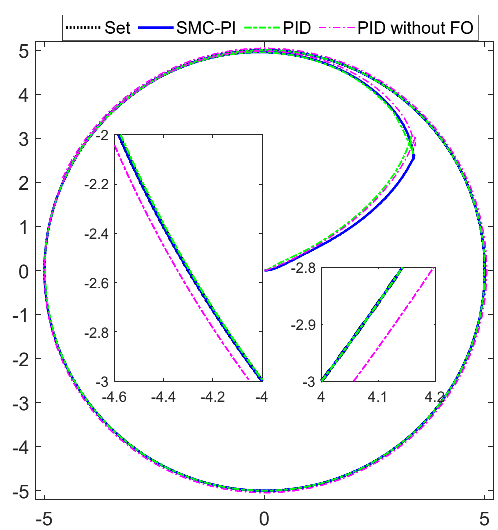

Figure 7. Tracking follows the set trajectory

Figure 3 depicts the effectiveness of the fault observer's observation capabilities. In the event of a fault in the robot's actuator, the fault observer detects the fault and applies corrective measures within the controller.

The system is subjected to faults when t>10s, as shown in Fig.7. The robot can flawlessly track the desired trajectory under the effect of actuator fault, and the tracking error quickly converges to zero (shown in Fig 4). Moreover, Figure 4 shows the performance and robustness of the SMC-PI-based fault-tolerant controller, which is superior to others.

Fig 5 also shows that the SMC-PI-based FTC is superior. The linear and angular velocities accelerate rapidly and stabilize when the robot moves to the reference trajectory.

V. CONCLUSION

The article focuses on constructing and validating a proposed the SMC-PI-based FTC for a two-wheel differential robot with considerations of the issues of faulty actuators. In order to minimize the impacts causes by the consequent errors of the system, a fault observer is designed to estimate and compensate for the errors in the controllers. The authors propose a fault-tolerant control law based on the flexible application of PID control with slip control and PI slip surface. Lyapunov stability theory and mathematical approach have been applied to prove the convergence and stability of the proposed control law. The simulation is done by MATLAB-Simulink to demonstrate the appropriateness of the proposed method and its effectiveness in tracking the trajectory of the proposed control law.

ACKNOWLEDGMENT

This research was funded by Vietnam's National project "Research, develop an intelligent mobile robot using different types of sensing technology and IoT platform, AI, and implemented in radioactive environment monitoring application", code: DTDLCN.19/23 of the CT1187 Physics development program in the period 2021- 2025

REFERENCES

[1] X. Qi, J.-T. Qi, D. Theilliol, Y.-M. Zhang, J.-D. Han, D.-L. Song and C.-S. Hua, "A review on fault diagnosis and fault tolerant control methods for single-rotor aerial vehicles," Journal of Intelligent & Robotic Systems, vol. 73, no. 1-4, pp. 535-555, January 2014.

[2] E. Naderi and K. Khorasani, "A data-driven approach to actuator and sensor fault detection, isolation and estimation in discrete-time linear systems," Automatica 85 (2017), pp. 165–178.

[3] Tae H. Lee, Chee Peng Lim, and Saeid Nahavandi, "Rodney G. Roberts, Observer-Based H∞ Fault-Tolerant Control for Linear Systems With Sensor and Actuator Faults", IEEE Systems Journal, Volume: 13, Issue: 2, June 2019, pp. 1981 - 1990.

[4] Li-Ying Hao, Lian-Sheng Zhou, "Fault-Tolerant Control of Linear Systems with Unmatched Uncertainties Based on Integral Sliding Mode Technique", Actuators 2022, 11(8), 241; https://doi.org/10.3390/act11080241

[5] Koubaa, A., Bennaceur, H., Chaari, I., et al., "Robot path planning and cooperation: foundations, algorithms and experimentations", (Springer, Berlin, 2018).

[6] Tao G.,'Direct adaptive actuator failure compensation control: a tutorial', J. Control Decis., 2014, 1, (1), pp. 75–101

[7] Li X.-J., Yang, G.-H.: "Robust adaptive fault-tolerant control for uncertain linear systems with actuator failures", IET Control Theory Applic., 2012, 6, (10), pp. 1544–1551

[8] Jin, X., Qin, J., Shi, Y., et al., 'Auxiliary fault tolerant control with actuator amplitude saturation and limited rate', IEEE Trans. Syst. Man Cybern. Syst., 2018, 48, (10), pp. 1816–1825

[9] Qin, J., Zhang, G., Zheng, W.X., et al., 'Adaptive sliding mode consensus tracking for second-order nonlinear multiagent systems with actuator faults', IEEE Trans. Cybern., 2019, 49, (5), pp. 1605–1615, DOI: 10.1109/ TCYB.2018.2805167

[10] Q. Y. Fan and G. H. Yang, "Active complementary control for affine nonlinear control systems with actuator faults", IEEE Trans. Cybern. PP (2017), no. 99, 1–12.

[11] M. Atitallah, M. Davoodi, and N. Meskin, "Event-triggered fault detection for networked control systems subject to packet dropout", Asian J. Control 20 (2018), no. 6, 2195–2206.

[12] Xiao-Zheng Jin, Ye-Xing Zhao, Hai Wang , Zhen Zhao, Xue-Ping Dong. "Adaptive fault-tolerant control of mobile robots with actuator faults and unknown parameters", IET Control Theory & Applications, Vol. 13 Iss. 11, 2019, pp. 1665-1672

[13] H. Zhou et al., "Adaptive fault–tolerant tracking control of nonaffine nonlinear systems with actuator failure", Abstr. Appl. Anal. 2014 (2014), no. 283403, 8

[14] Paul M. Frank, Birgit Köppen-Seliger, "Fuzzy logic and neural network applications to fault diagnosis", International Journal of Approximate Reasoning, Volume 16, Issue 1, January 1997, pp. 67-88

[15] Yang Cui, Guangren Duan, Xiaoping Liu & Hongyu Zheng, "Adaptive Fuzzy Fault-Tolerant Control of High-Order Nonlinear Systems: A Fully Actuated System Approach", International Journal of Fuzzy Systems, volume 25, pp.1895–1906 (2023).

[16] J. A. Farrell and M. M. Polycarpou, "Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches", Wiley-Interscience, Hoboken, NJ, USA, 2006.

[17] Xiao-Ni Zhang, Jian-Liang Wang, "Adaptive fault-tolerant tracking controller design against actuator stuck faults", Control and Decision Conference (CCDC), 2012 24th Chinese, 10.1109/CCDC.2012.6244655.

[18] 18. Imil Hamda Imran, Allahyar Montazeri, "An Adaptive Scheme to Estimate Unknown Parameters of an Unmanned Aerial Vehicle", 2020 International Conference Nonlinearity, Information and Robotics (NIR), 10.1109/NIR50484.2020.9290205

[19] Raj L, Czmerk A. "Modelling and simulation of the drivetrain of an omnidirectional mobile robot". Automatika: Časopis za Automatiku, Mjerenje, Elektroniku, Računarstvo i Komunikacije. 2017; 58(2): pp. 232–243.

[20] Chu Anh My,, Stanislav S. Makhanov, Nguyen A. Van, Vu M. Duc, "Modeling and computation of real-time applied torques and non-holonomic constraint forces/moment, and optimal design of wheels for an autonomous security robot tracking a moving target", Mathematics and Computers in Simulation, Vol. 170, pp.300–315, 2020. https://doi.org/10.1016/j.matcom.2019.11.002

[21] Takanori Fukao, Hiroshi Nakagawa, and Norihiko Adachi, "Adaptive Tracking Control of a Nonholonomic Mobile Robot", IEEE Transactions on Robotics And Automation, Vol. 16, No 5, October 2000.

[22] Ali Gholipour, M.J.Yazdanpanah, "Dynamic Tracking Control of Nonholonomic Mobile Robot With Model Reference Aadaption for Uuncertain Parameters", 2005.

[23] H. Khalil, Nonlinear Systems, Prentice Hall, Upper Saddle River, NJ, 2002.

[24] Åström, K. J., Hägglund, T. "PID controllers: Theory, design, and tuning (2nd ed.)". 1995, Instrument Society of America.

[25] Dorf, R. C., & Bishop, R. H. (2010). Modern Control Systems (12th ed.). Pearson.

[26] Ogata, K. "Modern Control Engineering", 2010, Prentice Hall.

[27] Kadu, C. B., Khandekar, A. A., & Patil, C. Y. (2018). Design of Sliding Mode Controller With Proportional Integral Sliding Surface for Robust Regulation and Tracking of Process Control Systems. Journal of Dynamic Systems, Measurement, and Control, 140(9), 091004. doi:10.1115/1.4039468

[28] Jinkun Liu, Sliding Mode Control Using MATLAB, 1st Edition - May 25, 2017. eBook ISBN: 9780128026700.