IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 November 2023, accepted 3 December 2023, date of publication 5 December 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339988

RESEARCH ARTICLE

Antifragile Control Systems: The Case of Mobile Robot Trajectory Tracking Under Uncertainty and Volatility

CRISTIAN AXENIE[1], (Senior Member, IEEE), AND MATTEO SAVERIANO[2], (Senior Member, IEEE)
[1]Department of Computer Science, Technische Hochschule Nürnberg Georg Simon Ohm, 90402 Nürnberg, Germany
[2]Department of Industrial Engineering (DII), University of Trento, 38123 Trento, Italy

Corresponding author: Cristian Axenie (cristian.axenie@th-nuernberg.de)

This work was supported by the European Union through NextGenerationEU under Grant CUP: E53D23001020001.

ABSTRACT Assistive care robots operate in cluttered, complex environments, akin to human residences, and need to face sensor and actuator faults during their operation without compromising safety. In these situations, the system must foresee and provide reactions that are beyond robust in addition to compensating for uncertainty and volatility in its functioning. We propose a new control design framework based on the principles of antifragility. In this work, we provide a formal concept of antifragile control and outline the design procedures for creating a mobile robot trajectory-tracking antifragile controller. An extended comparative evaluation against other controllers and a methodical investigation of the performance under parametrizable uncertainty and errors is also provided. Our findings demonstrate the effectiveness of antifragile closed-loop control in volatile and unanticipated circumstances.

INDEX TERMS Antifragile control, mobile robotics, trajectory tracking, uncertainty.

I. INTRODUCTION

Autonomous robotic vehicles are increasingly being adopted in everyday life, for instance through car-platooning applications [1], urban transportation services [2], and robotic wheelchairs [3]. The last class of vehicles is an effective solution for individuals with severe disabilities or handicaps who have no possibility of controlling a typical motorized wheelchair. In fact, robotic wheelchairs recognize the minimal input from the user, plan a route, and follow it within the constraints of the task's time constraints and the intricacy of the operating environment. Our goal in this exploratory work is to build a motion control algorithm that permits safe motion in situations where there is uncertainty and volatility about the operational states of the actuator and sensor. It should be noted that volatility pertains to the beginning and length of the fault in our study, whereas uncertainty defines the unknown amplitude and type of sensor and actuator faults.

The associate editor coordinating the review of this manuscript and approving it for publication was Nasim Ullah.

A. FRAGILITY-ROBUSTNESS-ANTIFRAGILITY SPECTRUM IN ROBOT CONTROL

Antifragility, as defined in [4] and [5], is a feature of a system that allows it to benefit from uncertainty, unpredictability, and volatility, in contrast to fragility. The reaction of an antifragile system to external perturbations or internal faults is beyond robust and resilient such that mild stresses can increase the system's future response by adding a significant anticipation component (see Fig. 1). In this work, we propose an alternative controller synthesis mechanism for mobile robots, extending the work in [9], based on the novel antifragile control framework introduced in [6] for therapy control and extended in [7] to traffic control. To provide an intuitive understanding of the antifragile control framework, we illustrate the fragility-robustness-antifragility spectrum inFig. 1. This is a basic representation of how different types of controllers would function when faced with escalating levels of uncertainty, such as sensor and actuator faults, such as wheel slippage, actuator fault, or sensor fault. Please note that this is a theoretical effect designed for

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

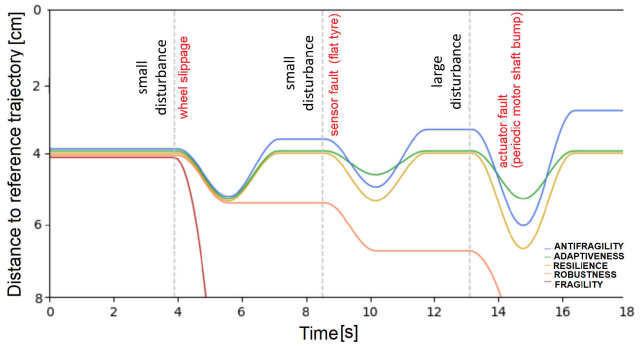

FIGURE 1. Fragility–robustness–antifragility spectrum in robot trajectory tracking. Existing approaches can compensate for an increase in amplitude and timing (i.e., uncertainty and volatility) of the disturbance of the system, while antifragile control aims at gaining from these adverse events.

pictorial reasons. The main goal of our work is to achieve an antifragile closed-loop control performance that not only gains from unforeseen, increasingly strong defects but also grows accustomed to them. An important note is that, in the current study, we extend the intrinsic and inherited fragility–robustness–antifragility detection heuristics of [5] through a novel type, termed *induced antifragility*. Our study proposes realizing antifragility through the design of a closed-loop control system that compensates for uncertainty (i.e. fault amplitude) and volatility (i.e. fault onset and duration) during trajectory tracking.

B. TRAJECTORY TRACKING FOR WHEELED MOBILE ROBOTS

Trajectory tracking control of nonholonomic mobile robots seeks to control a robot's motion to follow a specific time-varying trajectory. It is a basic motion control challenge that the robotics community has studied in great detail [9], [10], [11], [12], [13], [14], [15], [16]. Instead of offering yet another solution to an already well-known issue, our study aims to encourage the community to adopt a novel control synthesis strategy called antifragile control. Conventional methods obtain the reference trajectory by using a reference (virtual) robot; therefore, all kinematic constraints are implicitly taken into account in the reference trajectory. A stabilizing controller is then designed using a mix of feed-forward inputs estimated from the reference trajectory and feedback control rules [12], [13] or via Lyapunov stable time-varying state-tracking control laws [17], [18].

The design of robust controllers allows to compensate for uncertainty (e.g., wheel slip); these include dynamic modeling [14], robust [19], data-driven [20], and adaptive control [21], as well as sliding control [8], [15], [22]. Typical controllers aim to recover the desired robot behaviour by either "suppressing" uncertainty (i.e., robust control), modelling it (i.e., adaptive control), or predicting it (i.e., resilient control). In contrast, our antifragile controller pushes performance (i.e., minimal tracking error) through anticipation. Three key design components–explained in the sections that follow–help achieve this.

Uncertainty prediction can be effectively exploited to improve the controller response. In this regard, model predictive control anticipation capabilities have been used for mobile robot trajectory tracking [23], [24], [25]. On the other side, the goal of robust controllers is to provide (bounded) mediation actions that handle uncertainty, whereas our antifragile controller seeks to anticipate the uncertainty by accumulating "capacity" to accommodate the later effects. In the experiments, we compare our approach with the adaptive model predictive controller inspired by the work of [25].

Finally, the antifragile control is compared with a resilient control strategy where the control law is built to model specific uncertainties using fuzzy logic; the implementation is inspired by the work in [26]. It should be noted that this study is not an established framework prepared to overtake state-of-the-art, but rather an attempt to forge a new route in antifragile control synthesis with a real-world robotics application. We are excited to challenge the community to evaluate the framework instead of the solution to a well-known issue.

C. CONTRIBUTIONS

The main contributions of our study are:

- a control system design mapping the mobile robot trajectory tracking problem under sensor and actuator faults to the fragile–robust–antifragile continuum;
- an implementation of an antifragile controller for mobile robot trajectory tracking with closed-loop benefits from variability (i.e. fault amplitude) and volatility (i.e. fault onset and duration);
- a systematic characterization of mobile robot trajectory tracking problem under uncertainty and volatility (i.e., environment conditions, sensor and actuator faults in a space–time–intensity reference system);

II. MATERIALS AND METHODS

This section is dedicated to the models and tools we employed in our study and discusses the proposed approach for controller synthesis within the antifragile control framework.

A. WHEELED MOBILE ROBOT TRAJECTORY TRACKING

In our study, we consider a differential drive wheelchair, as shown in Fig. 2. In trajectory tracking mode, the real mobile robot must track a virtual mobile robot's trajectory under time constraints. The robotic setup was developed in [9] and [22]. The control objective is to compute the velocity of the robot such that its pose $P_r = [x_r, y_r, \phi_r]^\top$ follows a reference trajectory of the virtual robot $P_d = [x_d, y_d, \phi_d]^\top$. Such robots are subject to nonholonomic constraints, making motion perpendicular to the wheels impossible. The velocity of P_r is assumed in the direction of the axis of symmetry and, additionally, the wheels do not skid.

To improve tracking performance, we consider the robot's dynamics and divide the controller synthesis into two phases:

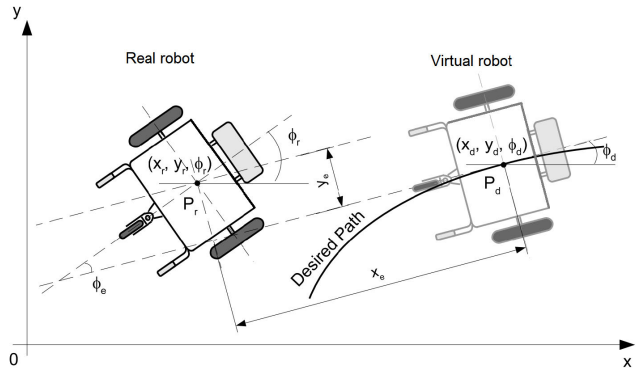

FIGURE 2. Description of the mobile robot trajectory tracking problem. Adapted with permission from [9].

- an inner loop that can be utilized to control both the linear and angular velocities – dynamic-level control.
- an outer loop to control the pose of the robot in the xOy reference frame– kinematic-level.

For the inner loop, we consider the DC motor model in [9], whereas, for the outer loop, the kinematic mobile robot model in [9]

$$\begin{cases} \dot{x}_r(t) = v_r(t)\cos(\phi_r(t)) \\ \dot{y}_r(t) = v_r(t)\sin(\phi_r(t)) \\ \dot{\phi}_r(t) = \omega_r, \end{cases} \tag{1}$$

where x_r and y_r are the Cartesian coordinates of the geometric centre of the mobile robot, v_r is the linear and ω_r is the angular velocity of the robot, and ϕ_r is the robot's heading angle. Knowing the wheel radius r and the distance between the wheels $2b$, it is straightforward to compute the Robot Dynamics Transform in Fig. 3, i.e., the relation between the linear v and angular ω velocities of the robot and the individual wheel angular velocities ω_R, ω_L.

The trajectory tracking errors can be described by the vector $[x_e, y_e, \phi_e]^\top$ (see Fig. 2). The designed controller needs to generate a command vector $[v_c, \omega_c]^\top$. Considering the model in (1), the error vector is given by

$$\begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} = \begin{bmatrix} \cos(\phi_d) & \sin(\phi_d) & 0 \\ -\sin(\phi_d) & \cos(\phi_d) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \phi_r - \phi_d \end{bmatrix}, \tag{2}$$

where the vector $[x_d, y_d, \phi_d]^\top$ is the virtual robot pose. The corresponding error derivatives are then computed as

$$\begin{cases} \dot{x}_e(t) = -v_d + v_r\cos(\phi_e) + y_e\omega_d \\ \dot{y}_e(t) = v_r\sin(\phi_e) - x_e\omega_d \\ \dot{\phi}_e(t) = \omega_r - \omega_d, \end{cases} \tag{3}$$

where v_d and ω_d are the desired robot linear and angular velocities, respectively. Note that a feasible desired trajectory for the mobile robot is pre-specified by the velocity planner in [27]. This approach generates speed profiles and trajectories that are comfortable for humans, as validated in [15].

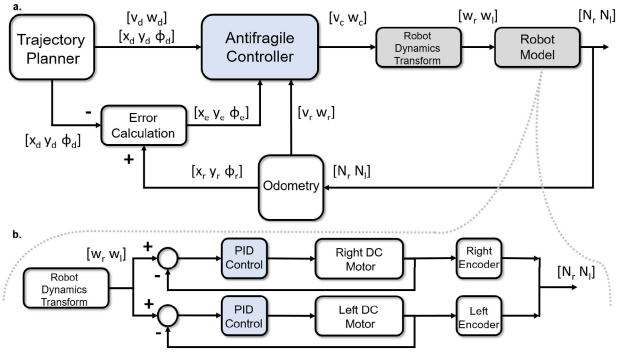

FIGURE 3. The proposed antifragile control framework consists of two nested loops. a) In the outer loop the antifragile controller generates the desired trajectory. b) The inner loop controls the individual velocities of the wheels.

B. ANTIFRAGILE CONTROL

This section is dedicated to introducing the mathematical apparatus of antifragile control, going from its theory and principles to the control synthesis for robot trajectory tracking under uncertainty. The control framework is shown in Fig. 3.

1) PRELIMINARIES

Let us consider the dynamical system associated with the mobile robot, assumed to be fully observable and controllable, in the form

$$\begin{cases} \dot{x} = f(x, u, d_x) \\ z = g(x, u, d_z), \end{cases} \tag{4}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the input, $z \in \mathbb{R}^m$ is the output, $d_x \in \mathbb{R}^n$ and $d_z \in \mathbb{R}^m$ are faults acting on the state and output of the system, respectively. Considering system (4), we can reformulate the definition of *antifragility* given in [5].

Definition 1 (Local Antifragility): The system (4) is locally antifragile iff $\forall\, x(t_0) \in \mathcal{X} \subset \mathbb{R}^n$, *for* $u = 0$, $d_x \in \mathcal{D}_x \subset \mathbb{R}^n$, *and* $d_z \in \mathcal{D}_z \subset \mathbb{R}^m$, *the error* $z - y_d = g(x, 0, d_z) - y_d$, *where* $y_d \in \mathbb{R}^m$ *is the desired output, is a convex function.*

Definition 2 (Global Antifragility): The system (4) is globally antifragile iff $\forall\, x(t_0) \in \mathbb{R}^n$, *for* $u = 0$, $d_x \in \mathcal{D}_x \subset \mathbb{R}^n$, *and* $d_z \in \mathcal{D}_z \subset \mathbb{R}^m$, *the error* $z - y_d = g(x, 0, d_z) - y_d$, *where* $y_d \in \mathbb{R}^m$ *is the desired output, is a convex function.*

Consequently, *fragility* is defined as the "opposite" of antifragility, i.e.,

Definition 3 (Fragility): The system (4) is fragile iff $\exists\, x(t_0) \in \mathcal{X} \subset \mathbb{R}^n$ *such that, for* $u = 0$, $d_x \in \mathcal{D}_x \subset \mathbb{R}^n$, *and* $d_z \in \mathcal{D}_z \subset \mathbb{R}^m$, *the error* $z - y_d = g(x, 0, d_z) - y_d$, *where* $y_d \in \mathbb{R}^m$ *is the desired output, is a concave function.*

An uncontrolled dynamics ($u = 0$) can be fragile in nature, but it is possible to *induce* an antifragile behaviour through control.

Definition 4 (Induced Antifragility): The system (4) is locally inducedly antifragile iff $\forall\, x(t_0) \in \mathcal{X} \subset \mathbb{R}^n$, *and for* $d_x \in \mathcal{D}_x \subset \mathbb{R}^n$ *and* $d_z \in \mathcal{D}_z \subset \mathbb{R}^m$, $\exists\, u$ *that renders the error* $z - y_d = g(x, u, d_z) - y_d$, *where* $y_d \in \mathbb{R}^m$ *is the desired*

output, a convex function. If the previous holds $\forall\, x(t_0) \in \mathbb{R}^n$, *we say that the system (4) is globally induced antifragile.*

Producing an antifragile behaviour in a feedback control loop (induced antifragility) demands a unique design and synthesis method in which: 1) *redundant overcompensation* may drive the system into an overshooting mode that accumulates extra capacity and capability in anticipation; 2) *structure-variability* can elicit stressors that carry intrinsic information which emerges only under volatility and unpredictability of the system dynamics affected by the application of a high-frequency component; 3) *time scales separation* of the interacting system's dynamics undergoes an order-reduction while driven towards the desired antifragile operation region. The following sections are dedicated to integrating the methodology above with the robot trajectory tracking problem.

2) CONTROL SYNTHESIS

The work of [6] and [7] was an attempt to cast the control design in geometric control and Riemannian geometry. This made it possible for us to operate in a coordinate-free setting by depending on a manifold's embedding into an expanded dynamical space, which allowed for the construction of control laws that were more straightforward and suitable for manifolds with curvature. In this work, a concrete control synthesis method is provided, while consolidating this methodology and reducing a number of prior assumptions.

Our goal is to design a controller that forces the robot to track a prescribed trajectory (i.e., a velocity-parametrized reference temporal evolution) with certain geometrical properties. The problem can be also formulated to compute a control signal (i.e., reference linear and angular velocities) such that the robot's state trajectory confines itself to desired dynamics where the error vector (x_e, y_e, ϕ_e) is minimized. In other words, in order to drive the closed-loop system state evolution to a manifold such that the longitudinal x_e, the lateral error y_e, and the angular error ϕ_e are internally mutually coupled leading to convergence of all three variables. In our control synthesis, we decouple the two internal control loops in Fig. 3 (see the darkly shaded boxes termed Antifragile Control and PID control) in order to describe the specific design steps of 1) redundant overcompensation; 2) structure-variability; and 3) time scales separation for uncertainty isolation.

3) TIME SCALE SEPARATION

Given the interactions between the two nested control loops in Fig. 3, in order to handle uncertainty and high-frequency phenomena, the closed-loop system separates the time scales of the loops. A very useful tool for such interventions is (singular) perturbation theory [28]. The high-frequency dynamics are considered in this paradigm by taking them into account on a different timeline. Similar to a parallel transport map on the Riemannian manifold of the system state trajectory, this transformation is accomplished by a dynamic change in the order of the controlled system as a parameter perturbation. Such a change in the controlled system dynamics is more "abrupt" than a normal perturbation to which the system is exposed—hence the singular perturbation. The main argument for using such an approach in our antifragile control design resides in the fact that such high-frequency phenomena are able to build capacity in reacting to high-amplitude changes in the robot's operation (e.g., wheel slipping, flat tire, and shaft bending). The core idea of time scale separation is to capture the dominant phenomena dynamics and then capture the stressors and is typically achieved by "outer" series expansions or "inner" boundary layer expansions [28], as shown in Fig. 4. More precisely, considering singularly perturbed dynamical systems, we can benefit from solutions with fast variation zones. These areas, which may be seen in the solution or its derivatives, are referred to as *layers*, and they frequently exist near the domain border (see Fig. 4). Inner solutions are found for the layers, whereas outer solutions are obtained for the regular distinguished limits. The uniform solution is described by the curvature (i.e., second-order effect) of the overlapping region between the inner and the outer layer. Interestingly, this can be exploited in our design to define fragile and antifragile control regimes (Definitions 1) to 4)) depending on the curvature in the overlap region of the solutions (i.e., attractors/solutions in the state space). From Definition 4), the antifragile region is the convex region of the solution curve (see Fig. 4a). Hence, the closed-loop system response is antifragile if the curvature is negative, otherwise is fragile.

In the case of robot trajectory tracking, the reference trajectory is a path, which is an explicit function of time (see Fig. 4b). To achieve a smooth robot movement, the trajectory must be twice differentiable to give continuous velocity and acceleration. In our experiments, we used the method in [8] to obtain smooth longitudinal and angular velocity profiles ($v(t)$ and $\omega(t)$) that are compatible with the characteristics of the actuators of the robot (i.e., DC motor regimes) and assigned total path length, and it must comply with human comfort travel.

Given the explicit expression of curvature $\kappa_{i,i+1}$ between consecutive points i and $i+1$ as

$$\kappa_{i,i+1}(t) = \frac{\dot{x}_{i,i+1}\ddot{y}_{i,i+1} - \ddot{x}_{i,i+1}\dot{y}_{i,i+1}}{\dot{x}^2_{i,i+1} + \dot{y}^2_{i,i+1}}, \tag{5}$$

and its sign, the computed signals $v(t)$ and $\omega(t)$ to move the robot from point p_1 to point p_2 will generate two possible paths, *fragile* (red) and *antifragile* green (i.e., antifragile control signals for tracking if the curvature is negative, otherwise fragile). Both fragile (red) and antifragile (green) are feasible solutions. The antifragile solution will limit the curvature and, implicitly, the magnitude of the control input $v(t)$ and $\omega(t)$. This reduces the stress on the robot's actuators and ensures higher robustness in the case of uncertainty while maintaining comfort. The fragile path and curvature will have more prominent curvature variation at the beginning and at

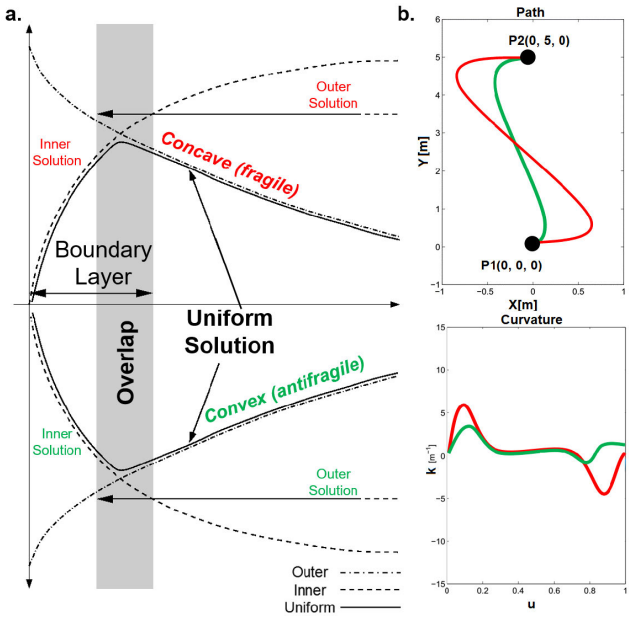

FIGURE 4. Singular perturbation for time scale separation in antifragile controller synthesis. Using boundary layers and matched asymptotic expansions to probe antifragile behaviour. a) Generic depiction of a boundary layer and the types of solutions in singularly perturbed dynamic systems. b) Mapping the boundary layers and shape (convexity/concavity) of the solution to robot velocity planning akin to the desired dynamics to track in the presence of uncertainty.

the closing of the spline respectively, which might reduce robustness in case of wheel slippage or mechanical damage during navigation.

4) REDUNDANT OVERCOMPENSATION

In the following, we revisit the core idea of time scale separation, graphically depicted in Fig. 4. Let us consider a more general form of the robot model in (1) as

$$\begin{cases} \dot{x} = f(x, z, \varepsilon, t), \ x(t_0) = x^0, \ x \in \mathbb{R}^n \\ \varepsilon \dot{z} = g(x, z, \varepsilon, t), \ z(t_0) = z^0, \ z \in \mathbb{R}^m, \end{cases} \tag{6}$$

where f, g are continuous differentiable functions of x, z, ε, t. The scalar $\varepsilon > 0$ quantifies all small parameters of the system (i.e., I_m, I_w, etc.), which in the antifragile control framework are termed as *stressors* for capacity build-up. Furthermore, considering two small time constants T_1 and T_2, we can assume that $T_1 = \varepsilon$ and $T_2 = \alpha\varepsilon$, where $\alpha = T_2/T_1$ is a known constant. Now, if we set $\varepsilon = 0$ in (6), the dimension of the state space of the system reduces from $n+m$ to n because the second equation degenerates into $0 = g(x, z, 0, t)$ where z can rapidly converge to a root of g due to its velocity $\dot{z} = g/\varepsilon$, which can be high if ε is small.

From a more intuitive perspective, the model in (6) is a reduced-order modelling technique, which allows us to convert the robot's dynamics simplification (reduction) into a parameter perturbation, called "singular". The solutions of the "slow" dynamics $x(t, \varepsilon)$ and the "fast" dynamics $z(t, \varepsilon)$ of the singularly perturbed system in (6) consist of a fast boundary layer and a slow quasi-steady-state, as shown in Fig. 4. From the Riemannian geometry perspective of antifragile control, there exists a manifold M_ε depending on ε that can be defined in the space $n + m$ of x and z such that $M_\varepsilon : z = \phi(x, \varepsilon)$. This reduces the dimension of the state space by restricting it to remain on M_ε. The redundant overcompensation and time scale separation's explicit concept and implementation is now introduced. For this purpose, we utilize a Proportional Integral Derivative (PID) controller for every DC motor to achieve redundant overcompensation. The problem is then formulated in the Laplace domain with complex frequency s in order to concentrate on the singular perturbation design. Later we will come back to the time domain to describe the actual time-scale separation.

The transfer function of the PID controller is $u(s)/e(s) = K_p(1 + K_D s + K_I \frac{1}{s})$ where the error $e(s) = w(s) - y(s)$. Furthermore, we consider the DC motor model of [9], where L is small in practice and may serve as our parameter ε. This maps to our generic formulation in (6) if $\omega = x$, $i = z$ and $R \neq 0$. Model reduction is addressed by ignoring the inductance L and solving $-k\omega - Ri + u = 0$ to obtain the value of the current $i = \frac{u - k\omega}{R}$ which we then use to obtain the first-order model of the DC motor in the form $J\dot{\omega} = -\frac{k^2}{R}\omega + \frac{k}{R}u$. This accounts for finding the manifold M_ε and restricting the DC motor dynamics to remain on it.

In our design, the goal is to express that the integral effect of the PID control is much slower than the proportional and the derivative components. The singular perturbation theory supports us in the task of rewriting the control law u given the fact that K_P and K_D offer speed and stability of the system, respectively, whereas K_I reduces the error e to zero. Assuming that $K_I = \varepsilon \hat{K}_I$, and changing notation as to $k_1 = K_P$, $k_2 = K_P K_D$, and $k_3 = K_P \hat{K}_I$, the control law is

$$u(s) = (k_1 + k_2 s + \varepsilon k_3 \frac{1}{s}) e(s). \tag{7}$$

The fast variables will build capacity for the antifragile response to uncertain events in the robot's operation, such as wheel slipping, flat tire, and DC motor actuator shaft bending. More precisely, the choice of k_1 and k_2 must be done such that in the PD part of the motor controller, the system matrix is Hurwitzian.

5) STRUCTURE VARIABILITY

Variable Structure Control (VSC) offers a very powerful tool for handling uncertainty in closed-loop [29], by providing a powerful reaction to minimal deviations from a chosen constraint. Typically, VSC is practically implemented through sliding mode control [29].

In our framework, VSC is related to redundant overcompensation, as shown in Fig. 4 and 5. The antifragile controller needs to provide a proper control signal (i.e., a pair (v, ω)), such that the path from origin to destination (see Fig. 5) is tracked under time constraints, uncertainty about the driving surface, actuator failures, and with increased comfort (i.e., minimal curvature). To deal with uncertainty about the

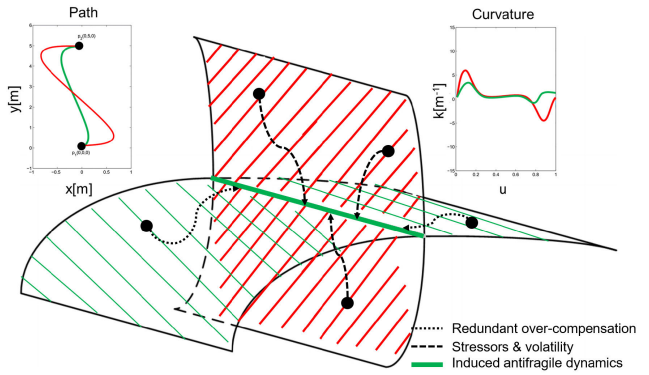

FIGURE 5. VSC through sliding modes in antifragile control synthesis with fragile (red) and antifragile (green) control signals and dynamics. The antifragile control manages to drive the system's dynamics toward the antifragile region benefiting from redundant overcompensation, stressors, and volatility.

driving surface and actuator failures, this is accomplished by properly synthesizing the control law (see initial conditions of the system dynamics converging from the green manifold to the induced antifragile dynamics from Fig. 5). When starting from a fragile region of the system's solutions (the red manifold in Fig. 5) the controller handles stressors and volatility (i.e., increased curvature of the trajectory and then tracking the desired path) by using the provided "inertia" to converge to the induced antifragile dynamics.

Now, to attain such dynamics under induced antifragile control, the following sub-section covers the relevant design steps of a VSC concerning our problem.

a: DESIRED MANIFOLD SELECTION

To obtain system performance during the sliding motion, a sliding manifold must be selected in this control design stage that has a lower order than the system. This stage is rather application-dependent, so we'll explain why we chose the trajectory tracking problem we did. We start from the canonical form of a sliding manifold s depending on system state dynamics x (see (6) for the general dynamical system formulation) in (8).

$$\dot{s} = \frac{\partial s}{\partial x}\dot{x} = \frac{\partial x}{\partial x}s(x) = \lambda_1 x_1 + \lambda_2 x_2 + \ldots + x_n = 0 \quad (8)$$

where the coefficients λ_i in $\dot{s}$ define the desired characteristics of the sliding mode, that is the characteristics of the closed-loop system after the manifold reaching phase, as broadly described in [29]. Finding these parameters is typically formulated as an optimization problem, and solved using linear programming techniques (e.g., Linear Quadratic (LQ) approach), as shown in [29]. Here, a criteria for a second order system $J = \int_{t_s}^{\infty}(x_1^\top Q_{11}x_2 + 2x_1^\top Q_{12}x_2 + x_2^\top Q_2 x_2)dt$ was minimized to get the optimal system motion manifold. Considering $Q_{12} = 0$ then the optimal control $x_2 = -Q_{22}^{-1}A_{12}^\top P x_1 = -kx_1$ where P is a p.d. matrix solution of the Riccati equation $A_{11}^\top P + PA_{11} - PA_{12}Q_{22}^{-1}A_{12}^\top P = -Q_{11}$ where A is the input matrix of the system. The switching function is obtained by considering $s(x) = kx_1 + x_2 = [Q_{22}^{-1}A_{12}^\top P, I]x$.

In our case, the desired motion manifold choice imposes that the longitudinal error x_e, the lateral error y_e, and the angular error ϕ_e are intrinsically coupled in order to guarantee mutual convergence. Given the robot error in the outer loop (see Fig. 2) defined in Equation 2, the desired motion manifolds are chosen as below.

$$\begin{cases} s_1 = \dot{x}_e + \lambda_1 x_e \\ s_2 = \dot{y}_e + \lambda_2 y_e + \lambda_0 \text{sgn}(y_e)\phi_e \end{cases} \quad (9)$$

with $\lambda_0, \lambda_1, \lambda_2 > 0$. Interestingly, if s_1 converges to 0 then x_e converges to 0. Additionally, if s_2 converges to 0, then at steady state $\dot{y}_e = -\lambda_2 y_e - \lambda_0 \text{sgn}(y_e)\phi_e$. Here, for negative lateral error $y_e < 0$ then $\dot{y}_e > 0$ if and only if $\lambda_0 < \lambda_2 \frac{|y_e|}{|\phi_e|}$ and for a positive lateral error $y_e > 0$ then $\dot{y}_e < 0$ if and only if $\lambda_0 < \lambda_2 \frac{|y_e|}{|\phi_e|}$.

b: CONTROL LAW DESIGN

In this step, we need to design a switched feedback control law that satisfies the reaching condition (see Fig. 5) and drives the system trajectory to the manifold in finite time and keeps it there thereafter. In this study, we consider the analysis of [8] and [9] that uses Gao's reaching law [30]. This approach employs the differential equation $\dot{s} = -Q\text{sgn}(s) - Ph(s)$, where $Q = \text{diag}[q_1, q_2, \ldots, q_n]$ with $q_i > 0, i = 1, \ldots, n$; $P = \text{diag}[p_1, p_2, \ldots, p_n]$, with $p_i > 0, i = 1, \ldots, n$; $\text{sgn}(s) = [\text{sgn}(s_1), \text{sgn}(s_2), \ldots, \text{sgn}(s_m)]^\top$; $h(s) = [h_1(s_1), h_2(s_2), \ldots, h_m(s_m)]^\top$; and $s_i h_i(s) > 0$ with $h_i(0) = 0$. The reaching time needed by the system state x to move from an initial state to the switching manifold s_i is finite and given by

$$T_i = \frac{1}{p_i}\ln\frac{p_i|s_i| + q_i}{q_i} \quad (10)$$

With the reaching law equation in conjunction, we can now identify the control rule u that directs the robot along the designated tracking trajectory. In our situation, the control law is calculated using the first derivative (i.e., the velocity) of $s(x)$ along the reaching mode trajectory (see Fig. 5) as $\dot{s} = \frac{\partial s}{\partial x}(A(x) + B(x)u) = -Q\text{sgn}(s) - Ph(s)$ where, in the generic form, A is the state transformation matrix and B is the control input gain matrix. We then have the control law given by $u = -(\frac{\partial s}{\partial x}A(x) + Q\text{sgn}(s) + Ph(s))(\frac{\partial s}{\partial x}B(x))^\top$. In this instance, the resultant sliding mode is not preassigned but instead adopts a first-order switching scheme's natural state trajectory, as demonstrated in [29]. As shown in Fig. 5, the switching takes place depending on the location in the state space of the initial state.

In our particular case, we choose the control law u, derived in [8] and [9], defined as

$$\dot{s} = -Qs - P\text{sgn}(s), \quad (11)$$

with $P, Q > 0$. Opposite to the approach of [29], we use the proportional term $-Qs$ instead of the $\text{sgn}(s)$ to force the system's state to approach the switching manifold faster when $\dot{s}$ is large, while the discontinuous (magnitude) component is given by $h(s) = \text{sgn}(s)$ in the second term (i.e.. the constant

rate reaching). Now, given the ordinary form for control of the mobile robot

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{12}$$

and the derivative of the manifold (9) as

$$\begin{cases} \dot{s}_1 = \ddot{x}_e + \lambda_1 \dot{x}_e \\ \dot{s}_2 = \ddot{y}_e + \lambda_2 \dot{y}_e + \lambda_0 \text{sgn}(y_e)\dot{\phi}_e \end{cases} \tag{13}$$

Through mathematical manipulation, we obtain the control law $u = [v_c, \omega_c]^\top$ where the linear acceleration is
and the angular velocity is

$$\omega_c = \frac{-Q_2 s_2 - P_2 \text{sgn}(s_2) - \lambda_2 \dot{y}_e - \dot{v}_r \sin(\phi_e) + \dot{\omega}_d x_e + \omega_d \dot{x}_e}{v_e \cos(\phi_e) + \lambda_0 \text{sgn}(y_e)}. \tag{15}$$

Note that, based on the study of [8], the sign function sgn(·) in the control signals can be replaced in the practical implementation by the saturation function sat(·) with thresholds to reduce the chattering phenomenon. Now, we define the Lyapunov function candidate, introduced in [9] and [22],

$$V = \frac{1}{2} s^\top s. \tag{16}$$

The time derivative $\dot{V}$ is given by

$$\begin{aligned} \dot{V} &= s_1(-Q_1 s_1 - P_1 \text{sgn}(s_1)) + s_2(-Q_2 s_2 - P_2 \text{sgn}(s_2)) \\ &= -s^\top Q s - P_1 |s_1| - P_2 |s_2|. \end{aligned} \tag{17}$$

For $\dot{V}$ to be negative semi-definite, we choose Q_i and P_i such that $Q_i P_i \geq 0$. Then, given that $V > 0$ and that $\dot{V} \leq 0$, the control law is stable in the Lyapunov sense, as shown in [8]. Finally, the single-wheel velocity commands for the mobile robot are given by

$$\begin{cases} \Omega_r = \dfrac{v_c + b\omega_c}{r} \\ \Omega_l = \dfrac{v_c - b\omega_c}{r}, \end{cases} \tag{18}$$

where, following the conventions in Fig. 2, r is radius of the driving wheels, b is half the distance between the driving wheels, v_c is the computed control velocity, and ω_c the computed control angular velocity (see 14, as shown at the bottom of the next page and 15). These values are subsequently sent to the inner loop of the closed-loop system in Fig. 3, where the encoder revolutions N_r and N_l are available from odometric computations.

C. COMPETITIVE CONTROL ALGORITHMS

For the trajectory tracking problem, we chose competitive approaches among the most well-known approaches within the spectrum outlined in Figure 1. It should be noted that the goal of this exploratory work is to provide the community with a novel antifragile control synthesis method that benefits from uncertainty.

1) ROBUST CONTROL

For the robust control, we have chosen sliding mode control, as a very powerful method for robot trajectory tracking control, because it shares the advantages of variable structure controller design. The specific control synthesis is based on the work of [15]. In our experiments, the sliding mode controller is denoted as ROBUST. The proposed controller used also variable structure synthesis based on equivalent control

$$\begin{cases} u_{eq1}(t) = \dfrac{-D_1(t)}{\alpha(t)\cos(\phi_e(t))} \\ u_{eq2}(t) = \dfrac{-D_2(t)}{\beta} \end{cases} \tag{19}$$

where r is the wheel radius, $\alpha = 1/rm(t)$ and $\beta(t) = b/rI(t)$ uncertainty parameters in mass m and inertia I, and $2b$ is the robot's base width. In (19), D_1 and D_2 are two functions of the kinematic error derivative in (3).

2) ADAPTIVE RECEDING HORIZON CONTROL

To approach adaptive receding horizon control, we considered model predictive control (MPC) as a suitable candidate given its prediction capabilities which contrast well with the anticipation capabilities of the antifragile controller. More precisely, our design is based on the controller design of [25]. The suggested controller uses the robot's dynamic model as the controller model, which increases tracking accuracy while simultaneously accounting for the robot's dynamic stability during the tracking procedure. Additionally, by adaptively raising the weight of the cost function, the driving comfort issue caused by the use of a typical MPC controller when the vehicle deviates from the intended course is resolved. The model predictive controller is referred to as ADAPTIVE in our experiments. In order to enable the robot to precisely follow the target trajectory and achieve lateral stability, MPC-based trajectory tracking control aims to minimize the error between the anticipated output variables and the reference values. Therefore, the cost function was constructed as follows

$$J = \|Q(y(t) - \hat{y}_{ref}(t))\|^2 + \|Ru(t)\|^2 \tag{20}$$

where Q and R are weighting matrices of the controlled outputs and inputs, $y(t)$ is the 2D location and the heading angle, $\hat{y}_{ref}(t)$ is the reference location and the heading angle in prediction horizon, and $u(t)$ is the control input vector.

3) RESILIENT CONTROL

Finally, in order to represent resilient controller design, a fuzzy logic controller is considered as it provides an effective approach to approximate any smooth nonlinear dynamics in the form of IF-THEN rules. The concrete implementation in our experiments is based on the work of [26]. This study proposes a trajectory-tracking method that imitates human driving behaviour using a fuzzy logic set of rules. Estimated information about the next curve in front of the robot is sent into the fuzzy system, and its corresponding

output is the cruising velocity required for the robot to safely follow the path within the allotted time. The fuzzy logic controller is referred to as RESILIENT throughout our study where a 4-rule Takagi-Sugeno-Kang fuzzy inference system was developed. The model provides an output in a crisp format that can be immediately applied to the actuators of the robot. The control signal is

$$u = [v_l; v_r] = [k_d(t)d_e + k_t(t)\phi_e; k_d(t)d_e - k_t(t)\phi_e], \quad (21)$$

where d_e is the Euclidian distance error in Cartesian space, ϕ_e is the heading error of the robot, and k_d and k_t positive sub-unit proportional gain factors.

III. EXPERIMENTS AND RESULTS

In order to evaluate the control strategies and demonstrate the benefits that an antifragile design brings, we have designed a systematic analysis and evaluation framework that:

1) generates reference trajectories for the closed-loop robot control,
2) supports the induction of user-defined uncertainty injection, e.g., wheel slippage, actuator fault, or sensor fault,
3) supports the parametrization (i.e., timing, duration, amplitude, frequency) of user-defined uncertainty injection,
4) compares the performance when different types of uncertainty and/or faults are injected.

Considered controllers are parameterized as follows:

- ROBUST control: separated x_e and ϕ_e surfaces in the sliding mode design; PID control parametrization;
- ADAPTIVE control: MPC with cost function based on lateral position error y_e and heading error ϕ_e;
- RESILIENT control: 2 inputs (d_e and $|\phi_e|$), 2 outputs (v_l, v_r) controller; 4 IF-THEN rules;
- ANTIFRAGILE Control: combined x_e, y_e errors and separated ϕ_e surfaces in the variable structure component design; singular perturbation PID control.

All the experiments, analysis, and additional experiments not discussed in this paper, can be reproduced through the codebase available on GitHub.[1]

A. IDEAL TRAJECTORY TRACKING ROBOT CONTROL

We first analyze the basic behaviour of the selected control approaches on the basic (fault-free) task. This will help us get some intuition of how each control solves the trajectory tracking.

We observed that profiles of the velocities and accelerations are very close to the profile of the reference ones (see Fig. 6). The trademarks of ANTIFRAGILE control in

[1]https://gitlab.com/akii-microlab/antifragile-robot-control/

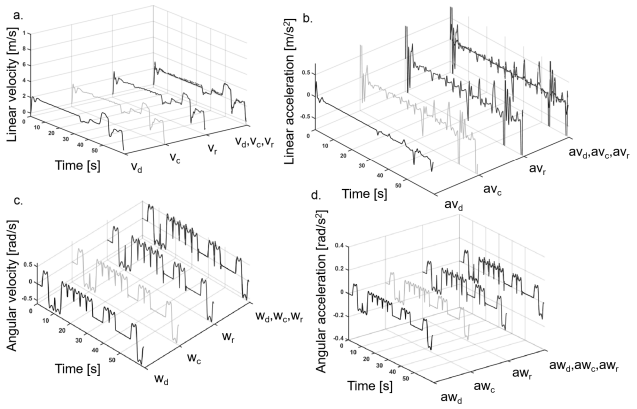

FIGURE 6. Analysis of the ANTIFRAGILE controller: a) linear velocity, b) linear acceleration, c) angular velocity, and d) angular acceleration.

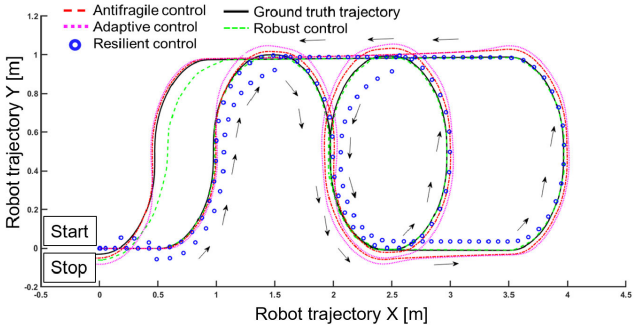

FIGURE 7. Comparative qualitative results on trajectory tracking.

the actual control signals for the robot motion are already visible in Fig. 6. As an example, the variable structure control component of the antifragile control influences the rate of change of linear velocity and, consequently, capacity building in handling fast-changing curvature values apparent through an overshoot in the trajectory (see also the rightmost loop in Fig. 7). Interestingly, the capacity-building feature of the ANTIFRAGILE controller is active (i.e., overshooting) when exiting a section of the trajectory from high curvature to low curvature, whereas the anticipation feature is active when exiting a section of the trajectory from low curvature to high-curvature (see Fig. 7 a - leftmost loop toward STOP). When considering the kinematic assessment of the ANTIFRAGILE trajectory tracking control in Fig. 8, the longitudinal error x_e is kept around 0, with a deviation of maximum 3 cm, whereas the lateral error y_e varies largely due to the often changes in the direction of the robot (see Fig. 7). However, the controller compensates jointly through the variable structure component (i.e. high-frequency changes in the linear and angular velocity control signal in Fig. 6), for both x_e and y_e, for an overall 5 cm maximum deviation. The angular error ϕ_e is also kept low,

$$\dot{v}_c = \frac{-Q_1 s_1 - P_1 \text{sgn}(s_1) - \lambda_1 \dot{x}_e - \dot{\omega}_d y_e - \omega_d \dot{y}_e + v_r \dot{\phi}_e \sin(\phi_e) + \dot{v}_d}{\cos(\phi_e)}, \quad (14)$$

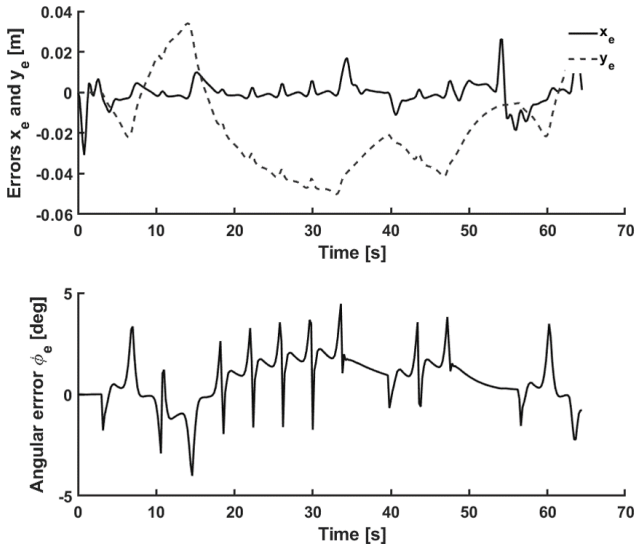

FIGURE 8. Trajectory tracking ANTIFRAGILE control analysis on the kinematic performance: the longitudinal error x_e, the lateral error y_e, and the heading error ϕ_e.

with a maximum deviation of 4 degrees, despite the highly curved trajectory in Fig. 7, which is only attained due to the time scale separation and redundant overcompensation components of the ANTIFRAGILE control.

B. FAULTS AND UNCERTAINTY INJECTION SYSTEM

The fault and uncertainty injection system is based on the work of [22]. Essentially, an estimate of the robot state in the presence of uncertainty and faults is produced by encapsulating changing robot dynamics according to the type of uncertainty and fault. This is achieved by modelling uncertainty and faults through a set of Kalman filters. Stated otherwise, we utilize the Kalman filter's capacity to minimize the estimated error covariance under specific conditions, thereby implementing an optimal estimator using a series of prediction-correction equations. More specifically, in our tests, every Kalman filter captures a robot's kinematic model, but with a distinct set of parameters (i.e., representing different types of defects or uncertainties).

The core idea behind this framework resides in the fact that for the same input vector (with noise), each filter computes a prediction of the robot's state vector. Each filter is associated with a certain form of uncertainty or fault. Thus, we take into account 1) a nominal filter that corresponds to the fault-free robot operation; 2) a filter that has the same robot kinematic model but modified parameters to emphasize the right tire flat fault (i.e., the right wheel radius has a smaller value progressively); 3) a filter that models and predicts the dynamics of a left tire flat fault; 4) a filter that models and predicts an actuator shaft bending of the right wheel; and 5) a filter that models and predicts an actuator shaft bending of the left driving wheel of the robot, respectively. Besides the state estimate each filter generates a measurement vector estimate during the prediction stage, which is used in the correction stage of the filter.

TABLE 1. Quantitative performance evaluation for the different controllers. We report the RMSE-based ranking of minimal robot pose deviations with rank as the ordered average RMSE in faulty and fault-free operation. A lower-ranking order is better.

Control System	Fault type					
	Fault-free	Bump left	Bump right	Flat left	Flat right	Rank
			x_e RMSE			
ROBUST	0.0156	1.3897	1.7950	0.1430	0.1434	3
ADAPTIVE	0.0036	1.3777	1.7830	0.1310	0.1314	2
RESILIENT	0.6948	5.5747	5.1694	6.8214	6.8219	4
ANTIFRAGILE	**0.0025**	**1.3766**	**1.7819**	**0.1299**	**0.1300**	**1**
			y_e RMSE			
ROBUST	0.0005	0.3538	1.2162	0.1689	0.6052	2
ADAPTIVE	0.0007	0.3540	1.2159	0.1691	0.6053	3
RESILIENT	0.0316	0.3848	1.2851	0.1999	0.6362	4
ANTIFRAGILE	**0.0002**	**0.3529**	**1.2170**	**0.1681**	**0.6043**	**1**
			ϕ_e RMSE			
ROBUST	**0.05521**	**0.2573**	**0.3441**	**0.2332**	**0.8642**	**1**
ADAPTIVE	0.0917	0.7628	0.0807	0.6785	1.3898	4
RESILIENT	0.5569	0.2976	0.3844	0.2133	0.8945	2
ANTIFRAGILE	0.0707	0.1838	0.1016	0.6195	1.3807	3

C. EVALUATION

For the evaluation of the four approaches for robot trajectory tracking control, we parametrize the faults and uncertainty injection system, described above, for 4 types of faults of the closed-loop system, namely: 2 sensor faults (e.g., perceiving a continuous wheel radius decrease during operation akin to a flat tire) and 2 actuator faults (e.g., a periodic eccentric mechanical motion of the DC motor shaft akin to a wheel bump). The faults are amplitude and time parametrized, basically assuming a progressive change in wheel radius over some time or a fixed amplitude increment of the wheel radius at periodic time intervals. Such parametrization is possible through the Kalman filter bank approach developed by [22]. More precisely, the faults parametrization, for the trajectory in Fig. 7 and analysis in Fig. 9, is as follows: the DC motor shaft bump amplitude is 1.5 cm and occurs periodically starting at time $t_{injection} = 20$ s of trajectory tracking (i.e., after the first half-loop in panels a, b, c, d); the flat tire assumes a time decaying wheel radius decrease from $r_{fault\_free} = 30$ cm to $r_{flat\_tire} = 26$ cm starting at $t_{injection} = 20$ s.

Intuitively, the effect each type of fault has upon the closed-loop system trajectory tracking control is different and dictated by the effects it has upon the kinematics and the dynamics of the robot. We have analyzed the effects of each of the 4 types of faults, in Fig. 9 when considering effects on a simple baseline control scheme (i.e., without any adaptation). This choice is motivated by the fact that we want to understand how each of the faults reflects in the robot's behaviour without any means to adapt.

As we can see, ANTIFRAGILE control dominates the position control of the robot in the presence of uncertainties, followed closely by the ROBUST and ADAPTIVE control strategies, and lastly by RESILIENT control. Interestingly, ANTIFRAGILE control places only third when it comes to controlling the heading of the robot, where ROBUST control and RESILIENT control excel due to explicit decoupling of the heading from the Cartesian positioning in the control law design. There are also more subtle aspects that we will unfold in the following section.

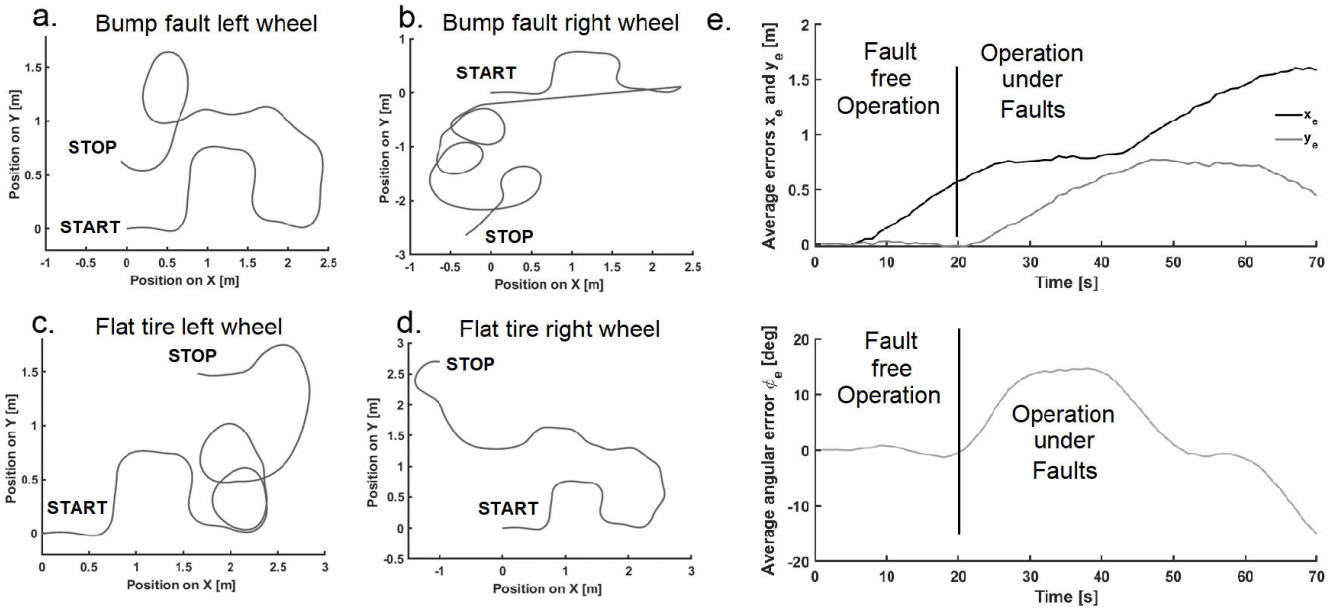

FIGURE 9. Qualitative trajectory tracking analysis in the presence of faults. Uncertainty is quantified by sensor and actuator faults amplitude. Volatility is quantified by fault onset and duration. a) Robot trajectory exposed to a DC motor shaft bump actuator fault on left robot wheel; b) Robot trajectory exposed to a DC motor shaft bump actuator fault on right robot wheel; c) Robot trajectory exposed to a flat tire sensor fault on left robot wheel; d) Robot trajectory exposed to a flat tire sensor fault on right robot wheel; e) The average longitudinal error x_e, lateral error y_e, and heading error ϕ_e over the 4 types of faults. The faults parametrization is as follows: the DC motor shaft bump amplitude is 1.5 cm and occurs periodically starting at time $t_{injection} = 20$ s of trajectory tracking (i.e. after the first half-loop in panels a, b, c, d); the flat tire assumes a time decaying wheel radius decreases from $r_o = 30$ cm to $r_f = 26$ cm starting at $t_{injection} = 20$ s.

The final experiment we performed in this study comes back to the fragility-robustness-antifragility spectrum in Fig. 1. More specifically, we aimed to assess the effectiveness of the four control mechanisms on cascaded failures, which can arise with varying magnitudes and at random times during the robot's operation. A special scenario was created in which we subsequently injected: actuator faults (such as a periodic eccentric mechanical motion of the DC motor shaft akin to a wheel bump), sensor faults (such as perceiving a continuous wheel radius decrease during operation akin to a flat tire), and changes in environment parameters (such as wheel slippage). We then compared the four strategies in terms of the tracking performance (i.e., RMSE on x_e, y_e, and ϕ_e) under the effect of the faults. Our findings are depicted in Fig. 10, where we can see the superior performance of the ANTIFRAGILE and ROBUST controllers which overcome the ADAPTIVE control and RESILIENT control. Interestingly, the diagram still captures the layout of the fragility-robustness-antifragility spectrum, where RESILIENT control has the loosest reaction (i.e., slow) to the occurrence of faults, but demonstrates in between a good stationary behaviour (i.e., see the blue trace in Fig. 10 between 25 s and 33 s where the position error doesn't increase, hence the fault was accommodated by the controller. In the case of ANTIFRAGILE and ROBUST, we can detect high oscillations due to the variable structure control law which keeps the position error as low as possible with the price of a high control activity. The ADAPTIVE controller manages to provide stable performance under the cascaded faults with comparable performance with the ANTIFRAGILE and ROBUST strategies.

IV. DISCUSSION

Trajectory tracking is a fundamental problem in mobile robot control and an even more fundamental control issue when considering uncertainty and sensor and actuator faults. Control strategies designed for tackling this problem need to synthesize control laws for the robot's actuators which compensate for longitudinal x_e, lateral y_e, and heading angle ϕ_e errors under uncertainty in both the operating environment and robot's sensors and actuators reliability. The landscape of solutions for such a multi-faceted problem is very diverse and rich in the methodological approaches used therein. Nonlinear control is the dominating approach for trajectory tracking, being used in mobile robots, such as the work of [31] and [32], but also in underwater systems, such as the work of [34], or flying robotics, such as the work of [35]. The aforementioned approaches, start from a basic nonlinear control formulation and extend the control synthesis with either discrete event triggering of state space changes [34], Euclidean distance damping–based adaptation [36], or explicit physics priors [37]. In our study, we introduce a novel control strategy termed ANTIFRAGILE control, which has the benefit of gaining from continuous exposure to uncertainty and reaching performance that is beyond robust. We have validated our hypothesis through a batch of experiments, an extensive evaluation, and the design of a framework for the evaluation of fault and uncertainty

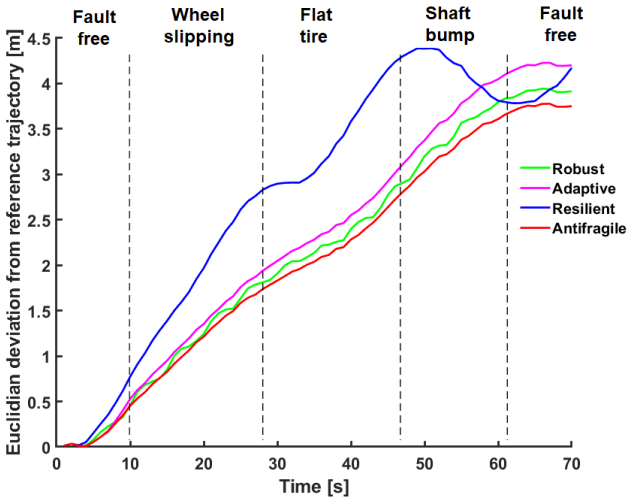

FIGURE 10. Trajectory tracking performance analysis in the presence of cascaded faults. After starting the operation with no faults (i.e. fault-free region) the faults and uncertainty injection system introduces one after the other the three types of faults at different times ($t_{slipping} = 10$ s, $t_{flat} = 28$ s, $t_{bump} = 46$ s, and back to fault-free from $t_{free} = 62$ s onward).

injection in mobile robot trajectory tracking control. The competitive algorithms were chosen among typical and relevant (i.e. see Figure 1) approaches for trajectory tracking, namely ROBUST control (i.e., a sliding mode controller), ADAPTIVE control (i.e., MPC), and RESILIENT control (i.e., fuzzy logic controller). The experiments and evaluation were designed to capture the comparative performance degradation of the closed-loop controllers in the presence of sensor and actuator faults.

A. TIME SCALE SEPARATION

A strong component in the ANTFRAGILE control synthesis is the time scale separation component responsible for the low-level actuator control robustness. Using singular perturbation theory, we have implemented time scale separation within the ANTFRAGILE controller based on the analysis of the dynamics boundary layers shapes (see Fig. 4) and the shape of the prescribed path and curvature quantities given as a reference for the actuators. Interestingly, we could obtain a separation of the control regimes in antifragile and fragile based solely on the curvature of the uniform solution shape of the closed-loop system, given by 5. This separation is then exploited in the computation of the actuator control signal (i.e., $v(t)$ and $w(t)$) which takes either the form of a ''rapid'' or ''slow'' dynamics. A similar behaviour, or at least comparable, is achieved in the ROBUST design through the variable structure control. More precisely, the lower-order design of the controller using equivalent control (see 19) accounts for a reduced-order technique analogous to the effect singular perturbation offers. The ADAPTIVE control attempts a time scale separation through the choice of a multi-scale cost function with regularizing terms for each temporal scale magnitude (see 20). This is especially visible in the MPC instantiation we considered where the prediction horizon can be weighted separately on ''fast'' and ''slow'' dynamics. Finally, the RESILIENT control can induce, in principle, such time-scale separation explicitly. In our case, this can be achieved through fuzzy inference rules that capture the co-variance of the first derivative of error terms and their rate of change in (21). Time scale separation is a design component that determines the low-level actuator control and the benefits of a curvature-aware synthesis of control law (see Fig. 4).

B. REDUNDANT OVER COMPENSATION

Redundant over-compensation refers to the capacity of the controller to build capacity to compensate (promptly) for uncertainty and faults. This ''capacity'' building can be seen as a measure of compensation, which goes beyond accommodating the uncertain event and up to gaining (i.e. sudden convergence of error) from the unexpected event. ANTIFRAGILE control uses redundant over-compensation when designing the low-level control of the actuators (see Fig. 5). After identifying the ''fast'' and the ''slow'' dynamics of the closed-loop system of the actuators, the design focuses on rewriting the dynamics such that the closed-loop system dynamics are described solely by the solution in the overlap region in Fig. 4, where convexity of the response can be probed through (5). To analyze the redundant over-compensation behaviour of the competitive control strategies, we start with a thorough overview of the experimental results in Fig. 7. Here, the differences in compensating for the curvature (i.e. second-order effects) of the prescribed dynamics (i.e. the trajectory is a place and a dynamics) are visible. More precisely, in Fig. 7 b one can observe that the ANTIFRAGILE and ROBUST controllers follow the prescribed trajectory closely (see Table 1 for quantitative assessment), with small magnitude overshooting in high-curvature regions. On the other side, the ADAPTIVE and RESILIENT controllers capture the overall inflexions of the trajectory but fail to smoothly capture highly convex regions and the end position. Finally, to get a more intuitive understanding of the benefits of redundant over-compensation, we analyze the results in Fig. 6. Here, the kinematic (i.e., linear and angular velocities) and the dynamic (i.e. linear and angular acceleration) quantities describing the robot's motion are analyzed, for the reference, control, and real velocities and accelerations. A trademark of ANTIFRAGILE control is the fact that the control linear velocity signal overshoots at regions where the curvature sign changes (see Fig. 6a) on the trajectory, visible also in the rate of change of velocity, depicted in Fig. 6b. These high-frequency changes are also determined by the variable structure control synthesis at the core of the ANTIFRAGILE design. This ''capacity'' building is also visible in the angular control signals, where both angular velocity control signals, depicted in Fig. 6c, and their rate of change surpasses shortly the prescribed values at the high-curvature inflexions of the trajectory. This behaviour is motivated by the simplified

dynamics of the DC motor which describe a proportional effect to changes in the dynamics of the "fast variables."

C. VARIABLE STRUCTURE CONTROL

The final ingredient of the ANTIFRAGILE control design is the variable structure control synthesis. This approach is highly used in the realm of robust control design as a means to inherently handle uncertainty, be it structured (i.e. parametric uncertainty) or unstructured (i.e. unmodelled dynamics). This is also the common design component between the ROBUST and ANTIFRAGILE controllers. Such a control pushes the system to a manifold that describes the prescribed dynamics of the closed-loop system and ensures that the system stays there. As mentioned earlier, the manifold becomes a place and a dynamics, as depicted in Fig. 5. Intuitively, the control signal to generate will be discontinuous in nature and stability is a strong prerequisite (see (14)–(17)). The induced behaviour of the variable structure control in both the ROBUST and ANTIFRAGILE controllers is visible in Fig. 6. This is even more clear when analyzing the performance in Table 1. Here, one can observe that the ANTIFRAGILE and ROBUST controllers outperform the ADAPTIVE and RESILIENT controllers in giving the lowest RMSE on longitudinal and lateral deviations when just the fault-free (baseline) scenario is taken into account. Due to ANTIFRAGILE's implicit weighting of the heading in the manifold design, the dominance is altered in the heading error, where it only ranks three (please refer to (8)). This is further emphasized in Fig. 8 and motivated by the fact that in trajectory tracking heading is secondary. In contrast, the overall (Euclidean) position needs to match as well as possible the prescribed trajectory. Due to the underlying model predictive control, the ADAPTIVE control does a comparatively good job across deviations RMSE in the fault-free scenarios, even better than the RESILIENT control which excels in the heading error minimization. When considering the scenarios with uncertainty and faults, we considered a performance evaluation for the different trajectory control algorithms under the impact of 4 types of faults (i.e., 2 sensor faults— modelled as flat tires—and 2 actuator faults— modelled as motor shaft periodic bump— of robot's driving wheels), as shown in Table 1. Overall, the ANTIFRAGILE control outperforms the other control techniques with minimum RMSE across all fault types, albeit with a rather narrow margin from ROBUST. In close quarters, the ROBUST control outperforms ANTIFRAGILE and the other techniques in minimizing orientation errors. ADAPTIVE control approaches ROBUST with a slight penalty that may depend on the cost function selection. Lastly, a slower changing response is offered via RESILIENT control (akin to the hypothetical situation in Fig. 1) but with a stable outcome. Finally, in our last and most extreme example, cascaded faults and uncertainty in the robot's trajectory tracking operation (see Fig. 10) were considered. The overall evaluation criteria were chosen based on the Euclidean deviation from the prescribed trajectory. As one can see, and also supported by the previous discussion and analysis, the experiments bring us closer to validating the hypothesis (visually described in Fig. 1). The analysis in Fig. 10 shows that ANTIFRAGILE control (red trace) offers the smallest deviation, with small regions (typically before a new fault occurs) where the errors decrease even more. Closely behind is the robust behaviour of the ROBUST controller, which displays a high-frequency oscillatory control law causing oscillations in the actuator's commands and eventually in the trajectory due to its variable structure control law (see the green trace in Fig. 10). By taking advantage of the underlying model's predictive properties and receding horizon, ADAPTIVE maximizes the benefits of MPC while delivering high performance. Ultimately, each inserted defect causes RESILIENT to respond slowly, increasing the total position inaccuracy.

V. CONCLUSION

Handling uncertainty in closed-loop robot control is still an openly debated and fruitful area of research. To tackle control synthesis in this setting, we present ANTIFRAGILE control, a novel design framework that aims to capture the unique characteristics of the system's response to control. In regions of the solutions space where the system is not only resistant to uncertainty and volatility but also capable of profiting from it and anticipating future uncertain events, first- and second-order effects offer useful guidance on when and how to issue control signals that can drive the system. This is the main reason for the current investigation and ANTIFRAGILE control. We were able to validate both the controller design for a robotic task and the ANTIFRAGILE control framework through the tests with parametrizable faults. Developing a coherent theory and framework based on the idea of induced antifragility in technical systems starts with this.

ACKNOWLEDGMENT

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or The European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] P. Halder and M. Althoff, "Minimum-violation velocity planning with temporal logic constraints," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 2520–2527.

[2] L. Chong and C. Osorio, "A simulation-based optimization algorithm for dynamic large-scale urban transportation problems," *Transp. Sci.*, vol. 52, no. 3, pp. 637–656, Jun. 2018.

[3] T. Carlson and Y. Demiris, "Collaborative control for a robotic wheelchair: Evaluation of performance, attention, and workload," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 42, no. 3, pp. 876–888, Jun. 2012.

[4] N. N. Taleb, *Antifragile: Things That Gain from Disorder*, vol. 3. New York, NY, USA: Random House, 2012.

[5] N. N. Taleb and R. Douady, "Mathematical definition, mapping, and detection of (anti) fragility," *Quant. Finance*, vol. 13, no. 11, pp. 1677–1689, 2013.

[6] C. Axenie, D. Kurz, and M. Saveriano, "Antifragile control systems: The case of an anti-symmetric network model of the tumor-immune-drug interactions," *Symmetry*, vol. 14, no. 10, p. 2034, Sep. 2022.

[7] C. Axenie and M. Grossi, "Antifragile control systems: The case of an oscillator-based network model of urban road traffic dynamics," 2022, *arXiv:2210.10460*.
[8] R. C. Solea, "Sliding mode control applied in trajectory-tracking of WMRs and autonomous vehicles," Ph.D. dissertation, Inst. Syst. Robot., Universidade Coimbra, Porto, Portugal, 2009.
[9] C. Axenie and R. Solea, "Real time control design for mobile robot fault tolerant control. Introducing the ARTEMIC powered mobile robot," in *Proc. IEEE/ASME Int. Conf. Mech. Embedded Syst. Appl.*, Jul. 2010, pp. 7–13.
[10] R. Fierro and F. L. Lewis, "Control of a nonholomic mobile robot: Backstepping kinematics into dynamics," *J. Robotic Syst.*, vol. 14, no. 3, pp. 149–163, Mar. 1997.
[11] A. M. Bloch, "An introduction to aspects of geometric control theory," in *Nonholonomic Mechanics and Control*. New York, NY, USA: Springer, 2015, pp. 199–233.
[12] A. D. Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and Dynamics of Multi-Body Systems*. Vienna, Austria: Springer, 1995, pp. 277–342.
[13] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, Nov. 2002.
[14] Y. Zhang, J. H. Chung, and S. A. Velinsky, "Variable structure control of a differentially steered wheeled mobile robot," *J. Intell. Robotic Syst.*, vol. 36, no. 3, pp. 301–314, Mar. 2003.
[15] R. Solea and U. Nunes, "Trajectory planning and sliding-mode control based trajectory-tracking for cybercars," *Integr. Comput.-Aided Eng.*, vol. 14, no. 1, pp. 33–47, Jan. 2007.
[16] Z. P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica*, vol. 33, no. 7, pp. 1393–1399, Jul. 1997.
[17] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in Cartesian space," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jan. 1991, pp. 1136–1137.
[18] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, pp. 384–389.
[19] P. Braun, C. M. Kellett, and L. Zaccarian, "Explicit construction of stabilizing robust avoidance controllers for linear systems with drift," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 595–610, Feb. 2021.
[20] W. Dong and K. D. Kuhnert, "Robust adaptive control of nonholonomic mobile robot with parameter and nonparameter uncertainties," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 261–266, Apr. 2005.
[21] F. Pourboghrat and M. P. Karlsson, "Adaptive control of dynamic mobile robots with nonholonomic constraints," *Comput. Electr. Eng.*, vol. 28, no. 4, pp. 241–253, Jul. 2002.
[22] C. Axenie and D. Cernega, "Adaptive sliding mode controller design for mobile robot fault tolerant control. Introducing ARTEMIC," in *Proc. 19th Int. Workshop Robot. Alpe-Adria-Danube Region (RAAD)*, Jun. 2010, pp. 253–259.
[23] F. Kuhne, W. F. Lages, and J. G. da Silva, "Model predictive control of a mobile robot using linearization," in *Proc. Mechatronics Robot.*, 2004, pp. 525–530.
[24] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017.
[25] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved MPC," *IEEE Access*, vol. 7, pp. 161064–161073, 2019.
[26] G. Antonelli, S. Chiaverini, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 2, pp. 211–221, Apr. 2007.
[27] R. Solea and U. Nunes, "Trajectory planning with velocity planner for fully-automated passenger vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 474–480.
[28] C. K. R. T. Jones, "Geometric singular perturbation theory," in *Dynamical Systems* (Lecture Notes in Mathematics), vol. 1609, R. Johnson, Ed. Berlin, Germany: Springer, 1995.
[29] V. I. Utkin, "Sliding mode control: Mathematical tools, design and applications," in *Nonlinear and Optimal Control Theory*. Berlin, Germany: Springer, 2008, pp. 289–347.
[30] W. Gao and J. C. Hung, "Variable structure control of nonlinear systems: A new approach," *IEEE Trans. Ind. Electron.*, vol. 40, no. 1, pp. 45–55, Feb. 1993.
[31] Y. Peng, P. Zhang, Z. Fang, S. Zheng, Z. Guo, and D. Xing, "Trajectory tracking control of the wheeled mobile robot based on the curve tracking algorithm," *J. Phys., Conf. Ser.*, vol. 2419, no. 1, Jan. 2023, Art. no. 012106.
[32] F. Xie, G. Liang, and Y.-R. Chien, "Highly robust adaptive sliding mode trajectory tracking control of autonomous vehicles," *Sensors*, vol. 23, no. 7, p. 3454, Mar. 2023.
[33] S. Drücker and R. Seifried, "Trajectory-tracking control from a multibody system dynamics perspective," *Multibody Syst. Dyn.*, vol. 58, nos. 3–4, pp. 341–363, Aug. 2023.
[34] J. Qin, J. Du, and J. Li, "Adaptive finite-time trajectory tracking event-triggered control scheme for underactuated surface vessels subject to input saturation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8809–8819, Aug. 2023.
[35] J. Baek and M. Kang, "A synthesized sliding-mode control for attitude trajectory tracking of quadrotor UAV systems," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 4, pp. 2189–2199, Aug. 2023.
[36] Q. Chen, Q. Zhang, Y. Hu, Y. Liu, and H. Wu, "Euclidean distance damping-based adaptive sliding mode fault-tolerant event-triggered trajectory-tracking control," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 237, no. 3, pp. 551–567, Mar. 2023.
[37] J. Tian and M. Yang, "Research on trajectory tracking and body attitude control of autonomous ground vehicle based on differential steering," *PLoS One*, vol. 18, no. 2, Feb. 2023, Art. no. e0273255.

CRISTIAN AXENIE (Senior Member, IEEE) received the B.Sc. degree in control systems engineering and the M.Sc. degree in advanced control engineering from the University of Galati, Romania, in 2009 and 2011, respectively, and the Dr.Eng.Sc. degree in neurorobotics from the Technical University of Munich, in 2016. He is currently a Professor with the Department of Computer Science, Technische Hochschule Nürnberg Georg Simon Ohm, Germany. Previously, he was a Staff Research Engineer with the Huawei Munich Research Center, Germany, and a Principal Scientist with the Audi Konfuzius-Institut Ingolstadt Laboratory, Germany. His research interests include physics-informed machine learning, multisensory fusion, and neuromorphic engineering. For more information, see https://www.th-nuernberg.de/en/person/axenie-cristian/.

MATTEO SAVERIANO (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in automatic control engineering from the University of Naples Federico II, Italy, in 2008 and 2011, respectively, and the Ph.D. degree from the Technical University of Munich, in 2017. He is currently an Assistant Professor with the Department of Industrial Engineering (DII), University of Trento, Italy. Previously, he was an Assistant Professor with the University of Innsbruck and a Postdoctoral Researcher with the German Aerospace Center (DLR). His research interests include robot learning, human-robot interaction, and understanding and interpreting human activities. He is an Associate Editor of IEEE Robotics and Automation Letters and *International Journal of Robotics Research*. For more information, see https://matteosaveriano.weebly.com/.