

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.1120000

# Model Predictive Contouring Control with Barrier and Lyapunov Functions for Stable Path-Following in UAV systems

Bryan S. Guevara<sup>1</sup>, Viviana Moya<sup>2</sup>, Luis F. Recalde<sup>1</sup>, David Pozo-Espin<sup>3</sup>, Daniel C. Gandolfo<sup>1</sup> and Juan M. Toibero<sup>1</sup>

<sup>1</sup>Instituto de Automática, Universidad Nacional de San Juan - CONICET, Av. San Martín Oeste 1109, San Juan J5400ARL, Argentina (e-mail: bguevara@inaut.unsj.edu.ar; lrecalde@inaut.unsj.edu.ar; dgandolfo@inaut.unsj.edu.ar; mtoibero@inaut.unsj.edu.ar)

<sup>2</sup>Facultad de Ciencias Técnicas, Universidad Internacional del Ecuador, Quito, Ecuador (e-mail: vimoyago@uide.edu.ec)

<sup>3</sup>Ingeniería en Electrónica y Automatización, Facultad de Ingeniería y Ciencias Aplicadas, Universidad de las Américas, Quito 170513, Ecuador (e-mails: david.pozo@udla.edu.ec)

Corresponding author: David Pozo-Espin (e-mail: david.pozo@udla.edu.ec).

The research was funded by the Universidad de las Américas (UDLA), Universidad Internacional del Ecuador (UIDE), and the Instituto de Automática (INAUT) of the Universidad Nacional de San Juan (UNSJ) - CONICET.

**ABSTRACT** In this study, we propose a novel method that integrates Nonlinear Model Predictive Contour Control (NMPCC) with an Exponentially Stabilizing Control Lyapunov Function (ES-CLF) and Exponential Higher-Order Control Barrier Functions to achieve stable path-following and obstacle avoidance in UAV systems. This framework enables unmanned aerial vehicles (UAVs) to safely navigate around both static and dynamic obstacles while strictly adhering to desired paths. The quaternion-based formulation ensures precise orientation and attitude control, while a robust optimization solver enforces the constraints imposed by the Control Lyapunov Function (CLF) and Control Barrier Functions (CBF), ensuring reliable real-time performance. The method was validated in a Model-in-the-Loop (MiL) environment, demonstrating effective path tracking and obstacle avoidance. The results highlight the framework's ability to minimize both orthogonal and tangential errors, ensuring stability and safety in complex environments.

**INDEX TERMS** MPCC, CLF, CBF, UAV, obstacle avoidance, path-following, CasADi, Acados.

## I. INTRODUCTION

THE increasing complexity of modern aerial systems has driven substantial advancements in control methodologies capable of managing the non-linear dynamics and safety-critical constraints of Unmanned Aerial Vehicles (UAVs) [1]. UAVs are required to operate in dynamic and uncertain environments where they must not only follow predefined paths but also respond adaptively to the presence of obstacles and interactions with other vehicles [2]. These scenarios place significant demands on control systems, which require simultaneous optimization of trajectory accuracy, safety, and real-time adaptability [3].

NMPCC has emerged as a promising control framework to address these challenges [4]–[6]. Unlike traditional non-linear model predictive control (NMPC), which focuses on minimizing time-indexed tracking errors, NMPCC emphasizes minimizing contouring and lag errors relative to a desired path [7]. This path-following optimization provides greater flexibility for UAVs to adapt their trajectories in

response to environmental changes, such as the sudden appearance of obstacles, while still maintaining overall mission objectives [8]. This adaptability makes NMPCC particularly well suited for complex tasks that require real-time responsiveness and high levels of autonomy [9].

The integration of NMPCC in UAV systems introduces additional challenges when operating in complex airspaces, requiring avoidance of both static and dynamic obstacles [10]. Ensuring safe, collision-free navigation under these conditions requires advanced control strategies that can dynamically adjust trajectories while maintaining the required stability and safety standards [11]. Traditional control methods, though effective in simpler environments, often lack the robustness and flexibility to handle the complexity of multi-UAV systems in real-world applications [12]–[14].

To address these limitations, we propose integrating NMPCC with CLF and CBF frameworks. CLFs offer a systematic numerical approach to ensuring system stability by enforcing the decrease of a Lyapunov function over time [15]–[18]. On

the other hand, CBFs enforce safety constraints, ensuring that UAVs operate within safe boundaries and avoid collisions [19], [20]. The combination of these control frameworks within the NMPCC structure results in a comprehensive approach that addresses both performance and safety, enabling UAVs to navigate complex and dynamic environments autonomously.

In addition to path-following and safety considerations, three-dimensional orientation is crucial for UAV control, particularly in dynamic environments where precise maneuvers are required [21]. To address this, we incorporate quaternion-based formulations into the control design. Quaternions offer several advantages over traditional Euler angles, such as avoiding gimbal lock and ensuring smooth, continuous rotation [22]. By leveraging the mathematical properties of quaternions, particularly their ability to map to the tangent space using the logarithmic map (Log) [23]–[25], we can effectively manage the rotational dynamics of UAVs, enhancing the robustness and accuracy of the overall control system. This quaternion-based approach is rooted in Lie theory and manifold principles, providing a mathematically rigorous foundation for handling the rotational behavior of UAVs.

Implementing this integrated framework, which combines NMPCC, CLFs, CBFs, and quaternion-based formulations, introduces considerable computational challenges, especially for real-time applications. To mitigate these challenges, we utilize ACADOS [26], an open-source software package optimized for solving optimal control problems with high computational efficiency, alongside CasADi [27], a symbolic framework for automatic differentiation and numerical optimization. These frameworks are particularly well-suited for UAV control tasks that demand rapid and precise decision-making, offering flexibility that enables seamless integration with the proposed control strategy. For the numerical integration of rotational dynamics, a fourth-order Runge-Kutta method is applied to ensure precise state updates.

In this study, we validate the proposed control strategy through extensive Model-in-the-Loop (MiL) simulations. These simulations demonstrate the practicality and robustness of the approach in complex environments. The results highlight the benefits of integrating NMPCC with advanced control techniques, offering a unified solution for safe and reliable UAV navigation in challenging operational settings.

In summary, the contributions of this paper are threefold:

- We present a novel integration of NMPCC with Control ES-CLF and higher-order CBF, providing a unified approach to dynamic obstacle avoidance in environments with multiple obstacles.
- We demonstrate the effectiveness of quaternion-based formulations, using the Log operator to map to the tangent space according to Lie theory and manifolds, improving the robustness and precision of UAV attitude control by ensuring smooth transitions and accurate orientation representation.

## A. OUTLINE

This paper is structured as follows. Section II introduces the dynamic model of the UAV system, providing the foundation for the control strategy. Section III presents the CLF used to ensure stability in the proposed framework. Section IV focuses on the CBF, which enforces safety constraints in the system. Section V explores the NMPCC approach, detailing its application to path-following and obstacle avoidance. Section VI provides an in-depth analysis of the experiments and results, demonstrating the performance of the proposed control strategy. Finally, Section VII concludes the paper, summarizing the key findings and outlining potential directions for future research.

## II. KINODYNAMIC OF UAV

We base the kinodynamic model of the UAV described in [28], where the state vector of the UAV is defined as  $\mathbf{x} = [\mathbf{p} \ \mathbf{v} \ \mathbf{q} \ \boldsymbol{\omega}]^T$ , where  $\mathbf{p} = [x, y, z]^T$  represents the UAV's position in the global frame, and  $\mathbf{v} = [v_x, v_y, v_z]^T$  is the velocity vector also in the global frame. The orientation of the UAV is represented by the quaternion  $\mathbf{q} = [q_w \ \mathbf{q}_v^T]^T$ , which defines the UAV's attitude. Finally, the angular velocity vector in the body frame is denoted by  $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ .

### A. TRANSLATIONAL DYNAMICS

The translational dynamics of the UAV is governed by the following equations:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= \mathbf{g} + \frac{1}{m} \mathbf{R} \mathbf{F}, \end{aligned}$$

where  $\mathbf{F} = [0 \ 0 \ F_z]^T$  is the thrust force vector, with  $F_z$  representing the vertical thrust component. The rotation matrix  $\mathbf{R}$  transforms the vectors from the body frame to the global frame, while the gravitational acceleration vector  $\mathbf{g} = [0 \ 0 \ -g]^T$  includes  $g$ , the acceleration due to gravity. The mass of the UAV is denoted by  $m$ .

### B. ROTATIONAL DYNAMICS

The rotational dynamics of the UAV is described by the following equation:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} (\boldsymbol{\tau} - \boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega})),$$

where  $\boldsymbol{\tau} = [\tau_x \ \tau_y \ \tau_z]$  is the moment vector applied to the UAV,  $\mathbf{I}$  is the inertia matrix, and  $\dot{\boldsymbol{\omega}}$  is the angular acceleration vector. The term  $\boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega})$ , known as the gyroscopic torque, accounts for the rotational effects due to angular velocity.

The evolution of the UAV's attitude over time, which represents the instantaneous kinematics, is given by the quaternion derivative with respect to time. In this context,  $\otimes$  denotes quaternion multiplication:

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \boldsymbol{\omega}.$$

This can also be expressed as:

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \begin{bmatrix} 0 & -\boldsymbol{\omega}^\top(t) \\ \boldsymbol{\omega}(t) & 0 \end{bmatrix} \mathbf{q}(t), \quad (1)$$

or equivalently:

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \mathbf{S}(\boldsymbol{\omega}(t)) \mathbf{q}(t), \quad (2)$$

where  $[\boldsymbol{\omega}(t)]_\times$  is a skew-symmetric matrix.

### III. CONTROL LYAPUNOV FUNCTION

Consider the UAV nonlinear dynamical system described by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad (3)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the state vector,  $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  represents the uncontrolled dynamics, and  $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$  is the control distribution matrix, defining how the control inputs  $\mathbf{u} \in \mathbb{R}^m$  affect the system.

A scalar function  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  is called a *Control Lyapunov Function* if it is continuously differentiable, positive definite, and satisfies  $V(\mathbf{x}_e) = 0$ , where  $\mathbf{x}_e$  represents the equilibrium point. This indicates that the system is at equilibrium when  $V(\mathbf{x}) = 0$ , and away from equilibrium when  $V(\mathbf{x}) > 0$  for all  $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{x}_e\}$ .

For  $V(\mathbf{x})$  to be effective as a CLF, its time derivative along the system trajectories under the control law  $\mathbf{u}$  should be negative definite. Mathematically, this requirement is expressed as:

$$\dot{V}(\mathbf{x}) = \nabla V(\mathbf{x}) \cdot \dot{\mathbf{x}} < 0 \quad \text{for all } \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{x}_e\}. \quad (4)$$

This condition implies that the function  $V(\mathbf{x})$  decreases strictly along the trajectories of the system for an appropriate choice of the control input  $\mathbf{u}$ , leading to the asymptotic stability of the system at the equilibrium point  $\mathbf{x}_e$ .

An *Exponentially Stabilizing Control Lyapunov Function* is a specialized form of a CLF that ensures exponential convergence of the state vector  $\mathbf{x}$  to the equilibrium point  $\mathbf{x}_e$ . The function  $V(\mathbf{x})$  is classified as an ES-CLF if it satisfies the following conditions [16]:

- There exist positive constants  $c_1, c_2 > 0$  such that:

$$c_1 \|\mathbf{x} - \mathbf{x}_e\|^2 \leq V(\mathbf{x}) \leq c_2 \|\mathbf{x} - \mathbf{x}_e\|^2, \quad \text{for all } \mathbf{x} \in \mathbb{R}^n. \quad (5)$$

- There exists a positive constant  $c_3 > 0$  and a continuous control law  $\mathbf{u} = \mathbf{k}(\mathbf{x})$  that ensures the following condition holds:

$$\dot{V}(\mathbf{x}) \leq -c_3 V(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{x}_e. \quad (6)$$

Integrating this inequality with respect to time yields:

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}(0))e^{-c_3 t}. \quad (7)$$

Given the quadratic bounds on  $V(\mathbf{x})$ , we can further deduce:

$$c_1 \|\mathbf{x}(t) - \mathbf{x}_e\|^2 \leq V(\mathbf{x}(t)) \leq V(\mathbf{x}(0))e^{-c_3 t} \leq c_2 \|\mathbf{x}(0) - \mathbf{x}_e\|^2 e^{-c_3 t}. \quad (8)$$

Taking the square root of both sides and rearranging terms, we obtain:

$$\|\mathbf{x}(t) - \mathbf{x}_e\| \leq \sqrt{\frac{c_2}{c_1}} \|\mathbf{x}(0) - \mathbf{x}_e\| e^{-\frac{c_3}{2} t}. \quad (9)$$

This condition implies that  $V(\mathbf{x})$  decays exponentially over time, guaranteeing that the state vector  $\mathbf{x}$  converges to the equilibrium point  $\mathbf{x}_e$  exponentially, with a rate of convergence determined by the constant  $c_3$ .

The use of an ES-CLF as a Lyapunov function in control design carries significant theoretical stability guarantees [29], [30]. This exponential stability implies that the system's state vector  $\mathbf{x}(t)$  converges to the equilibrium point  $\mathbf{x}_e$  at an exponential rate, which is a stronger form of stability compared to mere asymptotic stability.

### IV. CONTROL BARRIER FUNCTIONS

A *Control Barrier Function* is defined for a safe set  $\mathcal{C} \subset \mathbb{R}^n$  as:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}, \quad (10)$$

where  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  is a twice continuously differentiable scalar function.

To ensure that the system state remains within the safe set  $\mathcal{C}$ , we consider the successive Lie derivatives of  $h(\mathbf{x})$  along the system dynamics. The Lie derivative of  $h(\mathbf{x})$  with respect to  $\mathbf{f}(\mathbf{x})$  is defined as:

$$L_f h(\mathbf{x}) = \nabla h(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}), \quad (11)$$

and the Lie derivative of  $h(\mathbf{x})$  with respect to  $\mathbf{g}(\mathbf{x})$  is:

$$L_g h(\mathbf{x}) = \nabla h(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}). \quad (12)$$

Here,  $\nabla h(\mathbf{x})$  is the gradient of the function  $h(\mathbf{x})$  with respect to the state variables  $\mathbf{x}$ , and this gradient is multiplied by  $\mathbf{f}(\mathbf{x})$  and  $\mathbf{g}(\mathbf{x})$ , respectively.

The Lie derivative of order  $k$  of the function  $h(\mathbf{x})$  along the system dynamics is:

$$L_f^k h(\mathbf{x}) = \frac{d^k h(\mathbf{x})}{dt^k} \quad (13)$$

$$= \frac{\partial}{\partial \mathbf{x}} \left( L_f^{k-1} h(\mathbf{x}) \right) \cdot \mathbf{f}(\mathbf{x}), \quad (14)$$

Additionally, the cross Lie derivative of order  $k-1$  involving the function  $g(\mathbf{x})$  is expressed as:

$$L_g L_f^{k-1} h(\mathbf{x}) = \nabla \left( L_f^{k-1} h(\mathbf{x}) \right) \cdot \mathbf{g}(\mathbf{x}). \quad (15)$$

where  $L_f^{k-1} h(\mathbf{x})$  is the  $(k-1)$ -th order Lie derivative of  $h(\mathbf{x})$  with respect to  $\mathbf{f}(\mathbf{x})$ .

To handle more complex safety constraints, we introduce higher-order Control Barrier Functions, which are governed by the following condition:

$$L_f^k h(\mathbf{x}) + L_g L_f^{k-1} h(\mathbf{x})\mathbf{u} + \dots + L_g h(\mathbf{x})\mathbf{u} \geq -\alpha_k(h(\mathbf{x})), \quad (16)$$

where  $\alpha_k : \mathbb{R} \rightarrow \mathbb{R}$  is a class  $\mathcal{K}$  function, meaning it is continuous, strictly increasing, and  $\alpha_k(0) = 0$ .

The complete expression for  $k$  derivatives can be expanded as:

$$\sum_{i=0}^k \binom{k}{i} L_f^{k-i} L_g^i h(\mathbf{x}) \mathbf{u} \geq -\alpha_k(h(\mathbf{x})), \quad (17)$$

where  $\binom{k}{i}$  is the binomial coefficient.

This condition ensures that, under the action of the control input  $\mathbf{u}$ , the system will remain within the safe set  $\mathcal{C}$  defined by the function  $h(\mathbf{x})$ .

For  $h(\mathbf{x})$  to be an *Exponential Control Barrier Functions*, there must exist a gain vector  $\mathbf{K}_\alpha = [K_{\alpha,1}, K_{\alpha,2}, \dots, K_{\alpha,r}] \in \mathbb{R}^r$  such that for the system described by (3), the following condition holds:

$$\sup_{\mathbf{u} \in \mathcal{U}} [L_f^r h(\mathbf{x}) + L_g L_f^{r-1} h(\mathbf{x}) \mathbf{u}] \geq -\mathbf{K}_\alpha \eta_b(\mathbf{x}), \quad (18)$$

where the vector  $\eta_b(\mathbf{x})$  groups the derivatives of  $h(\mathbf{x})$  and its Lie derivatives up to order  $r - 1$ . The vector  $\eta_b(\mathbf{x})$  is constructed as:

$$\eta_b(\mathbf{x}) = \begin{pmatrix} h(\mathbf{x}) \\ L_f h(\mathbf{x}) \\ L_f^2 h(\mathbf{x}) \\ \vdots \\ L_f^{r-1} h(\mathbf{x}) \end{pmatrix}. \quad (19)$$

To compute the second-order Lie derivative of the barrier function, we apply the Lie derivative to  $L_f h(\mathbf{x})$  as follows:

$$L_f^2 h(\mathbf{x}) = \nabla(L_f h(\mathbf{x})) \cdot \mathbf{f}(\mathbf{x}), \quad (20)$$

$$L_g L_f h(\mathbf{x}) = \nabla(L_f h(\mathbf{x})) \cdot \mathbf{g}(\mathbf{x}). \quad (21)$$

This approach allows for the calculation of barrier functions up to the second order, which is essential for implementing constraints of the form discussed:

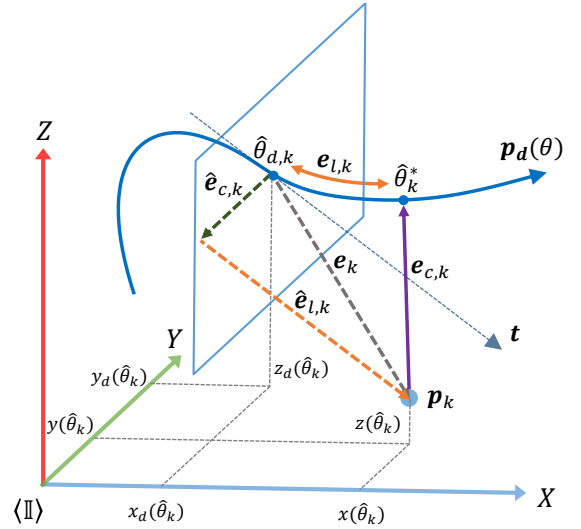
$$\ddot{h}(\mathbf{x}, \mathbf{u}) = L_f^2 h(\mathbf{x}) + L_g L_f h(\mathbf{x}) \mathbf{u}. \quad (22)$$

## V. MODEL PREDICTIVE CONTOURING CONTROL

Unlike traditional NMPC, the formulation of *Nonlinear Model Predictive Contouring Control* is used to minimize trajectory tracking errors while maximizing the speed along a given path over a finite prediction horizon  $l \in [t, t + N]$  [8]. This approach is particularly useful in three-dimensional (3D) environments, where the objective is to reduce both the contour error (orthogonal to the trajectory) and the lag error (parallel to the trajectory), while ensuring efficient progress along the path.

The desired trajectory is defined as a smooth curve parametrized by the arc length  $\theta$ , also referred to as *progress* along the path. The position at any point on the trajectory is given by  $\mathbf{p}_d(\theta) = [x_d(\theta) \ y_d(\theta) \ z_d(\theta)]^\top$ , and its derivative with respect to  $\theta$ , which corresponds to the tangent vector, is:

$$\mathbf{t}(\theta) = \mathbf{p}'_d(\theta) = \frac{d}{d\theta} \mathbf{p}_d(\theta).$$



**FIGURE 1.** The position error is projected onto orthogonal vectors, resulting in approximations of the contour and lag errors.

Since the trajectory is parametrized by arc length, the norm of the tangent vector is always one:

$$\|\mathbf{t}(\theta)\| = 1,$$

At any time  $t_k$ , the point  $\theta_k^*$  represents the location on the desired path closest to the current position  $\mathbf{p}(t_k)$ . However, since  $\theta_k^*$  is typically unknown during optimization, the system approximates it with  $\hat{\theta}_k$ , which serves as the best estimate of  $\theta_k^*$ . The total position error is defined as the difference between the current system position  $\mathbf{p}(t_k)$  and the desired position  $\mathbf{p}_d(\hat{\theta}_k)$  on the trajectory:

$$\mathbf{e}(\hat{\theta}_k) = \mathbf{p}(t_k) - \mathbf{p}_d(\hat{\theta}_k).$$

Although the system position is time-parametrized and the desired position is progress-parametrized, the comparison is valid because  $\hat{\theta}_k$  corresponds to the position on the trajectory at  $t_k$ .

As shown in Fig. 1, the total error can be decomposed into two orthogonal components: the *lag error*,  $\mathbf{e}_l(\hat{\theta}_k)$ , and the *contour error*,  $\mathbf{e}_c(\hat{\theta}_k)$ .

The lag error is defined as the difference in arc length between the current position  $\theta_k$  and the real minimizer  $\theta_k^*$ :

$$e_l(\theta_k^*) = \theta_k - \theta_k^*.$$

It is often approximated by projecting the total position error  $\mathbf{e}(\hat{\theta}_k)$  onto the tangent vector  $\mathbf{t}(\hat{\theta}_k)$ :

$$\mathbf{e}_l(\hat{\theta}_k) = (\mathbf{e}(\hat{\theta}_k) \cdot \mathbf{t}(\hat{\theta}_k)) \mathbf{t}(\hat{\theta}_k),$$

which results in a vector that represents the component of the total error along the direction of motion. Here,  $\mathbf{t}(\hat{\theta}_k) = \mathbf{p}'_d(\hat{\theta}_k)$ .

On the other hand, the contour error quantifies the deviation perpendicular to the trajectory. It is calculated by

subtracting the lag error from the total position error. Alternatively, the contour error can be computed by projecting the position error onto the normal to the tangent vector. This projection is performed using the matrix  $P_{ec}$ , which is defined as:

$$P_{ec} = I - \mathbf{t}(\hat{\theta}_k)\mathbf{t}^\top(\hat{\theta}_k),$$

where  $I$  is the identity matrix. The contour error is then given by:

$$\mathbf{e}_c(\hat{\theta}_k) = P_{ec} \mathbf{e}(\hat{\theta}_k).$$

The contour error can also be understood as the solution to the following minimization problem:

$$e_c(\hat{\theta}_k) = \min_{\theta} \|\mathbf{p}_k - \mathbf{p}_d(\theta)\|.$$

If the projection of the error onto the tangent direction  $\mathbf{e}_l(\hat{\theta}_k)$  is zero, the approximations  $\hat{\theta}_k$ ,  $\mathbf{e}_c(\hat{\theta}_k)$ , and  $\mathbf{e}_l(\hat{\theta}_k)$  coincide with their true values, ensuring accurate tracking.

Finally, the *velocity of progress*  $v_{\hat{\theta},k}$  quantifies the rate at which the system moves along the trajectory. It is calculated by projecting the system's velocity  $\mathbf{v}_k$  onto the tangent vector  $\mathbf{t}(\hat{\theta}_k)$ :

$$v_{\hat{\theta},k} = \mathbf{v}_k \cdot \mathbf{t}(\hat{\theta}_k),$$

which gives the instantaneous speed at which the system progresses along the tangent path.

#### A. OPTIMIZATION PROBLEM FOR NMPCC

In addition to minimizing deviations from the desired path and control effort, the NMPCC also considers maintaining the system's attitude orientation aligned with the direction of motion along the path.

This approach leverages the logarithm map  $\text{Log}(\cdot)$ , which maps the quaternion error from the unit quaternion space  $\mathbb{H}$  onto the tangent space  $\mathbb{R}^3$ . This map is defined as:

$$\text{Log}(\tilde{\mathbf{q}}) = 2\tilde{\mathbf{q}}_w \cdot \frac{\text{atan2}(\|\tilde{\mathbf{q}}_v\|, \tilde{\mathbf{q}}_w)}{\|\tilde{\mathbf{q}}_v\|}, \quad (23)$$

where  $\tilde{\mathbf{q}}_w \in \mathbb{R}$  is the scalar component and  $\tilde{\mathbf{q}}_v \in \mathbb{H}_p$  is the vector part of the quaternion error. This transformation is particularly useful for optimizing orientation errors in Euclidean space. The orientation error between a desired quaternion  $\mathbf{q}_d$  and the actual quaternion  $\mathbf{q}$  is computed as:

$$\tilde{\mathbf{q}} = \mathbf{q}_d \otimes \mathbf{q}^{-1}, \quad (24)$$

where  $\otimes$  denotes quaternion multiplication. The inverse of a quaternion can be calculated using its conjugate, denoted as  $\bar{\mathbf{q}}$ , along with its norm. Specifically, the inverse is given by:

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|^2}, \quad (25)$$

The quaternion manifold  $\mathcal{S}^3$ , representing unit quaternions, is a double cover of the special orthogonal group  $SO(3)$ , meaning that both  $\mathbf{q}$  and  $-\mathbf{q}$  describe the same rotation [31]. To avoid ambiguity arising from this property, it is essential to ensure that the real part  $q_w$  is positive; if  $q_w$  is negative, the quaternion should be negated before proceeding with any

computations. The optimization problem is then formulated as:

$$\begin{aligned} \min_{\mathbf{u}_k, \tilde{\mathbf{q}}_k} \quad & \sum_{k=0}^N \|\mathbf{e}_c(\hat{\theta}_k)\|_{\mathbf{Q}_c}^2 + \|\mathbf{e}_l(\hat{\theta}_k)\|_{\mathbf{Q}_l}^2 - \mu v_{\hat{\theta},k}^2 + \\ & \|\text{Log}(\tilde{\mathbf{q}}_k)\|_{\mathbf{Q}_q}^2 + \frac{1}{2} \|\mathbf{u}_k\|_{\mathbf{R}}^2 + \rho \zeta^2 \\ \text{subject to:} \quad & \mathbf{x}_{k+1} = f(\mathbf{x}_k) + g(\mathbf{x}_k)\mathbf{u}_k, \\ & \mathbf{x}_0 = \mathbf{x}(0), \\ & \mathbf{u}_k \in \mathbb{U}, \\ & \mathbf{x}_k \in \mathbb{X}, \\ & 0 \leq v_{\theta_k} \leq v_{\theta_{\max}}, \\ & \dot{V}(\mathbf{x}_k) \leq -\gamma V(\mathbf{x}_k) + \zeta, \\ & \ddot{h}(\mathbf{x}_k, \boldsymbol{\mu}_k) \geq -\mathbf{K}_\alpha \eta_b(\mathbf{x}_k). \end{aligned} \quad (26)$$

where the gains  $\mathbf{Q}_c$  and  $\mathbf{Q}_l$  represent the weights for the contour and lag errors, respectively.  $\mu$  adjusts the trade-off for higher progress speed  $v_{\hat{\theta}}$  to move as quickly as possible along the path, while  $\mathbf{Q}_q$  is the weight matrix for the quaternion error. Finally,  $\mathbf{R}$  penalizes the control effort;

In the context of NMPCC, a Lyapunov candidate function is introduced to ensure system stability while minimizing the error along the path. The Lyapunov candidate function  $V$  considers both the contour error  $\mathbf{e}_c(\hat{\theta}_k)$  and the lag error  $\mathbf{e}_l(\hat{\theta}_k)$ , and is defined as:

$$V = \frac{1}{2} \mathbf{e}_c^\top \mathbf{W}_c \mathbf{e}_c + \frac{1}{2} \mathbf{e}_l^\top \mathbf{W}_l \mathbf{e}_l,$$

where  $\mathbf{W}_c$  and  $\mathbf{W}_l$  are positive definite weighting matrices for the contour and lag errors, respectively.

The time derivative of  $V$  is then given by:

$$\dot{V} = \frac{\partial V}{\partial \mathbf{p}} \mathbf{v},$$

where the gradient of this function  $\frac{\partial V}{\partial \mathbf{p}}$  is computed with respect to the position vector  $\mathbf{p}$ , and  $\mathbf{v}$  is the velocity vector, previously defined.

To enforce the Lyapunov condition a slack variable  $\zeta \geq 0$  is introduced:

$$\dot{V} \leq -c_3 V + \zeta.$$

The slack variable  $\zeta$  introduces flexibility when constraints are too strict, allowing the system to remain feasible. To limit its impact,  $\zeta$  is penalized in the cost function with the term  $\rho \zeta^2$ , where  $\rho > 0$  controls the degree of penalization. This ensures feasibility in the Optimal Control Problem (OCP) while preserving stability. In real-time applications, the use of a slack variable becomes crucial for handling uncertainties and ensuring the optimization process remains feasible, especially under tight computational constraints.

The barrier function is designed to ensure that the distance between the UAV and the obstacle remains greater than the combined size of both, plus a safety margin. This function is expressed as:

$$h = \|\mathbf{p} - \mathbf{p}_{obs}\| - (r_{uav} + r_{obs} + \text{margin}),$$



where  $\mathbf{p}_{obs} = [x_{obs}, y_{obs}, z_{obs}]^T$  represents the position of the obstacle. The term  $r_{uav}$  corresponds to the UAV's radius,  $r_{obs}$  to the obstacle's radius, and the constant margin ensures a minimum additional distance to prevent collisions.

### B. ARC-LENGTH PARAMETRIZATION VIA THE BISECTION METHOD

To parametrize the desired trajectory  $\mathbf{p}_d(t)$  by arc length, we use the *bisection method* to determine the corresponding time  $t_k$  for each arc-length value  $\theta_k$ . Given a continuous trajectory  $\mathbf{p}_d(t)$ , the arc length from the initial point  $t_0$  to any point  $t_k$  is given by the following equation:

$$\theta_k = L(t_k) = \int_{t_0}^{t_k} \left\| \frac{d}{dt} \mathbf{p}_d(t) \right\| dt, \quad (27)$$

Since there is no explicit expression for  $t_k$  as a function of  $\theta_k$ , the bisection method is employed to numerically find  $t_k$  such that Eq. 27 is satisfied for the given  $\theta_k$ . To do so, we initialize the bisection algorithm with an interval  $[t_{low}, t_{high}]$ , where  $t_{low}$  and  $t_{high}$  are guesses that are expected to contain  $t_k$ . Typically,  $t_{low}$  can be initialized as  $t_0$ , the starting point of the trajectory, and  $t_{high}$  is chosen based on an estimate of where the desired arc length  $\theta_k$  is reached.

The bisection method, as outlined in Algorithm 1, iteratively refines the interval  $[t_{low}, t_{high}]$  to find the time  $t_k$  that corresponds to the desired arc length  $\theta_k$ . This process continues until the difference between the computed arc length  $L(t_{mid})$  and  $\theta_k$  falls below the specified tolerance  $\epsilon$ .

---

#### Algorithm 1 Time Calculation for $\theta_k$ Based on Arc Length

---

**Input:** Desired arc length  $\theta_k$ , initial interval  $[t_{low}, t_{high}]$ , tolerance  $\epsilon$ .

**while**  $|L(t_{mid}) - \theta_k| \geq \epsilon$  **do**

    Compute the midpoint  $t_{mid} = \frac{t_{low} + t_{high}}{2}$ .

    Calculate the cumulative arc length  $L(t_{mid}) = \int_{t_0}^{t_{mid}} \|\mathbf{p}'_d(t)\| dt$ .

**if**  $L(t_{mid}) < \theta_k$  **then**

        Update  $t_{low} = t_{mid}$ .

**else**

        Update  $t_{high} = t_{mid}$ .

**end if**

**end while**

**Output:**  $t_k = t_{mid}$  such that  $L(t_k) \approx \theta_k$ .

---

Once  $t_k$  is found, the corresponding position  $\mathbf{p}_d(\theta_k)$  is obtained as:

$$\mathbf{p}_d(\theta_k) = \mathbf{p}_d(t_k) = [x_d(t_k) \quad y_d(t_k) \quad z_d(t_k)]^T.$$

## VI. EXPERIMENTS AND RESULTS

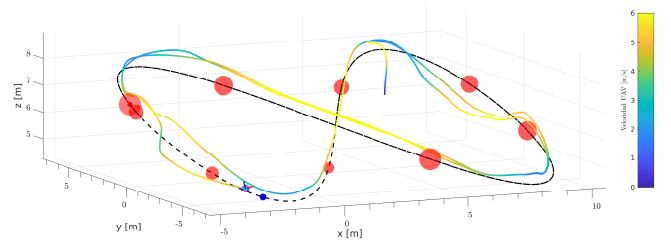
This section presents the results of applying the proposed NMPPC framework to a UAV system. The framework was tested in simulations where the UAV moved at high speeds in the presence of both static virtual obstacles and a mobile obstacle, in order to evaluate its path-following and obstacle-avoidance capabilities.

### A. EXPERIMENTAL SETUP

We conducted the experiments using a PC station with an AMD Ryzen 7 3700x and 16GB RAM, utilizing CasADi with ACADOS for optimal control. The controller was executed at 30 Hz with a prediction horizon of 30 steps (1 second). The UAV followed a figure-eight path, parametrized by arc length, with a maximum progress velocity  $v_{\theta_{max}}$  of 6 m/s. The UAV, with a radius of 0.15 m corresponding to a 6-inch frame [32], included a safety margin of 0.1 m. The contour and lag errors were weighted by  $\mathbf{Q}_c = 3\mathbf{I}_{3 \times 3}$  and  $\mathbf{Q}_l = \mathbf{I}_{3 \times 3}$ , respectively. The progress velocity was regulated by a gain  $\mu = 0.1$ , the control effort was penalized with  $\mathbf{R} = \text{diag}[0.02, 200, 200, 200]$  and the attitude error was weighted by  $\mathbf{Q}_q = \mathbf{I}_{3 \times 3}$ . The Lyapunov constraint,  $\mathbf{W}_c = \mathbf{W}_l = \mathbf{I}_{3 \times 3}$  and  $\rho = 100$  were used. A gain of  $\gamma = 0.9$  and  $\mathbf{K}_\alpha = [20, 8]^T$  for the static obstacle and  $\mathbf{K}_\alpha = [20, 15]^T$  for the mobile obstacle were applied. The following reference path was considered:  $x(t) = 4 \sin(0.04 \cdot t) + 1$ ,  $y(t) = 4 \sin(0.08 \cdot t)$ , and  $z(t) = 2 \sin(0.08 \cdot t) + 6$ . The quadrotor was required to complete the experimental path within a fixed duration of 30 seconds.

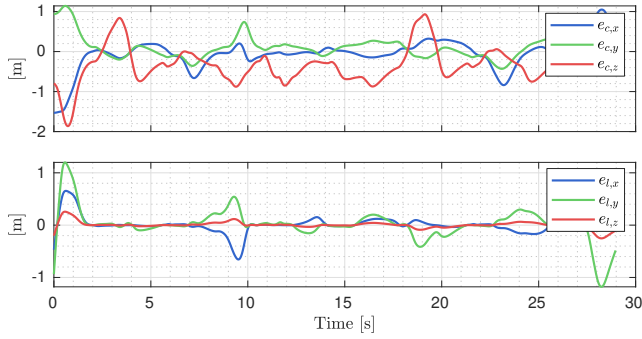
### B. SIMULATED PERFORMANCE

The MPCC framework was tested in simulations involving both static and mobile virtual obstacles to assess its path-following and obstacle-avoidance capabilities. During the test, a mobile obstacle moved along the same path but in the opposite direction, increasing the relative velocity between the UAV and the obstacle due to their opposing movements. Fig. 2 illustrates the UAV's trajectory as it navigates around both static and mobile obstacles, demonstrating the MPCC's effectiveness in minimizing contour and lag errors while ensuring consistent progress along the desired path.



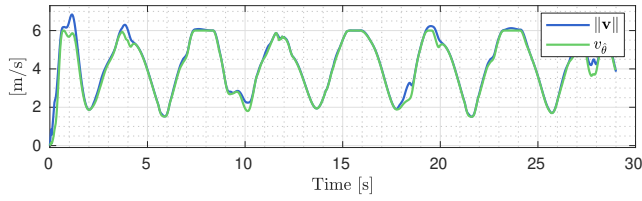
**FIGURE 2.** Flight path of UAV with MPCC in the presence of static obstacles.

The contour and lag errors recorded during the experiment are summarized in Fig. 3, demonstrating the effectiveness of MPCC in maintaining low error values, with instantaneous contour errors remaining below  $\|e_c\| < 1.97$  m and instantaneous lag errors below  $\|e_l\| < 1.87$  m, when obstacles are present. It is observed that during obstacle avoidance maneuvers, the contour error is the most affected, as the UAV adjusts its path to navigate around the obstacle.



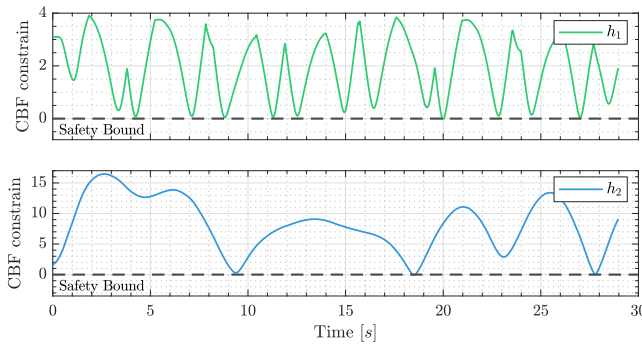
**FIGURE 3.** Contour and lag errors during path-following with obstacle avoidance.

In Fig. 4, the progress velocities and the norm of the velocity are shown. While both curves appear similar due to the maintained direction, it's important to note that the progress velocity is the projection of the system's velocity onto the reference's tangent vector. Additionally, the progress velocity adheres to the constraints set in the OCP, ensuring controlled advancement along the trajectory.



**FIGURE 4.** The system's instantaneous velocity and progress velocity.

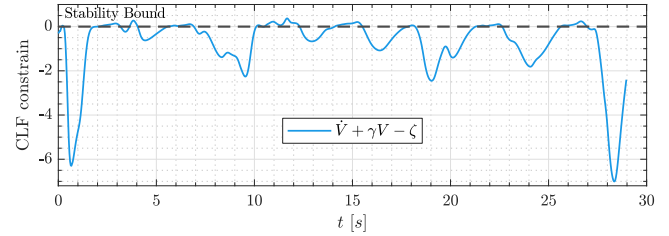
In Fig. 5, the combined ECBF values ( $h_1$  and  $h_2$ ) are displayed, demonstrating their effectiveness in enforcing safety constraints along the UAV's trajectory, particularly around static and moving obstacles.



**FIGURE 5.** CBF safety constraints.

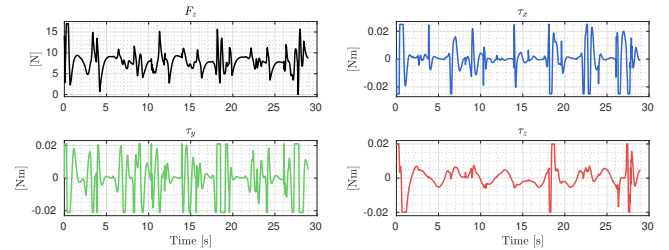
By ensuring that  $h(x) > 0$  throughout the experiment, the ECBF guarantees that the UAV avoids collisions and maintains safe operational limits throughout the flight. The minimum safety distance of between the obstacles was consistently maintained, with the CBF values remaining positive, thereby confirming that the safety measures were effectively enforced.

In Fig. 6, the ES-CLF stability constraint ensures that the system remains stable. Temporary increases in this constraint are observed when the UAV performs corrective maneuvers, particularly during obstacle avoidance. However, the inclusion of a slack variable allows for minimal violations when necessary, ensuring that the optimizer remains feasible. This ensures that the system remains operational, and despite these temporary deviations, the stability metric trends toward convergence as the UAV returns to its desired path, highlighting the system's robustness and ability to regain stability after corrective actions.



**FIGURE 6.** CLF stability constraints.

In Fig. 7, the control actions are shown, remaining within the limits set by the OCP. Notably, in the presence of obstacles, the control actions adjust effectively, ensuring obstacle avoidance without violating the imposed constraints. This illustrates the system's ability to handle complex scenarios while maintaining control stability.



**FIGURE 7.** Control actions within OCP limits.

## VII. CONCLUSION

The NMPCC framework provides an effective solution for tracking desired trajectories by minimizing both orthogonal and tangential errors while maximizing progress along the path. By integrating the trajectory geometry with the system dynamics, NMPCC enables precise and robust path tracking even in the presence of obstacles at high speeds.

The results from the ECLF and ECBF analyses confirm that the proposed NMPCC framework ensures both stability and safety during UAV path-following tasks. Future work will focus on extending this framework to multi-UAV systems and validating its performance in real-world flight tests.

Additionally, the NMPCC framework incorporates an attitude control strategy that uses the Log map to project quaternions into  $\mathbb{R}^3$ . This enables the direct minimization of attitude errors in the cost function, allowing the integration

of Lie algebra with standard Euclidean operations. As a result, the framework ensures precise alignment of the UAV's orientation with the desired trajectory, permitting complex maneuvers to be handled effectively in geometric space.

## ACKNOWLEDGMENTS

The authors are grateful for the support of INAUT - UNSJ, which provided them with the facilities and workspace necessary to carry out this research.

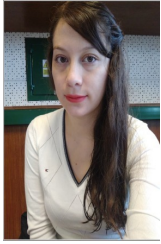
## REFERENCES

- [1] Y. Kumar, S. B. Roy, and P. B. Sujit, "Barrier lyapunov function based trajectory tracking controller for autonomous vehicles with guaranteed safety bounds," *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, pp. 722–728, 9 2020.
- [2] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020.
- [3] N. Gu, D. Wang, Z. Peng, and J. Wang, "Safety-critical containment maneuvering of underactuated autonomous surface vehicles based on neurodynamic optimization with control barrier functions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, pp. 2882–2895, 6 2023.
- [4] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Transactions on Robotics*, vol. 38, pp. 3340–3356, 12 2022.
- [5] A. Romero, R. Penicka, and D. Scaramuzza, "Time-optimal online replanning for agile quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 7, pp. 7730–7737, 7 2022.
- [6] D. Lam, C. Manzie, and M. Good, "Model predictive contouring control," *Proceedings of the IEEE Conference on Decision and Control*, pp. 6137–6142, 2010.
- [7] J. Xin, T. Xu, J. Zhu, H. Wang, and J. Peng, "Long short-term memory-based multi-robot trajectory planning: Learn from mpcc and make it better," *Advanced Intelligent Systems*, p. 2300703, 2024.
- [8] D. Lam, C. Manzie, and M. C. Good, "Model predictive contouring control for biaxial systems," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 552–559, 2013.
- [9] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, pp. 4459–4466, 10 2019.
- [10] M. Krinner, A. Romero, L. Bauersfeld, M. Zeilinger, A. Carron, and D. Scaramuzza, "Mpcc++: Model predictive contouring control for time-optimal flight with safety constraints," *Robotics: Science and Systems*, 2024.
- [11] D. Wang, Q. Pan, J. Hu, C. Zhao, and Y. Guo, "Mpcc-based path following control for a quadrotor with collision avoidance guaranteed in constrained environments," *IEEE International Symposium on Industrial Electronics*, vol. 2019-June, pp. 581–586, 6 2019.
- [12] Z. Du, H. Zhang, Z. Wang, and H. Yan, "Model predictive formation tracking-containment control for multi-uavs with obstacle avoidance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, pp. 3404–3414, 6 2024.
- [13] Z. Niu, X. Jia, and W. Yao, "Communication-free mpc-based neighbors trajectory prediction for distributed multi-uav motion planning," *IEEE Access*, vol. 10, pp. 13 481–13 489, 2022.
- [14] H. Huang, G. Zhu, Z. Fan, H. Zhai, Y. Cai, Z. Shi, Z. Dong, and Z. Hao, "Vision-based distributed multi-uav collision avoidance via deep reinforcement learning for navigation," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2022-October, pp. 13 745–13 752, 2022.
- [15] Z. Wang, T. Hu, and L. Long, "Multi-uav safe collaborative transportation based on adaptive control barrier function," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, pp. 6975–6983, 11 2023.
- [16] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.
- [17] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," *2019 18th European Control Conference, ECC 2019*, pp. 3420–3431, 6 2019.
- [18] K. Garg, J. Usevitch, J. Breeden, M. Black, D. Agrawal, H. Parwana, and D. Panagou, "Advances in the theory of control barrier functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems," *Annual Reviews in Control*, vol. 57, p. 100945, 1 2024.
- [19] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 10 2018.
- [20] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier-value functions for safety-critical control," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2021-December, pp. 6814–6821, 2021.
- [21] Z. Zheng, J. Li, Z. Guan, and Z. Zuo, "Constrained moving path following control for uav with robust control barrier function," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, pp. 1557–1570, 7 2023.
- [22] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 155–164, 10 2009.
- [23] J. Tang, S. Wu, B. Lan, Y. Dong, Y. Jin, G. Tian, W. A. Zhang, and L. Shi, "Gmpc: Geometric model predictive control for wheeled mobile robot trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 9, pp. 4822–4829, 5 2024.
- [24] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," 2021. [Online]. Available: <https://arxiv.org/abs/1812.01537>
- [25] G. Alcan, F. J. Abu-Dakka, and V. Kyriki, "Trajectory optimization on matrix lie groups with differential dynamic programming and nonlinear constraints," 2023. [Online]. Available: <https://arxiv.org/abs/2301.02018>
- [26] R. Verschuere, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, pp. 147–183, 3 2022.
- [27] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 3 2019.
- [28] A. Romero, R. Penicka, and D. Scaramuzza, "Time-optimal online replanning for agile quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 7, pp. 7730–7737, 7 2022.
- [29] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs," *Lecture Notes in Control and Information Sciences*, vol. 449 LNCIS, pp. 219–240, 2013.
- [30] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive clf-mpc with application to quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, pp. 565–572, 1 2022.
- [31] B. E. Jackson, K. Tracy, and Z. Manchester, "Planning with attitude," *IEEE Robotics and Automation Letters*, vol. 6, pp. 5658–5664, 7 2021.
- [32] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, 6 2022.



**BRYAN S. GUEVARA** graduated in Mechatronic Engineering from the University of the Armed Forces - ESPE in 2018. In 2024, he successfully earned his master's degree in Control Systems Engineering from the National University of San Juan. Currently, he is a doctoral student in Control Systems Engineering at the same university, supported by the DAAD scholarship (German Academic Exchange Service) through the Funding Program: Third Country Programme Latin America, 2022. His areas of interest are: aerial robotics, dynamic systems modeling, and optimal control.





include teleoperation systems and automatic control. In recent years she has been focusing also on Artificial Intelligence and Computer Vision.

**VIVIANA MOYA** is a Lecturer and Researcher at the Universidad Internacional del Ecuador. In 2021, she has a Ph.D. in Control Systems Engineering from The National University of San Juan (UNSJ) in Argentina. Her doctoral studies were supported by a DAAD scholarship from Germany. Prior to this, Viviana completed her undergraduate education in Electronics and Control Engineering at the Escuela Politécnica Nacional (EPN) in Quito, Ecuador, in 2016. Her professional interests



He's working currently with nonlinear control design, the inclusion of dynamic models and the applications of USVs (unmanned surface vessels) in environmental monitoring, surveillance and control.

**JUAN M. TOIBERO** graduated in Electronic Engineering from the National Technological University of Parana (Argentina) in 2002, received the Ph.D degree in Control Systems Engineering from the National University of San Juan (Argentina) in 2007. He is with the Instituto de Automática, National University of San Juan from 2002 as full professor and with CONICET from 2011 as full-time researcher. His research interests are related to the automatic control of mobile robotic platforms.



**LUIS F. RECALDE** graduated in Mechatronics Engineering from the University of the Armed Forces - ESPE in 2021. Currently, he is a master's student in Control Systems Engineering at the National University of San Juan, Argentina. His research interests include Nonlinear Model Predictive Control (NMPC) and Reinforcement Learning (RL). Over the past few years, he has actively merged control theory with machine learning.



Automation and Control area. From 2014 to 2020, he was directly responsible for the UITEC Unidad de Innovación Tecnológica at Universidad de Las Américas. Since 2014, he has been a professor in the Engineering in Electronics and Automation program at Universidad de Las Américas, Quito (Ecuador). His research mainly focuses on Robotics, Machine Learning, Control, and Computer Vision.

**DAVID POZO-ESPIN** obtained a Bachelor's degree in Electronics and Control Engineering from Escuela Politécnica Nacional in 2010, a Master's degree in Automatics and Robotics from Universidad Politécnica de Catalunya, Barcelona, Spain, in 2012, and a Master's degree in Artificial Intelligence from Universidad Internacional de Valencia in 2023. He worked as a full-time Adjunct Professor (2010-2011) and as an Assistant Professor (2013-2014) at Escuela Politécnica Nacional in the



included algorithms for management energy systems and optimal control strategies with application in unmanned aerial vehicles (UAV).

**DANIEL C. GANDOLFO** graduated in Electronic Engineering from the National University of San Juan (UNSJ), Argentina, in 2006. He has been working as automation engineer in the industry until 2009 and received the Ph.D. in Control Systems Engineering from the UNSJ in 2014. Currently, he is researcher of the Argentinean National Council for Scientific Research (CONICET), and an Associate Professor in the Institute of Automatics, UNSJ-CONICET, Argentina. His research interest