A Velocity qLMPC Algorithm for Path-Following with Obstacle Avoidance for Fixed-Wing UAVs

Ahmed Samir[1,4], Horacio M. Calderón[1,3], Herbert Werner[1],
Benjamin Herrmann[2], Leif Rieck[2], and Frank Thielecke[2]

***Abstract*— This paper tackles the problem of path-following control for fixed-wing unmanned aerial vehicles (UAVs), while accounting for wind disturbances and hindering obstacles. We introduce a novel predictive algorithm based on a quasi-linear parameter-varying (qLPV) model representation of the 3D kinematics of the fixed-wing aircraft. This approach allows us to utilize efficient *Quadratic Programming* (QP) solvers to find efficient and fast solutions to the *Optimal Control Problem* (OCP), typically within milliseconds. Additionally, it facilitates the incorporation of appropriate constraints aligned with the aircraft dynamics and obstacle constraints after further processing. In this paper, we demonstrate how the nonlinear obstacle constraints can also be represented in a qLPV form, making it feasible to handle them within our framework. Moreover, stability conditions can be directly derived based on the qLPV representation. The algorithm's effectiveness is demonstrated on an aerobatic unmanned aircraft with a successive-loop-closure (SLC) based attitude and stabilization controller. The evaluation is conducted across two scenarios previously used in experimental flights with the same aircraft. Each scenario involves nine waypoints, obstructive obstacles, and wind disturbances. The simulations begin with the kinematic model and are subsequently extended to a high-fidelity model of the UAV, resulting in successful path-following and obstacle avoidance with relatively low computational times.**

I. INTRODUCTION

In recent decades, UAVs have gained widespread use in diverse areas like mapping, surveillance, and agriculture [1]. This surge in UAV adoption has prompted advancements in aircraft autonomy, posing complex challenges in future aircraft technologies. Path-following control stands as one of the pivotal applications for aircraft autonomy. Existing literature highlights two main strategies for addressing the path-following problem: (a) the geometric strategies, and (b) the control-based strategies. Geometric strategies focus on adjusting yaw and pitch movements based on desired path changes. They are simple but may lack performance in windy or uncertain conditions. A method in this category is line-of-sight (LOS), see [2], [3], [4], and in the context of autonomous marine vehicle guidance [5]. Control-based strategies, use control theory methods to design controllers that ensure robustness against wind and uncertainties. Widely used methods include PID control [6] and techniques like LQR [7], sliding mode [8], and adaptive control [9]. Model predictive control (MPC) is another potent solution for path-following problem due to its ability to handle constraints through optimization and its predictive capability to assimilate information regarding the forthcoming path. Extensive research has focused on developing efficient nonlinear predictive control, using techniques like real-time iteration scheme [10], [11] and gradient based schemes [12]. However, the drawback of MPC is the potential complexity and memory demands in computing solutions, especially if the model and constraints are nonlinear, which are required if high tracking performance is desired. Moreover, striking a balance between capturing the nonlinear position dynamics and providing stability and performance guarantees remains a challenge in predictive path-following control design. In this paper, we aim to enhance the autonomy of fixed-wing UAVs by introducing a path-following capability through an innovative predictive algorithm that builds upon a qLPV representation of the aircraft's 3D kinematic model, addressing the challenges mentioned earlier through the utilization of the quasi-linear parameter-varying model predictive control (qLMPC) framework [13] and a quadratic problem formulation with linear constraints. This approach ensures efficient optimization and under certain assumptions, stability guarantees can be given. Recently, this framework was applied for stabilization and reference tracking of a flexible aircraft using Laguerre functions [14]. Moreover, we employ the qLMPC framework to address the path-following with obstacle avoidance problem, crucial in UAV tasks like surveillance or mapping. The challenge involves tracking a preset path while evading obstacles without deviating from the trajectory. Addressing obstacle constraints within our framework poses a nontrivial challenge, particularly due to their nonlinear nature. One potential solution involves linearizing these constraints to facilitate QP solving. However, a new challenge arises: to ensure the meaningfulness of these linearized constraints, they must be linearized with respect to a specific distance from the object. Consequently, the linearized constraints need to be updated at each time step, transforming them into qLPV constraints rather than linear constraints. For further insights into qLPV constraints, refer to [15]. This problem contrasts with obstacle avoidance in path-planning, e.g. in [16], where new paths are generated around obstacles using Spline-RRT (Rapidly exploring Random Tree) and velocity obstacle techniques. While significant, this method can be complex and computationally demanding for predictive control.

This work is a part of our broader endeavor to introduce

[1]Hamburg University of Technology, Institute of Control Systems, Hamburg, Germany
[2]Hamburg University of Technology, Institute of Aircraft Systems Engineering, Hamburg, Germany
[3]IAV GmbH, Research, Gifhorn, Germany
[4]Correspondence: ahmed.samir@tuhh.de

an efficient and innovative predictive control methodology for addressing the path-following problem using the qLMPC algorithm, which was recently introduced in [17]. Here, we integrate nonlinear obstacle constraints into our qLMPC framework using a synthetic waypoint guidance (SWG) path-planner, while ensuring the preservation of QP solving capabilities. Finally, we put this research to the test through simulations involving a high-fidelity model of the fixed-wing Unmanned Low-cost Testing and Research Aircraft (ULTRA-Extra) [18].

The paper is structured as follows: Section II outlines the path-following problem for fixed-wing UAVs within the qLMPC framework, in a cascaded structure. Here, we provide in-depth details of the construction of the qLPV model, the formulation of the OCP, and the integration of obstacle avoidance constraints. In Section III, we explore the synthetic waypoint path-planning approach. Section IV gives details on the fixed-wing aircraft considered in this study. Section V presents the simulation results. Finally, in Section VI, we conclude the paper by summarizing the main aspects and outline potential directions for future research.

A. Notation

The transpose of a matrix A is $A^\top$. The Jacobian of the function $f(x,u)$ with respect to the variable x is $\nabla_x f$. A positive (semi-)definite matrix Q is denoted as $Q \succ 0$ ($Q \succeq 0$). The Kronecker product between the matrices A and B is $A \otimes B$. A block diagonal matrix with a Q matrix repeated N times along the diagonal is $\tilde{Q} = \mathrm{diag}_N(Q)$.

II. Problem Setup

In this study, we propose a strategy that combines predictive path-following with static obstacle avoidance, while preserving fast computational processing using a synthetic waypoint path-planner. The research comprises four main stages: First, developing a qLPV model for a fixed-wing aircraft based on the 3D kinematic model (1). Second, formulating a suitable cost function with aircraft constraints to minimize the error between the aircraft and the path. Third, implementing the qLMPC algorithm. Lastly, integrating the synthetic waypoint path-planner [19] into our framework for versatile scenario testing. This section encompasses the initial three stages, with the final stage being addressed in the subsequent section. An overview of the entire system setup within a cascaded scheme is depicted in Fig.1, where the reference and aircraft position components are in earth-centered, earth-fixed (ECEF) coordinate system.

To achieve a qLPV model for a fixed-wing aircraft kinematics, velocity-based linearization [20] is used. This technique accurately transforms the nonlinear model into a qLPV form, distinct from Jacobian linearization's affine form. The resulting model is linear in both state and control input derivatives. This approach aligns well with offset-free MPC, enhancing disturbance robustness using incremental state and input values instead of absolute quantities. Moreover, we are specifically aiming to address the obstacle avoidance problem within the path-following framework, rather than the path-planning problem. In path-planning, obstacles are managed by re-planning path. However, in path-following, we aim to navigate around obstacles without changing the intended path. Instead, we utilize sensor data from the aircraft to adjust commands based on obstacle and aircraft positions. The challenge lies in balancing obstacle avoidance and staying close to the planned path. The path-following controller adapts commands to evade obstacles while minimizing path deviation. After successfully avoiding obstacles, the controller promptly adapts its commands to minimize errors and return to the planned path.

Consider the nonlinear 3D kinematic model of the aircraft

$$\underbrace{\begin{bmatrix}\dot{x}_\mathrm{p}\\ \dot{y}_\mathrm{p}\\ \dot{z}_\mathrm{p}\\ \dot{\chi}\\ \dot{\gamma}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} V_g\cos\gamma\cos\chi \\ V_g\cos\gamma\sin\chi \\ -V_g\sin\gamma \\ \frac{g}{V_g}\tan\phi \\ \frac{g}{V_g}(n_z\cos\phi-\cos\gamma)\end{bmatrix}}_{f(x,u)}, \tag{1}$$

where the state is $x = [x_\mathrm{p}, y_\mathrm{p}, z_\mathrm{p}, \chi, \gamma]^\top$ such that $x_\mathrm{p}, y_\mathrm{p}, z_\mathrm{p}$ are the 3D position components in ECEF coordinate system, χ is the aerodynamic course angle and γ is the flight path angle. The input is $u = [V_g, \phi, n_z]^\top$ such that V_g is the aircraft speed, ϕ is the bank angle, and n_z is the incremental load factor. The output is $y = h(x,u) = x$. Applying velocity-based linearization to (1), we derive the qLPV model

$$\underbrace{\begin{bmatrix}\dot{y}\\ \ddot{x}\end{bmatrix}}_{\dot{x}_v} = \underbrace{\begin{bmatrix}0 & \nabla_x h(x,u)\\ 0 & \nabla_x f(x,u)\end{bmatrix}}_{A_v}\underbrace{\begin{bmatrix}y\\ \dot{x}\end{bmatrix}}_{x_v} + \underbrace{\begin{bmatrix}\nabla_u h(x,u)\\ \nabla_u f(x,u)\end{bmatrix}}_{B_v}\dot{u}. \tag{2}$$

The system and input matrices are dependent on the states and control inputs. Thus, the obtained model is a qLPV model. It's important to carefully select appropriate initial conditions when using velocity-based linearization to ensure meaningful outcomes. Another important point to note is that the system matrix A_v has eigenvalues at 0, which signifies the existence of integral action within the obtained system. In order to utilize this model in a predictive scheme, the model is discretized using the Runge-Kutta 4 (RK4) method. The qLPV model described in equation (2) is updated at each time instant to obtain an LTI representation, allowing us to construct a quadratic cost function with linear constraints in the next step. The cost function can be expressed as follows

$$J = \sum_{i=1}^{N}\left(x_{v,k+i}^\top Q x_{v,k+i} + \Delta u_{k+i-1}^\top R \Delta u_{k+i-1} + e_{k+i}^\top T e_{k+i}\right). \tag{3}$$

Here, $x_{v,k+i}$ denotes the vector of augmented states resulting from the velocity-based linearization at sampling instant $k+i$, Δu_{k+i-1} denotes the control input increment at sampling instant $k+i-1$, and $e = r - x$ denotes the tracking error between the predicted aircraft position and the path. The weight on the state is $Q = Q^\top \succeq 0$, the weight on the control effort is $R = R^\top \succ 0$, and the weight on the tracking error

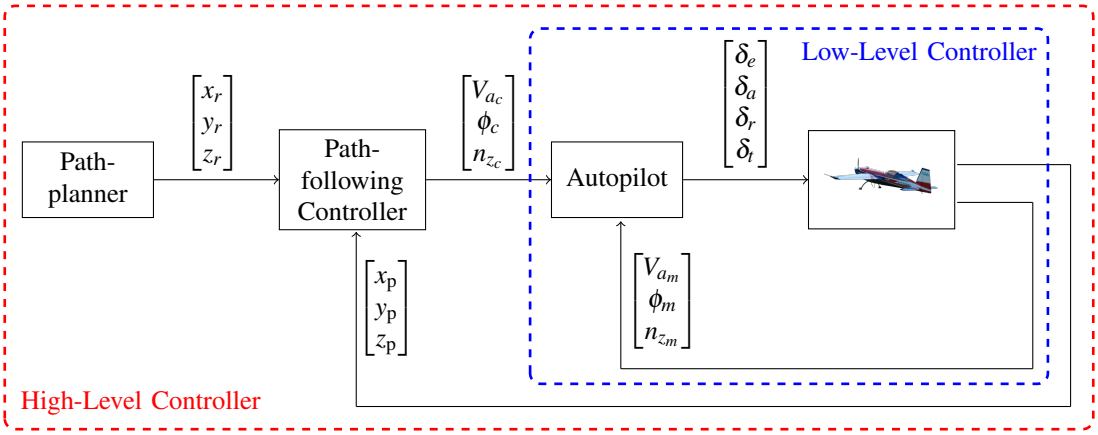

Fig. 1. Path-following problem setup where the position's reference is $[x_r, y_r, z_r]^\top$, the aircraft's position is $[x_p, y_p, z_p]^\top$, and the actuation signals for the elevators, ailerons, rudder and throttle are $[\delta_e, \delta_a, \delta_r, \delta_t]^\top$.

is $T = T^\top \succeq 0$. The prediction horizon is N. Using the prediction equation

$$X_k = H(P_k)x_{v,k} + S(P_k)\Delta U_k, \tag{4}$$

where $X_k \in \mathbb{R}^{Nn}$ is the vector of augmented states predictions such that $X_k = [x_{v,k+1}^\top \;\; x_{v,k+2}^\top \;\; \cdots \;\; x_{v,k+N}^\top]^\top$, $P_k = [\rho_k^\top \;\; \rho_{k+1}^\top \cdots \rho_{k+N-1}^\top]^\top$ is the vector of future scheduling variables, $\Delta U_k \in \mathbb{R}^{Nm}$ is the future control inputs' increments such that $\Delta U_k = [\Delta u_k^\top \;\; \Delta u_{k+1}^\top \;\; \cdots \;\; \Delta u_{k+N-1}^\top]^\top$, $H \in \mathbb{R}^{Nn \times n}$ is a matrix multiplied by the actual state $x_{v,k} \in \mathbb{R}^n$, $S \in \mathbb{R}^{Nn \times Nm}$ is a Toeplitz matrix multiplied by the vector of future control inputs' increments ΔU_k, n is the number of augmented states, and m is the number of control inputs, we can reformulate the cost function to be in a QP form as follows

$$J_k = \frac{1}{2}\Delta U_k^\top \underbrace{2(S^\top(P(k))\tilde{Q}S(P_k) + \tilde{R} + S^\top(P_k)\tilde{T}S(P_k))}_{H_s}\Delta U_k + \underbrace{2(x_{v,k}^\top H^\top(P_k)\tilde{Q}S(P_k) - R_k^\top \tilde{T}S(P_k) + x_{v,k}^\top H^\top(P_k)\tilde{C}^\top \tilde{T}S(P_k))}_{g^\top}\Delta U_k. \tag{5}$$

where H_s is the Hessian matrix of the QP, and g is the gradient vector. As previously mentioned, MPC offers the advantage of managing constraints. In our case, we establish constraints that align with our objective of achieving seamless path-following while adhering to the aircraft's dynamic and nonholonomic constraints. Furthermore, we incorporate appropriate constraints to circumvent potential obstacles along the trajectory. In particular, our constraints cover control input constraints and state constraints. If needed, we can also include output constraints to restrict the region in which the aircraft operates. Additionally, to ensure seamless transitions along the path, we impose constraints on the rate of change of control inputs. For obstacle avoidance integration, we introduce supplementary position state constraints. Let's consider a scenario where the aircraft is following a predefined path for a mission such as mapping or surveillance, and encounters an object along the planned path. The object is to be avoided while keeping the tracking error minimal. One possible approach is to impose a three-dimensional constraint on the aircraft's trajectory to create a buffer zone around the object [21]. To elaborate, let's assume the center of the object is represented by coordinates x_o, y_o, and z_o. We can then enforce a constraint on the distance d between the object's center and the aircraft's position. This distance (d), calculated in Euclidean space, can be determined by calculating the norm distance between the obstacle center and the aircraft position. Then, we can utilize this formula to establish a constraint between d and a specified distance r by ensuring that $d^2 > r^2$. This inequality can then be integrated into the existing set of constraints. However, in order to solve the optimization problem efficiently as a QP with linear constraints, it is necessary to linearize this constraint. The inequality constraint can be expressed as follows

$$(x_p - x_o)^2 + (y_p - y_o)^2 + (z_p - z_o)^2 > r^2. \tag{6}$$

Through linearization of the obstacle constraints and flipping the inequality to match the format used in the QP, we can express the obstacle avoidance inequality as follows:

$$\begin{bmatrix} 2x_o - 2x_p & 2y_o - 2y_p & 2z_o - 2z_p & 0 & 0 \end{bmatrix} x < -r^2 + x_o^2 + y_o^2 + z_o^2 - x_p^2 - y_p^2 - z_p^2. \tag{7}$$

Here, it is important to note that x_p, y_p, and z_p are updated online, which means that it can be treated as a qLPV constraint [15] with the constraint scheduling vector $\rho_k^c = [x_p, y_p, z_p]^\top$. It is worth mentioning that the constraint scheduling vector does not necessarily have to be the same as the system scheduling vector. Now, the cost function is in quadratic form, and the constraints take a linear form. Therefore, it becomes straightforward to solve a QP using (5) while adhering to the defined constraints and obtain

$$\Delta U_k^* = \min_{\Delta U_k} J_k \qquad s.t. \quad A\Delta U_k \leq b, \tag{8}$$

where $A \in \mathbb{R}^{q \times p}$ is a matrix multiplied by the decision variables ΔU_k and $b \in \mathbb{R}^q$ is the constraints vector. The scalar p indicates the number of decision variables, and q represents the number of constraints.

So far, we have discussed how to obtain a qLPV model that captures the nonlinear position dynamics of the aircraft using velocity-based linearization. We have also explained how to construct a cost function with integral action to achieve an offset-free path-following. The next step is to integrate the quadratic optimization problem into the controller algorithm. The qLMPC algorithm is summarised in Algorithm 1, where l represents the number of iterations.

Algorithm 1 QLMPC Algorithm [22]

Initialisation: Model (1), $\tilde{Q}$, $\tilde{R}$, $\tilde{T}$, N
  $k \leftarrow 0$
  Define $P^0 = 1_N \otimes \rho(x(0), u(0))$
  repeat
    $l \leftarrow 0$
    repeat
      Solve QP(8) with P_k^l for ΔU_k^l
      Predict state $X_k^l = H(P_k^l)x_k + S(P_k^l)\Delta U_k^l$
      Define $P_k^{l+1} = \rho(X_k^l, U_k^l)$
      $l \leftarrow l+1$
    until stop criterion
    Apply $u_k = u_{k-1} + \Delta u_k$
    Define $P_{k+1}^0 = \rho(X_k^l, U_k^l)$
    $k \leftarrow k+1$
  until end

III. Path-planner

In this section, we propose a way of using the synthetic waypoint guidance (SWG) algorithm, which was presented in [19] based on a pure pursuit guidance law to generate a continuous path derived from different waypoints. The resulting position components will serve as the reference signal for the path-following controller ($[x_r, y_r, z_r]^\top$ in Fig. 1). The SWG algorithm utilizes a flight path that is initially defined to establish the desired trajectory for the aircraft. The flight path waypoints are specified in inertial space with reference to a fixed local frame. Additionally, a synthetic waypoint (SW) is introduced at the location of the first real waypoint. The SWG algorithm operates by tracking the SW as it moves along the designated flight path. It is considered synthetic because its position on the flight path is determined as the position at which the aircraft intends to be within a specified time horizon. This time horizon is determined by the user and represents the time interval by which the aircraft trails the SW. As the aircraft approaches the SW on the flight path, it is re-positioned, thereby compelling the vehicle to adopt a new heading to pursue it. The previous discussion highlights that the SWG can be viewed as a virtual point tracking technique. To delve into the stability analysis of this algorithm, refer to [23].

We leverage this method to generate two distinct paths. In both scenarios, there are a total of nine waypoints, with the first and last waypoints being identical. On the one hand, the first scenario represents an aerodrome circuit path, characterized by varying altitudes ranging from 150 m to 250 m. On the other hand, the second scenario presents a more challenging trajectory, resembling a roller-coaster ride with altitudes ranging from 100 m to 300 m. It is important to note that these scenarios have been previously employed in flight experiments, albeit with the utilization of a nonlinear guidance law (NLGL) combined with cubic splines [24]. In this work, we integrate the SWG algorithm into the qLMPC framework to generate the required reference components in the ECEF coordinate system.

IV. ULTRA-Extra Aircraft and Controller Architechture

To demonstrate the effectiveness of the proposed algorithm, we utilize the ULTRA-Extra [18], a fixed-wing aircraft developed by the Institute of Aircraft Systems Engineering at Hamburg University of Technology. The ULTRA-Extra, depicted in Fig. 2, is a scaled replica of the Extra 330 ML aerobatic aircraft powered by a 7.2 kW electric motor, thus enabling experiments up to 20 minutes. With a mass of 24.6 kg and a wingspan of 3.1 m, it offers manual control through a remote controller or automated control via a flight control computer. A real-time computer manages measurement processes and guidance, navigation and control applications, while dual antennas provide precise heading and GPS position measurements.

Fig. 2. ULTRA-Extra aircraft.

A. Aircraft Modeling

For simulation, a comprehensive nonlinear model of the ULTRA-Extra is used, encompassing aerodynamics, actuator dynamics, and a model of the electric propulsion system. This model is based on data from a wind tunnel campaign and flight tests. Simulated wind and atmospheric turbulence adhere to MIL-F-8785C standards. Additionally, a 3D kinematic model is employed for path-following control and MPC purposes. This model incorporates primary control inputs such as the aircraft airspeed, bank angle, and the incremental load factor. The low-level controller uses elevators, ailerons, rudder, and throttle to track these control commands.

B. Stabilization and Attitude Controller Design

This section focuses on the design of the low-level controller using successive-loop-closure to track commands from

the predictive path-following controller. As the ULTRA-Extra is an aerobatic aircraft with negligible roll-yaw coupling effects, a cascaded controller structure based-on single-input single-output controllers is employed, following integrator chain principles. The low-level controller structure should align with the objective of executing 3D path-following commands, so we opt for the bank angle and incremental load factor as the primary control signals for lateral and longitudinal motions, respectively. An auto-throttle regulates airspeed. In Fig. 3, a simulated random scenario assesses the controller's performance in managing attitude changes and maintaining a sideslip angle of ($\beta = 0$) for clean flight and coordinated turns. Appropriate bandwidth selection is crucial for desired performance and stability, with emphasis on the highest bandwidth for the inner loops. Loop shaping techniques were employed for achieving control objectives.

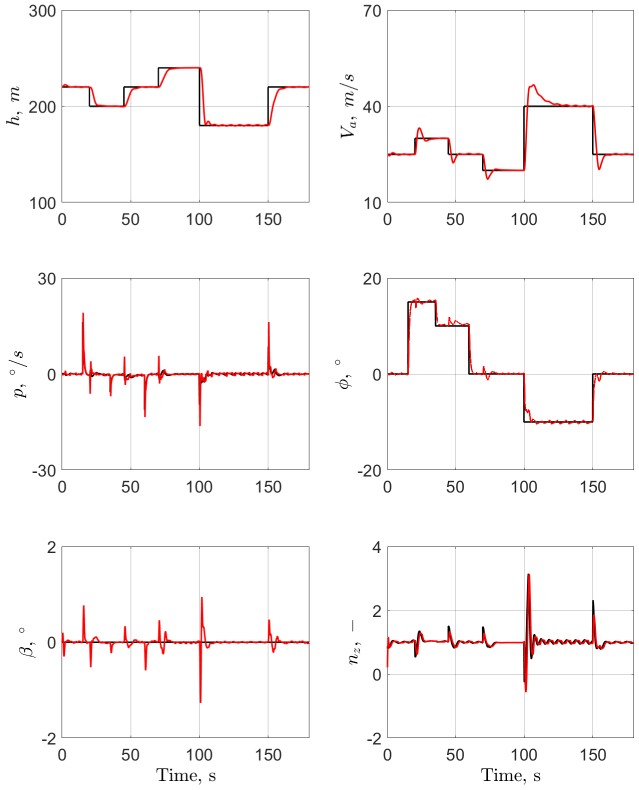

Fig. 3. Stabilization and attitude controller evaluation: command (—), measurement (—).

V. Simulation Results

In this section, we present simulation results demonstrating precise path-following with minimal errors and successful obstacle avoidance. We also discuss the computational times involved. Moreover, we provide insights into the stability analysis and convergence of the qLMPC algorithm, a critical research aspect requiring attention.

A. Tracking and Computational Time

Here, we evaluate the effectiveness and performance of the qLMPC path-following framework. In both scenarios, we consider the presence of two static obstacles, with radii ranging from 10 m to 25 m. For demonstration purposes, the obstacles are represented as cylinders in three-dimensional coordinates to simulate structures resembling buildings that the aircraft must navigate around during its mission. However, it is worth noting that representing them as spheres would be more appropriate. We assume that the pilot will guide the aircraft towards the vicinity of the first waypoint, at which the path-following mode is promptly initiated. In addition, we simulate a constant wind with an average speed of $V_w = 4$ m/s and an orientation of $\chi_w = 89°$ for the first scenario. For the second scenario, we simulate time-varying wind with V_w ranging from 1 m/s to 5 m/s and an orientation of $\chi_w = 335°$. We selected these values to closely match the wind conditions observed in a previous flight test conducted on the same aircraft. Our constraints encompass several critical factors. Firstly, we constrain the aircraft's speed within the range of 20-50 m/s, which aligns with the parameters of the designed low-level controller. This controller was developed based on Jacobian linearization across seven operating points within the same speed range. Moreover, it ensures that the predictive path-following controller does not increase or decrease the airspeed to undesired values (e.g. zero, which is not feasible for fixed-wing aircraft during flight). Secondly, we limit the bank angle to a range of -30° to 30° for comfortable flight during path-following mode, although this limitation is unnecessary for the ULTRA-Extra, given its aerobatic capabilities. Additionally, the incremental load factor is confined between -1 and 2.3. These constraints account for the structural limitations of the aircraft, as detailed in [24] for the same aircraft. Finally, to ensure seamless transitions along the path, we have introduced constraints on the rate of change of control inputs.

From Fig. 4 and Fig. 5, it is evident that the aircraft can smoothly follow the planned path with minimal tracking error until it encounters the first obstacle. At this juncture, the planned path adjusts as intended, but the aircraft's response is altered by the path-following algorithm, which enforces adherence to the defined hard obstacle constraints, necessitating the aircraft to maneuver around the obstacle without deviating significantly from the original path. Once the obstacle is cleared, the aircraft swiftly aims to minimize the tracking error and regain alignment with the planned path while adhering to its non-holonomic constraints. This process repeats when the aircraft encounters the second obstacle, following a similar behavior. The direction in which the aircraft maneuvers to avoid the obstacle is determined by the path-following controller based on the optimal solution of the optimization problem. In the first scenario depicted in Fig. 4, the controller chooses to navigate around the obstacle. However, in the second scenario shown in Fig. 5, the controller opts to deviate slightly in the yaw direction, resulting in a minor altitude loss. It is crucial to adhere to the

constraints defined in the optimization problem. Therefore, careful consideration should always be given to constructing an appropriate cost function with suitable constraints.

This approach is comparable to the one introduced in [25], where a customized nonlinear predictive controller was applied. However, the key difference lies in our utilization of a qLPV model for the problem, as opposed to using Taylor series linearization to convert the optimization problem into a quadratic form. It is important to note that in this study, the airspeed is considered as a controllable parameter by the path-following controller. This differs from the approach taken in previous works such as [24] and [26], where the airspeed was commanded separately in the context of using NLGL and nonlinear geometric differential path following guidance (NGDPFG) path-following algorithms. In Table I, we provided a summary of the computation times related to the qLMPC algorithm. 'Matrix' represents the time needed to construct the system and constraints matrices at each sampling instant, 'quadratic optimization problem (QP)' is the time needed to solve the quadratic problem, and 'qLMPC' is the total execution time for the entire algorithm, requiring only a single iteration ($l = 1$ in Algorithm 1). The optimization problem was solved using an 8-core processor running at 3.6 GHz, using the *quadprog* solver from the MATLAB Optimization Toolbox. In our implementation, a sampling time of 0.01 s was used for the path-following loop, 0.001 s for the low-level controller, and a prediction horizon of $N = 10$ steps was chosen. During initial controller tuning, our algorithm was applied to the simplified 3D kinematic model of the aircraft, introduced in eq. (1). This served as a preliminary step before applying the algorithm to the full-state model. We fine-tuned the controller for compatibility with the full model, primarily due to the model's finite bandwidth. No major structural changes to the controller were necessary.

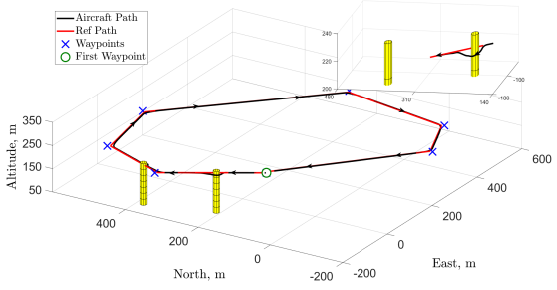

Fig. 4. Test scenario: 'Aerodrome circuit'.

TABLE I

COMPUTATIONAL TIMES OVERVIEW IN MILLI-SECONDS.

	Aerodrome Circuit Path			Roller Coaster Path		
	Min	Max	Mean	Min	Max	Mean
Matrix	1.231	5.678	2.234	1.133	5.541	2.122
QP	0.448	3.534	1.286	0.427	3.472	1.138
qLMPC	0.452	3.643	1.294	0.431	3.501	1.147

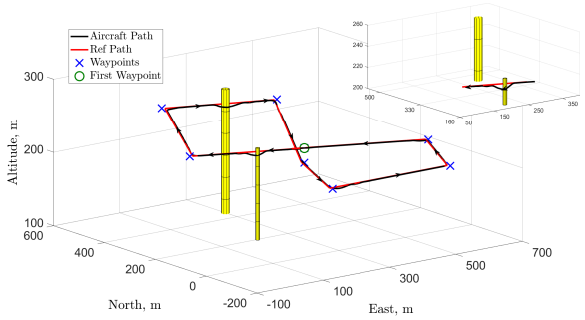

Fig. 5. Test scenario: 'Roller-coaster'.

B. Comment on Stability and Convergence of the Algorithm

An important research direction for the qLMPC algorithm involves examining its stability and convergence properties. To gain initial insights into the algorithm's stability, we've initiated an investigation, based on [22]. In this work, the authors tackled the stability concern by solving offline linear matrix inequalities (LMIs). It's worth noting that this analysis was conducted for a state-space qLPV model, and it may not be necessary for a velocity-based qLPV model, potentially alleviating the need for solving these LMIs. Regarding convergence properties, our research has also ventured into this aspect, as presented in [27]. In this study, the authors observed that the qLMPC algorithm doesn't always converge to the optimal solution; at times, it converges to a sub-optimal one. This issue is significant, and the paper introduces a solution to address it, focusing on unconstrained problems.

VI. CONCLUSION

In this paper, we have introduced a predictive path-following control law based on the qLPV model of the aircraft's position dynamics. This framework allowed us to efficiently solve a quadratic optimization problem with linear constraints. Our algorithm has proven its efficiency in achieving precise tracking and obstacle avoidance, even in the presence of constant and time-varying winds. In this study, we dealt with 3D obstacles, which inherently have nonlinear characteristics. To tackle this issue, we performed linearization to enable the solution of a QP and had to schedule these constraints for incorporation into the qLMPC framework, transforming them into qLPV constraints. To ensure a smooth and continuous path-following while respecting the dynamic and non-holonomic constraints of the aircraft, we employed a synthetic waypoint path-planner. This approach allowed us to generate a well-defined trajectory that can be easily followed by the aircraft. In future work, we plan to address the concerns highlighted in section V-B regarding stability and convergence properties of the proposed algorithm. Additionally, we aim to conduct hardware-in-the-loop (HIL) simulations as a preliminary step towards performing experiments on the real aircraft.

REFERENCES

[1] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned

aerial vehicles (uavs): A survey on civil applications and key research challenges," *Ieee Access*, vol. 7, pp. 48 572–48 634, 2019.

[2] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1092–1100, 2006.

[3] G. Conte, S. Duranti, and T. Merz, "Dynamic 3d path following for an autonomous helicopter," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 472–477, 2004, iFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004.

[4] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti, and M. A. D. Virgilio, "Algorithms for 3d uav path generation and tracking," *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5275–5280, 2006.

[5] N. Gu, D. Wang, Z. Peng, J. Wang, and Q.-L. Han, "Advances in line-of-sight guidance for path following of autonomous marine vehicles: An overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 12–28, 2023.

[6] M. Sun, R. Zhu, and X. Yang, "Uav path generation, path following and gimbal control," in *2008 IEEE International Conference on Networking, Sensing and Control*, 2008, pp. 870–873.

[7] A. Ratnoo, P. Sujit, and M. Kothari, "Adaptive optimal path following for high wind flights," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 985–12 990, 2011, 18th IFAC World Congress.

[8] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[9] A. P. Aguiar, I. Kaminer, R. Ghabcheloo, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, *Time-Coordinated Path Following of Multiple UAVs over Time-Varying Networks using L1 Adaptation*. American Institute of Aeronautics and Astronautics Inc. (AIAA), 2008.

[10] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[11] R. Verschueren, 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," 2020.

[12] T. Englert, A. Völz, F. Mesmer, S. Rhein, and K. Graichen, "A software framework for embedded nonlinear model predictive control using a gradient-based augmented lagrangian approach (grampc)," *Optimization and Engineering*, pp. 1–41, 2018.

[13] P. S. G. Cisneros, S. Voss, and H. Werner, "Efficient nonlinear model predictive control via quasi-LPV representation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, dec 2016.

[14] L. Rieck, B. Herrmann, F. Thielecke, and H. Werner, "Efficient quasi-linear model predictive control of a flexible aircraft based on laguerre functions," in *2023 American Control Conference (ACC)*, 2023, pp. 2855–2860.

[15] P. S. Cisneros, A. Sridharan, and H. Werner, "Constrained predictive control of a robotic manipulator using quasi-lpv representations," *IFAC-PapersOnLine*, vol. 51, no. 26, pp. 118–123, 2018, 2nd IFAC Workshop on Linear Parameter Varying Systems LPVS 2018.

[16] S. Zhang, T. Xu, H. Cheng, and F. Liang, "Collision avoidance of fixed-wing uavs in dynamic environments based on spline-rrt and velocity obstacle," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 48–58.

[17] A. S. Rezk, H. M. Calderón, H. Werner, B. Herrmann, and F. Thielecke, "Predictive path following control for fixed wing uavs using the qlmpc framework in the presence of wind disturbances," Jan 2024. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2024-1594

[18] M. Krings, B. Annighöfer, and F. Thielecke, "Ultra - unmanned low-cost testing research aircraft," in *American Control Conference*, 2013, pp. 1472–1477.

[19] E. D. B. Medagoda and P. W. Gibbens, "Synthetic-waypoint guidance algorithm for following a desired flight trajectory," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 601–606, 2010.

[20] P. S. G. Cisneros and H. Werner, "A velocity algorithm for nonlinear model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 1310–1315, 2021.

[21] M. P. Adhikari and A. H. J. de Ruiter, "Real-time autonomous obstacle avoidance for fixed-wing uavs using a dynamic model," *Journal of Aerospace Engineering*, vol. 33, no. 4, p. 04020027, 2020.

[22] P. S. González Cisneros and H. Werner, "Nonlinear model predictive control for models in quasi-linear parameter varying form," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 10, pp. 3945–3959, 2020.

[23] E. D. Medagoda, "Closed loop stability of synthetic waypoint guidance algorithm," *IFAC Proceedings Volumes*, vol. 43, no. 15, pp. 75–80, 2010, 18th IFAC Symposium on Automatic Control in Aerospace.

[24] N. Sedlmair, J. Theis, and F. Thielecke, "Design and experimental validation of uav control laws - 3d spline-path-following and easy-handling remote control," in *Proceedings of the 5th CEAS Conf. Guid., Navigation, Control*, 2019.

[25] M. P. Adhikari and A. H. J. de Ruiter, "Real-time autonomous obstacle avoidance for fixed-wing uavs using a dynamic model," *Journal of Aerospace Engineering*, vol. 33, no. 4, p. 04020027, 2020.

[26] N. Sedlmair, J. Theis, and F. Thielecke, "Experimental comparison of nonlinear guidance laws for unmanned aircraft," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 805–14 810, 2020, 21st IFAC World Congress.

[27] C. Hespe and H. Werner, "Convergence properties of fast quasi-lpv model predictive control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 3869–3874.