Predictive Path-Following with Stability Guarantee for Fixed-Wing UAVs Using the qLMPC Framework

Ahmed Samir[1,3], Horacio M. Calderón[1,2], and Herbert Werner[1]

Abstract— **This paper tackles the problem of path-following control for fixed-wing unmanned aerial vehicles (UAVs) in the presence of wind disturbances with stability guarantee constraints. Building upon our prior research, we continue to advance our novel predictive path-following algorithm, grounded in a quasi-linear parameter-varying (qLPV) model representing the 3D kinematics of fixed-wing UAVs. In this study, we have introduced additional stability guarantee constraints, all while retaining the high-efficiency path-following performance. This qLPV model, established through a velocity-based linearization strategy, enables us to implement offset-free model predictive control (MPC), known for its robustness against disturbances, and to solve a quadratic optimization problem at each time step, proving its efficacy in path-following applications. Furthermore, this representation permits the addition of few constraints to the optimization problem, ensuring stability without the necessity of solving complex offline LMIs. To evaluate the effectiveness of our algorithm, we tested it on a 24.6 kg aerobatic UAV, navigating through two scenarios with nine waypoints each. Our simulations began with a 3D kinematic model and progressed to a higher-fidelity one, showing strong performance with assured stability. We assume the proposed algorithm converges to the optimal solution, yet recent studies on unconstrained problems noted cases of suboptimal convergence.**

I. INTRODUCTION

UAVs, or drones, are widely applied across various sectors. In military and security operations, they provide real-time aerial intelligence, aiding surveillance and reconnaissance efforts without risking human lives. In disaster management, they assess affected areas, assist in search and rescue missions, and offer rapid, high-resolution imagery. In agriculture, UAVs optimize crop management, monitor soil conditions, and assess plant health [1], [2], while also being used for environmental monitoring, wildlife conservation, and infrastructure inspection, providing a cost-effective means of data collection in remote or hazardous locations.

In these scenarios, efficient navigation is essential for drones to fulfill their missions effectively. This entails techniques like waypoint-following, trajectory-tracking, and path-following. In this study, our focus is on the path-following problem.

Existing literature presents two main approaches to address the path-following problem: geometric strategies (e.g., [3], [4], [5], [6]), which primarily involve adjusting yaw and pitch movements based on desired path changes, offering simplicity but potentially lacking performance in windy or uncertain conditions; and control-based strategies (e.g., [7], [8], [9], [10]) that leverage control theory techniques to design controllers, ensuring robustness against wind and uncertainties.

Within the realm of control-based strategies, MPC stands out as a robust solution for the path-following problem due to its predictive capability and ability to handle constraints through optimization. Extensive research has focused on devising effective Nonlinear Model Predictive Control (NMPC) techniques, employing methodologies like real-time iteration schemes [11], [12] and gradient-based approaches [13]. However, MPC's complexity and memory requirements can be significant, especially with nonlinear models and constraints. Balancing nonlinear position dynamics with stability and performance guarantees remains a key challenge, addressed in various research efforts such as [14].

Ensuring stability in predictive algorithms involves defining conditions or constraints to bound system outputs. One approach, detailed in [15], entails solving offline Linear Matrix Inequalities (LMIs) for Linear Time-Invariant (LTI) systems, incorporating them as constraints within the optimization problem. Extending this to qLPV systems, [16] employed a gridding technique, albeit considered conservative. Notably, [17] presents stability conditions for velocity-based qLPV systems without requiring offline LMIs, applied successfully to stabilize a robotic arm.

This paper's key contribution lies in integrating the stability conditions from [17] into our predictive path-following framework [18], ensuring both efficient path-following performance and guaranteed stability. The framework we've introduced relies on the qLPV model of a fixed-wing UAV's 3D kinematic model. It employs a velocity-based qLPV model to accurately capture the nonlinear position dynamics inherent in fixed-wing aircraft. This approach tackles the aforementioned challenges by leveraging the quasi-linear parameter-varying model predictive control (qLMPC) framework [16], formulating the problem as a quadratic program with linear constraints for computational efficiency. Recently, this framework was employed to stabilize and achieve reference tracking for a flexible aircraft using Laguerre functions [19]. Moreover, it aligns with existing stability results from literature, previously applied based on qLPV systems to ensure stability. 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) [20].

The paper is structured as follows: Section II details the path-following problem for fixed-wing UAVs in the qLMPC framework, covering the construction of the qLPV model

[1] Hamburg University of Technology, Institute of Control Systems, Hamburg, Germany

[2] IAV GmbH, Research, Gifhorn, Germany

[3] Correspondence: ahmed.samir@tuhh.de

and the formulation of the optimal control problem (OCP) with stability guarantee constraints. Section III introduces a hybrid path-planning approach combining arc-length parameterization with cubic splines to create a continuous path from waypoints. Section IV provides a brief overview of the fixed-wing aircraft used. Section V presents simulation results. Finally, Section VI concludes by summarizing key points and suggesting future research directions.

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 $Q \succ 0$ ($Q \succeq 0$).

II. Problem Setup

In this section, we present a predictive path-following approach for fixed-wing UAVs, enhancing our prior algorithm [18] with stability guarantee conditions. We cover four main stages: 1) Developing a qLPV model using velocity-based linearization, 2) Constructing a stability-focused cost function, 3) Applying the qLMPC algorithm [16]. We address these stages here, leaving the path-planner discussion for the next section.

The path-following problem, illustrated in Fig. 1, follows a cascaded structure. The outer loop, initiated by the path-planner, generates reference position components (r) for the path-following controller. This controller generates commands (u_H) for the UAV's autopilot, which, in turn, instructs the elevators, ailerons, rudder, and throttle (u_L) to achieve precise path following.

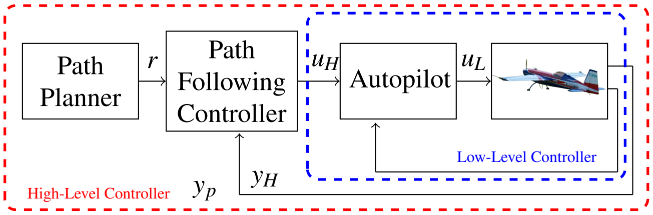

Fig. 1. Problem setup, where $r^\top = [x_r, y_r, z_r]$, $u_H^\top = [V_{a_c}, n_{z_c}, \phi_c]$, $u_L^\top = [\delta_e, \delta_a, \delta_r, \delta_t]$, $y_H^\top = [V_{a_m}, n_{z_m}, \phi_m]$, $y_p^\top = [x_p, y_p, z_p]$.

A. Optimization Problem

To derive a qLPV model for a fixed-wing UAV, we first employ a 3D kinematic model, which effectively captures most of its nonlinear positional dynamics. Then, we utilize velocity-based linearization [17] to derive the qLPV model. The 3D kinematic model used is

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

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

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

Note that the system and input matrices depend on the measured states and control inputs, rendering the model a qLPV. Another notable observation from the model derived in (2) is the presence of an eigenvalue at zero within the system matrix A_v, indicating integral action resulting from velocity-based linearization. To apply the acquired model in a predictive framework, it is discretized using the Runge-Kutta 4 (RK4) method. This yields an LTI model representation that can be updated at each time step. Subsequently, a quadratic cost function with linear constraints can be constructed, commonly 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 denotes the input increments, while $e = r - x$ represents 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 $R = R^\top \succ 0$, the weight on the tracking error is $T = T^\top \succeq 0$, and N is the prediction horizon. We can expand this cost function into a quadratic program, as demonstrated in [18]. Then, we can express the optimization problem as follows

$$\Delta U_k^* = \min_{\Delta U_k} \; J_k \tag{4a}$$

s.t.

$$\dot{x}_v = A_v x_v + B_v \dot{u} \tag{4b}$$

$$x_{v,k+i} \in \mathbb{X} \tag{4c}$$

$$u_{k+i-1} = u_{k-1} + \Sigma_{j=0}^{i} \Delta u_{k+j} \in \mathbb{U}, \; i \in [1,N] \tag{4d}$$

$$\Delta u_{k+i-1} \in \tilde{\mathbb{U}} \tag{4e}$$

$$y_{k+N} = y_{sp} \tag{4f}$$

$$\Delta x_{k+N} = 0 \tag{4g}$$

As previously highlighted, our approach excels in effectively managing constraints, crucial for achieving seamless path-following while respecting the fixed-wing UAV's nonholonomic constraints. State limitations (4c) regulate the aerodynamic course angle and flight-path angle, while constraints (4d) define maximum values for the bank angle, incremental load factor, and aircraft speed. Additional constraints (4e) ensure smooth changes in control inputs for seamless path-following. Constraints (4f) and (4g) guarantee algorithm stability, simplifying the quadratic program solution process.

$$\Delta U_k^* = \min_{\Delta U_k} J_k \qquad \text{s.t.} \quad A\Delta U_k \leq b, \tag{5}$$

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.

So far, we've outlined the creation of a qLPV model, capturing the aircraft's nonlinear position dynamics via velocity-based linearization. Additionally, we've detailed the cost function formulation, integrating integral action for offset-free path-following, which interfaces with QP solvers. Next, we integrate this QP into the controller algorithm. Algorithm 1 summarizes the qLMPC algorithm, where l denotes the iteration count.

Algorithm 1 QLMPC Algorithm [21]

```
Initialisation: Model, $\tilde{Q}$, $\tilde{R}$, $\tilde{T}$, N
  $k \leftarrow 0$
  Define $P^0 = 1_N \otimes \rho(x(0), u(0))$
  repeat
    $l \leftarrow 0$
    repeat
      Solve QP (5) with $P_k^l$ for $\Delta U_k^l$
      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
```

B. Stability of qLMPC

Below, we present stability conditions tailored for our proposed qLMPC algorithm, leveraging the velocity-based qLPV model's ability to map all equilibria to the origin ($\dot{x} = 0$, $\dot{u} = 0$). This eliminates the need for equilibria parameterization, which can be intricate for nonlinear systems. Our analysis extends the framework from [17], with essential adjustments to accommodate our application's nonholonomic constraints specific to fixed-wing UAVs.

Theorem 1 *[17] Let the terminal offset penalty be $P = \alpha Q_1$ for some $\alpha \geq 0$. Given a set point y_{sp}, the control law $\kappa(y_k, \dot{x}_k, y_{sp})$ derived from the solution of the optimization problem* (4a) *starting from a feasible state $[y^\top \ \dot{x}^\top]^\top$ stabilizes the system* (2)*, is recursively feasible and makes the output converge to one of the following*

1) *y_{sp} is $y_{sp} \in \mathbb{R}_y$*
2) *$\tilde{y}$ if $y_{sp} \notin \mathbb{R}_y$ where*

$$\tilde{y} = \arg\min_{y\in\mathbb{R}_y} ||y - y_{sp}||_P^2$$

where $\mathbb{R}_y$ is the admissible output set.

Proof: For a detailed proof, refer to [17]. ■

To solve (4a) with stability guarantee, constraints (4f) and (4g) are essential to validate Theorem 1. The inclusion of those two constraints guarantees that the obtained solution is both a setpoint, y_{sp}, and an equilibrium at the same time (Note that $y = x$).

Remark 1: The constraint (4f) can be incorporated into the cost function, transforming it into a soft constraint instead of a hard constraint. This acknowledges the practical challenges of precisely reaching the reference position due to aircraft dynamics and potential wind disturbances during UAV flight.

Remark 2: For similar reasons, the constraint (4g) needs to be relaxed to a small value ε, where $\underline{\varepsilon} \leq \Delta x_{k+N} \leq \overline{\varepsilon}$. This relaxation is essential as, at the end of the prediction horizon, the set position point should closely align with the reference position point, and the change rate of this position should ideally be minimal to ensure stability and effective path-following. However, achieving an exact zero change rate is impractical due to the continuous nature of the path, and reducing the aircraft's airspeed to zero during path-following would result in a crash.

Remark 3: Regarding the qLMPC algorithm's convergence properties, we assume consistent convergence to the optimal solution. However, in [22], sub-optimal solutions were sometimes observed. This study primarily focuses on state-space-based qLPV representations, not velocity-based ones, and exclusively deals with unconstrained problems. Further investigation is needed, particularly concerning velocity-based qLPV models for constrained problems like the one addressed here.

III. Path-planner

In this section, we employ a hybrid path-planning method from our prior research, combining arc-length parameterization [23] with cubic splines. This approach merges concepts from arc-length based paths for continuity and arc-length parameterized spline curves for real-time simulations [24] to create a smooth path from a sequence of waypoints. To achieve this, we introduce an additional state to represent the arc-length parameter, updated in real-time to track the path evolution. An extra constraint ensures accurate path progression. While advantageous for continuity, preplanning based on waypoint alterations is required. This mirrors the method used in [25], employing cubic splines for nonlinear guidance law (NLGL) path-following. The resulting path takes the following form: $\vec{S}_k(\tau) = [S_x^k(\tau) \ \ S_y^k(\tau) \ \ S_z^k(\tau)]^\top$, $k \in \{1, ..., n-1\}$. Here, $\vec{S}_k(\tau)$ is the cubic spline parametrized in arc-length, τ, that evolves online such that $\dot{\tau} = \lambda\tau + \nu$, where λ is a constant, chosen in this case as -10^{-3}, and ν is an additional virtual control input utilized to manage the speed of path evolution. Now the path $\mathscr{P}$ is regular, sufficiently differentiable, such that $\forall \tau \in [0, \hat{\tau}] : \tau \rightarrow p(\tau) \in \mathbb{X} \subseteq \mathbb{R}^n$ and $p(0) = 0$ holds. Therefore, the path $\mathscr{P}$ to be followed can be defined as $\mathscr{P} = \{p(\tau) \in \mathbb{R}^n : [0, \hat{\tau}] \rightarrow p(\tau)\}$.

The reference path will now be referred to as $[x_r(\tau) \ \ y_r(\tau) \ \ z_r(\tau) \ \ \psi_r(\tau) \ \ \gamma_r(\tau)]^\top$ and can be defined as

$$p(\tau) = \begin{bmatrix} a\tau^3 + b\tau^2 + c\tau + d \\ e\tau^3 + f\tau^2 + g\tau + h \\ i\tau^3 + j\tau^2 + k\tau + l \\ \tan^{-1}\dfrac{dy(\tau)}{dx(\tau)} \\ \sin^{-1}\dfrac{dz(\tau)}{\sqrt{dx^2(\tau) + dy^2(\tau)}} \end{bmatrix}.$$

Here, the coefficients $a, b, c, d, e, f, g, h, i, j, k$, and l are constants, determined based on the specific path waypoints and computed in advance. Then, the path is updated in real-time to accommodate variations in τ.

IV. ULTRA-Extra Aircraft and Controller Architechture

To illustrate the effectiveness of the proposed algorithm, we employ the ULTRA-Extra [20], a fixed-wing aircraft developed by the Institute of Aircraft Systems Engineering at Hamburg University of Technology. This model, depicted in Fig. 2, emulates the Extra-330 ML aerobatic aircraft, featuring a 7.2kW electric motor for experiments lasting up to 20 minutes. With a mass of 24.6 kg and a wingspan of 3.1 m, it offers manual or autonomous control via a remote controller or flight control computer.

Fig. 2. ULTRA-Extra Aircraft

A. Aircraft Modeling

In the simulations, we utilize a detailed nonlinear model of the ULTRA-Extra, considering aerodynamics, actuator dynamics, and the electric propulsion system. This model is crafted from data acquired through wind tunnel tests and flight experiments, while simulated wind conditions and atmospheric turbulence comply with MIL-F-8785C standards.

B. Stabilization and Attitude Controller Design

This section outlines the design of the low-level controller, employing successive-loop-closure to track commands from the predictive path-following controller. Given the ULTRA-Extra's aerobatic nature with minimal roll-yaw coupling, we use cascaded single-input single-output controllers, focusing on the bank angle and incremental load factor for lateral and longitudinal movements, respectively. The auto-throttle regulates airspeed. Figure 3 demonstrates a simulated scenario evaluating the controller's performance in managing attitude changes and maintaining a side-slip angle of $\beta = 0°$. Loop-shaping techniques were employed for effective control objectives. For detailed insights, refer to [18].

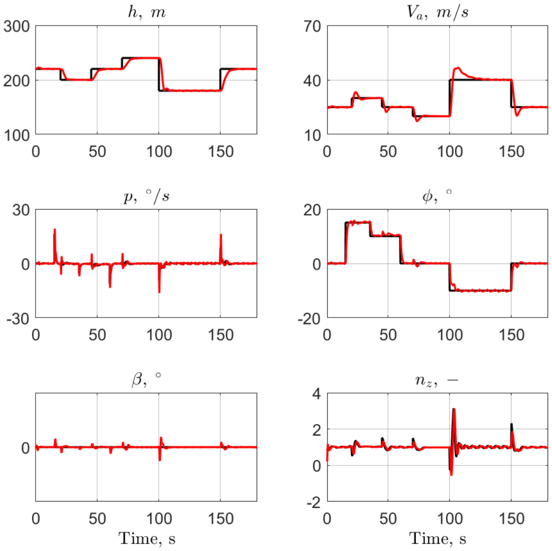

Fig. 3. Stabilization and attitude controller evaluation: command (—), measurement (—).

V. Simulation Results

In this section, we present simulation results to showcase the stability of our algorithm. We consider two scenarios with nine waypoints each, replicating 'Aerodrome circuit' and 'Roller-coaster' paths. The first scenario has altitude variations from 100 m to 300 m, with constant-speed wind ($V_w = 4$ m/s) and orientation $\chi_w = 89°$). The second scenario features time-varying wind ($V_w = 1 - 5$ m/s) and orientation $\chi_w = 335°$). These simulations closely resemble real-world wind conditions experienced by the same aircraft in prior flight experiments.

A. Tracking

Here, we assess the effectiveness of our qLMPC framework for path-following. Initially, we assume the pilot guides the aircraft near the first waypoint before activating path-following mode. We impose constraints to regulate the aircraft's behavior: speed limited to 20-50 m/s, consistent with the low-level controller's parameters derived from Jacobian linearization across seven operating points. This prevents undesirable airspeed fluctuations. Bank angle is confined within -30° to 30° for comfort, and the incremental load factor between -1 and 2.3, considering structural limitations. Additionally, constraints on control input rate changes ensure smooth transitions along the path.

Figures 4 and 7 depict seamless path-following in the planned scenarios. Notably, our approach integrates airspeed as a control parameter, differing from prior methods such as [25] and [26], where airspeed was independently commanded, particularly in NLGL and nonlinear geometric differential path following guidance (NGDPFG) algorithms. We solved the optimization problem using an 8-core 3.6 GHz processor and the *quadprog* solver in MATLAB Optimization Toolbox. Initially, the algorithm was implemented with a simplified 3D kinematic model for controller tuning, ensuring performance refinement before transitioning to the full-state model. Fine-tuning on the complete model accommodated its finite bandwidth without significant structural modifications to the controller's core.

B. Stability

To ensure algorithm stability, we leverage the velocity-based approach in formulating the qLPV model, offering advantages over other NMPC-based methods. As explained in section II, two additional constraints are imposed on the cost function to ensure stability. The first constraint aligns the final prediction horizon solution with the setpoint ($y_{k+N} = y_{sp}$), where y_{sp} is the reference from the path-planner. The second constraint ensures this solution corresponds to a steady-state ($\Delta x_{k+N} = 0$).
In both scenarios, Figures 5 and 8 validate the achievement of the first stability constraint, ensuring that the solution obtained at the prediction horizon's terminus closely aligns with the reference setpoint. Similarly, Figures 6 and 9 validate the successful fulfillment of the second stability constraint, indicating proximity to equilibrium. As discussed in Remark 2, we slightly relaxed this constraint to maintain performance without compromising stability. Notably, the terminal constraints do not apply to the arc-length parameter state due to its role in path evolution, which remains unconstrained to ensure correct path tracking. This aspect, managed by the path-planner, doesn't compromise stability and can be omitted if employing an alternative path-planner, such as the synthetic waypoint path-planner introduced in [18].

VI. CONCLUSION

In this paper, we refine our predictive path-following algorithm by integrating stability constraints. Leveraging a qLPV model of the aircraft's position dynamics, we efficiently address path-following via a quadratic optimization problem, ensuring both effectiveness and stability. Our path generation method, combining arc-length parameterization with cubic splines, yields a seamless, continuous path while accommodating dynamic and non-holonomic constraints. Simulation results demonstrate effective path-following with assured stability. Future research will delve into convergence properties under inequality constraints and explore hardware-in-the-loop (HIL) simulations for real-world validation.

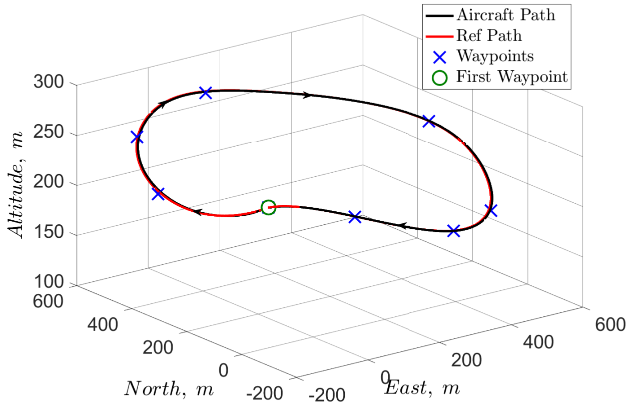

Fig. 4. Scenario 1 (Tracking): Aerodrome Circuit

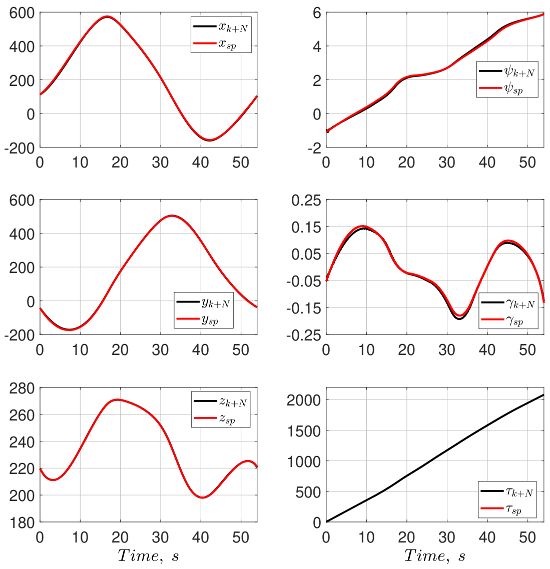

Fig. 5. Scenario 1 (Stability-First constraint ($y_{k+N} \approx y_{sp}$))

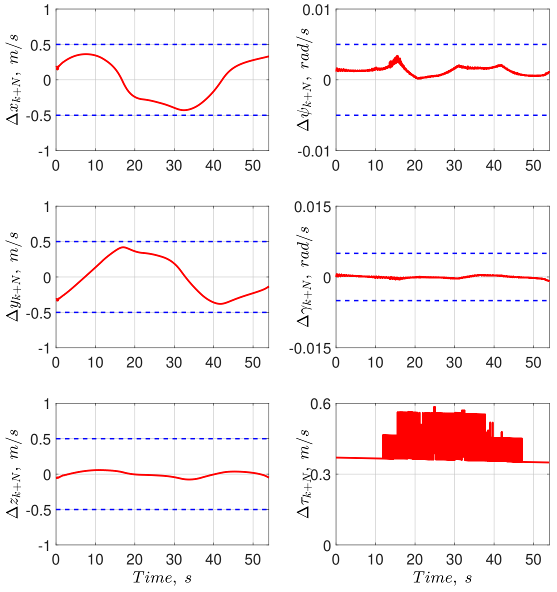

Fig. 6. Scenario 1 (Stability-Second constraint ($\Delta x_{k+N} \approx 0$))

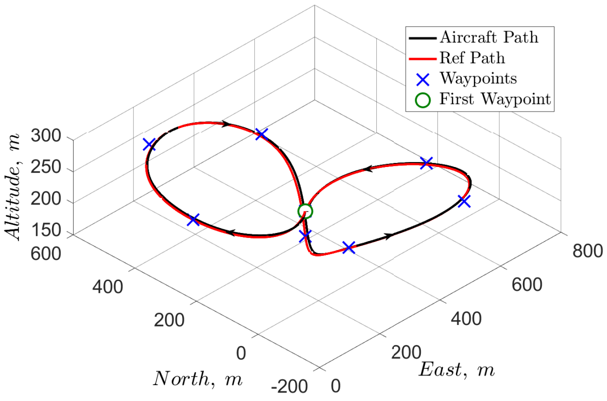

Fig. 7. Scenario 2 (Tracking): Roller-coaster

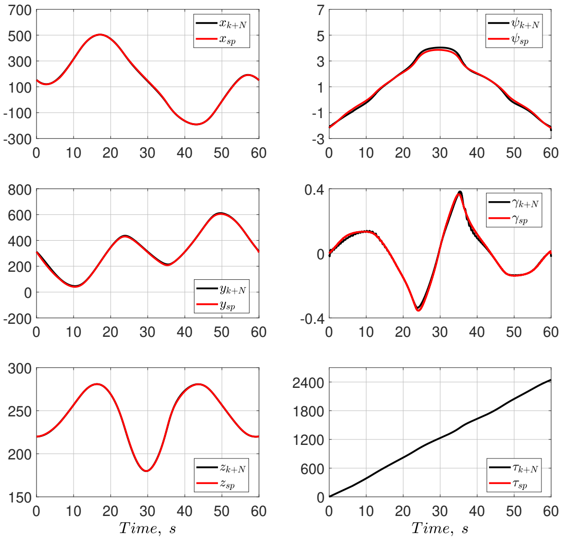

Fig. 8. Scenario 2 (Stability-First constraint ($y_{k+N} \approx y_{sp}$))

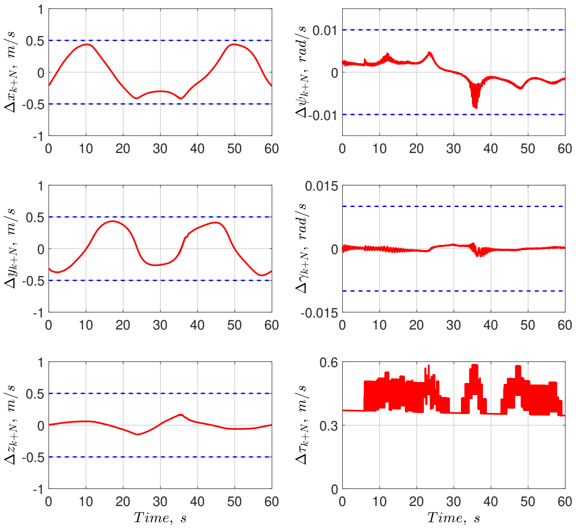

Fig. 9. Scenario 2 (Stability-Second constraint ($\Delta x_{k+N} \approx 0$))

REFERENCES

[1] J. G. A. Barbedo, "A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses," *Drones*, vol. 3, no. 2, 2019.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[15] J. Löfberg, "Linear model predictive control stability and robustness," 2001.

[16] 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.

[17] 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.

[18] A. Samir, 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," accepted for publication in American Institute of Aeronautics and Astronautics (AIAA), Jan 2024. [Online]. Available: https://github.com/AhmedSamierSaied/AIAA2024

[19] 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.

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

[21] 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.

[22] 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.

[23] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, dec 2009.

[24] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," in *Proc. 5th International Conference on Curves and Surfaces*, vol. 387396, 2002.

[25] 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.

[26] ——, "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.