Experimental validation of zonotopic Tube-MPC applied to a Hexacopter

Delansnay Gilles, Laurent Dewasme, Alain Vande Wouwer
System, Estimation, Control and Optimization
University of MONS
Mons, Belgium
gilles.delansnay@umons.ac.be, alain.vandewouwer@umons.ac.be

***Abstract*—In this paper, we present a real-time control strategy for a unmanned aerial vehicle operating under external disturbances, utilizing a Tube-based Model Predictive Control (Tube-MPC) framework integrated with a zonotopic representation. The approach ensures robust control by enforcing a bounded deviation between the real system trajectory and the nominal trajectory. The UAV's control problem is formulated as a linear MPC for nominal control, and a state feedback controller for disturbance rejection, using zonotopes to compute robust invariant sets. The proposed method is tested on a UAV equipped with a Pixhawk 4 flight controller and Raspberry Pi 5, where the system's robustness is validated in outdoor experiments under disturbances. The results demonstrate both the system's performance and safety under real-world conditions.**

***Index Terms*—Tube model predicitve control, zonotope, UAV, expriment, experimental, robust, robustness, drone**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have become increasingly popular in a wide range of applications, from environmental monitoring and precision agriculture to search and rescue operations [1], [2]. However, ensuring reliable and robust control of UAVs in real-world environments poses significant challenges. These systems must operate effectively in the face of uncertainties and disturbances such as wind gusts, sensor noise, and variations in the external environment. In this context, designing a control system capable of maintaining stability and performance despite these disturbances becomes crucial. Achieving this robustness in a computationally efficient manner is key, especially for real-time applications such as autonomous navigation and obstacle avoidance.

Model Predictive Control (MPC) [3] has emerged as a prominent method in UAV control due to its ability to handle multivariable control problems and enforce constraints on both states and control inputs. However, classical MPC assumes perfect knowledge of the system model and is sensitive to external disturbances. To address this limitation, Tube-based Model Predictive Control (Tube-MPC) has been introduced as a robust control framework. Tube-MPC enhances the resilience of MPC by constructing a "tube" around the nominal trajectory, ensuring that the actual trajectory of the system remains within a bounded deviation despite disturbances [4], [5].

In this work, we present the implementation of a real-time control algorithm based on Tube-MPC for a UAV system operating under disturbances. Our approach leverages zonotopic representations and reachability analysis to ensure robustness while maintaining computational efficiency. Zonotopes, as convex polytopes represented through generators, offer a compact and computationally feasible way to represent sets [6], [7]. Using zonotopes in our reachability analysis allows for the systematic tightening of constraints, ensuring that the UAV's actual trajectory remains within safety bounds despite uncertainties in the system.

The zonotopic framework is particularly well-suited for real-time control applications due to its ability to compute reachable sets efficiently. Unlike traditional polyhedral approaches, zonotopes can represent high-dimensional invariant sets with fewer parameters, reducing the computational burden [8]. By employing constrained zonotopes, we ensure that the control strategy adheres to the required safety and performance limits while accounting for disturbances in real time. Reachability analysis further enhances the robustness of the system by systematically accounting for all possible disturbances and ensuring that the system remains within a safe operational envelope [9], [10]. This is especially critical for UAVs, which often operate in dynamic and unpredictable environments.

The proposed control strategy comprises two main components: a linear MPC for the nominal system and a state feedback ancillary controller. The linear MPC computes optimal trajectories based on an augmented model that includes integrators for the UAV's position states, making it easier to track reference inputs for position and yaw angle. The ancillary controller, on the other hand, operates based on the real-time state of the UAV and corrects any deviations from the nominal trajectory caused by disturbances. By combining the predictive capabilities of MPC with the fast response of a state feedback controller, the Tube-MPC architecture achieves both robust performance and computational efficiency.

Our contributions in this paper are twofold: First, we demonstrate the effectiveness of Tube-MPC in managing disturbances for a UAV system in real-world conditions. Second, we show how zonotopic reachability analysis can be integrated into the Tube-MPC framework to handle uncertainties, ensuring both robustness and feasibility systematically. The UAV used in our experiments is equipped with a Pixhawk 4 flight controller and a Raspberry Pi 5, running the control algorithm in real time. The system is tested outdoors, with disturbances like wind gusts influencing the UAV's dynamics. Experimental results confirm the robustness of the proposed approach, with the UAV maintaining stability and performance despite significant disturbances.

The remainder of this paper is structured as follows: Section II describes the UAV system and the experimental setup. Section III provides an overview of the Tube-MPC framework and its integration with zonotopic reachability analysis. In Section IV, we present the experimental results, followed by a discussion in Section V. Finally, Section VI concludes the paper with a summary of our findings and future work directions.

II. Methodology

A. System Overview

The experimental setup consists of a hexacopter built on a DJI F550 frame with six DC brushless motors, as shown in Figure 1. The primary flight controller is a Pixhawk 4, equipped with an IMU and a GPS, which is connected to a Raspberry Pi 5 via the RX interface. The Raspberry Pi is responsible for gathering data from additional sensors, including a LiDAR and a stereo camera. The LiDAR provides accurate distance measurements for obstacle detection, while the stereo camera is used for depth perception and visual data, resulting in more precise navigation. The Raspberry Pi processes sensor data and computes the control strategy in real-time, leveraging the Tube-MPC approach. It then transmits command messages to the Pixhawk 4 using the MAVLink protocol. The goal is to achieve fully autonomous control, where the external computer only needs to send an initial goal, after which the UAV navigates independently. However, during the experiments, a master radio command is kept in the loop to ensure safety in case of malfunctions or unexpected behavior. This allows for manual intervention if necessary, providing an additional layer of security during real-world testing.

The system operates with a sampling time of 0.2 seconds, and the real-time data processing on the Raspberry Pi allows for rapid adjustments to the UAV's trajectory. The UAV is powered by a 4-cell Li-ion battery (14.8v). Table I provides the actual values and associated units of the system parameters.

TABLE I
UAV Specifications

Parameter	Value
Type	Hexarotor
Weight	3.37 kg
Height	37.5 cm
Wingspan	55 cm
Rotor radius	2.2 x 1.2 cm
KV	1000

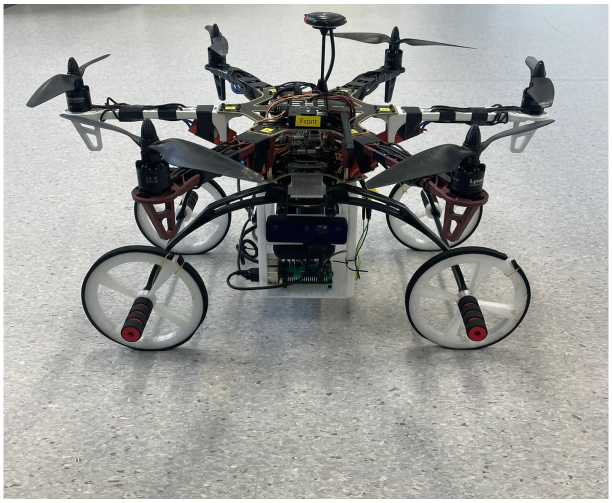

Fig. 1. Experimental setup - Hexacopter equipped with a Pixhawk 4, a GPS, batteries, a camera, a lidar and a Rasberry 5. The main frame used is a DJI550 with additional 3D printed parts.

B. System Identification

The system we aim to control is the closed-loop setup, where the Pixhawk 4 serves as the base controller to stabilize the UAV. Our goal is to enhance the robustness of this setup using Tube-MPC. Real experimental data was gathered by applying multiple-step inputs to the system. The system has four control inputs: the desired speeds along the three axes of the North-East-Down frame (which we associate with the xyz axes) and the yaw angle, measured relative to the z-axis (down axis). The outputs are the corresponding speeds and yaw angle. Figure 2 shows a sample of the data gathered during the step response experiments.

At first glance, linear dynamics predominantly drive the system's behavior. While noise and wind disturbances affect the output responses, the system follows the step inputs in a first-order fashion for the three speeds. However, the yaw angle exhibits slight oscillations. These observations are essential, as they will guide our control design, ensuring that the methods can run efficiently on embedded hardware at high frequencies.

The gathered data was split into training and testing sets for the identification process. We used MATLAB's System Identification Toolbox and applied subspace methods to identify a linear model with four states. Figure 3 shows the validation data of the identified model compared to the experimental data.

Overall, the system's dynamics are well represented. Some overshoots are observed in the speed responses, and oscillations in the yaw response were not fully captured. However, the control methods we will employ, particularly Tube-MPC, are designed to handle disturbances inherently. Therefore, we can consider these slight discrepancies between the model and real-world experiments as additional disturbances that the control strategy will compensate for.

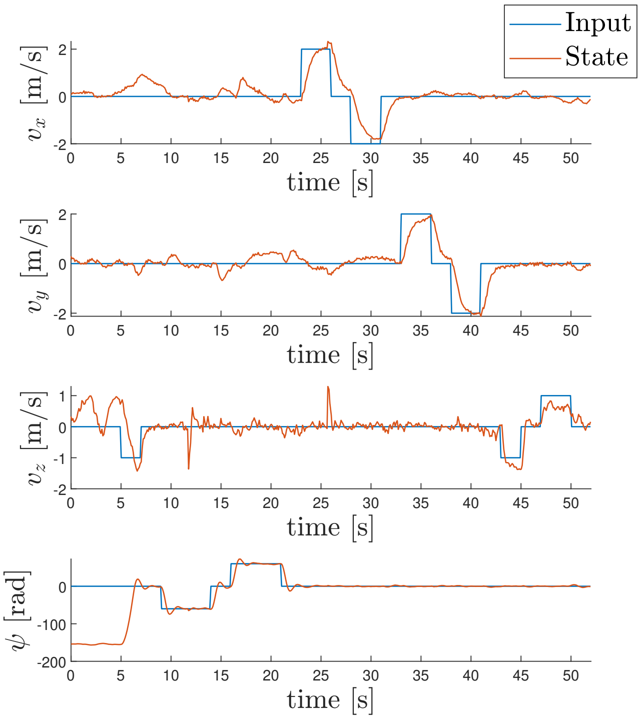

Fig. 2. Data for identification : UAV steps responses for four states : velocities along x, y and z axis and the yaw angle.

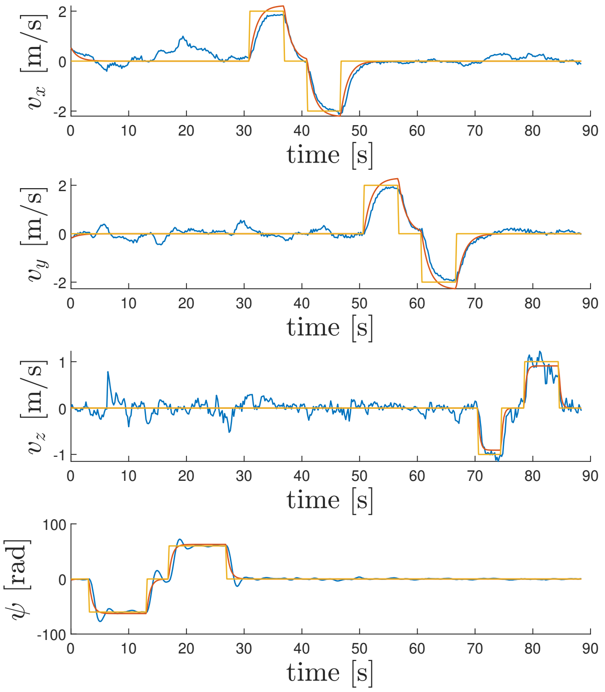

Fig. 3. Validation - Model vs Data - Yellow lines are the input data provided to the system, blue lines are the validation data used to measure the identified system performances shown in orange lines.

C. Tube Model predictive control

The Tube-based Model Predictive Control (Tube MPC) framework [4], [5] is a robust control strategy designed to handle uncertainties and disturbances in dynamic systems by enforcing a bounded deviation between the actual system trajectory and an ideal nominal trajectory. The fundamental concept is to keep the real system, affected by disturbances, within a "tube" that surrounds the optimal trajectory computed for the nominal model. Tube MPC ensures stability and robustness by maintaining the system within predefined limits despite external disturbances.

The system is generally modeled as discrete-time, nonlinear, and time-invariant, affected by disturbances:

$$\vec{x}(k+1) = f(\vec{x}(k), \vec{u}(k), \vec{w}(k)) \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^q \to \mathbb{R}^n$ defines the system's dynamics, $\vec{x}(k) \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector, $\vec{u}(k) \in \mathcal{U} \subseteq \mathbb{R}^r$ represents the control inputs, and $\vec{w}(k) \in \mathcal{W} \subseteq \mathbb{R}^q$ is a disturbance vector bounded by the compact set $\mathcal{W}$, which includes zero. The sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are all assumed to be compact.

For the nominal model, where no disturbances are present, the system evolves as:

$$\vec{\xi}(k+1) = f(\vec{\xi}(k), \vec{\nu}(k), 0) \tag{2}$$

where $\vec{\xi}(k) \in \mathbb{R}^n$ is the nominal state and $\vec{\nu}(k) \in \mathbb{R}^r$ is the nominal control input.

To maintain the actual system trajectory close to the nominal trajectory, an ancillary control law φ is employed, based on the actual state $\vec{x}(k)$, the nominal state $\vec{\xi}(k)$, and the nominal input $\vec{\nu}(k)$:

$$\vec{u}(k) = \varphi(\vec{x}(k), \vec{\xi}(k), \vec{\nu}(k)) \tag{3}$$

This ensures that the error, defined as the difference between the real and nominal trajectories, remains within a bounded region, often referred to as a robust invariant set Γ:

$$\vec{e}(k+1) = \vec{x}(k+1) - \vec{\xi}(k+1) \in \Gamma \tag{4}$$

By applying additional constraints on the nominal system through $\vec{\xi}(k) \in \Xi$ and $\vec{\nu}(k) \in \mathcal{V}$, the size of the "tube" surrounding the nominal trajectory can be modulated, ensuring that the real system adheres to safe operating limits, Fig 4.

The control strategy consists of an open-loop nominal controller that computes optimal trajectories and an ancillary controller that ensures the actual system remains close to these ideal trajectories, even in the presence of disturbances.

1) Implementation of Tube MPC: In the practical implementation, the Tube MPC is designed with a linear MPC for the nominal controller, using an augmented model obtained from the system identification process. The model is augmented by incorporating the dynamics of the position states x, y, and z, achieved by adding three integrators to the system. This augmentation simplifies reference tracking for position inputs, allowing the user to directly set references for x_{ref}, y_{ref}, z_{ref}, and yaw_{ref}. The nominal control is expressed as:

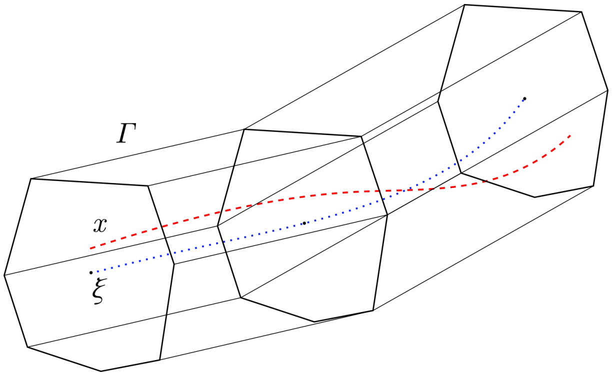

Fig. 4. Tube Model Predictive Control illustration. The nominal system (ξ) travels through its state space (blue line), while the real system (x) stays around the nominal trajectory (red line) within an invariant set (Γ) in zonotopic form (black lines).

$$
\begin{aligned}
\underset{\vec{\nu}}{\text{minimize}} \quad & \sum_{k=1}^{N-1} (\vec{\xi}(k) - \vec{\xi}_{ref}(k))^T Q (\vec{\xi}(k) - \vec{\xi}_{ref}(k)) \\
& + (\vec{\nu}(k) - \vec{\nu}_{ref}(k))^T R (\vec{\nu}(k) - \vec{\nu}_{ref}(k)) \\
& + (\vec{\xi}(N) - \vec{\xi}_{ref}(N))^T F (\vec{\xi}(N) - \vec{\xi}_{ref}(N)) \quad (5) \\
\text{subject to} \quad & \vec{\xi}(k+1) = A_a \vec{\xi}(k) + B_a \vec{\nu}(k), \\
& \vec{\xi}(k) \in \Xi, \quad \vec{\nu}(k) \in \mathcal{V},
\end{aligned}
$$

where $\vec{\xi}_{ref}$ and $\vec{\nu}_{ref}$ are the state and input references way points, respectively. The matrices A_a and B_a represent the linear dynamics of the augmented model. Q, R, and F are weight matrices that adjust the closed-loop behavior, while Ξ and $\mathcal{V}$ are constraint sets that ensure safety and feasibility. N denotes the receding horizon. This study utilizes the do-mpc library [11] in conjunction with the IPOPT solver to address the MPC problem. We expect the computational time to be less than half of the sampling time, given that this will constitute the primary computational burden.

The ancillary controller, responsible for ensuring the real system remains close to the nominal trajectory, is implemented as a state feedback controller. This controller operates based on the identified dynamics without the augmented integrators because the position estimates from the real system are subject to inaccuracies due to the reliance on GPS data. The ancillary controller receives the optimal nominal inputs from the linear MPC and computes the actual control inputs for the system.

The state feedback controller calculates the real control inputs using the following law:

$$\vec{u} = \vec{\nu}^* + K\left(\vec{x} - \vec{\xi}^*\right) \quad (6)$$

where $\vec{\nu}^*$ represents the optimal inputs from the nominal controller, and $\vec{\xi}^*$ represents the optimal velocities and yaw angle. The gain matrix K is designed using the Linear Quadratic Regulator (LQR) method.

This architecture allows the nominal controller to focus on trajectory optimization, while the ancillary controller ensures the real system remains within the desired "tube" around the nominal trajectory. A linear MPC simplifies the optimization problem, while the state feedback controller compensates for the actual system dynamics and disturbances in real time.

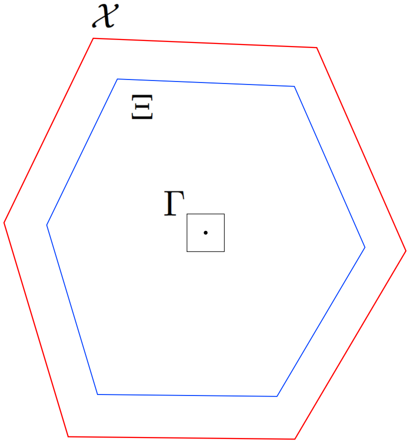

Fig. 5. Illustration of the tightening set for the nominal system. The actual system is constrained by the boundary $\mathcal{X}$ (red). Given the invariant set of the error system Γ (black), by tightening the nominal dynamics within Ξ (blue), we ensure that the disturbed system remains within $\mathcal{X}$ for any disturbances.

D. Constraint Sets

In our Tube MPC formulation, constraints play a crucial role in ensuring both the feasibility and safety of the control strategy. These constraints are applied to the nominal model and account for the disturbances that affect the real system, ensuring that the nominal trajectory remains within a "safe" region. Specifically, we impose state and input constraints on the nominal system:

$$\vec{\xi}(k) \in \Xi, \quad \vec{\nu}(k) \in \mathcal{V} \quad (7)$$

where $\Xi \subseteq \mathbb{R}^n$ is the set of allowable states, and $\mathcal{V} \subseteq \mathbb{R}^r$ represents the set of admissible control inputs. These constraints ensure that the system operates within physical and performance limits, such as actuator saturation and safety boundaries.

To account for disturbances, the constraints are tightened by incorporating a robust invariant set Γ around the nominal trajectory. This tightening ensures that the real system, under disturbances, remains within the safe operating region, Fig 5. Mathematically, we define the tightened constraints as:

$$\Xi = \mathcal{X} \ominus \Gamma, \quad \mathcal{V} = \mathcal{U} \ominus K\Gamma \quad (8)$$

where $\ominus$ denotes Minkowski subtraction, and K is the state feedback gain. The set Γ represents the bounds on the error between the actual and nominal trajectories, ensuring that the real system remains within the desired bounds $\mathcal{X} \subseteq \mathbb{R}^n$ despite disturbances, as well for the control input $\mathcal{U} \subseteq \mathbb{R}^r$.

E. Invariant Set Computation through Reachability Analysis

We employ reachability analysis to compute the robust invariant set Γ. This approach quantifies the set of states the system can reach under all possible disturbances, ensuring that the system stays within this set, even in the worst-case disturbance scenario.

The system dynamics for the error between the actual and nominal trajectories are described as:

$$\vec{e}(k+1) = (A + BK)\vec{e}(k) + \vec{w}(k) \tag{9}$$

where $\vec{e}(k) = \vec{x}(k) - \vec{\xi}(k)$ is the error between the actual state $\vec{x}(k)$ and the nominal state $\vec{\xi}(k)$.

Using reachability analysis [9], we compute the reachable set $\mathcal{R}(N)$ of the error system over a finite horizon N:

$$\mathcal{R}(N) = \bigcup_{k=0}^{N-1} \left((A+BK)^k \mathcal{W} \oplus \sum_{i=0}^{k-1} (A+BK)^i \mathcal{W} \right) \tag{10}$$

where $\oplus$ denotes Minkowski addition. Over a sufficiently long horizon, the set $\mathcal{R}(N)$ converges to the robust invariant set Γ, ensuring the system error remains bounded.

In practice, we approximate the invariant set Γ using iterative methods [12] by leveraging the CORA toolbox [13] and constrained zonotopes [8] for set representation. A constrained zonotope is capable of representing any convex polytope and is defined by a center $\vec{c} \in \mathbb{R}^n$, a generator matrix $G \in \mathbb{R}^{n \times n_g}$, a constraint matrix $A_c \in \mathbb{R}^{n_c \times p}$, and a constraint offset $\vec{b} \in \mathbb{R}^{n_c}$:

$$CZ := \left\{ \vec{c} + \sum_{k=1}^{p} \alpha_k G_{(:,k)} \mid \sum_{k=1}^{p} \alpha_k A_{c(:,k)} = \vec{b}, \quad \alpha_k \in [-1, 1] \right\}$$

By defining a constrained zonotope for initial conditions and another for inputs, reachable sets in the form of constrained zonotopes can be computed.

Once the invariant set Γ is computed, the state and input constraints of the nominal MPC are tightened using the set Γ. The tightened sets, represented by constrained zonotopes, can either be implemented into the MPC by converting them into a half-space (H-rep) representation, i.e., a series of linear equalities and inequalities, or by keeping the generator (G-rep) representation and adding an additional optimization variable (α_k) to the problem [14], [15].

III. Results

The results of the method were validated through a series of experiments. For this purpose, reference steps were applied to the four outputs of the system: x_{ref}, y_{ref}, z_{ref}, and yaw$_{\text{ref}}$. Each step change was introduced sequentially to test the controller's ability to track setpoint changes in real time, while compensating for external disturbances.

Figure 6 illustrates a MATLAB simulation of the system under the Tube-MPC framework, showing the expected output of the system without disturbances.

To assess consistency and robustness, the actual experiment was repeated four times under the same reference steps as in the simulations. The tests were performed outdoors, where the UAV was subject to environmental disturbances like wind.

Figures 7 illustrates the UAV's speed along the x-axis. Despite external disturbances, the UAV consistently followed the path induced by the nominal controller, demonstrating the efficacy of the ancillary feedback controller in compensating for deviations from the nominal trajectory. The Root Mean Squared Error (RMSE) between the nominal trajectories and the system states is computed in Table II. The consistency across the runs highlights the reliability of the proposed control scheme for UAV applications in dynamic environments. A larger error is observed in the yaw angle, as discussed in the identification section.

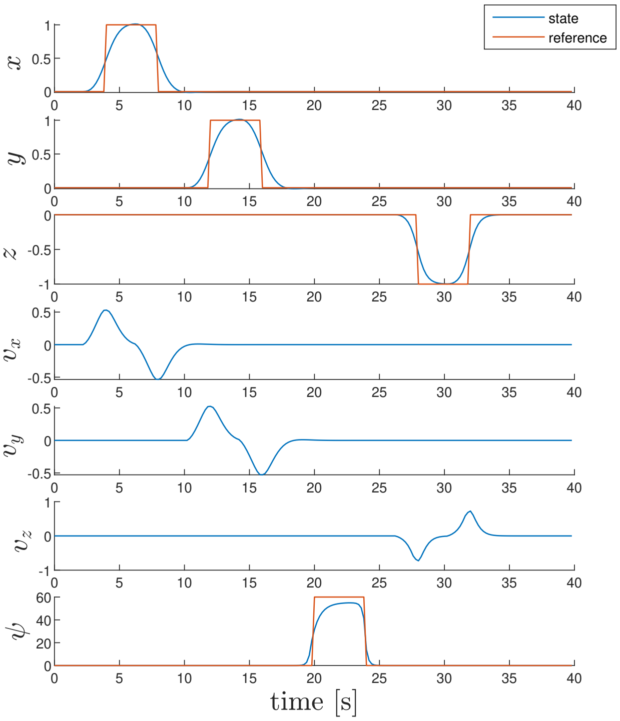

Fig. 6. Simulation of the system without disturbances using the Tube-MPC framework (MATLAB). States are shown in blue, and references are in orange.

TABLE II
RMSE values for different experiments

RMSE	v_x	v_y	v_z	ψ
Exp 1	0.1267	0.1549	0.1078	7.5205
Exp 2	0.1583	0.1317	0.0970	6.0900
Exp 3	0.1334	0.1431	0.0991	6.3427
Exp 4	0.1094	0.1307	0.1106	6.3947

Figure 8 shows the evolution of the nominal states, the states, the nominal inputs, and inputs in the first experiment, with the same references as in simulations. The nominal speed generated by the MPC layer is followed tightly by the ancillary, which aims to drive the system toward the nominal behavior. The main difference appearing is the magnitude of the states, which may come from identification issues, as well as the yaw dynamics, which has been simplified to a one-order system. However, the control can handle disturbances, both wind and dynamic uncertainties.

The average computation time for the entire control loop

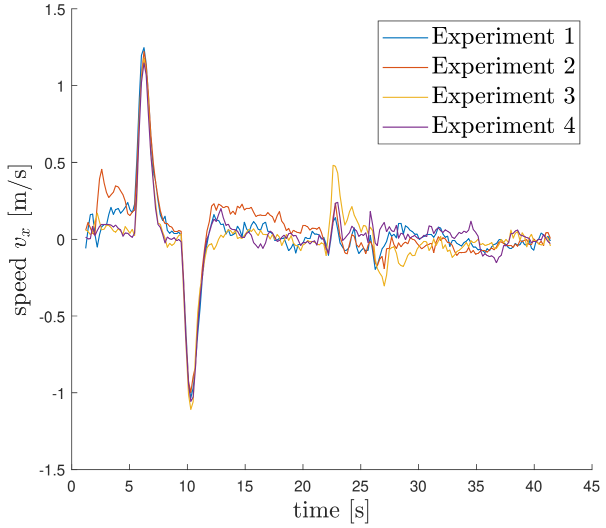

Fig. 7. System speed along the x-axis during the four different experiments.

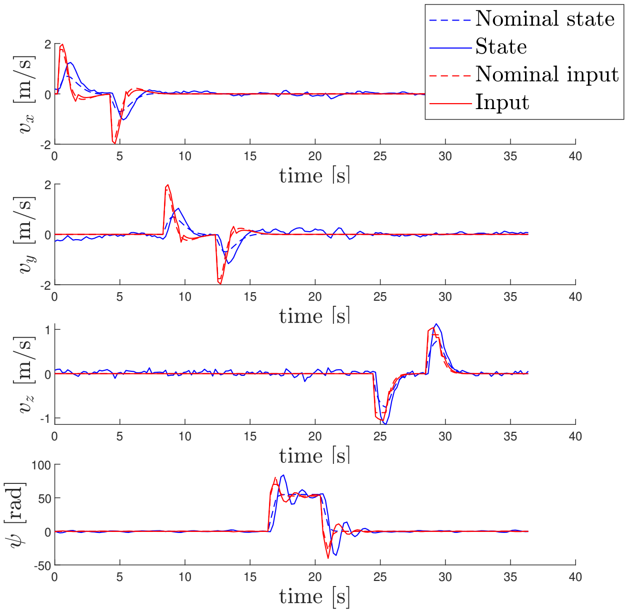

Fig. 8. Data of the first experiment: nominal states are shown as dotted blue lines, while the actual states are shown as solid blue lines. Nominal inputs are represented by dotted red lines, and the actual inputs are shown as solid red lines.

on the Raspberry Pi 5 is 0.052 seconds, with an upper limit observed of 0.077 seconds for a horizon of 10 steps. This respects the imposed sampling time and leaves additional room for other computations.

IV. Conclusion

This work introduces a robust control approach for UAVs under disturbances by combining Tube-MPC and a zonotopic framework with reachability analysis. The integration of zonotopes to compute robust invariant sets ensures that the actual UAV trajectory remains close to the nominal trajectory, despite disturbances. The real-time implementation of the control strategy was demonstrated on a UAV platform using the Pixhawk 4 and Raspberry Pi 5, validating the effectiveness of the method in an outdoor environment subject to external disturbances. The experiments show that the system can achieve robust trajectory tracking, highlighting the strength of the proposed approach in maintaining safe operation and effective disturbance rejection. Future work could explore scaling this approach to more complex UAV missions and obstacle avoidance.

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] F. Nex and F. Remondino, “Uav for 3d mapping applications: a review,” *Applied Geomatics*, vol. 6, no. 1, p. 1–15, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1007/s12518-013-0120-x

[3] C. E. García, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice—a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[4] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, “Tube-based robust nonlinear model predictive control,” *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[5] F. A. Bayer, M. A. Müller, and F. Allgöwer, “Tube-based robust economic model predictive control,” *Journal of Process Control*, vol. 24, no. 8, pp. 1237–1246, 2014, economic nonlinear model predictive control.

[6] P. McMullen, “On zonotopes,” *Transactions of the American Mathematical Society*, vol. 159, no. 0, p. 91–109, 1971. [Online]. Available: http://dx.doi.org/10.1090/S0002-9947-1971-0279689-2

[7] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *Hybrid Systems: Computation and Control*, M. Morari and L. Thiele, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 291–305.

[8] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, “Constrained zonotopes: A new tool for set-based estimation and fault detection,” *Automatica*, vol. 69, pp. 126–136, 2016.

[9] M. Althoff, G. Frehse, and A. Girard, “Set propagation techniques for reachability analysis,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. Volume 4, 2021, pp. 369–395, 2021. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-control-071420-081941

[10] G. Delansnay, A. V. Wouwer, H. G.Harno, and Y. Kim, “Zonotopic reachability analysis of multirotor aircraft,” in *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*, 2021, pp. 576–581.

[11] F. Fiedler, B. Karg, L. Lüken, D. Brandner, M. Heinlein, F. Brabender, and S. Lucia, “do-mpc: Towards fair nonlinear and robust model predictive control,” *Control Engineering Practice*, vol. 140, p. 105676, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066123002459

[12] G. Delansnay, L. Dewasme, and A. Vande Wouwer, “[submitted] tube-based nonlinear model predictive control of multirotor aircraft using a polynomial zonotopic framework,” *book chapter to the Springer volume ”Nonlinear and Constrained Control - Applications, Synergies, Challenges and Opportunities.”*, 2024.

[13] M. Althoff and N. Kochdumper, *CORA 2024 Manual*, Technische Universität München, 87548 Garching, Germany, 2018.

[14] N. Kochdumper and M. Althoff, “Constrained polynomial zonotopes,” *Acta Informatica*, vol. 60, no. 3, p. 279–316, May 2023. [Online]. Available: http://dx.doi.org/10.1007/s00236-023-00437-5

[15] G. Delansnay and A. V. Wouwer, “Design of a reference governor in a zonotopic framework applied to a quadrotor under feedback linearization control strategies,” *Journal of Intelligent; Robotic Systems*, vol. 109, no. 1, Aug. 2023. [Online]. Available: http://dx.doi.org/10.1007/s10846-023-01947-7