Autonomous Navigation of a Quadcopter with Varying Payload Mass

-

Checkpoint 2

Gokul Prabhakaran
Department of Robotics
University of Michigan
Ann Arbor, Michigan, USA
Uniqname: gok

Federico Seghizzi
Department of Robotics
University of Michigan
Ann Arbor, Michigan, USA
Uniqname: seghizzi

I. Introduction

Across various industries, multi-rotor UAVs are slowly replacing helicopters for video-capture applications. Nonetheless, the latter still outperform the former in terms of goods transportation capabilities. In light of the recent development of ultra-high payload multi-rotor UAVs [1–3], this project aims to develop a framework that would enable the autonomous use of these vehicles for transportation and delivery purposes.

This project will focus on two key aspects of the problem: generating optimal trajectories through waypoints for the vehicle to complete its mission, and controlling the UAV's trajectory in spite of disturbances such as wind and varying mass. The perceptive behavior of the vehicle, combined with its ability to generate trajectories in response to it, are not considered as they are highly application-specific.

II. Vehicle Model

A. Vehicle

Throughout this project, a simple quad-rotor UAV with an X-configuration will be considered. This will be assumed to be left-right and forward-backward symmetric, with a payload case of fixed geometry hanging beneath its body. Figure 1 displays the assumed geometry for this quadcopter, along with its system diagram complete with inputs, outputs, and states.

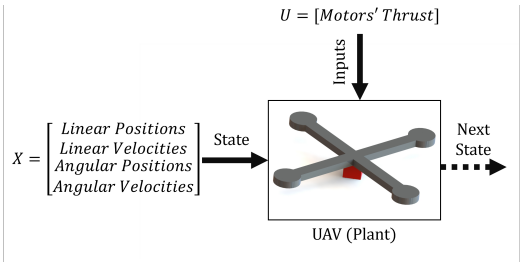

Fig. 1. Block diagram representation of the system, with inputs, outputs, and states.

The essential modeling parameters and constant for such a device are gathered in Table I. Given the focus of the project, the mass of the payload will be considered to vary both continuously and discontinuously as a function of time only; therefore this and the moments of inertia are not reported below.

TABLE I
Constants and parameters used for the quadrotor analysis. Parameters without a reference were derived from CAD software or computations.

Symbol	Name	Value	Units
A_x	Vehicle area projected on the y-z body plane	$7.5e^{-3}$	m^2
A_y	Vehicle area projected on the x-z body plane	$7.5e^{-3}$	m^2
A_z	Vehicle area projected on the x-y body plane	$1.3e^{-2}$	m^2
l	Distance of the rotors from the CoM	0.23 [4]	m
K_{rd}	Rotational drag coefficient (equal for all three axes)	0.1 [4]	Nms
K_{td}	Translational drag coefficient (equal for all three directions)	0.62	kg/m^3
K_m	Moment coefficient for the rotor-prop combination	$2.4e^{-2}$ [4]	m
g	Acceleration from gravity	9.81	m/s^2
F_{Tmin}	Minimum Thrust from the Motor-Propeller Combination	0	N
F_{Tmax}	Maximum Thrust from the Motor-Propeller Combination	$5 \times mg/4$	N
$\dot{F}_{Tmin}$	Minimum Thrust Change Rate from the Motor-Propeller Combination	$-5 \times mg/4$	N/s
$\dot{F}_{Tmax}$	Maximum Thrust Change Rate from the Motor-Propeller Combination	$2.5 \times mg/4$	N/s

B. Reference Systems

Figure 2 shows the adopted frames of reference for the problem. The vehicle frame is chosen such that the positive X axis is aligned with the direction of forward flight, whilst the

positive Z axis is aligned with the direction of thrust from the rotors. The origin of the frame is located in the center of mass (CoM) of the unloaded vehicle. In this frame, the rotational velocity variables reported in Equation 1 are defined.

$$X_{RV} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{1}$$

As for the inertial frame, this is assumed to be tied to a point on Earth's surface, which is assumed to be flat and static. The Z axis points vertically upwards, in an equal but opposite direction to the acceleration due to gravity, whilst the X and Y lie on the plane of the Earth. In this frame, the rotational displacement, translational displacement, and translational velocity variables reported in Equation 2 are defined.

$$X_{TD} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad X_{TV} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad X_{RD} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{2}$$

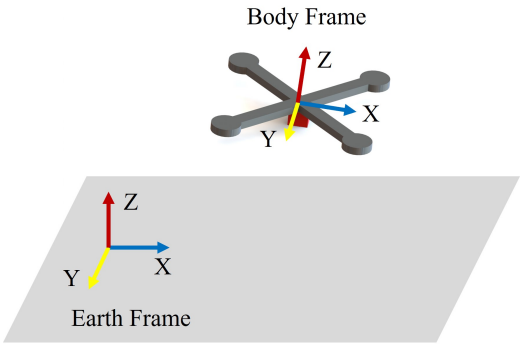

Fig. 2. Representation of the employed reference frames.

C. Equations of Motion

Given the above parameters and frames of reference, the Equations of Motion (EoM) can be written as shown in Equations 3 to 14.

1) Translation kinematic equations:

$$\dot{x} = \dot{x} \tag{3}$$

$$\dot{y} = \dot{y} \tag{4}$$

$$\dot{z} = \dot{z} \tag{5}$$

Where $\dot{x}$, $\dot{y}$ and $\dot{z}$ on the right-hand-side represent the variables in the state vector, whilst those on the left-hand side refer to the derivatives of x, y and z in the state derivative vector.

2) Rotational kinematic equations:

$$\dot{\phi} = p + q(s_\phi t_\theta) + r(c_\phi t_\theta) \tag{6}$$

$$\dot{\theta} = q(c_\phi) - v(s_\phi) \tag{7}$$

$$\dot{\psi} = q(s_\phi sec_\theta) + v(c_\phi sec_\theta) \tag{8}$$

3) Translational dynamics equations:

$$\ddot{x} = \frac{(s_\phi s_\psi + s_\psi s_\theta c_\phi)\sum_{i=1}^{4} F_{T_i} - k_a A_x(\dot{x}_w - \dot{x})^2}{m} \tag{9}$$

$$\ddot{y} = \frac{(-s_\phi c_\psi + s_\psi s_\theta c_\phi)\sum_{i=1}^{4} F_{T_i} - k_a A_y(\dot{y}_w - \dot{y})^2}{m} \tag{10}$$

$$\ddot{z} = \frac{-mg + (c_\phi c_\theta)\sum_{i=1}^{4} F_{T_i} - k_a A_z(\dot{z}_w - \dot{z})^2}{m} \tag{11}$$

Where $\dot{x}_w$, $\dot{y}_w$, and $\dot{z}_w$ are the body frame wind's velocity.

4) Rotational dynamics equations:

$$\dot{p} = \frac{l(-F_{T_1} - F_{T_2} + F_{T_3} + F_{T_4}) - (I_z - Iy)qr - K_{D_x}p}{I_x} \tag{12}$$

$$\dot{q} = \frac{l(-F_{T_1} + F_{T_2} + F_{T_3} - F_{T_4}) - (I_x - I_z)pr - K_{D_y}q}{I_y} \tag{13}$$

$$\dot{r} = \frac{K_m(F_{T_1} - F_{T_2} + F_{T_3} - F_{T_4}) - (I_y - Ix)qp - K_{D_z}r}{I_z} \tag{14}$$

Note that, given the symmetry assumptions made above, the I_{xz}, I_{xy}, and I_{yz} terms are all zero and are not included.

D. Disturbances

Given the considered application, the quadrotor is expected to experience two main disturbances: changes in mass, as the vehicle releases its payload; changes in wind speed, as the vehicle moves through the atmosphere.

The mass will be varied according to Equations 15 and 16, capturing both discontinuous and continuous variations. The associated inertia terms will vary as a function of mass; to simplify the problem, a linear relation between these was experimentally derived in CAD and is reported in Equation 17.

$$m_{total} = \begin{cases} m_{body} + m_{payload}, & \text{if } 0 <= t <= t_{drop} \\ m_{body}, & \text{otherwise} \end{cases} \tag{15}$$

$$m_{total} = m_{body} + k_{mass}t \tag{16}$$

$$\begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} = \begin{bmatrix} I_{x-body} + 8.304e^{-4}\Delta m \\ I_{y-body} + 8.304e^{-4}\Delta m \\ I_{z-body} + 4.075e^{-4}\Delta m \end{bmatrix} \tag{17}$$

Wind velocity will be similarly varied according to Equations 18 and 19. A more realistic wind behavior is also modeled via Equation 20. The relation between these absolute wind velocities and the wind velocities in the X-Y directions (Z direction is not considered due to the low operating altitudes considered [5]) is trivial and defined by the wind direction d.

$$W_{total} = \begin{cases} W_0 + W_{Excess}, & \text{if } t = t_{gust} \\ W_0, & \text{otherwise} \end{cases} \tag{18}$$

$$W = W_0 + k_{wind} * t \tag{19}$$

$$\begin{cases} W = \mathcal{N}(W_0, \frac{W_{excsess}}{3}) \\ d = \mathcal{N}(d_0, \frac{15}{3}) \end{cases} \tag{20}$$

The values considered for the purpose of this project are reported in Table II.

TABLE II
DISTURBANCE VALUES USED FOR THE QUADROTOR ANALYSIS. PARAMETERS WITHOUT A REFERENCE WERE ARBITRARILY CHOSEN.

Symbol	Name	Value	Units
m_{body}	Quadcopter mass w/o payload	0.65 [4]	kg
$m_{payload}$	Payload mass	$1.5m_{body}$ [6]	kg
d_0	Wind direction relative to the X axis of the Earth Frame	-45	deg
W_0	Constant wind speed	7 [7]	m/s
W_{excess}	Additional wind speed	3 [7]	m/s

E. Assumptions

The model reported above rests on a variety of important assumptions:

1) The vehicle is assumed rigid since the deflections it would experience would have a negligible effect on its motion.
2) The vehicle's propellers are assumed rigid since their short span would only result in minimum "flapping".
3) The vehicle is assumed perfectly left-right and forward-backward symmetric, given the numerous quadcopter designs featuring such characteristics.
4) The Earth is assumed flat and a valid inertial reference system, given the comparably small operating range, flight time, and cruising velocities of the quadcopter.
5) The acceleration due to gravity is assumed to be constant and uniform since its variations would have negligible effects on the quadcopter's motion.
6) The density of the atmosphere is assumed constant such that the thrust generated by the motors is constant for a given rotational velocity. This is acceptable given the reduced altitudes quadrotors typically operate in.
7) The motion of the atmosphere relative to the quadcopter is considered uniform at a given point in time. Additionally, it will be assumed that it can be comprehensively expressed by a set of velocity vectors in either the inertial or body frames. This is acceptable given the quadcopter's small scale relative to atmospheric phenomena.
8) The vehicle is assumed to be subject exclusively to aerodynamic drag, gravitational effects, and thrust acting as expressed in Section II.B and II.C.
9) Given the limited roll, pitch, and yaw rates, the gyroscopic effects of the entire quadcopter's rotation are assumed negligible. Furthermore, given the small size and mass of the propellers, the gyroscopic effects of their rotation are also assumed negligible.
10) Given the limited angle of attack the quadcopter would experience for this application, the projected areas associated with aerodynamic drag are assumed constant.
11) The reference frames are assumed to be oriented as expressed in Section II.B.

III. LINEARIZED SYSTEM

Given the nonlinear EoM presented above, the system had to be linearized to simplify the development of an effective controller.

A. Linearization

*1) **Equilibrium Point**:* The equilibrium point selected for linearizing the system was the hover condition. In this condition, state variables are set to 0, and control inputs are set to perfectly compensate for the weight of the UAV:

$$X_{eq} = 0^{12\times 1}, \quad U_{eq} = \frac{mg}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{21}$$

This condition is the trivial equilibrium condition, as it can easily be seen that it will ensure all EoM presented in Section II.C go to zero. Whilst other equilibrium points may exist, in practice this condition is used even when developing complex controllers [4, 8]. Alternative equilibria were thus not considered.

*2) **Derived System**:* Given the above equilibrium condition, the system was linearized by simply taking the jacobians of the EoM with respect to the state vector and the control vector, and evaluating said jacobians with the values in Equation 21. This produced the system system reported in Equation 22, with the A and B matrices defined as in Equation 23 and 24.

$$\dot{X} = A \cdot X + B \cdot U \tag{22}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.34m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -6.34m & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-0.1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-0.1}{I_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-0.1}{I_z} \end{bmatrix} \tag{23}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{2.52}{m} & \frac{2.52}{m} & \frac{2.52}{m} & \frac{2.52}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{-0.58}{Ix} & \frac{-0.58}{Ix} & \frac{0.58}{Ix} & \frac{0.58}{Ix} \\ \frac{-0.58}{Iy} & \frac{0.58}{Iy} & \frac{-0.58}{Iy} & \frac{0.58}{Iy} \\ \frac{6.03e^{-2}}{Iz} & \frac{-6.03e^{-2}}{Iz} & \frac{6.03e^{-2}}{Iz} & \frac{-6.03e^{-2}}{Iz} \end{bmatrix} \tag{24}$$

B. Performance Assessment

*1) **Non-Linear vs Linear System Comparison**:* To evaluate whether the linearized system reported in Equation 22 performed similarly to its non-linearized counterpart in Equations 3 to 14, the two were simulated assuming no payload mass or payload mass variation, and no wind. The initial conditions and fixed inputs employed for these simulations are reported in Table III. Whilst limited for most quadcopters, these maneuvers are what a transport/delivery UAV strictly needs to perform to successfully complete its mission, and are therefore deemed appropriate given the scope of the project.

TABLE III
OPEN-LOOP INITIAL CONDITIONS AND CONTROL INPUTS FOR COMMON QUADCOPTER MANEUVERS. ONLY NON-ZERO INITIAL CONDITIONS FOR THE STATE VECTOR ARE SPECIFIED.

Description	$\mathbf{X(0)}$	$\mathbf{U}$
Static Hover	All Zeros	$u_{1,2,3,4} = 1.5846N$
Steady Ascent	$w = 1m/s$	$u_{1,2,3,4} = 1.5915N$
Steady Straight Flight	$u = 2m/s$, $w = 7.8e^{-3}m/s$, $\theta = 0.22°$	$u_{1,2,3,4} = 1.5845N$

Results for the open-loop simulations are presented in Figures 3 to 5, where each state variable is plotted against time for both models.

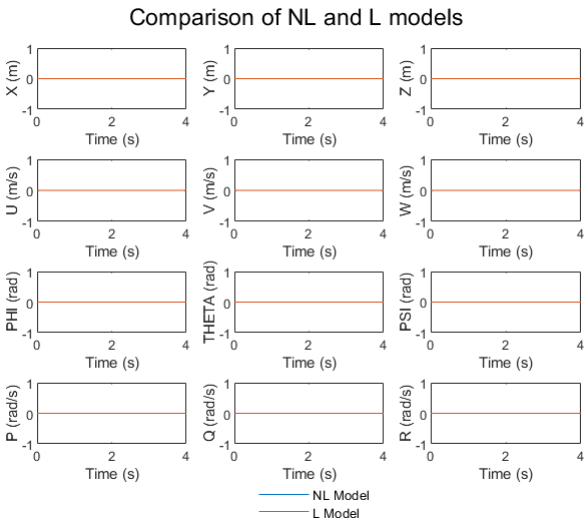

Fig. 3. Comparison between Linear and Non-Linear Quadcopter models for a hover condition.

As can be seen, the linearized model can somewhat adequately predict the performance of the non-linearized model. Specifically, the linearized model's states are at most $\sim 5\%$ off from their non-linearized counterpart for every second of simulation. Whilst this could become very significant over long maneuvers, it is an inherent and unavoidable limitation of the linearization of the numerous square velocities terms within the system. Overall, however, since the maneuvers and limited velocities needed for a transport or delivery-focused quadcopter are unlikely to meaningfully differ from those presented above, the model is likely acceptable for this project.

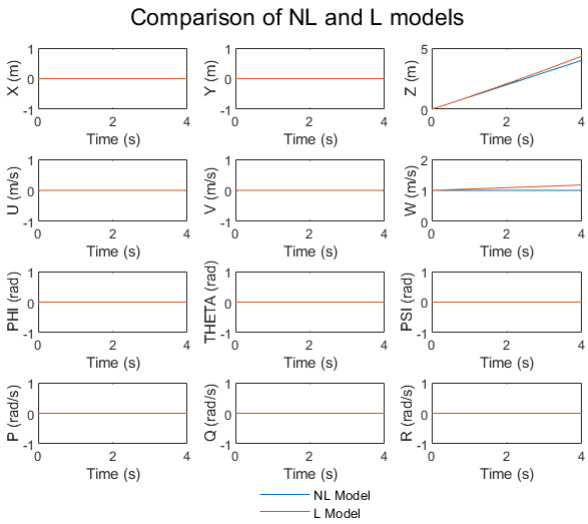

Fig. 4. Comparison between Linear and Non-Linear Quadcopter models for a steady-state ascent condition.

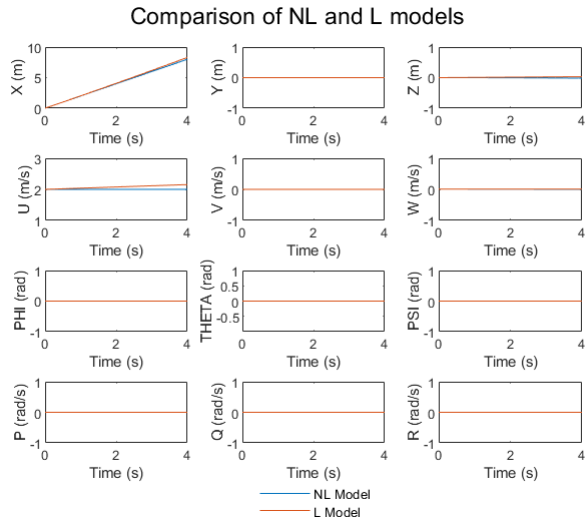

Fig. 5. Comparison between Linear and Non-Linear Quadcopter models for a steady straight flight condition.

*2) **Open-Loop Stability Analysis**:* Given the viability of the above-presented linearized system, its stability had to be evaluated. This was done by evaluating the eigenvalues of the A matrix, as reported in Equations 25 to 28. The corresponding eigenvectors are impractical to report and will be thus simply described qualitatively.

$$\lambda_1 = \frac{-0.1}{I_x} \tag{25}$$

$$\lambda_2 = \frac{-0.1}{I_y} \tag{26}$$

$$\lambda_2 = \frac{-0.1}{I_z} \tag{27}$$

$$\lambda_{4-12} = 0 \tag{28}$$

Considering each eigenvalue:

- λ_1: This eigenvalue will always have a real negative value, and represents stable non-oscillatory behavior. The corresponding eigenvector has terms associated primarily with ϕ and p, and meaningfully smaller terms associated with y and $\dot{y}$. This mode clearly highlights the coupling between these states, and disturbances along them will die down with the system eventually returning to an equilibrium condition, albeit perhaps with a steady-state error.
- λ_2: This eigenvalue presents analogous behavior to λ_1. The corresponding eigenvector features terms associated with θ, q, x, and $\dot{x}$, coherently with the theoretically expected coupling between these states.
- λ_3: This eigenvalue will also always be a pure negative number, and will consequently also feature stable behavior. Its eigenvector however only presents terms for the ψ and r states, due to reduced coupling with the rest of the system.
- λ_{4-12}: The eigenvalue of 0 is repeated nine times, and represents critically stable behavior. If the system is disturbed along the specific or generalized eigenvectors, it will simply settle at the disturbed value.

Based on the above, it can be said that the system is at best critically stable.

*3) **Controllability Analysis**:* To develop a controller that compensates for this critically stable behavior, the controllability of the system must be considered. This was evaluated through a Popov–Belevitch–Hautus (PBH) Test [9], which confirmed that all 12 modes are controllable, suggesting that the system is fully controllable.

IV. Controlled System

A. Controller Design

The controller developed for the quadrotor is a Model Predictive Controller (MPC). This was chosen based on its effective ability to reject disturbances whilst being able to follow trajectories specifying both displacements and velocities even for dynamically complex systems [10], with constraints. Two variants of this controller have been implemented. Firstly, a Linear MPC (L-MPC) relying on a discretized and linearized model. Secondly, a Non-Linearized MPC (NL-MPC), capturing the entire dynamics of the system presented in Section II, albeit at the cost of a superior computational complexity.

*1) **Linear MPC**:* To develop the L-MPC, the system had to first be converted to discrete-time using the 4-term Runge-Kutta (RK4) method. This was then linearized following the methodology and conditions highlighted in Section III.

Considering the controller itself, MPC is a trajectory-following algorithm that minimizes a specified objective function while conforming to any specified constraints, including dynamics constraints. At every time step k, the L-MPC optimizes a sequence of rotor thrust inputs as shown in Equations 29 and 30.

$$\mathbf{u}(k) = [\mathbf{u}(k|k) \dots \mathbf{u}(k+h-1|k)]^T \tag{29}$$

$$\mathbf{x}_{seq}(k) = [\mathbf{x}(k|k) \dots \mathbf{x}(k+h|k)]^T \tag{30}$$

In the equations above, h is the simulation horizon, $\mathbf{u}(k)$ is the candidate control sequence computed by the L-MPC at time step k, and $\mathbf{x}_{seq}(k)$ is the corresponding state sequence at that same time step k. The first control input $\mathbf{u}(k|k)$ in the sequence $\mathbf{u}(k)$, is applied to the system, altering the state from $\mathbf{x}(k)$ to $\mathbf{x}(k+1)$. The L-MPC step is performed again at step $k+1$ and the process continues until the quadrotor reaches the end of the specified reference trajectory.

The control sequence $\mathbf{u}(k)$ is obtained by performing an optimization with a quadratic cost function given in equation 31 with linear constraints. To solve this convex problem, the CVX [11] library was used within MATLAB to compute the global minimum.

$$\begin{aligned} &\min_{\Delta x_{1:h}, \Delta u_{1:h-1}} \\ &\sum_{i=1}^{h-1} \Big[(\mathbf{x}(k+i|k) - \mathbf{x}_{i,ref})^T Q (\mathbf{x}(k+i|k) - \mathbf{x}_{i,ref}) \\ &\quad + (\mathbf{u}(k+i|k) - \mathbf{u}_{i,ref})^T R (\mathbf{u}(k+i|k) - \mathbf{u}_{i,ref}) \Big] \end{aligned} \tag{31}$$

The cost function in Equation 31 minimizes the error between the reference trajectory and the state and control variables to be optimized, $\mathbf{x}(k)$ and $\mathbf{u}(k)$. By minimizing the difference between the reference states and the simulated states, the quadcopter will follow the trajectory as best as possible. The diagonal weight matrix Q specifies the relative importance of tracking each state variable.

Similarly, by minimizing the difference between the reference control inputs and the simulated control inputs, the quadcopter will fly as close to the critically stable hover condition as possible. The R vector specifies the relative importance of tracking each control input.

When performing the minimization via Equation 31, linear constraints are imposed to render the model convex. Firstly, the discrete-time linearized system is imposed as a linear constraint to ensure that the optimized control inputs will comply with the drone dynamics. Secondly, the motor-propeller's maximum thrust and thrust change are enforced via simple inequalities.

In this implementation, the mass of the system and the wind velocities are varied with time, based on Equations 15 through 20; these are assumed to be known perfectly. Instead of modeling these changes as disturbances, the mass, inertial matrix, and wind velocities are directly expressed in the quadrotor's state equation. This results in a discrete-time-varying linearized dynamics equation, with $A(t)$ and $B(t)$ matrices needing to be recomputed at each time step t.

*2) **Non-Linear Time-Varying MPC**:* The L-MPC can employ only quadratic cost functions, linear dynamic models, and linear constraints. Given the highly non-linear nature of the model considered and the risk of its linearization not adequately capturing its behavior, a Non-Linear MPC (NL-MPC) was thus also developed.

NL-MPC solves a non-linear optimization problem with non-linear problem constraints, objectives, or dynamics equations. In this NL-MPC implementation, the non-linear drone dynamics equations were used, whilst the cost function and constraints were set to match those of the L-MPC case. This modification required the use of a Sequential Quadratic Program (SQP) to complete the optimization and find a local minima to the non-linear problem [12, 13].

As for the L-MPC, the state equations would change through time due to the presence of varying mass, inertia, and wind, requiring the reformulation of the NL-MPC at every time step.

B. Controller Tuning

*1) **Non-Tuned Parameters**:* Not all of the tunable parameters reported in Section IV.A underwent a tuning procedure. Specifically, the weight matrices Q and R, and the time interval ΔT were simply set as expressed in Equation 32.

$$Q = I^{12\times12}, \quad R = 0.1I^{1\times4}, \quad \Delta t = 0.05 \tag{32}$$

Considering first the Q matrix, this is set to identity to ensure each of the state variables of the system has equal weight. Specifically, the translational variables are important given how they define the trajectory the quadrotor should follow; the rotational ones, on the other hand, incentivize the controller to keep the designated orientation, set to be level throughout the entire trajectory.

The R matrix instead presents all control input weights set to 0.1. The equal weighting is sensible given how the quadcopter is assumed to be fully symmetric and with four identical motors. The magnitude is chosen to be one order of magnitude smaller than that of the state weights to ensure that, whilst optimal control inputs are chosen, they do not come at the cost of trajectory tracking.

Finally, the time interval ΔT is set to 0.05 to emulate the sampling rate of the slowest sensor a quadcopter would be expected to feature: a GPS [14].

*2) **Tuned Parameters**:* The variables that were tuned were thus only the simulation horizon h for the L-MPC, and the prediction and control horizons p and c for the NL-MPC. These were tuned following a brute-force method: p and h were both varied between 5 and 25 in increments of 5, whilst c was varied between 2 and 5 in increments of 1. The extremes of the latter were chosen based on MATLAB's recommended defaults [12]; the former were instead based on literature values [10].

Each possible value (or combination of values in the case of NL-MPC) was evaluated through four test cases, summarized in Table IV.

TABLE IV
TEST-CASES FOR CONTROLLER TUNING AND STABILITY EVALUATION

Test ID	Description	Physical Significance	Disturbance Equation
A	Wind Gust	Step-Input for x, y, $\dot{x}$ and $\dot{y}$	18
B	Discrete Mass Drop	Step-Input for z and $\dot{z}$	15
C	Increasing Wind	Ramp-Input for x, y, $\dot{x}$ and $\dot{y}$	19
D	Continuos Mass Drop	Ramp-Input for z and $\dot{z}$	16

To evaluate the differently tuned controllers' ability to compensate for these ramp and step inputs, the weighted error metric ϵ reported in Equation 33 was used.

$$\epsilon = \frac{E^{0-1}_{mean} + E^{0-1}_{min} + E^{0-1}_{max} + 3T^{0-1}_{comp}}{6} \tag{33}$$

Where E^{0-1}_{mean}, E^{0-1}_{min}, and E^{0-1}_{max} are the mean, minimum, and maximum of the 2-norms taken between the desired and actual trajectories at every time step, scaled to be within 0 and 1. Note that the 2-norm is taken only between translational variables, as the orientation is not relevant for the considered application. T^{0-1}_{comp} is the computing time required to complete the entire simulation, also scaled between 0 and 1. The weight of the latter term is set to match the cumulative weight of the other three, to penalize diminishing improvements attainable with ever-longer horizons.

Figures 6 and 7 report the average ϵ for the four test cases at each possible tuning.

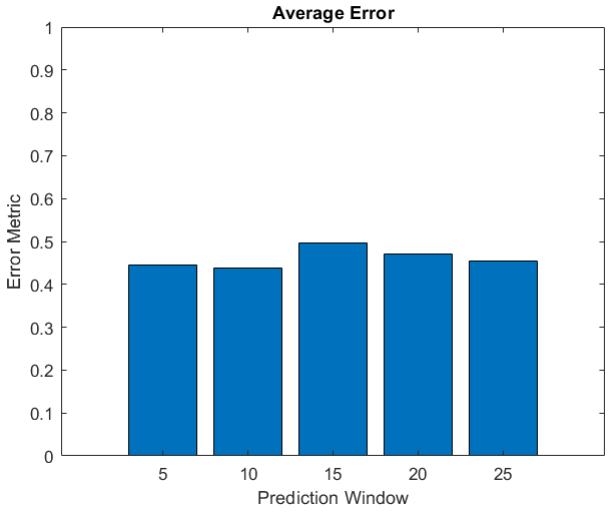

Fig. 6. Bar-graph displaying the average ϵ across the four tests for the tuning procedure for the L-MPC.

In the L-MPC case, one can see how increasing the prediction window has little effect; this indicates that any improvements in the system's performance are compensated evenly by increases in computational time. Indeed the ϵ values are all within $\pm2.5\%$ of one another, and consequently any h

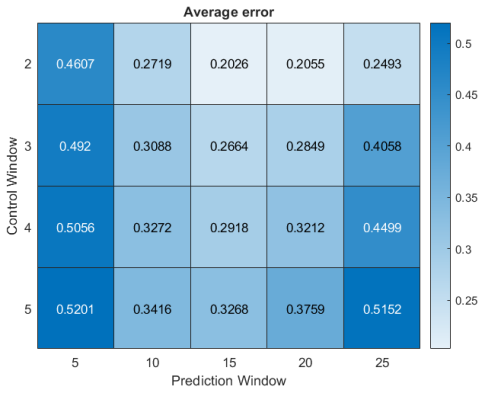

Fig. 7. Heatmap displaying the average ϵ across the four tests for the tuning procedure for the NL-MPC.

could be selected; for the purpose of this project, the value of $h = 25$ is used. On the other hand, in the NL-MPC system, p and c have meaningful effects on the performance of the system, and it is clear that optimal performance is achieved with $p = 15$ and $c = 2$. This ensures the algorithm needs to solve a comparably simple optimization problem, whilst incentivising aggressive response to disturbances and adequately capturing the system dynamics over an extended period of time.

C. Performance Assesment

Whilst the preliminary analysis reported above would appear to suggest that the NL-MPC system performs better than the L-MPC one, this is not accurate: note that the ϵ metric reported above is a normalized metric, unsuitable for such comparisons. To effectively select the optimal controller, two different analyses must be performed: one focussing on the stability of the system, and another addressing the system's ability to track a relevant trajectory globally.

1) ***Stability & Disturbance Rejection****:* The stability of the system is closely related to its ability to reject disturbances. Given the complex controllers employed, a root-locus plot could not be evaluated. Instead, the system's performance is evaluated via the four test cases highlighted in Section IV.B. Figures 8 reports the relevant step- and ramp-response plots.

Comparing the two controllers, these share similar responses to the simulated disturbances: both compensate for errors in velocity more effectively than they do for errors in displacement, with the latter presenting larger steady-state errors (SSE) when compared to the former. This is expected: since the velocity throughout the trajectory should be constant, the controllers try to amend it as soon as possible to minimize its error contribution through the simulation horizon; conversely, the position in the X and Y varies continuously, and the controller can sacrifice short-term position offsets as long as the quadrotor is moving in the overall general direction throughout the considered horizon. This is confirmed by the fact that Z's SSE is smaller than X and Y's since this is also set to a constant value. Both controllers additionally compensate for step disturbances more rapidly than they do for ramp disturbances, although the latter present lower maximum errors than the former do. This is also coherent with theory: at each time interval, the disturbance to be corrected is smaller than that of a large step, making it easier for the controllers to compensate it; the overall length of this disturbance does however last longer than the step disturbance, requiring more time to compensate.

Irrespective of these similarities, however, it is clear that the NL-MPC is more effective at rejecting disturbances than its linear counterpart. Specifically, the NL-MPC presents a lower SSE, maximum error, and settling time than the L-MPC for all the test cases considered.

2) ***Trajectory Tracking****:* To evaluate the controller's ability to track a trajectory, two relevant scenarios were considered: a delivery mission embodied by Equation 15, where a discrete payload is released at a designated location, and a dissemination mission represented by Equation 16, where the payload is released continuously during flight. The latter represents applications involving fire suppression in the outdoors or fertilization of fields; the former represents applications such as parcel deliveries or humanitarian aid provision. Throughout both missions, the drone will fly through the waypoints listed in Table V following the trajectory reported in Figure 9; this was generated to mimic the flight paths of a delivery quadcopter moving from a launch point to a landing point, and releasing its payload at altitude. Note that orientations of the quadrotor are all set to 0 throughout the trajectory and hence are omitted from the figure. The disturbances due to wind followed Equation 20.

TABLE V
POSITION AND VELOCITIES DEFYING THE WAYPOINTS. ALL UNITS IN S.I.

WP #	**(X, Y, Z)**	**($\dot{X}$, $\dot{Y}$, $\dot{Z}$) Delivery**	**($\dot{X}$, $\dot{Y}$, $\dot{Z}$) Dissemination**
1	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
2	(0, 0, 30)	(0, 0, 0)	(0, 0, 0)
3	(150, 150, 30)	(20, 20, 0)	(20, 20, 0)
4	(300, 300, 30)	(0, 0, 0)	(20, 20, 0)
5	(450, 450, 30)	(20, 20, 0)	(20, 20, 0)
6	(600, 600, 30)	(0, 0, 0)	(0, 0, 0)
7	(600, 600, 0)	(0, 0, 0)	(0, 0, 0)

The quadrotor's ability to globally follow the designated trajectory will be evaluated through the 2-norm taken between this and the actually performed trajectory at every time step. Data for these results is reported in Figure 10. The ability to reach designated waypoints will instead be evaluated through the minimum 2-norm taken between each waypoint and the entire trajectory; this data is visualized in Figure 11. For both metrics, only the translational state variables are considered, since orientation is irrelevant for the application considered. Additionally, in this assessment, the 2-norm for the displacement states (E_{xyz}) will be taken separately from the 2-norm for the velocities ($E_{\dot{x}\dot{y}\dot{z}}$).

To assess whether a quadrotor has arrived at a waypoint, the following criteria are considered. Position-wise, it should be within 1m of its designated position when landing, taking

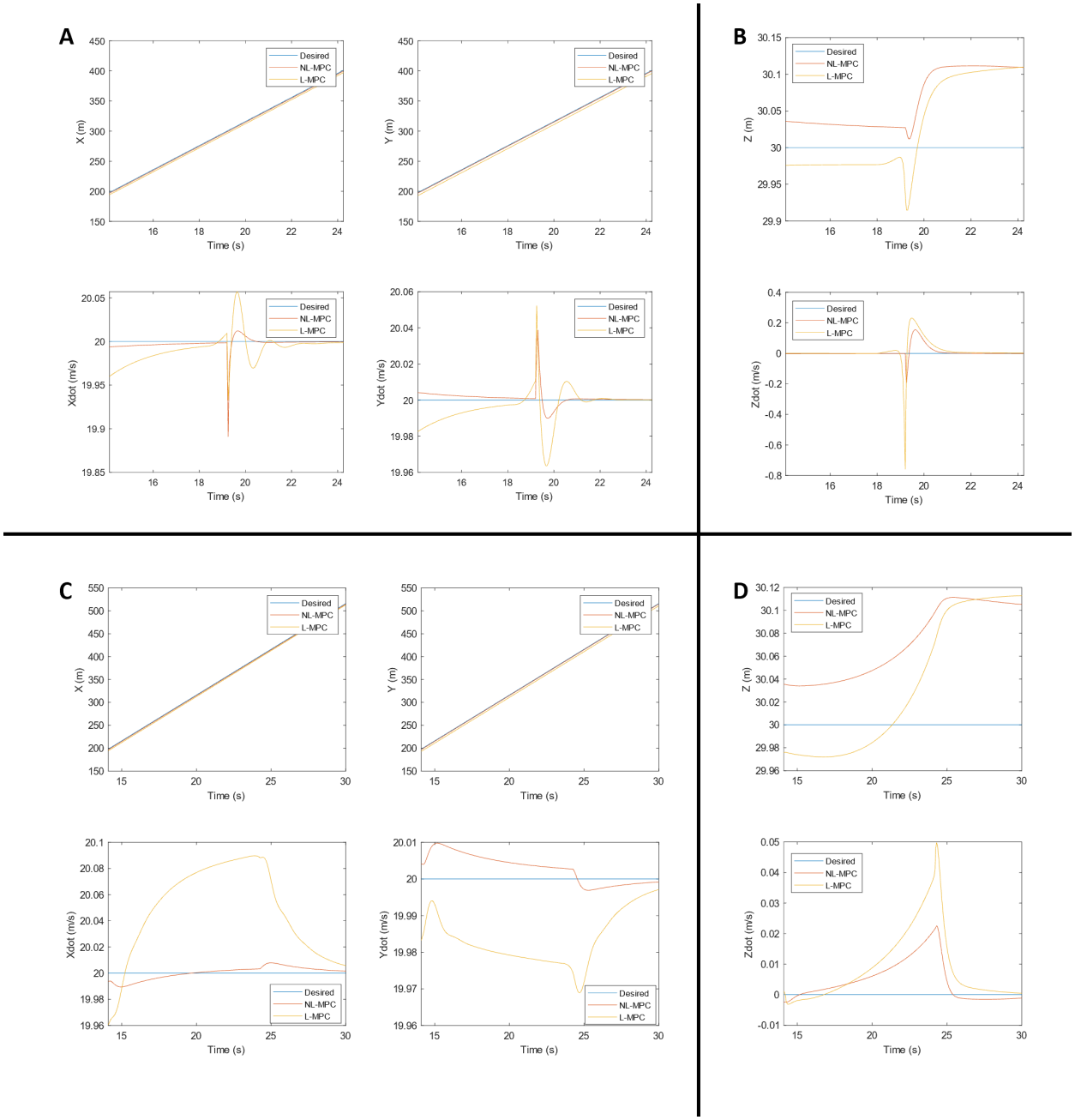

Fig. 8. Step- and ramp- responses for the L-MPC and NL-MPC. The letter labels correspond to the "Test ID" from Table IV. Note that plots for each test case are reported only for the state variables primarily affected by the disturbance.

off, or releasing its payload, or within 5m when flying. The latter limit is representative of the ample space the quadcopter can fly through when operating at its cruising altitude, whilst the former was chosen to reflect the more constrained flight envelope it must follow when closer to the ground or releasing its cargo. In terms of velocity, the quadrotor should be within 0.1m/s of its designated velocity when landing, taking off, or performing a payload release operation, or within 0.5m/s of its designated velocity during normal operations. As before, the choice of a tighter limit for special operations reflects the superior precision requirements needed for their successful operations. Figure 11's color coding reflects whether these limits were met for each waypoint.

The data reported in Figure 10 clearly indicates that, whilst the NL-MPC can adequately track the designated trajectory, the same is not true for the L-MPC. Indeed, the average error of L-MPC throughout the delivery mission are 6m and 2.68m/s, and 5.10m and 1.73m/s for the dissemination mission; conversely, NL-MPC presents 1.89m and 0.77m/s, and 1.77m and 0.49m/s for the same missions. Irrespective of this, it is important to note that even the NL-MPC presents high errors when performing rapid changes in direction, such

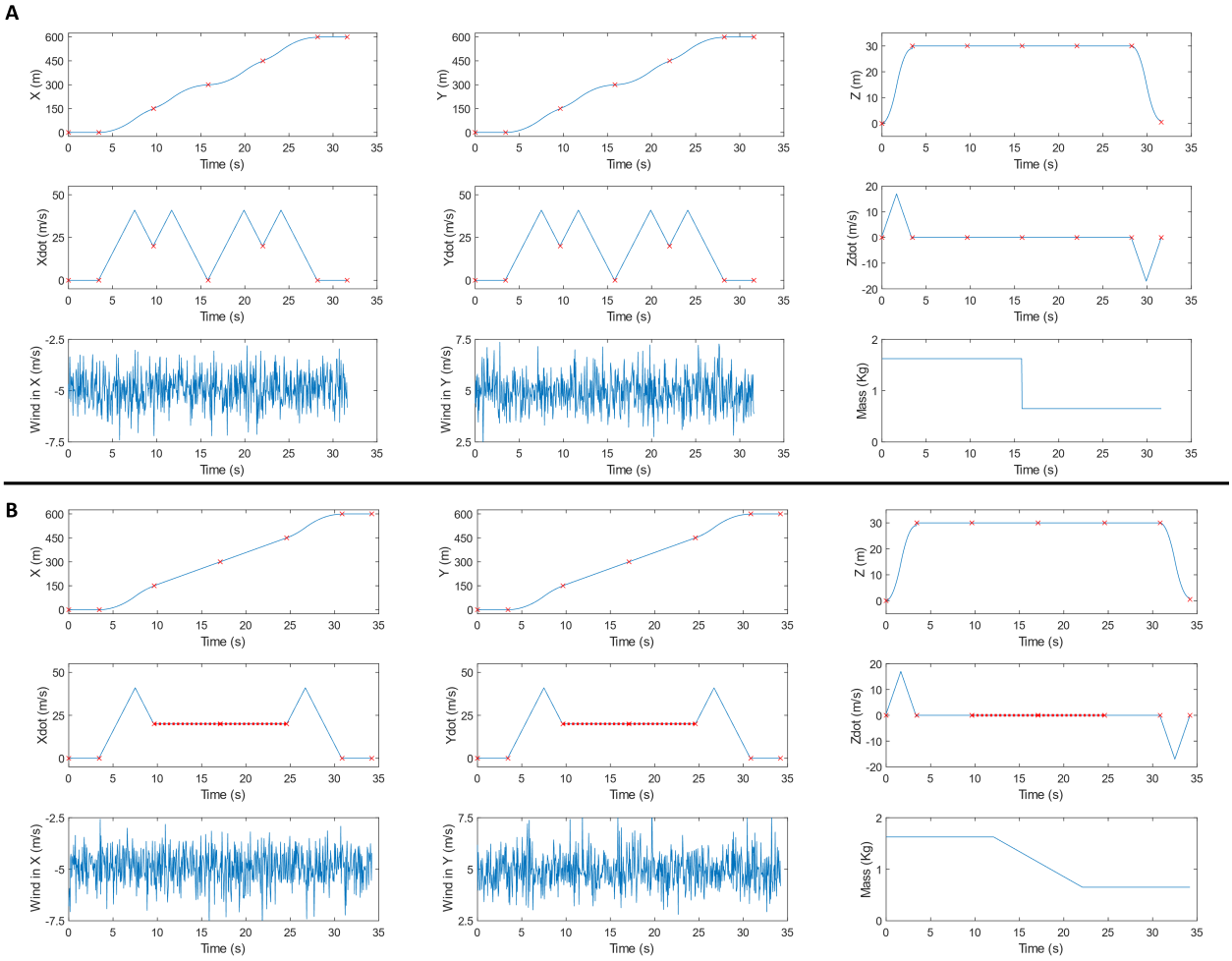

Fig. 9. Trajectory, mass disturbances, and wind variation for a quadrotor. The red "x" indicate the specified waypoints and corresponding velocities. The dashed red line indicates a constant velocity enforcement. **A**: Delivery mission with payload deployment at altitude; **B**: Dissemination mission with payload deployment at altitude.

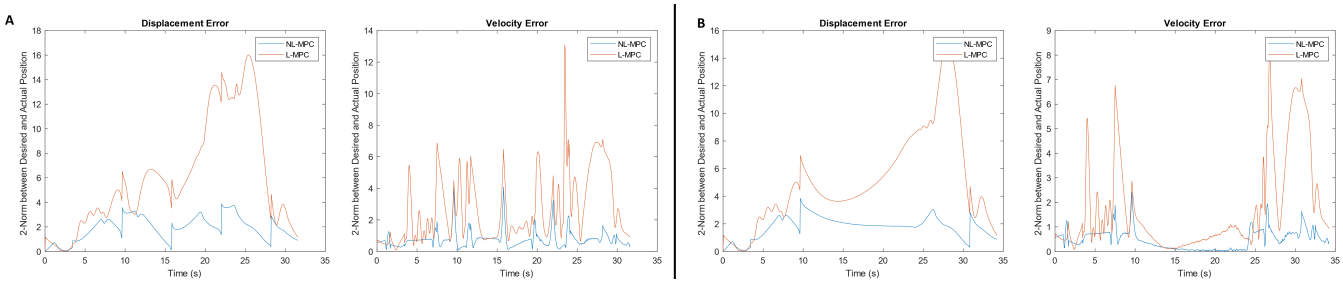

Fig. 10. Error between commanded and desired displacement and velocity throughout the mission. **A**: Delivery mission with payload deployment at altitude; **B**: Dissemination mission with payload deployment at altitude.

as for waypoints 2 and 6, or in responses to meaningful disturbances, such as instantaneously after the release of the discrete payload.

The NL-MPC outperforms its linear counterpart also when driving the drone to designated waypoints at designated velocities; specifically, the L-MPC fails to maintain the velocity within the established tolerance and does not consistently meet the more stringent position requirements applied during landing. Conversely, the NL-MPC maintains the quadcopter well within tolerance at all waypoints; the only waypoint where it appears to struggle is #6, where the drone must rapidly decelerate to a stop and then begin descent. This performance is nontheless still within the selected limits, and most importantly does not carry over to following waypoints.

A

WP #	L-MPC Displacement	L-MPC Velocity	NL-MPC Displacement	NL-MPC Velocity
1★	0.01	0.16	0.00	0.07
2	0.59	0.21	0.84	0.34
3	1.03	0.63	0.60	0.06
4★	0.27	0.21	0.44	0.34
5	2.25	0.63	0.91	0.06
6	2.72	0.50	2.54	0.31
7★	1.12	0.00	0.90	0.00

B

WP #	L-MPC Displacement	L-MPC Velocity	NL-MPC Displacement	NL-MPC Velocity
1★	0.01	0.16	0.00	0.07
2	0.59	0.21	0.85	0.34
3★	1.15	0.11	0.66	0.00
4★	0.29	0.11	0.56	0.00
5★	2.03	0.11	0.60	0.00
6	2.89	0.50	2.43	0.32
7★	1.15	0.00	0.86	0.00

Fig. 11. Error between commanded and desired displacement and velocity at each waypoint. The color-coding reflects whether the error is within the designated tolerance. The star next to the waypoint indicate whether the quadrotor is taking off/landing/releasing its payload; for these rows, tighter tollerances are employed. **A**: Delivery mission with payload deployment at altitude; **B**: Dissemination mission with payload deployment at altitude.

Based on all the above, it is thus clear that L-MPC is inadequate for the application considered. Conversely, the developed NL-MPC can effectively be used to control the quadcopter during both delivery and dissemination missions, along realistic trajectories several hundreds of meters long, whilst being disturbed by wind with varying intensity and direction coherent with what would be expected at the relevant altitudes of operation.

REFERENCES

[1] Griff aviation. https://www.griffaviation.com/. Griff Aviation. Accessed October 2, 2023.

[2] D. Inc. Draganfly inc. - heavy lift uav. https://draganfly.com/products/heavy-lift/. Draganfly Inc. Accessed October 2, 2023.

[3] Freefly systems - alta x. https://freeflysystems.com/alta-x. Freefly Systems. Accessed October 2, 2023.

[4] M. Islam, M. Okasha, and M. M. Idres, "Dynamics and control of quadcopter using linear model predictive control approach," *IOP Conference Series: Materials Science and Engineering*, vol. 270, p. 012007, 2017.

[5] N. C. for Environmental Information. Wind monitoring. National Centers for Environmental Information. Accessed: November 7, 2023. [Online]. Available: https://www.ncei.noaa.gov/access/monitoring/wind/

[6] F. Systems. Freefly alta x drone specifications. Freefly Systems. Accessed: November 7, 2023. [Online]. Available: https://freeflysystems.com/alta-x/specs

[7] U. D. of Energy. U.s. wind turbine database. U.S. Department of Energy. Accessed: November 7, 2023. [Online]. Available: https://windexchange.energy.gov/maps-data/325

[8] R. Kumar, M. Bhargavapuri, A. M. Deshpande, S. Sridhar, K. Cohen, and M. Kumar, "Quaternion feedback based autonomous control of a quadcopter uav with thrust vectoring rotors," in *2020 American Control Conference (ACC)*, 2020, pp. 3828–3833.

[9] B. Gillespie. (2020) PBH Tests - ME564 - Linear Systems Theory. University of Michigan. [Online]. Available: https://umich.instructure.com/courses/548697/external_tools/53084

[10] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process dynamics and control*. John Wiley & Sons, 2016.

[11] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[12] MathWorks. Nonlinear model predictive control (nl mpc). MathWorks. Accessed:November 7, 2023. [Online]. Available: https://www.mathworks.com/help/mpc/ref/nlmpc.html

[13] K. Schittkowski and Y.-X. Yuan, *Sequential Quadratic Programming Methods*. John Wiley Sons, Ltd, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470400531.eorms0984

[14] V. Automotive. Gps accuracy. MathWorks. Accessed: November 7, 2023. [Online]. Available: https://www.vboxautomotive.co.uk/index.php/en/component/content/article/2-products/75-gps-accuracy