Stability Guarantees for Translational Thrust-Propelled Vehicles Dynamics Through NMPC Designs

Ngoc Thinh Nguyen, Ionela Prodan, Laurent Lefèvre

Stability guarantees for translational thrust-propelled vehicles dynamics through NMPC designs

Ngoc Thinh Nguyen, Ionela Prodan and Laurent Lefèvre

***Abstract*—We propose two different NMPC (Nonlinear Model Predictive Control) schemes (with and without terminal cost and constraints) for stabilizing the translational dynamics of thrust-propelled vehicles. Both approaches make use of an elaborated nonlinear feedback linearization controller and its associated ellipsoidal invariant set under restrictive input constraints, hence, guaranteeing the closed-loop asymptotic stability. The terminal constraint set of the corresponding NMPC design is easy to tune due to its clear formulation expressed directly in terms of the tuning variables while for the NMPC scheme without terminal constraint, the design allows to stabilize the system with a significantly shorter prediction horizon in comparison with existing method in the literature. Simulation and experimental tests over a nano-drone platform validate the proposed approaches.**

***Index Terms*—NMPC, terminal constraint, unconstrained NMPC, feedback linearization, ellipsoidal invariant set, asymptotic stability, VTOL (Vertical Take-Off and Landing) vehicles.**

I. Introduction

NOWADAYS, there is a tremendous interest in the research community on UAVs (Unmanned Aerial Vehicles) and thrust-propelled vehicles (like quadcopters, hexacopters), in particular [1], [2]. Usually, the drone platforms already have their built-in controllers which control the rotors to track the four inputs which in turn govern the thrust level and the three Euler angles [2]–[4]. Therefore, more recently, controlling the thrust propelled vehicles requires only the maneuvering of their translation dynamics [4]. This is already a challenging task as the dynamics are not only strongly nonlinear but also subject to many operating constraints [1]–[3].

Designing a controller respecting the system dynamics and constraints can be fulfilled by considering an Model Predictive Control (MPC) [5] approach. This control strategy provides the input obtained by solving a finite horizon optimal control problem where the dynamics and constraints are explicitly taken into account. It is well known that the computational burden of the MPC-based optimization problem may pose serious issues in many time-critical applications. However, with recent technological advances, it becomes possible to employ MPC for many applications requiring fast response [6] as is the case in the control and coordination of aerial vehicles [1], [7]. E.g., [1] approximates the NMPC (Nonliner MPC) of stabilizing a human-sized quadcopter vehicle with a quadratic problem and succeeds in solving it by using a Cortex A9 800-MHz microcontroller. [7] employs an MPC scheme to generate quadcopter trajectories in real time by formulating the problem using the system's jerks. Also, it has been proved that a linear MPC-based optimization can be even executed over a 8-bit microcontroller by using a convex lifting approach [8]. Another important issue of MPC is that it does not provide an explicit controller equation [5], thus, causing difficulty in analyzing the stability and the feasibility of the closed-loop scheme (except explicit MPC whose numerical limitations make it difficult for many real-time implementations [9], [10]). The use of MPC laws without stability guarantees may prove shortsighted in certain applications. In particular, the authority loss for attitude and position control in UAVs will not only risk the mission objectives but lead to critical behavior. In the literature, the stability of an MPC controller can be achieved either by adding a terminal constraint [10]–[12] or by enlarging the prediction horizon [13]. In this paper, we employ both approaches to design two different NMPC schemes for stabilizing the translation dynamics of thrust-propelled vehicles. These designs make use of an elaborated feedback linearization controller which was introduced in the previous work of the authors [2] and was successfully applied to a nano-drone. In here, we employ this controller as the local controller (i.e., locally constraint admissible) to design the two NMPC schemes with guaranteed stability. Note that the role of a local controller is to ensure that once inside the terminal region, the system's trajectories remain inside. Thus, the local controller only acts as a guarantee of feasibility and asymptotic stability and is not actually used throughout the simulation.

Next we enumerate this paper's contributions which, to the best of our knowledge, are new to the state of the art. We:

i) propose an NMPC design with terminal ellipsoidal region which is invariant and constraint admissible under a nonlinear feedback linearization controller.
ii) propose an NMPC design without terminal constraint and the corresponding tuning procedure for choosing the optimal prediction horizon which guarantees the stability.
iii) validate the NMPC controllers through simulation and experimental testing over the Crazyflie 2.0 nano-quadcopter platform [3].

The paper structure is as follows. Section II presents first the translational thrust-propelled systems, then, the principles of the two NMPC design schemes, i.e., NMPC with and without terminal constraint. Section III details several properties of the elaborated feedback linearization controller which are employed in Section IV to ensure the stability of the two

N.T. Nguyen is with the Univ. of Lübeck, Institute for Robotics and Cognitive Systems, 23562 Lübeck, Germany (email: nguyen@rob.uni-luebeck.de).

I. Prodan and L. Lefèvre are with Univ. Grenoble Alpes, Grenoble INP Institute of Engineering and Management, LCIS, 26000 Valence, France (email: {ionela.prodan,laurent.lefevre} @ lcis.grenoble-inp.fr). This work was done while N.T. Nguyen was pursuing his PhD degree at laboratory LCIS.

NMPC designs. Section VI presents the simulation results and Section VII shows the corresponding experimental validation. Section VIII presents the conclusions and the future work.

Notations: Denote by $\boldsymbol{I}_n$ the identity matrix of size n and by $0_{m\times n}$ the zero matrix of size $m \times n$. $\mathbf{0}$ represents a zero matrix of appropriate dimension inferred from the context. Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalues of the square matrix A having only real eigenvalues. For a vector $\boldsymbol{x} \in \mathbb{R}^n$ and a positive semi-definite matrix $P \in \mathbb{R}^{n\times n}$, $\|\boldsymbol{x}\|_P^2$ denotes the weighted norm $\boldsymbol{x}^\top P\boldsymbol{x}$ while $\|\boldsymbol{x}\|$ represents the Euclidean norm of the vector $\boldsymbol{x}$. Furthermore, denote by $\boldsymbol{x}(i|k)$ and $\boldsymbol{u}(i|k)$ the values of the state $\boldsymbol{x}$ and the input $\boldsymbol{u}$ at time instant i predicted upon the information available at time $k \in \mathbb{N}$.

II. Model dynamics and NMPC setups

A. System modeling

This section recapitulates the discrete translation dynamics of a standard thrust-propelled aircraft system. The dynamics are obtained by applying the Runge–Kutta fourth-order discretization method [14] to the continuous thrust-propelled system found in [2] and are expressed in their state-space representation as follows:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) = A\boldsymbol{x}_k + h_\psi(\boldsymbol{u}_k), \tag{1}$$

where the state $\boldsymbol{x}_k \in \mathbb{R}^6$ gathers the position and the velocity at the time step k, $\boldsymbol{x}_k \triangleq [x_k\ y_k\ z_k\ v_{x_k}\ v_{y_k}\ v_{z_k}]^\top$, the input $\boldsymbol{u}_k \in \mathbb{R}^3$, $\boldsymbol{u}_k \triangleq [T_k\ \phi_k\ \theta_k]^\top$ consists of the normalized thrust T (the normal thrust divided by the system's mass), the roll, ϕ, and pitch, θ, angles. The yaw angle $\psi \in [-\pi, \pi]$ is an assumed known constant influencing the system (usually considered to be zero as in [15] but here we aim to exploit a general fix value of ψ). Explicitly, the matrix A and the input terms $h_\psi(\boldsymbol{u}_k)$ are given by:

$$A = \begin{bmatrix} \boldsymbol{I}_3 & \Delta_t \boldsymbol{I}_3 \\ 0_{3\times 3} & \boldsymbol{I}_3 \end{bmatrix}, \tag{2}$$

$$h_\psi(\boldsymbol{u}_k) = \begin{bmatrix} \dfrac{\Delta_t^2}{2}\boldsymbol{I}_3 \\ \Delta_t \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} T_k(\mathrm{c}\,\phi_k\,\mathrm{s}\,\theta_k\,\mathrm{c}\,\psi + \mathrm{s}\,\phi_k\,\mathrm{s}\,\psi) \\ T_k(\mathrm{c}\,\phi_k\,\mathrm{s}\,\theta_k\,\mathrm{s}\,\psi - \mathrm{s}\,\phi_k\,\mathrm{c}\,\psi) \\ -g + T_k\,\mathrm{c}\,\phi_k\,\mathrm{c}\,\theta_k \end{bmatrix}, \tag{3}$$

where Δ_t is the discretization time step and g stands for the gravitational acceleration[1]. Note that, the discrete system (1) is linear in the state since the original continuous system taken from [2] already possesses this property. The vehicle has to respect thrust limits, with $T_{limit} > g$, and roll and pitch constraints as follows:

$$\boldsymbol{u}(k) \in \mathcal{U} = \{0 \le T_k \le T_{limit}, |\phi_k| \le \epsilon_c, |\theta_k| \le \epsilon_c\}, \tag{4}$$

with $\epsilon_c \in (0, \pi/2)$, the desired maximum value of the angles. Then, without loss of generality, we aim to stabilize the system (1) around the equilibrium point:

$$\boldsymbol{x}_e = \mathbf{0},\ \boldsymbol{u}_e = [g\ 0\ 0]^\top. \tag{5}$$

Note that, the thrust-propelled system (1) can be stabilized at any position $\boldsymbol{x}'_e \in \mathbb{R}^3$ with zero velocity by simply choosing the origin of the coordinate system at $\boldsymbol{x}'_e$.

B. NMPC with and without terminal constraint

Model Predictive Control (MPC) is a control strategy in which at each time step, an open-loop optimal control problem of prediction horizon N, subject to the system dynamics (1) and constraints on states (including the current state as the initial condition) and inputs (4), is solved to obtain an optimal control sequence. From the sequence, only the first element is applied to the system. At the next sampling instant, the state is measured again and introduced to the optimization problem. The process is iteratively executed to establish the closed-loop controlled system [5]. Hereinafter, we introduce two different NMPC setups, i.e., with and without terminal constraint for the considered dynamics (1) which are adopted from their general designs given in [11], [13]. We also recapitulate the corresponding design principles for guaranteeing the stability of the two NMPC schemes [5], [13].

1) NMPC with terminal constraint (NMPC-t): The NMPC design detailed hereby employs both the terminal cost and the terminal constraint[2]. For more details, the open-loop optimization problem $V_{T,N_T}(\boldsymbol{x}_k)$ at time step k with the measured state $\boldsymbol{x}_k$ is solved over the prediction horizon N_T:

$$V_{T,N_T}(\boldsymbol{x}_k) := \min_{\boldsymbol{u}(\cdot|k)} J_{T,N_T}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)), \tag{6}$$

subject to

$$\boldsymbol{x}(i+1|k) = f(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k)), \tag{7a}$$
$$\boldsymbol{u}(i|k) \in \mathcal{U},\ i \in \{0, \ldots, N_T - 1\}, \tag{7b}$$
$$\boldsymbol{x}(0|k) = \boldsymbol{x}_k,\ \boldsymbol{x}(N_T|k) \in \mathcal{X}_f, \tag{7c}$$

with the cost function $J_{T,N_T}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k))$ defined as:

$$J_{T,N_T}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)) = \sum_{i=0}^{N_T-1} \ell(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k)) + F(\boldsymbol{x}(N_T|k)).$$

The predicted state and input sequences employed for the optimization problem at time step k are denoted by $\boldsymbol{x}(\cdot|k) \triangleq [\boldsymbol{x}(0|k)\ \ldots\ \boldsymbol{x}(N_T|k)]$ and $\boldsymbol{u}(\cdot|k) \triangleq [\boldsymbol{u}(0|k)\ \ldots\ \boldsymbol{u}(N_T - 1|k)]$, respectively. The terminal region $\mathcal{X}_f \subset \mathbb{R}^{6\times 6}$ is designed according to the stability conditions introduced hereinafter. Furthermore, the stage cost, $\ell(\boldsymbol{x}, \boldsymbol{u})$, and terminal cost, $F(\boldsymbol{x})$, are defined as:

$$\ell(\boldsymbol{x}, \boldsymbol{u}) = \|\boldsymbol{x} - \boldsymbol{x}_e\|_Q^2 + \|\boldsymbol{u} - \boldsymbol{u}_e\|_R^2, \tag{8}$$
$$F(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_e\|_P^2, \tag{9}$$

in which, the symmetric matrices $Q \in \mathbb{R}^{6\times 6}$ (positive definite) and $R \in \mathbb{R}^{3\times 3}$ (positive semi-definite) are defined by the users, while the positive definite matrix $P \in \mathbb{R}^{6\times 6}$ is a solution of the Lyapunov equation introduced later (see (32)). Note that, the formulation of $\ell(\cdot)$ in (8) allows $R = \mathbf{0}$ as similar to [16], [17]. Assuming that the optimization problem (6)-(7) is successfully solved, it provides the optimal state and input trajectories $(\boldsymbol{x}^*(\cdot|k), \boldsymbol{u}^*(\cdot|k))$. Then, the MPC control action applied to the system (1) at time step k is defined as:

$$\mu_T(\boldsymbol{x}_k) = \boldsymbol{u}_T^*(0|k). \tag{10}$$

[1] We have used in (3) "c" and "s" to denote the $\cos(\cdot)$ and $\sin(\cdot)$ functions, respectively.

[2] Usually referred as *quasi-infinite horizon NMPC* [11].

Finally, the nominal closed-loop system is given by:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \mu_T(\boldsymbol{x}_k)). \tag{11}$$

Hereinafter, we summarize the NMPC principles given in [11], [18] to establish the sufficient conditions for the stability of the nominal closed-loop system (11).

Assumption 1: Consider the closed terminal set $\mathcal{X}_f$ associated to (6) and the terminal cost F from (9). We assume that for each $\boldsymbol{x}_k \in \mathcal{X}_f$ at any time step k, there exists an admissible control value[3] $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ such that the two following conditions hold:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)) \in \mathcal{X}_f, \tag{12}$$

$$F(\boldsymbol{x}_{k+1}) + \ell(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)) \leq F(\boldsymbol{x}_k). \tag{13}$$

Theorem 1: Let Assumption 1 hold. Then, the nominal closed-loop system (11) under the feedback law μ_T (10) from the NMPC-t scheme (6)-(7) achieves recursive feasibility[4] and asymptotic stability. □

Proof: See Chapter 5 in [18] and Theorem 1 in [11]. ■

2) NMPC without terminal constraint (u-NMPC): The NMPC scheme without terminal cost and terminal stabilizing constraint is also referred as unconstrained NMPC as in [18], [19], hence, being denoted by u-NMPC within the paper. The following optimization control problem $V_{U,N_U}(\boldsymbol{x}_k)$ at time k is solved over the prediction horizon N_U:

$$V_{U,N_U}(\boldsymbol{x}_k) := \min_{\boldsymbol{u}(\cdot|k)} J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)), \tag{14}$$

subject to

$$\boldsymbol{x}(i+1|k) = f(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k)), \tag{15a}$$

$$\boldsymbol{u}(i|k) \in \mathcal{U}, \ i \in \{0, \ldots, N_U - 1\}, \tag{15b}$$

$$\boldsymbol{x}(0|k) = \boldsymbol{x}_k, \tag{15c}$$

with the cost function $J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k))$ defined in terms of the stage cost ℓ from (8) as:

$$J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)) = \sum_{i=0}^{N_U-1} \ell(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k). \tag{16}$$

Then, the u-NMPC control action and the nominal closed-loop system at time k are given by:

$$\mu_U(\boldsymbol{x}_k) = \boldsymbol{u}_U^*(0|k), \tag{17}$$

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \mu_U(\boldsymbol{x}_k)), \tag{18}$$

with $\boldsymbol{x}_U^*(\cdot|k)$ the optimal input trajectories. The stability of system (11) is well studied and presented in [16], [18] and summarized hereinafter.

Assumption 2: There exist constants $\gamma, c \in \mathbb{R}_{>0}$ such that for any $N_U \geq 2$ and for all the initial state $\boldsymbol{x}_0$ satisfying $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\|_Q^2 \leq c$, we have:

$$V_{U,N_U}(\boldsymbol{x}_k) \leq \gamma \|\boldsymbol{x}_0 - \boldsymbol{x}_e\|_Q^2. \tag{19}$$

Theorem 2: Let Assumption 2 hold. Then, there exists an $N_0 \in \mathbb{N}$, such that for the prediction horizon length $N_U \geq N_0$, the equilibrium point $\boldsymbol{x}_e$ is uniformly exponentially stable under the nominal closed-loop dynamics (11) for any initial state $\boldsymbol{x}_0$ satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. □

Proof: See Theorem 1 in [16] and Theorem 3.6 in [13]. At first, in [16], the authors show that $V_{N,U}(\boldsymbol{x}_k) \leq c\gamma$ implies that $V_{U,N_U}(\boldsymbol{x}_k) \leq \gamma\|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2$ with a case dictinction based on whether $\|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2 \leq c$ or not [16].
Secondly, whenever $V_{U,N_U}(\boldsymbol{x}_k) \leq \gamma\|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2$ holds, in [13], the authors show that $V_{U,N_U}(\boldsymbol{x}_k)$ decreases for all $N_U \geq N_0$ with N_0 given by (see Variant 3 in [13] for more details):

$$N_0 = 2 + \frac{\ln(\gamma - 1)}{\ln \gamma - \ln(\gamma - 1)}. \tag{20}$$

Lastly, the recursive feasibility and exponential stability are obtained with the initial condition $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. ■

Remark 1: One can impose state constraints into the two NMPC controllers by adding the following state condition into (7b) and (15b):

$$\boldsymbol{x}(i|k) \in \mathcal{X}, \tag{21}$$

with $\mathcal{X} \subseteq \mathbb{R}^6$, the non-empty state constraint set required to be convex and to contain the equilibrium point $\boldsymbol{x}_e$. Then, Theorems 1 and 2 accordingly require $\mathcal{X}_f \subseteq \mathcal{X}$ and $\boldsymbol{x}_k \in \mathcal{X}, \forall \|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2 \leq c$. Within this paper, we do not consider state constraints for the thrust-propelled dynamics (1) even though the proposed approaches provide the means to scale the terminal region $\mathcal{X}_f$ (by tuning its radius as will be discussed in Section III) and the set of $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\|_Q^2 \leq c$ (by increasing the value of Q). This is due to the fact that we want to analyze the maximum performance (convergence speed - shown by the velocity) of the NMPC controllers. Furthermore, the initial condition has to be chosen close to the equilibrium point for guaranteeing the stability (c.f. Theorem 2) due to hardware limitations (with largest possible prediction horizon of only 10 steps), hence, realistic constraints on position are never activated under our experimental tests. □

C. Motivation

Assumptions 1 and 2, which are required to design the two NMPC constructions provided in Sections IV-A and IV-B are hard to validate for a generic nonlinear system given as in (7a) and (15a). In the literature, the standard approach is to linearize the dynamics around the equilibrium point, from which it follows the construction of a linear controller and its associated invariant set (of ellipsoidal form as in [11], [16] or polyhedral form as in [12]). However, employing a linear controller for a general nonlinear system (7a),(15a) obviously restricts the corresponding invariant set (also serving as the terminal region $\mathcal{X}_f$ in (7)), hence, arguably reducing the efficiency of the NMPC-t controller (6)-(10) and leading to an impractically large prediction horizon for the u-NMPC controller (14)-(17). Thus, it is worthwhile to ask whether, for particular dynamics (as those shown in (1)), we may dispense with the linearized dynamics/linear controller construction and, instead, check Assumptions 1 and 2 by applying a feedback linearization controller.

In our opinion this helps not only to account for the nonlinearities of the system (1) but also to apply various tools available

[3] The notation $\boldsymbol{u}_{\text{loc}}$ stands for local controller, i.e., being locally admissible.

[4] The initial iteration successfully executed implies the feasibility of all the further steps.

for linear dynamics when considering the resulting closed-loop linear system (i.e., invariant set construction becomes much easier, as the difficulties are "pushed" in the nonlinear input). As is to be expected, the simplified closed-loop dynamics come at the price of a complex input formulation which, in turn, makes the input constraint and stage cost $\ell(\boldsymbol{x},\boldsymbol{u})$ in (4),(8) formulations difficult to handle. These difficulties will be tackled by employing previously obtained results of the authors [2], [20] and which will be recapitulated in the forthcoming sections.

III. Feedback linearization law and input constraint satisfaction

This section recapitulates the results of a feedback linearization controller for quadcopter trajectory tracking [2], [4] which will be employed as the local controller for guaranteeing the stability of the two NMPC controllers detailed in Section II-B. In general, feedback linearization control approach reduces the controlled system to an equivalent linear system (assuming no mismatches on state feedback and system dynamics, as similar to nominal NMPC applications in (10) and (17)). Thus, the method does not require any approximation and does not restrict the system to operate only in the neighborhood of the linearization point [21]. Let us start by introducing the standard feedback linearization law $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \triangleq \left[T_{\text{loc}}(\boldsymbol{x}_k)\ \phi_{\text{loc}}(\boldsymbol{x}_k)\ \theta_{\text{loc}}(\boldsymbol{x}_k)\right]^\top$ defined as follows (more details can be found in [2], [4]):

$$T_{\text{loc}}(\boldsymbol{x}_k) = \sqrt{u_x^2(\boldsymbol{x}_k) + u_y^2(\boldsymbol{x}_k) + (u_z(\boldsymbol{x}_k)+g)^2}, \tag{22a}$$

$$\phi_{\text{loc}}(\boldsymbol{x}_k) = \arcsin\left(\frac{u_x(\boldsymbol{x}_k)\,\mathrm{s}\,\psi - u_y(\boldsymbol{x}_k)\,\mathrm{c}\,\psi}{T_{\text{loc}}(\boldsymbol{x}_k)}\right), \tag{22b}$$

$$\theta_{\text{loc}}(\boldsymbol{x}_k) = \arctan\left(\frac{u_x(\boldsymbol{x}_k)\,\mathrm{c}\,\psi + u_y(\boldsymbol{x}_k)\,\mathrm{s}\,\psi}{u_z(\boldsymbol{x}_k)+g}\right), \tag{22c}$$

where the virtual control inputs $u_q(\boldsymbol{x}_k)$ with $q \in \{x,y,z\}$ are calculated in terms of the positions and velocities (i.e., $\boldsymbol{x}_k \triangleq [x_k\ y_k\ z_k\ v_{x_k}\ v_{y_k}\ v_{z_k}]^\top$ from (1)):

$$u_q(\boldsymbol{x}_k) = K_{1_q} q_k + K_{2_q} v_{q_k}, \tag{23}$$

with $K_{1_q}, K_{2_q} \in \mathbb{R}$ the control gains. If the feedback linearization control action $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ as in (22) is not affected by the saturation limit, i.e., $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ with $\mathcal{U}$ in (4), then $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ transforms the dynamics (1) into the following linear system:

$$\boldsymbol{x}_{k+1} = A_{cl}\boldsymbol{x}_k, \tag{24}$$

with the matrix $A_{cl} \in \mathbb{R}^{6\times 6}$ given by:

$$A_{cl} = \begin{bmatrix} \boldsymbol{I}_3 + \frac{\Delta_t^2}{2}K_1 & \Delta_t\boldsymbol{I}_3 + \frac{\Delta_t^2}{2}K_2 \\ \Delta_t K_1 & \boldsymbol{I}_3 + \Delta_t K_2 \end{bmatrix}, \tag{25}$$

and with $K_1 = \operatorname{diag}(K_{1_x}, K_{1_y}, K_{1_z})$ and $K_2 = \operatorname{diag}(K_{2_x}, K_{2_y}, K_{2_z})$. Furthermore, the control gains K_{1_q}, K_{2_q} are required to satisfy the following conditions:

$$-\frac{2}{\Delta_t} < K_{2_q} < \frac{\Delta_t}{2}K_{1_q} < 0, \tag{26}$$

which serve to establish the asymptotic stability of the resulted linear system (24) according to the Routh-Hurwitz criterion. Then, A_{cl} is a Schur matrix having all its eigenvalues strictly inside the unit circle.

However, the control action $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ has to respect the input constraints (i.e., $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ (4)) for the linearization to be valid. Thus, hereinafter, we will introduce some conditions on the virtual control inputs $u_x(\boldsymbol{x}_k)$, $u_y(\boldsymbol{x}_k)$ and $u_z(\boldsymbol{x}_k)$ in (23) to ensure the input constraint satisfaction.

Proposition 1: By choosing the values of the three positive saturation limits U_x, U_y and U_z such that:

$$U_z < g, \tag{27}$$

$$U_x^2 + U_y^2 \le (-U_z + g)^2 \tan^2 \epsilon_c, \tag{28}$$

$$\sqrt{U_x^2 + U_y^2 + (U_z+g)^2} \le T_{limit}, \tag{29}$$

we have that, if the virtual inputs (23) satisfy $|u_x(\boldsymbol{x}_k)| \le U_x$, $|u_y(\boldsymbol{x}_k)| \le U_y$ and $|u_z(\boldsymbol{x}_k)| \le U_z$, then, the followings hold:
(i) the feedback linearization control action $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ (22), for all value of $\psi \in [-\pi, \pi]$;
(ii) the values of $\|\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) - \boldsymbol{u}_e\|^2$ are bounded by:

$$\|\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) - \boldsymbol{u}_e\|^2 \le \boldsymbol{x}_k^\top (K^\top K + 2\Gamma)\boldsymbol{x}_k, \tag{30}$$

where $K = [K_1\ K_2]$ (24) and $\Gamma \in \mathbb{R}^{6\times 6}$ is given by:

$$\Gamma = \frac{1}{(-U_z+g)^2} K_{xy}^\top K_{xy}, \tag{31}$$

with $K_{xy} = \left[\operatorname{diag}(K_{1_x}, K_{1_y}, 0)\ \operatorname{diag}(K_{2_x}, K_{2_y}, 0)\right]$. □

Proof: The proof of point (i) is presented in previous works of the authors [2], [20] and the proof of point (ii) is given in Appendix A. The goal of the condition (27) is to constrain the thrust-propelled dynamics (1) not to free fall. Also, (28) and (29) are to guarantee the constraints on the angle ($|\phi_{\text{loc}}|, |\theta_{\text{loc}}| \le \epsilon_c$) and the thrust ($T_{\text{loc}} \le T_{limit}$), respectively, with ϵ_c, T_{limit} from (4). Further discussions on the origin and the advantages/ disadvantages of (27)-(29) are given in [2]. ■

By Proposition 1, we guarantee the input constraint satisfaction for the feedback linearization controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ (22) for the whole region of $|u_q(\boldsymbol{x_k})| \le U_q, \forall q \in \{x,y,z\}$ (23). Hereinafter, we will construct an ellipsoidal invariant set within this region, thus, being able to employ a simple analysis resulted from the linear system (24).

Let us consider a symmetric positive definite matrix $M \in \mathbb{R}^{6\times 6}$. We obtain the matrix $P \in \mathbb{R}^{6\times 6}$, also being symmetric positive definite, as the unique solution of the following Lyapunov equation:

$$A_{cl}^\top P A_{cl} = P - M. \tag{32}$$

with the Schur matrix A_{cl} from (24).

Proposition 2: Let us consider the set $\mathcal{X}_f$ defined as follows:

$$\mathcal{X}_f = \{\boldsymbol{x} \in \mathbb{R}^6 |\ \|\boldsymbol{x}\|_P^2 \le \delta\}, \tag{33}$$

with δ given by:

$$\delta = \lambda_{\min}(P) r^2, \text{ with } r^2 = \min_{q \in \{x,y,z\}} \left\{ \frac{U_q^2}{K_{1_q}^2 + K_{2_q}^2} \right\}, \tag{34}$$

and U_q from (27)-(29). Then, for all $\boldsymbol{x}_k \in \mathcal{X}_f$, we have that:
(i) the feedback linearization controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ (22),

for all $\psi \in [-\pi, \pi]$;
(ii) $\mathcal{X}_f$ is forward invariant[5] under $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$. Furthermore, $\boldsymbol{x}_{k+1}$ satisfies:

$$\|\boldsymbol{x}_{k+1}\|_P^2 \le \rho \|\boldsymbol{x}_k\|_P^2, \text{s.t. } \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)), \tag{35}$$

with $\rho \in (0,1)$ given by:

$$\rho = 1 - \lambda_{\min}(M)/\lambda_{\max}(P). \tag{36}$$

□

Proof: First, since the matrix P obtained from (32) is symmetric positive definite, for all $\boldsymbol{x}_k$ such that $\|\boldsymbol{x}_k\|_P^2 \le \delta$ with δ as in (34), we have that:

$$\lambda_{\min}(P)\|\boldsymbol{x}_k\|^2 \le \|\boldsymbol{x}_k\|_P^2 \le \delta, \tag{37}$$

thus, $\|\boldsymbol{x}_k\|^2 \le r^2$, which implies that $q_k^2 + v_{q_k}^2 \le r^2$ for all $q \in \{x, y, z\}$ with r as in (34). Hence, by using Cauchy-Schwarz inequality, the virtual inputs $u_q(\boldsymbol{x}_k)$ (23) satisfy:

$$|u_q(\boldsymbol{x}_k)| \le \sqrt{r^2(K_{1_q}^2 + K_{2_q}^2)} \le U_q. \tag{38}$$

Then, Proposition 1 provides that $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ and the system $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k))$ is linearized into the linear system $\boldsymbol{x}_{k+1} = A_{cl}\boldsymbol{x}_k$ (24). Hence, we arrive to:

$$\begin{aligned}\|\boldsymbol{x}_{k+1}\|_P^2 &= \boldsymbol{x}_k^\top A_{cl}^\top P A_{cl} \boldsymbol{x}_k = \|\boldsymbol{x}_k\|_P^2 - \|\boldsymbol{x}_k\|_M^2 \\ &\le \underbrace{\|\boldsymbol{x}_k\|_P^2 - \lambda_{\min}(M)/\lambda_{\max}(P)\|\boldsymbol{x}_k\|_P^2}_{\rho\|\boldsymbol{x}_k\|_P^2} < \delta.\end{aligned} \tag{39}$$

Thus, $\mathcal{X}_f$ is forward invariant and (35) is validated. ■

IV. Stability of different NMPC schemes

In the following, we gather all the results obtained from the feedback linearization controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ (22) to design two NMPC schemes with and without terminal constraint. We check that Assumptions 1 and 2 given in Section II-A hold, hence, guaranteeing the asymptotic stability property for stabilizing the thrust-propelled system (1).

A. NMPC with terminal constraint (NMPC-t)

Proposition 3: Consider the setup of the NMPC-t scheme (8)-(10) using the terminal region $\mathcal{X}_f$ in (33) with the matrix P from the Lyapunov equation (32). If the symmetric matrix M in (32) is chosen such that $M \succeq Q^*$ with the matrix Q^* verifying:

$$Q^* = Q + \lambda_{\max}(R)(K^\top K + 2\Gamma), \tag{40}$$

with Q and R from (8), K and Γ from (31), then, Assumption 1 is satisfied with the feedback linearization controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ from (22). Thus, the NMPC-t scheme (8)-(10) guarantees the recursive feasibility and the (nominal) asymptotic stability for the closed-loop dynamics (11). □

Proof: At first, since $M \succeq Q^*$ with Q^* as in (40), the matrices M and P obtained from (32) are both symmetric positive definite. Then, Proposition 2 holds and hence, $\mathcal{X}_f$ is forward invariant under $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ (35). Furthermore, since the virtual inputs $u_q(\boldsymbol{x}_k) \le U_q$, $\forall q \in \{x, y, z\}$ (38) for all $\boldsymbol{x}_k \in \mathcal{X}_f$, Proposition 1 provides $\|\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) - \boldsymbol{u}_e\|^2 \le (K^\top K + 2\Gamma)$ with K and Γ from (30)-(31). Thus, the stage cost $\ell(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k))$ is bounded by:

$$\ell(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)) \le \boldsymbol{x}_k^\top Q^* \boldsymbol{x}_k \tag{41}$$

with Q^* from (40). Then, by employing the terminal cost $F(\boldsymbol{x}_k) = \|\boldsymbol{x}_k - \boldsymbol{x}_e\|_P^2$ in (9), the condition (13) is verified as follows:

$$\begin{aligned}&F(f(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k))) - F(\boldsymbol{x}_k) + \ell(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)) \\ &\le \boldsymbol{x}_k^\top (A_{cl}^\top P A_{cl} - P)\boldsymbol{x}_k + \boldsymbol{x}_k^\top Q^* \boldsymbol{x}_k \\ &= \boldsymbol{x}_k^\top(-M + Q^*)\boldsymbol{x}_k \le \boldsymbol{0},\end{aligned} \tag{42}$$

in which P satisfies (32) and $M \succeq Q^*$ from (40). Hence, completing the proof. ■

Hereinafter, let us summarize the design procedure of the NMPC scheme with terminal constraint given in (8)–(10) for stabilizing the dynamics (1).

Procedure 1: **NMPC-t design**

1) Choose the positive definite symmetric matrices $Q \in \mathbb{R}^{6\times 6}$ and $R \in \mathbb{R}^{3\times 3}$ to formulate the stage cost in (8).
2) Choose the saturation limits U_x, U_y and U_z satisfying (27)–(29).
3) Choose the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ satisfying (26).
4) Find the matrix Q^* in (40), then, define the symmetric matrix $M \succeq Q^*$ and solve the Lyapunov equation (32) for P.
5) Find δ in (34) to obtain the terminal region $\mathcal{X}_f$ in (33).
6) Define the prediction horizon N_T such that the first NMPC iteration w.r.t. the initial state $\boldsymbol{x}_0$ is feasible. ■

Remark 2: For a predefined initial state $\boldsymbol{x}_0$, Step 6 in Procedure 1 requires the users to run the NMPC-t algorithm once, while progressively increasing prediction horizon until the optimization problem becomes feasible. However, note that, the computation time is greatly affected by any increase of the prediction length. Thus, in order to continue increasing the region of attraction when the prediction horizon N_T is already large, one can increase the size of the terminal constraint set $\mathcal{X}_f$ given in (33) instead. The parameters which affect most the size of the terminal region $\mathcal{X}_f$ (33) are the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ in (26). In general, decreasing the magnitudes of the control gains increase the size of the set $\mathcal{X}_f$ and vice versa. Therefore, when state constraints are considered as mentioned in Remark 1, the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ are required to be appropriately chosen such that $\mathcal{X}_f \subseteq \mathcal{X}$ with the state constraint set $\mathcal{X}$ from (21). □

B. NMPC without terminal constraint (u-NMPC)

Proposition 4: Consider the translational thrust-propelled dynamics (1) and the corresponding u-NMPC setup (14)-(18). By choosing the matrix $M \in \mathbb{R}^{6\times 6}$ used in (32) being symmetric positive definite and setting the constants $c, \gamma \in \mathbb{R}_{>0}$ from (19) as:

$$c = \delta\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}, \ \gamma = \frac{\lambda_{\max}(Q^*)\lambda_{\max}(P)}{\lambda_{\min}(Q)\lambda_{\min}(P)(1-\rho)}, \tag{43}$$

[5] A sub-level set of a Lyapunov function is a forward invariant set.

with δ in (34), Q in (8), Q^* in (40), ρ in (36) and P satisfying (32), then, Assumption 2 is satisfied. Furthermore, the equilibrium point $\boldsymbol{x}_e$ in (5) is uniformly exponentially stable under the nominal closed-loop dynamics (18) for all the initial states $\boldsymbol{x}_0$ satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. □

Proof: Since the matrices Q as in (8) and P from (32) are symmetric and positive definite, we have the following chain of inequalities:

$$\|\boldsymbol{x}_k\|_P^2 \leq \lambda_{\max}(P)\|\boldsymbol{x}_k\|^2 \leq \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\|\boldsymbol{x}_k\|_Q^2, \tag{44}$$

hence, $\|\boldsymbol{x}_k\|_Q^2 \leq c$ implies that $\|\boldsymbol{x}_k\|_P^2 \leq \delta$.
Thus, for all initial states $\boldsymbol{x}_k$ such that $\|\boldsymbol{x}_k\|_Q^2 \leq c$, Proposition 2 provides a feasible candidate consisting of the state and input trajectories $(\boldsymbol{x}_{\text{loc}}(\cdot|k), \boldsymbol{u}_{\text{loc}}(\cdot|k))$, of the prediction horizon length N_U, given by:

$$\boldsymbol{x}_{\text{loc}}(i+1|k) = f(\boldsymbol{x}_{\text{loc}}(i|k), \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_{\text{loc}}(i|k))), \tag{45}$$

with $i \in \{0, \ldots, N_U - 1\}$, the feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ from (22) and the initial condition $\boldsymbol{x}_{\text{loc}}(0|k) = \boldsymbol{x}_k$. Since V_{U,N_U} as in (14) is the optimal value function, it satisfies:

$$V_{U,N_U}(\boldsymbol{x}_k) \leq J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\cdot|k)). \tag{46}$$

Furthermore, similar to (41), the stage cost (8) of $J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\cdot|k))$ as in (16) is bounded as follows:

$$\ell(\boldsymbol{x}_{\text{loc}}(i|k), \boldsymbol{u}_{\text{loc}}(i|k)) \leq \|\boldsymbol{x}_{\text{loc}}(i|k)\|_{Q^*}^2, \tag{47}$$

with Q^* in (40). Then, since $\|\boldsymbol{x}_k\|_P^2 \leq \delta$, $\|\boldsymbol{x}_{\text{loc}}(i|k)\|_P^2$ exponentially converges as in (35). Thus, introducing (35) to (47) leads to:

$$\begin{aligned}\ell(\boldsymbol{x}_{\text{loc}}(i|k), \boldsymbol{u}_{\text{loc}}(i|k)) &\leq \frac{\lambda_{\max}(Q^*)}{\lambda_{\min}(P)}\rho^i\|\boldsymbol{x}_k\|_P^2 \\ &\leq \frac{\lambda_{\max}(Q^*)\lambda_{\max}(P)}{\lambda_{\min}(Q)\lambda_{\min}(P)}\rho^i\|\boldsymbol{x}_k\|_Q^2. \end{aligned}\tag{48}$$

Lastly, combining (46) and (48) implies that:

$$\begin{aligned}V_{U,N_U}(\boldsymbol{x}_k) &\leq \sum_{i=0}^{N_U-1} \ell(\boldsymbol{x}_{\text{loc}}(i|k), \boldsymbol{u}_{\text{loc}}(i|k)) \qquad (49)\\ &\leq \sum_{i=0}^{N_U-1} \rho^i \frac{\lambda_{\max}(Q^*)\lambda_{\max}(P)}{\lambda_{\min}(Q)\lambda_{\min}(P)}\|\boldsymbol{x}_k\|_Q^2 \\ &\leq \sum_{i=0}^{\infty} \rho^i \frac{\lambda_{\max}(Q^*)\lambda_{\max}(P)}{\lambda_{\min}(Q)\lambda_{\min}(P)}\|\boldsymbol{x}_k\|_Q^2 = \gamma\|\boldsymbol{x}_k\|_Q^2,\end{aligned}$$

with γ from (43). Thus, Assumption 2 is validated and the stability of the corresponding u-NMPC design follows Theorem 2. Hence, completing the proof. ■

In the following, the design procedure of the u-NMPC controller in (14)–(17) is summarized.

Procedure 2: **u-NMPC design**

1) 2) 3) Similar to the first three steps of Procedure 1.
4) Choose the symmetric positive definite matrix M in (32) by following the analysis detailed in Section V and solve the Lyapunov equation (32) for P.
5) Find δ as in (34), ρ as in (35), c and γ as in (43) in order to find N_0 given in (20).
6) Define the prediction horizon $N_U \geq N_0$ as in (20). ■

Remark 3: The region of attraction of the u-NMPC design (14)-(17) which consists of all the initial states $\boldsymbol{x}_0$ satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$ is difficult to estimate since it does not show an explicit requirement on the initial state $\boldsymbol{x}_0$ but requires solving the first u-NMPC iteration. Thus, the usual approach in the literature [13], [16] is to employ the condition $\|\boldsymbol{x}_0\|_Q^2 \leq c$ instead since $V_{U,N_U}(\boldsymbol{x}_0) \leq \gamma\|\boldsymbol{x}_0\|_Q^2$ holds as given in (49). However, it is worth noting that the region of $\|\boldsymbol{x}_0\|_Q^2 \leq c$ is very conservative w.r.t. the original region of attraction satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. □

V. Tuning the u-NMPC controller

It is well-known in the literature that the u-NMPC scheme requires a sophisticated tuning procedure in order to obtain the reasonable values of the required minimum prediction horizon N_0 and also the region of attraction which guarantee the stability [13], [16]. However, to the best of our knowledge, the tuning problems of the NMPC without terminal stabilizing constraints have been underestimated in various relating works [13], [16], [17], [19]. That is to say, people concentrate mostly on the stability proofs of their NMPC designs (e.g. as our contribution in Proposition 4), then, provide one illustrative example with specific parameters [16]. Theses examples actually aim to show how the results are obtained (e.g. in order for the readers to validate again the calculation process by themselves) but do not give the insight into the actual tuning process. For our particular u-NMPC design (14)-(17), the most influential parameters are the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ satisfying (26), the matrix M from (32) and the weighting matrices Q, R from (8) which are not easy to tune due to a large amount of decision variables. In Table I, we show that an arbitrary choice of these parameters can easily lead to an impractically large prediction horizon length while appropriate changes in the tuning parameters allow to reduce the required prediction horizon N_0 and to increase the domain of attraction (represented by $\|\mathsf{p}\|_Q^2 \leq c$ as in (19)).

In order to analyze how to tune the prediction horizon N_0 given in (20), we will mostly consider the parameter γ as in (43). By introducing ρ from (36) to (43), we can express γ as the multiplication of γ_1 and γ_2 as follows:

$$\gamma = \underbrace{\frac{\lambda_{\max}(Q^*)}{\lambda_{\min}(Q)}}_{\gamma_1}\underbrace{\frac{\lambda_{\max}^2(P)}{\lambda_{\min}(P)\lambda_{\min}(M)}}_{\gamma_2}, \tag{50}$$

with P obtained from solving (32) with a symmetric positive definite matrix $M \in \mathbb{R}^{6\times 6}$, Q in (8) and Q^* in (40). At first, from the formulation of N_0 given in (20), decreasing the value of γ as in (50) also reduces the prediction horizon N_0. Therefore, it is in our interest to minimize the values of γ_1 and γ_2 defined in (50). In order to reduce the decision variables, we follow some reasonable assumptions:

- We give equal importance to the motions along the three axes, i.e.:

$$Q = \text{diag}\{q_1, q_1, q_1, q_2, q_2, q_2\}, \tag{51}$$

$$K_{1_x} = K_{1_y} = K_{1_z} = k_1, \tag{52}$$

$$K_{2_x} = K_{2_y} = K_{2_z} = k_2. \tag{53}$$

Description	Q	R	K_1	K_2	M	N_0	c
First choice	$\begin{bmatrix}\boldsymbol{I}_3 & 0_{3\times3}\\ 0_{3\times3} & 0.1\boldsymbol{I}_3\end{bmatrix}$	$0.1\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$\boldsymbol{I}_6$	3121	0.024
Tuning R	$\begin{bmatrix}\boldsymbol{I}_3 & 0_{3\times3}\\ 0_{3\times3} & 0.1\boldsymbol{I}_3\end{bmatrix}$	$0.01\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$\boldsymbol{I}_6$	2786	0.024
Tuning Q	$\boldsymbol{I}_6$	$0.01\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$\boldsymbol{I}_6$	176.1	0.2397
Tuning K_1, K_2	$\boldsymbol{I}_6$	$0.01\boldsymbol{I}_3$	$-1.2\boldsymbol{I}_3$	$-0.9\boldsymbol{I}_3$	$\boldsymbol{I}_6$	154.1	0.2438
Tuning M	$\boldsymbol{I}_6$	$0.01\boldsymbol{I}_3$	$-1.2\boldsymbol{I}_3$	$-0.9\boldsymbol{I}_3$	$\begin{bmatrix}\boldsymbol{I}_3 & 0.1\boldsymbol{I}_3\\ 0.1\boldsymbol{I}_3 & 1.3\boldsymbol{I}_3\end{bmatrix}$	153.1	0.2708

TABLE I: Prediction horizon length w.r.t different tuning parameters.

with $q_1, q_2 > 0$ and $k_1, k_2 < 0$ satisfying (26). Consequently, the matrix A_{cl} in (24) becomes:

$$A_{cl} = \begin{bmatrix} \left(1+\frac{\Delta_t^2}{2}k_1\right)\boldsymbol{I}_3 & \Delta_t\left(1+\frac{\Delta_t}{2}k_2\right)\boldsymbol{I}_3 \\ \Delta_t k_1 \boldsymbol{I}_3 & (1+\Delta_t k_2)\boldsymbol{I}_3 \end{bmatrix}. \tag{54}$$

- The matrix $M \in \mathbb{R}^{6\times6}$ in (32) is parametrized as follows:

$$M = \begin{bmatrix} m_1\boldsymbol{I}_3 & m_3\boldsymbol{I}_3 \\ m_3\boldsymbol{I}_3 & m_2\boldsymbol{I}_3 \end{bmatrix}, \tag{55}$$

with $m_1 = 1$ and $m_2, m_3, m_4 \in \mathbb{R}$ satisfying the following conditions:

$$m_1 m_2 > m_3^2, \tag{56}$$

in which (56) is to guarantee the positive definiteness of M. With $m_1 = 1$, (56) is simplified to $m_2 > m_3^2$.

Remark 4: The reason for fixing $m_1 = 1$ in (55) is that any scaling on M results on the same scaling on P due to the Lyapunov equation (32). Then, all the eigenvalues of M and P are scaled similarly, hence, γ_1 in (50) does not change and therefore, fixing $m_1 = 1$ does not cause any loss of generality within the analysis (but reduces the numerical issues). □

Next, the Lyapunov equation (32) is linear in P which provides a full-rank system of linear equations with A_{cl} and M from (54)-(55). Thus, by solving this, the matrix P is explicitly given as follows:

$$P = \begin{bmatrix} p_1\boldsymbol{I}_3 & p_3\boldsymbol{I}_3 \\ p_3\boldsymbol{I}_3 & p_2\boldsymbol{I}_3 \end{bmatrix}, \tag{57}$$

with p_1, p_2, p_3 calculated as:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 1-a_1^2 & -2a_1a_3 & -a_3^2 \\ -a_2^2 & -2a_2a_4 & 1-a_4^2 \\ -a_1a_2 & 1-a_1a_4-a_2a_3 & -a_3a_4 \end{bmatrix}^{-1} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}, \tag{58}$$

in which, $a_1, \ldots, a_4$ are taken from $A_{cl} \triangleq \begin{bmatrix} a_1\boldsymbol{I}_3 & a_2\boldsymbol{I}_3 \\ a_3\boldsymbol{I}_3 & a_4\boldsymbol{I}_3 \end{bmatrix}$ with A_{cl} in (54). The eigenvalues of P from (57) are given by:

$$\lambda_{\substack{\text{max}\\ \text{min}}}(P) = \frac{1}{2}\left(p_1 + p_2 \pm \sqrt{(p_1-p_2)^2 + 4p_3^2}\right). \tag{59}$$

A. Tuning γ_1

By introducing the matrix Q from (51) and the control gains k_1, k_2 as in (52)–(53) to the formulation of the matrix Q^*(40), we obtain the maximum eigenvalue of Q^* as follows:

$$\lambda_{\text{max}}(Q^*) = \frac{1}{2}\Bigg(q_1 + q_2 + \overline{r}(k_1^2 + k_2^2) + \underbrace{\sqrt{\left(q_1 - q_2 + \overline{r}(k_1^2 - k_2^2)\right)^2 + 4\overline{r}^2k_1^2k_2^2}}_{\leq\sqrt{2(q_1-q_2)^2+2\overline{r}^2(k_1^4+k_2^4)}}\Bigg), \tag{60}$$

where $\overline{r}$ is directly proportional with the maximum eigenvalue of R:

$$\overline{r} = \lambda_{\text{max}}(R)\left(1 + \frac{2}{(-U_z+g)^2}\right), \tag{61}$$

with g the gravitational acceleration and $U_z \in \mathbb{R}_+, U_z < g$ from Proposition 1. Then, γ_1 is obtained by introducing (51)-(60) into (50) and is bounded by:

$$\frac{q_1 + q_2 + \overline{r}(k_1 + k_2)^2}{2\min(q_1, q_2)} \leq \gamma_1 \leq \frac{q_1 + q_2 + \overline{r}(k_1^2 + k_2^2) + \sqrt{2(q_1-q_2)^2 + 2\overline{r}^2(k_1^4 + k_2^4)}}{2\min(q_1, q_2)}, \tag{62}$$

in which, the first inequality is by introducing $\left(q_1 - q_2 + \overline{r}(k_1^2 - k_2^2)\right)^2 \geq 0$ to (60) and the latter one is by using the inequality given below (60). From (62), it can be observed that reducing the values of $|q_1 - q_2|$, $\lambda_{\text{max}}(R)$, k_1^2 and k_2^2 probably provide a smaller value of γ_1 (we use "probably" since the reductions of the aforementioned parameters actually make both the upper and lower bounds as in (62) smaller). Especially, in case of employing $q_1 = q_2 = q$ with $q \in \mathbb{R}^+$, γ_1 from (50) is simplified to γ_1^* as follows:

$$\gamma_1^* = 1 + \frac{\overline{r}(k_1^2 + k_2^2)}{q}, \tag{63}$$

which actually allows us to obtain a specific value of γ_1 (i.e., ≥ 1) by tuning only two weighting matrices Q and R from (8) regardless the predefined values of (k_1, k_2) as in (52)–(53). Therefore, for tuning γ_1 as in (50), we propose several general directions:

- decrease the ratio $\lambda_{\text{max}}(Q)/\lambda_{\text{min}}(Q)$ as much as possible.
- decrease the value of $\lambda_{\text{max}}(R)$ based on the employed values of (k_1, k_2) but bear in mind that a small value of R causes large input consumption.

- decrease the values of k_1^2 and k_2^2. However, this is not encouraged since it can cause an unexpected increase in the value of γ_2 defined as in (50) and probably result in a larger prediction horizon N_0.

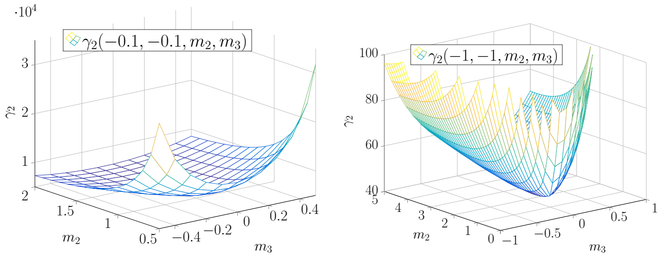
Remark 5: Formulation (63) explains the effects of the first two tuning steps given in Table I: i) reducing $\lambda_{\max}(R)$ and ii) eliminating the term $|q_1 - q_2|$. However, it does not explain for the rest of Table I which requires us to analyze the parameter γ_2 as detailed in the following. □

B. Tuning γ_2

Our approach is to construct an explicit formulation of γ_2 from (50) by using the parametrizations of M as in (55) and P as in (59). The function is expressed in terms of the tuning parameters, i.e., the control gains k_1, k_2 from (52)–(53) and the scalars m_2, m_3 from (55) as follows:

$$\gamma_2 \triangleq \gamma_2(k_1, k_2, m_2, m_3). \tag{64}$$

It will be in our interest to analyze and to find the (local) minimums of the function $\gamma_2(\cdot)$ from (64). Since the function is strongly nonlinear, non-convex and contains up to four variables, we have to divide the task into two steps:



(a) $(k_1, k_2) = (-0.1, -0.1)$ (b) $(k_1, k_2) = (-1, -1)$

Fig. 1: Illustration of $\gamma_2(k_1, k_2, m_2, m_3)$ as in (64) with different values of (k_1, k_2).

1. Find the optimal values of (m_2, m_3) which provide the local minimum value of γ_2 corresponding to a specific values of (k_1, k_2):

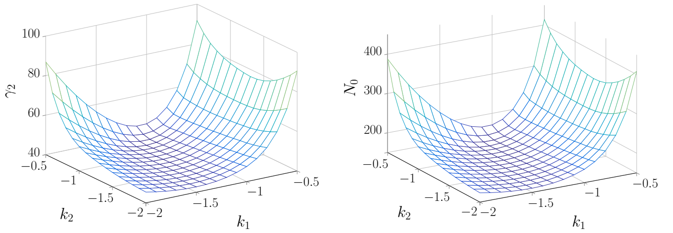
$$\gamma_2^*(k_1, k_2) = \min_{m_2, m_3} \gamma_2(k_1, k_2, m_2, m_3). \tag{65}$$

   subject to $m_2 > m_3^2$ as required in (56),

   $(m_2, m_3) \in [m_{2_{\min}}, m_{2_{\max}}] \times [-m_{3_{\max}}, m_{3_{\max}}]$,

   with $(m_{2_{\min}}, m_{2_{\max}}, m_{3_{\max}})$ positive scalars specifying the ranges of m_2 and m_3 as in (55) which are defined accordingly to the condition $m_2 > m_3^2$ as required in (56) and are chosen by user. The results of this step are illustrated in Figure 1 for $(k_1, k_2) = (-0.1, -0.1)$ and $(-1, -1)$.
2. Apply Step 1 for different choices of (k_1, k_2), then, compare the obtained minimum values γ_2^* as in (65) in order to provide the complete analysis as illustrated in Figure 2.

Remark 6: The solution of the optimization problem in (65) is obtained by checking a mesh grid of the variables (m_2, m_3) within the specific range $[m_{2_{\min}}, m_{2_{\max}}] \times [-m_{3_{\max}}, m_{3_{\max}}]$, hence, the solution's accuracy depends on the resolution of the mesh grid. However, the accuracy problem is not critical due to the fact that we always have to choose the prediction horizon $N_U \geq N_0$ (c.f. Theorem 2) and $N_U \in \mathbb{N}$ (e.g., it does not matter if we obtain $N_0 = 152.4$ instead of the precise value assumed to be 152.36 since the minimum prediction horizon needs to be an integer, which leads in both case to $N_U = 153$). Furthermore, using the mesh grid method provides very fast computing times. E.g., it takes only 0.08 seconds to construct the whole data points for Figure 2 which contains 51 × 41 points of (m_2, m_3) and 16 × 16 points of (k_1, k_2). □



(a) Illustration of the optimal function $\gamma_2^*(k_1, k_2)$ defined in (65) (b) N_0 obtained at the optimal choices of (m_2, m_3), $Q = \boldsymbol{I}_6$, $R = 0.01\boldsymbol{I}_3$

Fig. 2: Illustration of the analysis on N_0 from (20).

k_1 (52)	k_2 (53)	m_3 (55)	m_2 (55)	N_0 (20)	c (43)
−1.3	−1.1	0.2	1.6	153.8	0.206
−1.3	−1	0.1	1.3	153.5	0.215
−1.2	−0.9	0.1	1.3	153	0.272
−1.2	−0.8	0	1	152.2	0.287
−0.1	−0.1	**Method in [16],**		10^6	10^{-4}
−1	−1	using linear controller		5124	10^{-7}
−2	−2	and the linearized dynamics		1112	10^{-7}

TABLE II: Optimal values of (k_1, k_2, m_2, m_3) which provide the smallest N_0 in comparison with the method in [16] (using $Q = \boldsymbol{I}_6$ and $R = 0.01\boldsymbol{I}_3$ as in (16)).

We enumerate in Table II all the scenarios in which, we obtain the smallest value of the required prediction horizon N_0 by using the weighting matrices $Q = \boldsymbol{I}_6$ and $R = 0.01\boldsymbol{I}_3$. Table II also shows the corresponding values of c from (43) which helps identifying all the feasible initial states p_0, i.e., $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$ as stated in Theorem 2. All the choices of (k_1, k_2, m_2, m_3) gathered in Table II require the minimum prediction horizon around 153 $\sim$ 154 but they provide a large range of c from 0.205 to 0.287. Furthermore, using the control gains $(k_1, k_2) = (-1.2, -0.8)$ and $M = \boldsymbol{I}_6$ provides the shortest prediction horizon $N_0 = 153$ and the largest value of $c = 0.287$, hence, being the best choice in our analysis.
For comparison, we use the method proposed in [16] which employs a linear controller as well as the linearized model of the dynamics (1). The results are given in the last three lines in which both the value of c and the prediction horizon N_0 are much more conservative than the results of our proposed method. This is due to the restriction of using the linear controller for the nonlinear system (1). This strongly confirms the effectiveness of our NMPC design approach using the local FL controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ as defined in (22).

We also notice that even the shortest prediction horizon in our analysis $N_0 = 153$ steps is still extremely large for a real implementation. However, bear in mind that the obtained results (c.f. Table II) are still more promising than employing the linear controller as considered in [16]. This also indicates a big gap which still exists between the theory on NMPC design and their practical formulations since through various simulations and experimental tests, the u-NMPC controller (16)–(17) requires a prediction horizon of only 10 steps to stabilize the system (1). This is due to the fact that the u-NMPC controller can fully exploit the inputs of the system while a standard controller (e.g. FL controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ as in (22) or the linear controller as employed in [16]) enforces the inputs to follow their explicit formulations, hence, the convergence speed of the standard controller (e.g. ρ as in (35)) can not be fast enough to obtain this short prediction horizon of 10 steps. Also note that, better solutions can be found with more thorough parametrizations of the control gains (52)–(53) and of the matrix M (55). However, the trade-off is between increasing the computation time and the complexity of the optimization problem (65) (which may even cause infeasibility).

VI. Simulation results

To validate the presented results, we consider the simulation model (1) of a Crazyflie 2.0 nano-quadcopter platform [3]. It is assumed that the rotation dynamics are stabilized by a fast torque control mechanism [2]. As specified in (4), the vehicle has to respect thrust limits, with $T_{limit} = 2g$, and roll and pitch constraints, i.e., $\epsilon_c = 10°$ in (4)) which lead to the choice of $U_x = U_y = U_z = 1.0875$ which satisfy the conditions (27)-(28) as given in Tables III and IV. We fix the direction angle $\psi = 0$ and the initial state at $\boldsymbol{x}_0 = \begin{bmatrix} 0 & 0.15 & -0.1 & 0 & 0 & 0 \end{bmatrix}^\top$.
Hereinafter, we introduce first the numerical values of the designs of the two NMPC schemes, i.e., with the terminal constraint (denoted by NMPC-t) and without the terminal constraint (u-NMPC) detailed in Sections IV-A and IV-B, respectively. Their performances are analyzed and compared through simulation and experimental results. The NMPC optimization problems are implemented by using Pyomo [22] and solver IPOPT [23] in Python 3.0. The sampling time is fixed at $\Delta_t = 0.1$ seconds with Δ_t as in (2)-(3) which means that the NMPC computing time should be less than $\Delta_t = 0.1$ seconds for making possible the application to real experiments.

A. NMPC setup

1) NMPC-t: The chosen parameters of the NMPC-t control scheme (8)–(10) are gathered into Table III (c.f. Procedure 1). We choose the control gains $K_{1_q} = -5$, $K_{2_q} = -5$ for all $q \in \{x, y, z\}$ (26) so that the corresponding region $\mathcal{X}_f$ is large enough for employing the prediction horizon $N_T = 5$ and the controlled system also has a good convergence speed.

2) u-NMPC: Table IV presents the parameters of the u-NMPC control scheme (14)–(17). The values given in Table IV are the best choices according to Section V.

TABLE III: Parameters of the NMPC-t controller.

	Values
Q, R in (8)	$\boldsymbol{I}_6$, $0.1\boldsymbol{I}_3$
U_x, U_y, U_z in (27)–(29)	$1.0875,\ 1.0875,\ 1.0875$
K_{1_q}, K_{2_q}, $q \in \{x, y, z\}$ in (26)	$-5, -5$
M in (32) ($M \succeq Q^*$ (40))	$20\boldsymbol{I}_6$
P in (32)	$\begin{bmatrix} 22.02\boldsymbol{I}_3 & 10.92\boldsymbol{I}_3 \\ 10.92\boldsymbol{I}_3 & 12.68\boldsymbol{I}_3 \end{bmatrix}$
δ in (34)	0.1285
N_T in (6)	5

TABLE IV: Parameters of the u-NMPC controller.

	Values
Q, R in (8)	$\boldsymbol{I}_6$, $0.01\boldsymbol{I}_3$
U_x, U_y, U_z in (27)–(29)	$1.0875,\ 1.0875,\ 1.0875$
k_1, k_2, in (52)–(53)	$-1.2, -0.8$
M in (32)	$\boldsymbol{I}_6$
P in (32)	$\begin{bmatrix} 18.22\boldsymbol{I}_3 & 4.02\boldsymbol{I}_3 \\ 4.02\boldsymbol{I}_3 & 12.22\boldsymbol{I}_3 \end{bmatrix}$
δ in (34), ρ in (36)	5.78, 0.95
c, γ in (43)	0.29, 40.96
N_0 in (20), N_U in (14)	$N_0 = 152.22$, $N_U = 153$

B. Simulation results

We provide the simulation results for four scenarios:
Scenario 1: Stabilize the dynamics (1) with $\psi = 0$ using the NMPC-t controller detailed in (8)–(10) with the parameters gathered in Table III.
Using the u-NMPC controller given in (14)–(17) with the parameters in Table IV.
Scenario 3: Using the u-NMPC controller given in (14)–(17) with the prediction horizon of 10 steps (enough for stabilizing the system through simulations).
Scenario 4: Using the local feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ given in (22) with the control gains as in Table IV.

The results are given in Figures 3–6 where they are plotted in red, green, blue and black corresponding to the four aforementioned scenarios, respectively. Figure 3 presents the convergence of the state trajectories under simulation. Note that, only the results of the y and z axes are plotted since the system does not move along the x axis. The terminal region $\mathcal{X}_f \subset \mathbb{R}^6$ given in (33) of the NMPC-t scheme is illustrated by the yellow ellipsoid (which is obtained by slicing $\mathcal{X}_f$ along three $2D$ spaces of the position and the corresponding velocity, then, making a convex union of the three obtained $2D$ sets). All the closed-loop state trajectories under four scenarios converge to the origin but with different convergence speeds as being illustrated with more details in Figure 4 and also given in Table V. Also, the input results are shown in Figure 5 where all the proposed NMPC controllers respect the input constraints (4). We only provide the input results during the first one second for better clarity since they are all monotonously converging to the equilibrium $\boldsymbol{u}_e$ (5). Furthermore, Figure 6 and also Table V present the computing time per step of the three NMPC controllers employed under the first three scenarios.

Controller's information	Convergence time (95%) [s]	Computing time [ms]		
		Mean	Min	Max
Scenario 1	1.8	48.5	31.2	84.6
Scenario 2	3.1	243.7	209.3	298.4
Scenario 3	4.4	47.9	31.3	84.6
Scenario 4	2.1	Not applicable		

TABLE V: Comparison between different controllers.

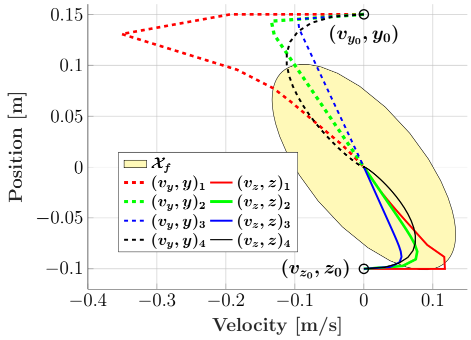


Fig. 3: Terminal region $\mathcal{X}_f$ (approximated illustration) and the trajectories $(y, v_y), (z, v_z)$ under different scenarios.

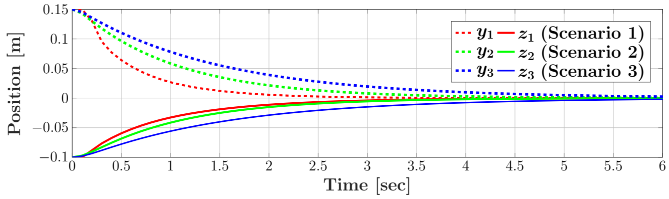


Fig. 4: Convergences of x, y, z under different scenarios.

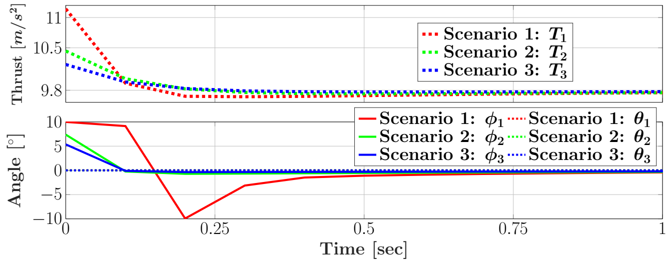


Fig. 5: Input values under different scenarios.

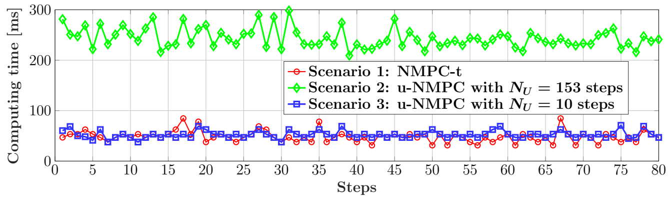


Fig. 6: Computation time under simulations.

From the presented simulation results, the NMPC-t controller employed under Scenario 1 obtains the shortest transient state (1.8 seconds as given in Table V) due to its appropriate terminal constraint set $\mathcal{X}_f$ (yellow ellipsoid in Figure 3). This is also better than using the feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ defined in (22) which provides the second shortest convergence time of 2.1 seconds. Next, when not using the terminal constraints, we can see that the u-NMPC controller under Scenario 2 (using the 153-step prediction horizon) obtains the transition time of 3.1 seconds while employing the u-NMPC controller with the prediction horizon of only 10 steps under Scenario 3 makes the states converging in longer time, i.e., 4.4 seconds. It is also trivial that the computation burden when using the 153-step prediction horizon under Scenario 2 is much heavier than those under other scenarios as can be seen from Figure 6. The average computing time of the NMPC controller with the 153-step prediction horizon is 243.7 milliseconds while it is only 47.9 second for the u-NMPC controller with the 10-step prediction horizon, being the best value in our analysis.

We also notice that the u-NMPC controller using the 153-step prediction horizon (which is found by our analysis given in Section V) is not ready for real implementation since its minimum computing time is already 209.3 milliseconds while the sampling time is fixed at 100 milliseconds. However, the proposed calculation process based on the feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ (22) already reduces significantly the required prediction horizon (153 steps) in comparison with the method using the standard linear controller introduced in [16] which requires thousands of steps to stabilize the system as mentioned in Table II.

In the next section, we show that both the NMPC-t and u-NMPC controllers considered in Scenarios 1 and 3 (using the 10-step prediction horizon) are successfully validated through experiments over a real quadcopter platform.

VII. Experimental results

This section introduces the experimental validation of the two NMPC controllers, NMPC with and without terminal constraint given in Sections IV-A and IV-B over the laboratory Crazyflie 2.0 (CF) nano-quadcopter. The CF has its inner controller running at 500 Hz to track the set-point of the thrust and the three angles sent from the station computer [2]. In the station computer, we implement the NMPC optimization problems (6) and (14) with the chosen sampling time $\Delta_t = 0.1$ seconds (fast enough to stabilize the system) by using Pyomo [22] and the IPOPT solver [23] in Python 3.0. In the following, we describe the limits of the platform and the mismatches between the theoretically nominal NMPC application and real implementation, then, illustrates the obtained results of the two proposed NMPC controllers.

A. Experimental validation limits and how to overcome them

Firstly, as introduced in Section II-A, the yaw angle ψ is assumed to be a known constant. However, maintaining a constant direction angle for an aerial vehicle is obviously impossible for long running times. Therefore, we try to stabilize the yaw angle of the CF around zero and update the actual yaw value to the NMPC optimization problem at each sampling time. This alternative approach still guarantees the nominal stability of the closed-loop scheme with a less strict assumption, i.e., requiring the yaw angle value to be constant only during the prediction horizon [6].

[6] The stability is still guaranteed since all the required ingredients detailed in Section III hold with a general constant yaw angle $\psi \in [-\pi, \pi]$.

Secondly, the execution time is always significant when considering an NMPC controller, especially for controlling the Crazyflie 2.0 quadcopter system with the required sampling time of 0.1 seconds (as illustrated in Figure 6 for simulation results). Thus, even with a perfect state feedback at time step k, we cannot obtain the MPC control action immediately at the same time step k as assumed in (10) and (17). Hence, we have to relax this assumption by introducing the obtained MPC input of time step k to the CF system at time step $k+1$.

Remark 7: If the reader is interested in using embedded NMPC with low-power hardware, a discussion on how to re-formulate the NMPC problems (6) and (14) into their approximated quadratic formulations, and thereafter, solve them by using a modified interior-point solver is detailed in [1]. Another discussion on solving a linear MPC problem with a 8-bit microcontroller by using a convex lifting method is presented in [8]. □

B. Experimental results

For experiments, we consider the two Scenarios 1 and 3 as given in Section VI with the parameters of the controllers detailed in Tables III and IV (the u-NMPC controller has its prediction horizon of 10 steps). Figure 7 presents the results of x (green lines), y (blue lines) and z (red lines) under two scenarios which shows that both the NMPC-t and u-NMPC controller succeed in stabilizing the CF quadcopter. It can be seen again that the NMPC-t provides faster convergence speed than the u-NMPC controller due to its terminal constraint (1.5 seconds vs. 3 seconds, from blue lines in Figure 7). The terminal constraint also results in more input consumption as clearly observed from Figure 8. The maximum values of the thrust and the roll angle of the NMPC-t (dashed lines) are significantly higher than the ones from the u-NMPC controller (solid lines). All the inputs respect their constraints as given in (4). The computing times under experiment of both NMPC controllers are given in Figure 9 which shows that the NMPC-t controller (with $N_T = 5$ steps and the terminal constraint set $\mathcal{X}_f$ as in (33)) requires more computation effort than the u-NMPC controller (with $N_U = 10$ steps) but both succeeds in running within the sampling time $\Delta_t = 0.1$ seconds.

VIII. Conclusion

This paper presented the application of a nonlinear feedback linearization controller for designing two different NMPC schemes, i.e., NMPC with and without terminal constraint, to stabilize the thrust-propelled vehicles dynamics with asymptotic stability guarantee. By using several standard mathematical inequalities, an ellipsoidal invariant set was constructed under the feedback linerization controller ensuring input constraint satisfaction. Within this invariant set, the feedback law was also bounded by the weighted norm of the corresponding state. These ingredients were gathered to ensure the satisfaction of the stability conditions of the two NMPC designs. The proposed control methods were validated through extensive simulations and experiments over a real quadcopter platform. Future works will analyze the mismatches on angle tracking and the delays on sending NMPC input to the controlled system as employed in the experiments.

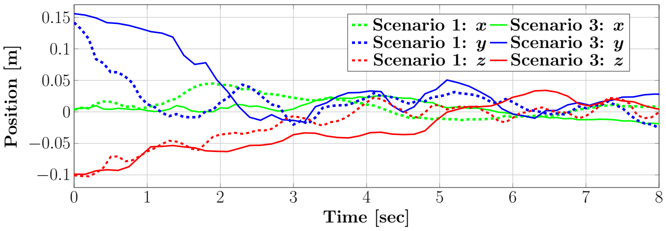


Fig. 7: Convergences of x, y, z under different scenarios.

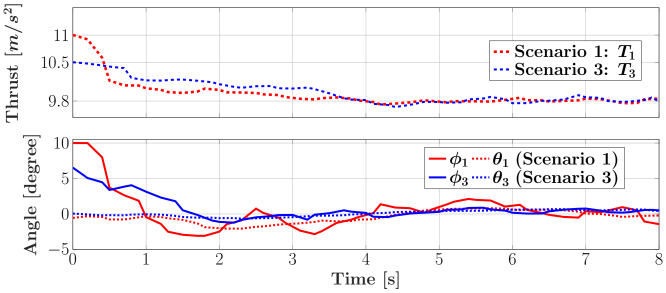


Fig. 8: Values of the input under different scenarios.

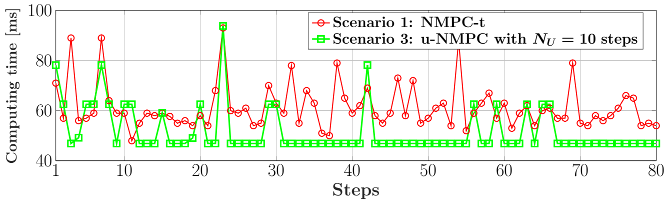


Fig. 9: Computation time under experiments using Pyomo in Python 3.0.

References

[1] A. Zanelli, G. Horn, G. Frison, and M. Diehl, “Nonlinear model predictive control of a human-sized quadrotor,” in *Proc. 2018 European Control Conference (ECC’18)*, Limassol, Cyprus, June 2018, pp. 1542–1547.

[2] N. T. Nguyen, I. Prodan, and L. Lefèvre, “Flat trajectory design and tracking with saturation guarantees: a nano-drone application,” *International Journal of Control*, pp. 1–14, Article in Press, August 2018.

[3] C. Budaciu, N. Botezatu, M. Kloetzer, and A. Burlacu, “On the evaluation of the crazyflie modular quadcopter system,” in *Proc. 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, 2019, pp. 1189–1195.

[4] N. T. Nguyen, I. Prodan, and L. Lefèvre, “Effective angular constrained trajectory generation for thrust-propelled vehicles,” in *Proc. 2018 European Control Conference (ECC’18)*, Limassol, Cyprus, June 2018, pp. 1833–1838.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[6] T. A. Badgwell and S. J. Qin, “Model-predictive control in practice,” *Encyclopedia of Systems and Control*, pp. 756–760, 2015.

[7] M. W. Mueller and R. D’Andrea, “A model predictive controller for quadrocopter state interception,” in *Proc. 2013 IEEE European Control Conference (ECC’13)*, Zürich, Switzerland, 2013, pp. 1383–1389.

[8] M. Gulan, G. Takács, N. A. Nguyen, S. Olaru, P. Rodriguez-Ayerbe, and B. Rohal’-Ilkiv, “Embedded linear model predictive control for 8-bit microcontrollers via convex lifting,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 697–10 704, 2017.

[9] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[10] S. Munir, M. Hovd, and S. Olaru, “Low complexity constrained control using higher degree lyapunov functions,” *Automatica*, vol. 98, pp. 215–222, 2018.

[11] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[12] M. Cannon, V. Deshmukh, and B. Kouvaritakis, "Nonlinear model predictive control with polytopic invariant sets," *Automatica*, vol. 39, no. 8, pp. 1487–1494, 2003.

[13] L. Grüne, "NMPC without terminal constraints," *4th IFAC Nonlinear Model Predictive Control Conference (NMPC 2012). IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 1–13, 2012.

[14] J. C. Butcher and G. Wanner, "Runge-kutta methods: some historical notes," *Applied Numerical Mathematics*, vol. 22, no. 1-3, pp. 113–151, 1996.

[15] H. Lu, C. Liu, L. Guo, and W.-H. Chen, "Constrained anti-disturbance control for a quadrotor based on differential flatness," *International Journal of Systems Science*, vol. 48, no. 6, pp. 1182–1193, 2017.

[16] J. Köhler, M. A. Müller, and F. Allgöwer, "Nonlinear reference tracking: An economic model predictive control perspective," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 254–269, 2018.

[17] D. Limón, T. Alamo, F. Salas, and E. F. Camacho, "On the stability of constrained mpc without terminal constraint," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 832–836, 2006.

[18] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer-Verlag, London, 2011.

[19] M. Reble and F. Allgöwer, "Unconstrained model predictive control and suboptimality estimates for nonlinear continuous-time systems," *Automatica*, vol. 48, no. 8, pp. 1812–1817, 2012.

[20] N. T. Nguyen, I. Prodan, and L. Lefèvre, "A stablizing nmpc design for thrust-propelled vehicles dynamics via feedback linearization," in *Proc. 2019 American Control Conference (ACC19)*, Philadelphia, USA, July 2019, pp. 2909–2914.

[21] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2005, vol. 3.

[22] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.

[23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

Appendix A
Proof of Proposition 1

By applying our previous results presented in [2], [4] to the nonlinear feedback laws (22b)-(22c), we have the following relations:

$$|\phi_{\text{loc}}(\boldsymbol{x}_k)|, |\theta_{\text{loc}}(\boldsymbol{x}_k)| \leq \epsilon, \ \forall \psi \in [-\pi, \pi], \tag{66}$$

where ϵ is defined as:

$$\epsilon = \arctan\left(\sqrt{\frac{u_x^2(\boldsymbol{x}_k) + u_y^2(\boldsymbol{x}_k)}{(u_z(\boldsymbol{x}_k) + g)^2}}\right). \tag{67}$$

Next, as detailed in Appendix B, we have that:

$$\begin{aligned}(T_{\text{loc}}(\boldsymbol{x}_k) - g)^2 \leq & \sum_{q \in \{x,y,z\}} \left(K_{1_q} q_k + K_{2_q} v_{q,k}\right)^2, \qquad (68)\\ & = \boldsymbol{x}_k^\top K^\top K \boldsymbol{x}_k.\end{aligned}$$

Furthermore, by applying $\arctan(p) \leq p, \ \forall p \geq 0$ to (67) and then, combining this result with (66), we arrive to:

$$\phi_{\text{loc}}^2(\boldsymbol{x}_k) \leq \frac{\sum_{q \in \{x,y\}} \left(K_{1_q} q_k + K_{2_q} v_{q,k}\right)^2}{(-U_z + g)^2} = \|\boldsymbol{x}_k\|_\Gamma^2, \tag{69}$$

in which the denominator is due to $|u_z(\boldsymbol{x}_k)| \leq U_z < g$. A similar result is obtained for $\theta_{\text{loc}}^2(\boldsymbol{x}_k)$. Finally, from (68)-(69), we have that:

$$\begin{aligned}\|\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) - \boldsymbol{u}_e\|^2 &= (T_{\text{loc}}(\boldsymbol{x}_k) - g)^2 + \phi_{\text{loc}}^2(\boldsymbol{x}_k) + \theta_{\text{loc}}^2(\boldsymbol{x}_k)\\ &\leq \boldsymbol{x}_k^\top (K^\top K + 2\Gamma)\boldsymbol{x}_k. \qquad (70)\end{aligned}$$

Appendix B
Proof of $(T_{\text{LOC}} - g)^2 \leq u_x^2 + u_y^2 + u_z^2$

By considering T_{loc} in (22a), we have that:

$$\begin{aligned}(T_{\text{loc}} - g)^2 =& u_x^2 + u_y^2 + u_z^2 \qquad (71)\\ & + 2g\left(u_z + g - \sqrt{u_x^2 + u_y^2 + (u_z + g)^2}\right).\end{aligned}$$

Next, if $u_z + g > 0$, we have that:

$$\sqrt{u_x^2 + u_y^2 + (u_z + g)^2} \geq |u_z + g| = u_z + g, \tag{72}$$

otherwise, if $u_z + g \leq 0$, it is straightforward to obtain:

$$u_z + g - \sqrt{u_x^2 + u_y^2 + (u_z + g)^2} \leq 0. \tag{73}$$

Thus, introducing (72) and (73) to (71) leads to:

$$(T_{\text{loc}} - g)^2 \leq u_x^2 + u_y^2 + u_z^2. \tag{74}$$

This result holds regardless the sign of $u_z + g$.

Ngoc Thinh Nguyen received the Engineering Degree in Mechatronics from the Hochiminh city University of Technology, Vietnam, in 2016 and the PhD in Control Engineering at the Laboratory of Conception and Integration of Systems (LCIS EA 3747), Valence, France, in 2019.
He is currently a post-doc researcher at Institute for Robotics and Cognitive Systems, Luebeck, Germany. His research interests include modeling and control for unmanned aerial vehicles (UAVs), optimal control and optimal trajectory generation.

Ionela Prodan received the B.S. degree from the Univ. Politehnica of Bucharest, Romania in 2009 and her PhD in Control Engineering from Supélec, Gif-sur-Yvette, France in 2012. She continued with a post-doctoral fellowship within the Chair on Systems Science and the Energetic Challenge - École Centrale Paris, France. From 2014 she is an Associated Professor at Univ. Grenoble Alpes, Laboratory of Conception and Integration of Systems (LCIS of Grenoble INP), Valence, France. Her research interests are multi-disciplinary with a core expertise in control and applied mathematics. These are encompassing constrained optimization-based control (Model Predictive Control via distributed and decentralized approaches), mixed-integer programming, set-theoretic methods and their applications to motion planning for autonomous vehicles and energy management in building-scale microgrids.

Laurent Lefèvre got his BSc from Ecole Polytechnique de Bruxelles (ULB, 1991) and his MSc in applied mathematics from Ecole Polytechnique de Louvain (UCL, 1994), both in Belgium. Then he moved to Lille (North of France) and got his PhD in control from Ecole Centrale de Lille (USTL, 1999). Since 2011 he got a full professor position with the Laboratoire de Conception et d'Intégration des Systèmes (at Grenoble Institute of Technology). His current domains of interest are numerical methods for control, distributed parameters systems and nonlinear control. He is involved in various project related to applications in water management, energy management for green buildings, microgrids and plasma physics.