J Syst Sci Complex (2021) 34: 1379–1400

Nonlinear Model Predictive Control-Based Guidance Algorithm for Quadrotor Trajectory Tracking with Obstacle Avoidance*

ZHAO Chunhui · WANG Dong · HU Jinwen · PAN Quan

DOI: 10.1007/s11424-021-0316-9
Received: 15 December 2020 / Revised: 27 March 2021

Abstract This paper studies a novel trajectory tracking guidance law for a quadrotor unmanned aerial vehicle (UAV) with obstacle avoidance based on nonlinear model predictive control (NMPC) scheme. By augmenting a reference position trajectory to a reference dynamical system, the authors formulate the tracking problem as a standard NMPC design problem to generate constrained reference velocity commands for autopilots. However, concerning the closed-loop stability, it is difficult to find a local static state feedback to construct the terminal constraint in the design of NMPC-based guidance law. In order to circumvent this issue, the authors introduce a contraction constraint as a stability constraint, which borrows the ideas from the Lyapunov's direct method and the backstepping technique. To achieve the obstacle avoidance extension, the authors impose a well-designed potential field function-based penalty term on the performance index. Considering the practical application, the heavy computational burden caused by solving the NMPC optimization problem online is alleviated by using the dynamical adjustment of the prediction horizon for the real-time control. Finally, extensive simulations and the real experiment are given to demonstrate the effectiveness of the proposed NMPC scheme.

Keywords Backstepping, input constraints, nonlinear model predictive control, obstacle avoidance, quadrotor UAV, trajectory tracking.

ZHAO Chunhui · WANG Dong (Corresponding Author) · HU Jinwen · PAN Quan
School of Automation, Northwestern Polytechnical University, Xi'an 710072, *China.* Email: zhaochunhui@nwpu.edu.cn; wangdongwarden@mail.nwpu.edu.cn; hujinwen@nwpu.edu.cn; quanpan@nwpu.edu.cn.
*This research was supported by the National Natural Science Foundation of China under Grant No. 62073264, in part by Key Research and Development Project of Shaanxi Province under Grant Nos. S2021-YF-ZDCXL-ZDLGY-0027, 2020ZDLGY06-02, and Fundamental Research Funds for the Central Universities under Grant No. 3102019ZDHKY02, in part by Aeronautical Science Foundation of China under Grant No. 2019ZA053008, in part by Natural Science Foundation for Young Scholars of China under Grant No. 61803309 and Science Foundation for Post Doctorate Research Grant No. 2018M633574.
◇ *This paper was recommended for publication by Editor GUO Jin.*

1 Introduction

In recent years, the applications of quadrotor unmanned aerial vehicles (UAVs) have significantly widened for diverse areas, such as aerial surveillance, transportation and power lines' maintenance[1]. Various autonomous commercial platforms appear on the market and become preferable, such as DJI and Parrot[2, 3]. For such ready-to-fly platforms, since the low-level autopilots are able to follow the given velocity references, users only need to design a guidance system (guidance law) in the outer loop that accesses position and velocity data from onboard sensors and sends guidance signals (velocity references signals) to the autopilot for transnational motion. Although these platforms facilitate the control system design, they also subject to the system constraints, e.g., the allowed maximum magnitude of input commands for autopilots and the limited computing power of onboard processors. Therefore, the motion controller design for such quadrotors up to today is stilling challenging.

Among diverse applications, trajectory tracking control, as one of the most critical motion control problems, has been receiving a growing interest. Linearization methods, such as feedback linearization[4] or Jacobian linearization[5], provide useful tools to cope with the nonlinearities. However, the main contradiction is that it requires accurate system model parameters, while they are rarely identified accurately. In order to tackle the parametric uncertainties, nonlinear control techniques are extensively resorted. In [6], a sliding mode controller is combined with a nonlinear extended state observer for the robust finite-time tracking control of wheeled mobile robots. This kind of combination is also applied to the quadrotor high stabilization flight control[7]. The Lyapunov-basd backstepping control (LBSC) is a mainstream method for tracking control of quadrotors because it can achieve the asymptotically stability subject to bounded disturbances[8]. In order to ensure the flight safety in constrained environments such as [2, 9], some collision avoidance strategies have been combined with these approaches for the integrated trajectory tracking and collision avoidance[10, 11]. Although the prescribed performance tracking control method in [12] addresses the performance constraints, it as well as the preceding control methods rarely consider input constraints explicitly in practical applications. Input constraints are often discarded in the controller design phase and then treated by heuristics, e.g., in terms of "saturation"and "antiwindup". In [13], the authors introduced a saturation function-based auxiliary system to counteract the input saturation in the design of SMC for attitude tracking. However, these heuristics don't consider the performance index with respect to power consumption explicitly and may result in poor control performances, and at least lose a certain degree of "optimality". Therefore, it is essential to design a trajectory tracking control scheme accounting for input constraints while achieving acceptable tracking performance.

An efficient alternative is model predictive control (MPC) because of its capability of handling control constraints systematically. The flexibility in formulating control problems allows MPC to digest any nonlinearity or time-varying characteristics of system models[14]. The salient features of handling system constraints and ensuring prescribed performance index make MPC widely used in chemical industry[15], industrial manipulators[16], aerospace engineering[17], mo-

bile robots[18]. Such an advanced control approach gradually becomes an appealing option for quadrotor guidance and control problems[3, 19]. In [3], a robust MPC law is design for an AR.Drone quadrotor for the waypoint tracking problem with the input constraint. In [19], flatness-based MPC for quadrotor trajectory tracking is proposed for balancing tracking performance and constraint satisfaction with fast computation. However, the classical MPC strategy for the motion control problem of quadrotor UAVs is quite limited because it is hard to find a local continuous time-invariant state feedback to design its terminal constraint for closed-loop stability by using linear matrix inequalities[20], or a feedback controller by using linearization is hard to be implemented in engineering[21]. Although the nonlinear MPC (NMPC) strategy with the large prediction horizon in [22] can remove the requirement for the terminal constraint, it greatly increases the computational complexity of solving NMPC optimization problem. Therefore, it is necessary to find a novel NMPC formulation that can make the tradeoff between specifying the maximum prediction horizon length without destabilizing the tracking control. In the NMPC framework, the most general treatments for collision avoidance rely on the collision avoidance constraint[23, 24]. Since the recursive feasibility is not considered in these works, these treatments may result in unfeasible problems.

Motivated by the above considerations, in order to handle the aforementioned issues, we aim to propose a stable trajectory-tracking guidance law based on a novel NMPC scheme. This guidance law generates the constrained reference velocity commands for autopilots to follow given reference trajectories. The specific contributions of this paper are as follows:

1) A novel NMPC optimization problem formulation for the integrated trajectory tracking and obstacle avoidance guidance law is proposed such that we can exploit the kinematic properties of the UAV motion under the reference velocity constraint. The obstacle avoidance capability is achieved by imposing a well-designed penalty term on the performance index, which is inspired by the artificial potential field approach.

2) For the quadrotor system, we introduce a stability constraint in our proposed NMPC-based trajectory tracking guidance law to guarantee the closed-loop stability and achieve the acceptable tracking performance. This stability constraint is derived from a time-varying state feedback auxiliary controller, which borrows the ideas from Lyapunov's direct method and backstepping approach.

3) The proposed NMPC scheme in essence provides a tradeoff between control performance and computational burden, which introduces an effective mechanism for reducing the computational burden with acceptable tracking performance. Extensive simulations and the experiment demonstrate its prominent performance.

The remainder of this article is organized as follows: In Section 2, the quadrotor UAV model is described and the trajectory tracking and obstacle avoidance problem is formulated. Section 3 presents the regularized NMPC optimization problem for trajectory tracking and obstacle avoidance. In Section 4, the auxiliary controller development and theoretical analysis of recursive feasibility and closed-loop stability are detailed. In Section 5, simulation and experimental results are provided to demonstrate the effectiveness and robustness of the proposed NMPC algorithm. In Section 6, some conclusive remarks and future work are discussed.

Notations: The absolute value is denoted by $|\cdot|$, the Euclidean norm is denoted by $||\cdot||$ and the infinity norm is denoted by $||\cdot||_\infty$. $||\boldsymbol{x}||_P^2$ denotes the weighted norm $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}$, where $\boldsymbol{P}$ is a positive-definite matrix. $\dot{\boldsymbol{\eta}}$ and $\ddot{\boldsymbol{\eta}}$ denote the derivative and the second derivative of $\boldsymbol{\eta}$ with respect to time. The column operation $[\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}, \cdots, \boldsymbol{x}_n^{\mathrm{T}}]^{\mathrm{T}}$ is written as $\mathrm{col}(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$. The superscript "T" represents the transposition, $\mathrm{diag}(\cdot)$ denotes the diagonal operation and $\max\{\cdot\}$ is the maximum elements of an array. The internal variables in the NMPC controller are denoted by a hat $(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{u}})$ to indicate the predicted values.

2 Preliminaries and Problem Formulation

2.1 Quadrotor UAV Modeling

For the translational motion of a quadrotor UAV, its kinematic equations can be described with the yaw attitude as follows[25]:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{v}_b, \tag{1}$$

where $\boldsymbol{\eta} = [x, y, z, \psi]^{\mathrm{T}}$ is the position and yaw vector in the inertial frame, $\boldsymbol{v}_b = [u, v, w, r]^{\mathrm{T}}$ is the velocity and yaw rate vector in the body frame,

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is the rotation matrix from the body frame to the inertial frame. In the body frame, the pitch and roll are considered to be neglectable. u and v are lying in xoy plane, where u is pointing to the front of the quadrotor and v to the left. w is aligned with the inertial frame z.

For commercial quadrotors, the measurements for the force, torque and rotor speed are unavailable in the closed-source configuration so that the Euler-Lagrange equations can not analyze the cause of the motion. Its relationship between the velocity responses and the reference velocity commands in the forward direction u_x, sidewards u_y, upwards u_z and the yaw angular velocity around the z-axis u_ψ is approximated by a following linear state space model via a precise parameter identification[23]:

$$\dot{\boldsymbol{v}}_b = \boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}\boldsymbol{u}, \tag{2}$$

where $\boldsymbol{u} = [u_x, u_y, u_z, u_\psi]^{\mathrm{T}}$, $\boldsymbol{S} = \mathrm{diag}([a_s, b_s, c_s, d_s])$, $\boldsymbol{F} = \mathrm{diag}([a_f, b_f, c_f, d_f])$, $a_s = -\frac{1}{\tau_x}$, $b_s = -\frac{1}{\tau_y}$, $c_s = -\frac{1}{\tau_z}$, $d_s = -\frac{1}{\tau_\psi}$, $a_f = \frac{K_x}{\tau_x}$, $b_f = \frac{K_y}{\tau_y}$, $c_f = \frac{K_z}{\tau_z}$, $d_f = \frac{K_\psi}{\tau_\psi}$, $K_x, K_y, K_z, K_\psi > 0$ are gains and $\tau_x, \tau_y, \tau_z, \tau_\psi > 0$ are time constants in the corresponding first-order transfer functions. Based on these considerations, we establish the following dynamic model for a quadrotor UAV trajectory tracking task:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{R}(\psi)\boldsymbol{v}_b \\ \boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}\boldsymbol{u} \end{bmatrix} \triangleq \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{3}$$

where $\boldsymbol{x} = \mathrm{col}(\boldsymbol{\eta}, \boldsymbol{v}_b) \in \mathbb{R}^8$ denotes the motion state and $\boldsymbol{u} \in \mathbb{R}^4$ is the control input.

2.2 Problem Formulation

To better describe the control objective, we now formulate the integrated trajectory tracking and obstacle avoidance problem. It is assumed that the environmental structure and obstacle location have been obtained by using some exploration algorithms with the onboard sensors, and thus a safe parameterized path $\mathcal{S}(\theta) = [x_{rr}(\theta), y_{rr}(\theta), z_{rr}(\theta)]^{\mathrm{T}}$ with the path parameter θ can be generated. Readers with an interest in the environment modeling can take reference to [26]. In order to keep the quadrotor in a prescribed safe distance r_s from the current quadrotor position $\boldsymbol{p}(t) = [(x(t), y(t), z(t)]^{\mathrm{T}}$ to the obstacle position $\boldsymbol{p}_o$ and track the safe time-parameterized reference trajectory $\boldsymbol{\eta}_r(t) = [x_r(t), y_r(t), z_r(t), \psi_r(t)]^{\mathrm{T}}$ generated by $\mathcal{S}(\theta)$, the real trajectory $\boldsymbol{\eta}(t)$ and the input $\boldsymbol{u}(t)$ of the quadrotor system are driven to satisfy:

- Forward direction: The quadrotor moves along the trajectory in forward direction, e.g., $\dot{x}_r(t) > 0$;
- Trajectory convergence: $\boldsymbol{\eta}(t)$ converges to the reference trajectory $\boldsymbol{\eta}_r$, e.g., $\lim_{t\to\infty} ||\boldsymbol{\eta}(t) - \boldsymbol{\eta}_r(t)|| = 0$;
- Input constraint satisfaction: For all $t \in [t_0, +\infty)$, the constraint on the input, e.g., $||\boldsymbol{u}(t)||_\infty \leq u_{\max}$ is satisfied, where $u_{\max}$ is the maximum magnitude;
- Obstacle avoidance: $||\boldsymbol{p}(t) - \boldsymbol{p}_o|| \geq r_s$.

3 Nonlinear Model Predictive Tracking Control

In this section, a reference trajectory augmentation is firstly generated. Next, the NMPC optimization problem for the trajectory tracking and obstacle avoidance is formulated, where the closed-loop stability is guaranteed.

3.1 Reference Trajectory Augmentation Generation

The reference path $\mathcal{S}(\theta)$ describes the desired position in the inertial frame. In general, θ is time-dependent. The time-parameterized reference position trajectory $\boldsymbol{p}(t)$ is generated based on a predetermined timing law $\theta(t) = v_t t$, e.g., $x_r(t) = x_{rr}(v_t t), y_r(t) = y_{rr}(v_t t), z_r(t) = z_{rr}(v_t t)$, where $v_t > 0$ denotes the velocity of the path progress in forward direction. To avoid singularities in the reference trajectory, we make the following assumption.

Assumption 3.1 The reference trajectory $\boldsymbol{p}(t)$ and its derivatives are bounded, satisfying that: $|x_r(t)| \leq \overline{x}$, $|y_r(t)| \leq \overline{y}$, $|z_r(t)| \leq \overline{z}$, $|\dot{x}_r| \leq \overline{x}_1$, $|\dot{y}_r| \leq \overline{y}_1$, $|\dot{z}_r| \leq \overline{z}_1$, $|\ddot{x}_r| \leq \overline{x}_2$, $|\ddot{y}_r| \leq \overline{y}_2$, $|\ddot{z}_r| \leq \overline{z}_2$.

For a continuous and smooth reference path, Assumption 3.1 is easy to be satisfied. Together with $\boldsymbol{p}(t)$ we have the reference augmentation as follows: Let $\boldsymbol{x}_r(t) = \mathrm{col}(\boldsymbol{\eta}_r(t), \boldsymbol{v}_{br}(t))$ with

$$\begin{aligned} \boldsymbol{\eta}_r(t) &= [x_r(t), y_r(t), z_r(t), \psi_r(t)]^{\mathrm{T}}, \\ \boldsymbol{v}_{br}(t) &= [u_r(t), v_r(t), w_r(t), r_r(t)]^{\mathrm{T}}, \end{aligned} \tag{4}$$

where

$$\psi_r(t) = \text{atan2}\{\dot{y}_r(t), \dot{x}_r(t)\},$$
$$u_r(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)}, \quad v_r(t) = 0, \quad w_r(t) = \dot{z}_r(t), \quad r_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)},$$

atan2$\{\cdot\}$ is the four-quadrant inverse tangent operator. The designed reference augmentation system, on the one hand, is restrained by the nonholonomic constraint such that the quadrotor modeled as the wheeled robots-like structure has smooth curve trajectory. On the other hand, it ensures that the reference state $\boldsymbol{x}_r(t) = \text{col}(\boldsymbol{\eta}_r(t), \boldsymbol{v}_{br}(t))$ fulfills the kinematic property of the UAV motion, that is, $\dot{\boldsymbol{\eta}}_r = \boldsymbol{R}(\psi_r)\boldsymbol{v}_{br}$. Therefore, each state of the dynamic model has a unique reference such that the singularity can be avoided.

Remark 3.1 The reference trajectory plays a momentous role in the trajectory tracking. Note that an appropriate timing law of $\theta(t)$ may be carefully designed such that the reference augmentation chosen in (4) fits the realistic constraints. Moreover, v_t decides the forward velocity of the quadrotor in the body frame. If v_t is small, the forward speed is slow so that the quadrotor spends more time to traverse the path. Instead, if it is large, it is hard for the quadrotor to converge to the path within a limited distance considering the initial position error, which may lead to an unsafe result. The influence of this parameter on trajectory tracking will be discussed in the simulation.

3.2 Obstacle Avoidance

Existing works on MPC-based trajectory tracking control rarely consider obstacle avoidance[3, 19]. Inspired by [10, 11, 25], we perform the switching behavior between obstacle avoidance and trajectory tracking by imposing the following cost function:

$$J^{oa}(\boldsymbol{x}(t)) = \begin{cases} \lambda, & ||\boldsymbol{p}(t) - \boldsymbol{p}_o|| < r_s \quad \text{(obstacle avoidance)}, \\ 0, & ||\boldsymbol{p}(t) - \boldsymbol{p}_o|| \geq r_s \quad \text{(trajectory tracking)}, \end{cases} \tag{5}$$

where $\lambda > 0$ is a well-tuned parameter according to the other goals in the system cost. Accordingly, λ is chosen with a big enough value for the sake of dominating the obstacle avoidance cost J^{ca} so that obstacle avoidance is prioritized over trajectory tracking. However, the lack of continuous differentiability at the condition border results in the non-differentiability of the cost function. This problem can be tackled by using the following continuous potential field function to approximate the obstacle avoidance behavior:

$$J^{oa}(\boldsymbol{x}(t)) \approx \rho(t) = \frac{\lambda}{1 + \mathrm{e}^{-k \cdot d(t)}}, \tag{6}$$

where $d(t) = r_s^2 - (\boldsymbol{p}(t) - \boldsymbol{p}_o)^{\mathrm{T}}(\boldsymbol{p}(t) - \boldsymbol{p}_o)$, $k > 0$ is a positive constant that denotes the smoothness of the obstacle avoidance trajectory. For the potential field function, the closer distance between the obstacle and the quadrotor results in the higher potential value; otherwise, it has the lower value, which can be omitted compared with the trajectory tracking goal in the cost function.

Remark 3.2 As the k value increases, the obstacle avoidance trajectory becomes more sharp, which results in that the system becomes more ill-conditioned and the optimization problem is more difficult to be solved. Therefore, k should be well-tuned according to the different environmental situations.

3.3 New Regularized NMPC Formulation

The relationship between the closed-loop state responses and the MPC control actions are expressed implicitly, which may not explicitly indicate the stability property like Lyapunov-based nonlinear controllers. Furthermore, due to the difference between predicted and closed-loop responses, there is no guarantee that an MPC controller based on a finite-horizon cost will achieve stability in the closed-loop operation[27]. In other words, the switching behavior from the avoidance to tracking may probably cause the unstability. Therefore, we regularize the NMPC optimization problem for the integrated trajectory tracking and obstacle avoidance and introduce a contraction constraint as a stability constraint to guarantee the closed-loop stability[28]. Based on the generated reference augmentation, a new regularized finite-horizon NMPC formulation ($\mathcal{P}_0(\boldsymbol{x})$) for the quadrotor trajectory tracking and obstacle avoidance guidance at time t_k is established as follows:

$$\mathcal{P}_0(\boldsymbol{x}): \min_{\widehat{\boldsymbol{u}}(\cdot)} J = \int_{t_k}^{t_k+T} ||\widetilde{\boldsymbol{x}}(s;t_k)||^2_{\boldsymbol{Q}} + J^{oa}(\widehat{\boldsymbol{x}}(s;t_k)) + ||\widehat{\boldsymbol{u}}(s;t_k)||^2_{\boldsymbol{R}}\, ds \tag{7a}$$

$$\text{s.t.} \quad \dot{\widehat{\boldsymbol{x}}}(s;t_k) = \boldsymbol{f}(\widehat{\boldsymbol{x}}(s;t_k), \widehat{\boldsymbol{u}}(s;t_k)), \quad s \in [t_k, t_k+T], \tag{7b}$$

$$\widehat{\boldsymbol{x}}(t_k;t_k) = \boldsymbol{x}(t_k), \tag{7c}$$

$$||\widehat{\boldsymbol{u}}(s;t_k)||_\infty \le u_{\max}, \quad s \in [t_k, t_k+T], \tag{7d}$$

$$\frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\widehat{\boldsymbol{x}}(t_k;t_k), \widehat{\boldsymbol{u}}(t_k;\widehat{\boldsymbol{x}}(t_k;t_k))) \le \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\widehat{\boldsymbol{x}}(t_k;t_k), \boldsymbol{h}(\widehat{\boldsymbol{x}}(t_k;t_k))), \tag{7e}$$

where $\widetilde{\boldsymbol{x}} = \widehat{\boldsymbol{x}} - \boldsymbol{x}_r$ is the predicted error state, T is the prediction horizon, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the positive-definite weighting matrices that define the cost. $\boldsymbol{h}(\boldsymbol{x})$ is the Lyapunov-based nonlinear tracking control law, $V(\boldsymbol{x})$ is the corresponding Lyapunov function.

Note that the performance index J includes integral operations and the constraints encompass derivative operations. Both the integral and derivative operations are performed numerically under the assumption that the discretization is stable with sufficient accuracy. The prediction horizon is divided into N steps. The step size δ is decided by $\delta = T/N$, which is interpreted as the sampling period. For ($\mathcal{P}_0(\boldsymbol{x})$), the discretized version ($\mathcal{P}_1(\boldsymbol{x})$) is solved at each sampling time t_k:

$$\mathcal{P}_1(\boldsymbol{x}) \quad \min_{\widehat{\boldsymbol{u}}_i} J = \sum_{i=0}^{N} (||\widetilde{\boldsymbol{x}}_i||^2_{\boldsymbol{Q}} + ||\widehat{\boldsymbol{u}}_i||^2_{\boldsymbol{R}} + J^{oa}(\widehat{\boldsymbol{x}}_i))\delta \tag{8a}$$

$$\text{s.t.} \quad \widehat{\boldsymbol{x}}_{i+1} = \widehat{\boldsymbol{x}}_i + f(\widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{u}}_i)\delta, \tag{8b}$$

$$\widehat{\boldsymbol{x}}_0 = \boldsymbol{x}(t_k), \tag{8c}$$

$$\boldsymbol{c}_{\text{in}}(\widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{u}}_i) \le \boldsymbol{0}, \tag{8d}$$

where $\boldsymbol{c}_{\text{in}}(\widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{u}}_i) \leq \boldsymbol{0}$ stands for the vector-valued inequality constraint including the input constraint and the stability constraint. The UAV system model (3) is discretized and denoted by $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + f(\boldsymbol{x}_i, \boldsymbol{u}_i)\delta$ with $\boldsymbol{x}_i = \boldsymbol{x}(i\delta)$ and $\boldsymbol{u}_i = \boldsymbol{u}(i\delta)$.

From (7e), it can be seen that the proposed NMPC scheme inherits the stability property of $h(\boldsymbol{x})$, which introduces the flexibility between control performance and computational efficiency. Obviously, it allows for specifying the prediction horizon without destabilizing the closed-loop system. In other words, it essentially improves the control performance and further reduces the computational complexity. Rather the prediction horizon can be adjusted properly in order to reduce the size of the NMPC optimization for the real-time control if the control algorithm is running on the low-cost onboard processor. Therefore, it is promising for such processors to perform the proposed NMPC. By virtue of the feedback of the computation time t_p for each iteration in real time, this adjustment is integrated into the NMPC scheme. The proposed NMPC algorithm is briefly described in Algorithm 1, where ΔN is the prediction horizon increment (or decrement), $\omega \in (0, 1)$ is the well-tuned weighting coefficient. The schematic illustration of the NMPC-based trajectory tracking guidance is shown in Figure 1.

Algorithm 1 NMPC Algorithm

1: Input the objective function J in (8)
2: Receive the measured state $\boldsymbol{x}(t_k)$ from the onboard sensors
3: Solve the optimization problem $(\mathcal{P}(\boldsymbol{x}))$ with $\widehat{\boldsymbol{x}}_0 = \boldsymbol{x}(t_k)$ and generate the (sub-) optimal solution $\boldsymbol{U}^* = [\widehat{\boldsymbol{u}}_0^*, \widehat{\boldsymbol{u}}_1^*, \cdots, \widehat{\boldsymbol{u}}_{N-1}^*]$
4: Record the computational time t_p. If $t_p > \delta$ then set $N = N - \Delta N$; else if $t_p < \omega\delta$ then set $N = N + \Delta N$; else $N = N$
5: Implement the control input $\boldsymbol{u}(t) = \widehat{\boldsymbol{u}}_0^*$ for one sampling period
6: $k = k + 1$, $t_{k+1} = t_k + \delta$, go to Step 2.

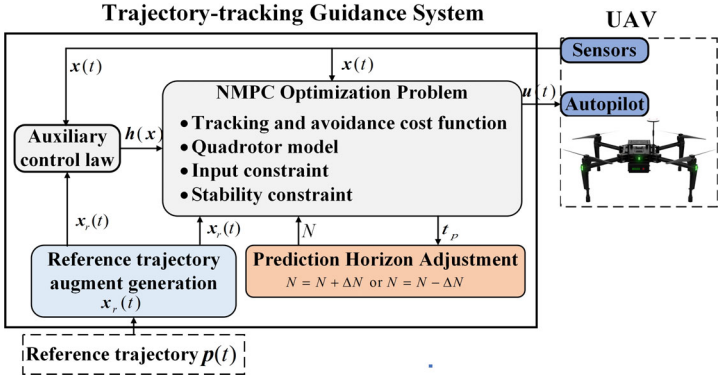

Figure 1 NMPC-based guidance system diagram

Remark 3.3 As far as the nonlinear dynamics are concerned, the optimization problem presents generic nonlinear programming problems, which is solved by the iterative optimization

algorithm SQP in numerical simulations and the efficient numerical algorithm C/GMRES in engineering practice. Since the best guaranteed solution to (7) is a local optimum, the suboptimal solutions are acceptable. Moreover, for the optimal solutions, their quality can be improved by prolonging the prediction horizon, which increases the computational burden accordingly. Owing to the stability constraint, the compatibility with suboptimal solutions introduces the flexibility between computational efficiency and control performance. For low-cost commercial platforms with limited computational resource, we can cut down the prediction horizon length for real-time control without destabilizing the tracking control.

4 Closed-Loop System Stability

The improved NMPC formulation for trajectory tracking and obstacle avoidance control of a quadrotor is presented above. Subsequently, we will discuss the design of the auxiliary state feedback control law $\boldsymbol{h}(\boldsymbol{x})$ for the stability constraint. The closed-loop stability is analyzed theoretically.

4.1 Auxiliary State Feedback Control Law Design

In the framework of the traditional MPC, a feasible input sequence candidate is composed of the optimized input sequence without the first element at the last time instant and a local stabilizing law. If an initial feasible solution is assumed, the recursive feasibility can be proved by inductive principle[29]. For $(\mathcal{P}_0(\boldsymbol{x}))$, it admits recursive feasibility if the auxiliary control law $\boldsymbol{h}(\boldsymbol{x})$ can be treated as a feasible solution under the condition that $||\boldsymbol{h}(\boldsymbol{x})||_\infty \leq u_{\max}$ can hold. In fact, we can construct $\boldsymbol{h}(\boldsymbol{x})$ by using any Lyapunov-based controller design technique. Considering the Lyapunov-based backstepping controller is extensively used in quadrotor systems, we construct $\boldsymbol{h}(\boldsymbol{x})$ by using the backstepping technique.

Consider a Lyapunov candidate as follows:

$$V_1 = \frac{1}{2}\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1, \tag{9}$$

where $\boldsymbol{z}_1 = \boldsymbol{\eta}_r - \boldsymbol{\eta}$ is the position error. Then,

$$\dot{V}_1 = \boldsymbol{z}_1^{\mathrm{T}}\dot{\boldsymbol{z}}_1 = \boldsymbol{z}_1^{\mathrm{T}}[\dot{\boldsymbol{\eta}}_r - R(\psi)\boldsymbol{v}_b] + \alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - \alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1, \tag{10}$$

where $\alpha_1 > 0$ is one of the LBSC parameters.

Further define the velocity error in the inertial frame

$$\boldsymbol{z}_2 \triangleq \dot{\boldsymbol{\eta}}_r - R(\psi)\boldsymbol{v}_b + \alpha_1\boldsymbol{z}_1 = \dot{\boldsymbol{z}}_1 + \alpha_1\boldsymbol{z}_1. \tag{11}$$

Then,

$$\begin{aligned}\dot{\boldsymbol{z}}_2 &= \ddot{\boldsymbol{\eta}}_r - \dot{R}(\psi)\boldsymbol{v}_b - R(\psi)\dot{\boldsymbol{v}}_b + \alpha_1\dot{\boldsymbol{z}}_1\\ &= \ddot{\boldsymbol{\eta}}_r - \dot{R}(\psi)\boldsymbol{v}_b - R(\psi)\dot{\boldsymbol{v}}_b + \alpha_1(\boldsymbol{z}_2 - \alpha_1\boldsymbol{z}_1).\end{aligned} \tag{12}$$

Exploiting (10) with (11), we have

$$\dot{V}_1 = \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_2 - \alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1. \tag{13}$$

The Lyapunov candidate is chosen as

$$V = V_1 + \frac{1}{2}\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_2. \tag{14}$$

Substituting (13) into (15) yields:

$$\begin{aligned}\dot{V} &= \dot{V}_1 + \boldsymbol{z}_2^{\mathrm{T}}\dot{\boldsymbol{z}}_2 \\ &= -\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 + \boldsymbol{z}_1^{\intercal}\boldsymbol{z}_2 + \boldsymbol{z}_2^{\mathrm{T}}\dot{\boldsymbol{z}}_2 \\ &= -\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - \alpha_2\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_2 + \alpha_2\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_2 + \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_2 + \boldsymbol{z}_2^{\mathrm{T}}\dot{\boldsymbol{z}}_2,\end{aligned} \tag{15}$$

where α_2 is another user-specified LBSC parameter.

Substituting (12) into (15), we can obtain

$$\dot{V}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x})) = -\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - \alpha_2\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_2 \tag{16}$$

under the following control law

$$\boldsymbol{h}(\boldsymbol{x}) = -\boldsymbol{F}^{-1}\boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}^{-1}\boldsymbol{R}^{\mathrm{T}}(\psi)\boldsymbol{\mu}, \tag{17}$$

where

$$\boldsymbol{\mu} = \ddot{\boldsymbol{\eta}}_r - \dot{\boldsymbol{R}}(\psi)\boldsymbol{v}_b + (\alpha_1 + \alpha_2)\boldsymbol{z}_2 + (1 - \alpha_1^2)\boldsymbol{z}_1. \tag{18}$$

Likewise, submitting (3), (11) and (12) into (15), we can get (19) under the NMPC control action $\boldsymbol{u}$:

$$\dot{V}(\boldsymbol{x}, \boldsymbol{u}) = -\alpha_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - \alpha_2\boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_2 + \boldsymbol{z}_2^{\mathrm{T}}[\boldsymbol{\mu} - \boldsymbol{R}(\psi)(\boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}\boldsymbol{u})]. \tag{19}$$

The contraction constraint (7e) is exploited for the detailed expression when $\dot{V}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x}))$ and $\dot{V}(\boldsymbol{x}, \boldsymbol{u})$ are substituted:

$$\widehat{\boldsymbol{z}}_2^{\mathrm{T}}(t_k; t_k)[\widehat{\boldsymbol{\mu}}(t_k; t_k) - \boldsymbol{R}(\widehat{\psi}(t_k; t_k))(\boldsymbol{S}\widehat{\boldsymbol{v}}_b(t_k; t_k) + \boldsymbol{F}\widehat{\boldsymbol{u}}(t_k; \widehat{\boldsymbol{x}}(t_k; t_k))] \leq 0.$$

Remark 4.1 The NMPC control action derived from $(\mathcal{P}_0(\boldsymbol{x}))$ is $\boldsymbol{u}(t_k) = \widehat{\boldsymbol{u}}^*(s; t_k), s \in [t_k, t_k + \delta]$, e.g., only the first element of the optimal solution sequence is sent into the autopilot. For the sake of improving computational efficiency, it is necessary to consider the contraction constraint for $s \in [t_k, t_k + \delta]$. Moreover, according to the LBSC law derivation process, we can conclude that this is the contraction constraint that allows one to prove that the NMPC-based on $(\mathcal{P}_0(\boldsymbol{x}))$ inherits the stability and robustness properties of LBSC.

4.2 Feasibility Analysis

Note that the auxiliary state feedback law must satisfy $||\boldsymbol{h}(\boldsymbol{x})||_\infty \leq u_{\max}$ so as to ensure feasibility of the NMPC optimization $(\mathcal{P}_0(\boldsymbol{x}))$. To this end, we obtain the following theoretical results for ensuring the feasibility:

Lemma 4.2 *Given that the quadrotor is controlled by the LBSC law* (17)*, the velocity* $\boldsymbol{v}_b$ *is bounded by*

$$||\boldsymbol{v}_b||_\infty \leq \sqrt{2}(\overline{\eta}_1 + ||\boldsymbol{z}_2(t_0)||_2 + \alpha_1||\boldsymbol{z}_1(t_0)||_2), \tag{20}$$

where $\boldsymbol{z}_1(t_0)$ *and* $\boldsymbol{z}_2(t_0)$ *are the position error and velocity error, respectively, at initial time* t_0 *(generally* $t_0 = 0$*).*

Proof Based on (1), we have

$$||\boldsymbol{v}_b||_\infty = ||\boldsymbol{R}^{\mathrm{T}}(\psi)\dot{\boldsymbol{\eta}}||_\infty \leq ||\boldsymbol{R}^{\mathrm{T}}(\psi)||_\infty ||\dot{\boldsymbol{\eta}}||_\infty \leq \sqrt{2}||\dot{\boldsymbol{\eta}}||_\infty. \tag{21}$$

From $\boldsymbol{z}_1 = \boldsymbol{\eta}_r - \boldsymbol{\eta}$ and $\dot{\boldsymbol{z}}_1 = \boldsymbol{z}_2 - \alpha_1 \boldsymbol{z}_1$, yield

$$||\dot{\boldsymbol{\eta}}||_\infty = ||\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{z}}_1|| \leq \overline{\eta}_1 + ||\dot{\boldsymbol{z}}_1||_\infty \leq \overline{\eta}_1 + ||\boldsymbol{z}_2||_\infty + \alpha_1 ||\boldsymbol{z}_1||_\infty. \tag{22}$$

Since $\dot{V} \leq 0$, there exist that $||\boldsymbol{z}_1(t_k)||_\infty \leq ||\boldsymbol{z}_1(t_k)||_2 \leq ||\boldsymbol{z}_1(t_0)||_2$ and $||\boldsymbol{z}_2(t_k)||_\infty \leq ||\boldsymbol{z}_2(t_k)||_2 \leq ||\boldsymbol{z}_2(t_0)||_2$. Therefore, (20) holds for $||\dot{\boldsymbol{\eta}}||_\infty \leq \overline{\eta}_1 + ||\boldsymbol{z}_2(t_0)||_2 + \alpha_1 ||\boldsymbol{z}_1(t_0)||_2$. This completes the proof. ∎

Theorem 4.3 *Given* $\overline{f} = ||F^{-1}||_\infty$, $\overline{s} = ||S||_\infty$, *e.g.,* $\overline{f} = \max\{a_f^{-1}, b_f^{-1}, c_f^{-1}, d_f^{-1}\}$, $\overline{s} = \max\{a_s, b_s, c_s, d_s\}$ *and the control parameters* α_1 *and* α_2 *are positive scalars. If the following condition can be satisfied:*

$$\sqrt{2} \cdot \overline{f}(\overline{s} \cdot l + \overline{\eta}_2 + 2\sqrt{2} \cdot l^2 + m) \leq u_{\max}, \tag{23}$$

where $l = \overline{\eta}_1 + ||\boldsymbol{z}_2(t_0)||_2 + \alpha_1 ||\boldsymbol{z}_1(t_0)||_2$, $m = (\alpha_1 + \alpha_2)||\boldsymbol{z}_2(t_0)||_2 + (1 - \alpha_1^2)||\boldsymbol{z}_1(t_0)||_2$, *then* $(\mathcal{P}_0(\boldsymbol{x}))$ *admits recursive feasibility.*

Proof Firstly, it is necessary to calculate $||h(\boldsymbol{x})||_\infty$ by taking infinity norm on the both sides of (17). Then, we obtain the following inequality:

$$||\boldsymbol{h}(\boldsymbol{x})||_\infty \leq ||\boldsymbol{F}^{-1}||_\infty ||\boldsymbol{S}||_\infty ||\boldsymbol{v}_b||_\infty + ||\boldsymbol{F}^{-1}||_\infty ||\boldsymbol{R}^{\mathrm{T}}(\psi)||_\infty ||\boldsymbol{\mu}||_\infty. \tag{24}$$

Next, we calculate $||\boldsymbol{v}_b||_\infty$, $||\boldsymbol{R}^{\mathrm{T}}(\psi)||_\infty$ and $||\boldsymbol{\mu}||_\infty$ respectively. According to the definition of $\boldsymbol{R}(\psi)$, we imply

$$||\boldsymbol{R}^{\mathrm{T}}(\psi)||_\infty = \max\{|\sin(\psi)| + |\cos(\psi)|, 1\} \leq \sqrt{2}. \tag{25}$$

Similarly, the following inequality holds for (18)

$$||\boldsymbol{\mu}||_\infty \leq \overline{\eta}_2 + ||\boldsymbol{\Omega}||_\infty ||\boldsymbol{v}_b||_\infty + (\alpha_1 + \alpha_2)||\boldsymbol{z}_2||_\infty + (1 - \alpha_1^2)||\boldsymbol{z}_1||_\infty, \tag{26}$$

where

$$\boldsymbol{\Omega} = -\dot{\boldsymbol{R}}(\psi) = \begin{bmatrix} \sin(\psi)r & \cos(\psi)r & 0 & 0 \\ -\cos(\psi)r & \sin(\psi)r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

with the fact that $||\boldsymbol{\Omega}||_\infty \leq \sqrt{2}||\boldsymbol{v}_b||_\infty$.

According to Lemma 4.2, there exists

$$||\boldsymbol{\mu}||_\infty \leq \overline{\eta}_2 + 2\sqrt{2} \cdot l^2 + m, \tag{27}$$

where $l = \overline{\eta}_1 + ||\boldsymbol{z}_2(t_0)||_2 + \alpha_1 ||\boldsymbol{z}_1(t_0)||_2$, $m = (\alpha_1 + \alpha_2)||\boldsymbol{z}_2(t_0)||_2 + (1 - \alpha_1^2)||\boldsymbol{z}_1(t_0)||_2$.

Substituting (20), (25) and (27) into (24), we have

$$||h(\boldsymbol{x})||_\infty \leq \sqrt{2} \cdot \overline{f}(\overline{s} \cdot l + \overline{\eta}_2 + 2\sqrt{2} \cdot l^2 + m). \tag{28}$$

Obviously, if (23) is satisfied, the sufficient condition of recursive feasibility, e.g., $||\boldsymbol{h}(\boldsymbol{x})||_\infty \leq u_{\max}$, is always satisfied at all times. This completes the proof. ∎

Remark 4.4 Theorem 4.3 restricts the selection range of the LBSC parameters by constructing (23) for recursive feasibility. Once the parameters are determined, a region of attract (ROA) $\mathcal{S}_\Omega = \{\boldsymbol{x} \in \mathbb{R}^8 | \sqrt{2} \cdot \overline{f}(\overline{s} \cdot l + \overline{\eta}_2 + 2\sqrt{2} \cdot l^2 + m) \leq u_{\max}\}$ for the solution is given by the set of all initial conditions from which it is possible to drive the state predictions to the equilibrium point. For the sake of facilitating the exploration of the possible optimal solution, it is desirable to make $\mathcal{S}_\Omega$ as large as possible in order to maximize the allowable operating region. Obviously, (23) allows for a larger ROA with smaller values of α_1 and α_2. Although the small α_1 and α_2 result in slow convergence under LBSC, the control performance is enhanced by the online optimization of NMPC.

4.3 Stability Analysis

Remember that the Lyapunov-based contraction constraint is constructed to ensure closed-loop stability under the recursive feasibility of $(\mathcal{P}_0(\boldsymbol{x}))$ guaranteed. Therefore, we have the following theoretical result:

Theorem 4.5 *Consider the auxiliary state feedback law $\boldsymbol{h}(\boldsymbol{x})$ in (17). Algorithm 1 enforces asymptotic convergence to the reference trajectory.*

Proof For the Lyapunov candidate $V(\boldsymbol{x})$ in (14), which is continuously differentiable and radically unbounded [30, Theorem 4.2], there exist $\mathcal{K}_\infty$ functions $\beta_i(\cdot), i = 1, 2, 3$ that satisfy the following inequalities by the converse Lyapunov theorem [30, Theorem 4.17]:

$$\beta_1(||\boldsymbol{x}||) \leq V(\boldsymbol{x}) \leq \beta_2(||\boldsymbol{x}||), \tag{29a}$$

$$\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x})) \leq -\beta_3(||\boldsymbol{x}||). \tag{29b}$$

Considering the stability constraint (7e) and that the control action $\boldsymbol{u}(t) = \boldsymbol{u}^*(s), s \in [t, t+\delta]$ in $(\mathcal{P}_0(\boldsymbol{x}))$ will be input for one sampling period each time, together with (29b) $V(\boldsymbol{x})$ in the closed-loop system under the NMPC control actions satisfies the following inequality:

$$\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \leq \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x})) \leq -\beta_3(||\boldsymbol{x}||).$$

Based on standard Lyapunov arguments (see [30, Theorem 4.9]), we claim that, $\widetilde{\boldsymbol{x}} = \boldsymbol{0}$ is asymptotically stable for the closed-loop system associated with the NMPC-based guidance law.

5 Simulation and Experimental Results

In this section, we design three simulation scenarios to validate the effectiveness of the proposed NMPC scheme. In the first scenario, we test the tracking performance for trajectory

tracking of a circle in three-dimensional space. In the second scenario, we exemplify the stability and robustness for trajectory tracking of a sinusoidal path in xoy plane by imposing the wind gust disturbances. In the third scenario, the obstacle avoidance function is tested for trajectory tracking of a safe straight line when going through a constrained environment. The simulation results illustrate the practical and theoretical advancement of the proposed NMPC method, including the excellent tracking performance, the robustness and the collision avoidance function. At last, we implement the gap traversal mission with the real flight platform by using the proposed NMPC guidance algorithm.

5.1 Parameters Selection

For the identified quadrotor UAV model parameters, its gains and time constants can be found in [23]. We choose the sampling period as $\delta = 0.1$ [s]. For the proposed NMPC controller, the weighting matrices $Q = \mathrm{diag}(10^3, 10^3, 10^3, 10^3, 100, 100, 100, 100)$ and $R = \mathrm{diag}(1, 1, 1, 1)$ are selected properly to achieve the trajectory convergence. The prediction horizon is $T = 7\delta$ [s]. The maximum allowed control signal $u_{\max}$ is 1 and the LBSC parameters are $\alpha_1 = 0.45, \alpha_2 = 1$.

5.2 Tracking Performance

The reference trajectory to be tested is the circle path defined in three-dimensional space: $x_r = \frac{1}{2}\cos\theta$, $y_r = \frac{1}{2}\sin\theta$, $z_r = 3 - 2\cos\theta$ with the timing law $\theta(t) = \frac{\pi}{20}t$. Since the model parameters are obtained by the system identification, there probably exist the identification errors. We assume a 30% model parameter error for the gains K_x, K_y, K_z, K_ψ. The trajectory tracking results are shown in Figures 2–4. The actual tracking trajectory steered by NMPC is drawn in the blue curve. The orange curve is the simulated trajectory using LBSC, while the green dashed circle curve is the reference trajectory. According to Figures 2 and 3, it can be clearly seen that both controllers could drive the quadrotor along the reference trajectory. Yet, the NMPC controller outperforms the LBSC controller. At the beginning of the tracking, it achieves more aggressive movements than LBSC to get the fastest possible convergence. As discussed in Remark 4.4, this is because NMPC searches for the best possible solution by leveraging online optimization, while the control parameters α_1 and α_2 with regard to LBSC are set as the small values for a large ROA. Figure 4 shows the computed control input signals for motion in x-axis, y-axis, z-axis and heading direction. All the control signals satisfy the control requirements as expected. However, the LBSC cannot fully take advantage of the allowed control input with the fixed control gain, which reveals the benefits of the proposed NMPC.

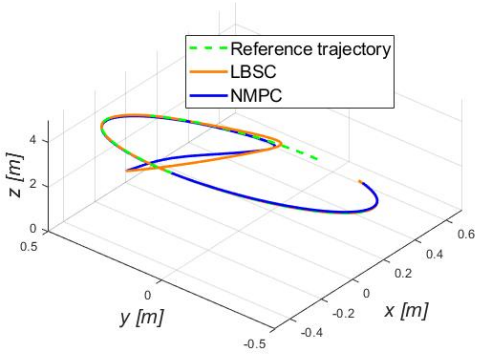

Figure 2 The quadrotor tracks the reference trajectory with the proposed NMPC guidance algorithm successfully

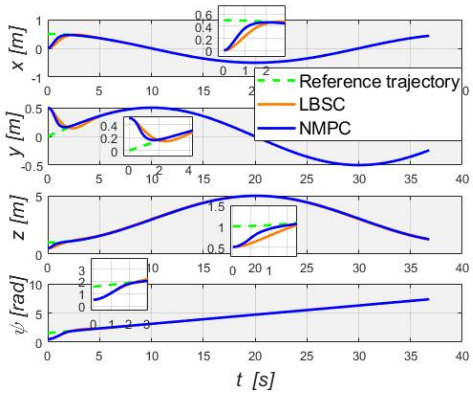

Figure 3 The state trajectories for the circle case

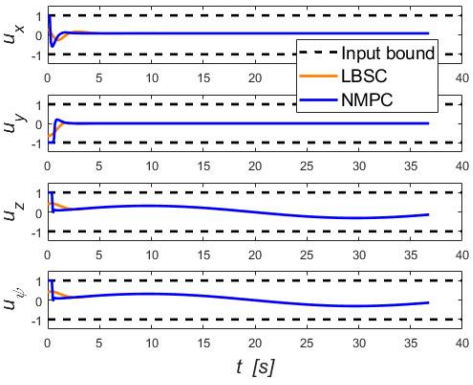

Figure 4 The control signals for the circle case

In order to better show the stability and robustness, a sinusoidal curve defined in xoy plane: $x_r = \theta,\ y_r = \sin\theta$ with $\theta = 0.2t$ is treated as the reference trajectory. In addition to the model uncertainties, the quadrotor is exposed to the wind gust disturbances in y-axis. The mathematical description of the wind gust is defined as:

$$d_{wind}(t) = \begin{cases} 0, & \text{others}, \\ \dfrac{V_{\max}}{2}\left(1 - \cos\left(2\pi\left(\dfrac{t - t_1}{t_p}\right)\right)\right), & t_1 < t < t_1 + t_p, \end{cases} \tag{30}$$

where $V_{\max}$ is the maximum wind speed, t_1 is the start time of the wind, t_p is the time duration. In this simulation, $V_{\max} = 0.2$ [m/s], $t_1 = 5$ [s], 15 [s] and 22 [s], $t_p = 3$ [s]. The simulation results are shown in Figures 5 and 6.

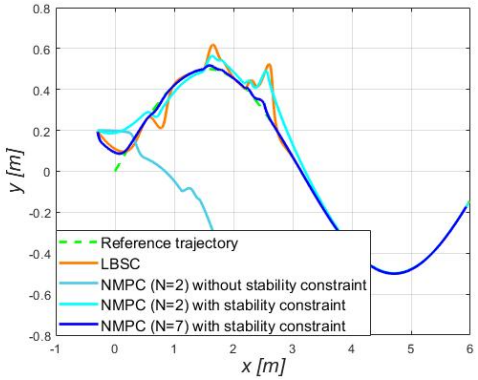

Figure 5 The tracking performance and robustness with the different NMPC parameters

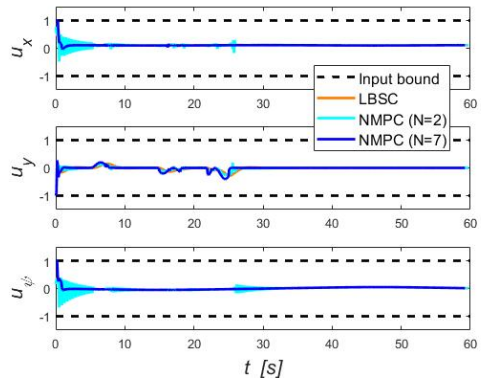

Figure 6 The control input for trajectory tracking with the wind gust disturbances

Figure 5 exemplifies that the closed-loop system controlled by the traditional NMPC without the stability constraint becomes unstable if the prediction horizon is set improperly, e.g., $T = 2\delta$ [s]. In contrast, the proposed NMPC demonstrates the better closed-loop performance with

the same prediction horizon, which highlights its advantage. For the pre-specified trajectory tracking, we can always design a set of proper parameters by trial and error to achieve the trajectory convergence. Nevertheless, for fully autonomous flight projects that involve arbitrary trajectory tracking, it is necessary to consider this stability constraint.

Figure 5 also demonstrates the robustness of NMPC with regard to the external disturbances. It can be seen that NMPC ($N = 7$) steers the quadrotor converging to the reference trajectory in the presence of model uncertainties and wind gust disturbances, while LBSC and NMPC ($N = 2$) exhibit visible tracking error. As discussed in Remark 3.3, on the one hand, this reveals that NMPC yields a better performance with a longer prediction horizon. However, extending the prediction horizon increases the computational burden. The prediction horizon may be restricted for real-time control. In practical applications, we should make a tradeoff between tracking performance and numerical efficiency. Nevertheless, with the stability constraint, we can easily make this tradeoff by specifying the maximum prediction horizon without destabilizing the tracking control. On the other hand, according to Figure 6, this also demonstrates that the prominent tracking performance of NMPC is comparably more acceptable than LBSC in terms of robustness. In essence, LBSC is a closed-loop state feedback control based on the tracking error while NMPC is an online open-loop optimal control. The proposed NMPC scheme can take advantage of online optimization to calculate appropriate control signals accounting for the compensation of the disturbances.

5.3 Tracking with Obstacle Avoidance

In order to testify the obstacle avoidance function, we set the initial position of the quadrotor at $x_0 = 0.3$ [m], $y_0 = 0$ [m] and the quadrotor flies through the gap along a safe straight line $x_r = \theta$, $y_r = -1$ with $\theta = 0.2t$. The parameters of the obstacle avoidance cost term $\lambda = 2000$, $k = 5$. The safety distance for the collision region is $r_s = 0.25$ [m]. The trajectories are shown in Figures 7 and 8. Obviously, when we choose the timing law parameter as $v_t = 0.3$, the quadrotor converges to the reference trajectory without any collision. But, lower v_t results in slower forward speed and spending more time for traversal as shown in Figure 8. When v_t is 0.6, without considering obstacle avoidance, the quadrotor enters the collision region (black dashed circle). In contrast, the proposed NMPC with the obstacle avoidance cost term steers the quadrotor avoiding the collision region.

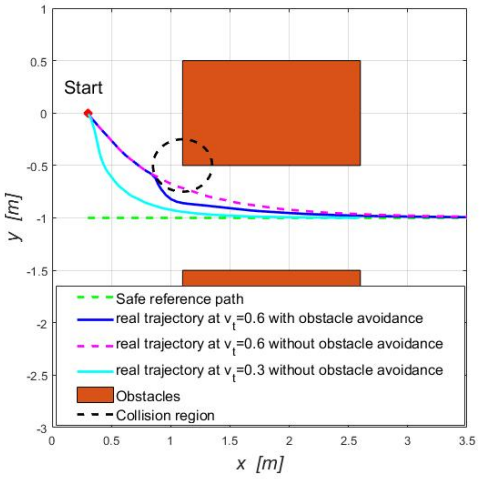

Figure 7 The trajectories comparison in obstacle avoidance with the difference timing laws

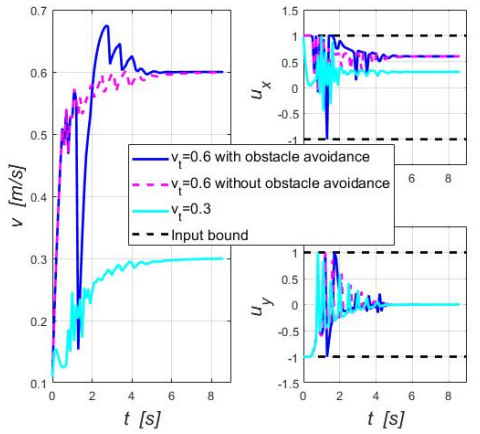

Figure 8 The evolution of the velocity v and control inputs u_x and u_y

5.4 Gap Traversal Experiment

To validate the trajectory tracking control performance of the proposed NMPC in practice, we do the gap traversal experiment by implementing this algorithm on the DJI M100 platform in comparison to LBSC with collision avoidance[31]. In addition to the onboard GPS/IMU and the processor Manifold, the platform is equipped with the onboard UTM-30XL 2D LIDAR for sensing the gap, UWB[32] for recording the position data in the local frame and DC-DC for power converting. The experimental scene and setup are shown in Figure 9. The blue square frame of 2×1.3 [m] is the gap. Obviously, the direct and convenient safe path is the straight line that is perpendicular to the gap plane through its center point. For simplification, we keep the quadrotor flying at the same altitude with the center of the frames by using 1D LIDAR for

altitude hold. The quadrotor is required to fly through the gap in forward direction without considering the collision in the vertical direction. When the gap is detected by the 2D LIDAR, its center's position is calculated by using the geometric method with the measured angles and distances and further the safe straight reference trajectory for traversal is generated with $v_t = 0.1$ [m/s]. The software module on the Manifold for detection and guidance is shown in Figure 10, which runs in the robot operating system (ROS) environment. According to the position reference trajectory, the guidance module generates the reference velocity commands for the autopilot. The control updating frequency in the guidance module is maintained at 10 [Hz]. The software development kit (SDK) of the DJI M100 is used to communicate with the autopilot via the UART serial port. The communication between the node of the guidance module and the SDK node is through the inter-process communication.

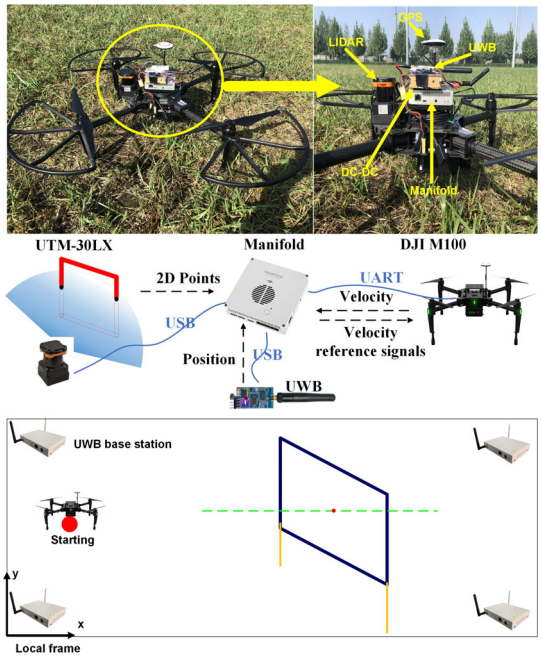

Figure 9 The experimental scene and hardware setup

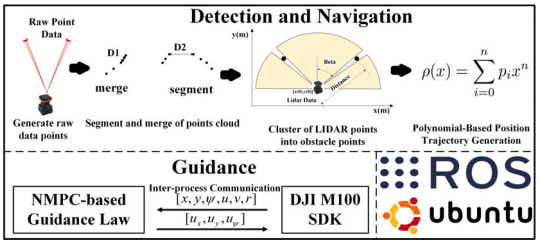

Figure 10 The software modules on the Manifold

The experimental results are shown in Figures 11–13. When the quadrotor approaches the gap, the gap is detected by UTM-30XL and the straight path is generated. Owing to the measurement errors for the angles and the distances to the gap sides, the calculated center point deviates from the real center point so that the straight path for traversal is close to one side of the gap. As the quadrotor traverses along the path, the distance to the collision region is reducing gradually. The obstacle avoidance function ensures that the quadrotor could avoid the collision and converge back to the straight line when it enters the collision region. Moreover, the guidance signals generated by the Manifold are always feasible for the real platform. In contrast, as we can see, the quadrotor has the slower trajectory convergence under LBSC than NMPC. The effect of the collision avoidance function is not significant. This is because that we have to reduce the controller parameters so that the control input signals stay within the bound. This results in the slow trajectory convergence and worse control performance.

Figure 11 The gap traversal of the quadrotor along a safe path

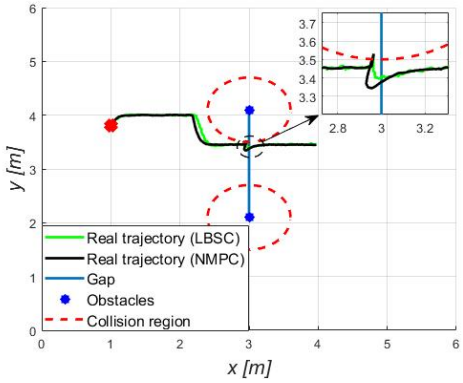

Figure 12 The real trajectory comparisons under the LBSC and NMPC in the experiment

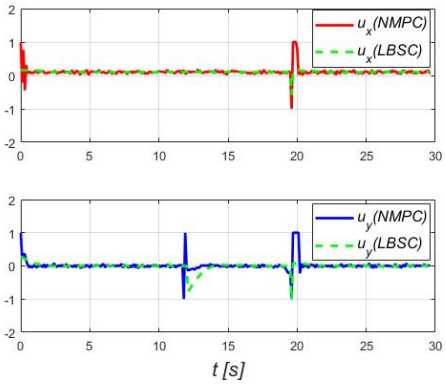

Figure 13 The control inputs u_x and u_y generated by LBSC and NMPC for trajectory tracking and obstacle avoidance

6 Conclusion

In this paper, we have proposed the NMPC guidance algorithm for the integrated trajectory tracking and obstacle avoidance control of a quadrotor UAV. We imposed the potential field function-based penalty term with well-tuned parameters on the tracking cost function for obstacle avoidance. Moreover, we incorporated a stability constraint into the online NMPC optimization problem to ensure the closed-loop stability. The simulations and experimental results demonstrated the effectiveness of the proposed NMPC.

As we can see in Figure 7, the parameter v_t of the timing law in the reference trajectory generation has a heavy influence on the tracking performance. As discussed in Remark 3.1, v_t should be carefully selected to fit the realistic constraint. Thus, how v_t is predetermined or whether v_t is capable of adaptive auto-adjustment according to tracking error is worth extensive researching in the near future. Moreover, when applying the proposed NMPC algorithm in safety critical applications, in addition to designing a collision-free trajectory, it is necessary to introduce the state constraint to limit the quadrotor position within the safe corridor for collision avoidance. Furthermore, NMPC could be combined with SMC or state observer to be more robust in the presence of external disturbances.

References

[1] Cai G, Dias J, and Seneviratne L, A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends, *Unmanned Systems*, 2014, **2**(2): 175–199.

[2] Sanket N J, Singh C D, and Ganguly K, GapFlyt: Active vision based minimalist structure-less gap detection for quadrotor flight, *IEEE Robotics and Automation Letters*, 2018, **3**(4): 2799–2806.

[3] Michel N, Bertrand S, Olaru S, et al., Design and flight experiments of a tube-based model

predictive controller for the AR.Drone 2.0 quadrotor, *Proceedings of the* 1*st IFAC Workshop on Robot Control*, Republic of Korea, 2019.

[4] Lotufo M A, Colangelo L, Perez-Montenegro C, et al., Embedded model control for UAV quadrotor via feedback linearization, *Proceedings of the* 20*th IFAC Symposium on Automatic Control in Aerospace*, Canada, 2016.

[5] Rigatos G, A nonlinear optimal control approach for tracked mobile robots, *Journal of Systems Science and Complexity*, 2021, DOI:10.1007/s11424-021-0036-1.

[6] Guo Y, Yu L, and Xu J, Robust finite-time trajectory tracking control of wheeled mobile robots with parametric uncertainties and disturbances, *Journal of Systems Science and Complexity*, 2019, **32**(5): 1358–1374.

[7] Kang B, Miao Y, Liu F, et al., A second-order sliding mode controller of quad-rotor UAV based on PID sliding mode surface with unbalanced load, *Journal of Systems Science and Complexity*, 2021, **34**(2): 520–536.

[8] Wang F, Chen W, Hand D, et al., Backstepping control of a quadrotor unmanned aerial vehicle based on multi-rate sampling, *Science China Information Sciences*, **62**(1): 19203.

[9] Jung S, Cho S, Lee D, et al., A direct visual servoing-based framework for the 2016 IROS autonomous drone racing challenge, *Journal of Field Robotics*, 2016, **35**(1): 146–166.

[10] Wang Y, Ma Y, Cai Z, et al., Quadrotor trajectory tracking and obstacle avoidance by chaotic grey wolf optimization-based backstepping control with sliding mode extended state observer, *Transactions of the Institute of Measurement and Control*, 2020, **42**(9): 1675–1689.

[11] AbdulSamed B N, Aldair A A, and Al-Mayyahi A, Robust trajectory tracking control and obstacles avoidance algorithm for quadrotor unmanned aerial vehicle, *Journal of Electrical Engineering & Technology*, 2020, **15**(2): 855–868.

[12] Jing Y, Liu Y, and Zhou S, Prescribed performance finite-time tracking control for uncertain nonlinear systems, *Journal of Systems Science and Complexity*, 2019, **32**(3): 803–817.

[13] Chen H and Song S, Robust chattering-free finite time attitude tracking control with input saturation, *Journal of Systems Science and Complexity*, 2019, **32**(6): 1597–1629.

[14] Shen C, Buckham B, and Shi Y, Modified C/GMRES algorithm for fast nonlinear model predictive tracking control of AUVs, *IEEE Transactions on Control Systems Technology*, 2017, **25**(5): 1896–1904.

[15] Ellis M, Liu J, and Christofides P D, *Economic Model Predictive Control: Theory, Formulations and Chemical Process Applications*, Springer, Switzerland, 2017,

[16] Faulwasser T, Weber T, Zometa P, et al., Implementation of nonlinear model predictive path-following control for an industrial robot, *IEEE Transactions on Control Systems Technology*, 2017, **25**(4): 1505–1511.

[17] Hu Q, Xie J, and Wang C, Dynamic path planning and trajectory tracking using MPC for satellite with collision avoidance, *ISA Transactions*, 2017, **84**: 128–141.

[18] Wang Y, Zheng H, Zong C, et al., Path-following control of autonomous ground vehicles using triple-step model predictive control, *Science China Information Sciences*, 2020, **63**(10): 209203.

[19] Greeff M and Schoellig A P, Flatness-based model predictive control for quadrotor trajectory tracking, *Proceedings of the* 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Spain, 2018.

[20] Liu X and Han C, Robust model predictive control of continuous uncertain systems, *Journal of Systems Science and Complexity*, 2008, **21**(2): 267–275.

[21] Chen H and Allgöwer F, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica*, 1998, **34**(10): 1205–1217.

[22] Worthmann K, Mehrez M W, Zanon M, et al., Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs, *IEEE Transactions on Control Systems Technology*, 2016, **24**(4): 1394–1406.

[23] Castillo-Lopez M, Sajadi-Alamdari S A, Sanchez-Lopez J L, et al., Model predictive control for aerial collision avoidance in dynamic environments, *Proceedings of the* 2018 *Mediterranean Conference on Control and Automation*, Croatia, 2018.

[24] Kamel M, Alonso-Mora J, Siegwart R, et al., Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control, *Proceedings of the* 2017 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Canada, 2017.

[25] Dentler J, Kannan S, Mendez M A O, et al., A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors, *Proceedings of the* 2016 *IEEE Conference on Control Applications*, Argentina, 2016.

[26] Blochliger F, Fehr M, Dymczyk M, et al., Topomap: Topological mapping and navigation based on visual SLAM maps, *Proceedings of the* 2018 *IEEE International Conference on Robotics and Automation*, Australia, 2018.

[27] Mayne D Q, Model predictive control: Recent developments and future promise, *Automatica*, 2014, **50**(12): 2967–2986.

[28] Munoz de la Pena D and Christofides P D, Lyapunov-based model predictive control of nonlinear systems subject to data losses, *IEEE Transactions on Automatic Control*, 2008, **53**(9): 2076–2089.

[29] Mayne D Q, Rawlings J B, Rao C V, et al., Constrained model predictive control: Stability and optimality, *Automatica*, 2000, **36**(6): 789–814.

[30] Khalil H K, *Nonlinear Systems*, Prentice, Upper Saddle River, NJ, 2002.

[31] Kowalczyk W, Rapid navigation function control for two-wheeled mobile robots, *Journal of Intelligent & Robotic Systems*, 2019, **93**(3): 687–697.

[32] Guo K, Qiu Z, Miao C, et al., Ultra-wideband-based localization for quadcopter navigation, *Unmanned Systems*, 2016, **4**(1): 23–34.