**SCIENCE CHINA**
Information Sciences

• **Supplementary File** •

# Nonlinear Model Predictive Control for Trajectory Tracking of Quadrotors Using Lyapunov Techniques

Dong Wang, Quan Pan, Jinwen Hu* & Chunhui Zhao

*School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi,* 710072, *P.R.China*

## Appendix A Reference Trajectory Augmentation Generation

The reference path $\mathcal{S}(\theta)$ describes the desired position in the inertial frame. In general, $\theta$ is time-dependent. The time-parameterized reference position trajectory $\boldsymbol{p}(t)$ is generated based on a predetermined timing law $\theta(t) = v_t t$, i.e., $x_r(t) = x_{rr}(v_t t), y_r(t) = y_{rr}(v_t t), z_r(t) = z_{rr}(v_t t)$, where $v_t > 0$ denotes the velocity of the path progress in forward direction. To avoid singularities in the reference trajectory, we generate the reference system as follows: Let $\boldsymbol{x}_r(t) = \mathrm{col}(\boldsymbol{\eta}_r(t), \boldsymbol{v}_{br}(t))$ with

$$\begin{aligned} \boldsymbol{\eta}_r(t) &= [x_r(t), y_r(t), z_r(t), \psi_r(t)]^\intercal, \\ \boldsymbol{v}_{br}(t) &= [u_r(t), v_r(t), w_r(t), r_r(t)]^\intercal, \end{aligned} \tag{A1}$$

where

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

atan2$\{\cdot\}$ is the four-quadrant inverse tangent operator.

**Remark 1.** The reference trajectory plays a momentous role in the trajectory tracking system. Notice that an appropriate timing law of $\theta(t)$ may be carefully designed such that the reference augmentation chosen in (A1) 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 in the presence of the initial position error, which may lead to an unsafe result. The influence of the parameter $v_t$ on trajectory tracking will be shown in Appendix F.3.

## Appendix B NMPC Algorithm

**Algorithm B1** NMPC Algorithm

1: Input the objective function $J$;
2: Receive the measured state $\boldsymbol{x}(t_k)$ from the onboard sensors;
3: Solve the optimization problem $\mathcal{P}(\boldsymbol{x})$ with $\hat{\boldsymbol{x}}(t_k; t_k) = \boldsymbol{x}(t_k)$ and generate the (sub-)optimal solution $\hat{\boldsymbol{u}}^*(s; t_k), s \in [t_k, t_k + T]$;
4: Implement the control input $\boldsymbol{u}(t) = \hat{\boldsymbol{u}}^*(s; t_k), s \in [t_k, t_k + \delta]$ for one sampling period;
5: $k = k + 1$, $t_{k+1} = t_k + \delta$, go to Step 2;

## Appendix C Stability constraint

Consider a Lyapunov candidate as follows:

$$V_1 = \frac{1}{2}\boldsymbol{z}_1^\intercal \boldsymbol{z}_1, \tag{C1}$$

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

$$\dot{V}_1 = \boldsymbol{z}_1^\intercal \dot{\boldsymbol{z}}_1 = \boldsymbol{z}_1^\intercal [\dot{\boldsymbol{\eta}}_r - R(\psi)\boldsymbol{v}_b] + \alpha_1 \boldsymbol{z}_1^\intercal \boldsymbol{z}_1 - \alpha_1 \boldsymbol{z}_1^\intercal \boldsymbol{z}_1, \tag{C2}$$

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

Further define the velocity error in the inertial frame

$$\boldsymbol{z}_2 \triangleq \dot{\boldsymbol{\eta}}_r - R(\psi)\boldsymbol{v}_b + \alpha_1 \boldsymbol{z}_1 = \dot{\boldsymbol{z}}_1 + \alpha_1 \boldsymbol{z}_1. \tag{C3}$$

* Corresponding author (email: hujinwen@nwpu.edu.cn)

Then,

$$\begin{aligned}\dot{\boldsymbol{z}}_2 &= \ddot{\boldsymbol{\eta}}_r - \dot{R}(\psi)\boldsymbol{v}_b - R(\psi)\dot{\boldsymbol{v}}_b + \alpha_1\dot{\boldsymbol{z}}_1 \\ &= \ddot{\boldsymbol{\eta}}_r - \dot{R}(\psi)\boldsymbol{v}_b - R(\psi)\dot{\boldsymbol{v}}_b + \alpha_1(\boldsymbol{z}_2 - \alpha_1\boldsymbol{z}_1).\end{aligned} \tag{C4}$$

Exploiting (C2) with (C3), we have

$$\dot{V}_1 = \boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_2 - \alpha_1\boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_1. \tag{C5}$$

The Lyapunov candidate is chosen as

$$V = V_1 + \frac{1}{2}\boldsymbol{z}_2^{\mathsf{T}}\boldsymbol{z}_2. \tag{C6}$$

Substituting (C5) into (C7) yields:

$$\begin{aligned}\dot{V} &= \dot{V}_1 + \boldsymbol{z}_2^{\mathsf{T}}\dot{\boldsymbol{z}}_2 \\ &= -\alpha_1\boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_1 + \boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_2 + \boldsymbol{z}_2^{\mathsf{T}}\dot{\boldsymbol{z}}_2 \\ &= -\alpha_1\boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_1 - \alpha_2\boldsymbol{z}_2^{\mathsf{T}}\boldsymbol{z}_2 + \alpha_2\boldsymbol{z}_2^{\mathsf{T}}\boldsymbol{z}_2 + \boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_2 + \boldsymbol{z}_2^{\mathsf{T}}\dot{\boldsymbol{z}}_2,\end{aligned} \tag{C7}$$

where $\alpha_2$ is another user-specified LBSC parameter.

Substituting (C4) into (C7), we can obtain

$$\dot{V}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x})) = -\alpha_1\boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_1 - \alpha_2\boldsymbol{z}_2^{\mathsf{T}}\boldsymbol{z}_2 \tag{C8}$$

under the following control law

$$\boldsymbol{h}(\boldsymbol{x}) = -\boldsymbol{F}^{-1}\boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}^{-1}\boldsymbol{R}^{\mathsf{T}}(\psi)\boldsymbol{\mu}, \tag{C9}$$

where

$$\boldsymbol{\mu} = \ddot{\boldsymbol{\eta}}_r - \dot{\boldsymbol{R}}(\psi)\boldsymbol{v}_b + (\alpha_1 + \alpha_2)\boldsymbol{z}_2 + (1 - \alpha_1^2)\boldsymbol{z}_1. \tag{C10}$$

Likewise, submitting $\dot{\boldsymbol{v}}_b = \boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}\boldsymbol{u}$, (C3) and (C4) into (C7), we can get (C11) under the NMPC control action $\boldsymbol{u}$:

$$\dot{V}(\boldsymbol{x}, \boldsymbol{u}) = -\alpha_1\boldsymbol{z}_1^{\mathsf{T}}\boldsymbol{z}_1 - \alpha_2\boldsymbol{z}_2^{\mathsf{T}}\boldsymbol{z}_2 + \boldsymbol{z}_2^{\mathsf{T}}[\boldsymbol{\mu} - \boldsymbol{R}(\psi)(\boldsymbol{S}\boldsymbol{v}_b + \boldsymbol{F}\boldsymbol{u})]. \tag{C11}$$

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

$$\hat{\boldsymbol{z}}_2^{\mathsf{T}}(t_k; t_k)[\hat{\boldsymbol{\mu}}(t_k; t_k) - \boldsymbol{R}(\hat{\psi}(t_k; t_k))(\boldsymbol{S}\hat{\boldsymbol{v}}_b(t_k; t_k) + \boldsymbol{F}\hat{\boldsymbol{u}}(t_k; t_k)] \leqslant 0.$$

## Appendix D Proof of Theorem 1

*Proof.* Firstly, it is necessary to calculate $||h(\boldsymbol{x})||_\infty$ by taking infinity norm on the both sides of (C9). Then, we obtain the following inequality:

$$||\boldsymbol{h}(\boldsymbol{x})||_\infty \leqslant ||\boldsymbol{F}^{-1}||_\infty||\boldsymbol{S}||_\infty||\boldsymbol{v}_b||_\infty + ||\boldsymbol{F}^{-1}||_\infty||\boldsymbol{R}^{\mathsf{T}}(\psi)||_\infty||\boldsymbol{\mu}||_\infty. \tag{D1}$$

Next, we calculate $||\boldsymbol{v}_b||_\infty$, $||\boldsymbol{R}^{\mathsf{T}}(\psi)||_\infty$ and $||\boldsymbol{\mu}||_\infty$ respectively.

Given

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

we imply

$$||\boldsymbol{R}^{\mathsf{T}}(\psi)||_\infty = \max\{|\sin(\psi)| + |\cos(\psi)|, 1\} \leqslant \sqrt{2}. \tag{D2}$$

Similarly, the following inequality holds for (C10)

$$||\boldsymbol{\mu}||_\infty \leqslant \bar{\eta}_2 + ||\boldsymbol{\Omega}||_\infty||\boldsymbol{v}_b||_\infty + (\alpha_1 + \alpha_2)||\boldsymbol{z}_2||_\infty + (1 - \alpha_1^2)||\boldsymbol{z}_1||_\infty, \tag{D3}$$

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 $||\boldsymbol{\Omega}||_\infty \leqslant \sqrt{2}||\boldsymbol{v}_b||_\infty$.

Based on the kinematic equations, we have

$$||\boldsymbol{v}_b||_\infty = ||\boldsymbol{R}^{\mathsf{T}}(\psi)\dot{\boldsymbol{\eta}}||_\infty \leqslant ||\boldsymbol{R}^{\mathsf{T}}(\psi)||_\infty||\dot{\boldsymbol{\eta}}||_\infty \leqslant \sqrt{2}||\dot{\boldsymbol{\eta}}||_\infty. \tag{D4}$$

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

$$||\dot{\boldsymbol{\eta}}||_\infty = ||\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{z}}_1|| \leqslant \bar{\eta}_1 + ||\dot{\boldsymbol{z}}_1||_\infty \leqslant \bar{\eta}_1 + ||\boldsymbol{z}_2||_\infty + \alpha_1||\boldsymbol{z}_1||_\infty. \tag{D5}$$

Since $\dot{V} \leqslant 0$, there exist that $||\boldsymbol{z}_1(t_k)||_\infty \leqslant ||\boldsymbol{z}_1(t_k)||_2 \leqslant ||\boldsymbol{z}_1(t_0)||_2$ and $||\boldsymbol{z}_2(t_k)||_\infty \leqslant ||\boldsymbol{z}_2(t_k)||_2 \leqslant ||\boldsymbol{z}_2(t_0)||_2$. Therefore,

$$||\boldsymbol{v}_b||_\infty \leqslant \sqrt{2}(\bar{\eta}_1 + ||\boldsymbol{z}_2(t_0)||_2 + \alpha_1||\boldsymbol{z}_1(t_0)||_2) \tag{D6}$$

holds for $||\dot{\boldsymbol{\eta}}||_\infty \leqslant \bar{\eta}_1 + ||\boldsymbol{z}_2(t_0)||_2 + \alpha_1||\boldsymbol{z}_1(t_0)||_2$. Then, we have

$$||\boldsymbol{\mu}||_\infty \leqslant \bar{\eta}_2 + 2\sqrt{2}\cdot l^2 + m, \tag{D7}$$

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

Substituting (D2), (D6) and (D7) into (D1), we have

$$||h(\boldsymbol{x})||_\infty \leqslant \sqrt{2}\cdot \bar{f}(\bar{s}\cdot l + \bar{\eta}_2 + 2\sqrt{2}\cdot l^2 + m). \tag{D8}$$

Obviously, if (3) is satisfied, the sufficient condition of recursive feasibility, e.g., $||\boldsymbol{h}(\boldsymbol{x})||_\infty \leqslant u_{\max}$, is always satisfied at all times. This completes the proof.

## Appendix E Proof of Theorem 2

*Proof.* The closed-loop system has an asymptotically stable equilibrium $\tilde{\boldsymbol{x}} = \boldsymbol{0}$ for the feedback control law $\boldsymbol{h}(\boldsymbol{x})$. $V(\boldsymbol{x})$ is the corresponding Lyapunov function. By the converse Lyapunov theorem [1], there exist functions $\beta_i(\cdot), i = 1, 2, 3$ of class $\mathcal{K}_\infty$ that satisfy the following inequalities:

$$\beta_1(||\boldsymbol{x}||) \leqslant V(\boldsymbol{x}) \leqslant \beta_2(||\boldsymbol{x}||), \tag{E1a}$$

$$\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\mathbf{f}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x})) \leqslant -\beta_3(||\boldsymbol{x}||). \tag{E1b}$$

Owing to the stability constraint (1d), together with (E1b) the $V(\boldsymbol{x})$ in the closed-loop system under the NMPC control action $\boldsymbol{u}(t)$ satisfies the following inequality:

$$\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\mathbf{f}(\boldsymbol{x}, \boldsymbol{u}) \leqslant \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\mathbf{f}(\boldsymbol{x}, \boldsymbol{h}(\boldsymbol{x})) \leqslant -\beta_3(||\boldsymbol{x}||). \tag{E2}$$

Based on standard Lyapunov arguments [1, Theorem 4.9], we claim that $\tilde{\boldsymbol{x}} = \boldsymbol{0}$ is asymptotically stable for the closed-loop system under Algorithm B1, e.g., the quadrotor is guaranteed to converge to the reference trajectory. This completes the proof.

## Appendix F Simulation and Experimental Results

We set up three simulation scenarios to validate the effectiveness of the NMPC algorithm for the commercial quadrotor system. In the first scenario, we test the tracking performance for trajectory tracking of the circle in three-dimensional space. In the second scenario, we exemplify the stability and robustness for trajectory tracking of the sinusoidal path in $xoy$ plane by introducing considerable wind gust disturbances. In the third scenario, the obstacle avoidance is tested for trajectory tracking of the safe straight line when going through the 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 on the real DJI M100 flight platform by using the proposed NMPC algorithm.

## Appendix F.1 Parameters Selection

For the identified DJI M100 model parameters, its gains and time constants can be found in [2]. For the NMPC controller, the following parameters are used: sampling period is $\delta = 0.1[s]$, prediction horizon is $T = 7\delta[s]$, the weighting matrices of cost function are chosen as $Q = \mathrm{diag}(10^3, 10^3, 10^3, 10^3, 10, 10, 10, 10)$, $R = diag(1, 1, 1, 1)$, the maximum allowed control signal $u_{max}$ is 1, LBSC parameters $\alpha_1 = 0.45, \alpha_2 = 1$. diag() is the diagonal operation.

## Appendix F.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 $K_x, K_y, K_z, K_\psi$. The trajectory tracking results are shown in Figures F1-F3. 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 F1 and F2, it can be clearly seen that both controllers could drive the quadrotor along the reference trajectory. Further comparing the tracking performance, we find that with the NMPC controller the quadrotor converges faster than using the LBSC controller. As discussed in Remark 2, this is because the NMPC searches for the best possible solution by leveraging online optimization, while the control parameters $\alpha_1$ and $\alpha_2$ are selected to be small for a large ROA. Figure F3 shows the computed control input signals for motion in x-axis, y-axis, z-axis and heading direction. We observe that at the beginning of the tracking, the NMPC controller makes more aggressive movements than LBSC to get the fastest possible convergence. The LBSC with the fixed control gain cannot make fully use of the allowed control input, which reveals the advantage of NMPC. As expected, all the control commands stay within the permitted range of magnitude.

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:

$$d_{wind}(t) = \begin{cases} 0,\ \text{others} \\ \frac{V_{\max}}{2}(1-\cos(2\pi(\frac{t-t_1}{t_p}))),\ t_1 < t < t_1 + t_p \end{cases}, \tag{F1}$$

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 F4 and F5.

Figure F4 exemplifies the lost stability. If the parameters are badly selected, the quadrotor controlled by NMPC may not converge to the reference trajectory. As can be seen from this figure, when we choose an unsuitable prediction horizon $N = 2$,

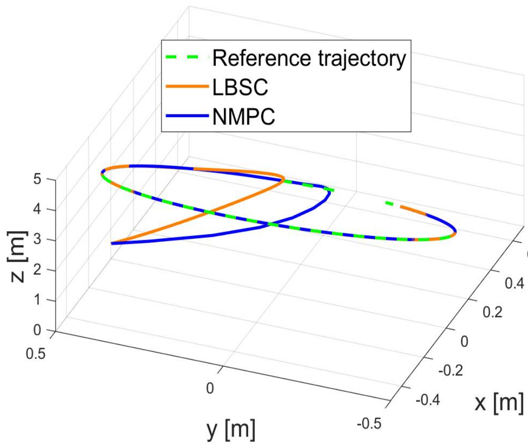


**Figure F1** The quadrotor successfully tracks the reference trajectory with the NMPC control algorithm.

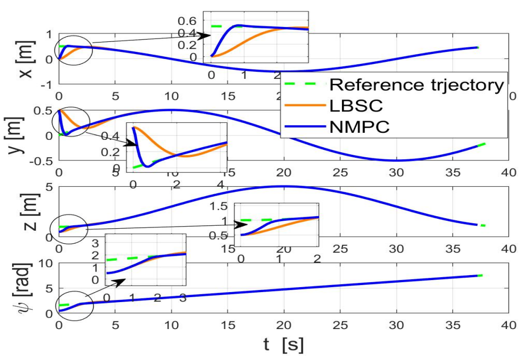


**Figure F2** State trajectories for the circle case

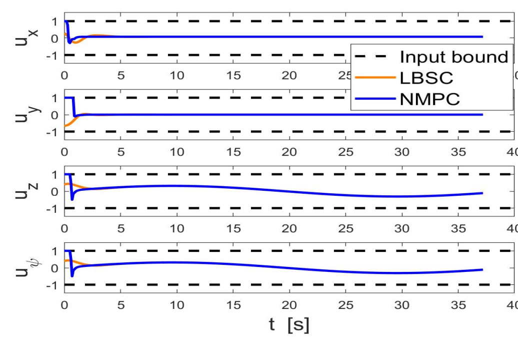


**Figure F3** The control command for the circle case

the quadrotor trajectory controlled by NMPC without the stability constraint will diverge. Similar instability can occur if the weighting matrices are badly chosen. In contrast, the NMPC with the stability constraint ($N = 2$) demonstrates the guaranteed closed-loop stability, which highlights its main advantages. Nevertheless, for tracking fixed types of desired trajectories, we are usually able to find a set of well-tuned parameters so that the NMPC without the stability constraint can still steer the quadrotor along the reference trajectory. In practical applications that involve tracking of arbitrary trajectories (such as fully autonomous flight in unknown environments), it is better to have this theoretical stability guarantee.

Figure F4 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. 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 burdens. 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 F5, this also demonstrates that the prominent tracking performance of NMPC is comparably more acceptable than LBSC in terms of control robustness. This is because LBSC is a closed-loop feedback control based on the tracking error while NMPC is an online open-loop optimal control. NMPC leverages online optimization to adjust an appropriate control input to well compensate the disturbances while LBSC is a fixed gain controller.

## Appendix F.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 results are shown in Figures F6 and F7. Obviously, when we set the timing law parameter $v_t = 0.3$, 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 F7. 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.

## Appendix F.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 [3]. In addition to the

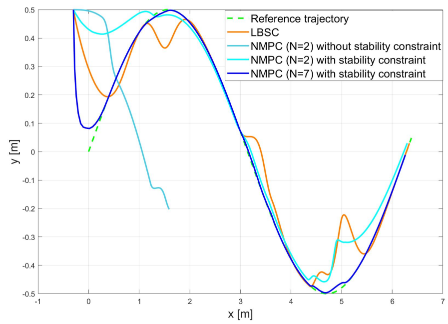


**Figure F4** The tracking performance and robustness with different NMPC parameters

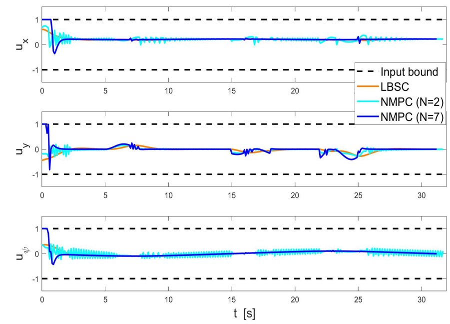


**Figure F5** The control input for trajectory tracking with wind gust disturbances

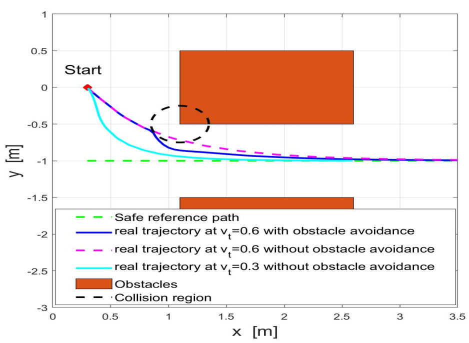


**Figure F6** Trajectories comparison

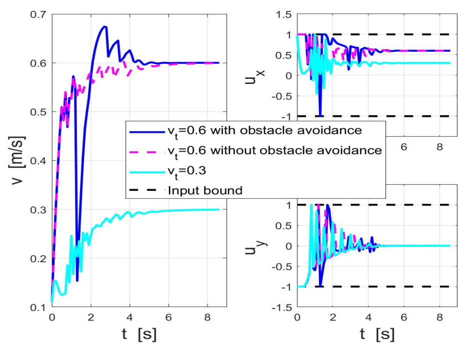


**Figure F7** The velocity $v$ and control inputs $u_x$ and $u_y$ evolution

onboard GPS/IMU and the processor Manifold, the platform is equipped with the onboard UTM-30XL 2D LIDAR for sensing the gap, UWB for recording the position data in the local frame and DC-DC for power converting. The experimental scene and setup are shown in Figure F8. The blue square frame of $2 \times 1.3[m]$ 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 experimental results are shown in Figures F9-F11. When the quadrotor approaches the gap, the gap is detected by UTM-30XL and the straight path is calculated. Owing to the measurement errors for the angles and distance 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 control 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.

## References

1 H. K. Khalil, Nonlinear Systems. Prentice Hall, 2002.
2 M. Castillo-Lopez, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, et al. Model predictive control for aerial collision avoidance in dynamic environments. In: Proceedings of the 2018 Mediterranean Conference on Control and Automation, Croatia, 2018.
3 W. Kowalczyk. Rapid navigation function control for two-wheeled mobile robots. Journal of Intelligent Robotic Systems, 2019, 93: 687-697.

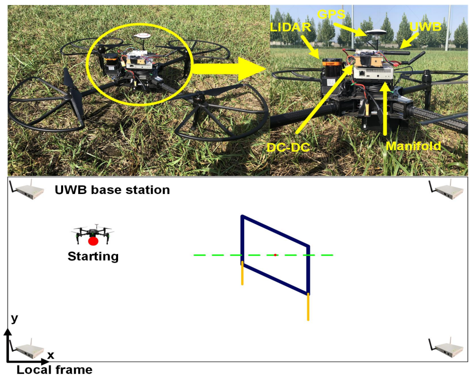


**Figure F8** The experimental setup

**Figure F9** The gap traversal of the quadrotor

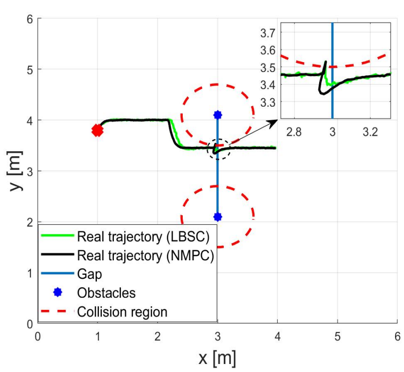


**Figure F10** The real trajectory comparisons under the LBSC and NMPC in the experiment.

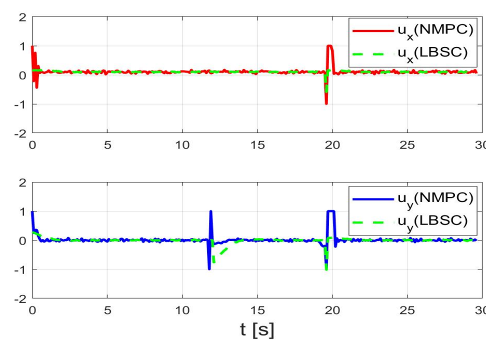


**Figure F11** The control inputs $u_x$ and $u_y$ generated by LBSC and NMPC for trajectory tracking and obstacle avoidance.