iLQR-Based Model Predictive Control for Trajectory Tracking of Quadrotor UAVs

Jun Ren, Mingyuan Zhai, Yanhui Tong, Lidong He, and Yudong Wang

***Abstract*—The problem of trajectory tracking for quadrotor unmanned aerial vehicles (UAVs) is investigated in this paper. An iterative linear quadratic regulator (iLQR) based model predictive control (MPC) strategy is proposed. The proposed iLQR-MPC strategy solves the nonlinear optimal control problem using the iLQR algorithm and implements the control in a receding horizon manner. A constrained iLQR algorithm is designed in the framework of the augmented Lagrangian method to solve the optimization problem induced by the trajectory tracking. Finally, a comparative simulation experiment is conducted to demonstrate the effectiveness and advantage of the proposed control strategy.**

***Index Terms*—Quadrotor unmanned aerial vehicles (UAVs), iterative linear quadratic regulator (iLQR), trajectory tracking, control algorithm**

I. Introduction

A. Motivation

In recent years, propelled by the swift progress in the aerospace technology, quadrotor unmanned aerial vehicles (UAVs) have garnered extensive applications across a multitude of sectors, owing to their unique performance advantages [1, 2]. These advantages are compact size, operational flexibility, and relatively low cost, making them ideal for the broad spectrum of applications. Within military contexts, these UAVs are capable of executing a variety of pivotal tasks including precision strikes, border surveillance, and military reconnaissance [3–5]. By virtue of their exceptional stealth and flexibility, quadcopter UAVs can undertake missions within hazardous environments, thereby mitigating risks to personnel and augmenting operational efficacy. In civilian domains, they are widely used in aerial transportation, power grid inspection, logistics distribution, and geological exploration. Amongst these applications, trajectory tracking capability emerges as a critical parameter for evaluating drone performance [6]. In these applications, the core challenges of trajectory tracking primarily include response speed and stability, precise control, and obstacle avoidance. During mission execution, the system must respond quickly to commands while maintaining stability and ensuring high tracking accuracy. Moreover, because of the nonlinear dynamic characteristics of quadcopters, the design of the controller poses a significant challenge.

In the areas of UAVs trajectory tracking problems, researchers have domestically and internationally conducted extensive investigations. These include methodologies such as PID control [7, 8], linear quadratic regulator (LQR) control [9–11], model reference adaptive control (MRAC) [12, 13], and model predictive control (MPC) [14, 15], among others. Because of the nonlinear, strongly coupled and underactuated characteristics of UAV [16], linear controllers often fail to effectively compensate for nonlinearities and external disturbances [17, 18]. Therefore, an increasing number of scholars are turning to nonlinear control methods. These approaches have not only further improved control accuracy and stability, but also enhanced the system's ability to resist external disturbances and internal uncertainties [19–21]. For example, an iterative linear quadratic regulator (iLQR) optimal control method is applied to the quadrotor UAV system with a cable-suspended payload [22], which exhibits high nonlinearity. This method not only achieves accurate tracking of the desired trajectory, but also effectively ensures the system's stability. A nonlinear model predictive control (NMPC) is used in Ref. [23] to extend the tracking of a desired trajectory to a reference dynamic system. To mitigate the computational burden associated with solving the corresponding NMPC optimization problem online, an improved continuous/generalized minimal residual algorithm is introduced. In Ref. [24], a data-driven model-free adaptive control method is employed to address the challenges of dynamic modeling and parameter identification in UAV trajectory tracking control, aiming to reduce the computational load of NMPC. In Ref. [25], a hierarchical controller for trajectory planning of tilt-propulsion UAVs is proposed. The controller features an inner loop based on the iLQR optimization and an outer loop employing feedback correction and forward rolling within an MPC framework. Experimental results show that the proposed controller achieves significantly better performance than the pure iLQR

Manuscript received: 27 August 2025; revised: 24 September 2025; accepted: 15 October 2025. *(Corresponding author: Yanhui Tong.)*

Citation: J. Ren, M. Zhai, Y. Tong, L. He, and Y. Wang, iLQR-based model predictive control for trajectory tracking of quadrotor UAVs, *Int. J. Intell. Control Syst.*, 2025, 30(4), 287–294.

Jun Ren and Yanhui Tong are with School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China (e-mail: M325123401@sues.edu.cn; nmgtong@sues.edu.cn).

Mingyuan Zhai is with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: zhaimy402@163.com).

Lidong He is with School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: lidonghe@njust.edu.cn).

Yudong Wang is with National Key Laboratory for Multi-modal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangyudong2018@ia.ac.cn).

Digital Object Identifier 10.62678/IJICS202512.10228

controller. However, constraint handling is not addressed in this design. A constrained iLQR algorithm is employed in Ref. [26] to handle general constraints. The constraints are first linearized, then shaped via a barrier function, and finally quadratized before being incorporated into the iLQR cost function.

B. Contribution

The main contributions of this paper are as follows:

(1) We propose an iLQR-based model predictive control framework for the trajectory tracking of quadrotor UAVs.

(2) We design a corresponding iLQR-MPC algarithm to solve the optimal trajectory tracking control via the augmented Lagrangian method.

(3) We conduct a comparative study beween the conventional MPC, iLQR, and iLQR-MPC in trajectory tracking of UAVs via simulation experiments.

C. Organization

The construction of the quadrotor UAV model is introduced in Section II. Subsequently, we propose the iLQR-based MPC control framework in Section III, comparative experiments are carried out in Section IV, and conclusions are summarized in Section V.

II. Problem Modeling and Formulation

According to Ref. [27], the final equations of the non-linear dynamic model of the quadrotor UAV, which derived via the Euler-Lagrange equations, are

$$\begin{cases} \ddot{x} = (\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi)\dfrac{F_z}{m}, \\ \ddot{y} = (\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi)\dfrac{F_z}{m}, \\ \ddot{z} = (\cos\varphi\cos\theta)\dfrac{F_z}{m} - g, \\ \ddot{\varphi} = \dfrac{(I_y - I_z)\dot{\theta}\dot{\psi}}{I_x} + \dfrac{M_x}{I_x}, \\ \ddot{\theta} = \dfrac{(I_z - I_x)\dot{\varphi}\dot{\psi}}{I_y} + \dfrac{M_y}{I_y}, \\ \ddot{\psi} = \dfrac{(I_x - I_y)\dot{\varphi}\dot{\theta}}{I_z} + \dfrac{M_z}{I_z} \end{cases} \tag{1}$$

where m is the mass of the quadrotor UAV, g is the gravitational acceleration, $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ represent its translational accelerations, respectively, ϕ, θ, and ψ denote the roll angle, pitch angle, and yaw angle, respectively, and I_x, I_y, and I_z represent the three axial moments of inertia of the quadrotor UAVs. $[M_x, M_y, M_z]^{\mathrm{T}}$ represents the moments about the roll, pitch, and yaw axes of the UAV, respectively. $L_a = L_b = L$ denote the lengths of the arms, defined as the distance from the propeller center to the UAV's center of gravity. α and β are the rotor's thrust coefficient and drag torque coefficient, respectively.

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} \alpha L_a(-w_2^2 + w_4^2) \\ \alpha L_b(-w_1^2 + w_3^2) \\ \beta(-w_1^2 + w_2^2 - w_3^2 + w_4^2) \end{bmatrix} \tag{2}$$

F_z is the total vertical thrust of the quadrotor UAV in flight, defined as

$$F_z = \alpha(w_1^2 + w_2^2 + w_3^2 + w_4^2) \tag{3}$$

The rotational speeds of the quadrotor UAV's four rotors are denoted as $[w_1, w_2, w_3, w_4]$. Define the system state vector as $x = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi}]^{\mathrm{T}}$ and the control input vector as $u = [w_1^2, w_2^2, w_3^2, w_4^2]^{\mathrm{T}}$. The paper focuses on constrained six-degree-of-freedom (6-DoF) trajectory tracking control and attitude stabilization for quadrotor UAVs. The control inputs are bounded by $u_{\min} \le u \le u_{\max}$, where $u_{\min}$ denotes the minimum thrust vector and $u_{\max}$ is the maximum thrust vector. Let the desired state trajectory be $x_d = [x_d, y_d, z_d, \dot{x}_d, \dot{y}_d, \dot{z}_d, \varphi_d, \theta_d, \psi_d, \dot{\varphi}_d, \dot{\theta}_d, \dot{\psi}_d]^{\mathrm{T}}$.

For the aforementioned nonlinear dynamic model, it can be expressed in the general discrete-time form

$$x(t+1) = f(x(t), u(t)) \tag{4}$$

This dynamic model governs the state transition from the current state $x(t)$ to the subsequent state $x(t+1)$ through the control input $u(t)$, where $x(t) \in \mathbb{R}^n$ denotes the state vector and $u(t) \in \mathbb{R}^m$ denotes the control input vector.

To guarantee the system to operate within prescribed bounds, both the state and control inputs are subjected to the constraint sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$, respectively. Specifically, the state vector is required to satisfy $x(t) \in \mathcal{X}$ at all times, while the control input must adhere to $u(t) \in \mathcal{U}$ during operation.

Based on Eq. (4), the resulting discrete optimization problem can be written as

$$\begin{aligned} \min_{x_{0:N}, u_{0:N-1}} \quad & \ell_N(x_N) + \sum_{i=0}^{N-1} \tau_i \ell_i(x_i, u_i), \\ \text{s.t.,} \quad & x_0 = f_0, \\ & x_{i+1} = f(x_i, u_i),\ i = 1, 2, ..., N-1; \\ & c_i(x_i, u_i) \leqslant 0,\ \forall i \end{aligned} \tag{5}$$

where N represents the number of prediction horizons, i represents the discrete time, x_i denotes the system state at time i, u_i represents the control input sequence, and f_0 represents the initial state. ℓ_i represents the operating cost, τ_i represents the attenuation coefficient, and ℓ_N represents the sum of the terminal costs. $c_i(x_i, u_i)$ denotes the inequality constraints.

III. Trajectory Tracking Controller Design

A. Augmented Lagrangian iLQR Method

The iLQR controller relies on the linearized system dynamics and a quadratic approximation of the cost function. This approach is particularly effective for nonlinear problems. However, when dealing with the complex nonlinearity and strict constraints, these approximations may lead to large linearization errors or inaccurate quadratic approximations, potentially causing the optimization problem to fail to converge [28, 29]. In the problem studied in this paper, there is a physical constraint issue regarding the control input. To address this, the following constraints are defined

$$c_i = [\max(0, u_i - u_{\max}, u_{\min} - u_i)] \tag{6}$$

thus, the cost function with constraints is

$$J_i(x_i,u_i)=\sum_{i=0}^{N-1}(\tau_i\ell_i(x_i,u_i))+\ell_N(x_N)+C(u_i),\ x_N\in\mathcal{X}_f \tag{7}$$

note that Q, R, and Q_f denote the state weighting matrix, the input weighting matrix, and the terminal state weighting matrix, respectively. Moreover, $Q\geq 0$, $R\geq 0$, and $Q_f\geq 0$ are expressed as

$$\begin{aligned}&\ell_i(x_i,u_i)=\parallel x(i)-x_d(i)\parallel_Q^2+\parallel u(i)-u_d(i)\parallel_R^2,\\&\ell_N(x_N)=\parallel x(N)-x_d(N)\parallel_{Q_f}^2\end{aligned} \tag{8}$$

$C(u_i)$ represents the penalty term. One of the simplest methods for solving constrained optimization problems is to move the constraints into the cost function and iteratively increase the penalty for approaching or violating the constraints. This paper adopts the augmented Lagrangian method to improve the penalty method by maintaining the estimation of the Lagrange multipliers related to the constraints [30].

$$C(u_i)=\frac{1}{2}c_i^{\mathrm{T}}I_\mu c_i+\lambda_k^{\mathrm{T}}c_i \tag{9}$$

where I_μ is the matrix that activates the constraints that are not satisfied, λ is the Lagrange multiplier, and

$$I_\mu=\begin{cases}0, \text{ otherwise};\\ \mu_i, \text{ if } c_i>0\end{cases} \tag{10}$$

The Lagrange multipliers update the rule as follows (typically $1<\kappa\leqslant 10$)

$$\lambda_i^+=\lambda_i+\mu_i c_i,\ \mu^+=\kappa\mu_i \tag{11}$$

When all four quantities in the control sequence u_k satisfy the constraint conditions, the matrix I_μ becomes a zero matrix, indicating that no constraint is violated. However, if the j-th quantity in u_k does not satisfy the constraint condition, the j-th row element of vector $C(u_k)$ will be non-zero, while the other elements remain zero. Correspondingly, the j-th diagonal element of matrix I_μ will also be non-zero, while the other diagonal elements will be zero. This results in an increase in the cost function value compared with the previous iteration.

In subsequent iterations, if this control quantity continues to violate the constraint conditions, its corresponding cost will continue to accumulate and increase. This mechanism ensures that the optimization process gradually tends toward a state where all constraint conditions are satisfied.

Since the iLQR algorithm is mainly used for trajectory planning, it pays more attention to the terminal position state. Given that this paper focuses on the trajectory tracking, it places more emphasis on the current position state. Therefore, a τ_i coefficient is added, and the value of τ_i decreases as i increases. α is typically $1<\alpha\leqslant 50$, $n\in N$.

$$\tau_i=\begin{cases}\dfrac{\alpha N}{i},\ i>n;\\ 1,\ i\leqslant n\end{cases} \tag{12}$$

B. iLQR-MPC Method

This paper will solve the iLQR controller within the control logic framework of MPC. The main steps of the MPC algorithm are as follows (at time k):

Step 1: Read the current state of the system at the current moment.

Step 2: Solve for the optimal control sequence.

Step 3: Only apply the first control quantity.

The core step of the MPC algorithm is step 2, of which the cost function J is set up and solved. The design of the iLQR-based MPC controller replaces step 2 with the iLQR algorithm to iteratively optimize and solve the optimal control sequence. The execution logic is shown in Fig. 1.

In Fig. 1, X_d represents a series of reference trajectory matrices, U_0 represents the initially given control sequence, x_0 represents the initial state, and the warm-starting step calculates the state sequence x and the initial cost function J_0 based on x_0 and U_0, providing a good initialization for the controller initially. Then, the state sequence x is used to perform the trajectory optimization algorithm. First, the iLQR algorithm is executed. Second, it checks whether the control sequence after the current iteration satisfies the constraints. If the constraints are not satisfied, the Lagrange multiplier will be updated. Finally, only the first control sequence is retained.

For the iLQR algorithm, the prediction horizon length and the number of iterations can affect the solver speed [31]. Because of the receding horizon nature of MPC, within an MPC controller framework, the number of iterations per time step can be significantly fewer compared with the standalone iLQR. However, the overall number of iterations is greater, which may result in a longer total computation time for the iLQR-based MPC algorithm compared with the single iLQR solution.

Nevertheless, this characteristic acts as a feedback mechanism, effectively correcting accumulated errors and significantly improving robustness.

In contrast, standard NMPC requires solving a constrained nonlinear optimization problem at each time step, which involves high computational cost. The iLQR algorithm, on the

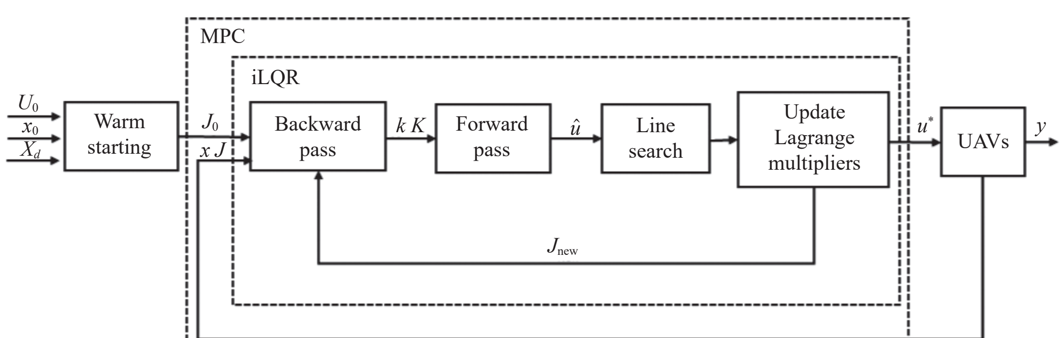


Fig. 1 iLQR-based MPC controller frame.

other hand, employs a structured approach based on the local linearization of dynamics and backward pass, iteratively refining the control sequence to approximate the optimal solution. As a result, its per-iteration computation is much more efficient than that of the general NMPC solvers.

We take $V(x,i)$ as an optimal value function, and the dynamic programming estimate form of the optimal value function is

$$V(x,i)=\min_{u}\left[\tau\ell(x,u)+V(f(x,u),i+1)+C(u_i)\right] \tag{13}$$

This equation can be expanded as a function of perturbations as

$$\begin{aligned}Q(\delta x,\delta u)&=\tau\ell(x+\delta x,u+\delta u)-\tau\ell(x,u)+C(u+\delta u)-\\&C(u)+V(f(x+\delta x,u+\delta u),i+1)-V(f(x,u),i+1)\end{aligned} \tag{14}$$

It is expanded to the second order in the following form

$$Q(\delta x,\delta u)\approx\frac{1}{2}\begin{pmatrix}1\\\delta x\\\delta u\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}0&Q_x^{\mathrm{T}}&Q_u^{\mathrm{T}}\\Q_x&Q_{xx}&Q_{xu}\\Q_u&Q_{ux}&Q_{uu}\end{pmatrix}\begin{pmatrix}1\\\delta x\\\delta u\end{pmatrix} \tag{15}$$

To simplify notation, V' is used in place of $V'\equiv V(i+1)$. In the last three equations, the last terms are ignored in iLQR because they are the second-order terms of dynamics.

$$\begin{aligned}&Q_x=\tau\ell_x+f_x^{\mathrm{T}}V_x',\\&Q_u=\tau\ell_u+f_u^{\mathrm{T}}V_x'+I_\mu c_u+\lambda,\\&Q_{xx}=\tau\ell_{xx}+f_x^{\mathrm{T}}V_{xx}'f_x+V_x'f_{xx},\\&Q_{uu}=\tau\ell_{uu}+f_u^{\mathrm{T}}V_{xx}'f_u+V_x'f_{xx}+I_\mu,\\&Q_{ux}=\tau\ell_{ux}+f_u^{\mathrm{T}}V_{xx}'f_x+V_x'f_{ux}\end{aligned} \tag{16}$$

Then, the quadratic approximation of the value function with respect to the input δu is minimized to obtain the optimal solution

$$\delta u^*=-Q_{uu}^{-1}(Q_u+Q_{ux}\delta x) \tag{17}$$

Therefore, the feed forward correction matrix and the feedback gain matrix are written as

$$\begin{aligned}&k_u=-Q_{uu}^{-1}Q_u,\\&K_u=-Q_{uu}^{-1}Q_{ux}\end{aligned} \tag{18}$$

Substituting the optimal policy into Eq. (15), we get

$$\begin{aligned}&\Delta\ V(i)=-\frac{1}{2}Q_uQ_{uu}^{-1}Q_u,\\&V_x(i)=Q_x-Q_{xu}Q_{uu}^{-1}Q_u,\\&V_{xx}(i)=Q_{xx}-Q_{xu}Q_{uu}^{-1}Q_{ux}\end{aligned} \tag{19}$$

Equation (15) can be used iteratively with Eq. (16) from the end of the time horizon to the beginning to compute the estimations of the control gains and the value function.

The final component of iLQR algorithm is the forward pass, which constitutes the forward propagation of the new trajectory. This is done through the following equations

$$\begin{aligned}&\hat{x}(1)=x(1),\\&\hat{u}(i)=u(i)+k_u(i)+K_u(i)(\hat{x}(i)-x(i)),\\&\hat{x}(i+1)=f(\hat{x}(i),\hat{u}(i))\end{aligned} \tag{20}$$

C. Regularization and Line Search

This paper mainly refers to Ref. [32] to improve the basic algorithm. A regularization factor γ and a term that penalizes state deviations are introduced to enhance the algorithm's robustness against the nonpositive definite control cost Hessian matrices. The improved formulas are

$$\begin{aligned}&\tilde{Q}_{uu}=\tau\ell_{uu}+C_{uu}+f_u^{\mathrm{T}}(V_{xx}'+\gamma I_n)f_u+V_x'\cdot f_{uu},\\&\tilde{Q}_{ux}=\ell_{ux}+f_u^{\mathrm{T}}(V_{xx}'+\gamma I_n)f_x+V_x'\cdot f_{ux},\\&k_u=-\tilde{Q}_{uu}^{-1}\tilde{Q}_u,\\&K_u=-\tilde{Q}_{uu}^{-1}\tilde{Q}_{ux}\end{aligned} \tag{21}$$

Substituting the improved feed forward correction matrix and feedback gain matrix into Eq. (19), it becomes

$$\begin{aligned}&\Delta V(i)=-\frac{1}{2}{k_u}^{\mathrm{T}}\tilde{Q}_{uu}k_u+{k_u}^{\mathrm{T}}Q_u,\\&V_x(i)=Q_x+{K_u}^{\mathrm{T}}\tilde{Q}_{uu}k_u+{K_u}^{\mathrm{T}}Q_u+\tilde{Q}_{ux}^{\mathrm{T}}k_u,\\&V_{xx}(i)=Q_{xx}+{K_u}^{\mathrm{T}}\tilde{Q}_{uu}K_u+{K_u}^{\mathrm{T}}\tilde{Q}_{ux}+\tilde{Q}_{ux}^{\mathrm{T}}K_u\end{aligned} \tag{22}$$

Considering that our goal is to reduce the cost function after iterations, it is crucial to distinguish whether the cost function decreases after an iteration, which determines whether to accept the new control law. As a further improvement, a backtracking line search parameter $0<\varepsilon\leqslant 1$ is added to Eq. (20). The scaled version of the entire step becomes

$$\hat{u}(i)=u(i)+\varepsilon k_u(i)+K_u(i)(\hat{x}(i)-x(i)) \tag{23}$$

Therefore, it is possible to decide whether to accept the improvement at a given iteration based on the ratio z between the actual cost reduction and the expected cost reduction.

$$\begin{aligned}&z=\left(J\left(u_{1,2,\ldots,N-1}\right)-J\left(\hat{u}_{1,2,\ldots,N-1}\right)\right)/\Delta J(\varepsilon),\\&\Delta J(\varepsilon)=\varepsilon\sum_{i=1}^{N-1}k_u(i)^{\mathrm{T}}Q_u(i)+\frac{\varepsilon^2}{2}\sum_{i=1}^{N-1}k_u(i)^{\mathrm{T}}\tilde{Q}_{uu}(i)k_u(i)\end{aligned} \tag{24}$$

To determine whether to accept the iteration, the ratio of the actual cost reduction of the new trajectory to the actual cost of the trajectory is calculated. If the actual reduction is sufficient, the adjustment of this step is accepted. After each iteration, continuous optimization is carried out by changing the value of ε.

D. iLQR-MPC Algorithm

We developed a trajectory tracking control method based on iLQR-MPC, achieving trajectory tracking control within a limited time domain. By implementing iLQR within the MPC framework, the tracking error can be continuously reduced through the iterative cycles. Algorithm 1 summarizes the proposed method.

IV. Simulation

In the simulation part, this paper also sets up two sets of control experiments. All the prediction horizon is 40, sampling time $T=0.1$ s. Initial position is $[0,0,0]$. The initial control sequence is selected within the horizon time span, with a sampling interval of T, according to the variation pattern of the following trigonometric function.

Algorithm 1 Trajectory tracking control based on iLQR-MPC

Input

Initial state: $x(0)$

Initial control law: $u(i)$

Reference trajectory: x_d

Warm starting

Output $x(i+1) = f(x(i), u(i))$

$$J_0(x_i, u_i) = \sum_{i=0}^{N-1} \tau_i \times \ell(x_i, u_i) + \ell_f(x_N) + C(u_i)$$

repeat

  repeat

    Backward pass

    Regularization and variable calculation: $Q_x, Q_u, \tilde{Q}_{uu}, \tilde{Q}_{ux}, k_u, K_u, V_x, V_{xx}$

    repeat

      Line search

      Change the value of ε $(0 < \varepsilon < 1)$

      Forward pass

      $\hat{u}(i) = u(i) + k_u(i) + K_u(i)(\hat{x}(i) - x(i))$

      $\hat{x}(i+1) = f(\hat{x}(i), \hat{u}(i))$

      Updata Lagrange multiplier

      New cost function:

$$\hat{J}(\hat{x}_i, \hat{u}_i) = \sum_{t=0}^{N-1} \tau_i \times \ell(\hat{x}_i, \hat{u}_i) + \ell_f(x_N) + C(u_i)$$

      Calculate variables: $\Delta J(\varepsilon)$

      $z = (J(x_i, u_i) - \hat{J}(\hat{x}_i, \hat{u}_i))/\Delta J(\varepsilon)$

    Until the expected cost decreases: $\Delta J(\varepsilon) > 0$, $z > 0$

  Until iteration completed

  Update the optimal control law: u^*

  Only apply the first one

  Update state trajectory: $x(i+1) = f(x(i), u^*(i))$

Update cost function: J

Until MPC completed

$$\begin{cases} u_{1,3} = 4 + \sin(t), \\ u_{2,4} = 3 + \cos(t) \end{cases} \tag{25}$$

The MPC controller is implemented using MATLAB's fmincon function with the sequential quadratic programming (SQP) algorithm. The iLQR controller is configured with 10 iterations, whereas the iLQR-MPC controller is configured with 2 iterations.

The weight matrix of the cost function is specified in Table 1. In the iLQR-MPC algorithm, α is set to 10. The reference trajectories are set as a helical ascending curve. The reference trajectory equations are given by

$$\begin{cases} x = 1 + \sin(t), \\ y = \cos(t), \\ z = t \end{cases} \tag{26}$$

Fig. 2 shows the trajectory tracking situations of the quadrotor aircraft under three control strategies over time, respectively. It can be seen that all control methods can follow the reference trajectory well, but there are differences in their dynamic response and steady-state accuracy. Among them, the iLQR method shows a relatively fast response speed in the initial stage, but it experiences large fluctuations in the subsequent process. The MPC method maintains a relatively smooth trajectory throughout the tracking process, but its convergence speed is relatively slow. The iLQR-MPC control method combines the advantages of both, ensuring a fast response and also achieving a high steady-state accuracy. In particular, in the z-axis direction, its trajectory is the closest to the reference trajectory. Fig. 2(d) presents the tracking error changes of the three control strategies throughout the tracking process. It can be seen that the iLQR method has a smaller error in the initial stage, but as the tracking time increases, its error gradually increases and stabilizes. The error of the MPC method changes relatively smoothly, but its steady-state error is relatively high. The iLQR-MPC method maintains a low error level throughout the tracking process, especially in the steady-state stage, and its error is significantly lower than the other two methods, indicating that this hybrid control strategy has a significant effect in improving tracking accuracy.

Fig. 3 presents the changes in the Euler angles of the unmanned aerial vehicle and its tracking performance. From this figure, it is clearly observable that the controllers have effectively stabilized the changes in the Euler angles. Fig. 3(d) shows the changes in tracking errors for the four control strategies throughout the state process. It can be seen that the iLQR method has a smaller error in the initial stage, but as the tracking time increases, its error gradually increases and stabilizes. The error of the MPC method changes relatively smoothly, but its steady-state error is relatively higher. The iLQR-MPC method maintains a lower error level throughout the tracking process, especially in the steady-state stage, its error is significantly lower than the other two methods.

Fig. 4 illustrates the input response of the unmanned aircraft. All three methods can effectively constrain the input variables.

As can be seen from the runtime in Table 1, although the iLQR-based MPC control strategy has a longer solving time than the standard iLQR algorithm, it achieves faster convergence and superior robustness. Moreover, its execution speed is faster than that of the conventional MPC approach.

V. CONCLUSIONS

This paper proposes an iLQR-based MPC control strategy for the trajectory tracking of quadrotor UAVs. By using the augmented Lagrangian method, a corresponding iLQR-MPC algorithm is designed to implement the proposed strategy. It is

Table 1 Weight matrix and runtime.

Weight matrix	Q	R	Q_f	Runtime (s)
iLQR	diag{10, 10, 10, 1, 1, 1, 10, 10, 10, 1, 1, 1}	diag{1}	diag{10}	4.12
MPC	diag{1, 1, 1, 0.1, 0.1, 0.1, 1, 1, 1, 0.1, 0.1, 0.1}	diag{0.01}	diag{10}	14.30
iLQR-MPC	diag{100, 100, 100, 1, 1, 1, 100, 100, 100, 1, 1, 1}	diag{1}	diag{100}	10.80

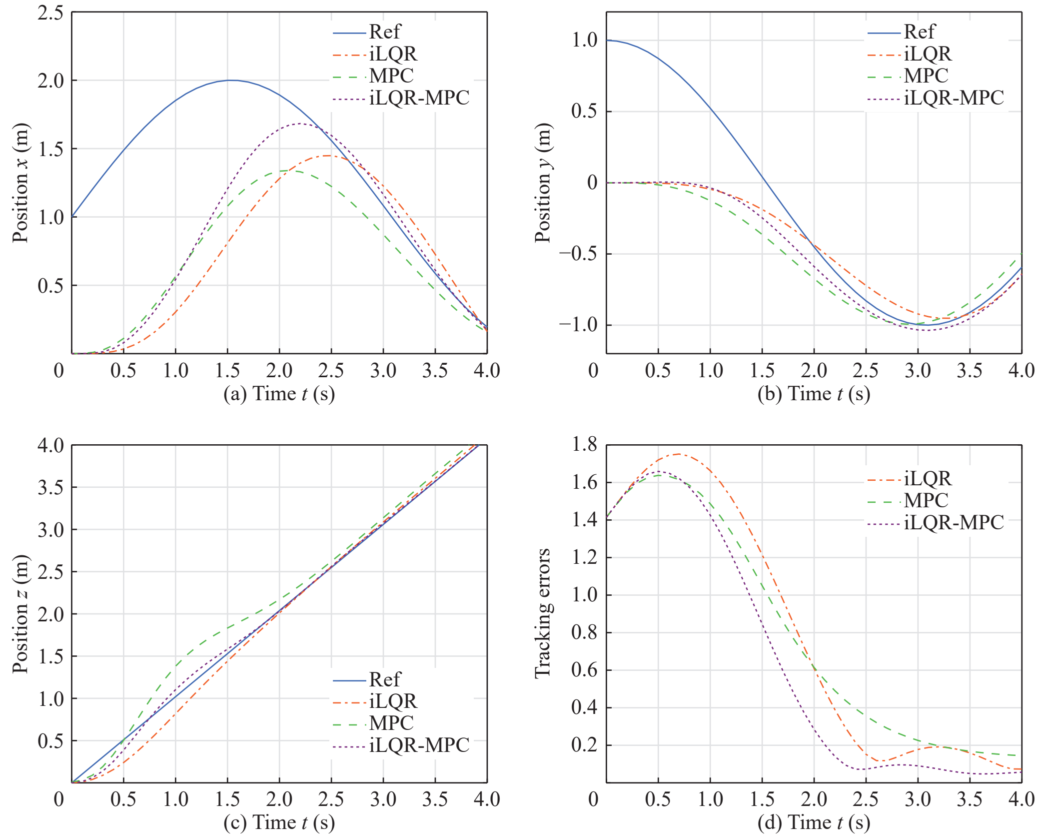


Fig. 2 Trajectory tracking and tracking errors in three-axis direction.

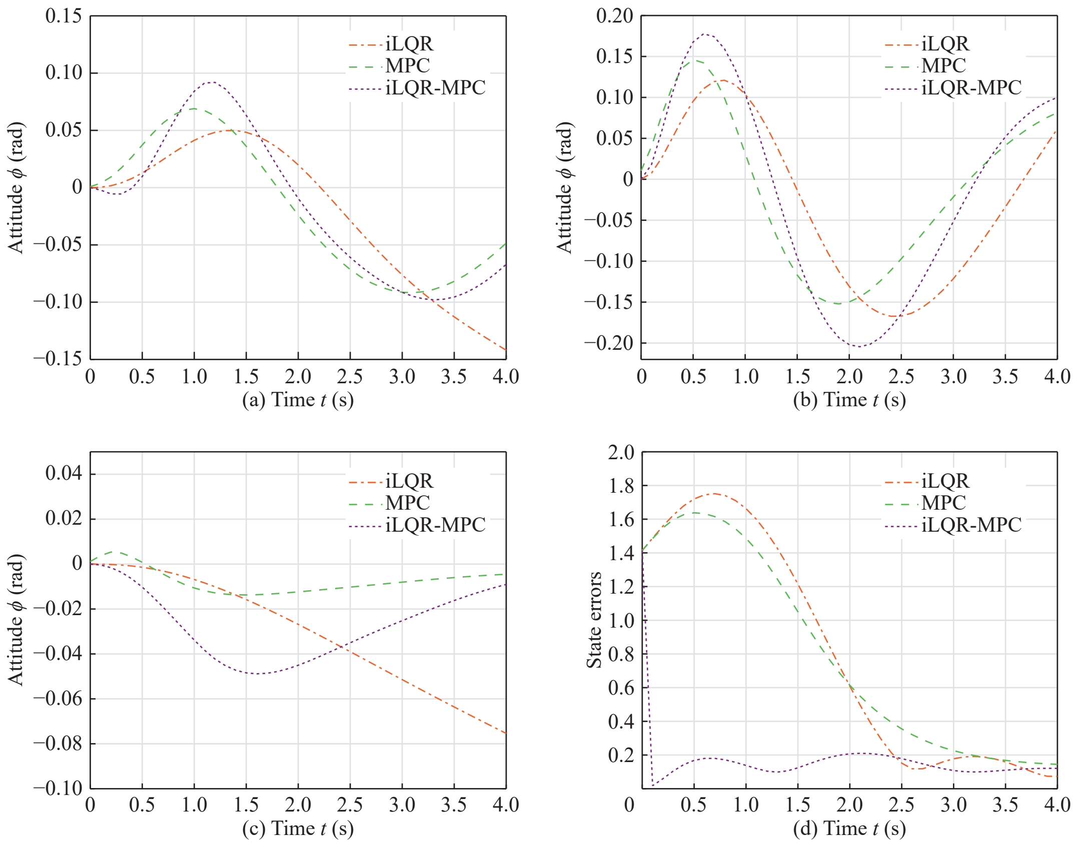


Fig. 3 Euler angle state change and state error.

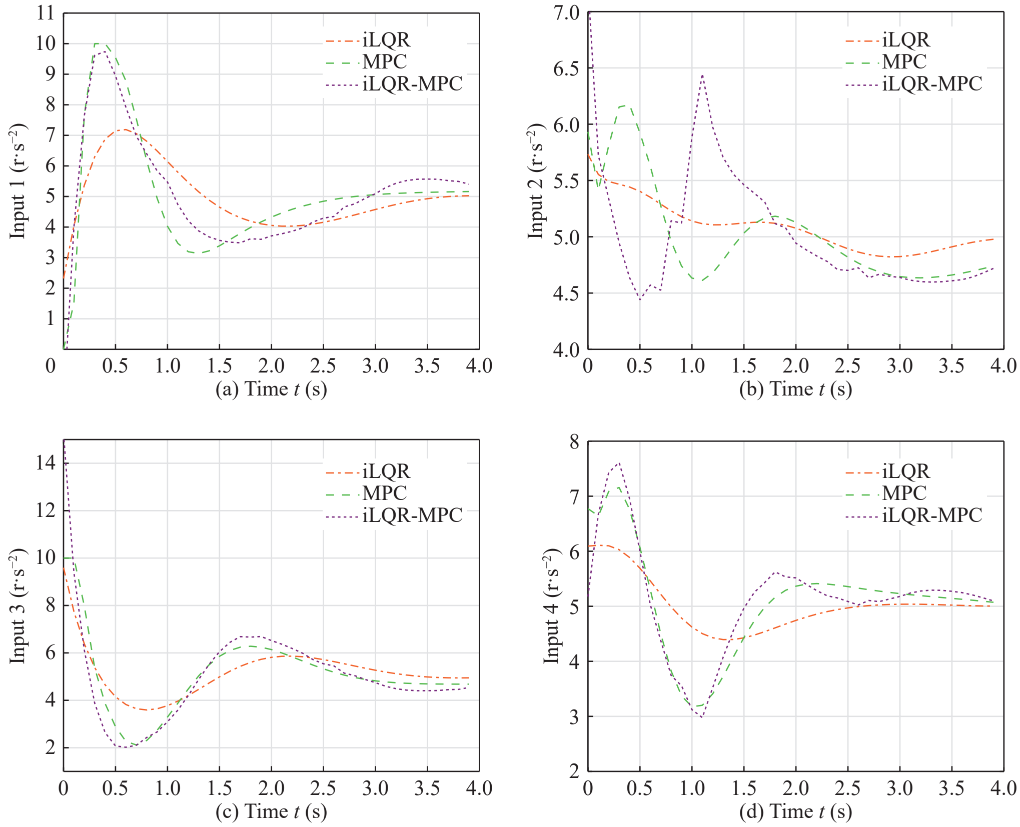


Fig. 4 Input response.

verified that the trajectory tracking performance with the iLQR-MPC control is superior to that of the MPC control and the iLQR control through simulation experiments. In the future, we will apply this strategy to a quadrotor UAV to demonstrate its practical application potential and devote to improve its robustness in the highly dynamic environments. In conclusion, this paper provides a solution for trajectory tracking control and lays a solid foundation for subsequent research, indicating multiple potential directions for future development.

REFERENCES

[1] A. Khalid, Z. Mushtaq, S. Arif, K. Zeb, M. A. Khan, and S. Bakshi, Control schemes for quadrotor UAV: Taxonomy and survey, *ACM Comput. Surv.*, 2024, 56(5), 124.

[2] J. Baek and M. Kang, A synthesized sliding-mode control for attitude trajectory tracking of quadrotor UAV systems, *IEEE/ASME Trans. Mechatron.*, 2023, 28(4), 2189–2199.

[3] I. Lopez-Sanchez and J. Moreno-Valenzuela, PID control of quadrotor UAVs: A survey, *Annu. Rev. Control*, 2023, 56, 100900.

[4] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and Z. Gao, UAV trajectory planning for AoI-minimal data collection in UAV-aided IoT networks by transformer, *IEEE Trans. Wirel. Commun.*, 2023, 22(2), 1343–1358.

[5] S. Zeghlache, H. Rahali, A. Djerioui, H. Mekki, L. Benyettou, and M. F. Benkhoris, Adaptive integral sliding mode control with chattering elimination considering the actuator faults and external disturbances for trajectory tracking of 4Y octocopter aircraft, *Processes*, 2024, 12(11), 2431.

[6] S. Ghambari, M. Golabi, L. Jourdan, J. Lepagnot, and L. Idoumghar, UAV path planning techniques: A survey, *RAIRO-Oper. Res.*, 2024, 58(4), 2951–2989.

[7] A. A. Najm and I. K. Ibraheem, Nonlinear PID controller design for a 6-DOF UAV quadrotor system, *Eng. Sci. Technol. Int. J.*, 2019, 22(4), 1087–1097.

[8] F. Parivash and A. Ghasemi, Trajectory tracking control for a quadrotor using fuzzy PID control scheme, in *Proc. 4th International Conference on Knowledge-Based Engineering and Innovation*, Tehran, Iran, 2017, 553–558.

[9] L. Martins, C. Cardeira, and P. Oliveira, Linear quadratic regulator for trajectory tracking of a quadrotor, *IFAC-PapersOnLine*, 2019, 52(12), 176–181.

[10] J. Xu, J. Liu, Z. Zhang, and W. Wang, Q-learning for linear quadratic optimal control with terminal state constraint, *Int. J. Intell. Control Syst.*, 2024, 29(3), 134–140.

[11] T. Issac, F. Thomas, and S. J. Mija, Trajectory tracking of unmanned helicopter using optimized integral LQR controller, in *Proc. 4th International Conference on Inventive Systems and Control*, Coimbatore, India, 2020, 324–328.

[12] N. A. Bakshi, Model reference adaptive control of quadrotor UAVs: A neural network perspective, in *Adaptive Robust Control Systems*. London, UK: IntechOpen, 2018.

[13] M. Menebo, L. Negash, and D. Shiferaw, Neural network based model reference adaptive control of quadrotor UAV for precision agriculture, in *Proc. 2nd Conference on Pan-African Conference on Artificial Intelligence*, Addis Ababa, Ethiopia, 2024, 171–193.

[14] M. A. Najafqolian, K. Alipour, R. Mousavifard, and B. Tarvirdizadeh, Control of aerial robots using convex QP LMPC and learning-based explicit-MPC, *IEEE Trans. Industr. Inform.*, 2024, 20(9), 10883–10891.

[15] A. Iskander, O. Elkassed, and A. El-Badawy, Minimum snap trajectory tracking for a quadrotor UAV using nonlinear model predictive control, in *Proc. 2nd Novel Intelligent and Leading Emerging Sciences Conference*, Giza, Egypt, 2020, 344–349.

[16] Z. Zuo, Trajectory tracking control design with command-filtered compensation for a quadrotor, *IET Control Theory Appl.*, 2010, 4(11), 2343–2355.

[17] S. Martini, M. Stefanovic, A. Rizzo, M. J. Rutherford, P. Livreri, and K. P. Valavanis, A benchmark framework for testing, evaluation, and comparison of quadrotor linear and nonlinear controllers, in *Proc. 2023 International Conference on Unmanned Aircraft Systems*, Warsaw, Poland, 2023, 471–478.

[18] T. Espinoza, A. Dzul, and M. Llama, Linear and nonlinear controllers applied to fixed-wing UAV, *Int. J. Adv. Robot. Syst.*, 2013, 10(1), 33.

[19] L. Xu, B. Tian, Z. Li, X. Zhang, and Q. Liu, Event-triggered MPC with nonlinear high-gain disturbance observer for trajectory tracking of quadrotor, in *Proc. 43rd Chinese Control Conference*, Kunming, China, 2024, 3000–3005.

[20] H. Yang, M. K. M. Jaffar, and M. W. Otte, Trajectory tracking while stabilizing an inverted pendulum on a quadcopter using adaptive model-predictive control, in *Proc. AIAA SCITECH 2024 Forum*, Orlando, FL, USA, 2024, 0713.

[21] B. S. Guevara, L. F. Recalde, J. Varela-Aldás, V. H. Andaluz, D. C. Gandolfo, and J. M. Toibero, A comparative study between NMPC and baseline feedback controllers for UAV trajectory tracking, *Drones*, 2023, 7(2), 144.

[22] Y. Alothman and D. Gu, Quadrotor transporting cable-suspended load using iterative linear quadratic regulator (iLQR) optimal control, in *Proc. 8th Computer Science and Electronic Engineering*, Colchester, UK, 2016, 168–173.

[23] D. Wang, Q. Pan, Y. Shi, J. Hu, and C. Zhao, Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment, *IEEE Trans. Cybern.*, 2021, 51(10), 5057–5068.

[24] D. Yuan and Y. Wang, An unmanned vehicle trajectory tracking method based on improved model-free adaptive control algorithm, in *Proc. 9th Data Driven Control and Learning Systems Conference*, Liuzhou, China, 2020, 996–1002.

[25] J. Xia and Z. Zhou, Model predictive control based on ILQR for tilt-propulsion UAV, *Aerospace*, 2022, 9(11), 688.

[26] J. Chen, W. Zhan, and M. Tomizuka, Constrained iterative LQR for on-road autonomous driving motion planning, in *Proc. 20th International Conference on Intelligent Transportation Systems*, Yokohama, Japan, 2017, 1–7.

[27] N. Ahmed and M. J. Er, Command-filtered robust trajectory tracking control for aggressive maneuvers of quadrotor UAV with multiple unknown disturbances, *Eng. Sci. Technol. Int. J.*, 2024, 59, 101858.

[28] J. Chen, W. Zhan, and M. Tomizuka, Autonomous driving motion planning with constrained iterative LQR, *IEEE Trans. Intell. Veh.*, 2019, 4(2), 244–254.

[29] Y. Pan, Q. Lin, H. Shah, and J. M. Dolan, Safe planning for self-driving via adaptive constrained ILQR, in *Proc. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, USA, 2020, 2377–2383.

[30] S. Kawarabayashi, K. Uchiyama, and K. Masuda, Trajectory optimization with collision avoidance for quadrotor using augmented Lagrangian iLQR, in *Proc. 18th International Conference on Control, Automation, Robotics and Vision*, Dubai, United Arab Emirates, 2024, 645–650.

[31] V. Kurtz and H. Lin, Contact-implicit trajectory optimization with hydroelastic contact and iLQR, in *Proc. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyoto, Japan, 2022, 8829–8834.

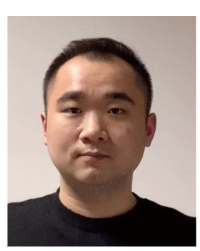
[32] Y. Tassa, T. Erez, and E. Todorov, Synthesis and stabilization of complex behaviors through online trajectory optimization, in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 2012, 4906–4913.

Jun Ren received the BS degree in electrical engineering and automation from Xinxiang Institute of Engineering, China, in 2023. He is currently pursuing the MS degree in engineering science at Shanghai University of Engineering Science, Shanghai, China. His research interests include flight control systems.

Mingyuan Zhai received the BS degree in automation and the MS degree in power electronics and transmission from Northeastern University, Shenyang, China, in 2007 and 2009, respectively. Since 2009, he has been working at Flight Control Department of Shenyang Aircraft Design and Research Institute as a senior control engineer. He is currently pursuing the PhD degree in control science and engineering at College of Information Science and Engineering, Northeastern University, Shenyang, China. His research interests include the design of advanced flight control systems.

Yanhui Tong received the BS degree in automation and the MS degree in navigation, guidance, and control from Northeastern University, Shenyang, China, in 2007 and 2009, respectively, and the PhD degree in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2013. From November 2013 to September 2017, he worked at Shanghai Institute of Electro-Mechanical Engineering, Shanghai Academy of Spaceflight Technology, Shanghai, China, as a control system engineer. From July 2022 to August 2023, he was a visiting scholar at University of Duisburg-Essen, Duisburg, Germany. He is currently an associate professor at School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China. His research interests include switched system control, model predictive control and their applications in flight control systems, and robotic systems.

Lidong He received the BEng degree in mechanical engineering from Zhejiang Ocean University, Zhoushan, China, in 2005, the MS degree from Northeastern University, Shenyang, China, in 2008, and the PhD degree in control science and engineering from Shanghai Jiaotong University, Shanghai, China, in 2014. Between 2010 and 2011, he was a visiting student at Science and Technology, Hong Kong University, Hong Kong, China. From 2014 to 2016, he was a postdoctoral researcher at Zhejiang University, Hangzhou, China. In 2016, he joined School of Automation, Nanjing University of Science and Technology, Nanjing, China. He is currently an associate professor. His research interests include secure estimation and control for cyber physical systems and its application in unmanned systems. He is an active reviewer for many international journals.



Yudong Wang received the PhD degree in computer application technology from Institute of Automation, Chinese Academy of Sciences (CASIA), China, in 2022. Since 2022, he has been working at National Key Laboratory for Multi-modal Artificial Intelligence Systems, CASIA, China. His research interests include self-learning optimal control, industrial anomaly detection based on deep learning, etc.