2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Attitude Control for Morphing Quadrotor through Model Predictive Control with Constraints*

Na Zhao, Yudong Luo†, Chaojun Qin, Xi Luo, Rong Chen and Yantao Shen

Abstract— **Morphing quadrotors that can be potentially applied to confined spaces such as warehouses, tanks, and pipelines have flourished in recent years. Most work has focused on the mechanical feasibility of the morphing systems and high-level flight controller design, with limited discussions on low-level control. In this paper, a constrained model predictive control (MPC) is proposed and applied to solve the attitude control problem of a morphing quadrotor. Prior to controller design, a custom-built morphing quadrotor is introduced with the kinematic and dynamic models established and corresponding issues and challenges presented. In the controller, to eliminate the steady-state error, an embedded integrator is adopted by exploiting the differential variables; then, the constraints of the morphing quadrotor are incorporated into the MPC formulation to simulate real flight conditions, and an orthonormal function is employed to approximate the control input sequences in the controller to alleviate the computational burden. In the comparative studies, several scenarios are considered to demonstrate the effectiveness of the proposed control strategy in attitude control.**

I. Introduction

Morphing quadrotors have occupied a prominent place in the domain of aerial rotor systems in the most recent years due to their capabilities of geometric adaptation to small openings and cluttered spaces as well as morph-enabled various functions and performance [1]–[6]. Most current related work focuses on mechanism design and high-level motion planning, while few studies exist on their low-level flight controller design [7]. The performance of the low-level controller for a morphing quadrotor is very important because varying geometric parameters induced inertial variation, model and dynamics uncertainties, and the limits of the physical system would directly affect the attitude of the quadrotor, which further deteriorates the tracking performance [7], making the study on this topic very challenging.

Model predictive control (MPC) performs well in dealing with multiple variables, constraints, and online process optimization [8], and it also demonstrates the simplicity of the design framework in handling many complex issues. The general idea of MPC is to incorporate state information, future references and constraints into the control, and predict how a system can be controlled in an optimum way. Compared to the standard PID control, which can be considered "reactive" because it steers the system towards a desired state based on previous state information. When exposed to unknown dynamics such as wind, variable payloads, and voltage sag, a PID controller can be far from optimal [9]. MPC can be considered "proactive" because it guides the system towards a future reference based on state information and a prediction of how the system can be controlled. Besides, different from the classic linear quadratic regulator (LQR), MPC solves the optimization problem along a moving horizon window, while LQR works within a fixed window. This property allows MPC to deal more freely with optimization problems with different types of constraints and uncertainties. From this point of view, MPC is very suitable for solving the aforementioned problems existing in morphing quadrotors.

However, the most criticized problem of MPC is its large computational cost as it involves lots of matrix and vector operations and the control inputs required to be computed online in each sampling time by taking all constraints into the control. This problem becomes even more prominent in practical implementations of a complex dynamic system, where a large control horizon is required, thus resulting in dramatically increased computation. To address this issue, a hierarchical MPC is proposed in [10], which divides the controller into multiple layers such as position, attitude, and motor with different sampling times for real-time implementation. Besides, a model reduction method is presented in [11] to alleviate the computational burden. In [8], [12], the approach that uses orthonormal functions such as Laguerre function and Kautz function are proposed to replace the forward shift operators to capture the future control trajectory for reducing computation load as this method uses fewer optimization variables. Similar work can be found in [13], [14]. Although quite a few works have been done on applying MPC to conventional quadrotors [15]–[17], its application to morphing quadrotors is still a new attempt. In [18], a model predictive framework for collision-free navigation of a morphing quadrotor is proposed by adding environment constraints (eg. restricted entrances). The simulation results demonstrate the feasibility of the morphing quadrotor passing through different scenarios, such as spherical and cubic entrances. In [19], an MPC that switches between the different structural formations is explored. However, little work has addressed the controller performance under disturbances and

*This work was supported by the National Natural Science Foundation of China under Grant 52305009 and the Fundamental Research Funds for the Central Universities under Grant 3132023266.

Na Zhao, Yudong Luo†, Chaojun Qin and Rong Chen are with the Department of Computer Science and Technology, Dalian Maritime University, Dalian, Liaoning 116026, China.

Xi Luo is with Yichang Testing Tech. Research Institution, Yichang, Hubei 410083, China.

Yantao Shen is with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557, USA.

†Corresponding author. Email: *ydluo@dlmu.edu.cn*.

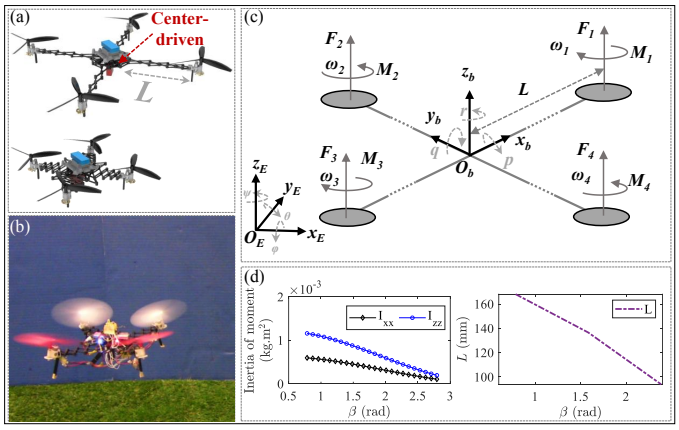

Fig. 1. Morphing quadrotor system: (a) actuation, (b) prototype, (c) coordinate system, and (d) varying inertia property.

the real-time performance.

In this paper, we first introduce the custom-built morphing quadrotor that requires only one central actuator and build the corresponding kinematic and dynamic model. Then, we construct an MPC with constraints based on an augmented model with integrators to eliminate the steady-state error to ensure accurate trajectory tracking. The proposed control framework can handle multiple state variables and control inputs with constraints during different configurations and the induced inertial variations caused by structural transformations. To reduce the computation cost, the Laguerre function is employed to approximate the control input sequence. Given that in the particular case of a quadrotor, which is a Multiple-Input Multiple-Output (MIMO) system, a state space model is easier to construct and better suited for evaluating stability and robustness, which will be employed in the following work. To the best of our knowledge, this is the first attempt to apply MPC with an embedded integrator into the control of the morphing quadrotor.

The rest of the paper is organized as follows. Section II states the problem that exists in the morphing quadrotor and presents the morphing system dynamics. Section III discusses the design of the constrained MPC controller based on an augmented model. To capture the performance of low-level control, this section details the MPC controller for the attitude loop. Section IV presents the comparative results in different scenarios. Finally, the work is summarized, and the discussion is concluded.

II. Quadrotor System and Dynamic Modeling

A. Morphing system

The morphing quadrotor is shown in Fig. 1(a) and (b), which is designed based on a center-driven planar closed-loop mechanism [20], [21]. The mechanism consists of four identical arms with a single servomotor at the center of the body frame. Compared to the morphing mechanisms currently used in many morphing quadrotors, the main advantages of adopting this type of mechanism for constructing the morphing quadrotor are as follows: i) fewer actuators are used to achieve a larger morphing ratio, and ii) the morphing mechanism is evenly distributed, which structurally ensures the self-balance of the body and avoids excessive use of controller leveling [20].

The following definitions and symbols are given to better describe the system. The radius of the quadrotor is defined as L, referring to the distance between the center of the quadrotor and that of the propeller motor, as illustrated in Fig. 1(c). Different from the conventional quadrotor, L is a varying parameter for the morphing quadrotor, which is related to the controlled deployable angle of the servomotor, denoted as β_s. As investigated in [20], the identified length is $\widetilde{L} \in$[0.09517, 0.1681] m, and inertia of moment are as follows: $\widetilde{I}_{xx} = \widetilde{I}_{yy} \in [0.1, 0.6] \times 10^{-3}$ kg·m^2, and $\widetilde{I}_{zz} \in [0.2, 1.2] \times 10^{-3}$ kg·m^2. Here is the expression of the inertia of the moment around each axis with regard to the deployable angle $\beta_s \in [45°, 160°]$.

B. System dynamics

For a clear description of the dynamics of the morphing system, we let $\mathcal{B} = \{x_b y_b z_b\}$ represent the body frame of the quadrotor, and $\mathcal{E} = \{x_E y_E z_E\}$ represent the fixed inertial frame, as shown in Fig. 1(c). The origin of $\mathcal{B}$ coincides with the mass center of the quadrotor, designated as O_b. Define $\boldsymbol{p} = [x, y, z]^{\mathrm{T}}$ as the position of the quadrotor along x, y, and z expressed in the fixed inertial frame $\mathcal{E}$, and $\boldsymbol{\gamma} = [\phi, \theta, \psi]^{\mathrm{T}}$ as the attitude of the quadrotor with respect to $\mathcal{E}$. F_i, M_i, and ω_i $(i = 1 \sim 4)$ correspond to the force, the moment, the angular velocity generated by the i^{th} motor. τ_x, τ_y, and τ_z represent the torque around corresponding axis in $\mathcal{B}$ frame.

We start with the analysis of forces and torques exerted on the quadrotor. The total thrust that is generated by four motors is $T = \sum_{i=1}^{4} F_i$, which is independent of the morphology as the quadrotor morphs parallel to the z_b-axis. Therefore, the corresponding dynamics can be achieved by following the standard dynamic model. The torque vector, $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^{\mathrm{T}}$, is expressed as

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} (F_2 - F_4)L \\ (F_3 - F_1)L \\ M_2 + M_4 - M_1 - M_3 \end{bmatrix} \tag{1}$$

The dynamic model of the quadrotor can be divided into two subsystems, translational movement (position) and rotation movement (attitude).

Position subsystem. The translational dynamics of the quadrotor are expressed as follows:

$$m\ddot{\boldsymbol{p}} = [0 \ 0 \ -mg]^{\mathrm{T}} + \bar{\boldsymbol{R}}_{\boldsymbol{1}} T, \tag{2}$$

where m is the total mass of the quadrotor, g is the gravity acceleration, $\bar{\boldsymbol{R}}_1$ is the transform matrix with $\bar{\boldsymbol{R}}_{\boldsymbol{1}} = [c\phi c\psi s\theta + s\phi s\psi \quad c\phi s\psi s\theta - c\psi s\phi \quad c\phi c\theta]^{\mathrm{T}}$, and cx and sx represent trigonometric functions $\cos x$ and $\sin x$, respectively.

The translational dynamics (2) can be approximated around the hovering condition, where small attitude angles are assumed. This assumption makes sense because one of the original purposes of designing a morphing quadrotor was to reduce or even avoid maneuvering flights and fly as horizontally as possible [22]. Hence, the translational

dynamics around the hovering condition can be expressed as

$$\ddot{\boldsymbol{p}} = [0\ 0\ -g]^{\mathrm{T}} + \bar{\boldsymbol{R}}_{\boldsymbol{1new}}[\phi\ \theta\ T]^{\mathrm{T}}, \tag{3}$$

where $\bar{\boldsymbol{R}}_{\boldsymbol{1new}} = [0\ g\ 0; -g\ 0\ 0; 0\ 0\ \frac{1}{m}]^{\mathrm{T}}$.

Attitude subsystem. Denote the rotational velocity vector, also termed as body rate, as $\boldsymbol{\omega} = [p\ q\ r]^{\mathrm{T}}$ to describe the angular motion of the body in the body frame. The attitude dynamics of the quadrotor are expressed as

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}(-\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{\tau}), \tag{4}$$

where the inertia matrix, $\boldsymbol{I} = diag(\widetilde{I}_{xx}, \widetilde{I}_{yy}, \widetilde{I}_{zz})$, only consists of diagonal terms as the quad-rotor is assumed to be symmetric along its axes.

The rotational velocities $\boldsymbol{\omega} = [p\ q\ r]^{\mathrm{T}}$ in the body frame are then converted to the Euler angle velocities in the fixed inertial frame as follows:

$$[\dot{\phi}\quad \dot{\theta}\quad \dot{\psi}]^{\mathrm{T}} = \bar{\boldsymbol{R}}_{\boldsymbol{2}}[p\quad q\quad r]^{\mathrm{T}} \tag{5}$$

with $\bar{\boldsymbol{R}}_{\boldsymbol{2}} = [1\ t\theta s\phi\ t\theta c\phi; 0\ c\phi\ -s\phi; 0\ s\phi/c\theta\ c\phi/c\theta]$, where $tx = \tan x$, and the small angle assumption allows the system to avoid singularities ($c\theta \neq 0$) and make the sinusoidal value zero, thus $[\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T \approx [p\ q\ r]^{\mathrm{T}}$.

Therefore, the attitude dynamics (4) can be linearlized around the operating point as (6), which satisfies $\dot{\phi} \approx 0$, $\dot{\theta} \approx 0$, and $\dot{\psi} \approx 0$.

$$[\ddot{\phi}\quad \ddot{\theta}\quad \ddot{\psi}]^{\mathrm{T}} = [\frac{\tau_x}{\widetilde{I}_{xx}}\quad \frac{\tau_y}{\widetilde{I}_{yy}}\quad \frac{\tau_z}{\widetilde{I}_{zz}}]^{\mathrm{T}} \tag{6}$$

C. Generalized state-space model

With (3) and (6), the following two linear subsystems can be described as

$$\begin{aligned} [\dot{\boldsymbol{p}}\quad \ddot{\boldsymbol{p}}]^{\mathrm{T}} &= A_1[\boldsymbol{p}\quad \dot{\boldsymbol{p}}]^{\mathrm{T}} + B_1\boldsymbol{u}_1 \\ \boldsymbol{y} &= C_1[\boldsymbol{p}\quad \dot{\boldsymbol{p}}]^{\mathrm{T}} \end{aligned} \tag{7}$$

$$\begin{aligned} [\dot{\boldsymbol{\gamma}}\quad \ddot{\boldsymbol{\gamma}}]^{\mathrm{T}} &= A_2[\boldsymbol{\gamma}\quad \dot{\boldsymbol{\gamma}}]^{\mathrm{T}} + B_2\boldsymbol{u}_2 \\ \boldsymbol{y} &= C_2[\boldsymbol{p}\quad \dot{\boldsymbol{p}}]^{\mathrm{T}} \end{aligned} \tag{8}$$

where $\boldsymbol{u}_1 = [\phi, \theta, T]^{\mathrm{T}}$, $\boldsymbol{u}_2 = [\tau_x, \tau_y, \tau_z]^{\mathrm{T}}$, ($A_1$, B_1, C_1) and (A_2, B_2, C_2) are used to define the position and attitude subsystems in the form of state-space equations, respectively.

It can be concluded that the position subsystem that (A_1, B_1, C_1) defines is a linear time-invariant (LTI) system, which is not different from the conventional system, while the attitude subsystem that (A_2, B_2, C_2) defines is subject to the varying inertia matrix, which makes (8) become a linear parameter-varying (LPV) system. For this reason, the attitude subsystem is highlighted in the following content.

Consider employing an explicit fourth-order Runge-Kutta method, $\boldsymbol{x}(k+1) = \boldsymbol{f}_{RK4}(\boldsymbol{x}(k), \boldsymbol{u}(k), \Delta t_{RK4})$ to incorporate the dynamics in discrete time with an integration step Δt_{RK4} [15]. Hence, the generalized form of the attitude subsystem can be expressed as

$$\begin{aligned} \boldsymbol{x}(k+1) &= A_{2d}(\beta_s)\boldsymbol{x}(k) + B_{2d}(\beta_s)\boldsymbol{u}_2(k) \\ \boldsymbol{y}(k) &= C_{2d}\boldsymbol{x}(k) \end{aligned} \tag{9}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{N_x}$ is the state vector, $\boldsymbol{u}(k) \in \mathbb{R}^{N_u}$ is the control input variables, $\boldsymbol{y}(k) \in \mathbb{R}^{N_y}$ is the controlled output variables, and the subscript d indicates discretization.

III. Controller Formulation

This section presents the formulation of the MPC by incorporating varying parameters and practical constraints to solve the tracking problem and preserve the quadrotor's stability. Fig. 2 illustrates the MPC scheme of the morphing quadrotor, where a predefined reference trajectory is generated by the position subsystem to be tracked to produce the desired thrust, T, as well as roll angle and pitch angle, ϕ_d and θ_d, respectively, thus guiding the attitude subsystem to solve the control inputs, τ_x, τ_y, τ_z, for stabilization.

A. State-space model with embedded integrator

When $\boldsymbol{x}(k)$ is measurable, the attitude subsystem (9) can be used directly as a prediction model. To reduce static errors, this section first presents the incremental state-space model based on (9) by embedding an integrator. Let's parameterize the problem of (9) using the difference of the state variable and the input increments, $\Delta x(k+1) = x(k+1) - x(k)$ and $\Delta u(k) = u(k) - u(k-1)$ to reduce or eliminate static errors. The model (9) can be modified to an incremental model (10) with a new state variable vector $\boldsymbol{x}(k) = [\Delta x(k), y(k)]$ and a new input variable $\Delta \boldsymbol{u}(k)$ to the state space model.

$$\begin{aligned} \underbrace{\begin{bmatrix} \Delta \boldsymbol{x}(k+1) \\ \boldsymbol{y}(k+1) \end{bmatrix}}_{\boldsymbol{x}(k+1)} &= A \underbrace{\begin{bmatrix} \Delta \boldsymbol{x}(k) \\ \boldsymbol{y}(k) \end{bmatrix}}_{\boldsymbol{x}(k)} + B\Delta \boldsymbol{u}(k), \\ \boldsymbol{y}(k) &= C[\Delta \boldsymbol{x}(k)\ \boldsymbol{y}(k)]^{\mathrm{T}}, \end{aligned} \tag{10}$$

where $A = \left[A_{2d}(\beta_s)\ O; C_{2d}A_{2d}(\beta_s)\ \boldsymbol{I}_{N_y \times N_y}\right]$, $B = [B_{2d}\ C_{2d}B_{2d}]^{\mathrm{T}}$, and $C = [\boldsymbol{O}_{N_x \times N_x}\ 1]$.

At this point, the augmented model of the morphing quadrotor is established, and the next step in the design of a predictive control system is to calculate the predicted plant output with the future control signal as the adjustable variables, consider the constraints, as well as conduct the optimization based on the cost function.

B. Prediction of states and outputs

The prediction is described within a pre-selected prediction horizon, thus we define the prediction horizon as N_p and the control horizon as N_c, which satisfies $N_c \leq N_p$. The future state variables are $x(k+1|k), x(k+1|k), \cdots, x(k+m|k), \cdots, x(k+N_p|k)$, where $x(k+N_p|k) = A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1)$. In a similar way, the predicted output can be described as $y(k+N_p|k) = CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1)$. Define vectors $\boldsymbol{Y}$ and $\Delta \boldsymbol{U}$ as $\boldsymbol{Y} = [y(k+1|k)\ y(k+2|k)\ y(k+3|k) \cdots y(k+N_p|k)]^T$ and $\Delta \boldsymbol{U} = [\Delta u(k)\ \Delta u(k+1)\ \Delta u(k+2) \cdots \Delta u(k+N_c-1)]^T$, then a compact matrix can be described as

$$\boldsymbol{Y} = \boldsymbol{F}\boldsymbol{x}(k) + \boldsymbol{H}\Delta \boldsymbol{U}, \tag{11}$$

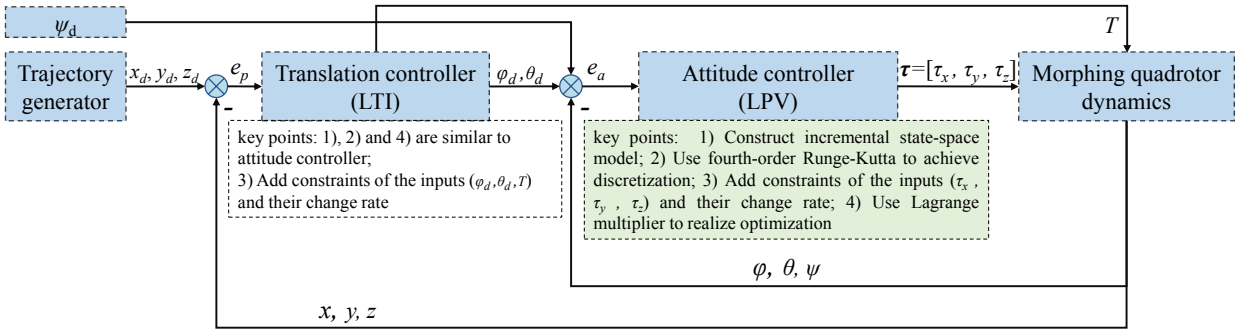

Fig. 2. MPC scheme of the morphing quadrotor.

where $\boldsymbol{F} = [CA \quad CA^2 \quad CA^3 \cdots \quad CA^{N_p}]^{\mathrm{T}}$, and

$$\boldsymbol{H} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ CA^2B & CAB & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & \cdots & CA^{N_p-N_c}B \end{bmatrix}.$$

C. Constraints handling

In actual trajectory tracking tasks, the morphing quadrotor is subject to physical constraints such as restricted dimensions and limited control capacity, which can be either soft constraints that may be violated if necessary or hard constraints that must be satisfied. Consider the case of the quadrotor, which achieves control by varying the angular speeds of each motor, thus adjusting the force varying within a range dependent on the motor specification.

In this study, we consider the control input $\boldsymbol{u}(k)$ and their rates $\Delta\boldsymbol{u}(k)$ in the MPC design, $\boldsymbol{u}_{min}(k) \leq \boldsymbol{u}(k) \leq \boldsymbol{u}_{max}(k)$ and $\Delta\boldsymbol{u}_{min} \leq \Delta\boldsymbol{u}(k) \leq \Delta\boldsymbol{u}_{max}$ with $\forall k \geq 0$, where subscripts min and max indicate the predefined lower and upper bounds, respectively. This means when designing the position subsystem (7), the constraints on the control inputs, $\boldsymbol{u}(k) = [\phi(k), \theta(k), T(k)]^T$ and their rates, $\Delta\boldsymbol{u}(k) = [\Delta\phi(k), \Delta\theta(k), \Delta T(k)]^T$ will be considered. Similarly, the constraints on the control inputs, $\boldsymbol{u}(k) = [\tau_x(k), \tau_y(k), \tau_z(k)]^T$ and their rates, $\Delta\boldsymbol{u}(k) = [\Delta\tau_x, \Delta\tau_y, \Delta\tau_z]^T$ will be considered in the attitude subsystem (8). The constraints on the rates of input change can be grouped into the form,

$$[-I \ I]^T \Delta\boldsymbol{u} \leq [-\Delta u_{min} \ \Delta u_{max}]^T. \tag{12}$$

Then, the input constraints can be written in a form that incorporates the rates of input change.

$$\boldsymbol{u} = \boldsymbol{C}_1 u(k-1) + \boldsymbol{C}_2 \Delta u, \tag{13}$$

where $\boldsymbol{u} = [u(k) \ u(k+1) \ \ldots \ u(k+N_c-1)]^{\mathrm{T}} \in [\boldsymbol{u}_{min}, \boldsymbol{u}_{max}]$, $\Delta u = [\Delta u(k) \ \Delta u(k+1) \ \ldots \ \Delta u(k+N_c-1)]^{\mathrm{T}} \in [\Delta\boldsymbol{u}_{min}, \Delta\boldsymbol{u}_{max}]$, $C_1 = [I \ I \ \ldots \ I \ I]^{\mathrm{T}}_{N_c \times 1}$, and $C_2 = [I \ 0 \ 0 \ \cdots \ 0; I \ I \ 0 \ \cdots \ 0; \cdots; I \ I \ I \ \cdots \ I]_{N_c \times N_c}$.

By combining (12) and (13), the following compact linear constraint inequality can be obtained.

$$\underbrace{\begin{bmatrix} -\boldsymbol{C}_2 \\ \boldsymbol{C}_2 \\ -I \\ I \end{bmatrix}}_{[\boldsymbol{CC}]} \Delta\boldsymbol{U} \leq \underbrace{\begin{bmatrix} -\boldsymbol{U}_{min} + \boldsymbol{C}_1\boldsymbol{u}(k-1) \\ \boldsymbol{U}_{max} - \boldsymbol{C}_1\boldsymbol{u}(k-1) \\ -\Delta\boldsymbol{U}_{min} \\ \Delta\boldsymbol{U}_{max} \end{bmatrix}}_{[\boldsymbol{DD}]}, \tag{14}$$

where $k = 1, \cdots, N_p$.

D. Optimization

In the most general form, MPC stabilizes a system subject to its dynamics along the desired reference values by minimizing a cost function that is comprised of the error between predicted outputs, $\boldsymbol{Y}$ and reference values, $\boldsymbol{R}_s$, subject to constraints on the system. Finally, the model predictive control in the presence of hard constraints is proposed as finding the parameter vector $\Delta\boldsymbol{U}$ that minimizes

$$\boldsymbol{J} = (\boldsymbol{R}_s - \boldsymbol{Y})^T(\boldsymbol{R}_s - \boldsymbol{Y}) + \Delta\boldsymbol{U}^T W \boldsymbol{\Delta} U, \tag{15}$$

which subjects to the inequality constraints (14) with the initial value $\boldsymbol{x}(0) = \boldsymbol{x}(t_0)$.

Since the cost function, $\boldsymbol{J}$ is quadratic, and the constraints are linear inequalities, the problem of figuring out an optimal predictive control becomes finding an optimal solution to a standard quadratic programming problem. W is a diagonal matrix in the form that $W = r_w I_{Nc \times Nc} (r_w \geq 0)$, where r_w is used as a tuning parameter for the desired closed-loop performance.

Substituting (11) into (15), we can acquire $\boldsymbol{J} = (\boldsymbol{R}_s - \boldsymbol{F}\boldsymbol{x}(k))^T(\boldsymbol{R}_s - \boldsymbol{F}\boldsymbol{x}(k)) - 2\Delta\boldsymbol{u}^T\boldsymbol{H}^T(\boldsymbol{R}_s - \boldsymbol{F}\boldsymbol{x}(k)) + \Delta\boldsymbol{u}^T(\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{W})\Delta\boldsymbol{u}$. Taking the first derivative of the cost function $\boldsymbol{J}$, we then get a new objective function,

$$\boldsymbol{J} = \tfrac{1}{2}\Delta\boldsymbol{u}^T\zeta\Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T\boldsymbol{\eta} \tag{16}$$

with constrains $[CC]\Delta\boldsymbol{u} \leq [DD]$, where $\boldsymbol{\zeta} = 2(\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{W})$ and $\boldsymbol{\eta} = -2\boldsymbol{H}^T(\boldsymbol{R}_s - \boldsymbol{F}\boldsymbol{x}(k))$, which are compatible matrices in the quadratic programming problem.

To solve this quadratic programming problem and reduce the computational load, a method of Lagrange multiplier, is used to optimize (16), which can be derived as follows with λ as the decision variable, $\min_{\lambda \geq 0}[\frac{1}{2}\lambda^T\boldsymbol{K}_1\lambda + \lambda^T\boldsymbol{K}_2 +$

TABLE I

PARAMETERS OF THE MORPHING QUADROTOR

m (kg)	0.12	g (m/s^2)	9.81
β	45°	94.5°	160°
L (m)	$L_{\text{min}} = 0.095$	L_{med}=0.132	L_{max}=0.168
I_{xx} (kg·m^2)	0.1×10^{-3}	0.4×10^{-3}	0.6×10^{-3}
I_{yy} (kg·m^2)	0.1×10^{-3}	0.4×10^{-3}	0.6×10^{-3}
I_{zz} (kg·m^2)	0.2×10^{-3}	0.8×10^{-3}	1.2×10^{-3}

$\frac{1}{2}[\boldsymbol{DD}]^T\boldsymbol{\zeta}^{-1}[\boldsymbol{DD}]]$, which subjects to $\lambda \geq 0$, and the matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are given by $\boldsymbol{K}_1 = [\boldsymbol{CC}]\boldsymbol{\zeta}^{-1}[\boldsymbol{CC}]^T$ and $\boldsymbol{K_2} = [\boldsymbol{DD}] + [\boldsymbol{CC}]\boldsymbol{\zeta}^{-1}\boldsymbol{\eta}$. The primal variable vector $\Delta\boldsymbol{u}$ is attained by

$$\Delta\boldsymbol{u} = -\boldsymbol{\zeta}^{-1}(\boldsymbol{\eta} + [\boldsymbol{CC}]^T\lambda). \tag{17}$$

IV. COMPARATIVE STUDIES AND RESULTS

The purpose of designing the model prediction controller is to track the desired trajectory $\boldsymbol{p}_d = [x_d, y_d, z_d]^T$ by applying constraints to the control inputs including limiting Euler angles and thrust within a range in the position loop to avoid large Euler angles that may cause the quadrotor depart from the operating region, as well as limiting torques within a range in the attitude loop to avoid the saturation of the motor's output. As MPC considers constraints, and with that comes inherent advantages over other controllers, instead of demonstrating the comparison of different controllers, this study focuses on the application of MPC with constrains to the morphing quadrotor and investigate the attitude performance of the qudarotor under different parameters and physical system conditions to prove its feasibility. In the following section, the attitude performance of the quadrotor will be emphasized and several scenarios are considered to validate the performance of the proposed controller. Tab. I illustrates the physical parameters of the custom-built morphing quadrotor.

A. Test I: comparision of different operation frequencies

As one of the important parameters of MPC, the prediction time domain, which indicates how many time steps to predict in the future, determines the degree of the controller's prediction of the future states. Different prediction time domains produce different results. By choosing a hardware controller with suitable operation frequency, the controller can better predict the future output of the system and correct the input signal in time according to the current error. Below we tested the performance of the controller at different operating frequencies to provide a basis for hardware controller selection.

The quadrotor in different sizes at three operation frequencies was conducted. Fig. 3 reprsentatively demonstrates the attitude tracking performance of the quadrotor in minimum state in different sampling rates. The desired Euler angles of the quadrotor are [0.2 0.2 0.2] rad (solid black line). It can be noticed that the required control inputs $[\tau_1, \tau_2, \tau_3]$ with their changing rates $[\Delta\tau_1, \Delta\tau_2, \Delta\tau_3]$ are limited within a range (dashed black line). Besides, it can be seen from the Euler angles' plots $[\phi, \theta, \psi]$ that a higher operation frequency of

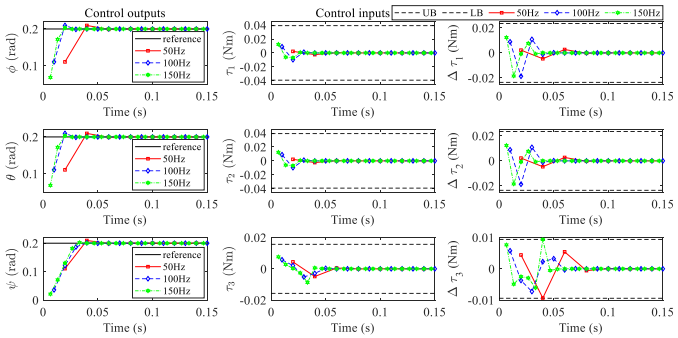

Fig. 3. Comparison of the control inputs and outputs of the quadrotor with different sampling rates (50 Hz, 100 Hz and 150 Hz). UB: upper bound, LB: lower bound.

the controller is conducive to a lower overshoot and faster stabilization. For example, it takes 0.07 seconds for roll angle ϕ to reach the desired 0.2 rad under 50 Hz with an overshoot of 0.209 rad, while 0.04 seconds under 100 Hz with an overshoot of 0.208 rad and 0.03 seconds under 150 Hz with an overshoot of 0.202 rad, which also applies to θ and ψ.

B. Test II: comparision of different prediction horizons

Immediately following Test I, we selected different prediction horizons, N_p to compare the attitude performance of the quadrotor. Empirically, the control horizon N_c is greater than 3 and satisfies $N_c \in [0.1, 0.2]N_p$, thus the following possible combinations are chosen for testing.

$$\begin{aligned}[N_p, N_c] =&[\{10,3\},\{20,3\},\{20,4\},\\ &\{30,3\},\{30,4\},\{30,5\},\{30,6\}]\end{aligned}$$

The results are plotted in Fig. 4, where three colored lines, red, blue, and green, represent the results under predicted horizons of 10, 20, and 30, respectively, and if the prediction horizon corresponds to different control horizons, then distinguish them by line styles. It can be noticed from the plots of ϕ, θ and ψ in Fig. 4 that when $[N_p, N_c]$=$[\{10,3\},\{20,3\},\{30,3\}]$, the overshoot, defined as os, is smaller than the rest of conditions, and follows $os_{\{30,3\}} < os_{\{20,3\}} < os_{\{10,3\}}$. This indicates that the control horizon $N_c = 3$ is the optimal value in this case.

To determine a suitable N_p, we further investigate the computation cost corresponding to each combination. The computation time of the controller in different conditions is collected on the same platform. The platform used to solve the optimization problem is a personal computer with AMD Ryzen 7 6800H with Radeon Graphics @ 3.2GHz processor with 16 GB of RAM and no parallel computing was used. The histogram in Fig. 4 demonstrates the average time cost of the five repeated experiments with different combinations of $[N_p, N_c]$ under different operation frequencies. In general, with a few exceptions, an increase in operation frequency or a decrease in the prediction horizon reduces the average calculation time. The computation time is defined as T_{ct} for ease of description. According to the bar plot, we have

$$T_{ct(50\text{Hz})} \in [0.33, 0.50]\text{ms} \leq \frac{1}{50\text{Hz}},$$

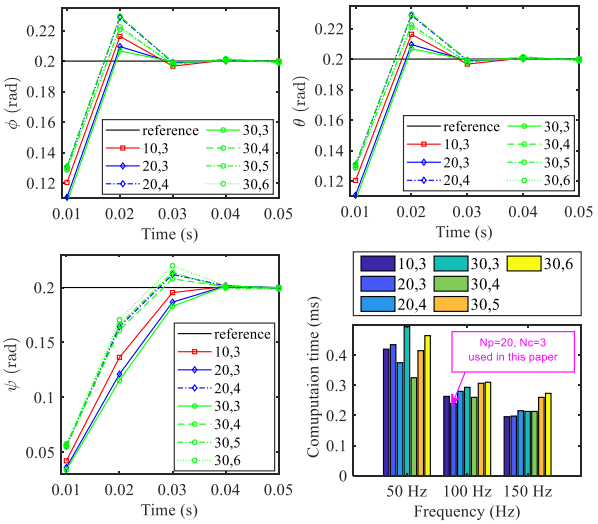

Fig. 4. Comparison of the attitude performance and computation cost with different N_c and N_p.

$$T_{ct(100\text{Hz})} \in [0.24, 0.31]\text{ms} \leq \frac{1}{100\text{Hz}},$$

and

$$T_{ct(150\text{Hz})} \in [0.20, 0.27]\text{ms} \leq \frac{1}{150\text{Hz}},$$

which reveals that the computation time is significantly lower than the one required by the controller frequency, meaning that it is enough to execute the optimization in a real-time scheme. Given that the hardware controller we used for implementation is 100 Hz, thus we chose the case with the least time expenditure, i.e. $N_p = 20$ and $N_c = 3$ in this study.

C. Test III: comparison of different sizes and robustness

Based on the results acquired from Test I and Test II, we further tested the performance of the quadrotor in different sizes and investigated their in-flight anti-disturbance capabilities with the controller frequency of 100 Hz, $N_p = 20$ and $N_c = 3$. Two indicative types of tests were conducted.

The first is the test of the quadrotor's attitude performance in different sizes without disturbance. It should be noted that the change in size leads to a change in constraints. For ease of description, here we define min, med, and max to represent the minimum, medium, and maximum states of the quadrotor, respectively, so each state has corresponding constraints, as shown in Fig. 5, where there are three pairs of constraints for the control inputs, represented by dashed red, blue, and green lines, respectively. It can be noticed that the proposed controller can figure out the optimal solution within the respective constraints to achieve the desired attitude angle with very small tracking error.

After the quadrotor stabilized, a disturbance was applied to test the control performance at different sizes. The random disturbance, denoted as rd, is defined as $rd_x = rand(1), rd_y = rand(1), rd_z = 0.1 \times rand(1)$, where $rand(1)$ is a normal distribution noise with a mean of 0 and variance of 1. Fig. 6 shows the response of control inputs and outputs of the quadrotor in three different sizes when

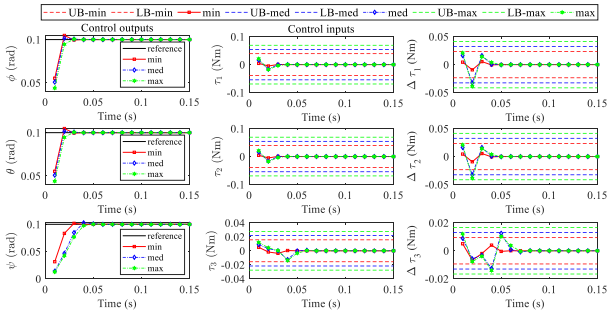

Fig. 5. Comparison of the attitude performance in different sizes (Controller frequency 100 Hz, and $[N_p, N_c] = [20, 3]$).

dealing with the disturbance applied at 0.3 sec. It can be seen that i) the proposed controller can eliminate the impact of disturbance within 0.05$\sim$ 0.07 seconds to stabilize the quadrotor, ii) all the control inputs and their corresponding changing rates are within the required constraints, and iii) when dealing with disturbance, the bigger quadrotor, which benefits from the increase of its own force arm, can instantly generate a larger torque to make the overshoot of the quadrotor smaller and stabilize faster.

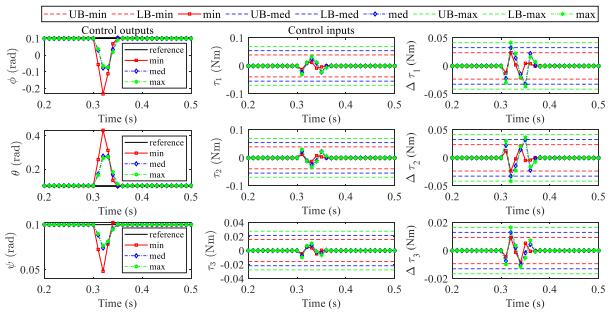

Fig. 6. Comparison of disturbance in different sizes under the condition of controller frequency 100 Hz, and $[N_p, N_c] = [20, 3]$.

V. Conclusion

To improve the current situation of little study on low-level controllers in the field of morphing quadrotors, a model predictive control considering constraints and embedded integrators is proposed to solve the problem of variable parameters in morphing quadrotors. Firstly, a custom-built morphing quadrotor was introduced to provide the kinematic and dynamic model for the following study. Then, an embedded integrator was employed in the state-space model to eliminate the steady-state error; fourth-order Runge-Kutta was used to discretize the system model; the Lagrange multiplier was implemented to approximate the control input sequences to reduce the time cost when executing the optimization. Several scenarios were conducted to demonstrate the tracking performance of the controller under different operation frequencies and different prediction horizons, and to prove the effectiveness of the proposed control strategy in disturbance rejection and stability. Our ongoing work is to adopt this method to the flight transition of different morphing states and include extensive experimental results to validate the performance of the proposed controller.

REFERENCES

[1] Y. Wu, F. Yang, Z. Wang, K. Wang, Y. Cao, C. Xu, and F. Gao, "Ring-rotor: A novel retractable ring-shaped quadrotor with aerial grasping and transportation capability," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2126–2133, 2023.

[2] N. Bucki, J. Tang, and M. W. Mueller, "Design and control of a midair-reconfigurable quadcopter using unactuated hinges," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 539–557, 2023.

[3] M. Zhao, K. Okada, and M. Inaba, "Enhanced modeling and control for multilinked aerial robot with two dof force vectoring apparatus," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 135–142, 2021.

[4] N. Zhao, Y. Luo, G. Wang, and Y. Shen, "A deployable articulated mechanism enabled in-flight morphing aerial gripper," *Mechanism and Machine Theory*, vol. 167, p. 104518, 2022.

[5] K. Patnaik and W. Zhang, "Towards reconfigurable and flexible multi-rotors," *International Journal of Intelligent Robotics and Applications*, vol. 5, no. 3, pp. 365–380, 2021.

[6] B. Mu and P. Chirarattananon, "Universal flying objects: Modular multirotor system for flight of rigid objects," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 458–471, 2019.

[7] K. Patnaik and W. Zhang, "Adaptive attitude control for foldable quadrotors," *IEEE Control Systems Letters*, 2023.

[8] L. Wang, *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

[9] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for uav attitude control," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, pp. 1–21, 2019.

[10] M. Bangura and R. Mahony, "Real-time model predictive control for quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11773–11780, 2014.

[11] M. Abdolhosseini, Y. M. Zhang, and C. A. Rabbath, "An efficient model predictive control scheme for an unmanned quadrotor helicopter," *Journal of Intelligent & Robotic Systems*, vol. 70, pp. 27–38, 2013.

[12] X. Chen and L. Wang, "Cascaded model predictive control of a quadrotor uav," in *2013 Australian Control Conference*, pp. 354–359, IEEE, 2013.

[13] J. A. Ligthart, P. Poksawat, L. Wang, and H. Nijmeijer, "Experimentally validated model predictive controller for a hexacopter," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4076–4081, 2017.

[14] A. Eskandarpour and I. Sharf, "A constrained error-based mpc for path following of quadrotor with stability analysis," *Nonlinear Dynamics*, vol. 99, pp. 899–918, 2020.

[15] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[16] D. Wang, Q. Pan, Y. Shi, J. Hu, and C. Zhao, "Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment," *IEEE Transactions on Cybernetics*, vol. 51, no. 10, pp. 5057–5068, 2021.

[17] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, "Nonlinear mpc for quadrotor fault-tolerant control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5047–5054, 2022.

[18] A. Papadimitriou, S. S. Mansouri, C. Kanellakis, and G. Nikolakopoulos, "Geometry aware nmpc scheme for morphing quadrotor navigation in restricted entrances," in *2021 European Control Conference (ECC)*, pp. 1597–1603, IEEE, 2021.

[19] A. Papadimitriou and G. Nikolakopoulos, "Switching model predictive control for online structural reformations of a foldable quadrotor," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 682–687, 2020.

[20] N. Zhao, W. Yang, C. Peng, G. Wang, and Y. Shen, "Comparative validation study on bioinspired morphology-adaptation flight performance of a morphing quad-rotor," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5145–5152, 2021.

[21] T. Yang, P. Li, Y. Shen, and Y. Liu, "Center-driven planar closed-loop mechanisms based on an angulated four-bar linkage," *Mechanism and Machine Theory*, 2023.

[22] N. Zhao, Y. Luo, H. Deng, and Y. Shen, "The deformable quadrotor: Design, kinematics and dynamics characterization, and flight performance validation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2391–2396, IEEE, 2017.