Model Predictive Control for Path Tracking of a VTOL Tail-sitter UAV in an HIL Simulation Environment

Boyang Li*, Weifeng Zhou† Jingxuan Sun‡ and Chih-Yung Wen§

The Hong Kong Polytechnic University, Hong Kong

Chih-Keng Chen ¶

National Taipei University of Technology, Taipei, Taiwan

This paper investigates the application of Model Predictive Control (MPC) for path tracking of a vertical takeoff and landing (VTOL) tail-sitter unmanned aerial vehicle (UAV) in hover flight. In this work, the nonlinear dynamic model of a quad-rotor tail-sitter UAV including the aerodynamic effect of the wing, propellers, and slipstream was developed. The cascaded MPC controllers were then built upon linearized dynamic models. Path tracking simulations were conducted in a hardware-in-loop (HIL) environment where the UAV model and controllers were running on a PC and a flight computer independently. The simulation results show that the proposed MPC controllers enable the UAV to perform good path tracking and the ability of disturbance rejection under limited onboard computation resource.

I. Nomenclature

C_L, C_D, C_M	=	lift, drag, and moment coefficient
$\bar{c}$	=	mean aerodynamic chord, m
$\mathbf{D}, \mathbf{L}$	=	drag, lift, N
d, l, s	=	distance, m
$\mathbf{e}$	=	feedback error
$\mathbf{F}$	=	drag, lift, N
H	=	prediction horizon
$\mathbf{I}$	=	inertia matrix, kg·m²
$\mathbf{J}$	=	objective function
K	=	system gain
k	=	control interval
m	=	mass, kg
$\mathbf{M}$	=	moment, N·m
N	=	control horizon
p, q, r	=	angular rate in body frame, deg/s
$\mathbf{P}$	=	position, m
$\mathbf{Q}, \mathbf{R}$	=	weight matrix
r	=	slipstream radius, m
$\mathbf{r}$	=	output reference
R	=	radius of the propeller, m
$\mathbb{R}, \mathbb{T}$	=	rotation matrix
S	=	reference area of wing, m²
$\mathbf{S}$	=	scale factors matrix
T	=	thrust, N
T_s	=	sampling time, s

*Ph.D. Student, Department of Mechanical Engineering, boyang.li@connect.polyu.hk, AIAA Student Member.

†M.Phil. Student, Department of Mechanical Engineering, chandler.zhou@connect.polyu.hk, AIAA Student Member.

‡Ph.D. Student, Department of Mechanical Engineering, jingxuan.j.sun@connect.polyu.hk, AIAA Student Member.

§Professor, Department of Mechanical Engineering, chihyung.wen@polyu.edu.hk, AIAA Associate Fellow.

¶Associate Professor, Department of Vehicle Engineering, ckchen@ntut.edu.tw.

$\mathbf{u}$	=	control input value
u, v, w	=	linear velocity in body frame, m/s
v_0	=	velocity of freestream, m/s
$\mathbf{v}$	=	velocity, m/s
$\mathbf{x}$	=	control state
$\mathbf{y}$	=	measured output value
$\mathbf{z}_k$	=	quadric programming decision factor
α	=	angle of attack, rad
$\boldsymbol{\omega}$	=	angular velocity, deg/s
ϕ, θ, ψ	=	Euler angles, deg
η	=	throttle of motor
ρ	=	air density, kg/m^3
τ	=	time constant, s

II. Introduction

Unmanned aerial vehicles (UAVs) are playing more and more critical roles in current society. They are designed to fit a vast variety of tasks such as cargo delivery, search-and-rescue, and aerial surveying. Vertical takeoff and landing (VTOL) UAVs can perform takeoff and landing without the necessity of runways or catapults while maintaining high aerodynamic efficiency during cruise stage. This advantage makes them very suitable to function in densely populated areas such as Hong Kong. There are several airframe configurations that can achieve VTOL, including tilt-wing, tilt-rotor, and tail-sitter. The tilt-wing design has engines mounted on the wing and the whole wing can rotate according to needs. The tilt-rotor has the wing fixed but rotatable engines. However, both the designs require a complex mechanical structure which severely decreases the reliability of the aerial system. The concept of a tail-sitter aircraft was firstly appeared in a patent by Nikola Tesla [?] in 1920s. It is a kind of configuration with the whole vehicle pitch down 90 degrees after takeoff. The tail-sitter UAV combines the advantages of simple structure, less weight penalty, and lower risk of failure. The cost is their high demanding to the control system during hover and transition period due to the aerodynamic nonlinearity of the wing at high angle of attack (AoA).

The tail-sitter design includes three kinds of common configuration according to the number of the propulsion source. One propeller tail-sitter UAVs [? ? ?] are similar to tradition fixed-wing UAV with landing sit mounted on the tail. This configuration is widely adopted for the micro aerial vehicle (MAV) due to the limited power and simple structure. Dual-rotor tail-sitter UAVs [? ? ?] have relatively more considerable takeoff weight and area of control surface immersed in the upstream, which can help to provide more actuator effort than the single propeller configuration. However, both the single and dual rotors configurations heavily depend on the effect of control surfaces for hover control, which result in poor performance in windy and turbulent outdoor environment. Quad-rotor designs [? ? ? ?] use the similar control principle with quadrotors during hover flight. The actuator effect provided by the differential thrust of propellers enable them with large control moments in every rotation axis. When it transits to level flight, the quad-rotor tail-sitter UAV can also choose to use control surface to achieve better efficiency. Despite the relatively large control actuator of quadrotor configuration, with the main wing area facing crosswind, it still needs a dedicate control system to ensure the tracking performance and disturbance rejection ability.

Model Predictive Control (MPC) is an optimal control using receding horizon structure. The finite time optimal control law is on-line computed by solving the optimization problem repeatedly. It is suitable for multi-input and multi-output (MIMO) systems which are governed by constrained dynamics, as the controller will account the hard constraints in the optimisation process compared to other control schemes [?]. Another advantage of MPC is that it allows the current timeslot to be optimized while keeping future timeslots into account. Nowadays, with the improvement of computation power, the application of MPC gradually moved to the control of nonlinear and fast process such as aerospace and robotic systems [?]. A. Bemporad [?] firstly proposed a hierarchical hybrid MPC controller for a quadrotor. The linear MPC controller was designed to stabilize the vehicle near the commanded set-points while an upper control layer used a hybrid MPC controller with slower sampling rate for obstacle avoidance. X. Chen. and L. Wang [?] also proposed a cascaded MPC controller for Quadrotor UAV and showed satisfactory performance. With the ability to account the limited output constraint during optimation, MPC is a promising control method for tail-sitter UAVs.

In this study, the proposed control target is a tail-sitter UAV with cross motor configuration, shown in Fig. 1. The primary airframe was modified from a commercial model called Skywalker X-5 with the rear motor removed.

The main objective of this paper is to verify the performance of MPC controller on this tail-sitter UAV in hovering stage especially for path tracking and the ability of disturbance rejection. A series of simulation were carried out in a hardware-in-the-loop (HIL) environment which shows the acceptable computation load of the MPC controller. The hardware integration also provides the primary onboard system for further experimental study.

Fig. 1 The UAV prototype "PolyU-Plus Tail-sitter".

The rest of paper is organized as follows. In Section III, the dynamic model of the UAV is described including the aerodynamics of the main wing and the propulsion system. The development of the cascaded MPC controllers is presented in Section IV. In Section V, the HIL simulation environment is described and the path tracking results are discussed. At last, the conclusion is presented in Section VI.

III. Dynamic Modeling

This section presents the development of the dynamic model of the tail-sitter UAV. The model includes a 6 degree-of-freedom (6-DOF) rigid body dynamics, an aerodynamic model of the wing and actuator model of the propulsion system.

A. Coordinate Systems

Two sets of reference system were used to describe the dynamics of the UAV. The fixed inertia coordinate system (O_e, X_e, Y_e, Z_e) is pointing North-East-Down (NED) direction of the earth. The mobile body coordinate system (O_b, X_b, Y_b, Z_b) is located at the center of gravity of the vehicle with the X_b and Y_b axis pointing to the bottom and right side of the wing respectively. The schematic diagram of the coordinate system is shown in Fig. 2.

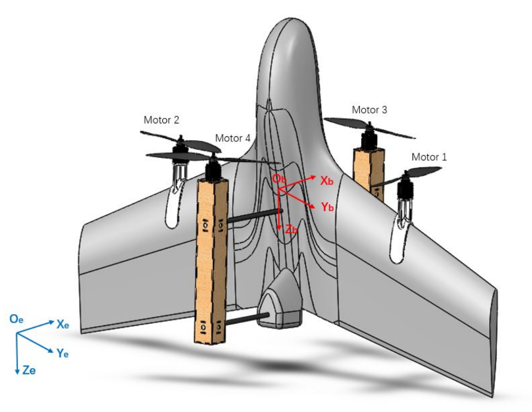

Fig. 2 The earth and body coordinate systems of the UAV.

B. Dynamics

The dynamic and kinematic model of the UAV can be described as

$$\begin{aligned}
\mathbf{v}_{\mathrm{I}} &= \mathbb{R}\mathbf{v}_{\mathrm{B}} \\
\boldsymbol{\omega}_{\mathrm{I}} &= \mathbb{T}\boldsymbol{\omega}_{\mathrm{B}} \\
m(\dot{\mathbf{v}}_{\mathrm{B}} + \boldsymbol{\omega}_{\mathrm{B}} \times \mathbf{v}_{\mathrm{B}}) &= \mathbf{F}_{\mathrm{B}} \\
\mathbf{I}\dot{\boldsymbol{\omega}}_{\mathrm{B}} + \boldsymbol{\omega}_{\mathrm{B}} \times (\mathbf{I}\boldsymbol{\omega}_{\mathrm{B}}) &= \mathbf{M}_{\mathrm{B}}.
\end{aligned} \tag{1}$$

$\mathbf{v}_{\mathrm{I}} = [\dot{x}\ \dot{y}\ \dot{z}]^T$ and $\boldsymbol{\omega}_{\mathrm{I}} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$ are the derivatives of position and Euler angle (roll, pitch, and yaw) in the inertia frame. $\mathbf{v}_{\mathrm{B}} = [u\ v\ w]^T$ and $\boldsymbol{\omega}_{\mathrm{B}} = [p\ q\ r]^T$ are the linear and angular velocity in the body frame. $\mathbb{R}$ and $\mathbb{T}$ are the rotation matrix that transfer linear and angular velocity from the body frame to the inertia frame. $\mathbf{I}$ denotes the inertia matrix and m is the mass of the vehicle.

The $\mathbf{F}_{\mathrm{B}}$ and $\mathbf{M}_{\mathrm{B}}$ are total force and moment act to the body frame, which can be presented as

$$\begin{aligned}
\mathbf{F}_{\mathrm{B}} &= \mathbf{F}_{\mathrm{aero}} + \mathbf{F}_{\mathrm{prop}} + \mathbb{R}^T\mathbf{F}_{\mathrm{g}} + \mathbf{F}_{\mathrm{d}} \\
\mathbf{M}_{\mathrm{B}} &= \mathbf{M}_{\mathrm{aero}} + \mathbf{M}_{\mathrm{prop}} + \mathbf{M}_{\mathrm{d}}.
\end{aligned} \tag{2}$$

$\mathbf{F}_{\mathrm{aero}}$ and $\mathbf{M}_{\mathrm{aero}}$ denote the aerodynamic forces and moments on the wing. $\mathbf{F}_{\mathrm{g}}$ is gravity force. $\mathbf{F}_{\mathrm{prop}}$ and $\mathbf{M}_{\mathrm{prop}}$ are the thrust and moments created by the rotating propellers. $\mathbf{F}_{\mathrm{d}}$ and $\mathbf{M}_{\mathrm{d}}$ are disturbance force and moment. The above model will be linearized on hover state based on small-disturbance theory in the next Section, which assumes the motion of the vehicle consists of small deviation about a steady condition.

C. Aerodynamic Model

In this vehicle, two of the propellers have their downstream passing through the wing. As a result, the aerodynamic effect must be considered in the hovering stage. The speed of the airflow is assumed to be uniform over the wing and assume there is no cross coupling between the left and the right wing. The wing was modeled into five parts based on a component breakdown approach [?], which are shown in Fig. 3.

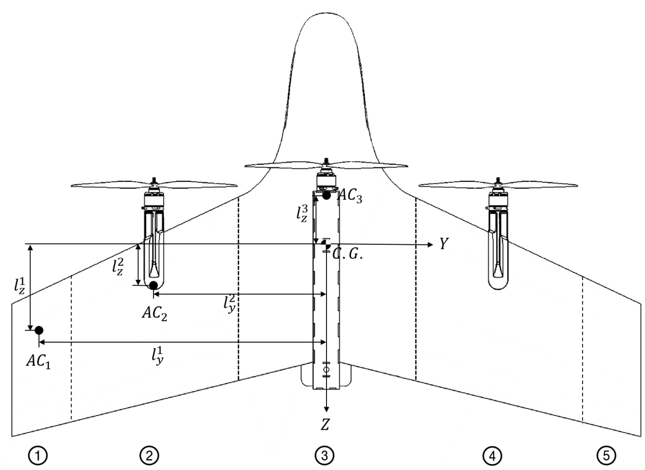

Fig. 3 Geometric parameters of the UAV for aerodynamic model.

The mean aerodynamic chord (MAC) and the position of aerodynamic center (AC) of the whole wing were calculated at the first place. The width of each segment is necessary since each segment will create lift, drag, and moments on its AC. The width of segments 2 & 4 was determined according to the slipstream radius r. By applying the momentum theory of flow, the radius r of the flow tube at distance s away to the propeller can be estimated by Eq.(**??**) [?], where R is the radius of the propeller.

$$r = R\sqrt{\frac{s^2 + R^2}{s + \sqrt{s^2 + R^2}}} \tag{3}$$

The induced velocity on segment 2 & 4 was strongly affected by the downstream of the propellers, which were consistently changed according to the throttle command. To obtain the flow condition of these parts, the airspeed $\mathbf{V}^{\mathrm{B}}_{\mathrm{as}}$ should be calculated by adding the contribution of propeller wash as

$$\mathbf{V}^{\mathrm{B}}_{\mathrm{as}} = \mathbb{R}^{\mathrm{B}}_{\mathrm{I}} \left(\mathbf{V}^{\mathrm{I}}_{\mathrm{wind}} + \mathbf{V}^{\mathrm{I}}_{\mathrm{gs}} \right) + \mathbf{V}^{\mathrm{B}}_{\mathrm{induce}}, \tag{4}$$

where the wind speed$\mathbf{V}^{\mathrm{I}}_{\mathrm{wind}}$ and ground speed$\mathbf{V}^{\mathrm{I}}_{\mathrm{gs}}$ were described in the inertia frame. $\mathbf{V}^{\mathrm{B}}_{\mathrm{induce}}$is the velocity of the propeller slipstream which is defined as

$$\mathbf{V}^{\mathrm{B}}_{\mathrm{induce}} = \begin{bmatrix} 0 \\ 0 \\ u_0 \end{bmatrix}. \tag{5}$$

u_0 can be estimated by using the continuity equation [**?**]:

$$u_0 = \frac{v_0 cos\alpha + \sqrt{v_0^2 \left(cos\alpha\right)^2 + \frac{2T}{\rho A}}}{2} \left[1 + \frac{s/R}{\sqrt{1 + \left(\frac{s}{R}\right)^2}} \right], \tag{6}$$

where α is the AoA, T is the thrust generated by the propeller, A is the area of propeller disk, v_0 is the velocity of freestream which can be expressed as:

$$v_0 = \left| \mathbf{V}^{\mathrm{I}}_{\mathrm{wind}} + \mathbf{V}^{\mathrm{I}}_{\mathrm{gs}} \right|. \tag{7}$$

The AoA of each segment is defined as:

$$\alpha = tan^{-1} \left(V^B_{\mathrm{as},x} / V^B_{\mathrm{as},z} \right), \tag{8}$$

where $V^B_{\mathrm{as},x}$ and $V^B_{\mathrm{as},z}$ are the local velocities of the segment in x and z of the body frame.

The effective airspeed on each of the wing segment is:

$$v_{\mathrm{eff}} = \left| \mathbf{V}^{\mathrm{B}}_{\mathrm{as}} \right|. \tag{9}$$

At the hovering stage, the side force and moment were ignored. A database of this airfoil which includes the lift coefficient C_L, drag coefficient C_D, and moment coefficient C_M at α from -180° - 180° and airspeed from 0 - 30 m/s was developed by conducting a series of wind tunnel experiment with the scaled model [**?**]. As the flow speed and AoA of each segment was measured and calculated, the aerodynamic coefficient can be determined by searching the data base and then the lift and drag of each segment can be calculated by:

$$\begin{aligned} L_i &= \tfrac{1}{2} \rho v^2_{eff,i} S_i C_{L,i} \\ D_i &= \tfrac{1}{2} \rho v^2_{eff,i} S_i C_{D,i}. \end{aligned} \tag{10}$$

The lift and drag on body coordinate can be express as:

$$\begin{aligned} \mathbf{L} &= \begin{bmatrix} -\sum L_i \cos\alpha_i \\ 0 \\ -\sum L_i \sin\alpha_i \end{bmatrix} \\ \mathbf{D} &= \begin{bmatrix} -\sum D_i \sin\alpha_i \\ 0 \\ \sum D_i \cos\alpha_i \end{bmatrix}, \end{aligned} \tag{11}$$

As a result, the aerodynamic force and moment can be expressed as:

$$\begin{aligned} \mathbf{F}_{\mathrm{aero}} &= \mathbf{L} + \mathbf{D} \\ \mathbf{M}_{\mathrm{aero}} &= \sum \begin{bmatrix} F^i_{\mathrm{aero},z} l^i_y \\ \frac{1}{2} C_{M,i} \rho v^2_{\mathrm{eff},i} S_i \bar{c}_i + F^i_{\mathrm{aero},x} l^i_z \\ F^i_{\mathrm{aero},x} l^i_y \end{bmatrix}. \end{aligned} \tag{12}$$

D. Actuator Model

The relationship between the throttles of the motor and force/moment of the propulsion system in hover state are developed by experiments. In the experiment, the motor and propeller are mounted on an ATI Mini40 6-DOF force/torque sensor [**?**] and then connected to an aluminum alloy bar fixed on the desk. The thrust and torque values are transduced by the sensor into an analog voltage and then collected by the data acquisition (DAQ) hardware and software system. The system configuration is shown in Fig. 4. Several sets of propeller and motor combinations were tested from 0 to 100% throttle value. The generated thrust and moments of the motor-propeller in vertical axis were collected. According to the thrust requirement and efficiency on the standard operation power, the combination of SunnySky X2212 KV980 brushless motor and APC 10*47 propeller were finally chosen.

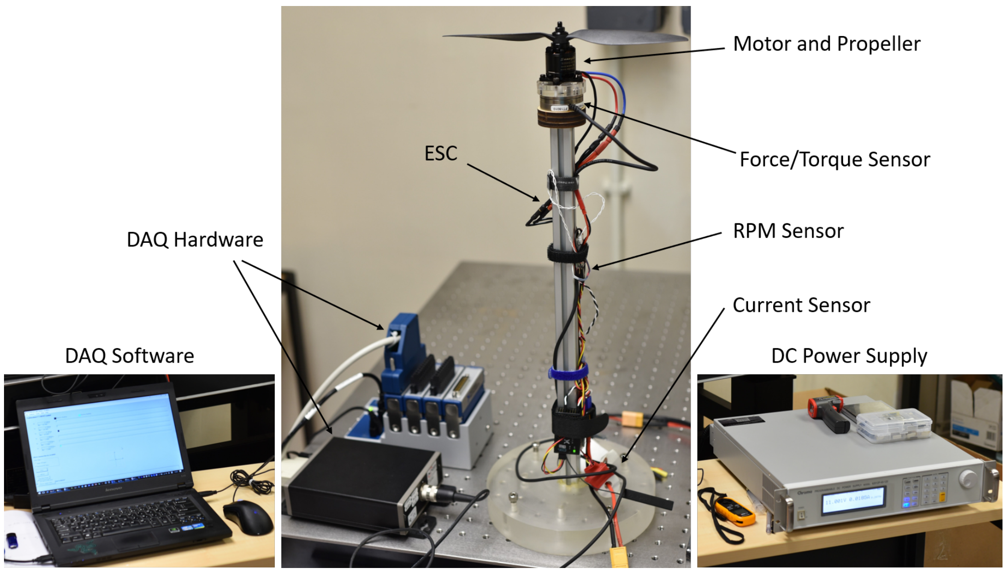

Fig. 4 The experimental setup of a thrust and moment test.

The non-linear relationship between throttle to the thrust and moment was modeled as a set of third-order polynomial functions for simulation use. The original experimental data and the fitted polynomial for thrust and moment of the chosen set of propulsion system are shown in Fig. 5.

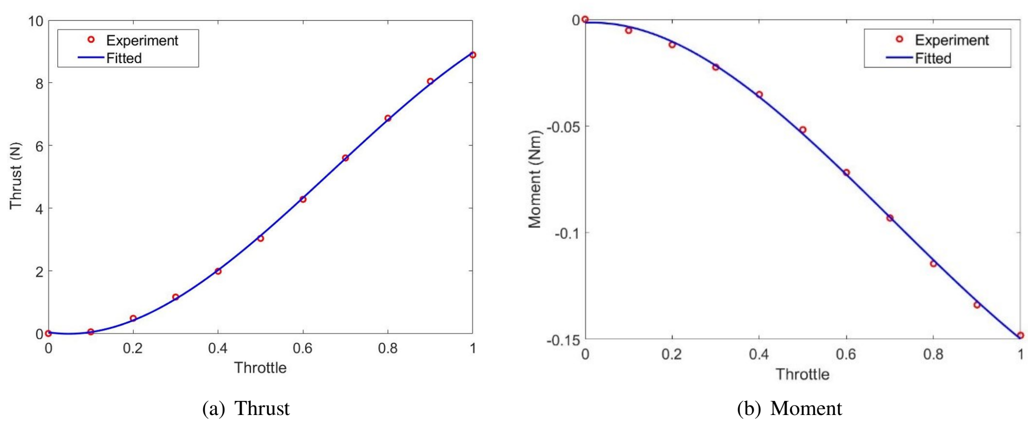

Fig. 5 The experiment data and fitted curve of the motor-propeller system.

For the whole UAV model, the thrust and moment created by the propulsion system can be defined as:

$$\mathbf{F}_{\mathrm{P}} = \begin{bmatrix} 0 \\ 0 \\ -(F_1 + F_2 + F_3 + F_4) \end{bmatrix}$$
$$\mathbf{M}_{\mathrm{P}} = \begin{bmatrix} (F_2 - F_1)l_y \\ (F_3 - F_4)l_x \\ M_1 + M_2 - M_3 - M_4 \end{bmatrix}, \tag{13}$$

where F_1, F_2, F_3, F_4 and M_1, M_2, M_3, M_4 are the thrust and moment of each motor and propeller. l_y and l_x are the distance between the center of mass to the motor along the y-axis and x-axis.

IV. Development of MPC Controllers

In this section, the dynamic model of the UAV is linearized near the operation point of hover state, and the MPC position and attitude controllers are then built according to the linear state-space model. The preview reference model was given to the cost function at the very beginning so that the controller can respond to the change of reference in beforehand. The actuator dynamics was also linearized to an actuator mapping to transform the controller output force and moment into four target throttle values.

A. Cascaded MPC Control Structure

Because of the required computation effort increases dramatically with the number of states increases in an MPC controller, the cascade structure was developed to reduce the onboard computation load. Two cascaded controllers with fewer states than a single loop controller will be more practical for onboard flight control mission. Meanwhile, this strategy allows a faster updating rate in the attitude control than the position control. Fig. 6 shows the schematic diagram of the control system, in which MPC control algorithm was applied to both the position controller and attitude controller. The "Actuator Map" is a multiplication of matrices, which converts the desired torque and moment signs into the required throttle η_i of each of the motors.

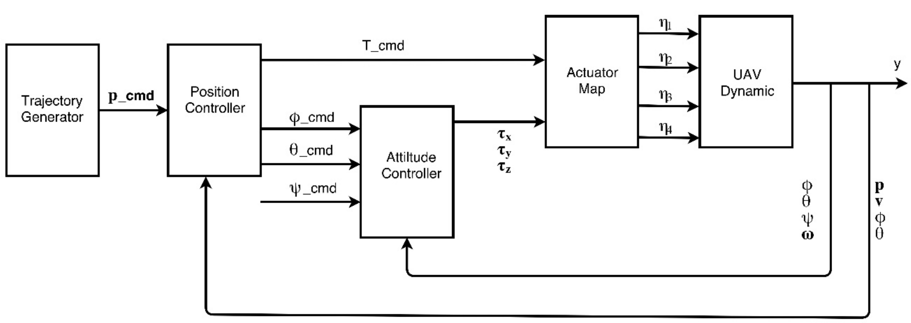

Fig. 6 The schematic diagram of the cascade control system.

B. Linear Steady-state Model

The dynamic model of the vehicle can be linearized around the hover state. We assumed that the motion of the vehicle consists of small deviation about a steady condition and approximates the sine function with its argument and the cosine function with unity. The products of deviation are also neglected. Because of the strong downstream flow, the pitching moment on the wing in hover state is significant. $M_{\mathrm{aero},y}$ were introduced into the model as unmeasured disturbances to eliminate the steady-state angular errors in y-axis (pitch motion). As long as the measured plant output deviates from its predicted trajectory, due to an unknown disturbance or a modeling error, the controller will asymptotically reject these sustained disturbances until the plant output returns to its desired trajectory, emulating an integral feedback controller [?]. In this study, the disturbance model was simplified to an integrator with dimensionless unity gain.

Following above assumptions, the linear dynamic and the kinematic model can be expressed as:

$$
\begin{aligned}
&\mathbf{v}_{\mathrm{I}} = \mathbf{v}_{\mathrm{B}} \\
&\omega_{\mathrm{I}} = \boldsymbol{\omega}_{\mathrm{B}} \\
&\dot{u} = -g\theta \\
&\dot{v} = g\phi \\
&\dot{w} = \ -F/m \\
&\dot{p} = M_x/I_x \\
&\dot{q} = (M + M_{\mathrm{aero},y})/I_y \\
&\dot{r} = M_z/I_z.
\end{aligned}
\tag{14}
$$

These linear equations can be written into state-space form as shown in Eq.(**??**), which is used to design the linear MPC controllers. The full state-space model was divided into two sub-models, the attitude model, and position model.

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\
\mathbf{y} &= \mathbf{Cx} + \mathbf{Du}
\end{aligned}
\tag{15}
$$

1) The attitude loop model
   The control system has the inner loop of attitude controller with state and input variables set up as $\mathbf{x_1} = [\phi \;\; \theta \;\; \psi \;\; p \;\; q \;\; r]^T$ and $\mathbf{u_1} = \begin{bmatrix} M_x & M_y & M_z \; M_{aero_y} \end{bmatrix}^T$. It's state space equation $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ was expanded and shown in Eq.(**??**). We assume all the states can be measured such that matric **C** and **D** are identity matrix and zero matrix respectively.

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
I_x^-1 & 0 & 0 & 0 \\
0 & I_y^-1 & 0 & I_y^-1 \\
0 & 0 & I_z^-1 & 0
\end{bmatrix}
\begin{bmatrix} M_x \\ M_y \\ M_z \\ M_{\mathrm{aero},y} \end{bmatrix}
\tag{16}
$$

2) The position loop model
   The outer loop is the position controller which has states and inputs as $\mathbf{x_2} = [x \;\; y \;\; z \;\; u \;\; v \;\; w \;\; \phi \;\; \theta]^T$ and $\mathbf{u_2} = [\phi_{cmd} \;\; \theta_{cmd} \;\; \psi_{cmd}]^T$. The $\dot{\phi}$ and $\dot{\theta}$ in $\dot{\mathbf{x}}_2$ were expressed by a set of first order transfer function [?]:

$$
\begin{aligned}
\dot{\phi} &= \tfrac{1}{\tau_\phi}(K_\phi \phi_{cmd} - \phi) \\
\dot{\theta} &= \tfrac{1}{\tau_\theta}(K_\theta \theta_{cmd} - \theta),
\end{aligned}
\tag{17}
$$

   where K_ϕ, K_θ and, τ_ϕ, τ_θ are gains and time constants of roll and pitch angle respectively. It's state space equation $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ was expanded and shown in Eq.(**??**). The matrix **C** is identity matrix and **D** is zero matrix.

$$
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -g \\
0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\tau_\phi^{-1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\tau_\theta^{-1}
\end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ u \\ v \\ w \\ \phi \\ \theta \end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & -1/m \\
K_\phi/\tau_\phi & 0 & 0 \\
0 & K_\theta/\tau_\theta & 0
\end{bmatrix}
\begin{bmatrix} \phi_{cmd} \\ \theta_{cmd} \\ T_{cmd} \end{bmatrix}
\tag{18}
$$

C. Objective Function

The aforementioned model will be employed into an objective function of the MPC controller that penalizes the output error as well as the control effort:

$$\mathbf{J}(\mathbf{z}_k) = \sum_{i=0}^{H-1} \{[\mathbf{e}_{\mathbf{y}}^T(k+i)\mathbf{Q}\mathbf{e}_{\mathbf{y}}(k+i)] + [\mathbf{e}_{\mathbf{u}}^T(k+i)\mathbf{R}_{\mathbf{u}}\mathbf{e}_{\mathbf{u}}(k+i)] + [\Delta\mathbf{u}^T(k+i)\mathbf{R}_{\Delta\mathbf{u}}\Delta\mathbf{u}(k+i)]\}, \tag{19}$$

where the operation time T consists of k control intervals. At the current control interval k, the cost function will consider a prediction horizon H, which consists of i steps. $Q, R_u$and $R_\Delta u$ are positive-semi-definite weight matrices, e_y, e_u and Δu represent error on output, error on input and the change of input, respectively and can be express as:

$$\begin{aligned} \mathbf{e}_{\mathbf{y}}\ (k+i) &= \mathbf{S}_{\mathbf{y}}^{-1}\ [\mathbf{r}(k+i+1|k) - \mathbf{y}(k+i+1|k)] \\ \mathbf{e}_{\mathbf{u}}\ (k+i) &= \mathbf{S}_{\mathbf{u}}^{-1}\ [\mathbf{u}_{\mathbf{target}}\ (k+i|k) - \mathbf{u}(k+i|k)] \\ \mathbf{\Delta u}(k+i) &= \mathbf{S}_{\mathbf{u}}^{-1}\ [\mathbf{u}(k+i|k) - \mathbf{u}(k+i-1|k)], \end{aligned} \tag{20}$$

where, $\mathbf{u}$ is the plant output reference value of the i-th prediction horizon step at the k-th control interval. $\mathbf{S}_y$ and $\mathbf{S}_u$ are diagonal matrix of scale factor of plant output and input in engineering units. The MPC constraints can be expressed as follow:

$$\begin{aligned} &\frac{y_{j,min}(i)}{s_j^y} \le \frac{y_j(k+i|k)}{s_j^y} \le \frac{y_{j,max}(i)}{s_j^y}, i = 1:H, j = 1:N_y \\ &\frac{u_{j,min}(i)}{s_j^u} \le \frac{u_j(k+i-1|k)}{s_j^u} \le \frac{u_{j,max}(i)}{s_j^u}, i = 1:H, j = 1:N_u \\ &\frac{\Delta u_{j,min}(i)}{s_j^u} \le \frac{\Delta u_j(k+i-1|k)}{s_j^u} \le \frac{\Delta u_{j,max}(i)}{s_j^u}, i = 1:H, j = 1:N_u \end{aligned} \tag{21}$$

where s_j^y and s_j^u are scale factors for the j-th output and input. N_y and N_u are the number of plant output and input. $y_{j,min}\ (i)$ and $y_{j,max}\ (i)$ are lower and upper bounds for the j-th plant output at the i-th prediction horizon step. Where $u_{j,min}\ (i)$, $u_{j,max}\ (i)$, $\Delta u_{j,min}\ (i)$and $\Delta u_{j,max}\ (i)$ are having similar meaning but of MV increment.

$\mathbf{z}_k$ is the Quadric Programming (QP) decision given by:

$$\mathbf{z}_k^T = \left[\mathbf{u}\,(k|\,k)^T \quad \mathbf{u}\,(k+1|\,k)^T \ldots\ \mathbf{u}\,(k+H-1|\,k)^T\right], \tag{22}$$

in which only the $\mathbf{u}(k|k)^T$ will be implemented at each time step.

D. Preview of Trajectory

Preview of MPC is an ability that the planned trajectory was fed into the controller in advance, such that the controller knew the trajectory ahead and it was tested by the position controller. According to the objective function (Eq.(**??**)), which is a summation of decisions of the whole prediction horizon. At the very beginning of an operation, a number of steps (same as the prediction horizon) of the trajectory will be loaded in advance to the objective function and will be considered all. Instead of the reference at just that time step. In the next time step, a new point of trajectory will be loaded and then considered. As the controller gradually sees the changes of the reference trajectory within the prediction horizon, the command can be given in beforehand of the trajectory differ. As a result, the controller will be able to start to act in seconds ahead, instead of responding after the error was measured.

E. Actuator Mapping

The control commands thrust T and moments $\mathbf{M}$ from both MPC controllers were fed into an actuator map that converts the received command signals into desired throttle of each motor. By applying the momentum theory, the steady-state thrust generated by a hovering rotor can be modeled as:

$$T = c_T\omega^2, \tag{23}$$

and the reaction moment can be modeled as:

$$M = c_M\omega^2 \tag{24}$$

c_T and c_M were obtained from an linear fitting of ω^2 to thrust T and to moment M by the previous experiment data. According to the 'plus' configuration of the vehicle, the linear relationship between the angular velocity and the desired thrust and moments can be modeled as:

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -d_1 c_T & d_1 c_T & 0 & 0 \\ 0 & 0 & d_2 c_T & -d_2 c \\ c_M & c_M & -c_M & -c_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \tag{25}$$

where d_1 and d_2 are the distance from the motors to the center of mass of the vehicle in x and y-axis. The required rotation speed of each motor can then be calculated by inverting Eq.(**??**) and obtain the corresponding throttle by a third order polynomial function fitted from the experimental data, which represents the relationship between ω^2 to throttle.

V. HIL Simulation Environment

A. HIL Simulation Structure

MPC controller usually has high computation cost since it relies on the online solution of the optimization problem. However, the onboard flight control unit usually has limited computation resource. The HIL simulation environment can help to test the real-time performance of the proposed flight controller and provide a reference for further flight experiments. In this paper, the HIL simulation environment was built based on MATLAB and Simulink Student Suite. The system is composed of two main parts, one is the real-time tail-sitter dynamic model running on a desktop PC and the other is the MPC controller running on the flight computer Odroid XU4 [**?**]. The schematic of the simulation system is shown in Fig. 7. The desktop PC has an Intel Core i7-4790 CPU @ 3.60 GHz while the flight computer has a Samsung Exynos5422 CPU @ 2GHz.

The UAV dynamic models described in Section II was built in Simulink. A Simulink tool named Real-Time Pacer for Simulink [**?**] was used to force the simulation run in wall clock time. The MPC controllers were firstly built in Simulink with Model Predictive Control Toolbox and then generated to a standalone ROS node by Simulink Coder [**?**]. This process transformed the Simulink blocks and MPC controllers into C/C++ code which could be built and executed on Odroid XU4 with high efficiency.

The model and controller are communicated by ROS (robot operating system) publish and subscribe manner [**?**]. This communication method is suitable for communication between multiple subsystems and is widely used in robotic systems. The Simulink model publishes the states of UAV including position, velocity, orientation and angular velocity to the MPC controller running on Odroid XU4. At the same time, the generated MPC ROS node subscribes the states of the UAV and computes the control commands to the Simulink dynamic model.

The whole system is a hardware-in-loop structure with flight controller running on an independent onboard computer. This environment can test the real-time computation load of the MPC controller and the effect of communication delay. After the HIL simulation, the controller hardware and software can be integrated on the UAV prototype which facilitates the further flight experiments.

B. Controller Parameters

Although the MPC controller requires less turning effort and the PID controller, some parameters still need to be adjusted to achieve reliable and robust performance. The sampling time is one of the main parameters to choose. As T_s decreases, rejection of unknown disturbance usually improves, however the computational effort will increase dramatically. Thus, the optimal choice is a balance of performance and computational load. T_s was set as 0.04 second in the following simulation, which means the control algorithm runs at 25 Hz. The prediction horizon, H is another important consideration. The prediction horizon cannot be too long since the plant unstable mode would dominate, and the controller memory requirements increase as well as the QP solution time increase. On the other hand, H cannot be too short because constraint violations would be unforeseen. In most of the time, H should vary inversely with T_s. The control horizon N must falls between 1-H. Regardless of the choice of N, when the controller operates, the first optimized manipulated variable (MV) move of the horizon is used and any others are discarded. Large N means more variables to be considerate in the QP solver at each control interval, which promotes slower computations but better performance in response.

Adjusting the weight of variables is critical to the performance. For output variable weight $\mathbf{Q}$, a higher weight means higher priority with rough guidelines of 0.05-20. For MV rate weight $\mathbf{R_{\Delta u}}$, it penalizes large MV changes in the optimization cost function. For example, simultaneously reduces the MV rate weight and increases the output variable

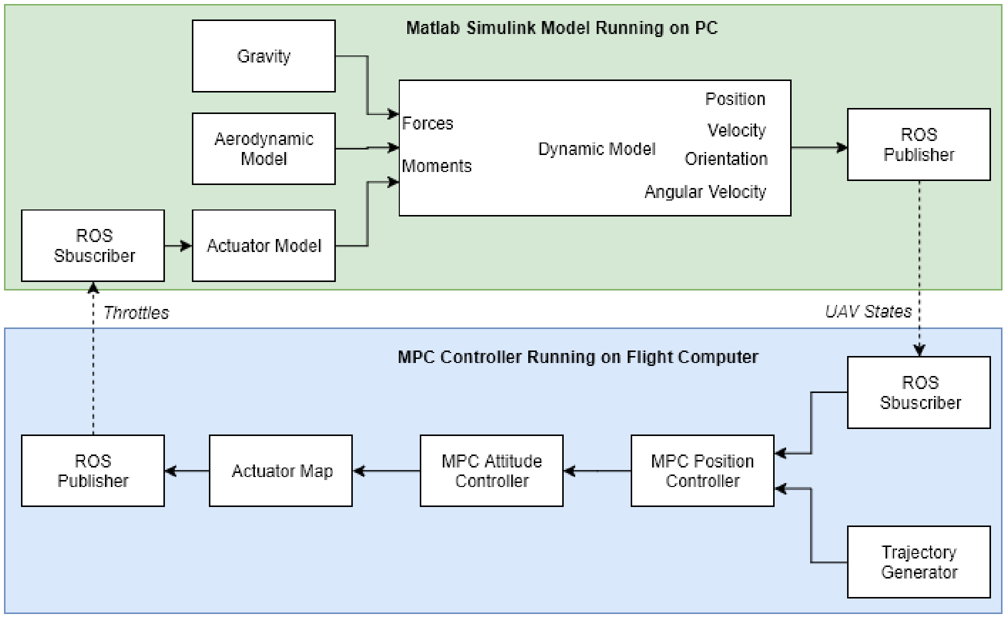

Fig. 7 The HIL simulation environment structure.

Table 1 Parameter setting of controllers

Parameters	Attitude Controller	Position Controller
$\mathbf{T}_s$	0.04	0.04
H	40	40
N	4	4
$\lvert\mathbf{x}\rvert$ constrain	$[\pi/6, \pi/6, \inf, 2, 2, 2]^T$	$[\inf, \inf, \inf, 20, 20, 20, \pi/6, \pi/6]^T$
$\lvert\mathbf{u}\rvert$ constrain	$[2.8, 2, 0.26]^T$	$[\pi/6, \pi/6, 10]^T$
$\mathbf{Q}$	$\mathbf{diag}\left\{\sqrt{5}, \sqrt{2}, \sqrt{10}, 0, 0, 0\right\}$	$\mathbf{diag}\left\{\sqrt{2}, \sqrt{2}, \sqrt{10}, 0, 0, 0, \sqrt{5}, \sqrt{5}\right\}$
$\mathbf{R_u}$	$\mathbf{diag}\,\{0, 0, 0\}$	$\mathbf{diag}\,\{0, 0, 0\}$
$\mathbf{R_\Delta u}$	$\mathbf{diag}\,\{0.2, 0.5, 0.2\}$	$\mathbf{diag}\,\{0.2, 0.8, 0.2\}$

(OV) weight will produce a more robust controller. Since the controller will limit the move and put higher priority on feedback at each control interval. MV weight were all set as zero in our simulation since there is no specific values to follow for the input as long as they stay in the constraints. MV increment weight $\mathbf{R_\Delta u}$ was set as 0.2-0.8 of each because a small value of weight can slightly penalize the move at each step, such that reduce the sensitivity of the system and increase robustness. The cost of change in pitch is slight higher than other since the aerodynamic moment in this direction is easily influenced by the change of velocity. The high frequency change of control output would cause vibration of angle. The key parameters of the two MPC controllers are shown in Table 1.

VI. Results and Discussions

Two typical reference paths were generated from trajectory generator and given to the MPC controller in three simulation scenarios to test the tracking and disturbance rejection performance. The initial condition of the UAV is in hover state with $\mathbf{x}_0 = \mathbf{0}$. The reference commands in X_e, Y_e and Z_e directions in different time was prepared in the trajectory generator and given to the system for tracking. The command given to the controller is plotted in dash lines and the solid lines indicate the response. The left column shows the command and response of the inner loop attitude controller and the right column shows that of the outer loop position controller.

A. Step Path Tracking without Trajectory Preview

Fig. 8 shows the simulation results of normal step path tracking. The step commands in all three directions were given to the UAV in 10s, 20, and 30s. At time $t = 0\ s$, $[X_{cmd}, Y_{cmd}, Z_{cmd}] = [0, 0, -3]$ was given, which commands the vehicle to climb up to 3 m in high and keep the altitude. At time of the $t = 10\ s$ and $t = 20\ s$, the other step position references in X_e and Y_e direction respectively were given. It is shown that vehicle reaches the desire position in about 3 s with small overshoot. The commanded attitude was consistently constrained by MPC controller within the region, which is -30 to 30 degree. Pitch is the most sensitive direction for this kind of tail-sitter UAV since the pitch moment generated by the main wing will variant with the change of attitude. Then the deviation about 5 degrees of pitch angle can be found whenever large movement.

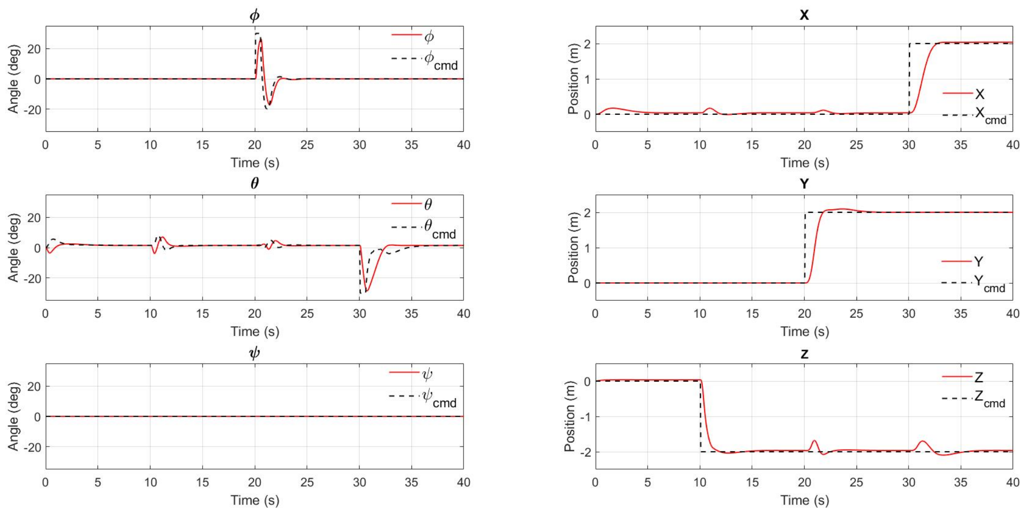

Fig. 8 Step path tracking results without preview.

B. Step Path Tracking with Trajectory Preview

In this simulation, the initial condition and path commands are similar to previous simulation. The difference is the entire trajectory was loaded to the controller before the simulation started. The controller then accounted the change of reference in ahead with the moving of the prediction horizon, shown in Fig. 9. The preview ability enables the vehicle to respond before the commanded time due to the included lookahead reference in the optimize function.

C. Spiral Path Tracking with Wind Disturbance

In the spiral path tracking simulation, the positions of the UAV in both Xe and Ye axis were controlled to follow two sinusoidal curves in different phase while the height of Z_e axis was kept increasing. A horizontal wind of 3 m/s was given to the vehicle from $t = 20\ s$ to $t = 30\ s$ from the $+X_e$ direction. The results of position and altitude response are shown in Fig. 10. It is shown that the positions in three axes can follow the commanded trajectory very well with preview. When the wind hit the vehicle at $t = 20\ s$, the vehicle pitched down with the influence brought by the change of AoA and airspeed. The attitude controller commands vehicle pitching up to overcome the effect of wind disturbance. When the wind stopped at $t = 20\ s$, we can find the vehicle recovered in about 2 second.

VII. Conclusion

In this paper, cascade MPC controllers were developed for a tail-sitter VTOL UAV in an HIL simulation environment for path tracking. The UAV dynamic model was built which included the aerodynamics of the main wing as well as the propellers. The MPC controllers were first turned in PC and then generated to ROS node which can run on a micro flight computer to verify the controller performance in an HIL environment. Two typical path tracking simulations were carried out in three scenarios. The simulation results show that the MPC controller has good tracking and disturbance

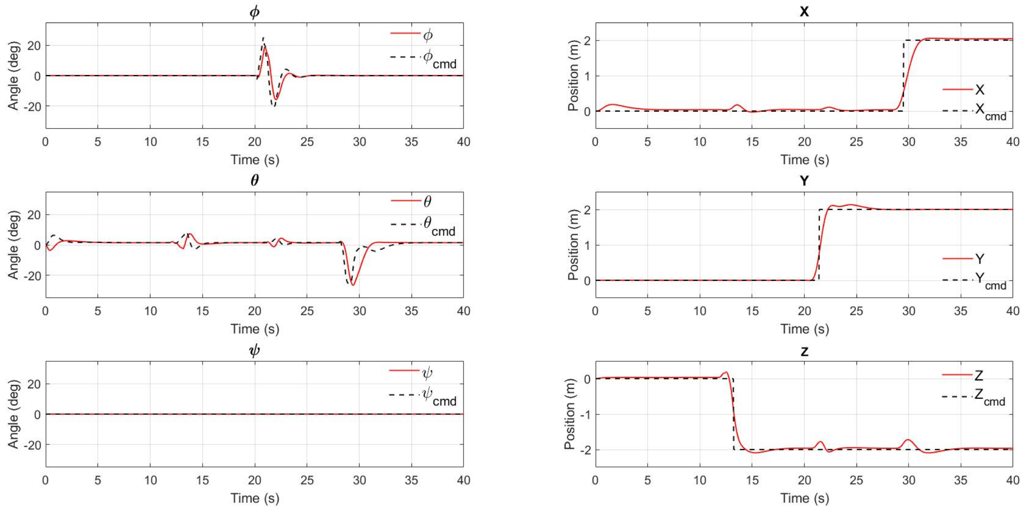

Fig. 9 Step path tracking results with preview.

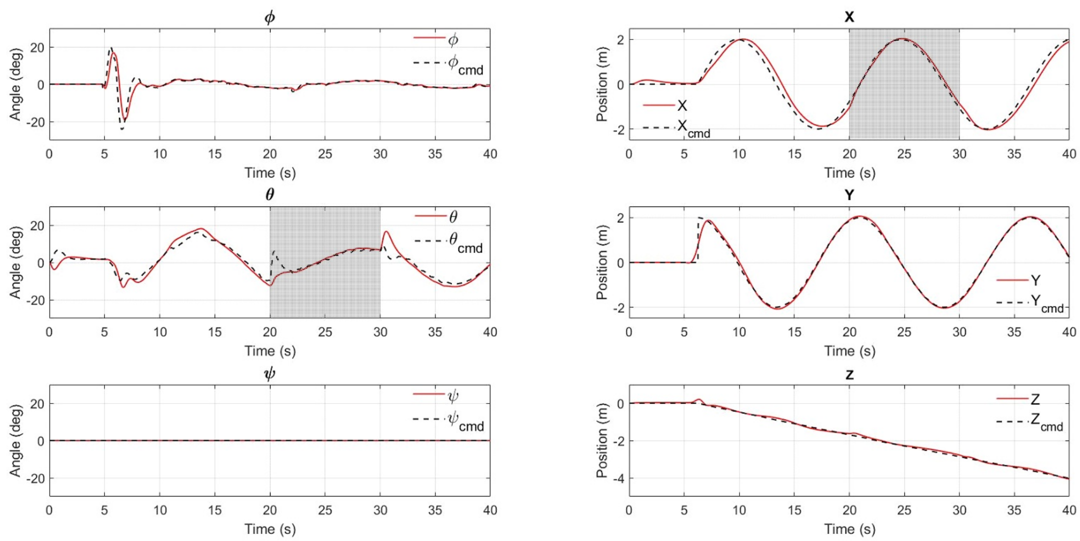

Fig. 10 Inclining spiral path tracking with trajectory preview on and wind disturbance from 20 s - 30 s.

rejection capability. In the future, the controller system will be transplanted into the indoor localization system to carry out experiment study.

Acknowledgments

This work is supported by Innovation and Technology Commission, Hong Kong under Contract No. ITS/334/15FP.

References

[] Nikola, T., "Apparatus for aerial transportation," , Jan. 3 1928. US Patent 1,655,114.

[] Shkarayev, S., Moschetta, J.-M., and Bataille, B., “Aerodynamic Design of Micro Air Vehicles for Vertical Flight,” *Journal of Aircraft*, Vol. 45, No. 5, 2008, pp. 1715–1724. doi:10.2514/1.35573.

[] Wang, J., Song, B., Wang, L., and Tang, W., “L1 Adaptive Dynamic Inversion Controller for an X-wing Tail-sitter MAV in Hover Flight,” *APISAT2014*, Vol. 99, Elsevier BV, 2015, pp. 969–974. doi:10.1016/j.proeng.2014.12.629.

[] Garcia, O., Sanchez, A., Escareño, J., and Lozano, R., “Tail-sitter UAV having one tilting rotor: Modeling, Control and Real-Time Experiments,” *IFAC Proceedings Volumes*, Vol. 41, No. 2, 2008, pp. 809 – 814. doi:https://doi.org/10.3182/20080706-5-KR-1001.00139, URL `http://www.sciencedirect.com/science/article/pii/S1474667016390498`, 17th IFAC World Congress.

[] Verling, S., Weibel, B., Boosfeld, M., Alexis, K., Burri, M., and Siegwart, R., “Full Attitude Control of a VTOL tailsitter UAV,” *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 3006–3012. doi:10.1109/icra.2016.7487466, URL `http://ieeexplore.ieee.org/ielx7/7478842/7487087/07487466.pdf?tp=&arnumber=7487466&isnumber=7487087`.

[] Sun, J., Li, B., Shen, L., Chen, C.-K., and Wen, C.-Y., “Dynamic Modeling and Hardware-In-Loop Simulation for a Tail-Sitter Unmanned Aerial Vehicle in Hovering Flight,” *AIAA Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-0811.

[] Ritz, R., and D’Andrea, R., “A Global Strategy for Tailsitter Hover Control,” *Springer Proceedings in Advanced Robotics*, Springer International Publishing, 2017, pp. 21–37. doi:10.1007/978-3-319-51532-8_2.

[] Zhang, F., Lyu, X., Wang, Y., Gu, H., and Li, Z., “Modeling and Flight Control Simulation of a Quadrotor Tailsitter VTOL UAV,” *AIAA Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-1561.

[] Yeo, D., Hrishikeshavan, V., and Chopra, I., “Gust Detection and Mitigation on a Quad Rotor Biplane,” *AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, 2016. doi:10.2514/6.2016-1531.

[] Theys, B., Vos, G. D., and Schutter, J. D., “A control approach for transitioning VTOL UAVs with continuously varying transition angle and controlled by differential thrust,” *Proc. Int. Conf. Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 118–125. doi:10.1109/ICUAS.2016.7502519.

[] Oosedo, A., Abiko, S., Konno, A., Koizumi, T., Furui, T., and Uchiyama, M., “Development of a quad rotor tail-sitter VTOL UAV without control surfaces and experimental verification,” *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 317–322. doi:10.1109/icra.2013.6630594.

[] Borrelli, F., Bemporad, A., and Morari, M., *Predictive Control for Linear and Hybrid Systems*, CAMBRIDGE UNIVERSITY PRESS, 2017.

[] Mayne, D. Q., “Model predictive control: Recent developments and future promise,” *Automatica*, Vol. 50, No. 12, 2014, pp. 2967–2986. doi:10.1016/j.automatica.2014.10.128.

[] Bemporad, A., Pascucci, C., and Rocchi, C., “Hierarchical and Hybrid Model Predictive Control of Quadcopter Air Vehicles,” *IFAC Proceedings Volumes*, Vol. 42, No. 17, 2009, pp. 14–19. doi:10.3182/20090916-3-es-3003.00004.

[] Chen, X., and Wang, L., “Cascaded model predictive control of a quadrotor UAV,” *2013 Australian Control Conference*, IEEE, 2013, pp. 354–359. doi:10.1109/aucc.2013.6697298.

[] Alexis, K., Tzes, A., and Nikolakopoulos, G., “Model predictive quadrotor control: attitude, altitude and position experimental studies,” *IET Control Theory & Applications*, Vol. 6, No. 12, 2012, pp. 1812–1827. doi:10.1049/iet-cta.2011.0348.

[] Bangura, M., and Mahony, R., “Real-time Model Predictive Control for Quadrotors,” *IFAC Proceedings Volumes*, Vol. 47, No. 3, 2014, pp. 11773–11780. doi:10.3182/20140824-6-za-1003.00203.

[] Kamel, M., Alexis, K., Achtelik, M., and Siegwart, R., “Fast nonlinear model predictive control for multicopter attitude tracking on SO(3),” *2015 IEEE Conference on Control Applications (CCA)*, IEEE, 2015, pp. 1398–1404. doi:10.1109/cca.2015.7320769.

[] Xu, P., “Aerodynamics of VTOL UAV,” mathesis, The Hong Kong Polytechnic University, Jul. 2017.

[] ATI, “F/T Sensor: Mini40,” `http://www.ati-ia.com/products/ft/ft_models.aspx?id=Mini40`, Nov. 2017.

[] MathWorks, “Simulink. MPC Modeling,” `http://www.ros.org/`, Nov. 2017.

[] Kamel, M., Burri, M., and Siegwart, R., “Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles,” *arXiv:1611.09240*, 2016. URL https://arxiv.org/abs/1611.09240.

[] Hardkernel, “ODROID-XU4,” http://www.hardkernel.com/main/products/prdt_info.php, Nov. 2017.

[] Vallabha, G., “Real-Time Pacer for Simulink,” https://cn.mathworks.com/matlabcentral/fileexchange/29107-real-time-pacer-for-simulink, Nov. 2017.

[] MathWorks, “Generate a Standalone ROS Node from Simulink®,” https://cn.mathworks.com/help/robotics/examples/generate-a-standalone-ros-node-from-simulink.html, Nov. 2017.

[] “ROS,” http://www.ros.org/, Nov. 2017.