

Position Control of a Tail-sitter UAV Using Successive Linearization Based Model Predictive Control

Weifeng Zhou¹, Boyang Li¹, Jingxuan Sun¹, Chih-Yung Wen¹, Chih-Keng Chen^{2*}

¹ Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong

² Department of Vehicle Engineering, National Taipei University of Technology, Taipei, Taiwan

* Fax: +886-935658801, Email address: ckchen@ntut.edu.tw

Keywords: Tail-sitter, VTOL, UAV, MPC, Aerodynamic, Disturbance

Abstract: A successive linearization based model predictive control (SLMPC) method is proposed to control a vertical take-off and landing (VTOL) tail-sitter unmanned aerial vehicle (UAV) in hovering flight. The dynamic model of the vehicle is derived, including a low-fidelity aerodynamic model and a propulsion system model. The position controller is developed by a state-space prediction model augmented with estimated disturbance and feedback integration terms. The time-varying weight in the objective function is included and the velocity of vehicle is considered as reference to improve the performance. The system is first tested in a software-in-loop environment followed by the real-time indoor flight tests. The results demonstrate the vehicle can precisely follow a trajectory and stably hold position under unsteady wind disturbance.

1. Introduction

Unmanned aerial vehicles (UAVs) play many roles in both consumer and commercial markets. Compared with manned aircrafts, UAVs are usually much more cost-effective, smaller, able to get closer to dangerous areas, and therefore, widely applied in search and rescue (B. Li, Jiang, Sun, Cai, & Wen, 2016; J. Sun, Li, Jiang, & Wen, 2016), emergency delivery (Lin, Shah, Mauntel, & Shah, 2018), infrastructure inspection (Deng, Wang, Huang, Tan, & Liu) etc. In addition to the fixed wing aircraft and helicopters, vertical take-off and landing (VTOL) UAVs have been developed to better fit a large variety of roles, especially in an urban area. VTOL UAVs combines the advantages of a multi-rotor and a fixed-wing aircraft. They can conduct missions without a runway or a catapult and undertake long-distance flights and reach high cruising speeds. These characteristics make them suitable for cities such as Hong Kong, which has a high-density urban population and many inhabited offshore islands. Among all kinds of VTOL UAVs, the tail-sitter vehicle is much less mechanically complicated, so it is lighter and has less risk of failure than other configurations, such as the tilt-rotor type (An, Zhang, Zhang, & Li, 2016), the tilt-wing type (Gregory, Ackerman, Snyder, & Rothhaar, 2015) and the hybrid type (Ozdemir, et al., 2013). Novel configurations for a tail-sitter vehicle with unique practical functionalities have been the subject of many studies (Pflimlin, Binetti, Souères, Hamel, & Trouchet, 2010; Sinha, Esden-Tempski, Forrette, Gibboney, & Horn, 2012; Wang, Chen, & Yuan, 2015).

Stable and robust hovering control is of higher priority than other flight stages, as hovering is the basic element of an entire flight. In the recent researches of (Hochstenbach, Notteboom, Theys, & De Schutter, 2015; Oosedo, et al., 2013; Oosedo, et al., 2012), the tail-sitter vehicle was controlled by a PID method. It is difficult for a traditional PID method to control a tail-sitter vertical with large nonlinearity even though a large amount of effort has been put into it to improving its capability. Because, due to the significant wing area, the tail-sitter vehicles are extremely sensitive to the windy environment during hover condition, yielding extra challenges for the control system. A control method that integrates both a vehicle model and an environment model will be an optimal choice.

The model predictive control (MPC) method is a multi-input-multi-output (MIMO) control scheme that uses a receding horizon strategy for both linear and nonlinear systems (Cannon, 2004; Deihl Moritz, 2009). As introduced in (Eduardo F. Camacho, 2007), the MPC method can optimise the current timeslot while keeping the future timeslot in the account. Constraints and weights are imposed on each variable during the optimisation process depending on physical characteristics. Very few control methodologies can simultaneously handle constraints and the optimisation problem in a systematic manner (Eren, et al., 2017). The prediction and optimization process is repeated at every sampling time to ensure performance. However, this generates a high computational burden. In (Zheng, 1997) a novel MPC algorithm was proposed to significantly lower the computational demand as the optimization problem depends only on the number of manipulated variable rather than the horizon. A suboptimal MPC scheme using the Dantzig-Wolfe algorithm was introduced in (Standardi, Poulsen, Jørgensen, & Sokoler, 2013) and successfully reduced the computation time. The study in (Peyrl, Zanarini, Besselmann, Liu, & Boéchat, 2014) aimed to shorten the execution-time by applying a parallel computing architecture to solve linear MPC problems. A forward error analysis was adopted by (Suardi, Longo, Kerrigan, & Constantinides, 2016) to compute the errors within bounds to improve the performance of a low precision computation MPC which naturally has low power consumption. As discussed in (Mayne, 2014), MPC is a relatively mature technique for linear and slow systems, such as processes in industry or chemical plants. With current improvements in overall computational capability, it is also used for nonlinear processes or processes with frequent changes in operating conditions, such as aerospace and robotic systems

A variety of strategies have been developed to improve the MPC control. A control strategy using a piecewise affine (PWA) dynamic modelling approach with a switching MPC method was proposed in (K. Alexis, Tzes, & Nikolakopoulos, 2012). It was successfully used to control a quadrotor UAV over a large flight envelope, which included aggressive attitudes and trajectory tracking in gusting wind. An efficient MPC scheme was developed in (Abdolhosseini, Zhang, & Rabbath, 2013). It uses a model reduction technique to downsize the state-space model to lower the computational requirement. An adaptive MPC scheme was used to control a traditional quadrotor vehicle in (Chikasha & Dube, 2017) and a single degree of freedom flapping-wing UAV in (Zhu & Zuo, 2017). The linear MPC method has successfully controlled the quadrotor UAV in (Bangura & Mahony, 2014; Zhao, Wang, Zhang, & Shen, 2017), it does not necessarily mean that it can control a tail-sitter vehicle, which has the strong nonlinear aerodynamic effect of the wings.

Disturbance modelling and rejection ability are important to a tail-sitter vehicle control as it usually faces prevailing wind or gust wind conditions. A linear robust explicit MPC controller was developed in (Kostas Alexis, Papachristos, Siegart, & Tzes, 2016) as a position control for a tri-rotor UAV to ensure minimum deviation even for the worst-case disturbance. With strictly defined boundaries for variables and the use of a prediction term, robustness and minimum peak performance were improved. A disturbance observer was designed in (Dong, Gu, Zhu, & Ding, 2014) to conduct agile trajectory following for a quadrotor UAV. The model for the rotor input delay was included while the external disturbance and model mismatch terms were added to the dynamic model. In (Liu, Chen, & Andrews, 2012), a nonlinear disturbance observer estimates the external force/torque from the wind turbulences and it is cooperate with a nonlinear MPC controller for small-scale helicopters. An active disturbance rejection control is used in (Ma, Xia, Li, & Chang, 2016) coupled with an extended state observer. The external and internal disturbance caused by unmodelled dynamic and parameter uncertainties is lumped as an external state and compensated in the control input. These technics can be applied to the controller development of a tail-sitter vehicle as its wing span will bring predictable and non-predictable disturbance to the system.

Motivated by the above discussion, this paper proposed the hovering position control of a tail-sitter VTOL UAV under unsteady wind condition by applying a successive linearization based MPC (SLMPC) method cooperated with disturbance models. The SLMPC allows regular update of parameters and nominal states in the prediction model at every time-step to cover a much larger control envelope than a traditional linear MPC controller. Compared to other MPC schemes discussed above, the major contribution of this study is the augmentation of an estimated disturbance term and a feedback integration term to the model of

the successive linearization based MPC. The aerodynamic force and moment are treated as disturbance that can be estimated and modelled. Model uncertainty and unknown disturbance are compensated by a feedback policy to achieve offset-free control.

The remainder of this paper is organised as follows. Section 2 describes the UAV configuration, the modelling of the propulsion system and the aerodynamic effects. The successive linearization based MPC controller and the optimisation problem is formulated in Section 3. Section 4 presents the software-in-loop (SIL) simulation results and section 5 discusses the experimental setup and results. Section 6 provides conclusions.

2. System Configuration and Modelling

In this section, the configuration and coordinate system are described, and the onboard avionics and propulsion system are detailed. The coordinate system and motion equations are defined. The characteristics of the propulsion system are then modelled using a thrust experiment and the aerodynamic effects are modelled theoretically.

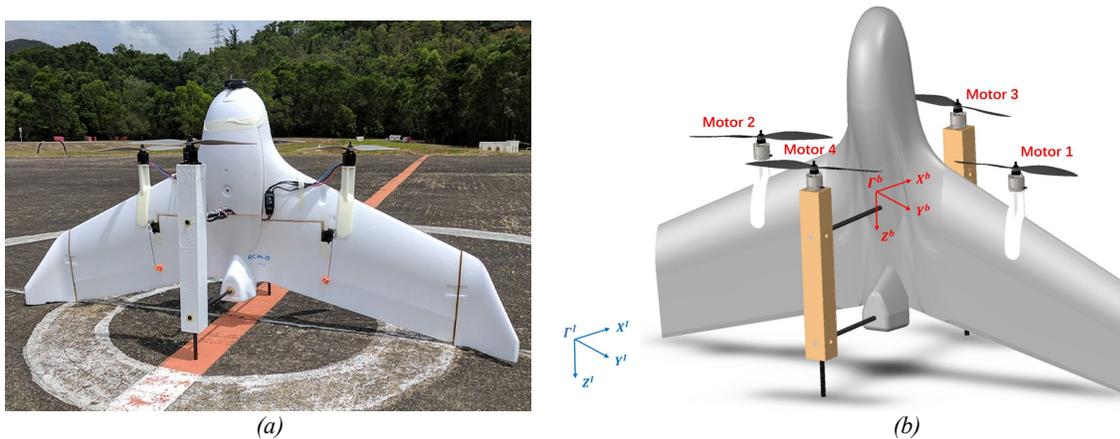


Fig. 1. (a) PolyU Plus tail-sitter VTOL vehicle, 1. (b) Definitions of the coordinate systems.

2.1. System Configuration and Coordinate System

The quadrotor tail-sitter UAV shown in Fig. 1a is modified from the commercial flying wing UAV platform ‘Skywalker X-5’, with its rear pusher propeller removed and replaced by four puller propellers. This UAV is different from the quadrotor type, as it has wings and therefore different aerodynamic effects. It uses four motors and propellers as the actuators, instead of the rudder, elevators, and ailerons used in fixed-wing aircraft. The ‘plus’ shape is chosen because this configuration better utilizes the wing. The propeller wash passes through a major part of the wing and increases the airspeed on this wing segment. However, it also results in an unwanted aerodynamic force and moment during the hovering phase, which must be corrected when designing the flight controller. The final prototype has a wingspan of 1.1m, with a mean aerodynamic chord (MAC) of 0.4m and an operating weight of 2kg.

To describe the dynamic of UAVs, two sets of reference systems are required, as shown in Fig. 1b. The fixed inertial coordinate system ($\Gamma^I: X^I, Y^I, Z^I$) points to the north, east and downward directions. The mobile body coordinate system ($\Gamma^b: X^b, Y^b, Z^b$), which is located at the vehicle’s centre of mass, has an x-axis, a y-axis and a z-axis pointing to motor 3, motor 1 and the tail of the vehicle in fig. 1(b), respectively.

2.2. Dynamic Modelling

The dynamic model is described as follows:

$$\dot{\mathbf{p}} = \mathbf{v} \quad (1a)$$

$$m\dot{\mathbf{v}} = \mathbf{R}_b^{\mathbf{I}} \mathbf{F}_b \quad (1b)$$

$$\boldsymbol{\Omega} = \mathbf{T}_b^{\mathbf{I}} \boldsymbol{\omega} \quad (1c)$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \mathbf{M}_b \quad (1d)$$

where $\mathbf{p} = [x \ y \ z]^T \in \mathbb{R}^3$ and $\mathbf{v} = [\dot{x} \ \dot{y} \ \dot{z}]^T \in \mathbb{R}^3$ are the position and velocity in Γ^I , respectively; $\mathbf{R}_b^{\mathbf{I}} \in \text{SO}(3)$ is the rotation matrix that transfers Γ^b into Γ^I ; $\boldsymbol{\Omega} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \in \mathbb{R}^3$ is the time derivative of the Euler angles (roll, pitch and yaw); $\boldsymbol{\omega} = [p \ q \ r]^T \in \mathbb{R}^3$ is the angular velocity in Γ^b ; $\mathbf{T}_b^{\mathbf{I}}$ is the non-singular matrix that relates the angular velocities to the rates of the Euler angles; \mathbf{I} is the inertial matrix; and m is the mass of the vehicle.

\mathbf{F}_b and \mathbf{M}_b are the force and moment in Γ^b , respectively, and are expressed as follows:

$$\mathbf{F}_b = \mathbf{F}_{aero} + \mathbf{F}_P + \mathbf{R}_I^b \mathbf{F}_g + \mathbf{F}_d \quad (2a)$$

$$\mathbf{M}_b = \mathbf{M}_{aero} + \mathbf{M}_P + \mathbf{M}_d \quad (2b)$$

where \mathbf{F}_{aero} and \mathbf{M}_{aero} are the aerodynamic force and moment, respectively; \mathbf{F}_g is the gravity force in Γ^I ; \mathbf{F}_P is the thrust on the negative z-direction; \mathbf{M}_P is the moment that is created by the propulsion system; The disturbance and model error are described by \mathbf{F}_d and \mathbf{M}_d .

2.3. Propulsion system modelling

The nonlinear relationship between the throttle commands and the force and moment of the propulsion system is determined by a set of experiments. During hovering, the wind speed is assumed to be zero, such that there is no coming flow. The motor and propeller are mounted on an ATI Mini40 6-DOF Force/Torque sensor and tested. The generated thrust and moment for the motor-propeller are recorded for 5 seconds at every 10% throttle increment. During the experiment, a number of propeller and motor pairs are tested. Depending on the requirement of the vehicle, the combination of a Sunnysky x2212-980kv brushless motor, 30A ESC and APC1047 propellers is selected for the propulsion system. This creates more than 3N of thrust at 50% throttle and provides a maximum thrust/weight ratio of approximately 2.7.

A fourth-order polynomial function is used to fit the non-linear relationship. The modelling process was described in detail in a previous work by the authors (Boyang Li, et al., 2018b). The thrust and moment created by the propulsion system can be calculated as follows:

$$\mathbf{F}_P = \begin{bmatrix} 0 \\ 0 \\ -(f_{T1} + f_{T2} + f_{T3} + f_{T4}) \end{bmatrix} \quad (3)$$

$$\mathbf{M}_P = \begin{bmatrix} (f_{T2} - f_{T1})l_y \\ (f_{T3} - f_{T4})l_x \\ M_1 + M_2 - M_3 - M_4 \end{bmatrix}$$

where f_{Ti} and M_i , $i \in [1,4]$, are the thrust and moment from each propeller, respectively, and l_y and l_x are the moment arms along the y-axis and x-axis, respectively.

2.4. Aerodynamic Modelling

For the current UAV configuration, a large portion of the wing of the vehicle is immersed in the induced flow generated by the propellers shown in segments 2 and 4 of Fig. 2. The speed of the airflow is considered uniform over the wing and it is assumed that there is no cross coupling between the left and right wings. The side force Y which yields the aerodynamic effect on the y -axis of the vehicle is assumed zero, as its magnitude is much smaller than the lift force L and drag force D .

Accordingly, the wing is separated into five segments, based on a component breakdown approach (Jingxuan Sun, Li, Shen, Chen, & Wen, 2017), to distinguish the segments with and without induced flow. The MAC and the position of the aerodynamic center (AC) of the entire wing are calculated first. Using Bernoulli's theory and the momentum theory of flow (Shkarayev, Moschetta, & Bataille, 2008), the radius of the flow tube (r) created by the propeller is calculated. Consequently, the widths of the other segments are easily obtained. The MAC and AC for each segment are then calculated.

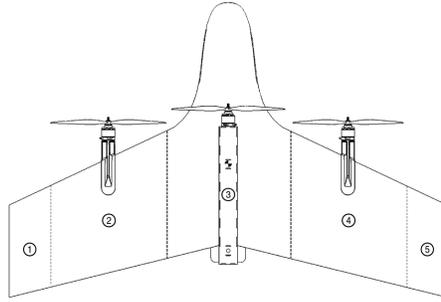


Fig. 2. Wing segments of the PolyU Plus tail-sitter vehicle.

To estimate the specific lift, drag and moment for the AC of each segment at different angles of attack and airspeeds, an aerodynamic database (Penghui, July, 2017) for this airfoil, including the lift coefficient (C_L), the drag coefficient (C_D) and the moment coefficient (C_M) at α from $-180^\circ \sim 180^\circ$ and the airspeed from $0 \sim 30$ m/s, is developed by conducting a series of wind tunnel experiments with a scaled model. The aerodynamic coefficients are searched in the database for different operational conditions and used to calculate the lift and drag, as follows:

$$\begin{cases} L_i = \frac{1}{2} \rho v_{e,i}^2 S_i C_{L,i} \\ D_i = \frac{1}{2} \rho v_{e,i}^2 S_i C_{D,i} \end{cases} \quad (4)$$

where i is the wing segment number; ρ is the density of air; and $v_{e,i}$, S_i and \bar{c}_i are the effective airspeed, surface area and MAC of the i^{th} segment, respectively. The transformation matrix then converts the freestream coordinates into body coordinates, as follows:

$$\mathbf{R}_f^b = \begin{bmatrix} -\cos\alpha_i & 0 & -\sin\alpha_i \\ 0 & 0 & 0 \\ -\sin\alpha_i & 0 & \cos\alpha_i \end{bmatrix} \quad (5)$$

where α_i is the angle of attack of the i^{th} segment and can be defined as:

$$\alpha = \tan^{-1}(V_{as,x}^b / V_{as,z}^b) \quad (6)$$

where $V_{as,x}^b$ and $V_{as,z}^b$ are the local airspeed velocity in the direction of x -axis and z -axis in body coordinate. To obtain the flow condition on the wing, the freestream velocity in body coordinate, \mathbf{V}_{as}^B , can be calculated by adding the contribution of propeller wash in flow field, which can be write as

$$\mathbf{V}_{as}^B = \mathbf{R}_I^B(\mathbf{V}_{wind}^I + \mathbf{V}_{gs}^I) + \mathbf{V}_{induce}^B \quad (7)$$

where the wind speed \mathbf{V}_{wind}^I and the ground speed \mathbf{V}_{gs}^I are described in the inertia frame (NED) and \mathbf{R}_I^B is the transformation matrix from the inertia frame to the body frame. \mathbf{V}_{induce}^B is the propeller slipstream velocity.

The aerodynamic force and moment are expressed as follows:

$$\mathbf{F}_{aero} = \begin{bmatrix} F_{aero,x} \\ F_{aero,y} \\ F_{aero,z} \end{bmatrix} = \sum_{i=1}^5 \mathbf{R}_f^b \begin{bmatrix} L_i \\ 0 \\ D_i \end{bmatrix} \quad (8)$$

$$\mathbf{M}_{aero} = \sum_{i=1}^5 \begin{bmatrix} F_{aero,z}^i l_y^i \\ \frac{1}{2} C_{M,i} \rho v_{eff,i}^2 S_i \bar{c}_i + F_{aero,x}^i l_z^i \\ F_{aero,x}^i l_y^i \end{bmatrix}$$

The aerodynamic force and moment are modeled and fed forwards to the controller as the estimated disturbance to minimize the model mismatch problem. This aerodynamic model is also used in the software-in-loop (SIL) simulation. In this study, the control surfaces were not used in the hovering flight. Because the four motors and propellers can provide much larger torque to control the vehicle compared to the control surfaces. The neglect of the control surface can significantly simplify the modelling process and the complexity of the controller design for hovering. During the transition, the aerodynamic effect becomes critical since the wing provides the lift rather than the propulsion system. As a result, the focus point of a transition control is completely different from a hovering control and more detail has been shown in the author's work [(Boyang Li, 2018)].

3. Development of Successive Linearization Based MPC Controller

This section presents the development of a successive linearization based MPC controller for the tail-sitter vehicle. Fig. 3 shows the cascaded control structure, where all states are assumed to be measurable by the onboard estimator. The cascaded structure has the advantage of reducing the onboard computational load. Both controllers have relatively few states, which is more practical for onboard flight control missions, as the computational effort increases dramatically as the number of MPC controller states increases. This strategy allows for a faster updating rate in the attitude control than the position control, lowering the possibility of crashing the UAV due to attitude loss. As a result, reliability and robustness are improved.

This study focuses on applying a successive linearization based MPC to position control, which generates thrust (f_T) and attitude commands ($\phi_{cmd}, \theta_{cmd}, \psi_{cmd}$) according to the reference trajectory (\mathbf{p}_{cmd}). A PID attitude controller then generates torque commands (τ_x, τ_y, τ_z) according to the angular movement commands. Actuator mapping then transfers the thrust and torque commands into the throttle (η) for each of the four motors according to the propulsion system model. Notably, the gravitational force term is added in the Actuator Mapping block, with the thrust command f_T calculated from the position

controller to determine the total thrust. The UAV dynamic includes a propulsion model (3) and an aerodynamic model (8). The details of the current SLMPC controller are introduced below

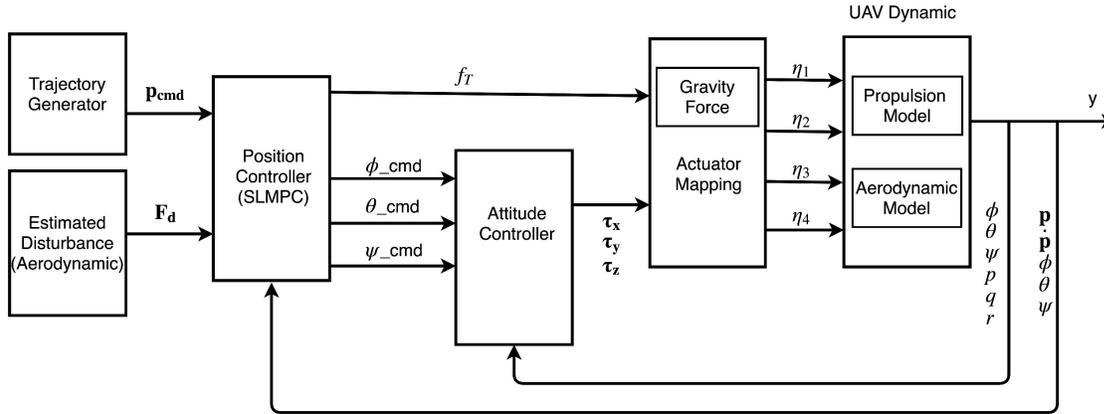


Fig. 3. Block diagram of the cascaded control structure of the SLMPC controller used in the PolyU Plus tail-sitter vehicle.

3.1. State-Space Model

The dynamic model of (1a) and (1b) can be written in a state-space form as follows:

$$\begin{aligned} \mathbf{x}(k+1) &= \bar{\mathbf{x}} + \mathbf{A}_x(\mathbf{x}(k) - \bar{\mathbf{x}}) + \mathbf{B}_u(\mathbf{u}(k) - \bar{\mathbf{u}}) \\ \mathbf{y}(k) &= \bar{\mathbf{y}} + \mathbf{C}_x(\mathbf{x}(k) - \bar{\mathbf{x}}) \end{aligned} \quad (9)$$

where $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$ and $\bar{\mathbf{y}}$ are the nominal state, input and output respectively. The system state, control input and control output are set up as follows:

$$\begin{aligned} \mathbf{x} &= [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T \\ \mathbf{u} &= [f_T \ \sin\phi \ \sin\theta \ \sin\psi]^T \\ \mathbf{y} &= [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T \end{aligned} \quad (10)$$

The control output (\mathbf{y}) is the same as the system state (\mathbf{x}), as all of the states are assumed to be fully measured in this study. The input signal is fed into an *arcsin* function and transformed into ϕ_{cmd} , θ_{cmd} and ψ_{cmd} before being sent to the attitude controller.

The defined state matrix \mathbf{A}_x , input matrix \mathbf{B}_u and \mathbf{C}_x are illustrated as:

$$\mathbf{A}_x = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}$$

$$\mathbf{B}_u = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{f_T \sin\psi}{2m} & \frac{f_T \cos\phi \cos\psi}{m} & \frac{f_T \sin\phi}{2m} \\ 0 & -\frac{f_T \cos\psi}{m} & \frac{f_T \cos\phi \sin\psi}{2m} & \frac{f_T \cos\phi \sin\theta}{2m} \\ \frac{\cos\phi \cos\theta}{m} & 0 & 0 & 0 \end{bmatrix} \quad (11)$$

$$\mathbf{C}_x = \mathbf{I}_{6 \times 6}$$

Non-linearity is adopted by successively discretizing the model at every time step, as the SLMPC can consistently update parameters and nominal values at every time step. While the operation condition varies, the nominal value can be updated, and a much larger control envelope can be covered without the risk of model mismatch, such as with the linear MPC model. At each time step, the generated command thrust (f_T) adjusts the altitude of the vehicle. Then f_T and the latest measured attitude angles (ϕ , θ and ψ) are used to update the transfer matrix (\mathbf{B}_u) and calculate a new control input.

3.2. Prediction Model

The schematic diagram shown in Fig. 4 demonstrates the structure of an MPC controller, in which $\hat{\mathbf{x}}$ is the predicted state, \mathbf{x}_{ref} is the reference state and $\hat{\mathbf{u}}$ is the predicted input. The predicted input sequence is the solution of an optimisation problem, which involves minimising the quadratic cost function over a finite prediction horizon within the constraint set. The unknown disturbance affects the plant and is mitigated by a feedback integration strategy. The estimated disturbance is fed into the plant along with the prediction model to allow the controller to consider its effect and reduce the error.

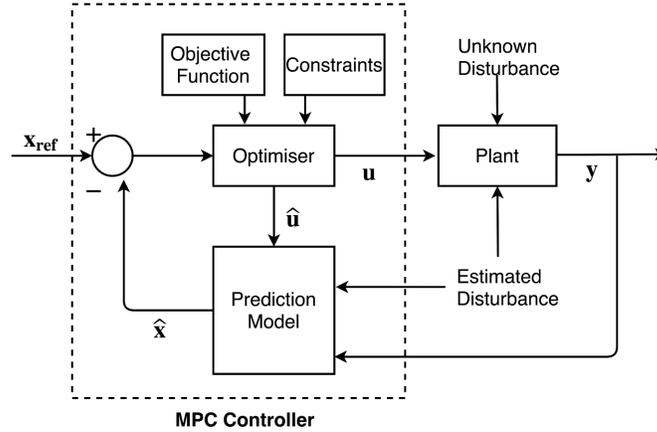


Fig. 4. Structure of the proposed MPC controller with an augmented system model.

The predicted input $\hat{\mathbf{u}}$ can be expressed as the summation of the last control input $\mathbf{u}(k - 1)$ and the input increment $\Delta \mathbf{u}$ as:

$$\begin{bmatrix} \hat{\mathbf{u}}(k|k) \\ \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \hat{\mathbf{u}}(k+H-1|k) \\ \vdots \\ \hat{\mathbf{u}}(k+P-1|k) \end{bmatrix} = \mathbf{I}_{P \times 1} \mathbf{u}(k-1) + \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ I & I & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & I \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I & I & \dots & \dots & I \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}(k|k) \\ \Delta \mathbf{u}(k+1|k) \\ \vdots \\ \Delta \mathbf{u}(k+H-1|k) \\ \vdots \\ \Delta \mathbf{u}(k+P-1|k) \end{bmatrix} \quad (12)$$

Here $\Delta \mathbf{u}(k+i|k)$ is the input increment at the k^{th} time step and the i^{th} prediction horizon, where P is the number of prediction horizons and H is the number of control horizon, which $1 \leq H \leq P$.

By propagating one step at a time of (9) and combining the relation in (12), the predicted state $\hat{\mathbf{x}}$ can be shown as:

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{S}^x \mathbf{x}(k) + \mathbf{S}^u \mathbf{u}(k-1) + \mathbf{S}^{\Delta u} \Delta \mathbf{u} - \mathbf{S}^{\bar{u}} \bar{\mathbf{u}} \quad (13)$$

where $\hat{\mathbf{x}} = [\hat{\mathbf{x}}(k|k) \ \hat{\mathbf{x}}(k+1|k) \ \dots \ \hat{\mathbf{x}}(k+P-1|k)]^T$ and

$$\mathbf{S}^x = \begin{bmatrix} \mathbf{A}_x \\ \mathbf{A}_x^2 \\ \vdots \\ \mathbf{A}_x^H \\ \vdots \\ \mathbf{A}_x^P \end{bmatrix}, \quad \mathbf{S}^u = \mathbf{S}^{\bar{u}} = \begin{bmatrix} \mathbf{B}_u \\ \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u \\ \vdots \\ \sum_{h=0}^H \mathbf{A}_x^h \mathbf{B}_u \mathbf{S}^u \\ \vdots \\ \sum_{h=0}^P \mathbf{A}_x^h \mathbf{B}_u \end{bmatrix},$$

$$\mathbf{S}^{\Delta u} = \begin{bmatrix} \mathbf{B}_u & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u & \mathbf{B}_u & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{h=0}^H \mathbf{A}_x^h \mathbf{B}_u & \vdots & \vdots & \mathbf{B}_u + \mathbf{A}_x \mathbf{B}_u & \mathbf{B}_u \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{h=0}^P \mathbf{A}_x^h \mathbf{B}_u & \dots & \dots & \dots & \sum_{h=0}^{P-H} \mathbf{A}_x^h \mathbf{B}_u \end{bmatrix}$$

The nominal value $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are updated in every time step and remain constant though out the whole prediction horizon same as the real time state $\mathbf{x}(k)$ and the last control input $\mathbf{u}(k-1)$. The predicted state can be substituted back into the output model of (9) and the predicted output can be expressed as:

$$\hat{\mathbf{y}} = \bar{\mathbf{y}} + \mathbf{C}_x \hat{\mathbf{x}} \quad (14)$$

3.3. Augmented Model

A normal feedback control strategy only takes corrective action once the effect of the disturbance is apparent. However, in this study, the aerodynamic disturbance is treated as estimated disturbance and added to the prediction model, so that the optimizer solves the problem by taking this effect into account. In the model, different wind speeds and wind directions will cause different aerodynamic forces, which will affect the vehicle's position. An experiment was conducted to measure the wind speed coming from a single wind direction. The experimental results of the mean wind speed and the variance of unsteady wind are presented in Section 5.1 and used to estimate the disturbance. To describe the disturbance in a general form, \mathbf{v} is used to represent the input of the estimated disturbance and added to (9). Accordingly, the prediction model is written as follows:

$$\mathbf{x}(k+1) = \bar{\mathbf{x}} + \mathbf{A}_x (\mathbf{x}(k) - \bar{\mathbf{x}}) + \mathbf{B}_u (\mathbf{u}(k) - \bar{\mathbf{u}}) + \mathbf{B}_v \mathbf{v}(k) \quad (15)$$

$$\mathbf{y}(k) = \bar{\mathbf{y}} + \mathbf{C}_x(\mathbf{x}(k) - \bar{\mathbf{x}})$$

If the direction and magnitude of the wind are known, the aerodynamic force and moment can be estimated (Section 2.4). For the position controller, the definitions of \mathbf{x} and \mathbf{u} remain unchanged, but \mathbf{F}_{aero} in (8) is set as \mathbf{v} and the corresponding transfer matrix (\mathbf{B}_v) is derived as follows:

$$\mathbf{B}_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/m & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/m \end{bmatrix} \quad (16)$$

According to the characteristics of the vehicle, an increase in thrust from propellers 1 and 2 (Fig. 1a) increases the induced velocities over wing segments 2 and 4 (Fig. 2). The vehicle pitches down, as there is an increase in the lift force on the AC of the wing and creates an extra moment. As a result, an increase in thrust induces a sudden unfavorable pitch movement and may cause a deviation in the position in the x -direction. Using this aerodynamic disturbance model, the controller can compensate for unwanted aerodynamic effects.

It is practically impossible to accurately model all the disturbances that act on a system. A feedback integration strategy is introduced to eliminate the effects of unknown disturbances, model mismatch, and measurement noise (Rawlings, Mayne, & Diehl, 2012; Rossiter, 2003; Zagrobelny, 2014). This strategy can easily augment the disturbance feedback policy to an already developed dynamic system. By adding extra terms to the prediction model, the SLMPC controller can achieve offset-free control in a similar way to the integration effect of a traditional PID control. The input of the unknown disturbance is expressed as follows:

$$\mathbf{d}(k) = \mathbf{y}_{ref}(k) - \mathbf{y}(k) \quad (17a)$$

$$\mathbf{x}_d(k+1) = \mathbf{x}_d(k) + \mathbf{d}(k) \quad (17b)$$

where \mathbf{d} is the input disturbance, which is the error between the measured output and the reference output, and \mathbf{x}_d is the state disturbance, which is added to the system state.

Combining (15) and (17), the augmented state-space model can be summarised as follows:

$$\begin{aligned} \mathbf{x}_A(k+1) &= \bar{\mathbf{x}}_A + \mathbf{A}(\mathbf{x}_A(k) - \bar{\mathbf{x}}_A) + \mathbf{B}(\mathbf{u}_t(k) - \bar{\mathbf{u}}_t) \\ \mathbf{y}(k) &= \bar{\mathbf{y}} + \mathbf{C}(\mathbf{x}_A(k) - \bar{\mathbf{x}}_A) \end{aligned} \quad (18)$$

where the augmented state is $\mathbf{x}_A(k) = [\mathbf{x}(k) \quad \mathbf{x}_d(k)]^T$ and the augmented nominal state is $\bar{\mathbf{x}}_A$. The total input is $\mathbf{u}_t = [\mathbf{u}(k) \quad \mathbf{v}(k) \quad \mathbf{d}(k)]^T$ and the nominal total input is $\bar{\mathbf{u}}_t$. The lumped matrix can be expressed as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_x & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_u & \mathbf{B}_v & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{C} = [\mathbf{C}_x \quad \mathbf{I}] \quad (19)$$

Note that $\bar{\mathbf{x}}_d = \bar{\mathbf{v}} = \bar{\mathbf{d}} = \mathbf{0}$ at all times, as there is no nominal value for any disturbance-related term a priori.

3.4. Objective Function and Constraints

A squared 2-norm composed of the output error and the input increment from each time instant to the prediction horizon is used as the objective function:

$$J(\mathbf{z}_k) = \sum_{i=0}^{P-1} \{[\mathbf{e}_y^T(k+i)\mathbf{Q}\mathbf{e}_y(k+i)] + [\Delta\mathbf{u}^T(k+i)\mathbf{R}_{\Delta u}\Delta\mathbf{u}(k+i)]\} \quad (20)$$

where \mathbf{Q} and $\mathbf{R}_{\Delta u}$ are positive-semi-definite matrices with weight information; and \mathbf{e}_y and $\Delta\mathbf{u}$ are the respective errors in the output and input increments, which can be expressed as follows:

$$\begin{aligned} \mathbf{e}_y(k+i) &= \mathbf{S}_y^{-1}[\mathbf{r}(k+i+1|k) - \hat{\mathbf{y}}(k+i+1|k)] \\ \Delta\mathbf{u}(k+i) &= \mathbf{S}_u^{-1}[\mathbf{u}(k+i|k) - \mathbf{u}(k+i-1|k)] \end{aligned} \quad (21)$$

The optimization process included the error between the current reference value \mathbf{r} and the predicted output $\hat{\mathbf{y}}$ of (12) such that the controller can take the future into account. If reference values are loaded into the optimiser in advance from 1 to $P-1$ steps, the optimiser can generate a predicted input that accounts for future error. If not, every reference value is considered a constant for the horizon. \mathbf{S}_y and \mathbf{S}_u are the diagonal matrices of the scale factors of the control output and input in engineering units.

Apart from the setup of the objective function, the variables are constrained according to the physical system and expressed as follows:

$$\mathbf{y}_{j,min}(i) \leq \mathbf{y}_j(k+i|k) \leq \mathbf{y}_{j,max}(i) \quad (22)$$

for $i = 1:P, j = 1:n_y$

$$\Delta\mathbf{u}_{j,min}(i) \leq \Delta\mathbf{u}_j(k+i-1|k) \leq \Delta\mathbf{u}_{j,max}(i) \quad (23)$$

for $i = 1:P, j = 1:n_u$

where n_y and n_u are the numbers of outputs and inputs, respectively; $\mathbf{y}_{j,min}(i)$ and $\mathbf{y}_{j,max}(i)$ are the lower and upper bounds of output, respectively, for the j^{th} plant output at the i^{th} horizon step; and $\Delta\mathbf{u}_{j,min}(i)$ and $\Delta\mathbf{u}_{j,max}(i)$ are the lower and upper bounds of input increments.

When the objective function and constraints are determined, quadratic programming (QP) problem can be derived. An open-loop optimization is repeated until the control horizon H is reached to eliminate the error between the reference state and the predicted state within the prediction horizon (P). The QP decision is obtained as follows:

$$\mathbf{z}_k^T = [\mathbf{u}(k|k)^T \ \mathbf{u}(k+1|k)^T \ \dots \ \mathbf{u}(k+H-1|k)^T \ \dots \ \mathbf{u}(k+P-1|k)^T] \quad (24)$$

where the control input \mathbf{u} will be calculated by solving the QP problem until the $H-1$ step and stays at the same value until the $P-1$ step. Only the first term is implemented as a control effort at each time step, according to the receding horizon control algorithm.

3.5. Time-Varying Weight

According to the objective function described in (20), when $\mathbf{e} \neq 0$, the objective function drives the output error towards zero, with a priority that depends on the weight parameters, to minimise the cost. In general, a step command creates a large, sudden output error such that the cost of the objective function increases significantly, and the control law consequently uses the maximum effort to eliminate it immediately. In most cases, this maximum effort causes excessive movement and often results in an unacceptable oscillatory response, especially in real-time flight tests.

To ease the system's response and reduce oscillation, a large value for input increment weight ($\mathbf{R}_{\Delta u}$) is used, which penalises large input increments ($\Delta \mathbf{u}$) at every control horizon. Form (20) the cost will increase as $\mathbf{R}_{\Delta u}$ increases for the same $\Delta \mathbf{u}$. Thus, in order to lower the cost, the controller will give smaller $\Delta \mathbf{u}$ as $\mathbf{R}_{\Delta u}$ increases. Meanwhile, the output of (20) can also be adjusted to ease the system response by varying the output variable weight (\mathbf{Q}) from one step to the next. In order to allow \mathbf{Q} varies as the horizon changes, (20) can be written as:

$$J(\mathbf{z}_k) = \sum_{i=0}^{P-1} \{[\mathbf{e}_y^T(k+i)\mathbf{Q}_i\mathbf{e}_y(k+i)] + [\Delta \mathbf{u}^T(k+i)\mathbf{R}_{\Delta u}\Delta \mathbf{u}(k+i)]\} \quad (25)$$

where \mathbf{Q}_i is a diagonal matrix containing weight information of each output variables and is described as:

$$\mathbf{Q}_i = \text{diag}[Q_i^1 \quad Q_i^2 \quad \dots \quad Q_i^j] \quad (26)$$

for $i = 0:P-1, j = 1:n_y$

According to (Wojsznis, Gudaz, Blevins, & Mehta, 2003), the coefficient in \mathbf{Q}_i is linearly increased from the beginning of the horizon to its set value. The cost will no longer increases abruptly, and the first move will be smaller, resulting in an increase of robustness. Although this approach complicates the tuning process, it provides an additional opportunity to adjust the control performance.

3.6. Velocity Reference

The position is controller to the desired setpoint by trying to bring the value of the position error to 0 at every time step. The position error is defined as follows:

$$\mathbf{p}_{error} = \mathbf{p}_{ref} - \mathbf{p}_{mea} \quad (27)$$

where \mathbf{p}_{ref} is the reference position and \mathbf{p}_{mea} is the measured position. Currently, the controller has state contains both the position and the velocity ($\mathbf{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$). Usually, only the position $[x \ y \ z]$ is given with weights in the objective function since in most of the case, a trajectory contains only position reference. However, the time derivative of the position error \mathbf{v}_{ref} can be set as a velocity reference for $[\dot{x} \ \dot{y} \ \dot{z}]$. In addition to the position weight, \mathbf{v}_{ref} is also given with weights in the objective function. Not only the position is considered and controlled by the optimization process, but also the velocity. The \mathbf{v}_{ref} is assumed to be the position error divided by a time constant T_c , as follows:

$$\mathbf{v}_{ref} = \frac{\mathbf{p}_{error}}{T_c} \quad (28)$$

The smaller the time constant is, the faster the response is. The weight of \mathbf{v}_{ref} in the objective function can create an effect similar to that of the derivative term for traditional PID control. When there is a large position error, the objective function generates a large command. As it nears the target, the position error becomes small, so the vehicle starts to slow down. This also reduces the overshoot effect.

4. Simulation

In this section, a set of SIL simulation aims to check the developed control algorithm and obtain a set of parameters that can perform a reliable control. The proposed control method is then put into a trajectory tracking mission to prove its performance by comparing with a linear MPC controller. The effectiveness of easing the control response without sacrificing the performance and energy efficiency by introducing the method of the time-varying weight and the velocity reference has also been tested.

4.1. Parameters and Settings

The values of the variables must be adjusted according to their importance in achieving reliable and robust performance. The sampling time (or the duration of each control interval; T_s) is set to 0.04 seconds for the simulation, so the control algorithm runs at 25 Hz. As T_s decreases, unknown disturbances are rejected more effectively, but the computational load increases dramatically. As a result, the optimal choice of sampling time is a balance between the control performance and the corresponding computational effort.

The prediction horizon (P) must not be too large, as the controller's memory requirement increases when solving the QP problem. However, P cannot be too small because constraint violations will then be unforeseen. In Fig. 4, the error between the predicted state $\hat{\mathbf{x}}$ and the reference state \mathbf{x}_{ref} is fed into the optimiser. If the future trajectory is used in advance, the future command signal can be considered by the objective function. As the optimiser generates the predicted input only after taking the whole prediction horizon into account, future variation in the trajectory will affect the position error in the prediction horizon and therefore affect the control input. At the next time step, the trajectory command will shift one step forwards and be added, such that the optimiser gradually sees the change in the reference trajectory. As a result, the control input can be given before rather than after the error has been measured. If T is introduced as the desired prediction duration, then $T \approx T_s \times P$. For example, if P is set to 50 the prediction duration is $T \approx 0.04 \times 50 = 2$ seconds ahead.

The control horizon (H) must fall between 1 and P . Regardless of the choice of H , when the controller operates, only the first optimised control move is used, and any others are discarded from (24). A small value for H means that there are fewer variables to be solved in the QP problem at each control interval. Therefore, computations are faster. However, optimisation is less effective and vice versa.

Non-negative scale vectors (\mathbf{S}) in (21) are divided by each plant input and output to generate dimensionless signals. If any signal has a significantly larger or smaller magnitude than the others, this defined scale factor is especially important.

Adjusting the weight of each variable is critical to the performance of the controller. For the output variable weights (\mathbf{Q}) in (20), a higher weight means a higher priority. The input increment weights ($\mathbf{R}_{\Delta u}$) penalise large input changes in the optimisation cost function. For example, simultaneously increasing the input increment weights and the corresponding output variable weights produces a more robust controller, as the controller limits movement and gives a higher priority to feedback at each control interval. The detailed parameter settings are presented in Table 1 for the SLMPC controller used in the 'PolyU Plus' tail-sitter vehicle.

Table 1. Parameter settings for the MPC controller.

| Parameter | Value |
|----------------|-----------------------------|
| T_s | 0.04 |
| P | 50 |
| H | 4 |
| Q | $diag\{5.5,4,5.5,2,2,2.5\}$ |
| $R_{\Delta u}$ | $diag\{10,35,30,25\}$ |
| $ y $ | $[inf, inf, inf, 5,5,5]^T$ |
| $ \Delta u $ | $[10,0.75,0.75,0.7]^T$ |
| S_y | I |
| S_u | $[15,0.8,1,1]^T$ |

4.2. The Effectiveness of Time-Varying Weight

The theoretical improvement brought about by adding a time-varying weight is introduced in Section 3.5. According to the experiences of flight tests, any large and abrupt control command is to be avoided since it will cause an excessive response of the vehicle and usually need extra effort for correction. In order to achieve stable and efficient control when a large and abrupt control command is encountered, the time-varying weight is a good measure to ease the response.

The simulation results of following a step command with measurement noise in the x -direction position are shown in Fig. 5. The results of fixed weights of $Q=2.0$ (green line) and $Q=5.5$ (blue line) and a time-varying weight (red line) are compared. According to the objective function in (25), a smaller fixed weight $Q=2.0$ in the x -position will make the cost smaller, compared with the case of $Q=5.5$. The priority of eliminating the error in x -direction is lower when $Q=2.0$, yielding a much larger overshoot in the x -direction and a much longer regression time. Increasing Q from 2.0 to 5.5 will reduce the overshoot in the x -direction and shorten the regression time, yet it creates large system responses in θ and q . Further increase in the fixed weight Q will deteriorate system responses in θ and q . The most significant improvement of the time-varying weight added, compared to the fixed weight $Q=5.5$, is that the maximum pitch angular rate (q) at the 5th second is reduced from from $-200^\circ/s$ to $-100^\circ/s$ and the maximum pitch angle (θ) also decreases. However, the minimum time required to reach the range of 5% error in controlling the x -axis position is similar. In the real flight, it is obvious that a lower maximum angular rate requires a much smaller sudden change in torque, which is strongly related to the rotation speed, and result in a lower energy consumption. Above all, the results indicate that the time-varying weight can ease the control responses and improve the robustness without compromising the performance.

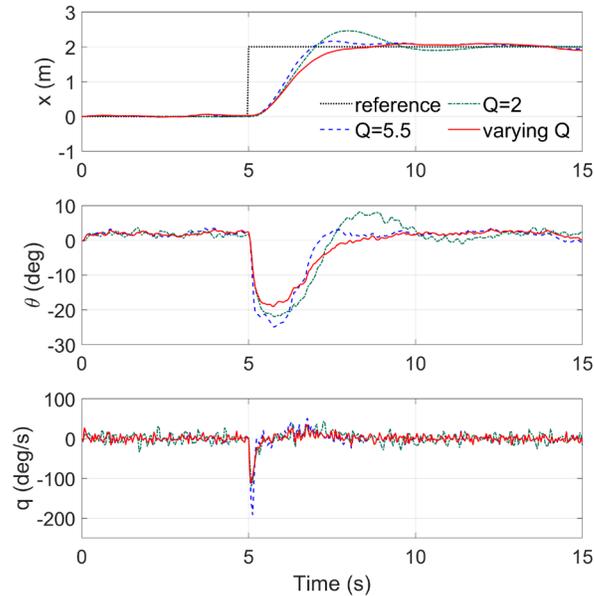


Fig. 5. Simulation results of step-command following with fixed weights of $Q=2.0$. and $Q=5.5$ and a varying weight.

4.3. The Effectiveness of Velocity Reference

A simulation result of following a step signal by the proposed SLMPC controller is shown in Fig. 6. For the SLMPC control itself (blue line), it tends to use maximum control effort to minimize the error between the reference and the output in order to achieve a smaller value of the objective function. As a result, a longer duration in the maximum pitch angle ($\theta \approx -20^\circ$) is commanded and reaches the setpoint earlier. However, the subsequent pitch up motion ($\theta > 0$) slows down the x -direction velocity, but overshoot has already been caused.

With the velocity reference added (red line), the pitch angle (θ) starts to reduce at the 6th second rather than at the 6.5th second with the original controller (blue line). Compared with the case without the velocity reference added (the blue line), this earlier decrease in the pitch angle can reduce the x -direction speed at an earlier stage and as a result, a much smaller overshoot is observed in the x -direction position (the red line). Notably, for the case without the velocity reference added (the blue line), there is a significant positive pitch movement between the 7th second and the 9th second to compensate the overshoot of the x -direction position. Meanwhile, the reaction time of the system is not compromised because both cases (lines) reach the reference value at the same time in the x -direction position. With this measure taken, in the real-time flight tests, the vehicle will have less fluctuation on both the attitude control and position control.

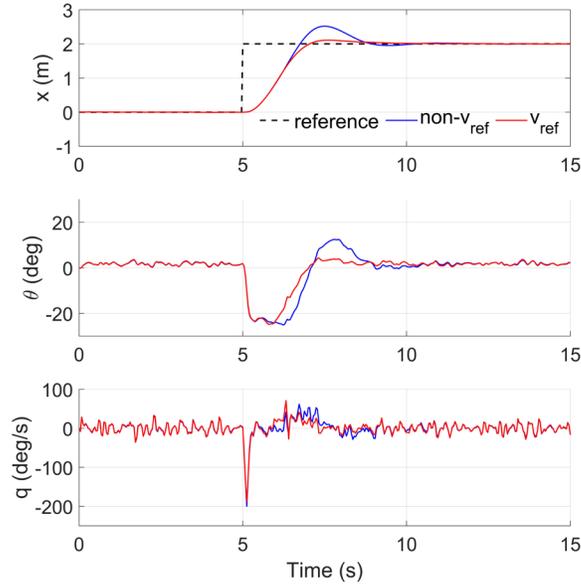
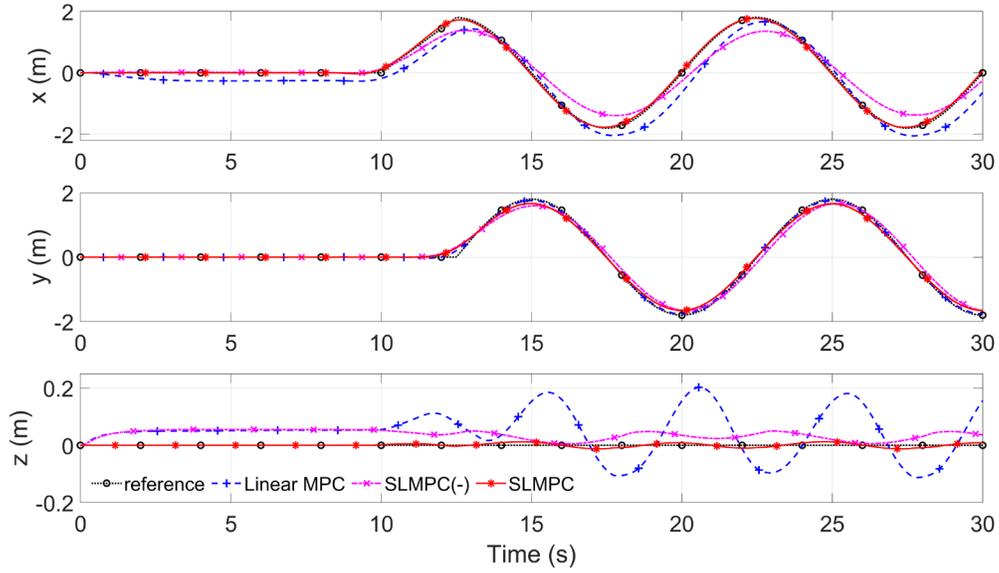


Fig. 6. Simulation results of step-command following without and with velocity as reference.

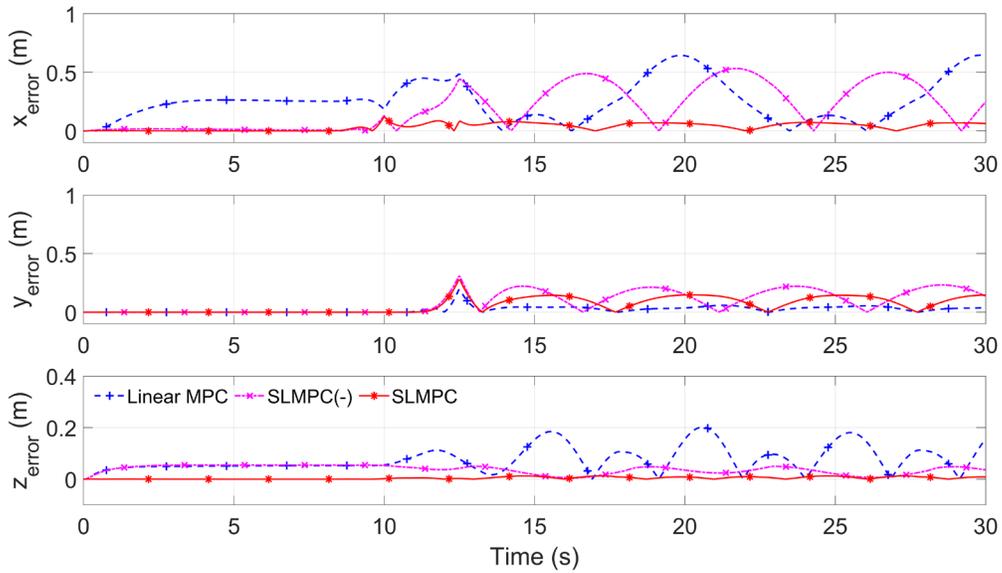
4.4. Comparison of SLMPC and Linear MPC

The linear MPC controller from the previous work of (B. Li, W. Zhou, J. Sun, C.-Y. Wen, & C.-K. Chen, 2018) and the proposed SLMPC controller is compared by performing a trajectory following mission simulation. The trajectory is a 4-meter diameter circular path in the same high in order to better distinguish the performance in altitude tracking. The performance on position tracking is shown in Fig. 7 (a). For the linear MPC controller (blue line), a steady-state error in the x -direction can be observed in Fig. 7 (b) due to the aerodynamic lift and drag of the wing itself has not being considered by the controller. When it starts to track the circle trajectory, significant unwanted variation in the z -direction can also be observed. According to the characteristics of this tail-sitter vehicle, the manoeuvrability in the x -direction is much weaker than the y -direction due to the wing. As a result, from Fig. 8, following the circle path requires maximum 5° in roll (ϕ) but 20° in pitch (θ). Whenever the pitch is large, the high drops ($+0.2m$ in z -direction). Since the linear MPC controller is developed base on a zero-degree Euler angle trim point. A 20° in pitch is too far from the trim point, the model is no longer matched and leading to inaccurate in the control effort. The z -direction is ignored while maximum effort has been given in keeping the x -direction causing this drop in high. The high is restored only when the priority of the x -direction is lower.

For the SLMPC without the augmented model (pink line), the performance in x -direction is also imprecise, no matter how the weight has been adjusted in the objective function. However, the z -direction has a slightly better performance than the linear MPC even though the steady-state error still exists. For the proposed SLMPC with the augmented model (red line), the following of the reference command in all direction is precise and stable without any steady-state error.



(a)



(b)

Fig. 7. (a) Position of trajectory following simulation of a circular path. (b) The corresponding

From Fig. 8, it can be observed that, the yaw angle (ψ) of the linear MPC has stayed in 0° while the yaw angle of SLMPC has deviated from it. As shown in Fig. 3, the yaw angle command (ψ_{cmd}) is generated by the MPC position controller and followed by the PID attitude controller. Although the yaw angle is rather irrelevant to the vehicle position, it is included in the control input in (10) and still affects the objective function and the position optimization. Unlike the linear MPC which commands the yaw angle to its trim point (0°), the yaw command from the SLMPC is its current yaw angle. Due to the aerodynamic characteristics of the vehicle, whenever there is a roll movement, the yaw changes. From Fig. 1(b), a roll movement is achieved by the difference of rotation speeds of propellers 1 and 2, which create a torque to roll. The difference in rotation speeds of propellers 1 and 2 will cause the different slipstream velocities on

the wing surface. As a result, uneven lift forces on two sides of the wing simultaneously cause an unwanted yaw motion. Real-time linearization is adapted by the SLMPC control which can immediately accept the current yaw angle on the optimization process and control the position with the lowest cost, consequently. Unlike the linear MPC control, there is no extra control effort spent on yawing the vehicle back to 0° .

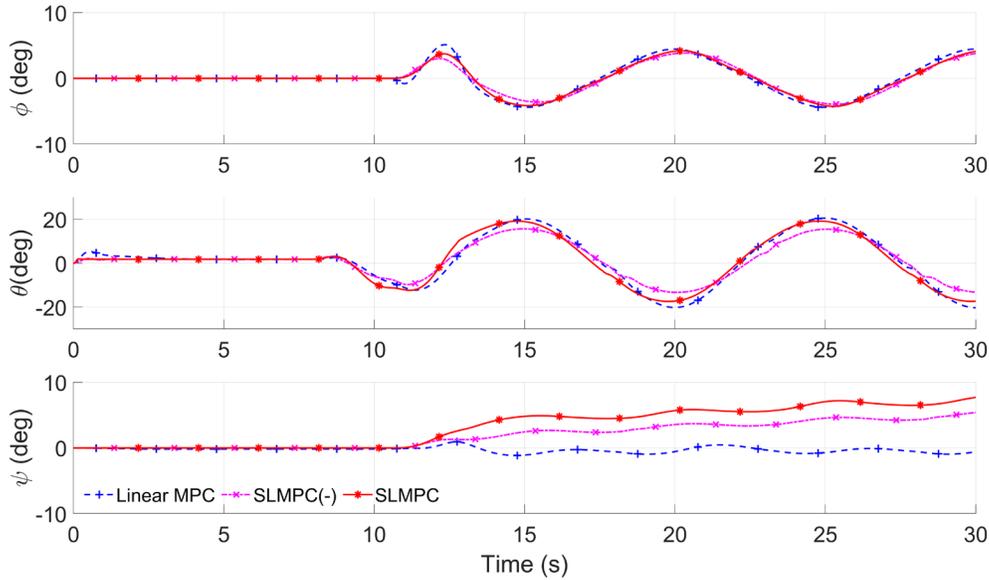


Fig. 8 The Euler angles of trajectory following simulation of a circular path.

Disturbance resistance controlled by the linear MPC and SLMPC with the wind of 1 m/s , 2 m/s and 3 m/s magnitudes coming from 0° to north (negative X^I -direction in Fig. 1) is presented in Fig 9. As shown, the shaded area indicates the period when the wind encounters the vehicle. The x -direction position error increases as the wind speed increases. Under the large wind speed of 3 m/s , the linear MPC cannot steer the vehicle back to reference value until the wind effect disappears. The vehicle becomes unstable when controlled by the SLMPC, because it fluctuates around the reference value and converges slowly. The corresponding root mean square errors ($RMSE = RMSE_x + RMSE_y + RMSE_z$) are presented in Fig. 10, with the wind speed ranging from 0.5 m/s to a tolerance of 3.3 m/s . The RMSE of SLMPC is smaller than that of the linear MPC and the performance of SLMPC in the extreme case of 3.3 m/s wind speed is much better than the linear MPC. The vehicle controlled by a linear MPC has failed to hold the position under a 3.3 m/s wind since it has been completely blown away in the x -direction before the cease of the wind.

The effectiveness of disturbance resistance by both the controllers with the wind of 3 m/s coming from 0° to 180° (clockwise) to the north is illustrated in Fig. 11. From the results, the variation of RMSE complies with the characteristics of the vehicle. The errors caused by the wind coming from $\pm x$ -direction (0° or 180°) is much larger than that from the y -direction (90°), since there is large wing surface area facing the wind in the $\pm x$ -direction. Overall, the RMSE of SLMPC is smaller than that of the linear MPC.

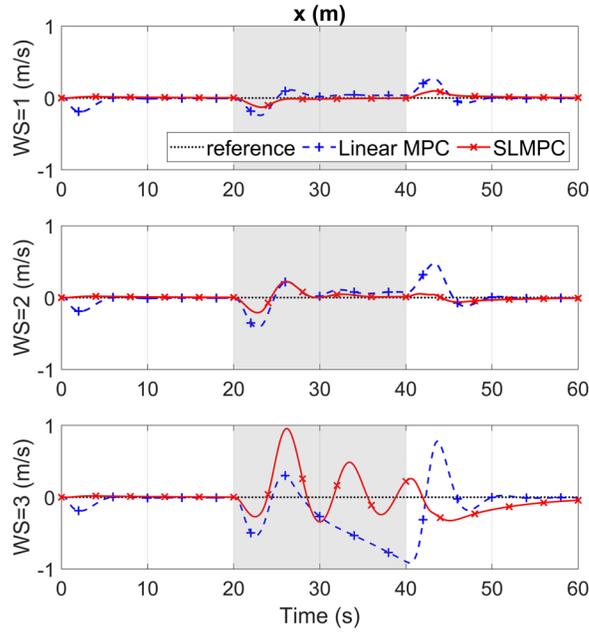


Fig. 9. Position control with disturbance rejection when the wind of 1 m/s, 2 m/s and 3 m/s magnitudes coming from 0° to north (negative X^l -direction in Fig. 1).

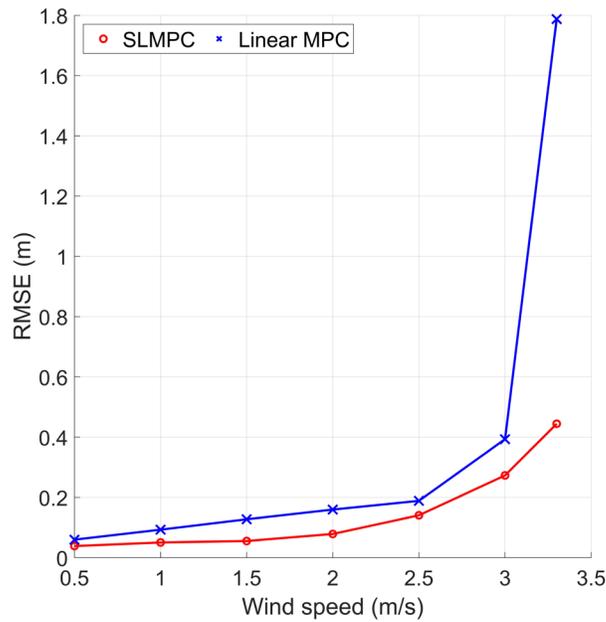


Fig. 10. Root mean square errors of controllers with disturbance rejection when the wind speed ranges from 0.5 m/s to 3.3 m/s and comes from 0° to north (negative X^l -direction in Fig. 1).

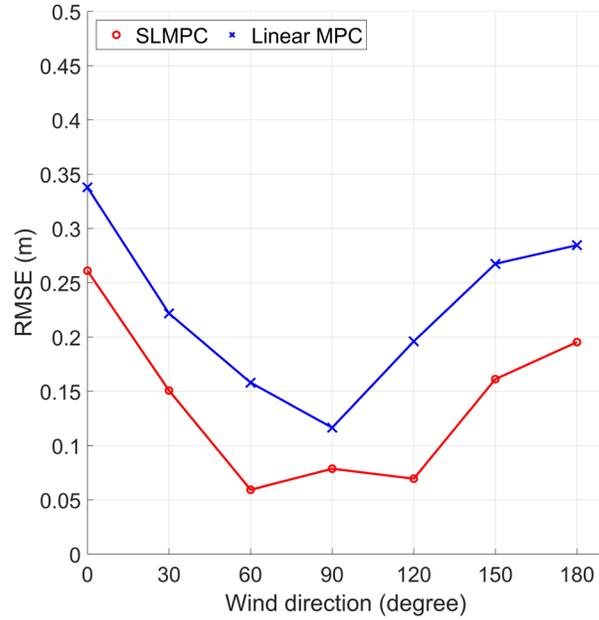


Fig. 11. Root mean square errors of controllers with disturbance rejection when the wind of 3 m/s coming from 0° to 180° .

5. Flight Experiments

The flight tests were conducted to further evaluate the hover flight stability of the tail-sitter vehicle in windy conditions in the aviation laboratory of The Hong Kong Polytechnic University with vehicle's position captures by a motion capture system. The developed SLMPC controller was compared with the PID controller and linear MPC controller. The artificial wind field is created to better control the experimental condition. The details of the experiment are presented followed by the discussion of the results. The worst-case flight tests experiment of disturbance rejection to hold the position of the vehicle while the vehicle's wing is directly facing the wind are performed.

5.1. Experimental Setup

The flight control unit (FCU) consists of commercial autopilot hardware, Pixhawk, and open-source PX4 firmware. An Odroid XU4 companion computer equipped with Samsung Exynos5422 Cortex™-A15 (2Ghz) and Cortex™-A7 Octa core CPUs is installed for real-time SLMPC computation to guarantee the computing efficiency of quadratic programming in the flight tests. A 2.4-GHz WiFi module, serving as the telemetry system, is connected to the companion computer to transmit data between the vehicle and the ground control station. The details of FCU and the communication network used in this study can be found in the authors' earlier work (Li, et al., 2018b).

Notably, the position control is handled by the Odroid, as it has a much powerful computational capability for the SLMPC algorithm. Meanwhile, the attitude control is handled by the basic FCU of the Pixhawk, which includes a PID control and an allocation mapping. The Odroid and the Pixhawk reciprocally exchange the real-time states of the vehicle and the command signals using the MAVLink protocol. The capability and functionality of the FCU has been well validated in real-time under a

Hardware-in-loop environment and presented in the authors' earlier work (B. Li, W. Zhou, J. Sun, C. Wen, & C. Chen, 2018; Jingxuan Sun, Li, Wen, & Chen, 2018).

The experiment is conducted indoors using a VICON motion capture system to obtain real-time positional information for the vehicle, as presented in Fig. 12.

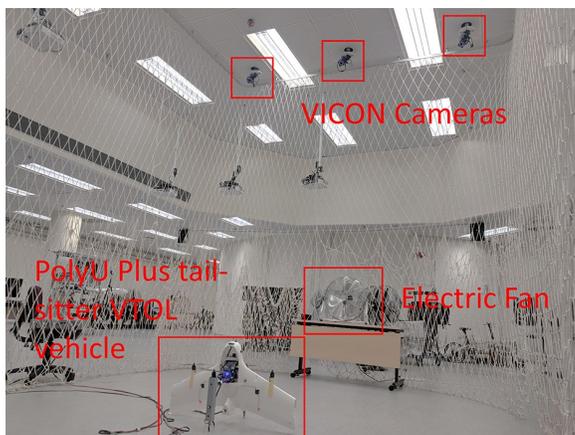


Fig. 12. Indoor flight experiment in a VICON system.

For the indoor flight tests, artificial wind conditions are generated to ensure a similar environmental condition for different flights. An industrial electric fan is placed approximately 3 m away from the position holding point. The Testo 480 digital meter was placed at the position holding point to measure the unsteady wind speed at a sampling frequency of 1 Hz. The result for a 3-minute measurement is demonstrated in Fig. 13. The mean wind speed is 1.88 m/s with a significant fluctuation of 0.58 m/s. This unsteady wind condition simulates a gusty winds outdoor environment. The wind direction and the measured mean value are fed into the estimated disturbance term in (15) and the aerodynamic force is estimated in (8).

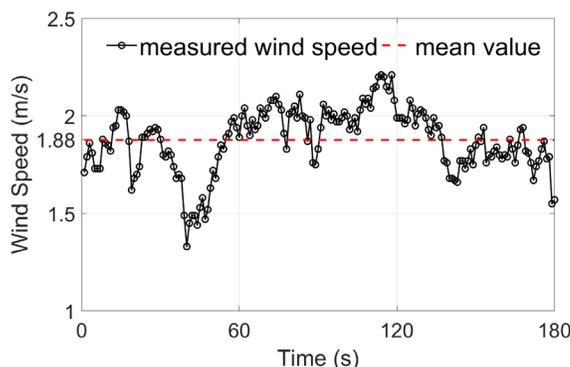


Fig. 13. Measured unsteady wind speed of the artificial wind field.

5.2. Flight Test Results

The designed mission is to hold the vehicle position at $[0, 0, -1]m$ in the x -, y - and z -directions with wind disturbance directly facing the vehicle wing. Zero in the x - and y -directions is the origin and -1 in the z -direction means that the vehicle is set to hover 1 m above the ground. A comparison of the position control performance in the indoor hovering tests, using a traditional PID controller, a linear MPC controller and an SLMPC controller, is presented in this section. The proposed SLMPC position controller is set up with the estimated aerodynamic disturbance and the feedback integration terms augmented. The wind speed

in the environment model is set to 1.88 ± 0.3 m/s according to the experimental results (Section 5.1). Horizontal wind in the negative x -direction is generated from the 30th second to the 50th second.

Case 1 (PID Control): A built-in PID controller in the open-source PX4 firmware has its parameters tuned via the Ziegler-Nichols method. The indoor flight test result under wind disturbance, using the tuned PID controller on the PolyU Plus tail-sitter VTOL vehicle, is shown in Fig. 14. When the wind hits the vehicle (the shadowed area), the vehicle is blown more than 0.5 m and gradually flies back to the setpoint in the x -direction. The integration term of the PID controller is known to affect the system when the error is large enough. Without this integration term, the vehicle would have difficulty eliminating the steady-state error, as a consistent aerodynamic force acts on the wing. Apart from the x -direction, there is a high-frequency oscillation in the roll angle (ϕ) and a large unfavourable movement of up to 30° in the yaw angle (ψ), although the performance in the y - and z -directions is acceptable. Notably, the abrupt changes in the yaw angle (ψ) and z -position during the last few seconds are caused by manual landing process in the experiment. The corresponding results for the root mean square error values of the errors in the x -, y - and z -directions under wind disturbance are shown in Table 2.

Table 2. Comparison of the root mean square error values (m) of the position error under wind disturbance, using the traditional PID, linear MPC (LMPC) and successive linearization based MPC (SLMPC) controllers.

| RMSE | x | y | z |
|-------|--------|--------|--------|
| PID | 0.2214 | 0.0889 | 0.1451 |
| LMPC | 0.1615 | 0.0478 | 0.0668 |
| SLMPC | 0.1298 | 0.0459 | 0.0413 |

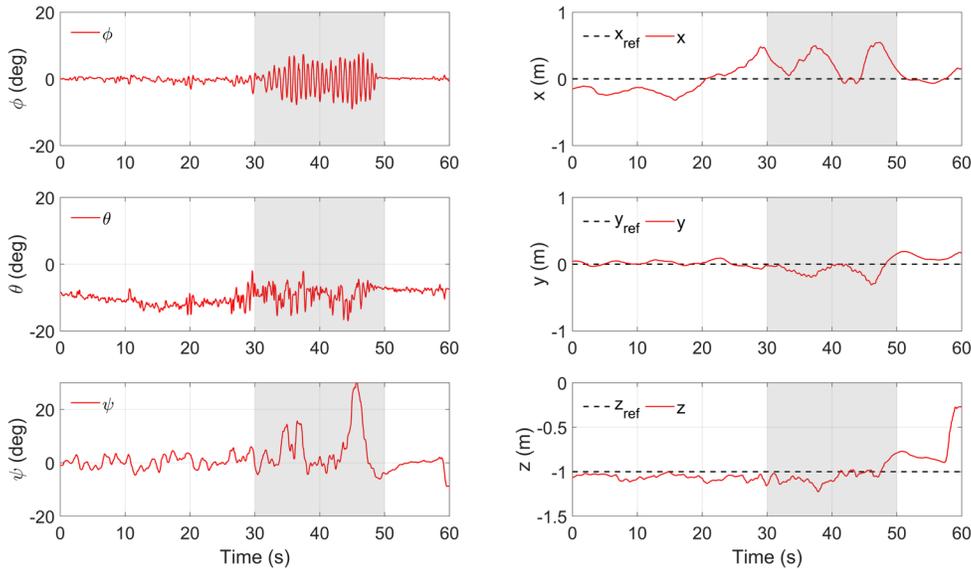


Fig. 14. Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the PID controller.

Case 2 (Linear MPC Control): The flight performance using a linear MPC position controller developed in the authors' previous work (Boyang Li, et al., 2018a) is presented in Fig. 15. Its performance in the x -direction is better than that of the previous PID controller, as it has a smaller variation of approximately 0.4 m and less peak movement. However, it does not control as effectively in the z -direction. There is a greater decrease in height and a more obvious oscillation in the z -direction. The attitude command signals also fluctuate, especially for the pitch (θ) command, fluctuating from $+10^\circ$ to -30° at approximately the 38^{th} second. As the linear MPC controller has its trim condition set around the hover point, which is 0° for all angles, any angular movement that is larger than 10° violates the predefined linear model and causes a model mismatch. However, the tendency to pitch down and then pitch up during windy conditions can be observed. From Table 2, the root mean square error values are smaller than those for the previous flight using the PID controller.

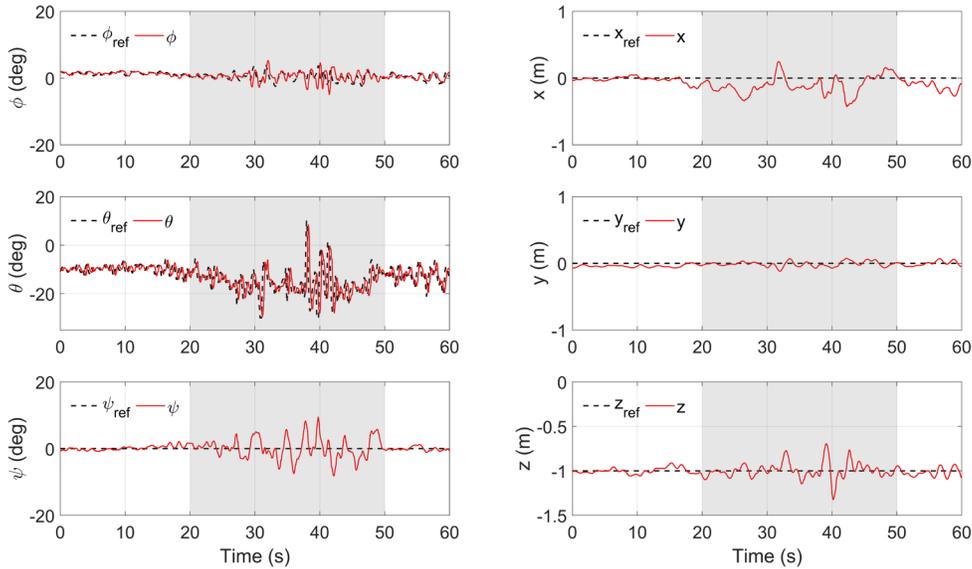


Fig. 15. Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the linear MPC controller.

Case 3 (Proposed SLMPC Control): The flight test results for the proposed SLMPC position controller are shown in Fig. 16. When the wind arrives, the pitch down command is clear and obvious. Compared with the linear MPC controller, the proposed controller shows a more stable and less oscillatory angular movement. The controls in the x - and z -directions are more precise. The consistent updating of the nominal states and model parameters prevents model mismatch for large angular movements. The yaw movement is mostly caused by the non-uniform wind field and the large moment arm of the wing. This effect occurs in all three cases but is much smaller for the MPC control (less than 10°) than the PID control (up to 30°). The root mean square error values of the position error under wind disturbance using the SLMPC controller are the smallest among all three tests, as shown in Table 2. The beauty of the SLMPC method lies in its ability to simultaneously optimise and adapt the real-time model to achieve minimum control effort and minimum model mismatch.

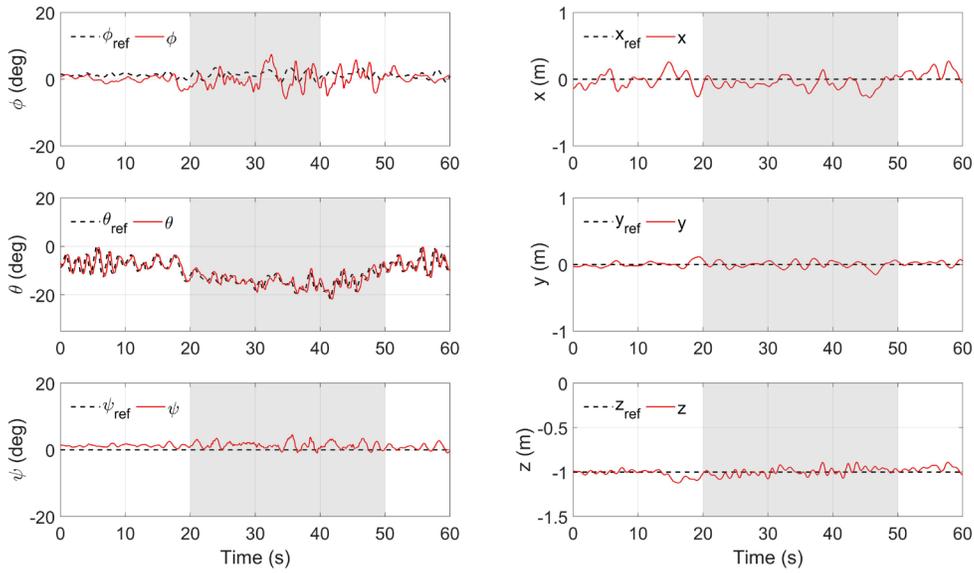


Fig. 16. Experimental results of attitude (left column) and position (right column) for the indoor UAV hovering tests under wind disturbance using the SLMPC controller.

6. Conclusion

In this study, a successive linearization based MPC control system is discussed and developed for a quad-rotor tail-sitter, which emphasizes the capability of wind disturbance rejection. The study started with the modelling of the tail-sitter VTOL vehicle that designed and built using a commercial off-of-shelf airframe, motors, propellers, flight controllers and other electrical instruments. Its dynamics and kinematics are described, and the propulsion system is modelled. The aerodynamic model is developed using an airfoil database to estimate the lift and drag on the wing under different operational conditions. A cascaded SLMPC controller using a prediction model, augmented with estimated disturbance and a feedback integration strategy is developed and the corresponding objective function is defined. Measures of time-varying weight and the velocity reference are taken to improve performance. The simulation and indoor flight test for disturbance rejection tasks compare a traditional PID controller, a linear MPC controller, and a SLMPC controller. The results show that the SLMPC has more precise hover position control in all three axes.

In this study, the information of the wind field is assumed to be available for the disturbance model of the MPC controller. In a future study, a wind field estimation method will be integrated into the control system to obtain more accurate wind information in real-time. Via real-time onboard measurement of the wind in different directions and at different magnitudes, outdoor flight experiments will be conducted in a natural environment of the prevailing wind and gusting wind conditions. Extension of the MPC controller to the transition and forward flight stages will also be conducted.

Acknowledgments

The authors acknowledge the financial support of the Innovation and Technology Commission of Hong Kong under grant number ITS/334/15FP and the Research Institute for Sustainable Urban Development of the Hong Kong Polytechnic University.

References

- Abdolhosseini M, Zhang YM, & Rabbath CA. An Efficient Model Predictive Control Scheme for an Unmanned Quadrotor Helicopter. *Journal of Intelligent & Robotic Systems* 2013;70:27-38.
- Alexis K, Papachristos C, Siegwart R, & Tzes A. Robust Model Predictive Flight Control of Unmanned Rotorcrafts. *Journal of Intelligent & Robotic Systems* 2016;81:443-469.
- Alexis K, Tzes A, & Nikolakopoulos G. Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications* 2012;6:1812-1827.
- An C, Zhang D, Zhang J, & Li T. A New structural configuration of tilting rotor unmanned aerial vehicle modeling. 2016 35th Chinese Control Conference (CCC). 2016. p. 2120-2125.
- Bangura M, & Mahony R. Real-time Model Predictive Control for Quadrotors. *IFAC Proceedings Volumes* 2014;47:11773-11780.
- Cannon M. Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control* 2004;28:229-237.
- Chikasha PN, & Dube C. Adaptive Model Predictive Control of a Quadrotor. *IFAC-PapersOnLine* 2017;50:157-162.
- Deihl Moritz HJF, Niels Haverbeke. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. *Nonlinear Model Predictive Control* 2009;Springer:391-417.
- Deng C, Wang S, Huang Z, Tan Z, & Liu J. Unmanned Aerial Vehicles for Power Line Inspection: A Cooperative Way in Platforms and Communications. *JCM*;9:687-692.
- Dong W, Gu G-Y, Zhu X, & Ding H. High-performance trajectory tracking control of a quadrotor with disturbance observer. *Sensors and Actuators A: Physical* 2014;211:67-77.
- Eduardo F. Camacho CBA. *Model Predictive Control*. 2 ed.: Springer-Verlag London; 2007.
- Eren U, Prach A, Koçer BB, Raković SV, Kayacan E, & Açıkmeşe B. Model Predictive Control in Aerospace Systems: Current State and Opportunities. *Journal of Guidance, Control, and Dynamics* 2017;40:1541-1566.
- Gregory IM, Ackerman K, Snyder S, & Rothhaar P. Adaptive control for tilt-wing VTOL UAV. 2015 American Control Conference (ACC). 2015. p. 2535-2535.
- Hochstenbach M, Notteboom C, Theys B, & De Schutter J. Design and Control of an Unmanned Aerial Vehicle for Autonomous Parcel Delivery with Transition from Vertical Take-off to Forward Flight – VertiKUL, a Quadcopter Tailsitter. *International Journal of Micro Air Vehicles* 2015;7:395-405.
- Li B. Model predictive hover control and transition optimization for a tail-sitter unmanned aerial vehicle. Dept of Mechanical Engineering. The Hong Kong Polytechnic University; 2018. p. 115.
- Li B, Jiang Y, Sun J, Cai L, & Wen CY. Development and Testing of a Two-UAV Communication Relay System. *Sensors (Basel)* 2016;16.
- Li B, Zhou W, Sun J, Wen C-Y, & Chen C-K. Development of Model Predictive Controller for a Tail-Sitter VTOL UAV in Hover Flight. *Sensors* 2018;18:2859.
- Li B, Zhou W, Sun J, Wen C, & Chen C. Model Predictive Control for Path Tracking of a VTOL Tailsitter UAV in an HIL Simulation Environment. 2018 AIAA Modeling and Simulation Technologies Conference. American Institute of Aeronautics and Astronautics; 2018.
- Lin CA, Shah K, Mauntel LCC, & Shah SA. Drone delivery of medications: Review of the landscape and legal considerations. *American Journal of Health-System Pharmacy* 2018;75:153-158.
- Liu C, Chen W-H, & Andrews J. Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers. *Control Engineering Practice* 2012;20:258-268.
- Ma D, Xia Y, Li T, & Chang K. Active disturbance rejection and predictive control strategy for a quadrotor helicopter. *IET Control Theory & Applications* 2016;10:2213-2222.
- Mayne DQ. Model predictive control: Recent developments and future promise. *Automatica* 2014;50:2967-2986.
- Oosedo A, Abiko S, Konno A, Koizumi T, Furui T, & Uchiyama M. Development of a quad rotor tail-sitter VTOL UAV without control surfaces and experimental verification. 2013.
- Oosedo A, Konno A, Matsumoto T, Go K, Masuko K, & Uchiyama M. Design and Attitude Control of a Quad-Rotor Tail-Sitter Vertical Takeoff and Landing Unmanned Aerial Vehicle. *Advanced Robotics* 2012;26:307-326.
- Ozdemir U, Aktas YO, Demirbag K, Erdem A, Kalaycıoğlu GD, Ozkol I, & Inalhan G. Design of a commercial hybrid VTOL UAV system. 2013 International Conference on Unmanned Aircraft Systems (ICUAS). 2013. p. 214-220.
- Penghui X. Aerodynamics of VTOL UAV. Mechanical Engineering. The Hong Kong Polytechnic University; July, 2017. p. 188.
- Peyrl H, Zanarini A, Besselmann T, Liu J, & Boéchat M-A. Parallel implementations of the fast gradient method for high-speed MPC. *Control Engineering Practice* 2014;33:22-34.
- Pfifflin JM, Binetti P, Souères P, Hamel T, & Trouchet D. Modeling and attitude control analysis of a ducted-fan micro aerial vehicle. *Control Engineering Practice* 2010;18:209-218.
- Rawlings JB, Mayne DQ, & Diehl MM. Disturbances and Zero Offset. *Model Predictive Control Theory and Design* 2012:47-49.
- Rossiter JA. *Model-Based Predictive Control: A Practical Approach*. CRC Press Control Series 2003:21-22.
- Shkarayev S, Moschetta JM, & Bataille B. Aerodynamic Design of Micro Air Vehicles for Vertical Flight. *Journal of Aircraft* 2008;45:1715-1724.
- Sinha P, Esden-Tempski P, Forrette CA, Gibboney JK, & Horn GM. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. 2012 IEEE Aerospace Conference. 2012. p. 1-17.

- Standardi L, Poulsen NK, Jørgensen JB, & Sokoler LE. Computational efficiency of economic MPC for power systems operation. IEEE PES ISGT Europe 2013. 2013. p. 1-5.
- Suardi A, Longo S, Kerrigan EC, & Constantinides GA. Explicit MPC: Hard constraint satisfaction under low precision arithmetic. *Control Engineering Practice* 2016;47:60-69.
- Sun J, Li B, Jiang Y, & Wen CY. A Camera-Based Target Detection and Positioning UAV System for Search and Rescue (SAR) Purposes. *Sensors (Basel)* 2016;16.
- Sun J, Li B, Shen L, Chen C-K, & 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.
- Sun J, Li B, Wen C-Y, & Chen C-K. Design and implementation of a real-time hardware-in-the-loop testing platform for a dual-rotor tail-sitter unmanned aerial vehicle. *Mechatronics* 2018;56:1-15.
- Wang X, Chen Z, & Yuan Z. Modeling and control of an agile tail-sitter aircraft. *Journal of the Franklin Institute* 2015;352:5437-5472.
- Wojcisz W, Gudaz J, Blevins T, & Mehta A. Practical approach to tuning MPC* *Based on Practical Approach to Tuning MPC by Wojcisz, Gudaz, Mehta, and Blevins, published in the Proceedings of the ISA 2001 Conference, September 10-13, 2001, Houston, TX [11]. *ISA Transactions* 2003;42:149-162.
- Zagrobely MA. MPC Performance Monitoring and Disturbance Model Identification. *Chemical Engineering*. UNIVERSITY OF WISCONSIN-MADISON; 2014. p. 71-76.
- Zhao S, Wang X, Zhang D, & Shen L. Model Predictive Control Based Integral Line-of-Sight Curved Path Following for Unmanned Aerial Vehicle. 2017.
- Zheng A. A computationally efficient nonlinear MPC algorithm. *Proceedings of the 1997 American Control Conference (Cat No97CH36041)*. 1997. p. 1623-1627 vol.1623.
- Zhu B, & Zuo Z. Adaptive model predictive control for set-point tracking of a single-DOF flapping-wing unmanned aerial vehicle. 2017 13th IEEE International Conference on Control & Automation (ICCA). 2017. p. 861-866.