Economic Model-Predictive Control for Aircraft Forced Landing: Framework and Two-Level Implementation

ZIHANG DONG
Imperial College London, London, U.K.

JINGJING JIANG
CUNJIA LIU
MATTHEW COOMBES
WEN-HUA CHEN, Fellow, IEEE
Loughborough University, Loughborough, U.K.

This article presents an optimization-based control framework for the autonomous forced landing of a fixed-wing unmanned aircraft (UA). A two-level model-predictive control (MPC) scheme is proposed to realize this framework, where an economic model-predictive control (EMPC) in a long piecewise constant fashion is proposed at the high level, while a short fixed-horizon linear time-varying MPC at the low level responds to fast dynamics of UA and tracks the optimal path provided by the high-level controller, alleviating computational burden compared to the high-frequency single-layer MPC scheme. Compared with a single EMPC setup with high sampling frequency, this hierarchical EMPC controller can significantly reduce the computational complexity and make it feasible to be implemented in real time. In addition, it also responds to disturbances (e.g., wind) and aircraft fast dynamics in a timely manner. The recursive feasibility and stability of the high- and low-level MPC schemes are established. The performance of the proposed EMPC forced landing function is illustrated by simulation case studies on both Aerosonde and Skywalker X8, compared favorably with competing schemes.

Manuscript received December 17, 2020; revised April 15, 2021 and July 15, 2021; released for publication August 27, 2021. Date of publication October 4, 2021; date of current version April 12, 2022.

DOI. No. 10.1109/TAES.2021.3117376

Refereeing of this contribution was handled by Y. B. Shhtessel.

This work was supported by the Engineering and Physical Sciences Research Council Established Career Fellowship under Grant EP/T005734/1.

Authors’ addresses: Zihang Dong is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K., E-mail: (zihang.dong14@imperial.ac.uk); Jingjing Jiang, Cunjia Liu, Matthew Coombes, and Wen-Hua Chen are with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, U.K., E-mail: (j.jiang2@lboro.ac.uk; c.liu5@lboro.ac.uk; M.J.Coombes@lboro.ac.uk; w.chen@lboro.ac.uk). (Corresponding author: Jingjing Jiang.)

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

I. INTRODUCTION

In recent years, unmanned aircraft (UA) has received considerable attention and development in both military and civilian applications due to its unique features: UA can significantly reduce the cost for training skilled pilots and guarantee human pilots’ safety for long-range remote sensing, delivery, and surveillance missions in dangerous areas.

However, without pilots, many safety issues arise. Among many others, an important consideration is how a UA deals with engine failure and other critical failures and carries out an emergency landing. As fixed-wing UA systems are primarily manufactured with a single engine, they are as vulnerable to lose power (due to engine failure) as any single-engine General Aviation aircraft [1]. Therefore,

NOMENCLATURE

PARAMETERS

\( X, Y, Z \) Position components [m].
\( V \) Velocity [m/s].
\( \alpha \) Angle of attack [rad].
\( \beta \) Side slip angle [rad].
\( \gamma \) Vertical flight path angle [rad].
\( \phi \) Roll angle [rad].
\( \theta \) Pitch angle [rad].
\( \psi \) Heading (yaw) angle [rad].
\( p \) Roll rate [rad/s].
\( q \) Pitch rate [rad/s].
\( r \) Yaw rate [rad/s].
\( L \) Force due to aerodynamic lift [N].
\( D \) Force due to aerodynamic drag [N].
\( M \) Aerodynamic pitching moment [kg \cdot m^2/s^2].
\( F_y \) Aerodynamic lateral force [N].
\( I \) Aerodynamic roll moment [kg \cdot m^2/s^2].
\( n \) Aerodynamic yaw moment [kg \cdot m^2/s^2].
\( \delta_a \) Control signal of the aileron deflection [rad].
\( \delta_e \) Control signal of the elevator deflection [rad].
\( \delta_r \) Control signal of the rudder deflection [rad].
\( m \) Mass [kg].
\( h \) Altitude [m].
\( b \) Wingspan [m].
\( S \) Wing area [m^2].
\( \omega_X \) Head/Tail wind [m/s].
\( \omega_Y \) Cross wind [m/s].
\( \omega_Z \) Vertical wind [m/s].
\( g \) Gravitational acceleration, 9.8 m/s^2.
\( \rho \) Density of air, 1.2682 kg/m^3.
\( e \) Oswald efficiency factor, 0.9.

COEFFICIENTS

\( C_l \) Aerodynamic lift force coefficient.
\( C_D \) Aerodynamic drag force coefficient.
\( C_M \) Coefficient of pitch moment.
\( C_l \) Aerodynamic lateral force coefficient.
\( C_{l} \) Coefficient of roll moment.
\( C_n \) Coefficient of yaw moment.
with an increased presence of UA over civilian populations and infrastructure assets, a reliable and robust system for forced landing is crucial in the deployment of UA systems to reduce the risk of casualties and asset losses. This is regarded as a bottleneck in the widespread applications of UA. Furthermore, autonomous forced landing systems can also be served as an advanced assistance function for human pilots in emergency situations.

The landing site selection and the associated reachability have been studied in previous research (see, e.g., [2]–[4]). A human forced landing approach technique named “high-key low-key” technique intending to give a human pilot good visibility to study the site and time to prepare the aircraft for landing has been introduced in [5], but this technique is not particularly suitable for UA especially in an environment with obstacles and low altitude. In [3] and [6]–[9], Dubins curve techniques, which would be reshaped to trochoids if the effects of uniform wind are taken into account, were developed for forced landing, where the height loss is minimized by flying the shortest path. However, these geometric curves might not be attainable for the aircraft as some dynamic behavior and constraints were not taken into account; for example, the desired bank angle was assumed to be achieved instantaneously. The authors of [10] and [11] assumed simplified kinematic modeling of the aircraft and studied the trajectory generation and optimization regardless of the analysis on how to design control actions to drive the aircraft. When the engine loses power, such kinematic models are not adequate to cope with maneuverability constraints imposed by loss of thrust.

Our objective is to find the optimal solution such that the aircraft reaches the target region above the landing site with minimum height loss. This will either increase the reachable range during the aircraft gliding or increase the safe margin for performing a landing exercise. To improve the autonomous landing performance of a fixed-wing UA, this article first proposes an optimal control framework by introducing the idea of economic model-predictive control (EMPC). Unlike many standard applications in industrial process control, which decompose the plant’s management and optimization into two levels (the real-time optimization (RTO) layer and the control layer), the EMPC looks into how to address economic optimization directly in real time (see [12]–[16] and references therein). The height loss of reaching a target region during the aircraft gliding or increase the safe margin for performing a landing exercise. To improve the autonomous landing performance of a fixed-wing UA, this article first proposes an optimal control framework by introducing the idea of economic model-predictive control (EMPC). Unlike many standard applications in industrial process control, which decompose the plant’s management and optimization into two levels (the real-time optimization (RTO) layer and the control layer), the EMPC looks into how to address economic optimization directly in real time (see [12]–[16] and references therein). The height loss of reaching a target region is selected as the only criterion and consequently used as the objective function in online optimization.

For applications like UA manoeuvring, in particular forced landing, the control objective is to reach a specified target region, in which it does not necessarily contain equilibrium points, while minimizing the height loss. This is different from the objective in [17], which is to recover a fixed-wing aircraft to a steady-state altitude from a deep stall using an EMPC controller. Hence, the notion of stability is compromised by that of completion [18]: once the target region is reached (at completion), the problem ends. In addition, for most applications of completion problems, it is desirable to have a finite completion time. Variable-horizon model-predictive control (MPC) [19], [20] and shrinking-horizon MPC [21], [22] have been shown to provide these features, and both types of MPC algorithms are able to relax the requirements on terminal ingredients\(^1\) that are central to the stability proofs. Since the variable horizon approach introduces an additional integer decision variable in the constrained optimization problem solved at each iteration, which might become intolerable for fast dynamic systems, the variable horizon is not employed in this article. In order to further alleviate the computational intensity of nonlinear optimization, a control framework employing a two-level control structure has been proposed in [25]. The high level adopted a piecewise constant input EMPC to generate baseline control commands by exploiting the nonlinear aircraft model, while the low level was designed based on linearized models around the reference trajectories provided by the high-level controller.

By integrating the advantages of shrinking horizon and piecewise constant control, this article proposes a two-level-optimization-based controller for a fixed-wing UA in forced landing applications. The main contributions of this article include the following:

1) For the first time, the forced landing problem is formulated in an EMPC framework, where the height loss and a target region around a forced landing site can be explicitly taken into account in the sense of optimal control. This study promotes a wider application of EMPC, particularly in aerospace sectors. Similar to the early development of MPC, most of the current applications of EMPC are in the process industry.

2) A two-level MPC scheme is proposed to implement this framework in trading off between many key factors, e.g., computational complexity, optimality, fast dynamics, and disturbance rejection. In this implementation, based on a simplified dynamic model, the high-level EMPC adopts shrinking horizon, piecewise constant control, and pure economic cost function so that it significantly reduces the workload of the online optimization; a fast linear time-varying MPC within a fixed horizon has been designed in order to reduce the effect of disturbances and uncertainties using its inherent robustness.\(^2\) With the optimal control actions determined by the two-level controller, numerical implementations have been performed on a high-fidelity six-degree-of-freedom aircraft model.

---

1Terminal ingredients include terminal constraint set, terminal cost, and local controller [23], [24].

2Inherent robustness refers to stability properties of a perturbed system in a closed loop with a nominal MPC design ignoring disturbances [24], [26], through repeated measurement. Meanwhile, this adoption of low-level MPC can also mitigate the demand of fast sampling frequency in high level while ensuring the reachability to the landing site.
3) Theoretic properties of the proposed hierarchical control scheme have been established, including recursive feasibility and closed-loop stability.

The rest of this article is organized as follows. Section II presents the mathematical model to describe the 3-D motion of a fixed-wing UA, the landing site with its associated target region, and the formal problem formulation. In Section III, the details of the proposed two-level MPC-based controllers are given, including the design of high-level shrinking horizon piecewise constant EMPC and low-level fixed horizon linear time-varying MPC. The closed-loop feasibility and stability results are analyzed in Section IV. Section V demonstrates the performance of the controller through illustrative examples. Finally, Section VI concludes this article.

II. EMPC FRAMEWORK FOR FORCED LANDING

A. Aircraft Modeling

This article aims to develop an optimal control framework for fixed-wing UA systems. Considering a right-handed coordinate system that coincides with the local east-north-up coordinates, specifically, we define the positive x-axis to be east, the positive y-axis to be north, and the positive z-axis to be up.

In the simulation, the aircraft dynamic model is described by high-dimensional nonlinear equations formulated in [27] and [28]. Nevertheless, to predict the long future aircraft trajectories within an acceptable computational complexity, a simplified aircraft dynamic model is employed to generate command guidance that drives the aircraft toward the desired region of a landing site. Specifically, the following set of equations [29]–[31] are used to describe the 3-D motion for a gliding aircraft over a flat Earth:

\[
\begin{align*}
\dot{X} &= V \cos \gamma \cos \psi \\
\dot{Y} &= V \cos \gamma \sin \psi \\
\dot{Z} &= V \sin \gamma \\
\dot{V} &= -\frac{D}{m} - g \sin \gamma \\
\dot{\gamma} &= \frac{L \cos \phi}{mV} - \frac{g}{V} \cos \gamma \\
\dot{\psi} &= \frac{L \sin \phi}{mV \cos \gamma} \\
\dot{\phi} &= \frac{C_L \alpha - C_D \phi}{k_\alpha}
\end{align*}
\]

(1)

where state vector \( x = [X \ Y \ Z \ V \ \gamma \ \psi \ \phi \ \alpha] \) and input command \( u = [\alpha \ \phi ] \). The state variables \( X, Y, \) and \( Z \) denote the position in a global coordinate system, \( V \) represents the airspeed of the aircraft, \( \gamma \) is the vertical path angle, \( \psi \) denotes the heading angle, \( \alpha \) is the angle of attack, and \( \phi \) is the roll angle. The control variables \( \bar{\alpha} \) and \( \bar{\phi} \) are the command angle of attack and roll angle, respectively. In particular, by adopting the techniques in [32] and [33], we employ the last two first-order differential equations in (1) to describe the dynamics of angle of attack and roll angle, relative to the piecewise constant input command \( u \), where \( k_{\alpha} \) and \( k_{\phi} \) are constant values. In short notation, the nonlinear representation is denoted by

\[
\dot{x}_t = f(x_t, u_t).
\]

(2)

Normally, the aerodynamic forces and moments in both longitudinal and lateral are determined by many factors. The full expression of aerodynamic functions is presented in the book [27]. For the dynamic model (1), the lift force \( L \) and drag force \( D \) are simplified, which are expressed as

\[
\begin{align*}
L &= \frac{1}{2} \rho SV^2 C_L(\alpha) \\
D &= \frac{1}{2} \rho SV^2 C_D(\alpha)
\end{align*}
\]

(3)

where \( C_L(\alpha) \) and \( C_D(\alpha) \) are nondimensional aerodynamic coefficients. For small angles of attack, these coefficients can be modeled with acceptable accuracy using linear approximations

\[
\begin{align*}
C_L(\alpha) &= C_{L_0} + C_{L_\alpha} \alpha \\
C_D(\alpha) &= C_{D_\phi} + \frac{(C_{L_0} + C_{L_\alpha} \alpha)^2}{\pi e \alpha R}
\end{align*}
\]

(4)

and \( \rho \) is the air density, \( S \) is the wing area, and \( AR = \frac{\gamma}{C} \) is the wing aspect ratio, with \( b \) denoting the wingspan.

The aircraft state variables are subject to physical and safety constraints that affect its maneuverability, which should be taken into account in the optimization problem. Mathematically, the feasible sets of system states and control inputs are denoted by \( x \in X \) and \( u \in U \), respectively. Letting \( x_t \) and \( u_t \) denote the instantaneous values of the state variable \( x \) and input variable \( u \) at time instant \( t \). Then, the (possibly coupled) pointwise-in-time state and input constraints to be considered are the following:

\[(x_t, u_t) \in Z \ \ \forall t \in \mathbb{R}_0 \]

(5)

where \( Z \subseteq X \times U \) is assumed to be compact. In particular, a realistic form of the aircraft constraints is explicitly formulated as

\[
\begin{align*}
\Phi(X, Y, Z) &\leq 0 \\
\alpha_{\text{min}} &\leq \alpha \leq \alpha_{\text{max}} \\
\psi_{\text{vert}} &\leq \psi \leq \psi_{\text{vert}} \\
V_{\text{min}} &\leq V \leq V_{\text{max}} \\
\gamma_{\text{min}} &\leq \gamma \leq \gamma_{\text{max}} \\
\alpha_{\text{min}} &\leq \alpha \leq \alpha_{\text{max}} \\
\phi_{\text{min}} &\leq \phi \leq \phi_{\text{max}} \\
\phi_{\text{min}} &\leq \phi \leq \phi_{\text{max}} \\
\phi_{\text{min}} &\leq \phi \leq \phi_{\text{max}}
\end{align*}
\]

(6)

where \( \psi_{\text{vert}} \) and \( \psi_{\text{horz}} \) are the vertical and horizontal structural load limitations of the acceleration components, respectively. The constraint \( \Phi(X, Y, Z) \leq 0 \) typically represents safe distance above ground. Moreover, it is worth to notice that the constraint on the angle of attack \( \alpha \) also relates to the accuracy of the aircraft model because the force parameters may be valid only in an interval of \( \alpha \).

B. Target Region

The objective of forced landing is to drive the aircraft to a specific destination. In the literature, quite often the destination landing site was simply regarded as a single
point which might not be appropriate in many practical scenarios. In practical applications, it is not necessary to ask the UA to land on a point. Instead, a corridor as shown in Fig. 1 is acceptable and so considered in this article.

To save height and design an achievable flight path, this article describes a predefined landing site in the 3-D space. Let us denote the two ending locations of the runway by

$$S_{\text{str}} = [X_{\text{str}} \ Y_{\text{str}} \ Z_{\text{str}}]^T$$

$$S_{\text{end}} = [X_{\text{end}} \ Y_{\text{end}} \ Z_{\text{end}}]^T$$

as shown in Fig. 1. Then, the destination can be expressed as a vector $S = S_{\text{end}} - S_{\text{str}}$. Accordingly, the horizontal angle of the landing site is calculated as

$$\theta_3 = \arctan \left( \frac{Y_{\text{end}} - Y_{\text{str}}}{X_{\text{end}} - X_{\text{str}}} \right).$$

Assuming that the required final straight-line landing distance is $L_s$ and the height should not be below $H_s$, the critical point $P_L = [X \ Y \ Z]^T$, where the aircraft targets to reach and starts the straight-line autoland process is

$$\begin{align*}
X &= X_{\text{str}} - L_s \cos(\theta_3) \\
Y &= Y_{\text{str}} - L_s \sin(\theta_3) \\
Z &= Z_{\text{str}} + H_s.
\end{align*}$$

In addition to the minimum required final height $H_s$, which is normally 500 ft or 152.4 m for the manned aircraft [34], before landing at the predefined landing site safely, the aircraft must be heading straight toward the runway with the heading angle $\psi = \theta_3$. Therefore, the targeted state for landing is $[X \ Y \ Z \ \psi]^T$.

In summary, the control objective of the forced landing and the height loss so that the aircraft has enough height to make turns in order to adjust its heading angle. Another important advantage of maintaining enough height is to preserve potential reachability to alternative landing sites in case of some unexpected emergencies. Therefore, the objective of the controller developed in this article is to maximize the terminal height $Z_N$.

To complete the forced landing task, the target region $X_f$ in Section II-B is adopted as the terminal region of the EMPC algorithm. In selecting a landing site, a reachability analysis using the trochoidal turn path in [3] and [8] or path planning techniques in [35]–[37] to any candidate landing site is conducted. Based on that, an estimated time
required to reach a candidate landing site can be calculated. According to the estimated time duration, the aircraft initiates its prediction horizon \( T = N \cdot T_d \) and starts the forced landing process from time instant \( t = 0 \). At each sampling time instant, a shrinking horizon EMPC at time \( t \in \mathbb{I}_{[0,N-1]} \), formulated as follows, is solved:

\[
\min_{u_{t:N}} - Z_{N|t} \quad (12)
\]

\[
\text{s.t.} \quad x_{t|t} = x_t, \quad (12a)
\]

\[
x_{k+1|t} = f_{T_d}(x_{k|t}, u_{k|t}) \quad (12b)
\]

\[
(x_{k|t}, u_{k|t}) \in \mathbb{Z}, \forall k \in \mathbb{I}_{[t,N-1]} \quad (12c)
\]

\[
x_{N|t} \in \mathcal{X}_f. \quad (12d)
\]

Note that, in this optimization problem, the number of decision variables \( u_{t:N} = [u_{t|t}, u_{t+1|t}, \ldots, u_{N|t}] \) are shrinking as the time instant \( t \) increases. After solving (12) at every time \( t \), following the standard MPC algorithm, only the first input element \( u_{t|t} \), as a constant value, is applied to control the aircraft over the time interval \( T_d \).

**Remark 1** In the standard hierarchical MPC algorithm, the traditional MPC controller is designed to track a path given by an RTO problem at the beginning of forced landing. To overcome the impact of uncertain wind along the path, many existing studies within the topic of wind estimate and compensation for small autonomous UA have been presented, e.g., disturbance observer [38], adaptive backstepping [39], vector field [40], etc. Since the primary objective of this article is to introduce the two-level EMPC framework, all of the formulations are deterministic, and combining the wind estimate techniques with the controller design is left for future research.

**Remark 2** The optimization problem (12) of “pure” economic nature is solved every \( T_d \) period and integrates the path planning and tracking problems. Constraint (12a) imposes the measured system state as the initial condition, and the future states are predicted according to constraint (12b). The pointwise-in-time constraint (12c) ensures the system feasibility within the prediction window. Moreover, the terminal constraint (12d) secures the aircraft to enter the desired terminal region \( \mathcal{X}_f \) at the end of the time horizon.

Although the EMPC controller generates a flyable trajectory between an original point (that includes the aircraft position, orientation, and velocity) and a destination runway while minimizing the loss in altitude, such a single layer method is prohibitive for real-time applications due to the required long prediction horizon and the nonlinearity of the optimization problem. For the practical application of operating an aircraft, a discretized model with a high sampling frequency has to be chosen to approximate the continuous-time dynamic system. If the sampling time interval \( T_d \) is selected to be sufficiently small, the MPC design based on the discretized model (11) can stabilize the original model (2). However, a smaller value of \( T_d \) will require a faster online optimization. To address the implementation issues of optimization problem (12), we aim to propose a two-level framework in the next section.

### III. TWO-LEVEL IMPLEMENTATION OF EMPC

This section presents an implementable two-level landing procedure that serves as a guideline for designing our autolanding algorithm. As shown in Fig. 2, the high-level EMPC planner adopts piecewise constant control actions to generate a feasible and optimal path for the aircraft at a lower sampling frequency. Then, a low-level time-varying MPC based on linearized models operates at a high sampling frequency to track the optimal path provided by the high-level EMPC planner.

#### A. High Level: Economic MPC Algorithm

We borrow the idea of piecewise constant MPC algorithm to yield a computationally efficient optimization problem [25]. A time interval \( T_h = \frac{H}{T_d} \) is introduced as the integration step for the high-level EMPC. In the conventional discrete-time MPC, \( T_s = T_d \), where \( T_d \) is the integration step for the lower-level design. Note that \( T_d \) is also the sampling time for the aircraft. However, in the piecewise constant MPC framework, the command inputs are held as constant values over several steps of \( T_d \) time period. For example, as shown in Fig. 3, the holding time duration \( T_s \) is calculated as \( T_s = H \cdot T_d \). In other words, for any prediction horizon \( N_h = \frac{H}{T_d} \), the control command generated by the high-level EMPC remains constant over \( H \) low-level MPC integration steps of \( T_d \), i.e., \( u_{k+1:H} = u_{k:H} \), \( \forall k \in \mathbb{I}_{[0,N_h-1]} \) and \( \forall i, j \in \mathbb{I}_{[0,H-1]} \).

Under the piecewise constant setup, the nonlinear discrete-time EMPC optimization problem subject to aircraft dynamics and various constraints at a high-level sampling time instant \( t = kH \) can be formulated as

\[
\min_{u_{t:N_h:H}} J_e(x_{t=N_h:H|t}, u_{t:(N_h-1):H|t}) \quad (13)
\]

\[
\text{s.t.} \quad x_{t|t} = x_t, \quad (13a)
\]

\[
x_{k+1|t} = f_{T_d}(x_{k|t}, u_{k|t}) \quad (13b)
\]

\[
(x_{k|t}, u_{k|t}) \in \mathbb{Z}, \quad (13c)
\]

\[
x_{N|t} \in \mathcal{X}_f. \quad (13d)
\]
with decision variables

\[ u_{t:(N_h-1)H} = \{ u_{t_1 H}, u_{t_1+2H}, \ldots, u_{t_1+(N_h-1)H} \} \]

and for all \( k \in \mathbb{Z}_{[0,N_h-1]}, \quad j \in \mathbb{Z}_{[0,H-1]} \) in (13b) and (13c).

The aim of the gliding stage is to maximize the terminal height of the aircraft when it enters the terminal region, so the objective function is defined as

\[ J_e(x_{t:(N_h-1)H}, u_{t:(N_h-1)H}) = -Z_{N_h H} \]  

(14)

**Remark 3** Note that the objective function (14) maximizes height only at the terminal state, which is intrinsically non-negative and fulfills certain properties useful for stability proof argument. Specifically, since the optimal height is directly adopted in the construction of a Lyapunov function, there exists a lower bound for the actual height of the aircraft when it enters the terminal region as in Theorem 1.

The optimization problem (13) is solved after every sampling interval \( T_d \) or equivalently \( H \) integration steps of \( T_d \), producing the optimal control sequence

\[ u^*_{t:(N_h-1)H} = \{ u^*_{t_1 H}, u^*_{t_1+2H}, \ldots, u^*_{t_1+(N_h-1)H} \} \]

and the corresponding state reference

\[ x^*_{t:N_h H} = \{ x^*_{t_1}, x^*_{t_1+H}, \ldots, x^*_{t_{N_h-1}H} \} \]

Following the standard MPC algorithm, only the first input element \( u^*_{t_1 H} \) and \( H \) states \( \{ x^*_{t_1}, x^*_{t_1+H}, \ldots, x^*_{t_{N_h-1}H} \} \) are sent to the low-level controller as the reference trajectories.

**B. Low Level: Linear Time-Varying MPC**

The proposed high-level EMPC scheme generates a baseline control by: 1) providing a longer time interval for solving the optimization problem via reducing the sampling rate and 2) reducing the number of variables to be optimized. However, the EMPC strategy becomes an open-loop optimal control within the interval \( T_s \). Unfortunately, due to the mismatch between the mathematical model used in the high-level EMPC and the real aircraft system, the noises, and disturbances in the process, this kind of optimal control might not perform as designed. Within the interval \( T_s \), the MPC cannot suppress any tracking error, so the bandwidth associated with the MPC may not be adequate for stabilizing and controlling the UA that have fast dynamics.

To overcome these difficulties in the implementation of EMPC with online optimization, a low-level controller has been adopted in the proposed framework. The high-level controller is the EMPC strategy described as before, which provides optimized state reference \( x^*_{t:(k+1)H-1} = x^*_{t:(k+1)H-1} | x_{t:(k+1)H-1} \) and the corresponding baseline control \( u^*_{t:(k+1)H-1} = u^*_{t:(k+1)H-1} | x_{t:(k+1)H-1} \) after every time interval \( T_s \), whereas the low-level controller is a linear feedback controller paralleled to EMPC that can provide stability around the optimized state reference in the presence of disturbances and uncertainties. The high-level controller runs at a large sampling time \( T_s \) due to the calculation time caused by solving nonlinear optimization problem (13). In contrast, the low-level controller works at a much higher sampling rate to reduce the impact of disturbances.

The low-level controller with sampling time interval \( T_d \) is designed based on the linearized aircraft model (11) around the reference state \( x^*_{t:(k+1)H-1} \) and control input \( u^*_{t:(k+1)H-1} \). The linearized system is expressed as

\[ \dot{x}_{t+1} = A(\delta) \tilde{x}_t + B(\delta) \tilde{u}_t \]

(15)

where \( \tilde{x}_t = x_t - x^*_t \) and \( \tilde{u}_t = u_t - u^*_t \) and \( A(\delta) = \frac{\partial f(x_t, u_t)}{\partial x} \mid_{(x^*_t, u^*_t)}, B(\delta) = \frac{\partial f(x_t, u_t)}{\partial u} \mid_{(x^*_t, u^*_t)} \), with \( \delta \) a parameter vector of the form \( \delta = [x^*_t, u^*_t]^T \), and it is known at every step. Correspondingly, the state-input constraints will be shifted and depend on \( \delta \), so the time-varying constraints for the error system (15) are denoted by

\[ \tilde{Z}(\delta) = \{ (\tilde{x}_t, \tilde{u}_t) \mid (\tilde{x}_t + x^*_t, \tilde{u}_t + u^*_t) \in \tilde{Z} \} \]

(16)

Letting \( \Delta \) represent the closed set containing all the possible values of \( \delta \), a specific model matrix pair \( (A(\delta), B(\delta)) \) is denoted by \( \lambda \in \Lambda \), where the set \( \Lambda \) is a multiplet description defined as

\[ \Lambda = \{ (A, B) \mid A = A(\delta), B = B(\delta), \delta \in \Delta \} \]

(17)

Note that the linear time-varying system (15) is dependent on \( x^*_{t:(k+1)H-1} \) and \( u^*_{t:(k+1)H-1} \), which are bounded in the high-level optimization. The parameter \( \delta \) is bounded by \( \Delta \), and each matrix pairs \( \lambda \in \Lambda \) is time invariant and depends on a known parameter \( \delta \).

Assume that a full measurement or estimate of the state \( x_t \) is available at the current time \( t \in \mathbb{Z}_{[0,H-1]} \). With a fixed prediction horizon \( N_\ell \in \mathbb{Z}_{\geq 0} \), we can formulate the following MPC problem:

\[ \min_{\tilde{u}_{t+1:N_\ell+1}^*} J_e(\tilde{x}_{t+1:N_\ell}^*, \tilde{u}_{t+1:N_\ell+1}^*) \]

(18)

subject to

\[ \tilde{x}_{t+j+1} = A(\delta_{t+j}) \tilde{x}_{t+j} + B(\delta_{t+j}) \tilde{u}_{t+j} \]

(18a)

\[ (\tilde{x}_{t+j+1}, \tilde{u}_{t+j}) \in \tilde{Z}(\delta_{t+j}) \forall j \in \mathbb{Z}_{[0,N_\ell-1]} \]

(18b)

\[ \tilde{x}_{t+N_\ell} \in \tilde{X}_{f} \]

(18d)

The objective function is to penalize the deviation of the predicted state and input sequences to the reference trajectories

\[ J_e(\tilde{x}_{t+1:N_\ell}^*, \tilde{u}_{t+1:N_\ell+1}^*) = \Psi(\tilde{x}_{t+N_\ell}^*) + \sum_{j=0}^{N_\ell-1} \tilde{x}_{t+j}^T Q \tilde{x}_{t+j} + \tilde{u}_{t+j}^T R \tilde{u}_{t+j} \]

where \( \Psi(\tilde{x}_{t+N_\ell}^*) \) represents the terminal cost at time \( t + N_\ell \) given the current sampling time instant \( t \). The cost function (18) is convex and the constraints (18a)–(18d) are linear. Therefore, it can be solved with efficient quadratic programming solvers. In the objective function, the matrices \( R \) and \( Q \) are positive definite to penalize the deviations from the reference input and state, respectively.

**Remark 4** According to the property of shrinking horizon EMPC, there are always solutions to optimization problem (13) in a certain environment if one solution can be found at the beginning. However, the low-level MPC may break down due to external disturbances. If this is the case, the
prediction horizon will be reinitialized, and an updated shrinking horizon high-level EMPC will proceed accordingly. In worse situations when the external disturbances are quite large, the high-level EMPC may become infeasible as well, indicating that the selected target region may be no longer reachable. Hence, the landing site selection algorithm [41] will be triggered and the UA will fly to a different destination.

Remark 5 Compared to an alternative scheme of the integration of high-level MPC and low-level proportional–integral–derivative (PID) tracking controller, the proposed two-level MPC framework solves the high-level EMPC optimization problem by means of piecewise-constant inputs at a low frequency and solves the low-level quadratic problem with a short prediction horizon. Therefore, it significantly reduces the computational load. More importantly, this article addresses the forced landing problem where the UA loses its power and operates in an extreme situation. The satisfaction of the constraints becomes more relevant and important, compared with normal aircraft operation with power. Therefore, the low-level controller, in addition to providing robustness for the high-level EMPC controller in the presence of uncertainties and disturbances, must respect all the physical and operational constraints while maximizing the maneuver capability. This is why we adopted a low-level MPC controller to track the references generated by the high-level EMPC. Finally, the employment of low-level MPC (rather than a simple PID tracking controller) provides a well-structured formulation and specification of stability analysis given in Theorem 2.

Let \( \bar{u}^*_{t:T} = \{ u^*_t, u^*_t + 1, u^*_t + 2, \ldots, u^*_t + N_t - 1 \} \) be the optimal solution of (18) at time \( t \). Following the fashion of standard MPC algorithm, the sum of the first element \( u^*_t \), from the low-level MPC and the reference input \( \bar{u}^*_t \), generated by the high-level EMPC is sent to the autopilot system to generate the real deflection control inputs, i.e., \( \delta_0, \delta_r \), and \( \delta_l \), as shown in Fig. 2. The proposed forced landing control scheme is then implemented to the six-degree-of-freedom dynamic system described in [27] and [28]. In the next sampling time, the optimal control problem (18) is solved again using the new state measurements.

When the model is linear time invariant, which corresponds to constant \( \delta \), a typical choice for \( \bar{X}_f \) is the maximal positive invariant set for the closed-loop system with state feedback control law \( u = Kx \), where \( K \) is the associated linear–quadratic regulator (LQR) gain (i.e., the unconstrained infinite time optimal controller gain). However, since the model (15) is linear time varying, there are several maximal positive invariant sets \( \bar{X}_f(\delta) \), and several different LQR feedback gain \( K(\delta) \), one for each \( \delta \in \Delta \) or equivalently \( \lambda \in \Lambda \). The maximal positive invariant set \( \bar{X}_f \) that is invariant for all \( \lambda \in \Lambda \) is explored in [42].

IV. FEASIBILITY AND STABILITY ANALYSIS

The stability of the EMPC employing Lyapunov function was first proved in [43] under the assumption of strong duality. Such a condition was subsequently relaxed by dissipativity in [12] and [13], where Amrit et al. have shown how to transform the economic cost to a standard positive-definite function and to choose the transformed cost-to-go as a candidate Lyapunov function. In analogy, this section explores the stability of the closed-loop nominal sequence under EMPC control actions.

For the particularly defined objective function (14) in the high level, we are able to claim our first main result.

Theorem 1 Suppose the optimization problem (13) has a solution at the beginning. Then, the high-level closed-loop system in the absence of disturbances using the proposed EMPC algorithm is recursively feasible, and the target region is stable in the sense that the aircraft reaches it in finite time. Moreover, letting that the optimization problem (13) has a solution at \( t = 0 \) and the predicted terminal height is \( Z_{dH|0} \), then the actual height \( Z_f \) when the aircraft enters the region is no worse than \( Z_{N_h H|0} \), viz., \( Z_f \geq Z_{N_h H|0} \).

Proof See Appendix A.

In the low-level tracking MPC with a fixed horizon, a tail term has to be added to the shifted input sequence to ensure recursive feasibility. The main idea is to find a local state feedback controller that stabilizes the system (15) inside an invariant terminal region \( \bar{X}_f \). Accordingly, some rigorous conditions of how to choose the terminal cost function \( \Psi(\cdot) \) and the terminal constraint set \( \bar{X}_f \) should be satisfied in order to ensure closed-loop stability and feasibility [23], [24]. In this respect, we make the following assumption:

Assumption 1 (see[23, Assumption 2.33]) There exists a terminal set \( \bar{X}_f \) including the origin as an interior point, a terminal control input \( u_f, \) and a terminal cost function \( \Psi(\cdot) \), such that for all \( x \in \bar{X}_f \) and \( \delta \in \Delta \), it holds that

\[
\begin{align*}
(i) & \quad (x, u_f) \in \tilde{Z}(\delta) \\
(ii) & \quad A(\delta)x + B(\delta)u_f \in \bar{X}_f \\
(iii) & \quad \Psi(A(\delta)x + B(\delta)u_f) - \Psi(x) \leq -x^TQx - u_f^TRu_f.
\end{align*}
\]

Remark 6 Assumption 1 preserves the corresponding basic stability conditions of [23, Assumption 2.33] for linear time-varying systems. Following the general approach in the MPC algorithm, we have selected a quadratic terminal cost function \( \Psi(x) = x^TPx \), where \( P \) is the state weight, and the terminal control policy \( u_f = K(\delta)x \). Then, Assumption 1(iii) can be written as

\[
A(\delta) + B(\delta)K(\delta))^T P(A(\delta) + B(\delta)K(\delta)) - P \leq -Q - K(\delta)^TRK(\delta).
\]

Notice that the above constraint implies that the quadratic function is a Lyapunov function, i.e., \( V(x) = x^TPx \); thus, the level sets of this Lyapunov function could be a choice of the terminal region \( \bar{X}_f \), and then, the other conditions in Assumption 1 are satisfied.
To simplify the calculation of terminal ingredients, one may consider a simple method that is to choose the following state equality as the terminal constraint (18d):

\[ \dot{x}_{t+N_t|t} = 0 \]  

(23)

which corresponds to \( \ddot{x}_f(\delta) = [0] \), \( K(\delta) = 0 \) and \( \Psi(\cdot) = 0 \) for all \( \delta \in \Delta \). This setup is applicable since the origin is an equilibrium point of the system (15) and trivially satisfies the terminal stability conditions in Assumption 1. However, the main drawback is also obvious that the constraints may cause the need of long horizon to obtain feasibility; otherwise, a control law with dead-beat behavior\(^3\) is necessary if the horizon is short.

We are now in the position to state the main result for the low-level control setup.

**THEOREM 2** Supposing that Assumption 1 holds, the low-level closed-loop system using time-varying MPC algorithm (18) is recursively feasible and stable.

**PROOF** See Appendix B.

From previous analysis, we know that if there is no uncertainty and disturbance, the aircraft always follows the trajectory obtained from the high-level EMPC controller until it reaches the target region. However, in reality, disturbances (such as winds) may deviate the aircraft from the reference trajectory. For a bounded disturbance along the reference trajectory, the error dynamic is governed by the linearized system (15). Let \( K_r \) denote the feedback gain, the control action is \( \bar{u}_r^* = K_r \ddot{x}_r \), and the resulting closed-loop system is expressed as

\[ \ddot{x}_{t+1} = (A(\delta_t) + B(\delta_t)K_r \ddot{x}_r + \omega_t. \]  

(24)

Note that the system (24) can be converted to a polytopic system with the vertices determined by the known boundary values of uncertainties, and the associated robust stability can be ensured by using the parameter-dependent Lyapunov function techniques [44], [45]. Moreover, the high-level EMPC resets the error state to zero every \( T_0 \) time interval because it synchronizes the real aircraft state as the new initial state of the future reference trajectory. Thus, the use of low-level MPC guarantees that the divergence between the real system state and the reference trajectory caused by a bounded disturbance \( \omega \) is bounded.

V. SIMULATION

Simulations were carried out to investigate the closed-loop performance of the proposed two-level EMPC forced landing system. The controller was tested on a high-fidelity aircraft model of Aerosonde, the dynamics of which are detailed in [27] and [28]. First, the advantage of using the two-level EMPC controller compared with following a Dubins curve is presented. Then, the forced landing maneuver of Aerosonde has been conducted in both no-wind and wind conditions to test the robustness of the proposed algorithm.

In addition, the performance of the closed-loop system with the proposed two-level EMPC controller targeting at different predefined landing sites has also been investigated. The comparison of our approach with the traditional control method (i.e., single-level EMPC with autopilot) demonstrating the effectiveness of our method against disturbances (i.e., winds). Finally, the generality of this algorithm has been validated via another type of aircraft, i.e., Skywalker X8 fixed-wing UA.

A. Performance of the Two-Level EMPC Scheme

Consider forced landing for an Aerosonde UA with the relevant parameters in Table I. The bounds in terms of both state and input constraints are \( \alpha_{\text{vert}}^\text{max} = 5 \text{ m/s}^2 \), \( \alpha_{\text{horz}}^\text{max} = -5 \text{ m/s}^2 \), \( \alpha_{\text{horz}}^\text{min} = -5 \text{ m/s}^2 \), \( V_{\text{max}} = 60 \text{ m/s} \), \( V_{\text{min}} = 15 \text{ m/s} \), \( \gamma_{\text{max}} = 20^\circ \), \( \gamma_{\text{min}} = -20^\circ \), \( \phi_{\text{max}} = 30^\circ \), \( \phi_{\text{min}} = -30^\circ \), \( \alpha_{\text{max}} = 20^\circ \), and \( \alpha_{\text{min}} = -20^\circ \). In addition, incremental restrictions are also imposed, e.g., \( |\dot{\omega}| \leq 5 \text{ /s} \) and \( |\phi| \leq 5 \text{ /s} \). The tolerance parameters of the target region \( X_f \) are defined as \( \varepsilon = 20^\circ \) and \( r = 100 \text{ m} \). The final landing distance \( L_S = 200 \text{ m} \). In all the simulation studies, the sampling time is chosen as \( T_d = 0.025 \text{ s} \).

Suppose the Aerosonde UA has an engine failure approximately 400 m away from an emergency landing site parameterized by \( S_{\text{sta}} = [6001000]^T \), \( S_{\text{end}} = [7001500]^T \). The initial conditions of this aircraft are specified in Table II. The time setting for high-level shrinking horizon piecewise constant EMPC is \( N_h = 15 \) and \( H = 40 \). This setup implies that the high-level EMPC sampling time is \( T_s = H \cdot T_d = 1 \text{ s} \) and the initial high-level prediction time duration is \( N_h \cdot T_s = 15 \text{ s} \). To cover the same prediction time window, the high-level prediction horizon \( N_h \) (without considering piecewise constant control) has to be increased from 15 to \( 600 \) steps, which is extremely large. For the low-level linear time-varying MPC, the state and input weights are \( Q = I \) and \( R = 0.01I \), where \( I \) is the identity matrix with proper dimensions. The low-level prediction horizon is selected as \( N_l = 20 \), which implies the corresponding time duration \( N_l \cdot T_d = 0.5 \text{ s} \). Moreover, the terminal equality constraint was adopted to avoid the specification of invariant sets and quadratic penalty functions.

---

\(^3\)Dead-beat behavior refers to a controller which can bring the system state to the equilibrium (origin) in a small number of steps.
Our control algorithm was implemented using MATLAB R2020a and solved by *fmincon* and *quadprog* functions, on a computer with two-core 3.50-GHz Intel(R) Xeon(R) E5-1650 processor and 32-GB RAM. For the high-level shrinking horizon EMPC, the longest computational time could occur in the first optimization problem with the longest prediction horizon (i.e., the most number of decision variables), which is 0.9884 s. Note that the computational time is also affected by the number of active constraints. Among all the optimization problems being solved at high-level sampling time instants, the problem at \( t = 3 \) s has the largest number of active constraints. Specifically, 65 out of 385 constraints are active. In this case, the computational time is 0.8330 s. Additionally, the computational time for the optimization problem at \( t = 10 \) without active constraints is 0.0735 s. Therefore, we can conclude that the high-level EMPC optimization problem can be solved within the sampling time interval \( T_s = 1 \) s.

Regarding the low-level tracking MPC with a fixed prediction horizon, it is a quadratic programming problem taking around 0.0025 s to find a solution, which is significantly smaller than \( T_d = 0.025 \) s. Moreover, further reduction in computation time for both levels can be easily obtained by increasing the corresponding sampling interval and utilizing more powerful machines.

Figs. 4 and 5 show the trajectories of the aircraft using the proposed two-level EMPC controller and tracking a Dubins curve. In Fig. 4, the critical point \( P_L = [421.1210.56152.40] \), which is represented by the blue dot. This critical point is connected to the target region by a purple dashed line, which represents the straight-line autoland- ing process before the aircraft can land on the yellow runway safely. The top view of the target area \( \mathcal{X}_{cyld} \) is encircled by the orange dotted line. The red solid curve is obtained by maximizing the terminal height, while the khaki solid one is obtained by following the black dash-dotted Dubins curve. Fig. 4 shows that the aircraft can complete the task and reach the target region by both methods. However, as can be seen in Fig. 5, the resulting terminal aircraft height using the proposed two-level EMPC is significantly higher than that following the Dubins curve. When the EMPC controller is used, the height loss is only about 10 m, whereas the Dubins method decreases the aircraft altitude by more than 150 m. Therefore, the EMPC method shows a great advantage in preserving aircraft height.

Then, to test the performance of the proposed controller in the presence of wind uncertainties, simulations in a no-wind environment and a “1-cos” deterministic wind environment [46] have been compared. The time histories of the wind on \( X, Y, \) and \( Z \) dimensions are shown in Fig. 6, where the negative sign means that the wind was blowing along the negative direction of the axis. Simulation results are displayed in Fig. 7.

According to Fig. 7, the aircraft heights in the wind environment (represented by the red line) are similar to that in a no-wind environment (represented by the blue curve) within the initial 4 s due to the relatively small accumulative wind impact. However, after 4 s, the aircraft states are increasingly dissimilar. Particularly, the magnitude of wind in the \( Z \)-direction is always negative, which is equivalent to a dragging force to the aircraft, so the height of the aircraft in the wind environment is lower. The negative values of wind in \( X \) and \( Y \) directions hinder the aircraft from the target region, so it costs an extra second for the aircraft to
reach it in the wind condition. Moreover, it is observed that the altitude of the aircraft increased between 5 and 10 s due to the velocity loss, while the aircraft turned toward the direction of landing site. In particular, the lifted altitude is higher at the expense of decreasing more speed when there is no wind. The final speeds of these two environment scenarios are almost the same, which implies that the aircraft has to spend more energy to fly against the wind as the associated final height is lower. The time histories of the aircraft’s angle of attack $\alpha$ and roll angle $\phi$ are shown in the bottom two subplots in Fig. 7, from which we can see that the two angles have similar patterns but different magnitudes when the aircraft was approaching to the target region because of the wind.

The time histories of deflection control sequences for closed-loop implementation are shown in Fig. 8. Given the command angles from the two-level controller, an autopilot system is used to decide the deflection angles corresponding to the command. Note that the deflection angles of the aircraft are bounded by $\pm 5^\circ$. The elevator deflection almost keeps at a constant after 5 s, which corresponds to the variation of height and velocity with a constant rate. The aileron and rudder deflections are used to adjust the lateral angles toward the target region.

B. Different Landing Sites

Four different landing sites have been selected to test the reachability of the aircraft from a common initial state. The parameters of each landing site are shown in Table III, and the corresponding flight paths are shown in Fig. 9, showing that the aircraft is able to reach the corresponding target region for all landing sites. This generalization on the reachability to multiple landing sites is useful when the control algorithm breaks down (see more details in Section V-D) for one landing site since it can direct the aircraft to other available ones.

C. Different Tracking Controller

The low-level controller can improve the performance of the high-level EMPC in the presence of uncertainties and disturbances. To evaluate the benefits of the two-level EMPC control framework, a comparison has been conducted with single-level EMPC integrated with a simple tracking controller, e.g., PID-based autopilot. For these two cases, the same wind as shown in Fig. 6 has been implemented. The resulting aircraft’s trajectories and deflection control inputs are shown in Figs. 10 and 11, respectively. As can be seen from the aircraft path view, the control method without low-level MPC cannot drive the aircraft into the target region. This is because autopilot in this case is tracking the piecewise constant command from high-level EMPC directly, rather than a command updated every $T_d$ s. Such comparison shows that adopting a low-level tracking MPC can update the command signal at a higher frequency.
and provide better robustness against the uncertain wind. Regarding the control inputs illustrated in Fig. 11, the two-level algorithm (represented by the blue line) is able to achieve better performance as the magnitudes of the deflection angles against the wind are smaller.

D. Different Aircraft Physical Parameters

Besides the Aerosonde model, the controller has also been implemented on a Skywalker X8 fixed-wing UA. Some basic parameters of this UA were collected from [47]. The X8 performed its forced landing starting from the origin to the same site as in Section V-A. Unlike Aerosonde, the X8 has a smaller size and the range of velocity is only between 10 and 20 m/s. The aircraft trajectories of these two types of UA under the zero wind condition are displayed in Fig. 12. Although the terminal height of X8 is lower, i.e., around 460 m, it can be controlled by the same two-level EMPC, demonstrating the wide applicability of the proposed control framework. Fig. 13 shows the airspeed and control surfaces of Aerosonde and X8. Note that the speed of X8 is almost constant at 15 m/s, and the angular deflections have more input oscillations compared to those of Aerosonde. This is because the X8 took turns at the beginning and near the end of its flight by adjusting the deflection angles more frequently to enter the terminal region with the desired heading angle while maximizing the height. Moreover, due to the relatively low velocity, the X8 has to spend more time to reach the target region.

VI. CONCLUSION

In this article, a novel two-level EMPC scheme is developed for fixed-wing UA systems in order to accomplish a successful emergency landing when the aircraft loses power. The proposed scheme integrates the high-level EMPC method, which aims to minimize the height loss, and the low-level time-varying tracking MPC, so that the closed-loop feasibility and stability are guaranteed. The effectiveness of the proposed approach has been demonstrated by several case studies. In particular, the robustness and generality of the proposed controller on the aircraft flight have been tested and analyzed. The benefits of using a two-level MPC framework, rather than the traditional high-level MPC planner with low-level simple tracking controller, have also
been investigated. In the future, we will combine the control algorithm with wind estimation techniques and investigate how to take advantage of favorable winds.

APPENDIX A
PROOF OF THEOREM 1

PROOF

1) Recursive feasibility: The optimization (13) is solved at time instants \( t = kH, \forall k \in [0, N_h - 1] \). To prove recursive feasibility, one can show that it is always possible to construct a feasible solution for the optimization problems with \( k \in [1, N_h - 1] \). Given the optimal solution \( u^*_k \) from the optimization at \( t = (k - 1)H \), there is a feasible control sequence at time instant \( kH \)

\[
\mathbf{u}_{kH:(N_h-1)H} = [\mathbf{u}^*_k H:\mathbf{u}^*_{k+1H:(N_h-1)H}] \tag{25}
\]

Thus, the recursive feasibility of the optimization problem (13) is guaranteed.

2) Stability: The proof of stability follows similar lines as in [18, proof of proposition]. We introduce an auxiliary variable, \( \chi \in [0, 1] \), where \( \chi = 1 \) represents the region \( \mathcal{X}_f \) that has not been reached and \( \chi = 0 \) indicates that the region \( \mathcal{X}_f \) has been reached at the current step or previous ones. At time instant \( t \), consider the modified cost function

\[
\tilde{J}_c(x_{t:N_h H}^*, \mathbf{u}_{t:(N_h-1)H}) = \tilde{Z} - Z_{N_h H}^* + \sum_{k=0}^{N_h-1} \chi_{kH}\] (26)

subject to constraints (13a)--(13d), where \( \tilde{Z} \) is a large positive value such that \( \tilde{Z} - Z_{N_h H}^* \) is always positive. Then, this modified problem admits the same optimal solution as the optimization problem (13) as their objective functions differ by a constant. Let us denote the optimal value of the modified objective function by

\[
V_c(x_t) = \tilde{J}_c(x_{t:N_h H}^*, \mathbf{u}_{t:(N_h-1)H}) \tag{27}
\]

When there is no mismatch between the model and the real plant and in the absence of disturbances, the measured state \( x_{t+H} \) at time instant \( t + H \) equals to the state predicted at instant \( t \). Hence, when the system proceeds to time instant \( t + H \), the initial control defined by (25) is feasible, and the corresponding performance index is

\[
\tilde{J}_c(x_{t+H:N_h H}^*, \mathbf{u}_{t+H:(N_h-1)H}) = V_c(x_t) - \chi_{t+H H} = \tilde{Z} - Z_{N_h H}^* + \sum_{k=2}^H \chi_{kH}\] (27)

where \( Z_{N_h H}^* \) is the optimal terminal altitude of the aircraft.

Note that at time instant \( t + H \), the online optimization will be performed to minimize the cost function initialized by (27). Thereby, by optimality, the resulting performance index implies

\[
V_c(x_{t+H}) \leq \tilde{J}_c(x_{t+H:N_h H}^*, \mathbf{u}_{t+H:(N_h-1)H}) = V_c(x_t) - \chi_{t+H H}.
\]

This indicates that the cost \( V_c \) decreases by at least one per sampling time as \( \chi_{t+H} = 1 \). Moreover, since \( V_c(x_t) \) is positive by construction, it can be adopted as a Lyapunov function for the high-level EMPC algorithm. In addition, \( \chi_t \to 0 \) indicating that the target region has been reached.

Furthermore, let us recall that the solution (25) obtained from the optimal solution at \( t = kH \) provides a suboptimal control sequence for the next sampling time instant \( t = (k + 1)H \), so the optimal value function \( J_c \) fulfills

\[
J_c(x_{(k+1)H:N_h H}^*(k+1)H), \mathbf{u}_{(k+1)H:(N_h-1)H}(k+1)H) \leq J_c(x_{(k)H:N_h H}^*(k)H), \mathbf{u}_{(k)H:(N_h-1)H}(k)H) \]

or equivalently

\[
Z_{N_h H}(k+1)H \geq Z_{N_h H}(k)H \]

which implies that the predicted final altitude is non-decreasing. Therefore, once the aircraft reaches the region and the prediction horizon shrinks to zero, the final height \( Z_f \) satisfies the inequality \( Z_f \geq Z_{N_h H}(0) \), where \( Z_{N_h H}(0) \) is the predicted final altitude at time instant \( t = 0 \).

APPENDIX B
PROOF OF THEOREM 2

The proof of Theorem 2 follows the similar line of that in Appendix A and the standard method of stability proof for time-varying systems in [23]. The difference is that for the fixed horizon algorithm, an additional tail term is necessary to ensure recursive feasibility. Due to the conditions (i) and (ii) in Assumption 1, recursive feasibility is ensured.

Next, we show the optimal value function

\[
V_c(x_t) = J_c(x_{t:N_h H}^*, \mathbf{u}_{t:(N_h-1)H})
\]

is a Lyapunov function and strictly decreasing. When the system proceeds from time \( t \) to \( t + 1 \), the sequences \( x_{(t+1)H}^* \) and \( u_{(t+1)H}^* \) are feasible for the optimization problem at instant \( t + 1 \); therefore, by optimality, it yields that

\[
V_c(x_{t+1}) \leq J_c(x_{(t+1)H:N_h H}^*(t+1)H), \mathbf{u}_{(t+1)H:(N_h-1)H}(t+1)H) + (K(\delta_{t+1}^N \hat{x}_{t+1}^N)')^T R(K(\delta_{t+1}^N \hat{x}_{t+1}^N)
\]

and the prediction horizon shrinks to zero, the final height \( Z_f \) satisfies the inequality \( Z_f \geq Z_{N_h H}(0) \), where \( Z_{N_h H}(0) \) is the predicted final altitude at time instant \( t = 0 \).
Jingjing Jiang received the B.Eng. degree in electronic and electrical engineering from the University of Birmingham, Birmingham, U.K., and the B.Eng. degree in measurement, control technology and instrument from the Harbin Institute of Technology, Harbin, China, in 2010, and the M.Sc. degree in control systems and the Ph.D. degree with a thesis on shared control for systems with constraints from Imperial College London, London, U.K., in 2011 and 2016, respectively. Since September 2018, she has been a Lecturer of Intelligent Mobility/Autonomous Vehicles with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K. After completing the Ph.D. degree, she was appointed as a Research Associate with the Control and Power Group, Imperial College London. She has authored or coauthored 30 papers in journals and conferences. Her research interests include control design of intelligent systems with constraints, human in the loop, driver assistance control, and autonomous systems.

Cunjia Liu received the Ph.D. degree in autonomous vehicle control from Loughborough University, Loughborough, U.K., in 2011. He was a Research Associate with the Department of Aeronautical and Automotive Engineering, Loughborough University, where he is currently a Reader in Unmanned Vehicles. He has a strong background in flight control, autonomous vehicles, and Bayesian estimation. His recent work focuses on the novel perception and decision algorithms for autonomous vehicles and their applications in security, industry, and agriculture domains.

Matthew Coombes is a Lecturer of Unmanned Vehicles with Loughborough University, Loughborough, U.K. He has considerable experience in developing and operating autonomous platforms, including system and control design, sensor integration, and testing. He has research experience in a number of areas related to autonomous vehicles. Some of which include unmanned aerial system (UAS) forced landing, situational awareness, autonomous taxiing, mission planning optimization for agricultural aerial surveying and crop spraying as well as sensor fusion and tracking for Counter-UAS. He has recently been awarded with projects from DASA and innovateUK on Counter UAS and Drone inspection totaling more than £250 K. He has overseen all laboratory practical activities for the Loughborough University Centre for Autonomous Systems since 2014.

Wen-Hua Chen (Fellow, IEEE) received the M.Sc. and Ph.D. degrees from the Department of Automatic Control, Northeast University, Shenyang, China, in 1989 and 1991, respectively. He is currently a Professor of Autonomous Vehicles with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K. From 1991 to 1996, he was a Lecturer with the Department of Automatic Control, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He held a research position and then a Lectureship in control engineering with the Centre for Systems and Control, University of Glasgow, Glasgow, U.K., from 1997 to 2000. He has authored or coauthored three books and 300 papers in journals and conferences. His research interests include the development of advanced control strategies and their applications in aerospace engineering, particularly in unmanned aircraft systems.

Dr. Chen is a fellow of the Institution of Engineering and Technology and the Institution of Mechanical Engineers. He is a recipient of the prestigious Engineering and Physical Sciences Research Council Established Career Fellowship Award.

Zhihang Dong received the Ph.D. degree in control systems from Imperial College London, London, U.K., in 2019. In 2019, he was a Part-time Research Associate with Loughborough University, Loughborough, U.K. He is currently a Research Associate with the Department of Electrical and Electronic Engineering, Imperial College London. His research interests include robust economic model-predictive control, flexible demand response in power systems, and the design of control strategies for the smart grids.