DOI: 10.13009/EUCASS2019-453

8TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS)

DOI: ADD DOINUMBER HERE

Iterative Learning Control for Precise Aircraft Trajectory Tracking in Continuous Descent Approaches

Almudena Buelta†, Alberto Olivares, Ernesto Staffetti
Universidad Rey Juan Carlos
Fuenlabrada, Madrid, Spain
almudenajose.buelta@urjc.es· alberto.olivares@urjc.es · ernesto.staffetti@urjc.es
†Corresponding author

Abstract

In this study, an iterative learning control method for precise aircraft trajectory tracking is presented. Given a trajectory to be followed by an aircraft whose dynamical model is known, the proposed algorithm improves the system performance in following the trajectory anticipating recurring disturbances and proactively compensating for them, using the spatial and temporal deviations suffered by previous flights. The results presented show a significant reduction of the trajectory tracking error in few iterations for a type of operations in which precision is essential: continuous descent approaches. The results prove the effectiveness of this method applied to commercial aircraft trajectory tracking.

1. Introduction

Global air traffic growth in both volume and complexity requires the modernization of the Air Traffic Management (ATM) system. The Single European Sky ATM Research[1] (SESAR) and other international initiatives work to develop an optimization framework for trajectory-based operations, which assigns four-dimensional (4D) trajectories to flights based on the stakeholders' preferences and priorities, improving flight efficiency and reducing environmental impact while also meeting the expected demands for increased capacity. A 4D trajectory consists of the three spatial dimensions of the aircraft path and the additional component of time, meaning that any delay is considered as a distortion of the trajectory as much as a deviation in the position of the aircraft. The designed 4D trajectories must be precisely followed by the aircraft to avoid conflicts with other aircraft and ensure the safety of the flight and an efficient exploitation of the airspace. However, mainly because of wind and temperature forecast errors or unpredictable weather events such as storms, some level of uncertainty will remain, resulting in deviations that cannot be compensated by the usual aircraft trajectory tracking controllers, which react to noise and unexpected disturbances as they occur [1].

The method here exposed addresses this problem by using an Iterative Learning Control (ILC) scheme which is able to improve the precision of the aircraft in following the trajectory taking into account the deviations suffered by previous flights. This method is especially suitable for air spaces with high traffic density, where time between consecutive flights is short enough to assume similar atmospheric conditions and therefore similar disturbances. This typically occurs in busy air spaces such as terminal maneuvering areas in arrival and departure procedures, in which precision in trajectory tracking is crucial. In particular, the ILC paradigm has been studied for Continuous Descent Approaches (CDAs). CDA operations are embraced in the general term Continuous Descent Operations (CDO), in which aircraft descend from the cruise altitude to the final approach fix at or near idle thrust, ideally in a low drag configuration, without level segments at low altitude, minimizing the need for high thrust levels to remain at a constant altitude and reducing the environmental impact. According to [2], the terms CDO and CDA are interchangeable and should be read and understood in the same context. This case is of special interest because for an effective CDA implementation, accurate trajectory tracking control schemes must be employed to avoid potential conflicts between the arriving and departing aircraft together with appropriate airspace design and effective Air Traffic Control (ATC) procedures. A CDA may be part of a Standard Terminal Arrival Route (STAR) procedure, in which case it is known as Optimized Profile Descent (OPD) [3].

Information sharing is key to the application of ILC to precise aircraft trajectory tracking, since this method is based on using data from previous aircraft to minimize trajectory deviations of the following flights. This information

[1] https://www.sesarju.eu/

DOI: 10.13009/EUCASS2019-453

is not yet available, though this obstacle will be overcome with the System Wide Information Management (SWIM), which is part of the ICAO Global Air Navigation Plan [4], and whose purpose is to perform a common platform where aircraft will be fully connected as an information node, enabling full participation in collaborative ATM processes with exchange of data, including meteorology and the detailed route of the aircraft defined in 4D. In this paper, intended and actually flown trajectories of previous aircraft are assumed to be available.

ILC is based on the idea that the performance of a control system that executes the same task multiple times can be improved by learning from previous executions. In particular, precision in the execution of a task can be improved by incorporating error information into the control law for subsequent iterations. In doing so, accurate performance can be achieved with low transient tracking error despite recurrent disturbances. ILC paradigm emerged from industrial robotics applications to improve trajectory tracking precision of robot manipulators in repetitive tasks [5], and has been widely applied over the last three decades. In [6], the main results in ILC analysis and design are surveyed. Several textbooks on ILC are available, such as [7], which treats both ILC for linear and nonlinear systems, and [8], which focuses on real time applications. ILC has recently been applied to precise trajectory tracking of aerial robots. In [9], an ILC approach has been applied to precise quadrocopter trajectory tracking. This work can be viewed as an extension of the results presented in [10], where a least-squares based learning rule was proposed to perform aggressive maneuvers with quadcopters, which consist in steering these systems quickly from one state to another. Other ILC paradigms have been applied to trajectory tracking for Unmanned Aerial Vehicles (UAV). See for example [11] and references therein. To the best knowledge of the authors, ILC has not been applied yet to precise aircraft trajectory tracking.

In this paper, the possibility of applying the optimization-based ILC method presented in [9] to improve precision in aircraft trajectory tracking has been investigated. This ILC scheme requires the system dynamics to be repetition-invariant, which means that the same dynamical system (in this context, the aircraft) executes the same task (flying the same planned trajectory) in each iteration. In a more realistic scenario, consecutive flights along the same path are in general carried out by different aircraft and follow different trajectories. The knowledge transfer problem among dynamical systems and among trajectories is not addressed in this paper due to its complexity, and will be subject of future research. The ILC problem is solved in two steps, both relying on a nominal model of the aircraft, in which input and state constraints are explicitly taken into account. The first step consists in the estimation of the model error and recurrent disturbances affecting the flight of an aircraft along a trajectory using a time-varying Kalman filter. In the second step, optimization techniques are employed to determine an updated control input to optimally compensate for the recurrent disturbances of the following aircraft trajectory tracking. An advantage of using the ILC method is that it can be made non intrusive with respect to the trajectory tracking control system of the aircraft, since a new reference trajectory can be computed as part of the input update and be fed into the flight management system of the following aircraft. This way, the feedback trajectory tracking control reduces non-repetitive disturbances while the ILC is intended to reject repetitive disturbances. The ILC method has been implemented in the MATLAB/Simulink platform, a graphical programming environment for modeling and simulating dynamical systems developed by MathWorks[2].

In the experiments presented in this paper, the CDA trajectories to be followed have been generated using optimal control techniques in which the actual dynamical model of the aircraft has been taken into account, ensuring the feasibility of the planned trajectories. The resulting optimal control problems have been solved using DIDO, a MATLAB application based on the pseudospectral method developed by Elissar Global[3]. This application, besides the optimal control and the corresponding optimal trajectory, returns the Hamiltonian, costates, path covectors, and endpoint covectors. This information, together with classical tools such as Pontryagin's Maximum Principle, is essential to verify the optimality of the numerical results.

The experiments and results have been carried out in a simulated environment. A realistic scenario has been built in the MATLAB/Simulink platform using commercial aircraft data from the Base of Aircraft Data (BADA), which is an Aircraft Performance Model (APM) developed and maintained by EUROCONTROL[4] with the collaboration of aircraft manufacturers and airlines, specifically designed for simulation and prediction of aircraft trajectories for research and operations in ATM. The BADA APM has two components: model specifications, which provide the theoretical models used to calculate aircraft performance parameters, and data sets, which give the aircraft-specific coefficients. There are two families of the BADA APM, based on the same modeling approach and with the same components: BADA Family 3 and BADA Family 4. The latest release of the first one has been used in this paper [12].

The application of the ILC technique to precise aircraft trajectory tracking would be an innovative solution to increase predictability of trajectories in the future trajectory-based operations paradigm. Higher precision in trajectory tracking implies an improvement of the aircraft performance predictability and therefore of the air traffic management system capacity. Airlines can also benefit from this higher predictability by reducing the number of alterations in following their designed trajectories, which entails a reduction of costs and emissions.

[2] https://es.mathworks.com/
[3] http://www.elissarglobal.com/industry/products/
[4] https://www-test.eurocontrol.int/services/bada

DOI: 10.13009/EUCASS2019-453

The paper is organized as follows. The model of the aircraft dynamics is derived in Section 2. The general ILC scheme is presented in Section 3 and the method for generating the CDA reference trajectory is presented in Section 4. The experimental setup is described in Section 5 and the results of the application of the ILC algorithm to aircraft trajectory tracking in Section 6. Finally, Section 7 contains the conclusions.

2. Aircraft dynamics

In this section, the dynamic model of the aircraft used in the simulated environment will be described.

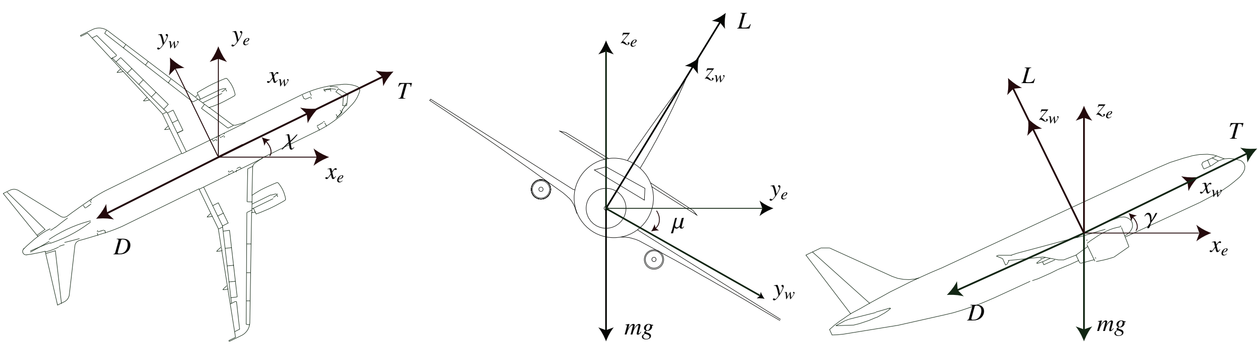
2.1 Equations of Motion

A common three-degree-of-freedom dynamic model has been used which describes the point variable-mass motion of the aircraft over a non-rotating flat Earth model [13]. In particular, a symmetric flight has been considered. Thus, it has been assumed that there is no sideslip and all forces lie in the plane of symmetry of the aircraft. The following equations of motion of the aircraft have been considered:

$$\begin{aligned}
\dot{V}(t) &= \frac{T(t) - D(h_e(t), V(t), C_L(t)) - m(t)\cdot g\cdot\sin\gamma(t)}{m(t)}\\
\dot{\chi}(t) &= \frac{L(h_e(t), V(t), C_L(t))\cdot\sin\mu(t)}{m(t)\cdot V(t)\cdot\cos\gamma(t)}\\
\dot{\gamma}(t) &= \frac{L(h_e(t), V(t), C_L(t))\cdot\cos\mu(t) - m(t)\cdot g\cdot\cos\gamma(t)}{m(t)\cdot V(t)}\\
\dot{x}_e(t) &= V(t)\cdot\cos\gamma(t)\cdot\cos\chi(t)\\
\dot{y}_e(t) &= V(t)\cdot\cos\gamma(t)\cdot\sin\chi(t)\\
\dot{h}_e(t) &= V(t)\cdot\sin\gamma(t)\\
\dot{m}(t) &= -T(t)\cdot\eta(V(t)).
\end{aligned} \tag{1}$$

The three dynamic equations in (1) are expressed in an aircraft-attached reference frame, the wind axes (x_w, y_w, z_w), and the three kinematic equations are expressed in a ground based reference frame, the Earth reference frame (x_e, y_e, z_e), as shown in Fig. 1.

The states of the system (1) are the true airspeed, V, the heading angle, χ, the flight path angle, γ, the position, x_e, y_e, h_e, and the aircraft mass, m. Thus, $\mathrm{x}(t) = (V(t), \chi(t), \gamma(t), \lambda_e(t), \theta_e(t), h_e(t), m(t))$. The control inputs are the bank angle, μ, the engine thrust, T, and the lift coefficient, C_L. Thus, $\mathrm{u}(t) = (T(t), \mu(t), C_L(t))$. Parameter η is the speed-dependent fuel efficiency coefficient. Lift, $L = C_L S\hat{q}$, and drag, $D = C_D S\hat{q}$, are the components of the aerodynamic force. Parameter S is the reference wing surface area and $\hat{q} = \frac{1}{2}\rho V^2$ is the dynamic pressure. A parabolic drag polar $C_D = C_{D0} + KC_L^2$, and an International Standard Atmosphere (ISA) model are assumed. The lift coefficient C_L is a known function of the angle of attack α and the Mach number.



(a) Top view (b) Front view (c) Lateral view

Figure 1: Aircraft state and forces.

DOI: 10.13009/EUCASS2019-453

2.2 Flight Envelope Constraints

Flight envelope constraints are derived from the geometry of the aircraft, structural limitations, engine power, and aerodynamic characteristics. The performance limitations model and the parameters have been obtained from BADA.

$$\begin{aligned} &0 \le h_e(t) \le \min[h_{M0}, h_u(t)], && \gamma_{min} \le \gamma(t) \le \gamma_{max}, \\ &M(t) \le M_{M0}, && m_{min} \le m(t) \le m_{max}, \\ &\dot{V}(t) \le \bar{a}_l, && C_v V_s(t) \le V(t) \le V_{Mo}, \\ &\dot{\gamma}(t) V(t) \le \bar{a}_n, && 0 \le C_L(t) \le C_{L_{max}}, \\ &T_{min}(t) \le T(t) \le T_{max}(t), && \mu(t) \le \bar{\mu}. \end{aligned} \tag{2}$$

In (2), h_{M0} is the maximum operational altitude and $h_u(t)$ is the maximum operative altitude at a given mass (it increases as fuel is burned). $M(t)$ is the Mach number and M_{M_0} is the maximum operating Mach number. C_v is the minimum speed coefficient, $V_s(t)$ is the stall speed, V_{M_0} is the maximum operating Calibrated Airspeed (CAS) and $\bar{a}_n$ and $\bar{a}_l$ are, respectively, the maximum normal and longitudinal accelerations for civilian aircraft. Finally, $T_{min}(t)$ and $T_{max}(t)$ correspond to the minimum and maximum available thrust, respectively, and $\bar{\mu}$ corresponds to the maximum bank angle due to structural limitations.

2.3 Longitudinal dynamics

In this work, ILC is used for precise trajectory tracking of the vertical profile of a CDA. Therefore, the motion of the aircraft is limited to a vertical plane, i.e., with constant course and thus constant heading angle χ. Without loss of generality, we assume that the heading angle is zero, that is $\chi = 0$. We also suppose that the aircraft performs a leveled wing flight, thus the bank angle is zero, that is $\mu = 0$.
The state variables are then

$$\mathrm{x}(t) = (V(t), \chi(t), \gamma(t), \lambda_e(t), \theta_e(t), h_e(t), m(t))$$

and the control variables,

$$\mathrm{u}(t) = (T(t), C_L(t)).$$

The equations of motion are reduced to:

$$\begin{aligned} \dot{V}(t) &= \frac{T(t) - D(h_e(t), V(t), C_L(t)) - m(t) \cdot g \cdot \sin\gamma(t)}{m(t)} \\ \dot{\gamma}(t) &= \frac{L(h_e(t), V(t), C_L(t)) - m(t) \cdot g \cdot \cos\gamma(t)}{m(t) \cdot V(t)} \\ \dot{x}_e(t) &= V(t) \cdot \cos\gamma(t) \\ \dot{h}_e(t) &= V(t) \cdot \sin\gamma(t) \\ \dot{m}(t) &= -T(t) \cdot \eta(V(t)). \end{aligned} \tag{3}$$

To perform an optimized descent, the throttle should be near the idle detent position. Then, the minimum and maximum thrust constraints in (2) are the thrust idle plus a negative and a positive margin, respectively. Apart from the thrust constraints, the flight envelope constraints remain the same as in (2), except for the one referring to the bank angle, since it is assumed to be zero.

3. Iterative Learning

In this section, following [9], the ILC method implemented in this paper for precise aircraft trajectory tracking will be introduced. The starting point of the learning algorithm is a time-varying nonlinear model of a real dynamic system:

$$\begin{aligned} \dot{\mathrm{x}}(t) &= \mathrm{f}(\mathrm{x}(t), \mathrm{u}(t), t) \\ \mathrm{y}(t) &= \mathrm{g}(\mathrm{x}(t), \mathrm{u}(t), t), \end{aligned} \tag{4}$$

DOI: 10.13009/EUCASS2019-453

where $\mathrm{u}(t) \in \mathbb{R}^{n_u}$ is the control input, $\mathrm{x}(t) \in \mathbb{R}^{n_x}$ is the state, and $\mathrm{y}(t) \in \mathbb{R}^{n_y}$ is the output, and f and g are assumed to be continuously differentiable in x and u. Constraints on the input $\mathrm{u}(t)$ and the state $\mathrm{x}(t)$, and their time derivatives are represented by

$$\mathrm{Z}\,\mathrm{q}(t) \leq \mathrm{q}_{max}, \tag{5}$$

where

$$\mathrm{q}(t) = \left[\mathrm{x}(t), \mathrm{u}(t), \dot{\mathrm{x}}(t), \dot{\mathrm{u}}(t), \ldots, \frac{d^m}{dt^m}\mathrm{x}(t), \frac{d^m}{dt^m}\mathrm{u}(t)\right] \tag{6}$$

and $\mathrm{q}_{max} \in \mathbb{R}^{n_q}$. The inequality is defined component-wise and n_q is the total number of constraints, Z is a constant matrix of appropriate dimensions. In this case, the time-varying nonlinear model of the aircraft and its constraints are described in (3) and (2), where we have assumed that all state variables can be measured, therefore $\mathrm{y}(t) = \mathrm{x}(t)$.

The goal of the presented learning algorithm is to precisely track a feasible predefined output trajectory $y^*(t)$ over a finite time interval $t \in \mathcal{T} = [t_0, t_f]$, with $t_f < \infty$. The desired output trajectory $y^*(t)$ is here the result of solving an optimal control problem based on the system dynamics (4).

The system's behavior (4) can be represented as a linear time-varying system

$$\begin{aligned} \dot{\tilde{x}}(t) &= A(t)\tilde{x}(t) + B(t)\tilde{u}(t) \\ \dot{\tilde{y}}(t) &= C(t)\tilde{x}(t) + D(t)\tilde{u}(t), \quad t \in \mathcal{T}, \end{aligned} \tag{7}$$

where the time-dependent matrices $A(t), B(t), C(t), D(t)$ are the corresponding Jacobian matrices of the nonlinear functions f and g with respect to x and u. The triple $(\tilde{u}(t), \tilde{x}(t), \tilde{y}(t))$ represent small deviations from the desired trajectory and its corresponding state and input $(x^*(t), u^*(t), y^*(t))$,

$$\begin{aligned} \tilde{u}(t) &= \mathrm{u}(t) - u^*(t), \\ \tilde{x}(t) &= \mathrm{x}(t) - x^*(t), \\ \tilde{y}(t) &= \mathrm{y}(t) - y^*(t). \end{aligned} \tag{8}$$

In a real system, measurements are only available at fixed time intervals, therefore a discrete-time representation is needed, which results in the following linear, time-varying difference equations,

$$\begin{aligned} \tilde{x}(k+1) &= A_D(k)\tilde{x}(k) + B_D(k)\tilde{u}(k) \\ \tilde{y}(k) &= C_D(k)\tilde{x}(k) + D_D(k)\tilde{u}(k), \end{aligned} \tag{9}$$

where $k \in \mathcal{K} = \{0, 1, ..., N-1\}, N < \infty$ represents the discrete-time index. The desired trajectory is represented by a N-sample sequence

$$(u^*(k), x^*(k+1), y^*(k+1)), \quad k \in \mathcal{K}, \tag{10}$$

with given initial state $x^*(0)$. The input and state constraints (5) are similarly transformed

$$Z\tilde{q}(k) \leq q_{max}(k), \tag{11}$$

where $\tilde{q}(k)$ is the deviation of $q(k)$ from the corresponding nominal values $q^*(k)$ defined analogously to (8), and discretized.

3.1 Lifted system representation

It is useful to replace the model described above by a lifted vector representation, mapping the finite input time series $\tilde{u}(k), k \in \mathcal{K}$ into the corresponding output time series $\tilde{y}(k), k \in \mathcal{K}$ for each trial [14]. The deviations with respect to the desired trajectory (10) are then

$$\begin{aligned} u &= [\tilde{u}(0), \tilde{u}(1), \ldots, \tilde{u}(N-1)]^T && \in \mathbb{R}^{Nn_u} \\ x &= [\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(N)]^T && \in \mathbb{R}^{Nn_x} \\ y &= [\tilde{y}(1), \tilde{y}(2), \ldots, \tilde{y}(N)]^T && \in \mathbb{R}^{Nn_y}. \end{aligned} \tag{12}$$

Using this notation, the linear system (9) can be described as

DOI: 10.13009/EUCASS2019-453

$$\begin{aligned} x &= Fu + d^0 \\ y &= Gx + Hu, \end{aligned} \tag{13}$$

The lifted matrix $F \in \mathbb{R}^{Nn_x \times Nn_u}$ is composed of the matrices $F_{(l,m)} \in \mathbb{R}^{n_x \times n_u}$, $1 \leq l, m \leq N$, that is

$$F = \begin{bmatrix} F_{(1,1)} & \dots & F_{(1,N)} \\ \vdots & \ddots & \vdots \\ F_{(N,1)} & \dots & F_{(N,N)} \end{bmatrix}, \tag{14}$$

where

$$F(l,m) = \begin{cases} A_D(l-1)\dots A_D(m)B_D(m-1) & \text{if} \quad m < l \\ B_D(m-1) & \text{if} \quad m = l \\ 0 & \text{if} \quad m > l. \end{cases}$$

Matrices G and H are block-diagonal and analogously defined by

$$G(l,m) = \begin{cases} C_D(l) & \text{if} \quad l = m \\ 0 & \text{otherwise} \end{cases}$$

and

$$H(l,m) = \begin{cases} D_D(l) & \text{if} \quad l = m \\ 0 & \text{otherwise,} \end{cases}$$

respectively, where, $G_{(l,m)} \in \mathbb{R}^{n_y \times n_x}$ and $H_{(l,m)} \in \mathbb{R}^{n_y \times n_u}$. Vector d^0 contains the free response of the system (9) to the initial deviation $\tilde{x}(0) = \tilde{x}_0 \in \mathbb{R}^{n_x}$,

$$d^0 = \left[(A_D(0)\tilde{x}_0)^T, (A_D(1)A_D(0)\tilde{x}_0)^T, \dots, \left(\prod_{i=0}^{N-1} A_D(i)\tilde{x}_0 \right)^T \right]^T.$$

3.2 Disturbance estimation

In order to take into account the recurrent nature of the problem setting, the system (13) is written as

$$\begin{aligned} x_j &= Fu_j + d_j \\ y_j &= Gx_j + Hu_j, \end{aligned} \tag{15}$$

where the subscript j indicates the jth execution of the desired task and d_j represents the repetitive disturbance along the reference trajectory, which shows only slight random changes, ω_j, between iterations. Taking into account process and measurement noise, captured in the random variable μ_j, the evolution of the learning over consecutive trials can be represented as a Kalman filter model [15]

$$\begin{aligned} d_j &= d_{j-1} + \omega_{j-1} \\ y_j &= Gd_j + (GF + H)u_j + \mu_j, \end{aligned} \tag{16}$$

where both stochastic zero-mean Gaussian white noise variables, $\omega_j \sim \mathcal{N}(0, \Omega_j)$ and $\mu_j \sim \mathcal{N}(0, M_j)$, are trial-uncorrelated and assumed to be independent. Matrices Ω_j and M_j represent the noise covariances and can be characterized as diagonal matrices. In the aircraft simulation used for this study, noise has been introduced as measurement errors and variations in the aircraft model and the wind speed.

The Kalman filter estimates the current error d_j taking into account the output signals $y_0, y_1, \dots, y_j$ from previous trials. Given the initial values of the error estimate, $\widehat{d_0}$, and the error covariance matrix, $P_0 = E[(d_0 - \widehat{d_0})(d_0 - \widehat{d_0})^T]$, the disturbance estimate is calculated as

$$\widehat{d_j} = \widehat{d}_{j-1} + K_j \left(y_j - G\widehat{d}_{j-1} - (GF + H)u_j \right), \tag{17}$$

where K_j is the optimal Kalman gain

$$K_j = \left(P_{j-1} + \Omega_{j-1} \right) G^T \left(G \left(P_{j-1} + \Omega_{j-1} \right) G^T + M_j \right)^{-1}.$$

DOI: 10.13009/EUCASS2019-453

3.3 Input update

The learning update consists in deriving a model-based update rule that computes a new control input $u_{j+1} \in \mathbb{R}^{Nn_u}$ in response to the estimated disturbance $\widehat{d_j}$, that is, minimizing the deviation from the nominal trajectory in the next trial. Since this deviation x_{j+1} is unknown, the expected value of x_{j+1} given all past measurements is considered,

$$E\left[x_{j+1} \middle| y_1, y_2, \ldots, y_j\right] = F u_{j+1} + \widehat{d_j}. \tag{18}$$

The update rule can be expressed by the following optimization problem:

$$\begin{aligned} &\min_{u_{j+1}} && \|F u_{j+1} + \widehat{d_j}\|_\ell + \alpha \|D u_{j+1}\|_\ell \\ &\text{subject to} && L u_{j+1} \leq q_{max}, \end{aligned} \tag{19}$$

where constraints (11) are explicitly taken into account, and $\alpha \geq 0$ and the matrix D have the aim of penalizing the input or approximations of its derivatives in order to enforce the smoothness of the optimal problem solution. The vector norm ℓ, with $\ell \in \{1, 2, \infty\}$, of the minimization problem (19) affects the result and convergence of the learning algorithm and should be chosen in accordance with the performance objectives.

The update law defined in (19) can be formulated as a standard convex optimization problem of the form

$$\begin{aligned} &\min_z \left(\frac{1}{2} z^T V z + v^T z\right) \\ &\text{subject to} \quad W z \leq w \quad \text{and} \quad \eta_1 \leq z \leq \eta_2, \end{aligned} \tag{20}$$

where $z \in \mathbb{R}^{n_z}$ represents the vector of decision variables. Vectors v, w and matrices V, W have appropriate dimensions. In the experiments herein exposed, the input is updated only if the state variables' deviations from the desired trajectory exceed a pre-specified value. Otherwise, the same input is applied to the next iteration. The maximum deviation allowed may be different for each state variable and each part of the trajectory, e.g., in the case of a CDA, deviations at the beginning of the descent phase may be greater than at the end part of the trajectory, where high precision is required.

A scaling of the original signals u(t), x(t), y(t) in (4) is essential to guarantee reasonable results in the optimization problem. The scaling, exemplarily shown on the system's state x(t), reads as

$$x^s = S_x x, \quad S_x \in \mathbb{R}^{Nn_x \times Nn_x} \tag{21}$$

with x^s representing the scaled version of a lifted state vector x and S_x being the corresponding scaling matrix, usually represented by a diagonal matrix. Additionally, a state weighting matrix may be useful to give greater importance to some of the state variables over the rest.

As mentioned in the introduction, one of the advantages of the iterative learning algorithm is its non intrusiveness with respect to the aircraft's existing trajectory tracking controller, since a new reference trajectory can be provided to the following aircraft rather than a control input. Once the updated input u_{j+1} is calculated in (19), the new reference trajectory is obtained by introducing this input into the lifted model, dismissing the disturbances except for the initial deviation error

$$\begin{aligned} \mathrm{x}_N &= x_N + x_d, \quad \text{with} \quad x_N = F u_{j+1} + d^0_{j+1} \\ \mathrm{y}_N &= y_N + y_d, \quad \text{with} \quad y_N = G x_N + H u_{j+1}, \end{aligned} \tag{22}$$

where $\mathrm{x}_N \in \mathbb{R}^{Nn_x}$ is the new reference state variable lifted vector, $\mathrm{y}_N \in \mathbb{R}^{Nn_y}$ is the new reference output lifted vector, and x_d and y_d are the lifted vectors of the desired state and output, respectively

$$\begin{aligned} x_d &= [x^*(1), x^*(2), \ldots, x^*(N)]^T \in \mathbb{R}^{Nn_x} \\ y_d &= [y^*(1), y^*(2), \ldots, y^*(N)]^T \in \mathbb{R}^{Nn_y}. \end{aligned} \tag{23}$$

4. Trajectory planning

In this section, the method to generate a reference trajectory to be followed will be described.

As said before, in this paper, the ILC paradigm is applied to follow a CDA trajectory in the the vertical plane generated using an optimal control technique in which the horizontal distance has been minimized. The european

DOI: 10.13009/EUCASS2019-453

definition of CDA, as approved by the stakeholders, is: an aircraft operating technique in which an arriving aircraft descends from an optimal position with minimum thrust and avoids level flight to the extent permitted by the safe operation of the aircraft and compliance with published procedures and ATC instructions. According to [3], an optimum CDA starts from the Top-Of-Descent (TOD) and uses descent profiles that reduce controller-pilot communications and segments of level flight. Furthermore it provides for a reduction in noise, fuel burn and emissions, while increasing flight stability and the predictability of flight path to both controllers and pilots. One of the most important requirements for successful CDA operations is predictability, thus precise trajectory tracking is essential.

A feasible state trajectory with its corresponding nominal input is the starting point of the iterative learning algorithm. The trajectory is generated via the MATLAB application DIDO using a nominal model of an Airbus A320 aircraft. The input to DIDO is the problem formulation in a structured format, including system dynamics, constraints and cost function. Besides the optimal control and the resulting trajectory, DIDO automatically outputs the Hamiltonian, costates, path covectors, and endpoint covectors for the verification of the solution. A generic CDA has been considered, starting at the TOD and continuously descending until reaching an altitude of 2500 m at a speed of 150 m/s. The final conditions of altitude and speed of the designed trajectory are compliant with a STAR to complete the landing.

5. Experimental Setup

In this section, the simulated environment in which the experiments have been carried out will be described. It is composed by:

- a realistic flight simulator,
- an estimator of the disturbances acting on the aircraft, and
- an ILC controller.

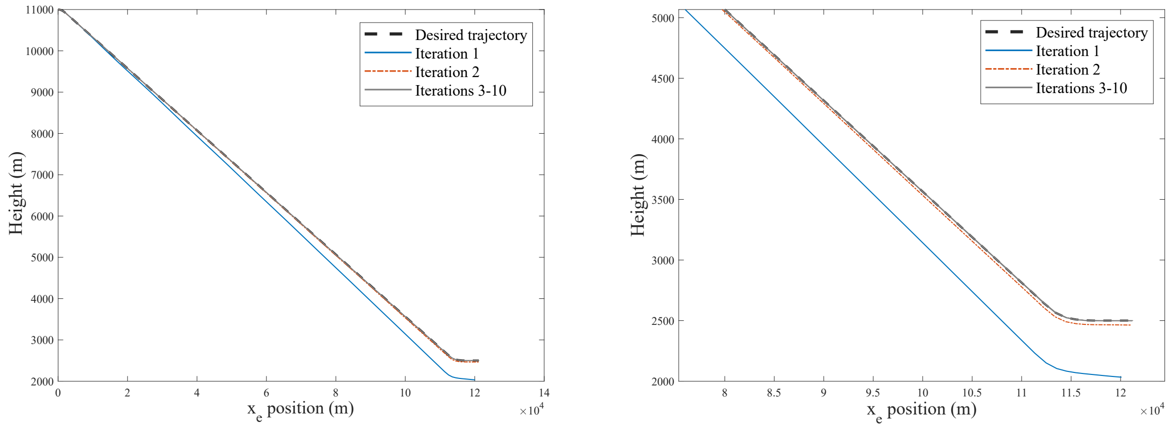
The ILC algorithm, which has been implemented in MATLAB, can be summarized as follows.

- Initialization. The first step of the algorithm is to load the trajectory to be followed. As said in Section 4, in this experiment an optimal desired trajectory has been calculated using a nominal model of the aircraft without taking into account perturbations. The desired trajectory, $y^* = x^*$, and the corresponding input, u^*, are loaded from a file provided by the optimal control software DIDO. This model does not coincide with the aircraft model used in the flight simulator, which is more realistic, containing perturbations. Then, settings and parameter values of the learning algorithm are introduced in the main program, the system is linearized about the desired trajectory, converted to a discrete-time system, and noted as a lifted domain representation. Finally, the Kalman gains are computed. To test the robustness of the ILC controller to modeling errors, the aircraft parameters entered here are slightly different from those used in the flight simulator and in trajectory planning. The optimization problem is now set up.
- j-th iteration. Using the most recent feed-forward input generated by the ILC algorithm, the corresponding trajectory of the aircraft is generated by the flight simulator. The simulator has been developed using a 3-DOF longitudinal model of an Airbus A320 aircraft implemented in Simulink, a widely used software in aircraft simulation [16]. The state and control variables are the same as those used in the ILC algorithm model. We assume that all states can be measured. Weather perturbations, model uncertainties and measurement noise are introduced in the simulated environment. After each execution of the trajectory tracking experiment, the results are stored and compared to the desired trajectory. The resulting error vector is fed into a Kalman filter providing an estimate of the tracking error, as described in Section 3.2.
- $j+1$-th iteration. Based on the tracking error estimated after the j-th iteration, the simulated aircraft is set to the initial position and the update step of the learning algorithm is executed, which, if necessary, determines a new reference trajectory to be tracked in the $j+1$-th iteration optimally compensating for the estimated error.

6. Results

In this section, the results of the application of the ILC scheme to precise aircraft trajectory tracking in CDAs will be reported. As said before, although the ILC algorithm is able to tackle 4D trajectories, in this section, for the sake of clarity, only the corresponding 3D paths will be reported.

DOI: 10.13009/EUCASS2019-453



(a) Complete path $x_e - h_e$.

(b) Detail of the final part of the path $x_e - h_e$.

Figure 2: Evolution of the path $x_e - h_e$ over iterations.

Fig. 2(a) shows the evolution of the tracking of the desired path over iterations. This desired path, related to the CDA obtained in the trajectory planning in Section 4, is shown in dashed black line. Figure 2(b) shows a detail of the same evolution in the final part of the path. The input applied in the first execution is the nominal input obtained in the desired trajectory generation, and the resulting trajectory tracked by the aircraft is generated by the flight simulator, as described in Section 5. After each execution the input is updated, if needed, and again the simulated aircraft tries to follow the new reference trajectory. As shown in Fig. 2(a) and Fig. 2(b), in the first execution the path followed by the simulator falls far below the desired one due to modeling and disturbance errors, but the ILC scheme rapidly learns from the first executions, compensating for the recurring disturbances and achieving a very precise tracking of the designed CDA trajectory after only three iterations.

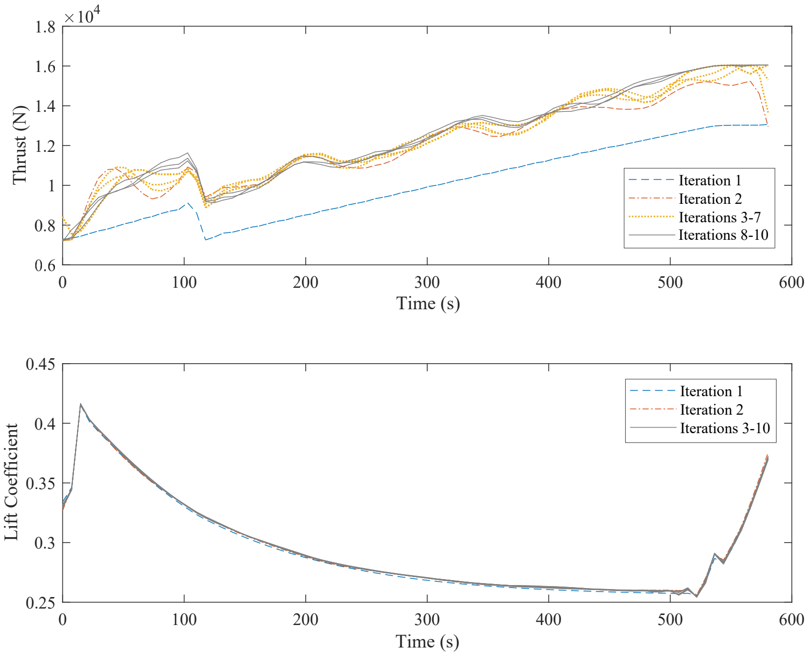


Figure 3: Evolution of the thrust and lift coefficient over iterations.

As shown in Fig. 3, the main control action is carried out by the thrust control input, which tends to converge over iterations whereas the lift coefficient input remains almost unchanged. Despite the convergence of the inputs, both the trajectory and the thrust vary over iterations due to non-repetitive disturbances, which would be compensated by

DOI: 10.13009/EUCASS2019-453

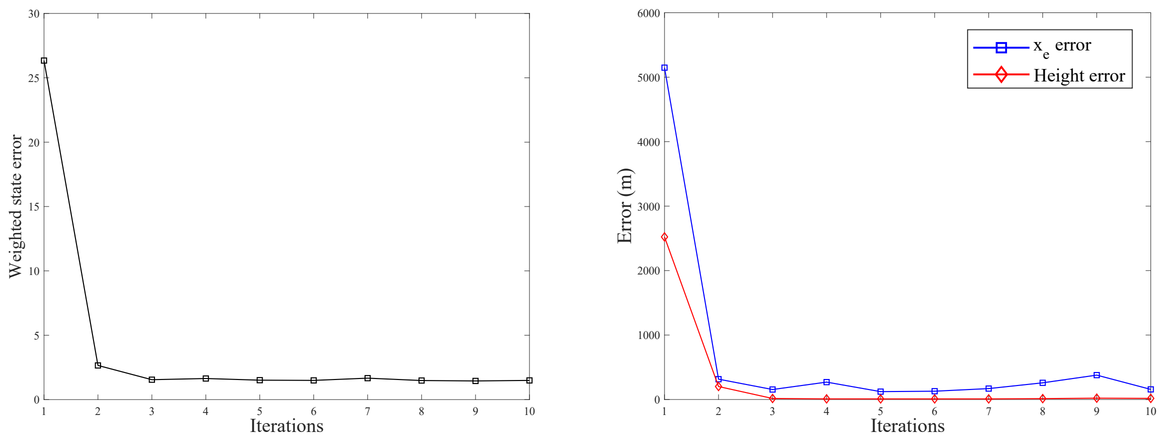
the feedback trajectory tracking controller of the aircraft.

The state weighted error in Fig. 4(a) allows the learning performance to be evaluated over iterations. It is calculated as:

$$e_{w,j} = \|S\,\mathrm{y}_j\|_2, \tag{24}$$

where S is the weighted scaling matrix of the state variables and y_j is the measured output vector. Similarly, the vertical and horizontal position errors are shown in Fig. 4(b), without scaling.

As expected, since the ILC scheme is intended to compensate for repetitive disturbances, the weighted state error and the horizontal and vertical position errors converge to the system noise level, and not to zero, due to non-repetitive disturbances.



(a) Weighted state error.

(b) Horizontal and vertical error.

Figure 4: Evolution of the weighted state and horizontal and vertical position errors over iterations.

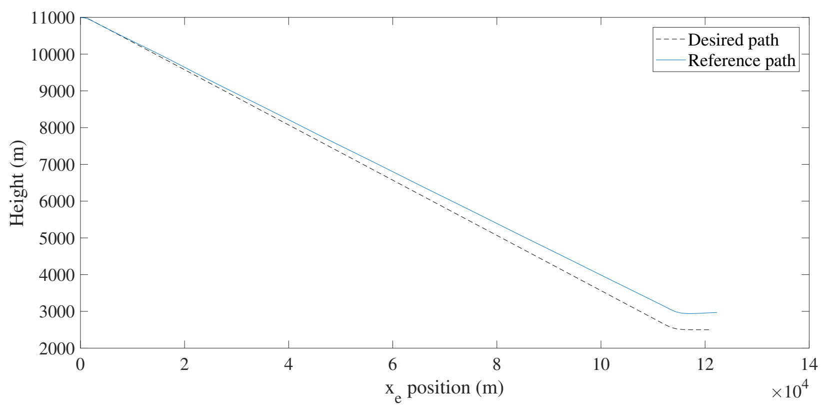


Figure 5: Reference path associated with the reference trajectory to be provided to the aircraft in the tenth iteration of the ILC algorithm to precisely track the desired trajectory.

As an example, the reference path generated in the tenth iteration of the ILC algorithm to be provided to the aircraft in the next iteration is depicted in Fig. 5. As explained before, the control paradigm here proposed is non-intrusive with respect to the underlying aircraft feedback controller since it calculates, at each iteration, a reference trajectory for the following aircraft rather than a control input to precisely track the desired trajectory.

7. Conclusions

Given an arrival procedure, the dynamical model of an aircraft and a trajectory to be followed compliant with the procedure, an optimization-based iterative learning approach has been applied to improve the precision of the aircraft

DOI: 10.13009/EUCASS2019-453

in following the trajectory, taking into account the deviations suffered by other flights operated with the same aircraft model and assuming that all the flights successively follow the same trajectory with short time-based separation and therefore are subject to similar recurrent disturbances. In this approach, optimality is pursued in both the estimation of the recurring disturbance and in the calculation of the input update, which optimally compensates for the disturbance.

The method proposed has been successfully applied to trajectory tracking of the simulation of commercial aircraft in continuous descent approaches. The descent flight of an Airbus A320 aircraft has been modeled in the vertical plane, assuming constant course and leveled wing flight and International Standard Atmosphere conditions. Weather perturbations, model uncertainties and measurement noise have been introduced in the model. These noise signals vary from iteration to iteration and are assumed to be trial-uncorrelated sequences of zero-mean Gaussian white noise. It has been shown that precision in aircraft trajectory tracking can be improved by pure feed-forward adaptation of the control input.

Future work will include applying iterative learning control techniques to determine the top of descent under given atmospheric conditions to improve precision in reaching the final fix in continuous descent approaches. Additionally, it is intended to extend the approach here proposed to flights not restricted to the vertical plane and applying it to other flight procedures.

8. Acknowledgments

This work has been partially supported by the grants number `TRA2017-91203-EXP` and `RTI2018-098471-B-C33` of the Spanish Government.

References

[1] A. Tewari, *Advanced Control of Aircraft, Spacecraft and Rockets*. Wiley, 2011.

[2] EUROCONTROL, "Continuous Descent. A Guide to Implementing Continuous Descent," Tech. Rep. , European Organization for the Safety of Air Navigation, 2011.

[3] ICAO, "Continuous Descent Operations (CDO) Manual," Tech. Rep. Doc 9931 AN/476, International Civil Aviation Organization, 2010.

[4] ICAO, "Global Air Navigation Plan," Tech. Rep. Doc 9750 AN/963, International Civil Aviation Organization, 2016.

[5] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[6] D. A. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, pp. 2039–2114, 2006.

[7] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*. Springer, 2003.

[8] J.-X. Xu, S. K. Panda, and T. H. Lee, *Real-time Iterative Learning Control: Design and Applications*. Springer, 2009.

[9] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Autonomous Robots*, vol. 33, no. 1-2, pp. 103–127, 2012.

[10] O. Purwin and R. D'Andrea, "Performing and extending aggressive maneuvers using iterative learning control," *Robotics and Autonomous Systems*, vol. 59, pp. 1–11, 2011.

[11] X. Liang, M. Zheng, and F. Zhang, "A scalable model-based learning algorithm with application to UAVs," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 839 – 844, 2018.

[12] V. Mouillet, "User Manual for the Base of Aircraft Data (BADA) Revision 3.14," Tech. Rep. , EUROCONTROL Experimental Centre, Brétigny, France, 2017.

[13] D. G. Hull, *Fundamentals of Airplane Flight Mechanics*. Springer-Verlag, 2007.

[14] B. Bamieh, J. B. Pearson, B. A. Francis, and A. Tannenbaum, "A lifting technique for linear periodic systems with applications to sampled-data control," *Systems & Control Letters*, vol. 17, no. 2, pp. 79 – 88, 1991.

DOI: 10.13009/EUCASS2019-453

[15] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[16] L. Moin, A. Baig, and V. Uddin, "State space model of an aircraft using Simulink," *International Journal of System Modeling and Simulation*, vol. 2, no. 4, pp. 1–6, 2017.