A Novel Robust MPC Based Aircraft Auto-Throttle for Performing 4D Contract Flights

Vangelis Petratos
Department of Mechanical
& Aeronautical Engineering
University of Patras
26504 Patras, Greece
petratos@upatras.gr

Dimitrios Dimogianopoulos
Department of Automation
Technological Education
Institute of Piraeus
122 44 Egaleo, Greece
dimogian@teipir.gr

Fotis Kopsaftopoulos
& Spilios Fassois
Department of Mechanical
& Aeronautical Engineering
University of Patras
26504 Patras, Greece
{fkopsaf, fassois}@mech.upatras.gr

***Abstract*—A novel robust auto-throttle controller for use within a Four-Dimensional Contract (3D+Time) flight context is introduced. Its design uses typical receding horizon techniques, with control values resulting from optimization of the predicted system response over future time intervals. The novelty is two-fold: First, the controller is designed to optimize the aircraft's fuel efficiency (represented by the Specific Fuel Consumption-SFC) along with its position during the 4D flight. Second, the control value admits a closed-form expression, which greatly simplifies its on-board calculation. Tests against conventional PID based auto-throttle controllers illustrate the current controller's superior robustness under challenging flight conditions (turbulence).**

I. Introduction

The current trend in Air Traffic Management (ATM) is the implementation of the Four-Dimensional flight concept, associating the 3D aircraft position to relevant time points (T) throughout the flight. Specifically, the Four-Dimensional Contracts (4DCos) are the succession of Four-Dimensional Way Points (4DWPs), describing the desired 3D aircraft position and the corresponding time. Flight safety in 4DCos is ensured by defining "safety bubbles", that is, the maximum allowable cross-track and along-track deviations from the aircraft's current position without conflicts with other aircraft. Although several new constraints related to the strict aircraft 4DCo compliance are introduced, the 4DCo concept provides new opportunities for a safer, environmental friendlier and less human-involved management of the airspace [1].

Efforts using the 4DCo concepts for optimizing the fuel use in terms of fuel burnt and emissions over a large part of the considered airspace have been reported in [2] and [3]. Both studies introduce fuel-optimal paths (climbs/descends in [2], constant altitude paths in [3]) either by minimizing a Hamiltonian [2] or by solving an optimal control problem [3]. Nevertheless, optimizing the overall fuel use for a part of the airspace obviously means similarly optimized flights of each aircraft using this space: In other words, the development of auto-throttle strategies suitably defined for both 4DCo compliance and optimal efficiency.

The basics of controlling an engine under multiple (mechanical or aerothermal) constraints has been extensively presented in [4]. Therein, the basic principles and limitations for each engine component are reviewed, and various control strategies respecting the typical requirements for sensors and actuators are given. Nonetheless, the control objective is to use the engine efficiently (for better fuel consumption and part life) rather than to satisfy flight-related constraints, as those found in the 4DCos. An effort towards defining engine control strategies for simultaneously optimizing objectives other than mechanical constraints, has been made in [5]. These strategies result from solving a nonlinear predictive control problem, and aim at simultaneously optimizing performance (thrust, specific fuel consumption) and operability (costs, in-flight mishaps and other flight-related parameters). Nevertheless, the solution is computed via an iterative procedure, meaning that its on-board implementation might become problematic. Similar efforts for designing controllers optimal with respect to multiple objectives (one of which is the fuel use) may be found in [7]. There, the response time during the engine's acceleration/deceleration and the fuel consumption are considered as objective functions, whereas a Wiener model with experimentally estimated parameters represents the gas turbine engine. The controller gains are tuned by Particle Swarm Optimization (PSO) techniques, meaning that the on-line gain computation depends on the PSO convergence rate.

Fuel efficiency and overall optimization of jet engines have often been linked to SFC minimization [8], [9]. Even though measuring the SFC is still a controversial subject, it is widely accepted that the SFC may be estimated either from thermodynamic models [10], or indirectly by thrust estimation via physics-based models [11] or from data provided by the Full Authority Digital Engine Control (FADEC) [12], as will be discussed later on. In any case, optimizing engine control for minimal SFC is already possible, and (if engine performance charts are available) may be even executed on-line [8], [9].

This paper aims at introducing an auto-throttle controller design, which achieves compliance with the aircraft's 4DCo, while simultaneously optimizing the fuel usage with respect to a desired set-point value. The controller design is based on typical receding horizon (that is, Model Predictive Control-MPC) techniques as in [5], with control values resulting from optimizing the predicted system response over future time intervals. The novelty lies in that the system response is now optimized over both fuel efficiency (in terms of SFC as in [8],

Asian Control Conference (ASCC), June 2013, Istanbul, Turkey

[9], [12]) and the aircraft position in the 4DCo bubble. The latter corresponds here to an operability objective [5] and, to the best of the authors' knowledge, it has never been used as such before. Another innovation of the currently proposed controller is related to the SFC representation. Given the possibility of SFC estimation, a linear AutoRegressive with eXogenous excitation (ARX) representation of the relationship between throttle and SFC may be identified. By the past, ARX models have often proved sufficient in representing various engine dynamics [13]. The advantage of this ARX representation is that an analytical closed-form solution to the MPC problem is feasible, as opposed to the iterative solution in [5]. Hence, the control value may be computed on-board in real-time without any computational concerns. The proposed controller is implemented on a Boeing 737 simulation software and tested via several flights conducted under normal or degraded conditions. Comparisons in such conditions are made with a PID controller, tuned in the (traditional) sense of achieving 4DCo compliance. The robustness advantage in favor of the currently proposed controller seems quite significant.

The paper is organized as follows: Section II presents the principles of controller design. Simulation results are given in Section III along with comparisons of the proposed controller with a classical control PID based auto-throttle. Finally, some concluding remarks are provided in section IV.

II. Principles of Controller Design

The controller design is based on three main elements:

(i) An ARX model representing the dynamics between throttle command and engine SFC.
(ii) The Basic Minimization Criterion (BMC), penalizing the mean square error between predicted (by the ARX model) and desired SFC values.
(iii) An additional (to the BMC) extension term, penalizing the aircraft non-compliance to the allocated 4DCo.

A. Modeling the Relationship Between Throttle & Engine SFC

The modeling of the relationship between throttle command and engine SFC may be based on a discrete-time model obtained via standard identification procedures [14]. In this study, an AutoRegressive with eXogenous excitation (ARX) model is used. An ARX(na, nb) model admits the form [14]:

$$y[t] + \sum_{i=1}^{na} a_i \cdot y[t-i] = \sum_{i=0}^{nb} b_i \cdot u[t-i] + w[t] \qquad (1)$$

or using the backshift operator ($\mathcal{B}^i \cdot y[t] \triangleq y[t-i]$):

$$A(\mathcal{B}) \cdot y[t] = B(\mathcal{B}) \cdot u[t] + w[t], \; w[t] \sim \text{iid}\, \mathcal{N}(0, \sigma_w^2) \qquad (2)$$

$$\begin{aligned} A(\mathcal{B}) &= 1 + a_1 \cdot \mathcal{B} + \ldots + a_{na} \cdot \mathcal{B}^{na} \\ B(\mathcal{B}) &= b_0 + b_1 \cdot \mathcal{B} + \ldots + b_{nb} \cdot \mathcal{B}^{nb} \end{aligned} \qquad (3)$$

with t designating the normalized discrete time[1] ($t = 1, 2, \ldots$), $y[t]$, $u[t]$ the measured output (SFC) and control input (throttle command) signals, respectively. The AutoRegressive (AR) and eXogenous (X) orders are noted as na and nb, respectively, whereas $A(\mathcal{B}), B(\mathcal{B})$ are the AR and X polynomials, respectively. The signal $w[t]$ is uncorrelated (white) with zero mean and variance σ_w^2. It coincides with the model based one-step-ahead prediction error, and is uncorrelated with the excitation $u[t]$. The symbol $\mathcal{N}(\cdot, \cdot)$ designates Gaussian distribution with the indicated mean and variance, and iid stands for identically independently distributed.

[1]The absolute time is $(t-1)T_s$, where T_s stands for the sampling period.

The model is parametrized in terms of the parameter vector $\boldsymbol{\theta} = [a_1 \ldots \; a_{na} \;\; b_0 \ldots \; b_{nb}]^T$ to be estimated from the measured signals. Model estimation may be achieved based on minimization of the Ordinary Least Squares (OLS) or Weighted Least Squares (WLS) criteria [14, pp. 8–9].

As previously stated, obtaining an SFC measurement mainly depends on the feasibility of estimating thrust. In general, thrust estimation may be performed either by using physics-based thermodynamic models as stand-alone thrust estimators [10], or as the basis for designing observers for on-line estimation [11]. Again, thrust may be estimated from data either collected or estimated by the digital avionics and communicated to FADEC system, as claimed in [12]. Note that choosing to minimize SFC for controller design purposes instead of, for instance, the pure fuel flow is due to the fact that the SFC perfectly describes fuel efficiency, since the fuel usage is related to the thrust produced. The SFC minimization is considered as a vital objective in engine development, since its reduction by 4% is, reportedly, a reason for implementing a new engine design [15]. For the same reasons, algorithms exploiting engine performance charts and attempting to optimize engine control on-line with respect to the SFC have already been reported [8], [9].

B. The Basic Minimization Criterion (BMC)

The Basic Minimization Criterion (BMC) penalizes the mean square error between predicted SFC and desired value at each time instant. The identified ARX model is used for predicting SFC values over a j-step-ahead prediction horizon based on the current and past throttle commands. Subsequently, the BMC minimization results in the future values of the control input (throttle command), which will achieve SFC values close to the desired (and predefined) ones.

Using the following identity (Diophantine Equation, [16]):

$$E_j(\mathcal{B}) \cdot A(\mathcal{B}) + \mathcal{B}^j \cdot F_j(\mathcal{B}) = 1, \qquad (4)$$

with $E_j(\mathcal{B})$ and $F_j(\mathcal{B})$ polynomials uniquely defined given $A(\mathcal{B})$, and the prediction horizon of j future steps. The optimal j-th-step-ahead predictor given measured output data up to time t and any given $u[t+j]$ for $j > 1$ is obtained by:

$$\widehat{y}[t+j|t] = G_j(\mathcal{B}) \cdot u[t+j] + F_j(\mathcal{B}) \cdot y[t], \qquad (5)$$

where $G_j(\mathcal{B}) \triangleq E_j(\mathcal{B}) \cdot B(\mathcal{B})$.

Denoting the coefficients of the polynomials $G_j(\mathcal{B})$ and $F_j(\mathcal{B})$ as $g_{j,0}, \ldots, g_{j,j-1}$ and $f_{j,0}, \ldots, f_{j,na-1}$, respectively, the vector of j estimated future outputs may be expressed as:

$$\widehat{\boldsymbol{y}} = \boldsymbol{G} \cdot \boldsymbol{u} + \boldsymbol{p} \qquad (6)$$

where:

$$\widehat{\boldsymbol{y}} = \left[\widehat{y}[t+1] \ \dots \ \widehat{y}[t+j]\right]^T, \ \boldsymbol{u} = \left[u[t+1] \ \dots \ u[t+j]\right]^T,$$

$$\boldsymbol{G} = \begin{bmatrix} g_{1,0} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ g_{j,j-1} & \cdots & g_{j,0} \end{bmatrix}, \ \boldsymbol{p} =$$

$$\begin{bmatrix} g_{1,nb} & \cdots & g_{1,1} & f_{1,na-1} & \cdots & f_{1,0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{j,j-1} & \cdots & g_{j,0} & f_{j,na-1} & \cdots & f_{j,0} \end{bmatrix} \cdot \begin{bmatrix} u[t-nb+1] \\ \vdots \\ u[t] \\ y[t-na+1] \\ \vdots \\ y[t] \end{bmatrix}$$

Now, consider the following minimization criterion:

$$J = \sum_{i=1}^{j} (y[t+i] - r[t+i])^2 + \mu \cdot \sum_{i=1}^{j} (u[t+i] - u[t+i-1])^2. \quad (7)$$

The first term of (7) penalizes the squared difference between the system's output SFC and the user-defined SFC signal $\boldsymbol{r} = \left[r[1] \ \dots \ r[j]\right]^T$. The second term penalizes the squared first differences of the future input control (throttle) commands, with μ being a user-defined weighting factor. This term is often used when the output-predicting model (here, the ARX model) has non-minimum phase characteristics [17]. It penalizes any abrupt activity of the throttle command and ensures smoother throttle operation. The second sum in (7) may be expressed as $\boldsymbol{D} \cdot \boldsymbol{u} - \boldsymbol{I_0}$, [18] with $\boldsymbol{I_0} = [\ u[t] \ 0 \ \cdots \ 0]^T$ and:

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}.$$

Hence, the minimization criterion may be expressed as:

$$J = (\boldsymbol{G}\cdot\boldsymbol{u}+\boldsymbol{p}-\boldsymbol{r})^T(\boldsymbol{G}\cdot\boldsymbol{u}+\boldsymbol{p}-\boldsymbol{r})+\mu\cdot(\boldsymbol{D}\cdot\boldsymbol{u}-\boldsymbol{I_0})^T(\boldsymbol{D}\cdot\boldsymbol{u}-\boldsymbol{I_0}). \quad (8)$$

As J is a quadratic function of the unknown control vector $\boldsymbol{u}$ (yet to be estimated), its minimum is computed by setting its partial derivative (with respect to $\boldsymbol{u}$) to zero. This yields:

$$\boldsymbol{u} = \left(\boldsymbol{G}^T\boldsymbol{G} + \mu \cdot \boldsymbol{D}^T\boldsymbol{D}\right)^{-1}\left(\mu \cdot \boldsymbol{I}_0 + \boldsymbol{G}^T(\boldsymbol{r} - \boldsymbol{p})\right). \quad (9)$$

Note that this is a closed-form solution of the control value $\boldsymbol{u}$. This gives a significant advantage in computational effort over alternative control methods where the solution results from optimization procedures [6]. Hence, no extra on-board computational effort is required, thus making the proposed controller easier to implement in real-time application.

C. The Extended Minimization Criterion

Apart from optimizing the SFC around some desired values, the controller must ensure aircraft compliance to the assigned

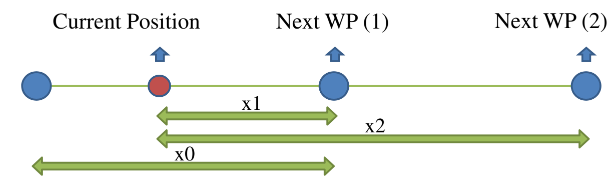

Fig. 1. The aircraft desired velocity at the current condition depends on WP(1) and WP(2). A component-velocity is calculated for each of these two WPs, dividing its distance from the aircraft with the remaining time-to-arrival. The desired velocity is a linear combination of the two component velocities.

4DCo. This is done by extending the BMC in (7) as follows:

$$J^{\text{EXT}} = J + \lambda_0[t] \cdot \sum_{i=1}^{j} u^2[t+i] + \lambda_1[t] \cdot \sum_{i=1}^{j} (1 - u[t+i])^2 \quad (10)$$

with $\lambda_0[t]$ and $\lambda_1[t]$ positive time-varying weighting factors, which are updated each time a new control value is computed, and are valid until the next control value is provided from J^{EXT} minimization. The first term penalizes the squares of the control signal: Thus, increasing $\lambda_0[t]$ will result in a lower throttle command. Similarly, the second term penalizes the squared differences between the control signal and the unity: Thus larger $\lambda_1[t]$ values favor increased throttle commands. The selection of unity is due to the assumption that the throttle command is normalized in the $[0, 1]$ interval.

In order to compute $\lambda_0[t]$ and $\lambda_1[t]$, first a desired aircraft ground velocity has to be defined. This velocity is a linear combination of two component ground velocities, which are associated to the distances and times-of-arrivals of the two WPs ahead from the current aircraft position, namely WP(1) and WP(2) in Fig. 1. The first component velocity is associated to WP(1) in Fig. 1, and is computed by dividing the distance x_1 (Fig. 1) with the estimated time-to-arrival. The same holds for the second component velocity, which is associated to WP(2). The desired ground velocity results from the linear combination of these two component ground velocities with weights W and $1 - W$ provided by a sigmoid function $W(x) = (erf(8 \cdot (x - 0.5)) + 1)/2$. The term $erf(\cdot)$ is the Gauss Error Function [19] and x is the distance to be covered until the next WP over the total distance x_0 between two WPs (Fig. 1). These time varying weights and the sigmoid function produce a smooth transition from the first component ground velocity to the second, as the aircraft reaches the next WP (Fig. 2, green dash-dot line). If a linear (instead of sigmoid) transition is used, the desired ground velocity is not continuous when the aircraft changes WP (Fig. 2, red line). Note, also, that linearly combining the two component velocities compensates for the fact that the first component (which would be a more obvious choice) becomes either very large or small when reaching the WP (Fig. 2, black line).

As seen, $W(x)$ is suitably defined for normalizing the input–output to the zero-one interval. The factor 8 is for regulating the sigmoid's curvature, that is, the smoothness of the transition and the flatness of its limits (Fig. 3), and is set by the user. As shown in Fig. 3, a curvature factor of 4 (black line)

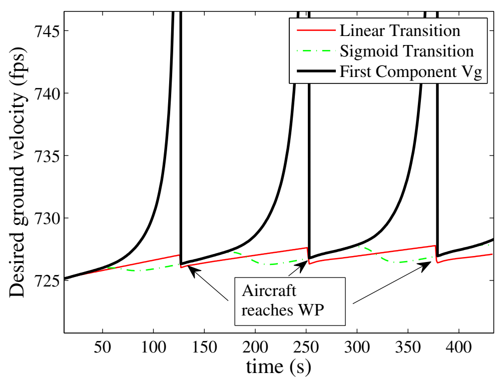

Fig. 2. Methods for defining the desired aircraft ground velocity.

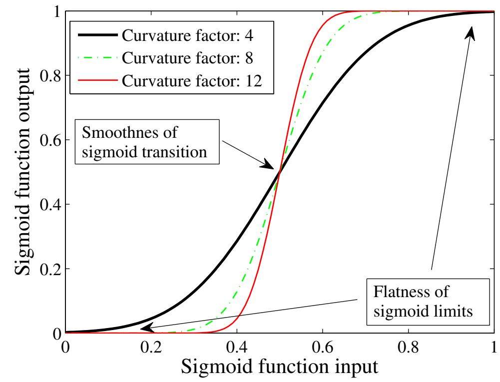

Fig. 3. The sigmoid function for various curvature factors.

produces a smooth transition, but without flat enough sigmoid limits. These have to be appropriately flat to compensate for discontinuities occurring when the aircraft reaches the next WP, as discussed previously. A curvature factor of 12 (Fig. 3, red line) produces flat enough sigmoid limits, but a sharp transition leading to sudden changes in the desired ground velocity if the two components differ significantly. Finally, a curvature factor of 8 (Fig. 3, green dash-dot line) produces a smooth transition and appropriately flat sigmoid limits.

The weighting factors $\lambda_0[t]$ and $\lambda_1[t]$ are defined as:

$$\lambda_0[t] = c_a \cdot e^{c_b(v_t[t]-v_d[t])}, \ \lambda_1[t] = c_a \cdot e^{c_b(v_d[t]-v_t[t])} \quad (11)$$

where c_a, c_b are user defined tuning parameters and $v_d[t]$, $v_t[t]$ the desired and actual aircraft ground velocities, respectively.

Note that whenever the actual velocity becomes greater than the desired value, $\lambda_0[t]$ increases, thus limiting the throttle magnitude due to the penalty imposed in (7). Inversely, the throttle magnitude increases when the actual velocity becomes smaller than the desired value. Finally, if the actual velocity is close to the desired one, both $\lambda_0[t]$ and $\lambda_1[t]$ will admit small values (close to c_a), allowing for commands favoring the achievement of desired SFC values over 4DCo compliance.

The extended minimization criterion admits the form:

$$J^{\text{EXT}} = J + \lambda_0[t] \cdot \boldsymbol{u}^T\boldsymbol{u} + \lambda_1[t] \cdot (\boldsymbol{I}_e - \boldsymbol{u})^T(\boldsymbol{I}_e - \boldsymbol{u}), \quad (12)$$

where $\boldsymbol{I}_e = [1 \ \dots \ 1]^T$. Minimizing J^{EXT} as in (8), (9) yields:

$$\boldsymbol{u} = \big(\boldsymbol{G}^T\boldsymbol{G} + \mu \cdot \boldsymbol{D}^T\boldsymbol{D} + (\lambda_0[t] + \lambda_1[t]) \cdot \boldsymbol{I}_e\big)^{-1} \cdot \big(\mu \cdot \boldsymbol{I}_0 + \lambda_1[t] \cdot \boldsymbol{I}_e + \boldsymbol{G}^T \cdot (\boldsymbol{r} - \boldsymbol{p})\big). \quad (13)$$

Figure 4 presents the proposed controller's block diagram.

III. Simulation Results

A. Identification of the Engine SFC Dynamics

A Boeing 737 using the open source JSBSim simulator [20] is used for obtaining throttle command and SFC data, as well as for implementing the designed controller. The flights considered for the ARX model identification procedure include constant altitude, speed and heading, and no turbulence, gusts and winds. The duration of each flight is 180 seconds and the sampling rate used is $f_s = 20$ Hz. Initial position and control inputs of the aircraft are provided via a trimming function of the flight simulator. The throttle command is selected as a low-pass Gaussian noise, whose mean value is the throttle control value obtained from the trimming function for the specific flight. This serves to properly excite the engine dynamics and ensure identifiability of the throttle–SFC relationship.

The SFC dynamics are represented by ARX models for various flight cases inside the flight envelope. The modeling procedure involves estimating ARX(na, nb) models of increasing na and nb orders until a minimum of the Bayesian Information Criterion (BIC) [14, pp. 505–507] is reached: This achieves a compromise between low Residual Sum of Squares (RSS) and a compact model. The final model is validated by checking its residual sequence for lack of correlation [14, pp. 512–513]. This identification procedure results in an ARX$(12, 12)$ model structure (see Fig. 5), with the parameters of a typical flight (starting at 31000 ft, with ground velocity

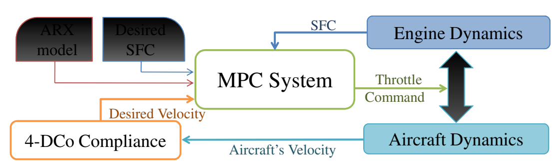

Fig. 4. The MPC based auto-throttle block diagram.

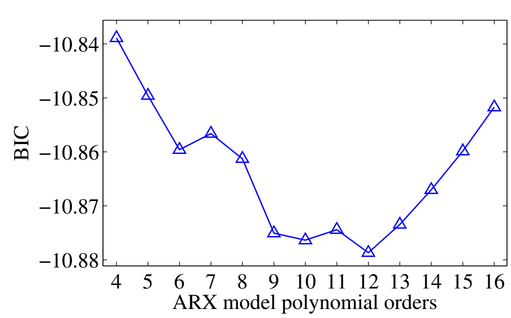

Fig. 5. Bayesian Information Criterion (BIC) order selection criterion for ARX(n, n) models.

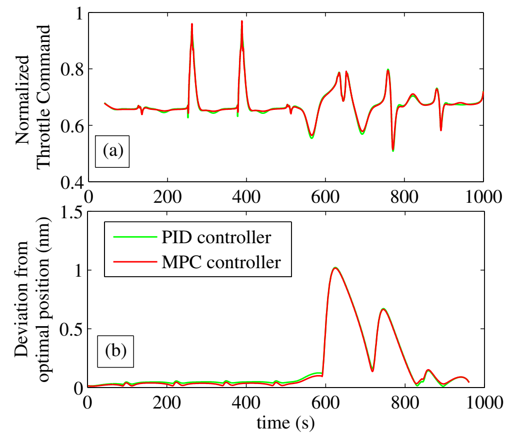

Fig. 6. Simulated B737 flight trajectory on 3-D space.

Fig. 7. Comparison of the proposed MPC and PID based controllers for an indicative turbulence-free cruise flight: (a) throttle command output and (b) 4DCo deviations (nautical miles, nm).

741 fps and initial throttle command equal to 0.52) presented in Table I.

B. Controller Assessment

The proposed controller is compared with a PID-based one, via simulated flight scenarios on cruise mode including turbulence and/or wind. Each flight's duration is 16 minutes and the 4DCo's feature a climb of 1000 ft at time $t = 350$ s, a turn of 25 degrees at $t = 630$ s, and an increase of speed by 25 fps at $t = 880$ s (see Fig. 6). Each flight case has different initial altitude and velocity values, obtained via the trimming function of the JSBSim simulator. At each simulation, the proposed controller uses the identified ARX model valid for the turbulence/ wind-free flight, and should achieve the nominal SFC value for the specific flight obtained by a previous (turbulence-free) simulation. The user defined parameters of the controller are computed as $c_a = 1$, $c_b = 0.3$, $\mu = 200$ and $j = 30$ samples (1.5 s), by a trial-and-error procedure. An indicative flight case is shown in Fig. 6.

Figure 7 presents the comparison of the proposed and PID based controllers for an indicative cruise flight with wind but no turbulence. Figure 7a depicts the throttle command from the two controllers. Clearly, they both produce an almost identical throttle command output under turbulence-free conditions. Figure 7b presents the 4DCo deviations for the same flight,

TABLE I
ARX(12,12) PARAMETERS FOR A TYPICAL FLIGHT (STARTING AT 31000 FT WITH GROUND VELOCITY OF 741 FPS AND THROTTLE AT 0.52).

AR Parameters		X Parameters	
a_1 = 0.237	a_8 = 0.133	b_0 = -0.206	b_7 = -0.114
a_2 = 0.213	a_9 = 0.075	b_1 = -0.191	b_8 = -0.108
a_3 = 0.218	a_{10} = 0.051	b_2 = -0.176	b_9 = -0.088
a_4 = 0.172	a_{11} = 0.049	b_3 = -0.168	b_{10} = -0.060
a_5 = 0.158	a_{12} = -0.038	b_4 = -0.153	b_{11} = -0.044
a_6 = 0.155		b_5 = -0.137	b_{12} = -0.016
a_7 = 0.156		b_6 = -0.125	

with the proposed controller showing only marginally better performance. The peak in the deviations at time $t = 630$ s is due to the 25 degrees turn at the same time. Although steep for a Boeing 737, this command demonstrates the controllers performance under challenging handling conditions.

Figure 8 presents indicative results for the two controllers for cruise flight with type A (that is, low level of) turbulence, as provided by the simulator. In this case, the throttle outputs of the two controllers differ significantly, with the proposed controller providing a throttle activity quite smoother than that of the PID. In the case of type B (that is, significant level of) turbulence (Fig. 9), the PID controller is highly affected by turbulence exhibiting high-frequency oscillations, while the proposed controller is hardly affected at all. In other words, the difference in throttle activity between the two controllers becomes more pronounced as the turbulence increases, with the PID based controller being highly affected. This shows that the proposed MPC controller is significantly more robust compared with classical solutions under flight conditions that the controllers were not designed for (turbulence). Furthermore, note that, although the throttle commands of the controllers differ significantly, their 4DCo compliance is quite similar. Nevertheless, any throttle activity similar to that from the PID based controller in Figs. 8 and 9 may not be considered as implementable and could constitute a threat for the system's long term reliability.

Finally, note that the SFC was regulated to a desired benchmark value, *not* a minimal one. In other words, the currently proposed controller favors the accurate 4DCo compliance and the enhanced robustness with respect to classical alternatives (PID based controllers) over the absolute SFC minimization.

IV. CONCLUSIONS

A novel auto-throttle controller aiming at simultaneously optimizing the aircraft position in 4DCo flight and regulating its fuel use (in terms of SFC) towards some desired value

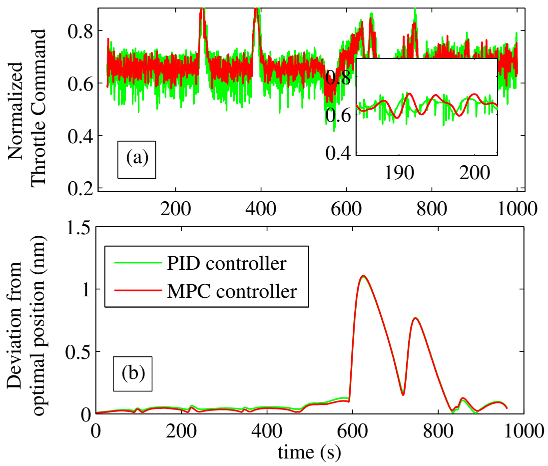

Fig. 8. Comparison of the MPC and PID based controllers for an indicative type A (low-level) turbulence cruise flight: (a) throttle command output and (b) 4DCo deviations (nautical miles, nm).

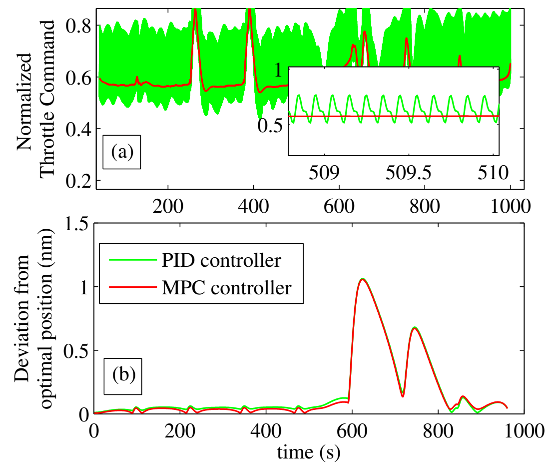

Fig. 9. Comparison of the MPC and PID based controllers for an indicative type B (high-level) turbulence cruise flight: (a) throttle command output and (b) 4DCo deviations (nautical miles, nm).

has been presented. The controller design is formulated as a MPC problem, with its solution providing the throttle control value. The innovation resides in the MPC criterion penalizing both excessive predicted SFC values and potential non-compliance of the aircraft to its 4DCo. Hence, the computed throttle control value takes simultaneously into account these two objectives. An additional innovation is related to the identification of a linear stochastic ARX representation for the relationship between throttle and SFC. This enables the formulation of a linear MPC problem, as opposed to the nonlinear MPC formulations used in other studies. Thus, in the present case an analytical closed-form solution to the MPC problem (in other words, the throttle control value) is obtainable. This is beneficial for the controller's on-board implementation, since no extra computational burden is imposed. Comparisons via simulated flights with a conventional PID based controller show that the proposed solution is more robust under challenging conditions (turbulence), while ensuring the aircraft compliance to the assigned 4DCo. Finally, in the current stage of development, emphasis was not given to achieving the absolute SFC minimization, which is the object of future work. Instead, the proposed controller achieves the SFC regulation towards a user-defined (realistic) value, while exhibiting superior robustness with respect to classical control (PID based) alternatives.

ACKNOWLEDGMENT

Research partially supported by the European Commission – FP7 Project No. 266296 4DCo–GC ("4 Dimension Contracts – Guidance and Control").

REFERENCES

[1] A. Joulia, Four Dimensional Contracts Guidance and Control European Project document "4DCo-GC2-AnnexI_v1_2".
[2] J.W. Burrows, Fuel Optimal Trajectory Computation. *Journal of Aircraft*, Vol. 19, No 4, 1982, pp. 324–329.
[3] A. Franco, D. Rivas, Minimum-Cost Cruise at Constant Altitude of Commercial Aircraft Including Wind Effects, Engineering Notes, *Journal of Guidance, Control and Dynamics*, Vol. 34, No. 4, 2011.
[4] H. A. Spang, H. and Brown, Control of Jet Engines, *Control Engineering Practice*, Vol. 7, No. 9, 1999, pp. 1043–1059.
[5] B. J. Brunell, Patent: Methods and Apparatus for Model Predictive Control of Aircraft Gas Turbine Engines, US 6823253, 2004.
[6] B. J. Brunell, R. R. Bitmead, A. J. Connolly, Nonlinear Model Predictive Control of an Aircraft Gas Turbine Engine, *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada USA, December 2002.
[7] M. Montazeri-Gh, S. Jafari, M.R. Ilkhani, Application of Particle Swarm Optimization in Gas Turbine Engine Fuel Controller Gain Tuning, *Engineering Optimization*, Vol. 44, No. 2, 2012.
[8] O. D. Lyantsev, T. V. Breikin, G. G. Kulikov, V. Y. Arkov, On-line Performance Optimisation of Aero Engine Control System, *Automatica*, Vol. 39, 2003, pp. 2115–2121.
[9] O. D. Lyantsev, Synthesis of Nonlinear Optimal Real-Time Automatic Control Systems for Gas Turbine Engines, *Journal of Computer and Systems Sciences International*, Vol. 46, No. 3, 2007, pp. 484–492.
[10] A. Kreiner, K. Lietzau, The Use of Onboard Real-Time Engine Models for Jet Engine Control, *MTU Aero Engines*, Germany (2003).
[11] M. Henriksson, T. Gronstedt, C. Breitholtz, Model-based on-board Turbofan Thrust Estimation, *Control Engineering Practice*, 2011, Vol. 19, pp. 602–610.
[12] R. H. Luppold, Patent: Apparatus and Method for Reducing Aircraft Fuel Consumption, US 20110202251, 2011.
[13] X. Dai, T. Breikin, Z. Gao, H. Wang, Dynamic Modelling and Robust Fault Detection of a Gas Turbine Engine, *American Control Conference*, Seattle, Washington USA, June 11-13, 2008.
[14] L. Ljung, System Identification: Theory for the User, 2nd edn. Upper Saddle River NJ: PTR Prentice Hall, 1999.
[15] O.D. Lyantsev, T.V. Breikin, G.G. Kulikov, V.Y. Arkov, On-line Performance Optimisation of Aero Engine Control System, *Automatica*, Vol. 39, 2003, pp. 2115–2121.
[16] D.W. Clarke, C.Mohtadi and P.S. Tuffs, Generalized Predictive Control – Part I. The Basic Algorithm, *Automatica*, Vol. 23, No 2, 1987, pp. 137–148.
[17] W. S. Levine, The Control Handbook, *CRC–IEEE press*, 2000, p. 808.
[18] J. K. Huusom, N. K. Poulsen, S. B. Jorgensen and J. B. Jorgensen, Tuning Methods for Offset Free MPC based on ARX Model Representations, *American Control Conference*, Baltimore, MD, USA, June 30–July 02, 2010.
[19] L. C. Andrews, Special Functions of Mathematics for Engineers, *SPIE Press*, 1998, pp. 110–111.
[20] http://jsbsim.sourceforge.net/