UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

A Multiobjective Trajectory Optimisation Method for Planning Environmentally Efficient Trajectories

Quintain McEnteggart
Cranfield University
Cranfield
Bedfordshire, MK43 0AL
Email: q.mcenteggart@cranfield.ac.uk

James Whidborne
Cranfield University
Cranfield
Bedfordshire, MK43 0AL
Email : j.f.whidborne@cranfield.ac.uk

***Abstract*—This paper proposes a multi-objective trajectory optimization method to be used in the planning of environmentally efficient commercial aircraft trajectories. The problem of finding environmentally efficient trajectories is treated as an optimal control problem that is solved by applying a direct method of trajectory optimisation. The method involves an inverse trajectory parameterisation technique, methods for the calculation of environmental objectives, and the use of a multi-objective version of the Differential Evolution algorithm. The principal benefit of the method relative to previous work is that it allows the fast generation of Pareto optimal fronts between several competing objectives. This allows the decision maker to make informed decisions about potential trade-offs between different environmental goals.**

I. Introduction

Over the last decade global passenger air traffic has increased by more than 45%, with similar levels of growth projected for the coming decade [1]. It has long been recognised that aviation benefits society, as a generator of wealth, and as an enabler to the exchange of ideas and culture between nations. However, increasing levels of air traffic will continue to impact the environment, with rising levels of aircraft emissions and higher numbers of people exposed to significant levels of aircraft noise [2].

The Advisory Council for Aviation Research in Europe (ACARE) is a group of representatives from the European commission, member states, industry and academia tasked with influencing the direction of European aviation research and development. The council, recognising the impact of aviation on the environment, has proposed to meet the challenges of sustainable aviation through the application of research and technology. To achieve this, it has created a strategic agenda for European aviation research with associated goals to be achieved by the year 2020. The environmental targets proposed by ACARE are (from a 2000 baseline) [3]

- Reduce fuel consumption and carbon dioxide (CO2) emissions by 50% per passenger kilometre.
- Reduce NOx emissions by 80%.
- Reduce perceived noise by 50% .

Of the targets, ACARE has proposed in its roadmap that 5-10% of the CO2 reduction be achieved through improved aircraft operations and air traffic management. This target is in line with the Single European Sky Air Traffic Management Research (SESAR) programme goal of reducing Air Traffic Management (ATM) related CO2 emissions by 10% per flight (from a 2005 baseline) [4]. Although no quantitative targets have been set for ATM related noise reduction, both initiatives recognise the role that improving aircraft operations has to play in reducing noise impact on communities around airports.

A particular focus of Air Traffic Management research related to delivering the ACARE goals is the design and delivery of trajectories that minimise environmental impact, referred to as green trajectories. Green trajectories, in the form of Continuous Descent Approaches (CDA), have already shown promise in delivering emissions and noise reductions to airport terminal areas [5]. The planning and optimization of green trajectories has been the subject of a number of theoretical studies. The Sourdine project, using expert analysis, developed a series of recommended noise abatement procedures for airport arrival and departures [6]. The trajectories were optimized to reduce noise under the flight-path for representative medium narrow-body and large wide-body aircraft. Simulations subsequently showed that large scale adoption of the Sourdine procedures could lead to significant reductions in noise footprints relative to conventional trajectories, although runway rates were adversely affected [7], [8]. Visser et al [9], [10] posed the problems of finding green arrival and departure trajectories as optimal control problems. This work used the direct collocation technique proposed by Hargreaves and Paris [11] to calculate optimal trajectories in terms of fuel burn and awakenings. Hebly et al [12] also used a collocation method and a weighted-sum cost function based on fuel burn and awakenings to optimise a Required Navigation (RNAV) departure procedure. Prats et al [13], again using a collocation method, recognised that the calculation of green trajectories can require the consideration of several conflicting optimisation criteria. To account for this, a lexiographic method for multi-objective optimisation was implemented. For the method, a hierarchy of importance for the objectives was established prior to the simulation. In this case, the method finds the minimum for the first objective and then seeks reductions in subsequent objectives providing

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

they do not increase the values of the objectives higher in the hierarchy. What results is a single Pareto optimal point that lies at the extreme of the first objective in the hierarchy.

When multiple conflicting objectives exist for an optimization problem, a single Pareto optimal point, or the minimum of a single objective, offer the Decision Maker (DM) very little information about potential tradeoffs between the optimization objectives. In environmental trajectory optimization, it is usually desirable to have multiple Pareto optimal points, where the tradeoffs in objectives, such as noise and fuel burn, can be assessed relative to each other.

The aim of this paper then is to solve the general multi-objective trajectory optimization problem for several types of environmental objectives, and to find the Pareto optimal set between minimised objectives. To do this, it is proposed to convert the optimal control problem to a Non Linear Programming (NLP) problem using a direct method of trajectory optimization that can be combined with aircraft emission and noise methods for the calculation of objectives. It is then proposed to solve the NLP problem for a Pareto optimal set using a stochastic evolutionary algorithm as the NLP solver. It is intended that the proposed method be useful to air traffic route designers and to airline flight planners as an approach that can be used to predict and optimize the environmental impact of commercial aircraft trajectory operations.

II. Problem

Stated generally, the multi-objective trajectory optimization problem can be stated as the problem of minimising an array of scalar objective functions

$$\min_{\mathbf{z},\mathbf{u}}[f_1(\mathbf{z},\mathbf{u}), f_2(\mathbf{z},\mathbf{u}), \ldots, f_q(\mathbf{z},\mathbf{u})]^T \tag{1}$$

where, in this work, the aircraft states are $\mathbf{z} = [x(t), y(t), h(t), v(t), \gamma(t), \chi(t)]^T$, the aircraft position is $\mathbf{r} = [x(t), y(t), h(t)]^T$, $v(t)$ is the airspeed, $\gamma(t)$ is the flight path angle and where $\chi(t)$ is the heading angle. The aircraft controls are then $\mathbf{u} = [T(t), n(t), \phi(t)]^T$ where $T(t)$ is thrust, $n(t)$ is load factor and $\phi(t)$ is the bank angle. The individual objectives of the array are then minimised subject to the inequality constraints to be satisfied

$$\mathbf{c}_i(\mathbf{z},\mathbf{u}) \leq 0 \tag{2}$$

When solving a multiobjective optimization problem, where n is the number of objectives, the objective vector $\mathbf{s}^*$ is Pareto optimal, if there does not exist another objective vector $\mathbf{s} \in S$ where $s_i \leq s_i^*$ for all $i = \{1, \ldots, n\}$ and where $s_j^* < s_j$ for at least one index of j, $j \in \{1, \ldots, n\}$. A Pareto front is a set of Pareto optimal solutions. A set is Pareto optimal if each solution in the set is Pareto optimal. For conflicting objectives, the Pareto optimal set should identify the extremes of the objectives, most fuel efficient and shortest path for instance, and allow for tradeoffs between objectives to be examined. This facilitates the search for solutions that offer the best balance between objectives.

III. Transcription

To solve the general optimal control problem it may be converted to a finite dimensional numerical optimization problem. Methods for transcription to a numerical problem are classified in Betts [14] as indirect and direct. Indirect methods involve forming the Hamiltonian of the system, estimating the costate variables, and finding a root of the two point boundary value problem. Direct methods involve discretising the optimal control problem and solving for the states and controls at a series of dividing nodes. Direct methods are preferred for this work because they do not require the definition of costate variables or constrained arcs and are therefore easier to apply to the problem.

Direct collocation methods use polynomials to parameterise the states and controls of the aircraft. Hargreaves and Paris [11] used piecewise cubic polynomials, while Fahroo et al [15], [16] and Benson et al [17] have used orthogonal polynomials with several different forms of collocation points. A significant drawback of collocation methods however, is that they can require large numbers of varied parameters. When applied to problems with large numbers of Air Traffic Control (ATC) constraints on aircraft position, speeds and rates of climb/descent, collocation methods may be very slow to evolve, first from infeasible to feasible solutions and then from feasible to a point along the global Pareto front. Yakimenko proposed an inverse method where the position states of the aircraft and their derivatives are parameterised using 7th degree polynomials [18]. Controls are then determined by inverting the state equations. The method significantly reduces the number of optimization variables required by analytically determining the polynomial coefficients from the prescribed states and controls at the boundaries ($t = 0$ and $t = t_f$). Instead of parameterising by time, Yakimenko adopted Taranenko's method of parameterising the polynomials by τ, creating a virtual arc $\tau \in [0, \tau_f]$. The relationship between time t and τ is defined as $\lambda = d\tau/dt$. The use of the relationship parameter λ allows the definition of aircraft velocity using a separate reference function, enabling the creation of a virtual speed profile along the trajectory path of the aircraft. The method, termed the Inverse Dynamics in the Virtual Domain (IDVD) method, is a fast trajectory optimization method and has been considered for real time implementation [18], [19]. The IDVD method was adopted for this work because the small number of varied parameters allowed the Differential Evolution (DE) algorithm used in this work to quickly evolve decision vectors through the differential mutation and crossover mechanisms. The DE algorithm was chosen because it had the potential, when combined with the IDVD method, to efficiently converge on detailed global Pareto fronts.

For the IDVD method, the flat earth Cartesian coordinates $r_j (j = 1, 2, 3)$ and their derivatives are parameterised from

the reference function (3) and its derivatives,

$$r(\tau)_j = \sum_{k=0}^{7} \frac{a_j k \tau^k}{\max(1, k(k-1))} \tag{3}$$

The coefficients of the polynomials are determined analytically from the coordinates and their derivatives at the boundaries ($\tau = 0$ and $\tau = \tau_f$) by making the coefficients the subjects of the following set of linear equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{\tau_f^2}{2} & \frac{\tau_f^3}{6} & \frac{\tau_f^4}{12} & \frac{\tau_f^5}{20} & \frac{\tau_f^6}{30} & \frac{\tau_f^7}{42} \\ 0 & 1 & \tau_f & \frac{\tau_f^2}{2} & \frac{\tau_f^3}{3} & \frac{\tau_f^4}{4} & \frac{\tau_f^5}{5} & \frac{\tau_f^6}{6} \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 & \tau_f^4 & \tau_f^5 \\ 0 & 0 & 0 & 1 & 2\tau & 3\tau^2 & 4\tau^3 & 5\tau^4 \end{bmatrix} \begin{bmatrix} a_{j0} \\ a_{j1} \\ a_{j2} \\ a_{j3} \\ a_{j4} \\ a_{j5} \\ a_{j6} \\ a_{j7} \end{bmatrix} = \begin{bmatrix} r_{j0} \\ r'_{j0} \\ r''_{j0} \\ r'''_{j0} \\ r_{jf} \\ r'_{jf} \\ r''_{jf} \\ r'''_{jf} \end{bmatrix} \tag{4}$$

To modify the polynomials, the variables iterated by the solver are then $\Xi = [x'''_{0,f}, y'''_{0,f}, h'''_{0,f}, v''_{0,f}, \tau_f]$. To transform the polynomials to the system dynamics, a point mass model is used. Therefore the state and controls are determined by inverting the following equations:

$$\begin{aligned} \dot{x} &= v\cos\gamma\cos\chi, & \dot{v} &= \frac{T-D}{m} - g\sin\gamma \\ \dot{y} &= v\sin\chi\cos\gamma, & \dot{\chi} &= \frac{gn\sin\phi}{v\cos\gamma} \\ \dot{h} &= v\sin\gamma, & \dot{\gamma} &= \frac{g}{v}(ncos\phi - \cos\gamma) \end{aligned} \tag{5}$$

The drag D is modelled with the aid of the BADA drag polars [20], aircraft mass is m, g is gravitational acceleration and n is the load factor. Conversions between the virtual and time domain are achieved by:

$$\begin{aligned} \dot{r} &= \lambda r'; \quad \ddot{r} = \lambda(r''\lambda + r'\lambda') \\ \dddot{r} &= \lambda^3 r''' + 3\lambda^2\lambda' r'' + (\lambda^2 + \lambda\lambda'^2)r' \end{aligned} \tag{6}$$

Once the state and control histories are determined from τ_o to τ_f, the following constraints are applied to ensure that they lie between defined limits determined with the aid of the BADA database [20]:

$$\begin{gathered} T \in [T_{min}, T_{max}] \quad n \in [n_{min}, n_{max}] \quad |\phi| \leq |\phi_{max}| \\ v \in [1.2v_{stall}, v_{max}] \quad \dot{v} \in [\dot{v}_{min}, \dot{v}_{max}] \quad \gamma \in [\gamma_{min}, \gamma_{max}] \end{gathered}$$

For the BADA parameters, Eurocontrol compensate for inaccuracies in the parameter estimation process by defining limits for both aircraft thrust and acceleration.

IV. Waypoint Constraints

For commercial aircraft trajectory optimization, constraints imposed by the operating environment must be considered. Calculated trajectories must be able to adhere to ATC constraints imposed by airspace sectorization, procedures and traffic flow corridors. Typically, operating environment and procedural restrictions manifest as constraints on the height, speed or path of the flight, or some combination of the three. Therefore a simple five dimensional constraints model is developed.

Waypoint fixes are defined by the user in 2 dimensions as $\mathbf{W}_r = [W_x, W_y]$. The aircraft's trajectory path can then be constrained to fly over the 2D fix. For each waypoint fix, the minimum distance between the trajectory path and the fix position is calculated as a 2D distance d_{min}, with a corresponding minimum time $t_{d_{min}}$,

$$d_{min} := \min_{t\in[t_0,t_f]} d(t) \text{ where } d(t) = \|r_{2D}(t) - \mathbf{W}_r\|$$

$$t_{d_{min}} := \min t \;\; s.t. \;\; d(t) = d_{min}$$

The aircraft is then constrained to fly within a distance radius of the centre point of the fix, where $\hat{d}$ is the upper constraint on d_{min},

$$\mathbf{c}_1(d_{min}) = d_{min} - \hat{d}$$

The aircraft can also be constrained to cross the fix at a specified height, speed and arrival time. The cross above constraints, $\underline{h}, \underline{v}, \underline{t}$, and cross below constraints, $\overline{h}, \overline{v}, \overline{t}$, constrain the minimum and maximum heights, speeds and time of the aircraft crossing the waypoint. The minimum and maximum can be constrained simultaneously to create height,speed and time windows at the waypoint.

$$\mathbf{c}_2(\mathbf{h}, \mathbf{v}, \mathbf{t}, t_{d_{min}}) = \begin{bmatrix} h(t_{d_{min}}) - \overline{h}_{t_{min}} \\ \underline{h}_{t_{min}} - h(t_{d_{min}}) \\ v(t_{d_{min}}) - \overline{v}_{t_{min}} \\ \underline{v}_{t_{min}} - v(t_{d_{min}}) \\ t_{d_{min}} - \overline{t}_{min} \\ \underline{t}_{min} - t_{d_{min}} \end{bmatrix}$$

V. Environmental models

For the calculation of civil aircraft noise impact local to airports, the most commonly used method in the field is the Noise Power Distance (NPD) method [21]. The NPD method utilises, for a number of noise metrics, tables of empirical data that relate the noise level calculated on the ground to the power utilised by the aircraft and the distance from the aircraft to the noise assessment point. Specifically, the noise level at a point is calculated as

$$noise\,level = f_n(P(t), d(t), \boldsymbol{\beta}), \quad \forall t \in [t_o, t_f] \tag{7}$$

where $P(t)$ is power, $d(t)$ is slant distance between the aircraft and the assessment point, $\boldsymbol{\beta}$ is a set of segment level correction terms, and t_o and t_f are respectively the start and end times of the trajectory. Power, in this instance, is corrected net thrust. The noise model chosen for use in this work is the Integrated Noise Model (INM). INM is a model developed by the Federal Aviation Administration (FAA) to assess the impact of civil aircraft noise on

communities local to airports. INM version 7 [22] is fully compatible with ECAC Doc 29, [21] guidance that provides a standardized methodology for the computation of noise contours around civil airports. INM is able to calculate several types of aircraft noise impact metrics, including the maximum A-weighted sound pressure level L_{Amax}, the single event exposure metrics Sound Exposure Level (SEL) and The Effective Perceived Noise Level $(EPNL)$. INM also allows the calculation of noise metric contours that can be used to calculate population exposures and awakenings.

For the calculation of aircraft emissions, the emissions CO2, water vapor (H2O) and sulphur oxides (SOx) are calculated using direct multipliers on fuel burn, such that their rate of emissions is:

$$\dot{e}(t) = f_e(F(t), \boldsymbol{\alpha}) \tag{8}$$

where $F(t)$ is the rate of fuel burn and $\boldsymbol{\alpha}$ is a set of fuel burn multipliers. For the work presented in this document, rate of fuel burn is calculated using the BADA fuel flow model [20], where $F(t)$ is a function of thrust and the thrust specific fuel consumption factor η such that:

$$F(t) = f_F(P(t), \eta) \tag{9}$$

The emissions for hydrocarbons (HC), carbon (CO), and oxides of nitrogen (NOx) are calculated using the Boeing Fuel Flow Methodology (BFFM) [23]. The ICAO emissions databank contains empirical information for certified engines that relate fuel burn to emissions indices at 4 different engine thrust settings. The BFFM offers a procedure for correcting the data for atmospheric conditions and for interpolating the data to calculate emissions of HC, CO, and NOx from fuel flow such that their rate of emissions is:

$$\dot{e}(t)_{BFFM} = f_{BFFM}(F(t), \mathbf{EI}) \tag{10}$$

where EI is a set of of emissions indices.

VI. Stochastic solver

NLP algorithms can be applied in the iterative solution of a wide array of optimization problems. NLP algorithms can be generally classified as direct search (no derivatives), gradient search or stochastic. Derivative based gradient solvers require the definition or estimation of the derivatives of the objective function, while derivative free optimisers compare only the objective function value. To reach a global optimum it helps if gradient and derivative free optimisers are initialised where the objectives and constraints are convex. Stochastic solvers however tend to find it easier to escape local minima. Like derivative free algorithms, stochastic algorithms compare only objective values between iterations but use probability functions in guiding the search for the optimum solution. More details on stochastic optimization methods can be found in [24]. Evolutionary algorithms are a popular type of stochastic method and are well suited to solving multi objective optimization problems. Evolutionary algorithms, at each step of an optimization, maintain a population of solutions. This, allows the algorithms to simultaneously explore different parts of the solution space to identify decision vectors that provide the minimum for each objective and also the Pareto optimal points that form a front between the minimums.

The stochastic solver chosen for this work was Differential Evolution [25]. The DE algorithm utilises the mechanism of differential mutation. Differential mutation is a self adaptive mechanism where 3 population vectors are randomly selected from the parent generation and the scaled difference between 2 of the vectors is added to the third. The DE algorithm has proved to be a simple yet effective method for handling global optimization problems [26]. DE requires only 3 configuration parameters for calibration and has shown promising results when used with the inverse dynamics method chosen for use in this research. Drury [27] tested the performance of the Inverse Dynamics method with a number of popular Non Linear Programming (NLP) algorithms. For 2000 different sets of boundary values, the DE algorithm achieved a convergence of 99.8% with a relative optimality score of 94%, outperforming all of the other NLP algorithms. A multiobjective variant of DE is defined in [25] where Lampinen's dominance based method for constrained optimization [28] is used to drive solutions towards a Pareto front. Madavan [29] subsequently showed that multiobjective DE could be supplemented with a nondominated sorting procedure and the crowding distance metric developed by Deb et al [30] for the Nondominated Sorting Genetic Algorithm (NSGAII). The result was a fast and powerful method that combined the self organising mechanism of differential mutation with the elitism and the diversity preservation of the NSGAII algorithm.

A. Main

The variant of DE chosen for this work was DE/rand/1/Bin [25]. The algorithm initialises by generating a population of random individuals between the user specified upper b_U and lower b_L parameter bounds. For each individual in this population, a candidate vector is created that is the algebraic combination of 3 parent vectors, further modified by crossover between the candidate and the target vectors. Each candidate vector is placed in an offspring population. Once a candidate population has been generated that is the same length as the parent population, it is appended to the parent population. The combined population is then sorted into fronts using nondominated sorting. For the production of the new population for the next generation, the size of the combined population is reduced to the initial population size by truncation. Each solution in the solution set is sorted into fronts (F) depending on the number of other solutions in the set each is dominated by. All fronts from the first nondominated front (F_1) upwards , whose combined length is less than or equal to the initial population size are preserved for the next generation. The individuals in the final front are sorted according to their crowding distance value. Individuals with large crowding distance values are moved toward the start of the front and the individuals with

Algorithm 1 A fast, self-adaptive, elitist, diversity preserving, multi-objective evolutionary algorithm

```
G = 1                                                  First generation
P = ∅                                                  Initialise initial population
for i := 1 to NP                                       Create NP real valued vectors
    p_i = ∅                                            Initialise population vector
    for j := 1 to D                                    Each population vector contains D real parameters
        x_{j,i,G} = rand(0,1)(b_{j,U} − b_{k,L}) + b_{j,L}   Create parameters within bounds
        p_i = ∪{x_{j,i,G}}                             Add the parameter to the population vector
    end
    P = ∪{p_i}                                         Add the population to the set of populations
end

while G < G_max                                        While current generation is less than final generation
    Q = ∅                                              Initialise offspring population
    for each p_{i,G} ∈ P_G                             For each population vector in the population
    r_1, r_2, r_3 ∈ {1, ..., NP}                       Select 3 random indexes
    F_S ∈ [0, 1]                                       Scale factor
    v_{i,G} = p_{r_1,G} + F_S(p_{r_2,G} − p_{r_3,G})   Create mutant population with Differential mutation
    CR ∈ [0, 1]                                        DE crossover parameter CR
    k ∈ {1, ..., D}                                    Random parameter index
    q_i = ∅                                            Initialise candidate vector
    for each x_{j,G} ∈ p_{i,G}                         For each parameter in the vector
        r = rand(0, 1)
        if r <= CR ∨ j == k                            Crossover between parent and the mutant vectors
            u_{j,G} = v_{j,G}
        else
            u_{j,G} = x_{j,G}
        end
        q_{i,G} ∪ {u_{j,G}}                            Add parameter to candidate vector
    end
    Q_G ∪ {q_{j,G}}                                    Add candidate vector to offspring population
    end
    R_G = P_G ∪ {Q_G}                                  Combine parent and offspring population
    F = fast nondominated sort(R_G)
    P_{G+1} = ∅ and l := 1                             All nondominated fronts of R_G
    while |P_{G+1}| + |F_l| ≤ |P_G|                    Until the new population P_{G+1} is filled
        crowding distance assignment(F_l)              Calculate the crowding distance in F_l
        P_{G+1} = P_G ∪ {F_l}                          Include the l^th nondominated front in the new population
        l = l + 1                                      Check the next front for inclusion
    end
    sort(F_l : F_end, ≺)                               Sort final front in descending order using domination
    P_{G+1} = P_{G+1} ∪ F_L[1 : (N − |P_{G+1}|)]       Truncate final front if required
    G = G + 1                                          Increment generation counter
end
```

small crowding distance values are moved towards the rear. If inclusion of the final front results on a population size greater than the initial population size, then the final front is truncated. This ensures that the solutions in the final front with the greatest solution diversity are retained for the next generation.

B. Nondominated Sorting

In general, for 2 feasible solutions where $\mathbf{p}, \mathbf{q} \in S$, $\mathbf{p}$ dominates $\mathbf{q}$ $(\mathbf{p} \prec \mathbf{q})$ if $\forall k : f_k(\mathbf{p}) \leq f_k(\mathbf{q}) \wedge \exists k : f_k(\mathbf{p}) < f_k(\mathbf{q})$ Nondominated sorting involves using domination to rank each solution into fronts that are sets of solutions with equal dominance ranking. Solutions that are not dominated by any other solutions are assigned to the first front F_1, Solutions that are dominated by 1 other individual will appear on the next front F_2, and so on until all solutions are assigned to the appropriate front.

C. Crowding Distance

The crowding distance metric is used to measure the distance along the same nondominated front from one solution to the 2 adjacent solutions. For each objective function, the greatest and smallest objective values are assigned an infinite crowding value, preserving the boundary value individuals. For each intermediate individual, its proximity to other individuals is determined by taking the normalised difference between the solutions either side of that solution. When this measure is summed over all individual's objective functions, a measure of the closeness between solutions is reached.

D. Selection

Unlike many Evolutionary Algorithms, where parent and child populations are compared to each other for fitness, for the NSGA-II method, the parent and child populations are appended to each other, and all selection occurs within the same population. Fitter individuals as determined by their nondomination rank and crowding distance are moved to the

front of the population, while less fit individuals are moved to the back of the population and may be subjected to truncation. As shown in Algorithm 2, domination is determined by the constraints, rank, and crowding distance of each individual. In Algorithm 2, the solution **i** dominates the solution **j** if both are feasible and **i** has either a lower nondominated rank ($rank$) or has a greater crowding distance value ($distance$) at the same nondominated rank. If **i** is feasible and **j** is infeasible then **i** dominates. If both solutions are infeasible, then the individual with the lowest overall constraint violation dominates.

Algorithm 2 Selection

$$\zeta(\mathbf{i}) = \sum_{k}^{m} max[0, g_k(\mathbf{i})]$$

$$\zeta(\mathbf{j}) = \sum_{k}^{m} max[0, g_k(\mathbf{j})]$$

$$\mathbf{i} \prec \mathbf{j} \ \ if \begin{cases} \begin{cases} \zeta(\mathbf{i}) \leq 0 \wedge \zeta(\mathbf{j}) \leq 0 \\ \wedge \\ i_{rank} < j_{rank} \\ \vee \\ [i_{rank} = j_{rank}] \wedge [i_{distance} > j_{distance}] \end{cases} \\ \vee \\ \begin{cases} \zeta(\mathbf{i}) \leq 0 \\ \wedge \\ \zeta(\mathbf{j}) > 0 \end{cases} \\ \vee \\ \begin{cases} \zeta(\mathbf{i}) > 0 \\ \wedge \\ \zeta(\mathbf{i}) \ < \zeta(\mathbf{j}) \end{cases} \end{cases}$$

VII. Results

A departing aircraft scenario was created to demonstrate the multi objective trajectory optimization method. In the scenario, a commercial aircraft is required to climb from an initial climb point below 500ft at the west of a large population centre, to an en-route connection point lying at 20,000ft on the far side of the population centre. The commercial aircraft simulated was the medium narrow-body Airbus A321 aircraft with twin International Aero Engine V2530 engines. The population was artificially created for the scenario, and consisted of 1.5 million people evenly distributed over an area of 45000 hectares. In addition to the constraints from Section III, the bank angle ϕ and the minimum climb gradient below 1000ft were constrained to $0(rad)$ and 12% respectively. For the first scenario the objectives chosen were the greenhouse gas Carbon Dioxide and the population enclosed within the 70 dB(A) Sound Equivalent Level footprint contour. SEL was used here as it forms the basic 'building block' of the Lden (Day-Evening-Night Average Sound Level) and Ldn (Day-Night Average Sound Level) contour calculations used to asses the community impact of aircraft noise.

Fig. 1 shows a Pareto front between the minimums of the two objectives. It can be seen from the front that there is a trade-off of approximately 900kg of CO2 between the most CO2 optimal trajectory and the most noise optimal trajectory. Similarly there is a trade-off of approximately 300,000 exposed people between the most noise optimal trajectory and the most CO2 optimal trajectory. Fig. 2 shows the trajectories for the two minimums. The aircraft trajectory for the minimum CO2

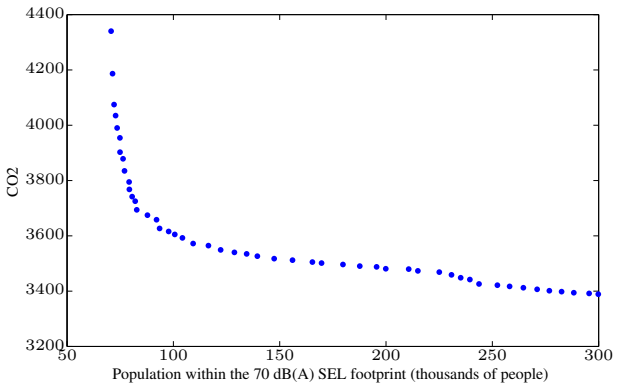

Fig. 1. Pareto front between CO2 emissions and the population within the 70dB(A) footprint

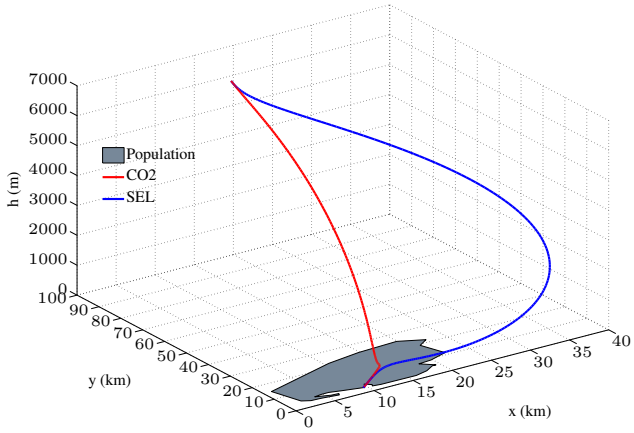

Fig. 2. Minimum noise and CO2 trajectories

objective, after climbing out to 1000ft takes a direct route over the population area to the target end point, minimising excess track miles, fuel burn and therefore CO2. The trajectory for the minimum noise objective, initially progresses directly to the east, avoiding over-flying the majority of the population area and therefore minimising the population exposed to noise. It can be seen from Fig. 2 and from Fig. 3 that the noise optimised trajectory climbs to a height of 1500ft where it reduces thrust and accelerates to zero flap speed while passing over a population region near to the airport. On clearing the population region, the aircraft is turned to the target end point while thrust is increased gradually to maximum climb thrust and the aircraft accelerates to en-route climb speed. The trajectory produced mimics closely the Sourdine close-in noise abatement procedure as both involve an initial climb at full

thrust followed by acceleration at reduced thrust and a gradual power increase.

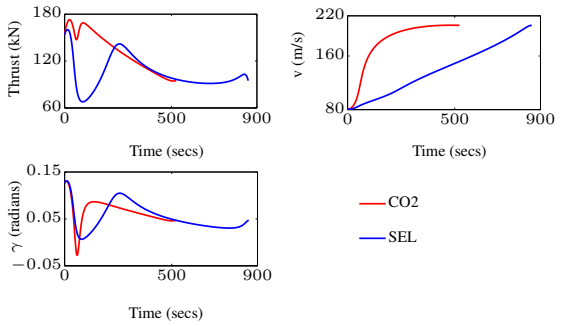

Fig. 3. Thrust, speed and flight path angle histories for the minimum noise and CO2 trajectories

A second scenario was created to examine the trade-offs between minimising for noise near to the airport and minimising for noise farther away from the airport. Boundary values remained as in the first scenario, however, all solution trajectories were constrained via the constraints in Section IV to lie along a common x, y ground path shown in Fig. 4.

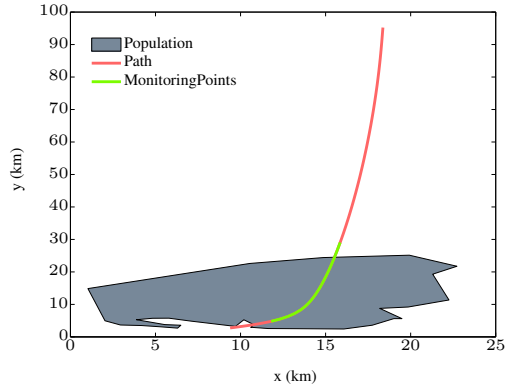

Fig. 4. Constrained x, y path with noise monitoring region

All improvements in objectives were therefore achieved through changes to the vertical trajectory. A region of noise monitoring points, shown in green in Fig. 4, were then placed at 1000m intervals from a distance of 3000m to 30,000 metres from start of roll. The region of 3000m to 15000m was defined as the close-in region and the region of 15,000 to 30,000 metres was defined as the far region. Two objectives were then the subject of the optimization, Average EPNL in the close-in region, and average EPNL in the far region. Fig. 5 shows the Pareto front between the average EPNL in the near region and average EPNL in the far region. It can be seen that there is a 2-3 EPNdB average EPNL tradeoff between optimising trajectories for the different regions.

It can be seen from Fig. 6 and from Fig. 7 that the near region optimization causes the aircraft to cutback its thrust and to assume a shallower flight path angle climb sooner than the far region optimization. As can be seen in Fig. 8, this results in

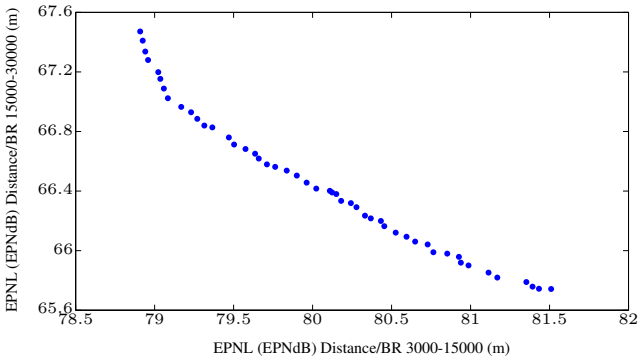

Fig. 5. Pareto front between close-in and far region average EPNL

a reduction in EPNL noise levels along the trajectory centreline at distances running from 3000 to 11000 metres. By contrast the far region optimization results in the aircraft climbing as high as possible using maximum thrust and flight path angle prior to the beginning of the far region. Once the far region is reached, the aircraft reduces thrust and climb angle, but the greater altitude attained allows the aircraft to fly higher over the far region increasing the noise attenuation distance and reducing noise levels on the ground. Once the far region has been passed, thrust is increase to accelerate the aircraft to enroute speed. Fig. 8, shows the corresponding higher close-region noise levels and lower far region noise levels that result from this trajectory.

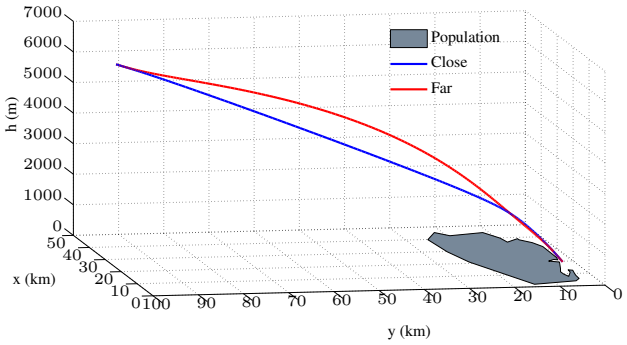

Fig. 6. Minimum close-in noise and far region noise trajectories

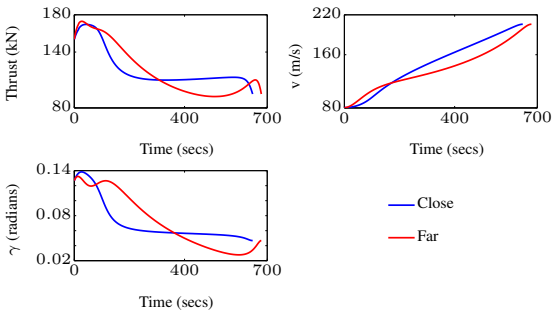

Fig. 7. Thrust, speed and flight path angle histories for the minimum close-in and far noise trajectories

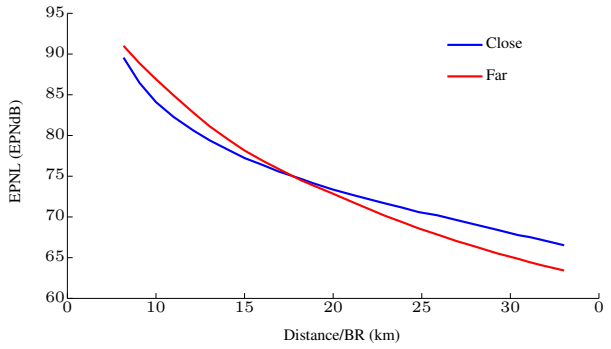

Fig. 8. Minimum close-in noise and far region noise trajectories

VIII. Conclusion

As can be seen from Section VII, under many circumstances, there is no one trajectory that minimises all environmental costs. Therefore when planning environmentally efficient trajectories, the trade-offs between the objectives must be considered. This work has proposed a multi-objective trajectory optimisation method that may be used to analyse the trade-offs between environmental objectives that arise when operating commercial aircraft in different ways. The work assumes the availability of an advanced Flight Management System (FMS) with auto-throttle capable of tracking the detailed trajectory solutions. However, each trajectory may be segmented into a smaller number of operational steps that would be suitable for a pilot to execute. Further work will involve the automated analysis of the Pareto fronts so that certain solutions from the front can be recommended to the user to aid their decision making. This is intended to be especially useful when analysing Pareto fronts between more than two objectives.

Acknowledgment

This work was completed thanks to funding support from Cranfield university. The authors would also like to thank Ken Lai from Cranfield University, Peter Hullah from Eurocontrol and Rick Drury from Flight Applications Limited for their help and guidance.

References

[1] Airbus, "Delivering the future: Global market forecast 2011-2030," 2011.
[2] ICAO, "Environmental report 2010," 2010.
[3] ACARE, "Strategic research agenda. volume 2," October 2002.
[4] European Commission, "European air traffic management master plan," March 2009.
[5] Committee on Aviation Environmental Protection, "Review of continuous descent approach (CDA) implementation and associated benefits," *Working Paper CAEP/7-WP/26*, 2006.
[6] Sourdine Consortium, "Study of optimisation procedures for decreasing the impact of noise (sourdine)," *Report D5*, 2001.
[7] NLR, "SourdineII noise results amsterdam schiphol," *D4-1-2b*, 2006.
[8] INECO, "SourdineII noise results madrid barajas," *D4-1-3b*, 2006.
[9] R. A. Visser, Hendrikus G. Wiljnen, "Optimization of noise abatement departure trajectories," *Journal of Aircraft*, vol. 38(4), p. 620627, 2001.
[10] ——, "Optimization of noise abatement arrival trajectories," *The Aeronautical Journal*, vol. 107(1076), p. 607615, 2003.
[11] C. Hargraves and S. Paris, "Direct trajectory optimization using nonlinear programming and collocation," in *Astrodynamics 1985*, vol. 1, 1986, pp. 3–12.
[12] S. Hebly and H. Visser, "Advanced noise abatement departure procedures: custom optimized departure profiles," in *AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI*, 2008.
[13] X. Prats, V. Puig, J. Quevedo, and F. Nejjari, "Lexicographic optimisation for optimal departure aircraft trajectories," *Aerospace Science and Technology*, vol. 14, no. 1, pp. 26–37, 2010.
[14] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, 1998.
[15] F. Fahroo and I. Ross, "Advances in pseudospectral methods for optimal control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.
[16] ——, "Direct trajectory optimization by a Chebyshev pseudospectral method," in *Proceedings of the 2000 American Control Conference, 2000.*, vol. 6. IEEE, 2000, pp. 3860–3864.
[17] D. Benson, G. Huntington, T. Thorvaldsen, and A. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance Control and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.
[18] O. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, 2000.
[19] C. Lai and J. Whidborne, "Real-time trajectory generation for collision avoidance with obstacle uncertainty," in *AIAA Guidance, Navigation and Control Conference, AIAA 2011-6598*, Portland OR, August 2011.
[20] A. Nuic, "User manual for the base of aircraft data," *(BADA) revision 3.7*, vol. 2010, p. 001, 2010.
[21] ECAC, "Standard method of computing noise contours around civil airports," *ECAC. CEAC Doc. 29*, 2010.
[22] J. Olmstead, G. Fleming, J. Gulding, C. Roof, P. Gerbi, and A. Rapoza, "Integrated noise model (inm) version 7.0 technical manual," *Report FAA-AEE-02-01, Office of Environment and Energy, Federal Aviation Administration*, 2002.
[23] S. Baughcum, T. G. Tritz, S. C. Henderson, and D. C. Pickett, *Scheduled civil aircraft emission inventories for 1992: Database development and analysis*. National Aeronautics and Space Administration, Langley Research Center, 1996.
[24] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.
[25] K. Price, R. Storn, and J. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer-Verlag New York Inc, 2005.
[26] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
[27] R. Drury, "Performance of NLP algorithms with inverse dynamics for near-real time trajectory generation," in *AIAA Guidance, Navigation, and Control Conference*, August 2011.
[28] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02.*, vol. 2. IEEE, 2002, pp. 1468–1473.
[29] N. Madavan, "Multiobjective optimization using a pareto differential evolution approach," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02.*, vol. 2. IEEE, 2002, pp. 1145–1150.
[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.