González-Arribas, D., Soler, M., López-Leonés, J., Casado, E. & Sanjurjo-Rivo, M. (2018). Automated optimal flight planning based on the aircraft intent description language. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(3), pp. 928–948.

DOI: 10.1177/0954410017751990

Keywords
multiphase optimal control, aircraft trajectory, ATM, Flight planning, Trajectory-based operation

Journal name
000(00):1–13
©The Author(s) 2010
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI:doi number
http://mms.sagepub.com

Automated Optimal Flight Planning based on the Aircraft Intent Description Language

Daniel González-Arribas [*]
Department of Bioengineering and Aerospace Engineering, Universidad Carlos III de Madrid, 28911 Madrid, Spain

Manuel Soler [†]
Department of Bioengineering and Aerospace Engineering, Universidad Carlos III de Madrid, 28911 Madrid, Spain

Javier López-Leonés [‡]
Boeing R&T Europe

Enrique Casado [§]
Boeing R&T Europe

Manuel Sanjurjo-Rivo
Department of Bioengineering and Aerospace Engineering, Universidad Carlos III de Madrid, 28911 Madrid, Spain [¶]

Abstract

The future Air Traffic Management system is to be built around the notion of trajectory-based operations. It will rely on automated tools related to trajectory prediction in order to define, share, revise, negotiate, and update the trajectory of the aircraft before and during the flight, in some case, in near real time. This paper illustrates how existing standards on trajectory description such as the Aircraft Intent Description Language (AIDL) can be enhanced including optimisation

[*] Corresponding author; e-mail: dagonza@ing.uc3m.es
[†] {masolera@ing.uc3m.es}
[‡] {javier.lopezleones@boeing.com}
[§] {enrique.casado@boeing.com}
[¶] {msanjurj@ing.uc3m.es}

capabilities based on numerical optimal control. AIDL is a formal language that has been created in order to describe aircraft intent information in a rigorous, unambiguous and flexible manner. It has been implemented in a platform for a modular design of the trajectory generation process. A case study is presented to explore its effectiveness and identify the requirements and needs to generate optimised aircraft intents with higher automation and flexibility. Preliminary results show the suitability of numerical optimal control to design optimised aircraft intents based on AIDL.

Nomenclature

δ_L	Elevator setting
δ_T	Throttle level
δ_{HL}	High lift configuration parameter
δ_{LD}	Coordinated ailerons and rudder deflection
δ_{LG}	Landing gear configuration parameter
δ_{SB}	Speed brakes configuration parameter
γ	Flight path angle
λ	Longitude
$\mathcal{S}$	Sequence of switching conditions
$\mathcal{X}_k$	Valid state set for phase k
μ	Bank angle
ϕ	Latitude
Φ_k	Meyer (terminal cost) term of phase k
ϕ_k	Path constraints function of phase k
ψ	Final conditions function
Σ	Set of differential-algebraic dynamic subsystems
Σ_k	Differential-algebraic dynamic subsystem for phase k
ϑ_k^{eq}	Equality interior point constraints of phase k
ϑ_k^{ieq}	Inequality interior point constraints of phase k
C_L	Lift coefficient
f_k	Dynamic function for phase k
g_k	Constraint function for phase k
h_e	Geodetic altitude

H_p	Pressure altitude
J	Cost functional
L_k	Lagrangian cost term of phase k
m	Mass
N	Number of phases
n_u	Dimension of the control space
n_x	Dimension of the state space
p	Vector of parameters
T	Thrust
t	Time
u	Control vector
x	State vector
x_I	Initial conditions vector
U_k	Valid control set for phase k
CAS	Calibrated Airspeed
CI	Cost index
TAS	True Airspeed

1. Introduction

The Air Traffic Management (ATM) system is responsible for safe, efficient and sustainable operation in civil aviation. During its lifetime, the ATM system has evolved from an early primitive structure composed by a set of simple operation rules to its current form, consisting of complex network of ATM layers, communication, navigation, and surveillance subsystems. A paradigm shift in the current ATM system is being pursued in order to: address the continuous growth of air traffic demand by increasing the capacity of the system; increase the competitiveness by reducing the overall cost of ATM services provision and the operational costs incurred by airline companies; and tackle the growing concerns over both the environmental impact and the level of safety of air transportation by reducing aviation environmental fingerprint and increasing safety, respectively (1; 2).

In the future ATM system, the trajectory will be put at the center of a new set of operating procedures that are collectively labeled as Trajectory-Based Operations (TBO) (3). These concepts will be employed to enhance capabilities in all aspects of the trajectory management (planning, sharing, agreeing and synchronising) and will lead to ATM performance improvements in terms of efficiency, capacity, safety and environmental impact.

A core concept in TBO is the idea of business trajectory. The business trajectory is defined as the planned trajectory that would, under the fulfilment of certain operational restrictions, meet best airline business interests, taking into consideration different stakeholders' preferences, e.g., Air Navigation Service Providers (ANSPs), Network Manager (NM), and airports. TBO require the ATM system to introduce profound innovations to enable the envisioned changes. Some of these include

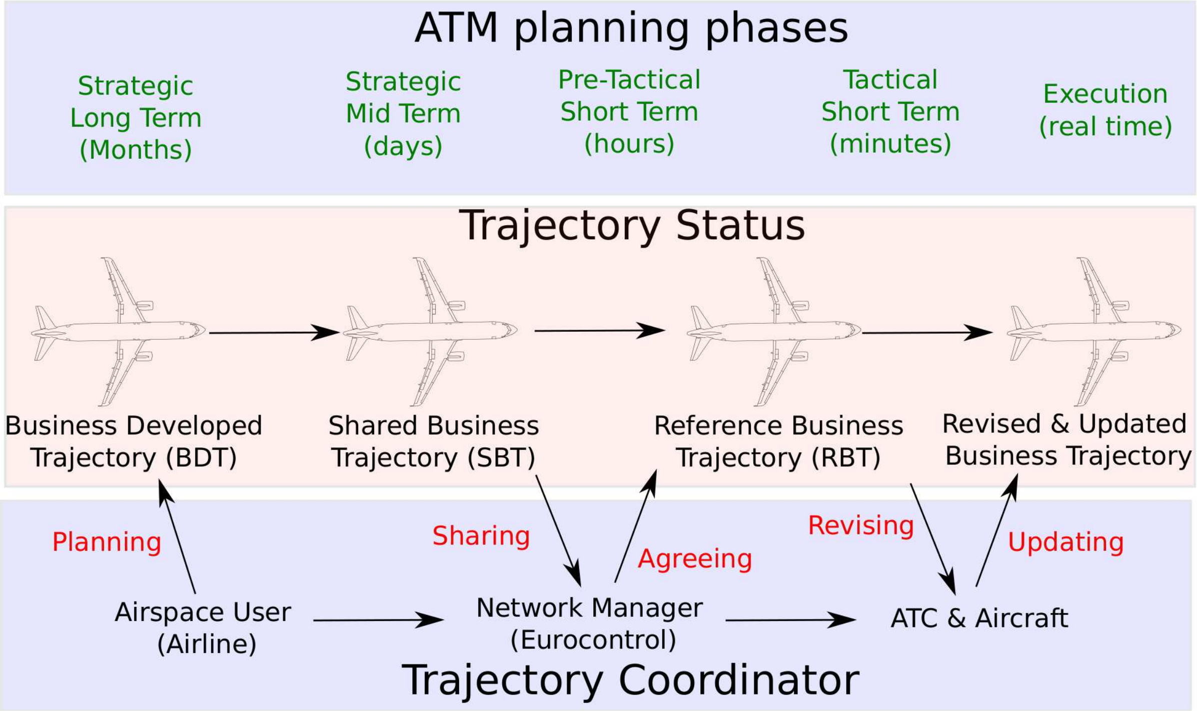

ATM planning phases
Strategic Long Term (Months)
Strategic Mid Term (days)
Pre-Tactical Short Term (hours)
Tactical Short Term (minutes)
Execution (real time)
Trajectory Status
Business Developed Trajectory (BDT)
Shared Business Trajectory (SBT)
Reference Business Trajectory (RBT)
Revised & Updated Business Trajectory
Planning
Sharing
Agreeing
Revising
Updating
Airspace User (Airline)
Network Manager (Eurocontrol)
ATC & Aircraft
Trajectory Coordinator

Fig. 1. Trajectory Based Operations concept.

collaborative decision-making processes, better data sharing and usage, and advanced decision support tools for human operators.

The business trajectory can be conceived as a contract between the above mentioned stakeholders, i.e., the airline, the ANSPs, the airports, and the NM (in the case of Europe, Eurocontrol). The lifecycle of the business trajectory is illustrated in Figure 1. In the strategic phase (long term), the airlines calculate their preferred trajectories, resulting in the business development trajectory (BDT). Eventually, in a process that goes from months before to the day of operation, all BDTs will be shared with the network manager (coordinating all ANSPs), becoming the shared business trajectory (SBT). With these SBTs, the ANSPs can assess airspace configurations, routes, and their allocation of resources. The network manager, having visibility of all SBTs (demand of flights) and ANSPs resources (capacity), can identify possible capacity and demand imbalances, approving, delaying or rerouting the different SBTs. After all the stakeholders' interests are taken into account and an adequate solution has been found, it will become the reference business trajectory (RBT). The airline will then plan to fly the RBT and the ANSPs/airports will provide ATM services for the flight according to the RBT. Afterwards, during flight execution, RBT might be impacted by factors such as for instance de-conflicting, real-time queuing, or weather hazards. Therefore, the RBT might be tactically revised, negotiated, and updated during execution.

Within this envisioned framework, trajectory prediction will rely on automated tools that require formal and structured descriptions of flight plans and trajectories that will be processed by decision support tools (DSTs) unambiguously. By expressing this trajectory-related information through well-defined rules, it will be able to be exchanged, negotiated, and processed by automatic systems. This idea motivated the development the Aircraft Intent Description Language (AIDL), introduced in (4) and further extended in (5) and (6).

AIDL is a formal language, defined over an alphabet and a grammar, which specifies a framework to model guidance modes and flight commands to drive the aircraft from a specified initial state to a final state while fulfilling certain constraints over the trajectory. The fundamental contribution of this paper is to enhance the AIDL intent generation capabilities with optimisation functionality by means of enriching the AIDL alphabet and grammar. This requires addressing the complexity of language translation between an intent-related language, AIDL, and a sample-trajectory-based language (6) resulting from the solution to an aircraft trajectory optimisation problem.

An aircraft trajectory optimisation problem can be modelled as an optimal control problem. In the optimal control framework, a controlled dynamical system described by a set of differential or differential-algebraic equations and a set of constraints is considered; the optimal control problem is to find a control input history (as well as the resulting trajectory of the system) that will minimise an objective function while meeting the specified initial and final conditions (7; 8). Applications to aircraft trajectory optimisation problems using optimal control include, for instance, (9), (10), (11; 12; 13), and (14). In these works, different objective functions are stated; examples include the minimisation of fuel, time, weighed fuel and time (also referred to as cost index), climate impact cost, etc., to solve realistic aircraft trajectory problems under some of the operational constraints that typically apply in ATM. Gardi et al. in (15) provide a thorough and comprehensive review. Foreseen benefits in fuel consumption, flight times, and emissions are claimed. In (11), it was found that the use free-routing trajectories can lead to around 11% (short-haul flights) and 6% (medium-haul flights) reduction in fuel consumption (and a proportional decrease in CO_2 emissions) compared to the current ATM paradigm. The results in (14) showed that the employment of a continuous cruise phase procedure could produce fuel savings between 1% and 2% of the total trip fuel for an A320 (and a reduction of trip times between 1% and 5% of total trip time).

Fuel and emissions' savings have been also confirmed in real-world trials. Two such examples are Project AIRE (Atlantic Interoperability initiative to Reduce Emissions) (16) and AIRE-2.[1]. Project AIRE (16) involved more than 1000 trial flights and found a reduction in CO_2 emissions of about 0.4 tonnes per flight.

Unfortunately, the realisation of foreseen savings will not be possible without automated tools related to trajectory prediction that allow continuous exchange of trajectory information, including revision, update, and synchronisation of trajectories at a macroscopic level (traffic scale level). The merging of aircraft optimisation algorithms with AIDL language preserves the needed standardisation while introducing the capacity to plan, negotiate, and update environmentally friendlier, cheaper trajectories under the future envisioned ATM.

The approach followed to solve the trajectory optimisation problem can be described in the following manner (12) (17): in first place, a multiphase optimal control problem with unknown switching times is formulated. Then, this problem is transcribed through a direct collocation method (in this paper, a Hermite-Simpson discretisation scheme) to a nonlinear optimisation problem (NLP), which can then be solved with specialised NLP solver software such as IPOPT (18). Then, in order to achieve automatic translation between the sampled trajectory and AIDL, a software library and a graphical user interface have been built capable of automating this process in a simply, user friendly manner.

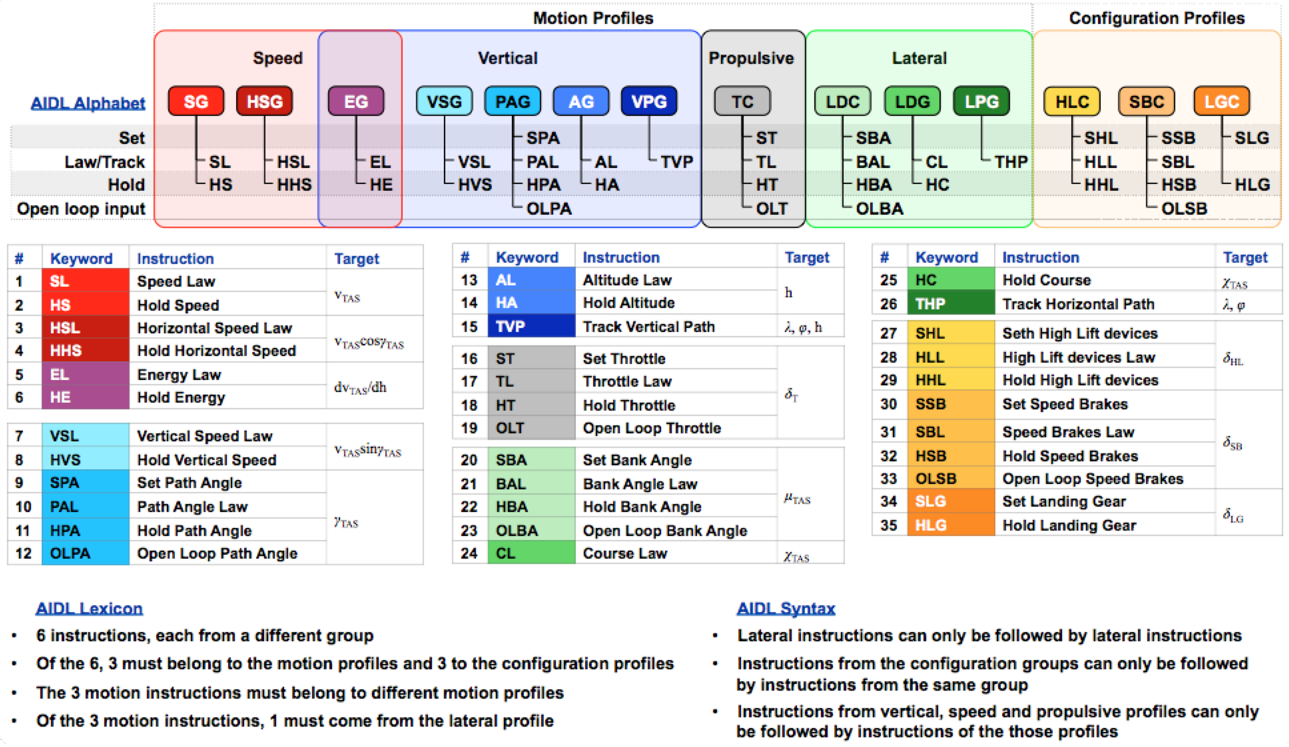
This paper is structured as follows. First, AIDL is described in Section 2 and the multiphase optimal control framework in Section 3. Then, our main contribution is introduced, the development of the AIDL optimisation extensions and the optimal intent generation infrastructure in Section 4.2. Afterwards, a case study is described and the results are analysed in order to test the present work in Section 5. Finally, conclusions are drawn and future lines of work are outlined in Section 6.

2. Aircraft Intent Description Language (AIDL)

AIDL was conceived as a formal language that enabled the description of an 'aircraft intent' (a sequence of operations determining how to control and guide an aircraft) in an univocal, rigorous, and standard fashion. As a formal language, it is composed by an alphabet and a grammar; in the case of the AIDL, we describe them in sections 2.1 and 2.2, respectively.

[1] http://www.sesarju.eu/newsroom/brochures-publications/aire-2-reports

2.1. AIDL alphabet



#	Keyword	Instruction	Target
1	SL	Speed Law	v_{TAS}
2	HS	Hold Speed	
3	HSL	Horizontal Speed Law	$v_{TAS}\cos\gamma_{TAS}$
4	HHS	Hold Horizontal Speed	
5	EL	Energy Law	dv_{TAS}/dh
6	HE	Hold Energy	
7	VSL	Vertical Speed Law	$v_{TAS}\sin\gamma_{TAS}$
8	HVS	Hold Vertical Speed	
9	SPA	Set Path Angle	γ_{TAS}
10	PAL	Path Angle Law	
11	HPA	Hold Path Angle	
12	OLPA	Open Loop Path Angle	

#	Keyword	Instruction	Target
13	AL	Altitude Law	h
14	HA	Hold Altitude	
15	TVP	Track Vertical Path	λ, φ, h
16	ST	Set Throttle	δ_T
17	TL	Throttle Law	
18	HT	Hold Throttle	
19	OLT	Open Loop Throttle	
20	SBA	Set Bank Angle	μ_{TAS}
21	BAL	Bank Angle Law	
22	HBA	Hold Bank Angle	
23	OLBA	Open Loop Bank Angle	
24	CL	Course Law	χ_{TAS}

#	Keyword	Instruction	Target
25	HC	Hold Course	χ_{TAS}
26	THP	Track Horizontal Path	λ, φ
27	SHL	Seth High Lift devices	δ_{HL}
28	HLL	High Lift devices Law	
29	HHL	Hold High Lift devices	
30	SSB	Set Speed Brakes	δ_{SB}
31	SBL	Speed Brakes Law	
32	HSB	Hold Speed Brakes	
33	OLSB	Open Loop Speed Brakes	
34	SLG	Set Landing Gear	δ_{LG}
35	HLG	Hold Landing Gear	

AIDL Lexicon

- 6 instructions, each from a different group
- Of the 6, 3 must belong to the motion profiles and 3 to the configuration profiles
- The 3 motion instructions must belong to different motion profiles
- Of the 3 motion instructions, 1 must come from the lateral profile

AIDL Syntax

- Lateral instructions can only be followed by lateral instructions
- Instructions from the configuration groups can only be followed by instructions from the same group
- Instructions from vertical, speed and propulsive profiles can only be followed by instructions of the those profiles

Fig. 2. AIDL alphabet Σ_{AIDL}

The AIDL alphabet is defined over the finite set of instructions that can be executed by certain Trajectory Engine (TE).[2] Figure 2 lists the set of instructions that conform the AIDL alphabet, herein referred to as as Σ_{AIDL}.

The Σ_{AIDL} instructions are classified into five groups according to the form in which they determine aircraft behaviour:

***Set** instructions* represent *target modes* that model the change of a given flight variable or configuration from an initial value towards a target value.

Example 2.1. The landing gear extraction is an example of a *Set* instruction, modeled by means of SLG (Set Landing Gear).

The possible motion aspects whose transition is controlled by *Set* instructions are the aircraft attitude and the power plant settings, together with all three configuration parameters, i.e., high lift devices, speed brakes, and landing gear. Their execution time is in all cases in the order of a few seconds.

***Law** instructions* describe *advanced modes* of governing an aircraft. With these instructions, it is possible to control any measurable flight variable or configuration through functions that might be implemented in the FMS or chosen by the pilot.

Example 2.2. Flying at a given Mach number is an example of a *Law* instruction, modelled by means of SL (Speed Law).

[2] TE refers to any computational infrastructure capable of integrating the resulting set of differential algebraic equations governing the motion of the aircraft.

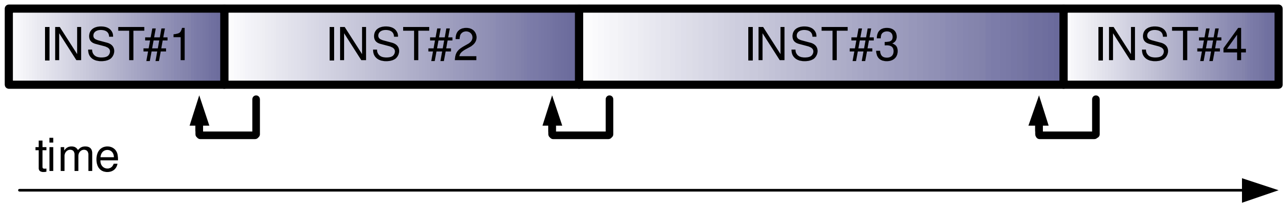


Fig. 3. Instruction thread

Either the pilot (by reading the information from the flight instruments) or the corresponding LNAV (lateral navigation) and VNAV (vertical navigation) guidance modes would ensure that the desired laws are followed by dynamically changing the controls (such as the throttle settings or the control surfaces) and configurations of the aircraft.

***Hold** instructions* describe *basic modes* of governing the aircraft. A hold mode is essentially a *Law* mode where the aircraft is controlled in order to keep the desired variable or configuration aspect constant.

Example 2.3. `Maintaining the current rate of climb is an example of a` *Hold* `instruction, modelled by means of` HVS `(Hold Vertical Speed).`

A *Hold* instruction can target amy magnitude that can be measured or computed by the aircraft sensors and systems.

***Open Loop** instructions* describe modes where the pilot directly sets (possibly depending on time) a configuration or control regardless of the state of the aircraft.

Example 2.4. `The gradual forward motion of the throttle lever by the pilot is an example of an` *Open Loop* `instruction, modelled by means of` OLT `(Open Loop Throttle).`

The number of *Open Loop* instructions in the Σ_{AIDL} alphabet is thus limited by the number of controls available to the pilot.

***Track** instructions* model *advanced modes* of governing an aircraft inside a given geometrical structure. They do not depend on time.

Example 2.5. `The lateral navigation controlled by the FMS to follow a particular path between two navaids can be modelled using a` TLP.

2.2. AIDL Grammar

The AIDL grammar is a set of rules that specify how AIDL instructions can be combined into valid AIDL instances strings. The rules allow for both sequential (when the execution intervals of each instruction are non-overlapping and continuous) and simultaneous (with overlapping intervals) combination of the instructions

It contains rules governing how to combine instructions both sequentially (instructions with contiguous, non overlapping execution intervals) and simultaneously (instructions with overlapping action intervals). They are needed in order to ensure that the AIDL intent describes in unambiguous fashion the trajectory of the aircraft.

Threads A *thread* can be defined as a time ordered sequence of instructions connected by means of linked triggers so that one and only one instruction is active at all times. Figure 3 sketches a thread composed by four instructions.

The system of equations governing the motion of an aircraft is derived from a 3DoF point-mass dynamical model of the aircraft motion, e.g., that in BADA 4 (19),[3] and needs to be integrated to obtain an aircraft trajectory. Considering

[3] for more information please visit http://www.eurocontrol.int/services/bada

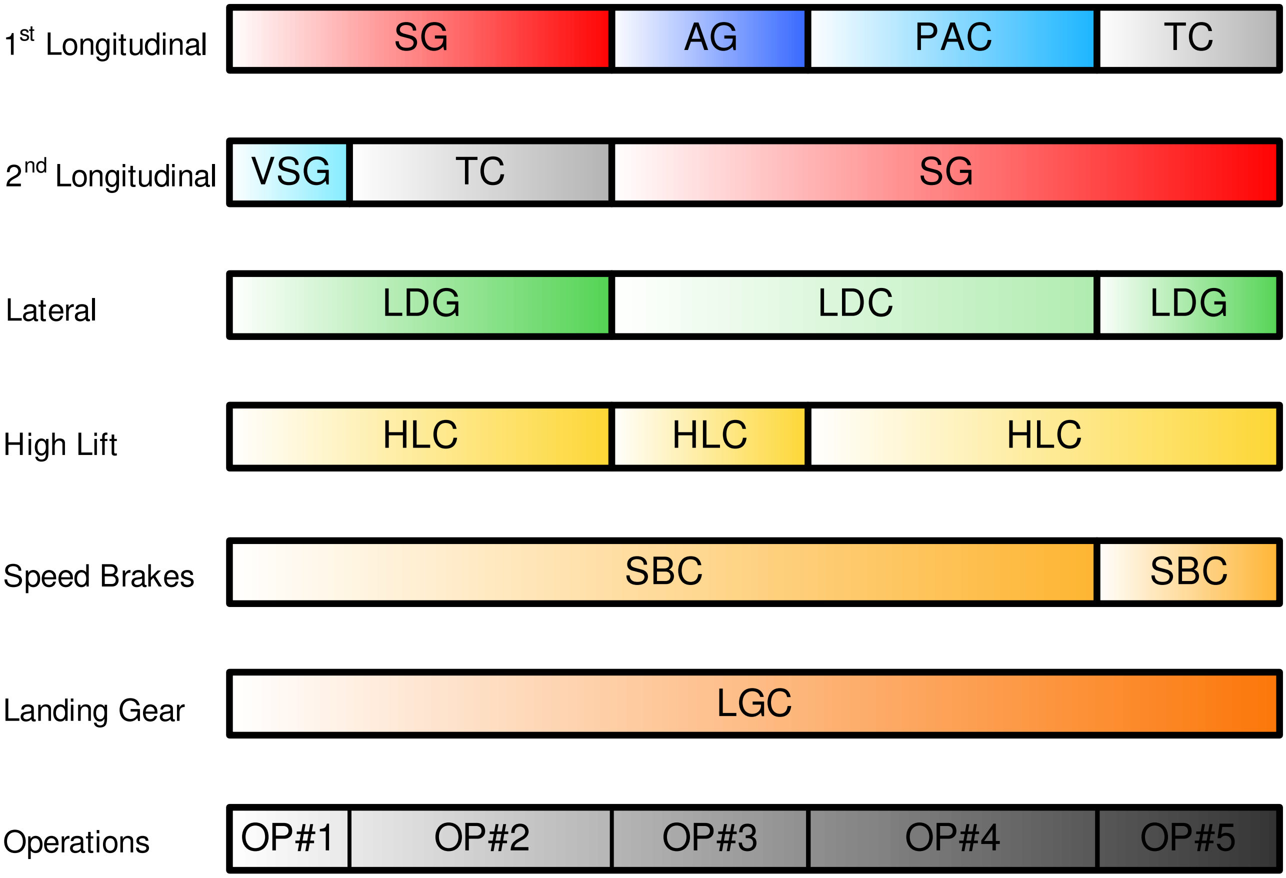


Fig. 4. Aircraft intent sequence (composed of the 6 threads) and operations resulting from it

three additional degrees of freedom associated to the aerodynamic configuration (high lifts, speed brakes, landing gear) the systems contains six degrees of freedom. By definition, each instruction closes a single degree of freedom, and thus a necessary (although not sufficient) condition for an aircraft intent sequence to be valid is that six instructions need to be active all times. This condition can be fulfilled by placing all instructions along six overlapping threads to generate an *aircraft intent sequence* as shown in Figure 4.

Triggers Instructions within a thread are connected by triggers.

AIDL allows five types of triggers (as illustrated in Figure 5):

1. **Fixed triggers** are basic conditions on distance or time. When time or distance, as desired, meets the specified value, the trigger condition is met.
2. **Floating triggers** specify the state conditions that are to be met to trigger the end of the operation. Reaching the cruising altitude is an example of floating trigger.
3. **Default triggers** are specific to a Set instruction and a particular aircraft type. The aircraft model specifies the transition law towards the desired aircraft state or configuration and thus determines when the corresponding default trigger is met. For example, a default trigger after a Set instruction for the deployment of High-lift devices will be met when the aircraft model determines that the high-lift devices have reached their new configuration.
4. **Linked triggers** are triggers that point to triggers in other threads and the trigger condition is met exactly at the same time as the condition of the designated trigger. They can therefore be used to synchronise threads.

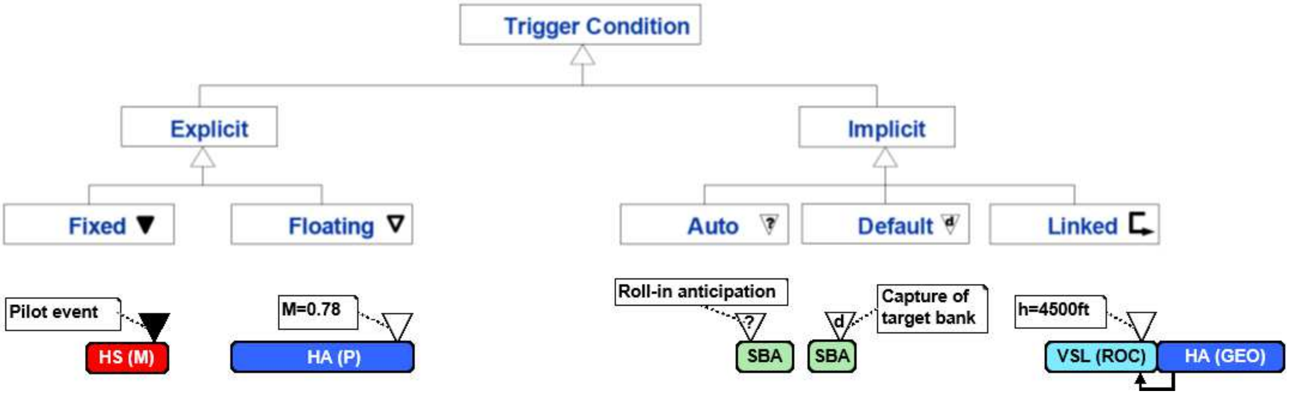


Fig. 5. AIDL triggers

5. **Auto triggers** are triggers that refer to a future condition on the aircraft state. On example is an auto trigger for the Top of Descent in order to meet a future condition.

Compatibility among Threads An aircraft intent sequence guarantees that there are six active instructions at all times, but does nothing to ensure the compatibility of these instructions among themselves. The mathematical problem composed by the equations of motion contains six degrees of freedom that must be closed by the six active instructions, each of them adding its effect constraint as an algebraic equation that closes a specific degree of freedom.

In a real aircraft, the six degrees of freedom of the aircraft motion are closed by the different controls at the disposal of the pilot or the Flight Management System (FMS). They are the following:

- Throttle lever, denoted by throttle control parameter δ_{T}.
- Elevator, denoted by longitudinal control parameter δ_{L}.
- Ailerons and rudder coordinated action.
  denoted by lateral directional control parameter δ_{LD}.
- High lift devices position, denoted by high lift configuration parameter δ_{HL}.
- Landing gear position, denoted by landing gear configuration parameter δ_{LG}.
- Speed brakes position, denoted by speed brakes control parameter δ_{SB}.

Table 1 presents the relationship between instruction profiles, control parameters, and sequence threads.

AIDL instructions capture different basic commands, guidance modes, and control inputs implemented through one of the above six parameters, and materialised in the control of a specific *target variable*. It can be derived from the above that the six instructions that are active at any time in the aircraft intent sequence are *compatible* if their effects can be simultaneously implemented by means of the six available controls listed above. If this is so, their effect constraints complement the equations of motion in a way that results in a well posed mathematical problem with a unique solution, which is the aircraft trajectory for the period of time for which the six instructions are simultaneously active (4). An example of properly placed triggers can be found in Figure 6.

Sequences By placing each instruction in the thread(s) assigned to its profile, as shown in Figure 4[4], the six instructions of the sequence active at any time will be compatible as long as the active instructions belonging to the 1st and 2nd longitudinal profiles do not belong to the same profile (either speed, vertical, or propulsive). The sequence shown in Figure 4 is in

[4] Figure 4 does not show any triggers for clarity, and replaces the instruction keyword with that of its group.

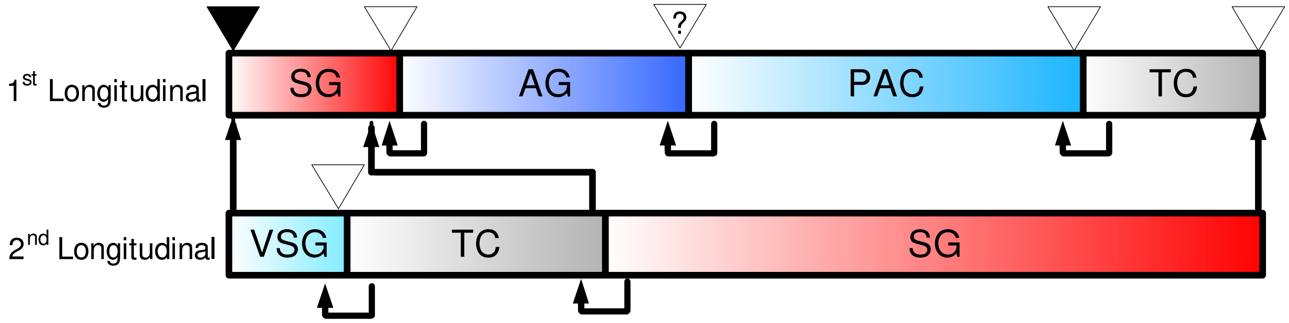


Fig. 6. Properly placed triggers

Profile	Groups	Parameter	Thread
Speed	**SG**, **HSG**, **EG**	δ_{L} or δ_{T}	Long. (1st & 2nd)
Vertical	**EG**, **VSG**, **PAC** **AG**, **VPG**	δ_{L} or δ_{T}	
Propulsive	**TC**	δ_{T}	
Lateral	**LDC**, **LDG**, **LPG**	δ_{LD}	Lateral
High Lift	**HLC**	δ_{HL}	High Lift
Speed Brakes	**SBC**	δ_{SB}	Speed Brakes
Landing Gear	**LGC**	δ_{LG}	Landing Gear

Table 1. Relationship between instruction profiles and sequence threads.

principle properly composed as the six combinations between instructions of the two longitudinal threads correspond to instructions of different profiles.

Operations An *operation* can be defined as a segment of the trajectory that corresponds to the simultaneous application of a unique set of instructions during the *operation interval* such that each operation is associated with a single instruction per thread (note that the reverse is not true, i.e., a single instruction can span several operations)

Example 2.6. Figure 4 shows the different operations that could correspond to the illustrated aircraft intent.[5] As explained above, the operations resulting from a given aircraft intent sequence cannot be determined until the trajectory integration occurs.

3. Multiphase optimal control

The motion of the aircraft can be modelled as a multiple phase process. Each phase corresponds to a differential-algebraic dynamic subsystem within set $\Sigma = \{\Sigma_0, \Sigma_1, \ldots, \Sigma_{N-1}\}$, so that $\Sigma_k = \{f_k : \mathcal{X}_k \times \mathcal{U}_k \times \mathbb{R}^{n_{l_k}} \longrightarrow \mathbb{R}^{n_{x_k}} | k \in \mathcal{K};\ g_k : \mathcal{X}_k \times \mathcal{U}_k \times \mathbb{R}^{n_{l_k}} \longrightarrow \mathbb{R}^{n_{g_k}} | k \in \mathcal{K}\}$, where f_k represents the right-hand side of the differential equation $\dot{x}(t) = f_k(x(t), u(t), p)$ for subsystem k, g_k denotes the algebraic constraints $0 = g_k(x(t), u(t), p)$ for subsystem k, with $k = 0, \ldots, N-1$ representing phase index. $\mathcal{X}_k \in \mathbb{R}^{n_{x_k}} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U}_k \in \mathbb{R}^{n_{u_k}} \subseteq \mathbb{R}^{n_u}$ are the state and control sets for subsystem k, respectively. $x(t) \in \mathbb{R}^{n_x}$ represents a n_x-dimensional piecewise state variable and $u(t) \in \mathbb{R}^{n_u}$ a n_u-dimensional piecewise control input. $p \in \mathbb{R}^{n_p}$ represents a vector of parameters.

Let

$$t_I = \tilde{t}_0 \leq \tilde{t}_1 \leq \cdots \leq \tilde{t}_N = t_F$$

[5] Triggers have been omitted for the sake of clarity.

be the switching times between phases. In other words, at time $\tilde{t}_k$ the dynamic subsystem changes from Σ_{k-1} to Σ_k. Thus, during time subinterval $[\tilde{t}_k, \tilde{t}_{k+1}]$, the evolution of the process is governed by dynamic subsystem Σ_k. Within subinterval $[\tilde{t}_{N-1}, \tilde{t}_N]$ the active dynamic subsystem is Σ_{N-1}.

Let us define the sequence of switching conditions in the set $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{N-1}\}$. $\mathcal{S} = \mathcal{S}_{\mathcal{A}} \cup \mathcal{S}_C$ consists of logic constraints that relate continuous state and switching mode. $\mathcal{S}_{\mathcal{A}}$ includes the set of autonomous (also referred to as internally forced) switchings, and $\mathcal{S}_C$ corresponds to the set of controlled (also referred to as externally forced) switchings. As illustration: an autonomous switch occurs when the state trajectory intersects a certain set of the state space; then the system is forced to switch from subsystem 1 to subsystem 2. In the case of controlled switches, only when the state belongs to a certain set, a transition from subsystem 1 to subsystem 2 is possible. This controlled switch might take place in response to the control law.

3.1. Problem statement

Consider a sequence of active dynamical subsystems $\Sigma = \{\Sigma_0, \Sigma_1, \ldots, \Sigma_{N-1}\}$ (and their corresponding sets of constraints) together with a sequence of switching conditions $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{N-1}\}$ at switching times $t_I = \tilde{t}_0 \le \tilde{t}_1 \le \cdots \le \tilde{t}_N = t_F$.

The multiphase optimal control problem can be formulated as follows:

Problem 3.1 (***Multiphase Optimal Control Problem***)**.**

$$\begin{aligned}
&\mathit{min}\ J(t, x(t), u(t), p) = \sum_{k=0}^{N-1} \Bigg[\Phi_k(\tilde{t}_{k+1}, x(\tilde{t}_{k+1})) \\
&+ \int_{\tilde{t}_k}^{\tilde{t}_{k+1}} L_k(x(t), u(t), t, p) dt \Bigg]. && \text{(1a)} \\
&\textit{Subject to:} \\
&\dot{x}(t) = f_k(x(t), u(t), t, p), k = 0, \ldots, N-1, && \text{(1b)} \\
&0 = g_k(x(t), u(t), t, p), k = 0, \ldots, N-1, && \text{(1c)} \\
&x(t_I) = x_I, && \text{(1d)} \\
&\psi(x(t_F)) = 0, && \text{(1e)} \\
&\phi_{l_k} \le \phi_k(x(t), u(t), t, p) \le \phi_{u_k}, k = 0, \ldots, N-1, && \text{(1f)} \\
&0 = \vartheta_k^{eq}(x(\tilde{t}_{k+1}), \tilde{t}_{k+1}, p), k = 0, \ldots, N-2, && \text{(1g)} \\
&0 \le \vartheta_k^{ieq}(x(\tilde{t}_{k+1}), \tilde{t}_{k+1}, p), k = 0, \ldots, N-2. && \text{(1h)}
\end{aligned}$$

The objective function $J : [t_I, t_F] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}$ is given in multiphase Bolza form. It is expressed as the summatory on index $k = 0, \ldots, N$ of the sum of a multiphase Mayer term $\Phi_k(\tilde{t}_{k+1}, x(\tilde{t}_{k+1}))$ and a multiphase Lagrange term $\int_{\tilde{t}_k}^{\tilde{t}_{k+1}} L_k(x(t), u(t), t, p)dt$. Functions $\Phi_k : [\tilde{t}_k, \tilde{t}_{k+1}] \times \mathbb{R}^{n_{x_k}} \to \mathbb{R}$ and $L_k : [\tilde{t}_k, \tilde{t}_{k+1}] \times \mathbb{R}^{n_{x_k}} \times \mathbb{R}^{n_{u_k}} \times \mathbb{R}^{n_{p_k}} \to \mathbb{R}$ are assumed to be twice differentiable. Dimensions $n_{x_k}, n_{u_k}, n_{z_k}, n_{p_k}$ can be different for each phase. f_k is assumed to be piecewise Lipschitz continuous within the time interval $[\tilde{t}_k, \tilde{t}_{k+1}]$, and the derivative of the algebraic right hand side function g_k with respect to x, i.e., $\frac{\partial g_k}{\partial x} \in \mathbb{R}^{n_{x_k} \times n_{x_k}}$, is assumed to be regular within the time interval $[\tilde{t}_k, \tilde{t}_{k+1}]$. $x_I \in \mathbb{R}^{n_x}$ is the vector of initial conditions and the function $\psi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_q}$, assumed to be twice differentiable, contains the final conditions. The system must satisfy multiphase algebraic path constraints within the time interval $[\tilde{t}_k, \tilde{t}_{k+1}]$ given by the functions $\phi_k : [\tilde{t}_k, \tilde{t}_{k+1}] \times \mathbb{R}^{n_{x_k}} \times \mathbb{R}^{n_{u_k}} \times \mathbb{R}^{n_{p_k}} \to \mathbb{R}^{n_{\phi_k}}$ with lower bound $\phi_{l_k} \in \mathbb{R}^{n_{\phi_k}}$ and upper bound $\phi_{u_k} \in \mathbb{R}^{n_{\phi_k}}$. Function ϕ_k is assumed to be twice differentiable. Dimension n_{ϕ_k} can be different for each phase. The system must also satisfy equality and inequality interior point constraints given by the functions $\vartheta_k^{eq} \in \mathbb{R}^{n_{eq_k}}$ and $\vartheta_k^{ieq} \in \mathbb{R}^{n_{ieq_k}}$, which are assumed to be twice differentiable. Dimensions n_{eq_k} and n_{ieq_k} can be different for each phase.

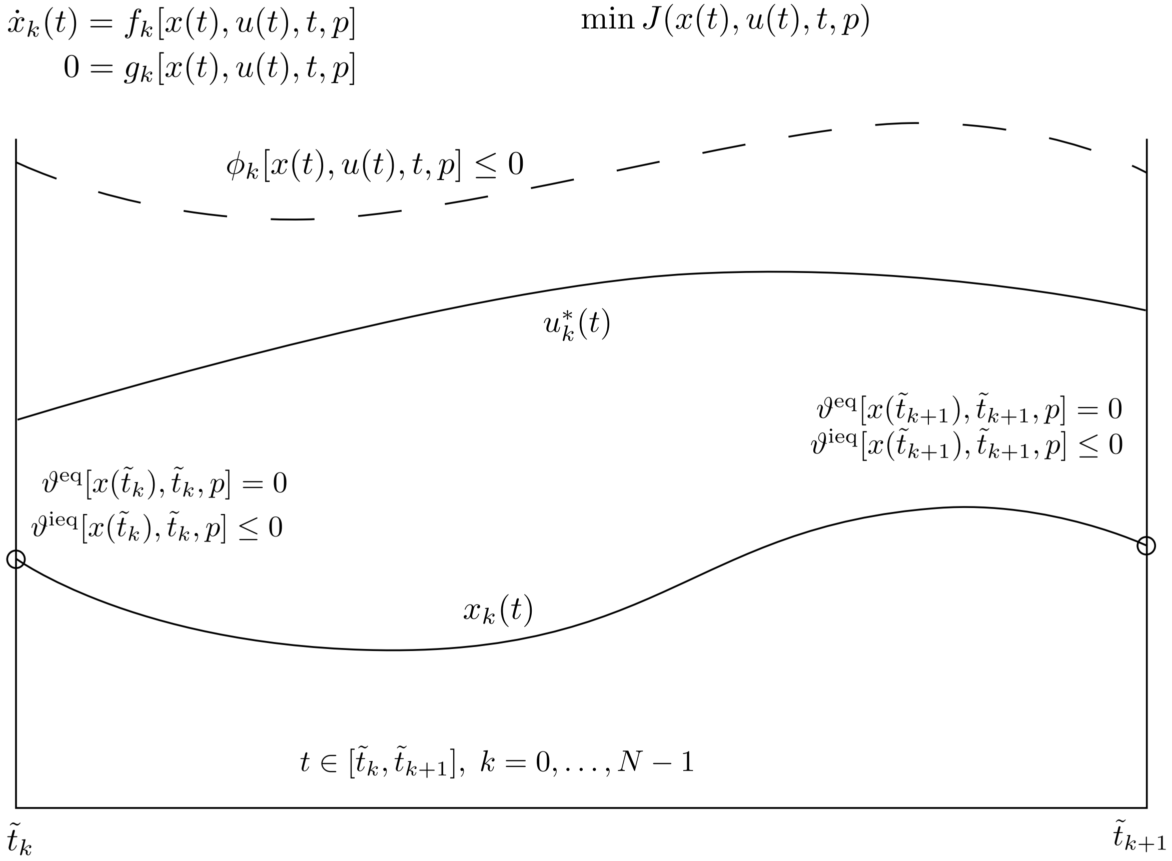

$\dot{x}_k(t) = f_k[x(t), u(t), t, p]$
$0 = g_k[x(t), u(t), t, p]$
$\min J(x(t), u(t), t, p)$
$\phi_k[x(t), u(t), t, p] \leq 0$
$u_k^*(t)$
$\vartheta^{\text{eq}}[x(\tilde{t}_{k+1}), \tilde{t}_{k+1}, p] = 0$
$\vartheta^{\text{ieq}}[x(\tilde{t}_{k+1}), \tilde{t}_{k+1}, p] \leq 0$
$\vartheta^{\text{eq}}[x(\tilde{t}_k), \tilde{t}_k, p] = 0$
$\vartheta^{\text{ieq}}[x(\tilde{t}_k), \tilde{t}_k, p] \leq 0$
$x_k(t)$
$t \in [\tilde{t}_k, \tilde{t}_{k+1}],\ k = 0, \ldots, N-1$
$\tilde{t}_k$
$\tilde{t}_{k+1}$

Fig. 7. Multiphase optimal control problem.

Figure 7 illustrates the statement of the problem schematically. As an illustration, in the objective function (1a), the Lagrange term describes an instantaneous cost that is accumulated over time, while the Mayer term represents an additional cost that depends on the terminal state of the system. One example of a Lagrange term is the total amount of energy spent in a system; one example of a Mayer term is the final time, to be minimised. (1b) and (1c) represent the differential-algebraic equation system that describes the trajectory of the system; (1f) models path constraints that restrict the allowed trajectories and might be related to physical or regulatory limits that must not be breached; (1d) and (1e) are the boundary conditions of the problem; (1g) and (1h) model constraints applied on the phase transitions, e.g., to reach a waypoint (equality interior point constraint on the position of the aircraft) within a given time window (inequality interior point constraint on the time at which the position is reached).

The multiphase optimal control problem is then finding an admissible control $u^*(t)$ and the switching instants, $(\tilde{t}_1^*, \ldots, \tilde{t}_{N-1}^*)$, such that the set of differential-algebraic subsystems follows an admissible trajectory $(x^*(t), u^*(t), p^*)$ between the initial and final state that minimises the performance index $J(x(t), u(t), t, p)$. The final time, $\tilde{t}_F = \tilde{t}_N$, may be fixed or left free, depending on the problem.

3.2. Solution approach

Problem reformulation The multiphase optimal control problem is tackled first by making the unknown switching times part of the state. Then a new independent variable is introduced with respect to which the switching times are fixed. In this reformulated problem, a linear relation between the new variable and time exists. The linear term (the slope in other word) of this linear relation changes on each interval between switches. These slopes represent indeed time scaling factors that determine the optimal switching times. The equivalent problem can be considered a conventional optimal control problem. Interested readers are referred to (17) for mathematical details.

Solving continuous time optimal control problems Practical optimal control problem can be complex and non-linear, and finding analytical solutions is usually not a feasible approach even when considering simple problems. Therefore, numerical methods are employed instead in order to obtain a practical solutions. Numerical methods in optimal control can be classified in three broad families: dynamic programming (DP) methods (which try to approximate a solution of the Hamilton-Jacobi-Bellman PDE (20) that describes the sufficient optimality conditions of the problem), indirect methods (which seek to solve the two-point boundary value problem arising from the Pontryagin's maximum principle (21) that describes the necessary optimality conditions of the problem), and direct methods (that approximate the trajectory as a finite-dimensional parametrisation and then apply nonlinear optimisation algorithms to this discretisation (8))

The direct application of DP is not practical in the case of continuous systems with a moderate dimension of the state space because of the *curse of dimensionality* [6]. Regarding indirect methods, many drawbacks arise (8, chapter 4.3). Some major drawbacks of such methods are: 1) numerical difficulties, 2) the a priori lack of knowledge about the sequence of arcs where inequality constraints are binding or not, and 3) the difficulty of generating an adequate initial guess for the iterative resolution algorithms, since there is no intuitive physical interpretation of the adjoint variables

These practical difficulties imply that indirect methods are not functional for the kind of general optimal control problems with unknown user-described complex structures that should be faced. Instead, direct methods have been selected; while they may be less accurate than indirect methods (22), they are more flexible, easier to implement, and robust (depending on the problem). These advantages explain why they have been widely used for the solution of optimal control problems within the field of aerospace engineering. Comprehensive sources for direct and indirect methods for trajectory optimisation can be found in (23; 24; 25).

Direct methods operate by discretising the infinite-dimensional trajectory into a finite set of variables. Then, a Non-Linear Programming (NLP) problem is formulated with these discretised variables and solved using nonlinear optimisation algorithms. The process is often summarised as “first discretise, then optimise”. They can be classified into direct shooting (single (26) and multiple (27)) and direct collocation (28). Direct collocation methods, in particular, are currently popular in aerospace optimal problems (23; 25; 29)

Direct collocation methods Collocation methods describe the state trajectory as an interpolant on a set of points and enforce the dynamic equations at a (potentially different) set of collocation nodes (28; 29; 30). At each collocation node, an NLP constraint (called defect constraint) is created equating the state derivatives (related to the state values by the interpolation rule) to the dynamical function evaluated on that point, ensuring that the equations of motion are approximately satisfied.

Several collocation schemes have been proposed in the literature. One of the simplest ones is Hermite-Simpson discretisation (31; 32), which subdivides the time interval into a number of subintervals and a third-order Hermite interpolating polynomial is used as interpolant within the time subintervals and the equations of motion are collocated at the midpoint of each subinterval (which is analogous to the Simpson integration rule). Higher-order variants can also be obtained with higher-order quadrature rules; they are called Hermite-Legendre-Gauss-Lobatto (HLGL) methods (30). In this work, a Hermite-Simpson discretisation is employed.

Pseudospectral methods are a more recent development, relying on a global Lagrange interpolant on a set of Legendre-Gauss quadrature points. These methods can offer spectral accuracy for smooth problems, but their direct application to highly constrained complex problems does not offer such advantage unless employed within an adaptive scheme (33). See (34; 29; 25) for surveys that discuss pseudospectral methods in detail.

[6] While some techniques exist in order to ameliorate the curse of dimensionality, usually grouped under the banner of *Approximate Dynamic Programming*, they raise their own challenges as well.

The resulting NLPproblem can be solved with general-purpose nonlinear optimisation algorithms (35), usually through modelling interfaces such as AMPL.[7] The open source NLPsolver IPOPT (Interior Point Optimizer) (18) is used, which implements an interior point algorithm that is able to handle large sparse problems. It is hosted at COIN-OR (www.coin-or.org).

4. AIDL-Optimal Control Integration

4.1. AIDL optimisation Extensions

The AIDL language is extended in order to include optimal control capabilities by providing Optimal Guidance Laws (OGL) and Optimal Trigger Conditions (OTC).

Proposed OGL include new instructions such as:

- OSL - Optimal Speed Law.
- OHSL - Optimal Horizontal Speed Law.
- OEL - Optimal Energy Law.
- OVSL - Optimal Vertical Speed Law.
- OPAL - Optimal Path Angle Law.
- OAL - Optimal Altitude Law.
- OTL - Optimal Throttle Law.
- OBAL - Optimal Bank Angle Law.
- OCL - Optimal Course Law.

Proposed OTC include:

- Optimal Floating.

Both new triggers and instructions are compliant with the AIDL formal language as described in Section 2.

***Optimal Guidance Law** instructions* model *advanced modes* of guiding and controlling an aircraft in a manner such that a given optimisation criteria, e.g., minimum fuel, is fulfilled.

Example 4.1. Cruising at the optimal cruising speed, expressed in terms of Mach, CAS (calibrated airspeed), or TAS (true airspeed), is an example of an *Optimal Law* instruction, modelled by means of OSL (Optimal Speed Law). Notice that this OSL results from the solution to an optimal control problem.

***Optimal Floating** triggers* specify the state conditions that trigger the transition between one instruction and the following within one thread. They are left as variables to be determined by the solution to the optimal control problem.

Example 4.2. A classical speed profile during climb phase would be constituted by a CAS operation followed by a Mach operation, in other words, a HS (CAS) instruction followed by a HS (Mach) instruction. Using AIDL, one must specify a transition Mach number by means of a floating trigger, e.g., M=0.78, to trigger the instruction transition. Using an Optimal floating trigger, the transition Mach (and thus the Mach at which the operation would be performed) results to the solution for the optimal control problem.

[7] http://ampl.com/

4.2. Library Architecture and User Interface

A conventional AIDL workflow is illustrated in Figure 8. Starting from a Flight Intent, which represents a description of the basic components of the flight, an AIDL aircraft intent is built by the Intent Generation Engine through the use of heuristics and operational context information. The AIDL intent can then be processed by the Trajectory Prediction Engine (TPE) that produces a computed trajectory.

Figure 9 describes the integration of the optimisation toolkit within the existing AIDL pipeline. Instead of using the IGE, the AIDL intent is now produced by the Trajectory optimisation Engine, which finds the optimal trajectory with respect to an optimisation criterion that is defined by the user (usually, a cost-index-based combination of flight time and fuel burnt). This trajectory is synthesised to an optimised AIDL intent that can be employed in the same fashion as a conventional AIDL intent.

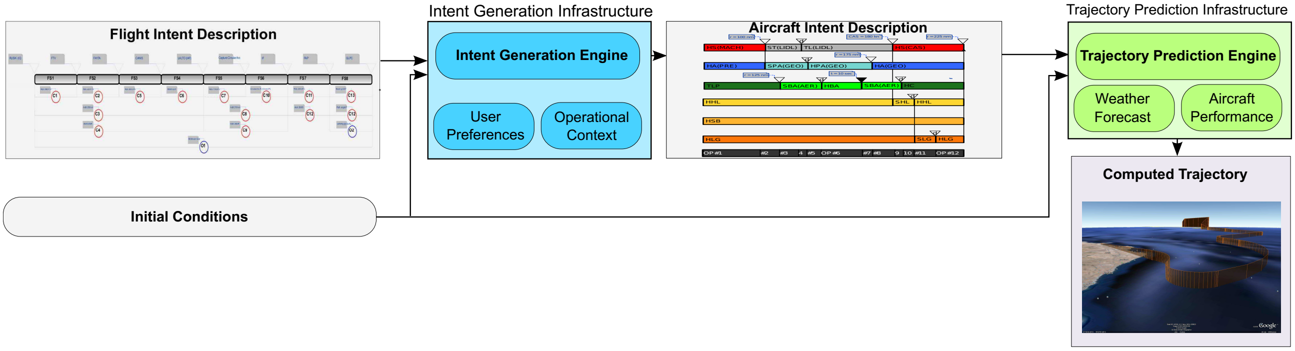

Flight Intent Description
Initial Conditions
Intent Generation Infrastructure
Intent Generation Engine
User Preferences
Operational Context
Aircraft Intent Description
Trajectory Prediction Infrastructure
Trajectory Prediction Engine
Weather Forecast
Aircraft Performance
Computed Trajectory

Fig. 8. AIDL Based Aircraft Intent.

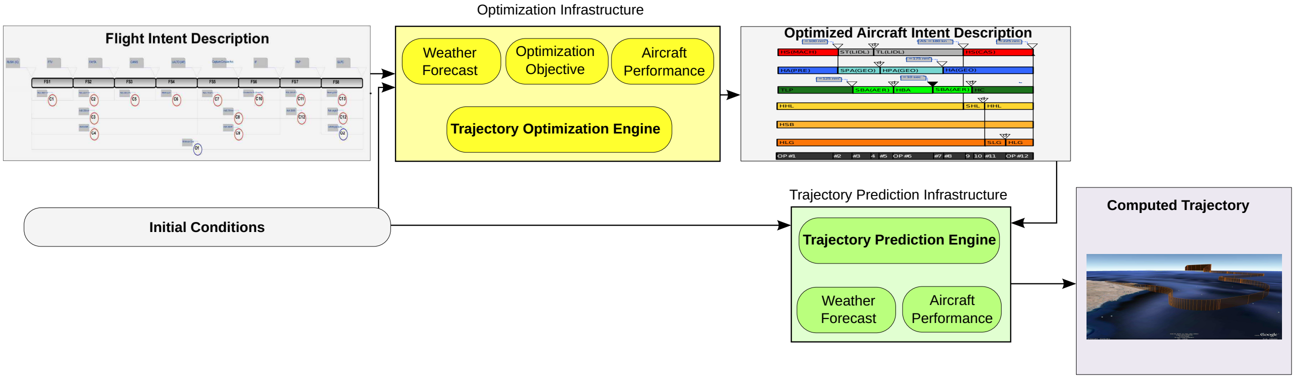

Flight Intent Description
Initial Conditions
Optimization Infrastructure
Weather Forecast
Optimization Objective
Aircraft Performance
Trajectory Optimization Engine
Optimized Aircraft Intent Description
Trajectory Prediction Infrastructure
Trajectory Prediction Engine
Weather Forecast
Aircraft Performance
Computed Trajectory

Fig. 9. Optimal Control AIDL based Aircraft Intent. Notice that the yellow box (optimisation Infrastructure) corresponds to the system depicted in Figure 10.

The optimisation infrastructure, written in the Python programming language, is composed by three main components:

- The Transcription Engine takes an optimal control problem formulation and generates the corresponding nonlinear optimisation problem (NLP) using a direct method.
- The Intent Synthesiser creates AIDL aircraft intents from the results of the optimisation process.
- A Graphical User Interface (GUI) that allows the user to manually control the optimisation process. It has been built with the Qt4 toolkit through the PySide libraries. Note that the entire functionality accessed through the GUI is also available through entirely automatic tools.

The trajectory optimisation process can be decomposed into the following steps.

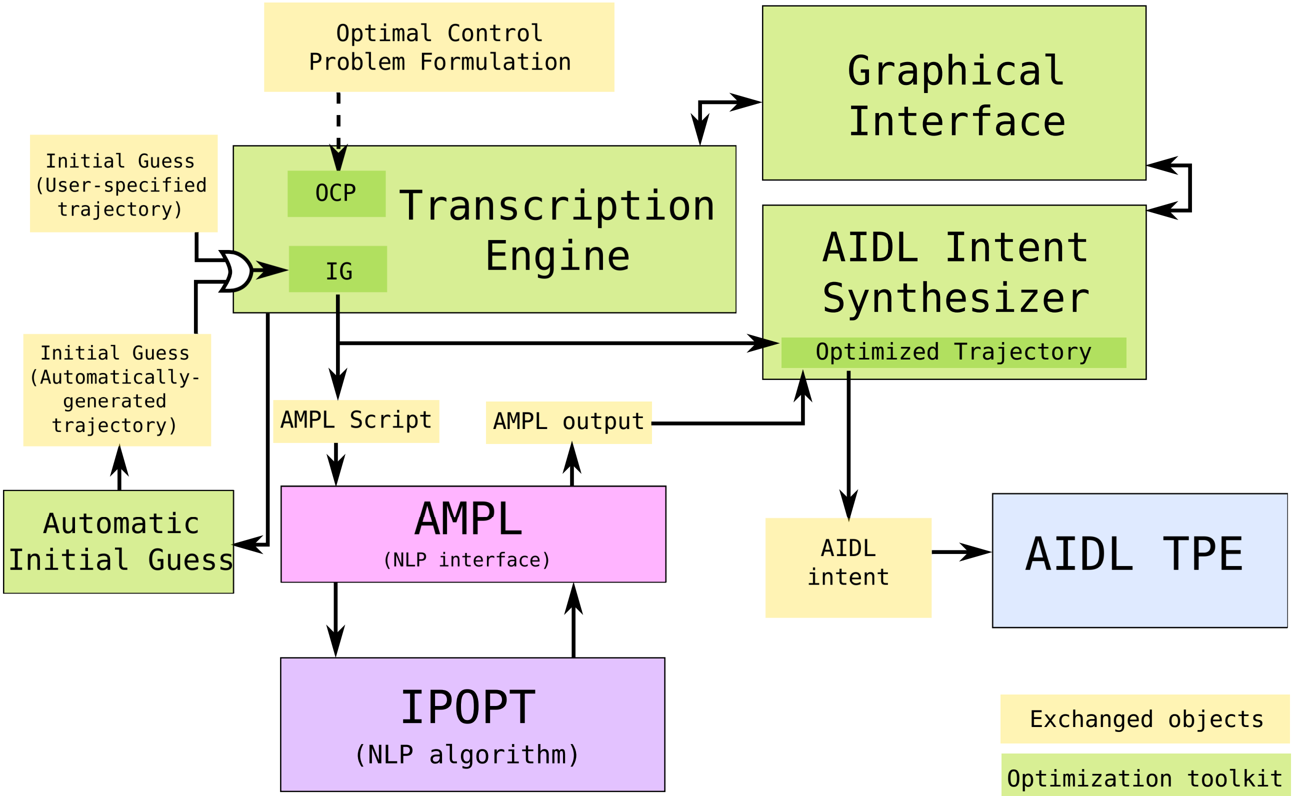


Fig. 10. Architecture of the Trajectory optimisation Engine.

- Formulation of the Optimal Control Problem (OCP). This can be achieved either by using the objects provided by the `OptGen` library directly, by using the GUI, by editing the plain-text files that serve as an input format or through the usage of an automatic tool to create them from Flight Intents.
- Generation of the initial guess: a trajectory that fulfils the boundary conditions of the problems (with the same aircraft, weather and earth model) but is not necessarily optimal. This initial guess will be used to initialise the NLPproblem. This initial guess can be obtained from a trajectory computed from an AIDL intent (possibly a non-optimised AIDL intent) or can be generated automatically by the procedure described in Section 4.2
- Transcription of the OCP into a discrete NLP. An AMPL based model is created, containing all the information from the NLPand the initial guess.
- Solution of the discrete problem with an NLPalgorithm using the interior-point algorithm IPOPT (18). In this stage, AMPL is called on the generated model, which in turn calls IPOPT to solve the model.
- Synthesis of an aircraft intent from AMPL output containing the results of the NLP.

The Transcription Engine The Transcription Engine takes the following inputs, which define the optimal control problem:

- The initial conditions.
- The aircraft performance model: BADA 3 or BADA 4.
- The list of flight phases, which correspond to phases in the optimal control problem.
- The cost functional to be optimised.
- The Earth model, e.g., flat, spherical, or WSG84.
- The atmosphere model, based on ISA.

- The meteorological forecast.

Each flight phase is characterised by the following objects:

- Transcription scheme parameters, including type of discretisation method and number of nodes per phase.
- Aerodynamic configuration of the aircraft during the phase.
- A list of extra constraints representing operational constraints (since flight envelope constraints are contained in the aircraft performance model).
- Aircraft performance model.
- List of final conditions of the phase. Naturally, the final conditions of the last phase are the final conditions of the problem.

Additionally, the library contains the following interfaces:

- Routines that load/save the OCP from/to plain-text files.
- Routines that generate the AMPL script.
- Routines that either load the initial guess from an already-generated trajectory or call the initial guess generation routines.

Automated Initial Guess An initial guess is required to feed any NLPsolver. Although NLPsolvers are designed to converge with poor initial guesses, good initial guesses have a favorable impact on the performance of the solver (25; 8). An advanced user should be able to provide the solver with a suitable initial guess. Nonetheless, automatic generation of adequate initial guesses improves the robustness of the system and eases the job of the user. This is the rationale to develop, in the context of this work, an algorithm to generate automatically an initial guess in an arbitrary scenario.

The main criteria to define the initial guess has been the fulfilment of the kinematical and dynamical constraints. Again, although this condition is not required by the solver, it helps to improve the convergence to the optimal solution. Therefore, first, the required information of the kinetic variables should be completed and, second, the control laws should be chosen to determine the dynamical degrees of freedom.

Concerning the former, the data provided by the user about the trajectory in the vertical and horizontal profile depends on the type of problem. In some cases the trajectory is almost completely defined by the user's constraints whereas in other cases, the trajectory is almost free and only origin and destination are set. The algorithm fulfils the constraints with the following principle: constant heading between waypoints and vertical profile with an ascent, a cruise and a descent phase. When altitude constraints are present, the algorithm considers a series of ascents/descents to satisfy them.

Concerning the dynamical description of the initial guess, ascent and descent are performed with constant flight path angle and the velocity of maximum aerodynamic efficiency, whereas cruise phases are flown at constant altitude and velocity of maximum aerodynamic efficiency.

The automatic initial guess generation has proven to be reliable, and robust, thus enhancing the usability of the tool and allowing the user to focus on the optimisation problem.

The Intent Synthesiser The Intent Synthesiser contains the classes and routines that read the AMPL output and then build an AIDL intent from this output. The optimised intent is generated according to the following procedure. First, the ground distance covered by each phase of the solution trajectory of the optimal control problem is computed. An intent structure is generated, with each thread being composed by a sequence of instructions separated by fixed trigger conditions related to distance. Then, each phase of the optimal control formulation will be mapped to a single AIDL operation by adding the corresponding instruction in each of the AIDL threads, with linked triggers that ensure that the instructions corresponding to the same phase end at the same time, thus generating a unique operation. Figure 11 illustrates an example of the output transcription.

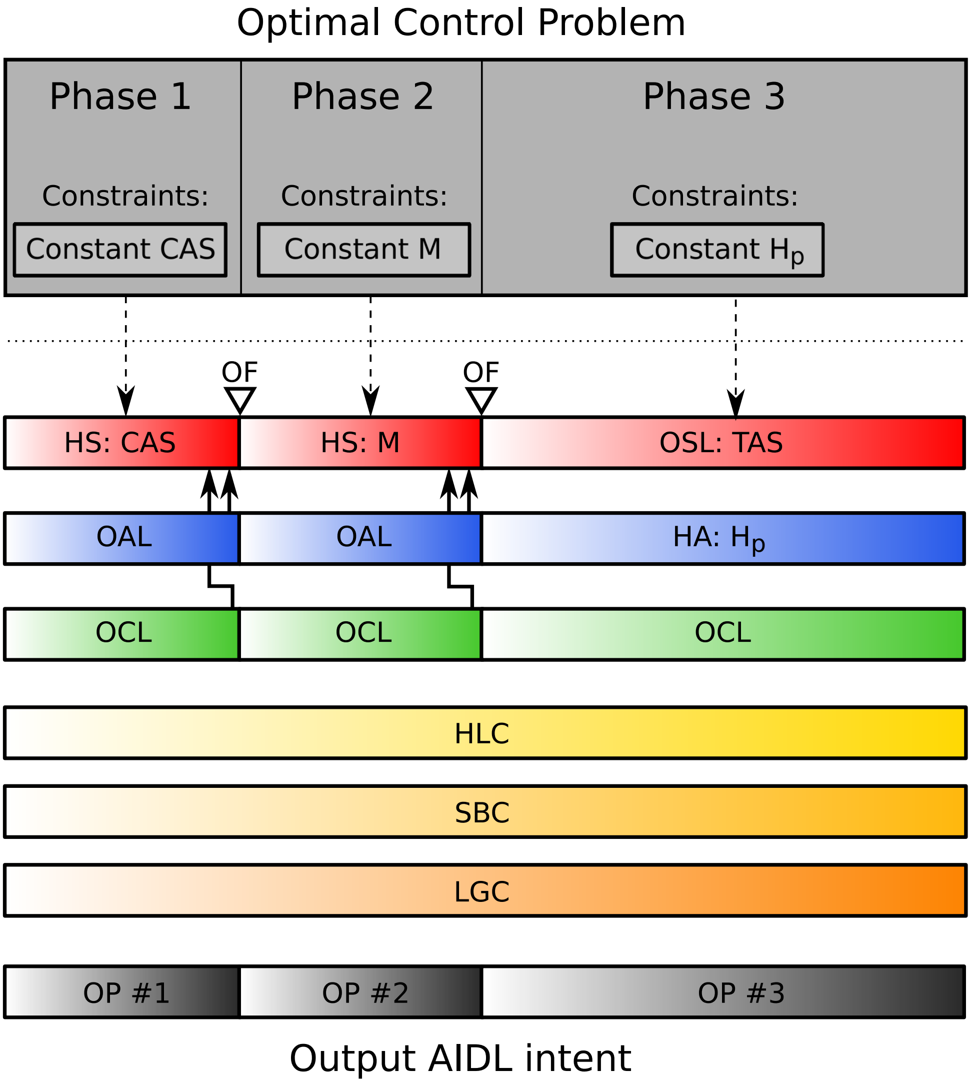


Fig. 11. Example transcription of an Optimal Control problem to AIDL output

Each instruction relates to the a trajectory phase, and each thread is composed by the same structure (same triggers). Then, for each phase:

- Any custom restrictions applied to the phase of the optimal control problem (such as constant speed or altitude) are transcribed to the corresponding Hold Speed or Hold Altitude instruction.
- If there is no speed restriction, add a Speed Law specifying TAS as a function of the distance, where the function is a cubic spline interpolant of the TAS of the solution trajectory in the discretisation nodes of the direct transcription.
- If there is no altitude restriction, add an Altitude Law specifying the geopotential pressure altitude as a function of the distance, where the function is a cubic spline interpolant of the altitude of the solution trajectory in the discretisation nodes of the direct transcription.
- Add a Course Law instruction specifying the geographic heading as a function of the distance, where the function is a cubic spline interpolant of the geographic heading of the solution trajectory in the discretisation nodes of the direct transcription.
- Add the Set High Lift devices, Set Speed Brakes and Set Landing Gear instructions corresponding to the aerodynamic configuration of the phase to the last three AIDL threads.

5. Case study

A case study is proposed in order to test the implemented functionality. In the main analysis, three aircraft intents for the same flight will be compared:

- A non-optimised aircraft intent designed with a heuristic that applies standard flight rules and implemented as a typical AIDL instance. This will be herein termed **AIDL reference intent**.
- An optimised aircraft intent that flies through the same geographical route (in other words, same lateral profile) as the reference intent. This will be herein termed **AIDL optimised intent**.
- An "ideal" aircraft intent where both vertical and lateral profiles are set free. This will be herein termed **AIDL free-routing intent**.

The three different aircraft Intents are presented in Figure 12, respectively. As it can be noticed, Figure 12.a (that represents the AIDL reference intent) includes hold speed instructions and floating triggers that univocally define ascent, cruise, and descent profiles. For instance, by combined action of hold instructions and floating triggers, the CAS ascent speed is set to 280 kt, the transition Mach to 0.78, and the cruising altitude to 36700 ft. Figure 12.b (that represents the AIDL optimised intent) includes some of the optimisation capabilities previously explained, namely: optimal altitude laws for ascent/descent; an optimal speed law for the cruise; optimal floating triggers. Both Fig. 12.a and Fig. 12.b follow the same lateral path, as the third thread illustrates.[8] Fig. 12.c presents a free-routing intent, in which a single operation suffices: the intent will be univocally defined by: OSL, OAL, and an OTL, resulting in a so-called continuous operation (orthodromic route together with optimal altitude/speed profile). This illustrates the potentiality of the framework towards the design of more efficient flights.

The analysed case is based on a flight departing from LFPG and arriving to EPWA through STAR BIMPA4U[9]. The objective function to be minimised is the fuel consumption. The aircraft is a Boeing B737-800. Additional assumptions include: International Standard Atmosphere (ISA) model and spherical Earth. Moreover, the aircraft is assumed to fly always with clean configuration (no flaps deflected, speed breaks off, and landing gear up), as the takeoff and landing of the aircraft is not modelled. Boundary conditions are presented in Table 2.

Two different experimental set-ups will be used:

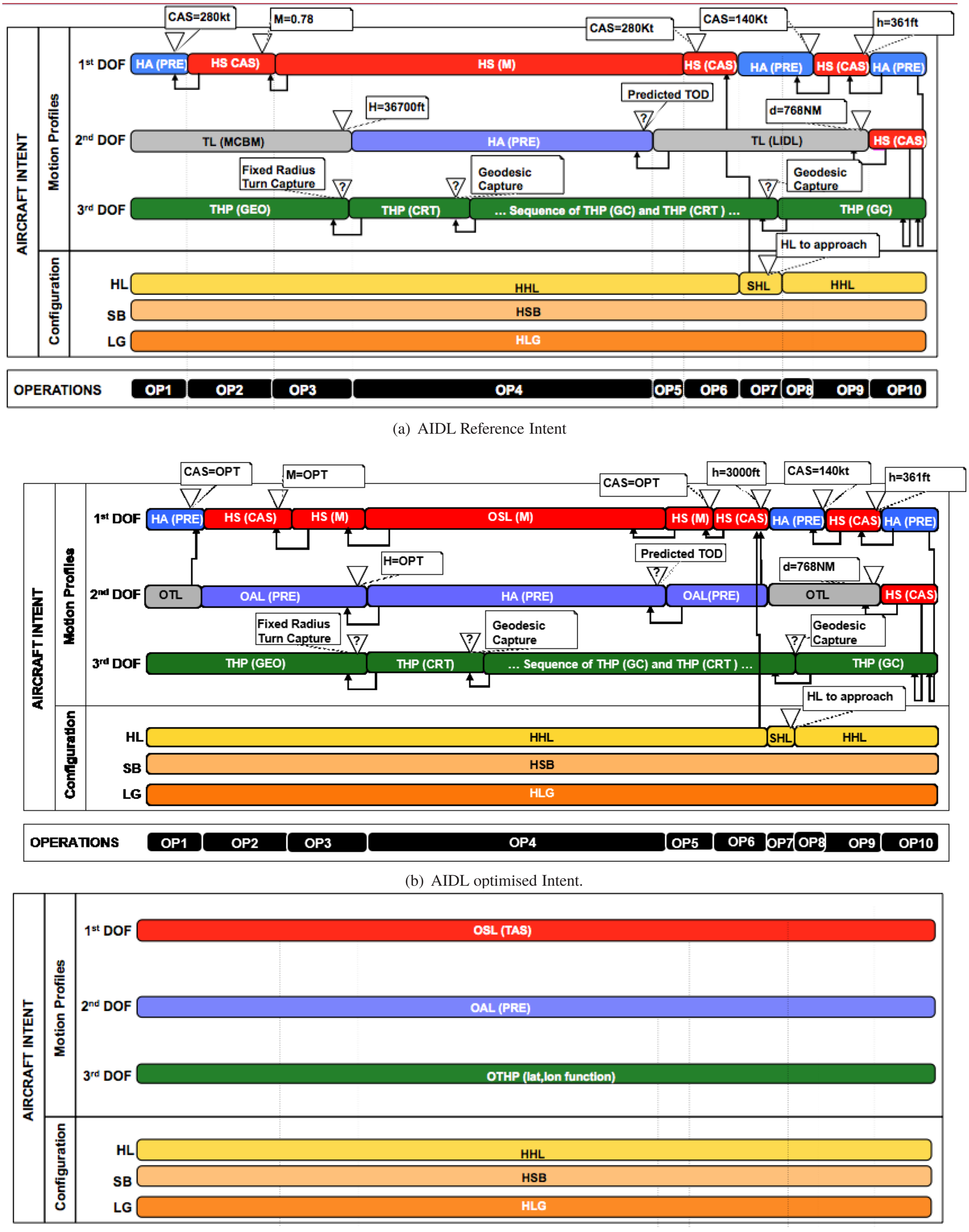
- The main analysis will be made considering a BADA 3.11 (36) aircraft dynamical model, calm conditions, and the objective of minimizing fuel burnt.
- A secondary analysis is presented to show the robustness of the approach and its suitability to incorporate advanced modeling features. The AIDL optimised intent is run with the BADA4 performance model, under forecasted wind conditions and for different values of the cost index (CI); that is, a cost functional of the form:

$$J = -m_f + \mathrm{CI} \cdot t_f$$

  will be minimised. As CI increases and the user places more weight on shorter flight times than lower fuel burns, the optimiser will find solutions with higher airspeeds (and thus higher drag, thrust and fuel burn) which arrive to the destination faster. wind fields are based on the Global Forecast System (GFS) projections, produced by the National Centers for Environmental Prediction (NCEP). A forecast produced the 21st of January, 2016 at 00:00 UTC is used.

[8] It should be noted that the number of instructions within the third thread (all of them derived from waypoints' overfly) have been intentionally reduced for the sake of clarity in the figure. The instruction *Sequence of THP (GC) and THP (CRT) reflects a series of arcs connecting all waypoints of the flight.*

[9] Notice that only clean configuration phases are considered



(a) AIDL Reference Intent

(b) AIDL optimised Intent.

(c) AIDL Free-Routing Intent

Fig. 12. Pansa aircraft AIDL intents.

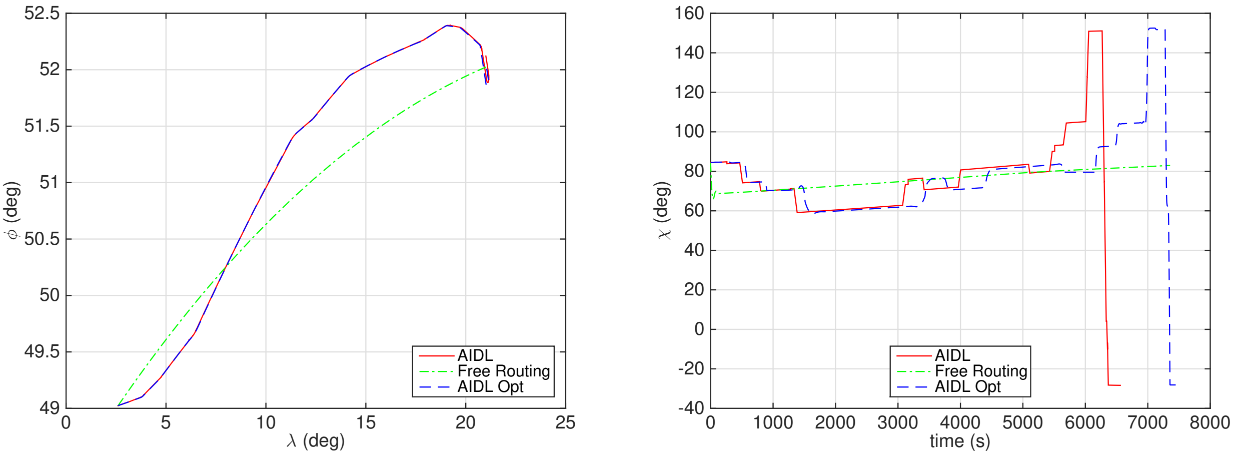
Table 2. Boundary conditions

Bound.	Time [s]	h [m]	V_{TAS} [m/s]	γ [rad]	m [kg]	λ [rad]	θ [rad]	χ
Ini.	0	533.4	90	0	49393.5	0.85562626	0.045305107	1.47307525
Fin.	-	533.4	-	-	-	0.36792995	0.908186924	-0.491238

5.1. Main analysis results

The two optimised trajectories are computed using the described methodology in Section 3.2. The resulting large-scale NLP problem has been solved using IPOPT. A discretisation grid with a total of $N = 220$ (20 per phase) sample points has been used. The resulting NLP problem had 3825 variables, 3032 equality constraints, and 3388 inequality constraints. The total computational time on a Mac OS X 2.56 GHz laptop was 456 s. This computational time is compatible not only with strategic, but also with tactical trajectory planning.

Flight optimal path, optimal control law, and the evolution of the state variables over time are also obtained as part of the solution. State variables and control inputs are shown in Figure 13-16, including speed and altitude profiles, heading angle and flight path angle, the lateral path (latitude-longitude), the mass of the aircraft, and the control inputs (coefficient of lift, thrust, and back angle in the context of the optimal control problem). Table 3 shows climb, cruise, and descent speed profiles' values, including top of descent fix. The flight times and fuel consumptions for the three cases, including potential savings, are reported in Table 4.



(a) λ-ϕ

(b) t-χ

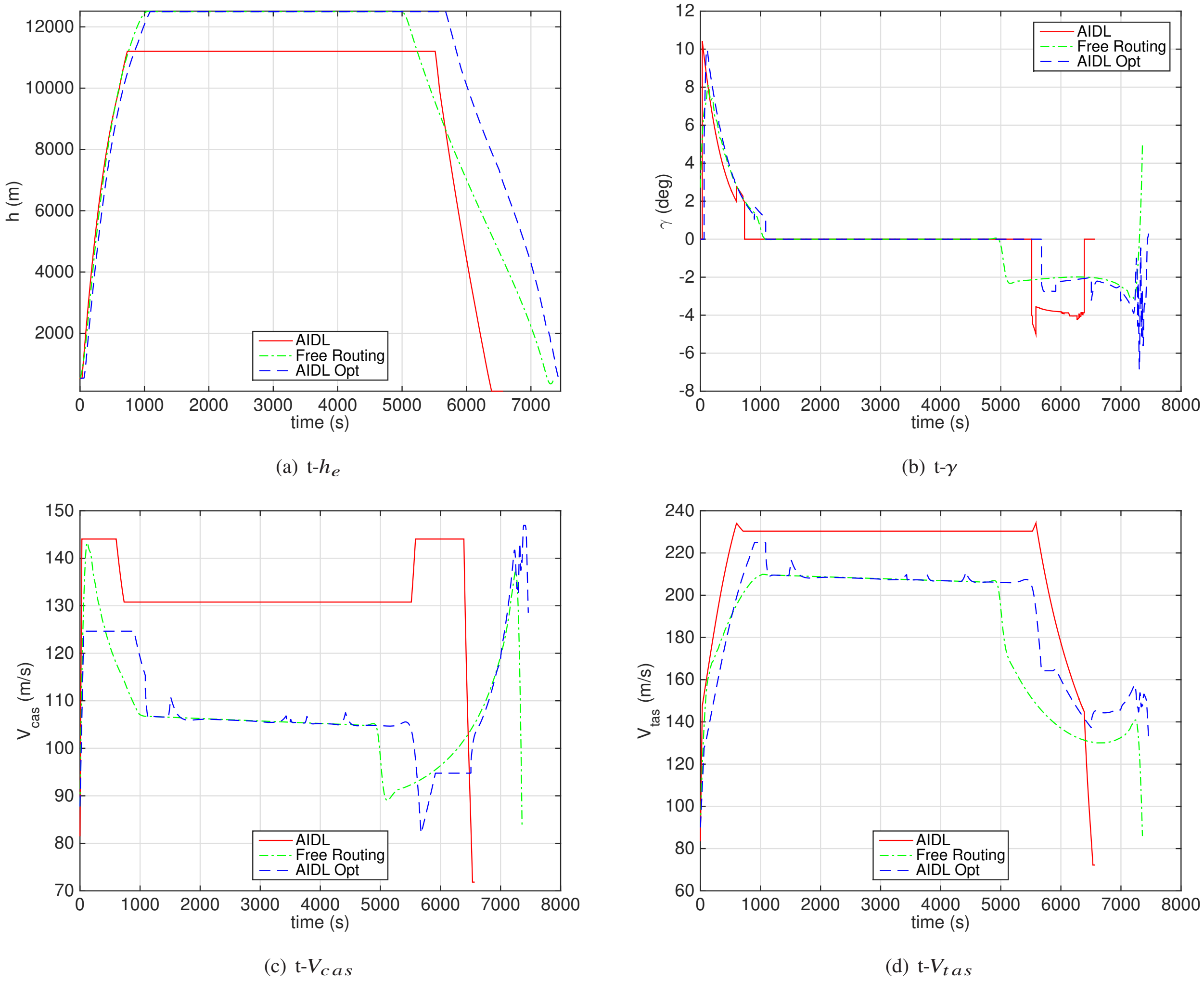
Fig. 13. Lateral path and heading profile.

Table 3. Results: vertical profile

Case	Climb Profile	Cruise Profile	Top of Descent [rad]	Descent Profile
AIDL	*CAS* 144 - *M* 0.78	*h* 11200 - M 0.78	$\lambda = 0.336062$; $\theta = 0.914408$	*M* 0.78 - *CAS* 144
Opt-AIDL	OAL & *CAS* 124.6 - *M* 0.762	*h* 12496 - OSL	$\lambda = 0.312057$; $\theta = 0.912142$	OAL & *M* 0.569 - *CAS* 97
Free Routing	OSL & OAL	OSL & OAL	-	OSL & OAL

Table 4. Results: flight times and consumptions

Case	Flight Time [sec.]	Diff. in Flight Time	Final mass [kg]	Fuel consumption [kg]	Diff. in Fuel consumption
AIDL	6559	Baseline	45523	3780.5	Baseline
Opt-AIDL	7461	+902 sec. (+ 13.75%)	45973.7	3419.8	-450.7 kg (-11.6 %)
Free Routing	7357	+798 sec. (+ 12.17%)	46153	3240.5	-630 kg (-16.2%)



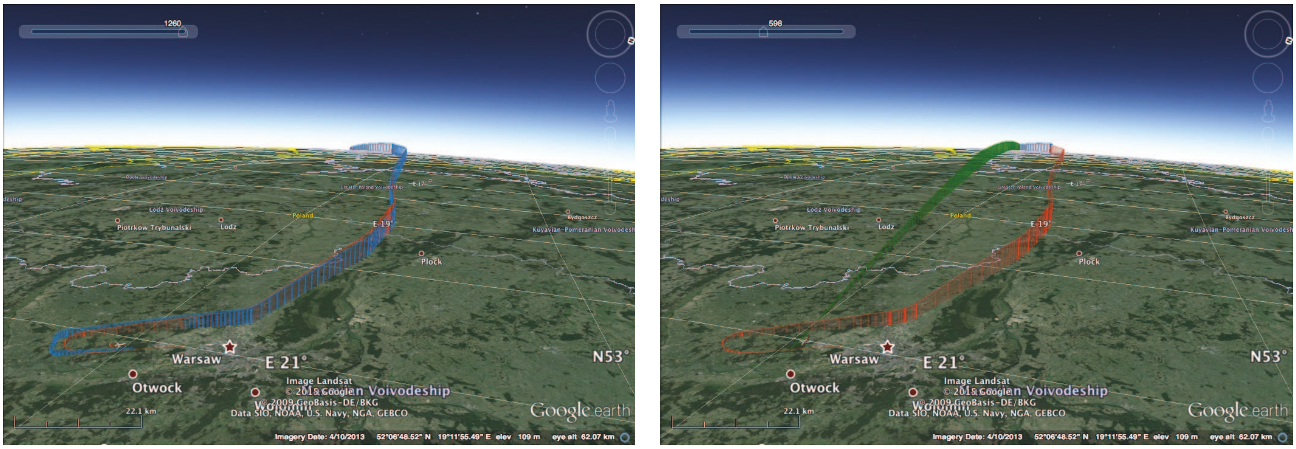
(a) t-h_e

(b) t-γ

(c) t-V_{cas}

(d) t-V_{tas}

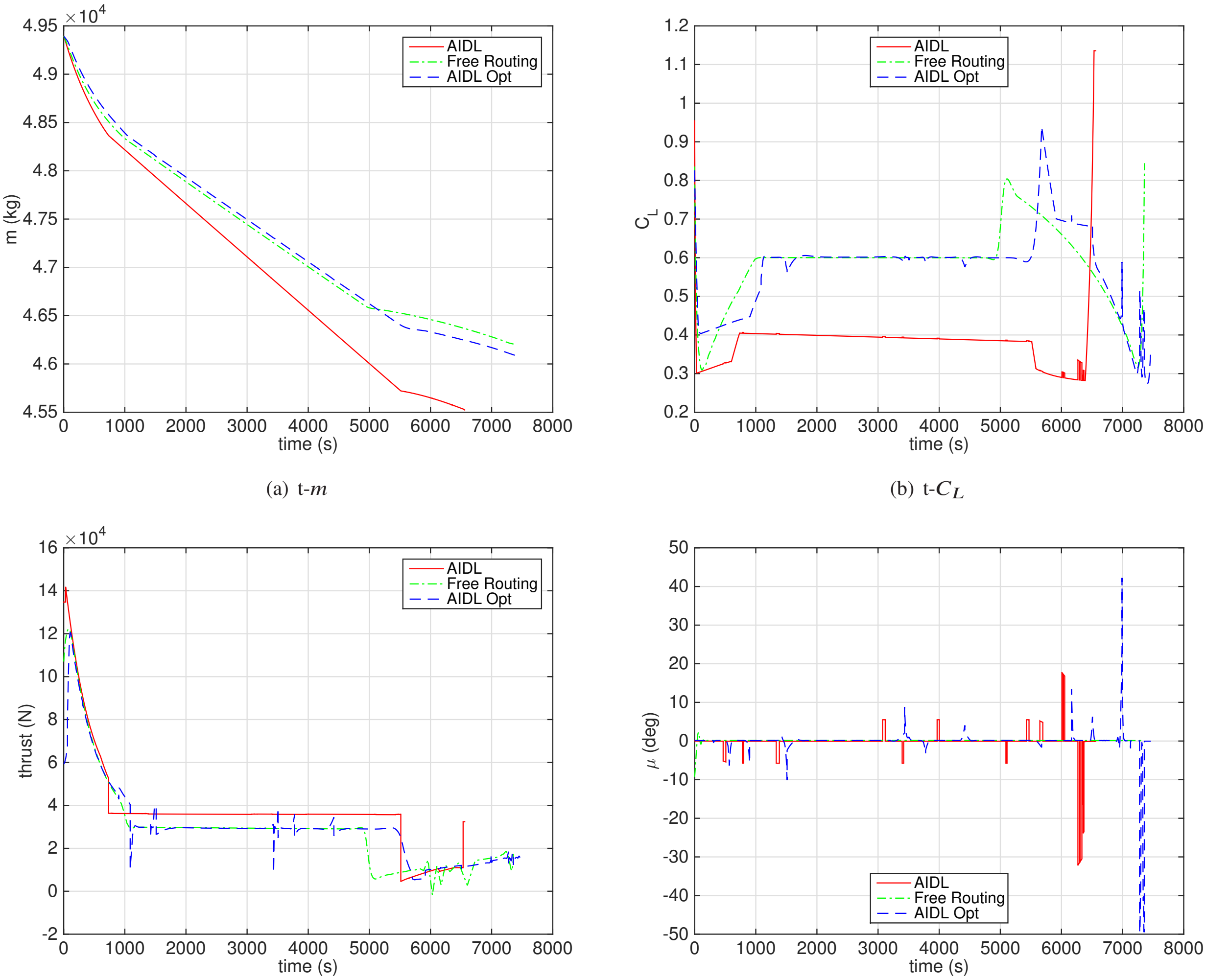
Fig. 14. Altitude profile, flight path angle, calibrated airspeed and true airspeed.



(a) AIDL-OptAIDL

(b) AIDL-FreeRouting

Fig. 15. Pansa Google



(a) t-m

(b) t-C_L

(c) t-T

(d) t-μ

Fig. 16. Mass evolution, lift coefficient, thrust profile and bank angle.

For the sake of presenting a fairer comparison, a heuristic algorithm has been implemented to search for more efficient solution using the current AIDL infrastructure. We explore a subset of possible solutions, allowing ascent/descent Mach to vary from 0.65 to 0.82 (varying by 0.01), ascent/descent CAS to vary from 240 to 330 kts (varying by 10 kts), and cruise Geopotential altitude to vary from 10400 to 12600 m (varying by 200 m). All together 3564 runs. Note that the speed profiles are considered to be symmetrical (ascent/descent). The obtained solutions are presented in Figure 17. It should be pointed out that some of the combination are out of the flight envelope and, thus, infeasible.

According to the Heuristic, the minimum fuel (non-optimized) AIDL aircraft intent would be that with an ascent/descent velocity profile with CAS = 240 kts and M = 0.7, and a Geopotential altitude of h= 12400 m. Its associated fuel consumption and flight times are 3525.9 kg and 7381.9 sec., respectively. Comparing this solution with the opt-AIDL, savings would be of 106 kg. (3%) at the cost of an extra flight time of 71 sec. (1%). These results are presented in Table 5.

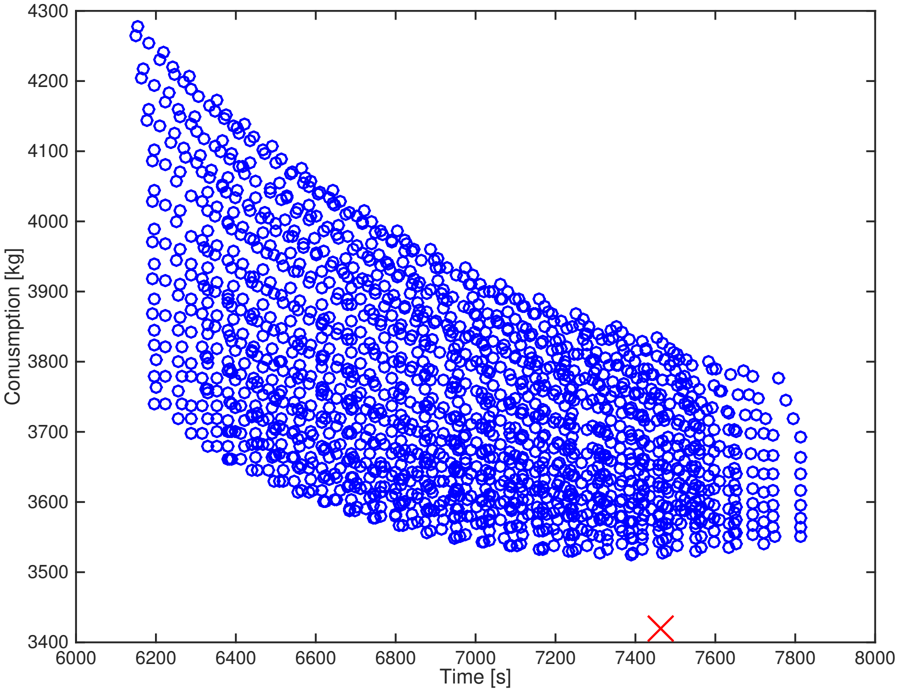


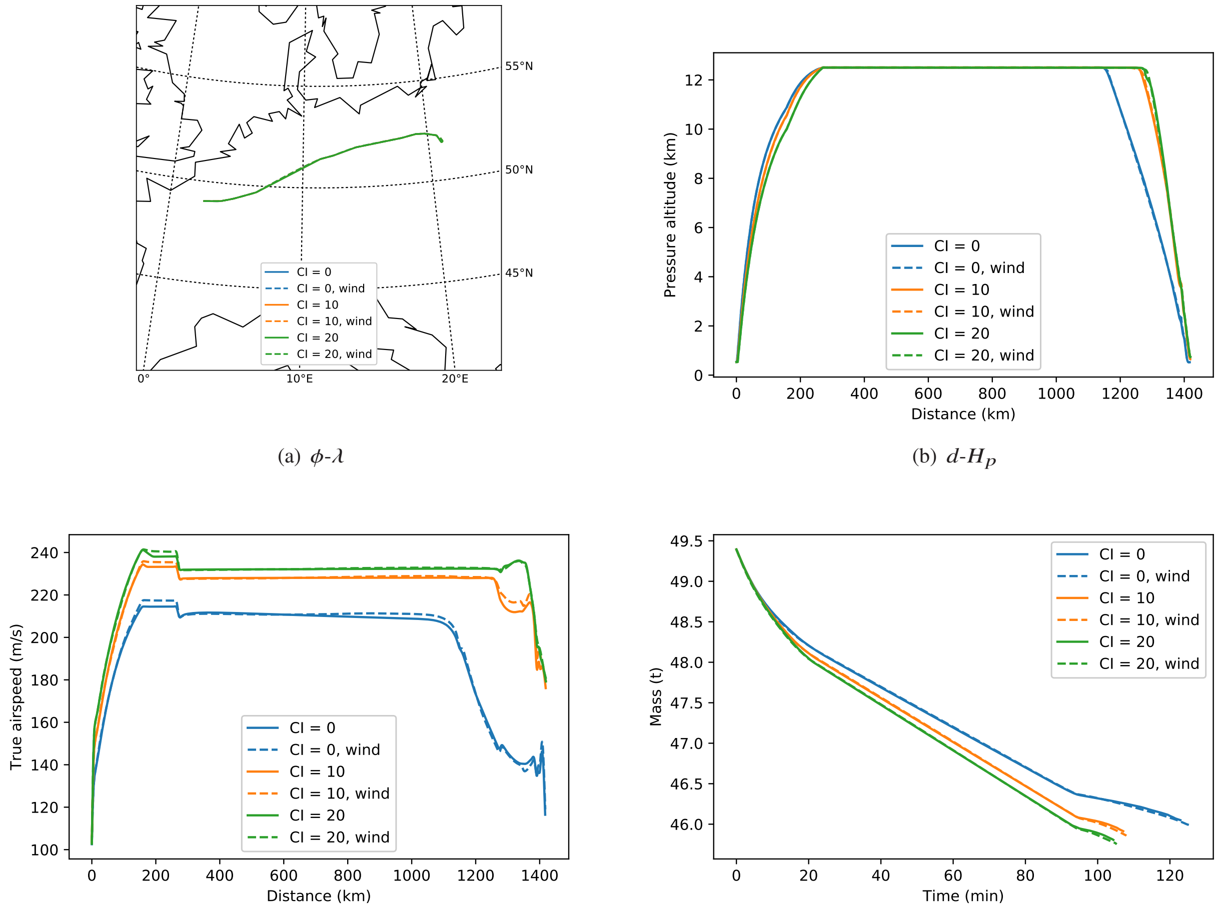
Fig. 17. Heuristic search for conventional AIDL solutions (blue dots representing different Ascent CAS/M, Cruise Altitudes, Descent M/CAS) compared to the opt-AIDL solution (red cross).

Table 5. Heuristic Results: flight times and consumptions

Case	Flight Time [sec.]	Diff. in Flight Time	Final mass [kg]	Fuel consumption [kg]	Diff. in Fuel consumption
Heuristic AIDL	7381,9	Baseline 2	45777,6	3525.9	Baseline 2
Opt-AIDL	7461	+79.1 sec. (+ 1%)	45973.7	3419.8	-106 kg (-3 %)
Free Routing	7357	-24,9 sec. (- 0.3%)	46153	3240.5	-285 kg (-8%)

5.2. Secondary analysis results

The main features of the solution trajectories of the secondary analysis are illustrated in Figure 18. In this analysis, the AIDL optimised intent with BADA 4 as aircraft performance model and different cost indexes has been computed. Also wind in included in some instances. The purpose of the second analysis in to illustrate the suitability of the approach for modeling more realistic flight planning problems. The lateral path is very similar in all cases, as the waypoints leave little room for alternatives. The altitude profile features a steeper climb, but a longer descent for the CI = 0 case; as CI increases, the climb angle decreases while the descent becomes sharper. The true airspeed profile increases as CI rises, as it can be expected; the difference is markedly higher during the descent phase. Finally, the mass evolution shows the cost of reducing flight time in terms of increased fuel burn.



(a) ϕ-λ

(b) d-H_p

(c) d-v

(d) t-m

Fig. 18. BADA 4 results: lateral path, altitude profile, airspeed profile and mass evolution

5.3. Discussion

Efficiency metrics As indicated in Table 4, potential fuel savings for the instance herein solved (a typical intra-european medium-haul flight) can reach up to 450 kg, around 11.6 %. What is more important, the optimised intent simulation yields fuel savings that are very close to the theoretical optimal (associated herein to the free-routing intent). The fuel savings achieved result, in turn, in longer flight times: around 15 min. in this case. This can go against most of airlines' policies, in which flight time is somehow traded-off against fuel consumption using the so called cost index (CI). In this particular case study, a minimum-fuel problem is solved (in other words, a zero cost index problem), and thus the goal is to minimise fuel consumption regardless of flight time (which results in a slower but more efficient flight). The simulation of flights using different cost indexes is herein limited by the aircraft performance model used (BADA 3), which does not include compressibility effect in its drag curve. As a consequence, the fuel consumption penalty due to flying close to transition Mach number is not high enough and, for a high enough cost index, solution trajectories tend to fly at the maximum allowed speeds. More realistic computation could be obtained by using BADA 4, which includes compressibility effects.

Performances The discussion is now turned to analysing flight performances, including both lateral and vertical (speed and altitude) profiles. Figure 13 plots lat-long and heading. It can be observed how both the AIDL reference intent and the AIDL optimised intent follow the same lateral path. On the contrary, AIDL free-routing intent results in a different path. Indeed, most of the inefficiencies above mentioned (comparing AIDL optimised intent and AIDL free-routing intent) can be attributed to the lateral profile, i.e., the AIDL optimised intent is enforced to navigate though a structure of airways connecting waypoints, whereas the lateral solution to the AIDL free-routing intent is, as one would expected, an orthodromic between origin and destination.

Vertical profiles can be decomposed into altitude and speed profiles. The altitude profiles are shown in Figure 14.a. Speed profiles (CAS and TAS) are shown in Figure 14.c and Figure 14.d, respectively. Whereas CAS-MACH climbing profiles are similar, differences in the MACH-CAS descent profiles are more evident (the different CAS-MACH values are due to the inclusion of optimal trigger instead of floating triggers). Indeed, the optimised intent descent results in smoother rates of descent.

Another relevant result is that AIDL optimised intent allows to set an Optimal Speed Law (OSL), which results to the solution to the multiphase optimal control problem. The OSL results in a slightly decreasing speed profile, where the optimal speed at constant altitude slightly decreases as fuel is burnt during the flight. Also, It should be noted that discrepancies between cruising altitudes are somehow biased: whereas the AIDL reference intent is enforced (by means of a floating trigger) to cruise at an existing flight level (flight levels are defined every 1000 ft), the AIDL optimised intent includes an optimal trigger, i.e., the cruising altitude is a variable to be optimised. This results in a cruising altitude that does not necessarily correspond to an allowed flight level and thus would be operationally infeasible. Including however the set of possible flight levels would result in a decision mechanism that is out of the scope of this study.

Figure 16 depicts the three control inputs of the multiphase optimal control problem, namely: lift coefficient; thrust; and bank angle. Using AIDL infrastructure, these can be easily transformed into the real controls, i.e., δ_T, δ_L, and δ_{LD} (coordinated action of ailerons and rudder). It can be noticed that the AIDL reference intent presents a lower lift coefficient when compared to the AIDL optimised intent, corresponding to the higher velocities for the former. Both thrust and bank angle present profiles that are smooth, and respond to a similar structure. Naturally, the bank angle is non-zero when a turn is performed, specially when the aircraft flies the STAR. The thrust profiles are very similar both during ascent (maximum) and descent (idle); however, the AIDL optimised intent ones during the cruise are lower than the AIDL reference intent, associated with lower speeds.

Operational aspects From an operational point of view, the proposed system is suitable for automatic systems based on AIDL. However, when compared with manually-crafted AIDL, the efficiency gains often come at the expense of readability. This is an acceptable trade-off: this system is geared towards a concept of operations with higher levels of automation and flexible trajectories represent only the most general kind of problem that can be solved with the tool. Indeed, an appropriate usage of the available constraints lets the user define simpler optimised intents that generate more readable AIDL sequences. Another possibility, which nevertheless falls out of the scope of this paper, would be to generate a post-processing system that takes the output of a flexible problem and tries to approximate it with readable AIDL, therefore sacrificing a bit of optimality to recover readability (for example, by discretising the lateral path into segments between waypoints, each one with a specified constant airspeed and altitude or rate of climb or descent).

A more significant limitation, within the current concept of operations, is the treatment of the altitude profile. The system handles flexible problems, with a free and continuous altitude profile that does not fit within the current ATM paradigms, based on quantised flight levels. However, the user can, again, make use of the available constraints in order to specify a flight level sequence. Because the transition times would be free for the optimiser to find, even a pre-defined flight level sequence can be flexible enough for the solution to be close to optimal. Alternatively, a post-processing system such as the one proposed in this section could discretise a continuous solution into discrete flight levels.

6. Conclusions

The future ATM to be built around the notion of business trajectory will rely on automated tools related to trajectory prediction in order to define, share, revise, negotiate, and update the trajectory of the aircraft before and during the flight, in some case in near real time. Based on numerical optimal control, this paper illustrates how existing standards on trajectory description, as it is case of AIDL, can be enhanced including optimisation capabilities. Results illustrate how important fuel savings can be achieved while at the same time complying with AIDL rules, so that the trajectory can be defined without ambiguity and thus interchanged and negotiated. The results herein presented represent a key technological milestone towards fulfilling the goals of SESAR, combining the required automation tools and procedures with potential savings in terms of fuel and environmental fingerprint.

Future work is oriented towards improving the computational performance of the system in order to speed up the optimization process. An important factor in this regard is the quality of the initial guess: we are testing improvements to the automated initial guess heuristics in order to improve the convergence of the optimization algorithm. Last but not least, it should be noted that the current multiphase optimal control framework requires a predefined sequence of operations. Different algorithms are being tested to circumvent this drawback, solving cases in which the sequence of events is not predefined a priori, e.g., providing a set of possible triggers or not pre-defining the number of operations needed; these techniques, however, rely on alternative optimisation strategies (combining discrete variables that model decision making processes with continuous variables) as the one that we have described in this paper.

References

[1] SESAR Master Plan, 2nd Edition. SESAR Joint Undertaking; 2012. [Online] Available: http://ec.europa.eu/transport/modes/air/sesar [Retrieved 03/06/2016].

[2] NEXTGEN. Concept of Operations for the Next Generation Air Transport System, 2007.;. "http://www.jpdo.gov/library/NextGen_v2.0.pdf".

[3] Sesar Consortium. The ATM Target Concept, SESAR Definition Phase Milestone Deliverable 3. SESAR Consortium; September 2007. http://www.eurocontrol.int/sesar/public [Retrieved 01/09/2009].

[4] López-Leonés J. The Aircraft Intent Description Language [PhD Thesis]. University of Glasgow. University of Glasgow; 2007.

[5] Besada J, Frontera G, Crespo J, Casado E, López-Leonés J. Automated aircraft trajectory prediction based on a formal intent-related language processing. IEEE Transactions on Intelligent Transportation Systems. 2013;14(3):1067–1082.

[6] Frontera G, Besada J, Bernardos A, Casado E, López-Leonés J. Formal Intent-Based Trajectory Description Languages. IEEE Transactions on Intelligent Transportation Systems. 2014 Aug;15(4):1550–1566.

[7] Bryson AE, Ho Y. Applied Optimal Control. Wiley; 1975.

[8] Betts JT. Practical Methods for Optimal Control and Estimation using Nonlinear Programming. Advances in Design and Control. Society for Industrial and Applied Mathematics; 2010.

[9] Betts JT, Cramer EJ. Application of direct transcription to commercial aircraft trajectory optimization. Journal of Guidance, Control, and Dynamics. 1995;18(1):151–159.

[10] Jacobsen M, Ringertz UT. Airspace constraints in aircraft emission trajectory optimization. Journal of Aircraft. 2010;47(4):1256–1265.

[11] Soler M, Zapata D, Olivares A, Staffetti E. Framework for aircraft 4D trajectory planning towards an efficient air traffic management. Journal of Aircraft. 2012;49(1):341–348.

[12] Soler M, Olivares A, Staffetti E. Multiphase Optimal Control Framework for Commercial Aircraft Four-Dimensional Flight-Planning Problems. Journal of Aircraft. 2015;52(1):274–286.

[13] Soler M, Zou B, Hansen M. Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach. Transportation Research Part C: Emerging Technologies. 2014;48:172–194.

[14] Dalmau R, Prats X. Fuel and time savings by flying continuous cruise climbs: Estimating the benefit pools for maximum range operations. Transportation Research Part D: Transport and Environment. 2015;35(62-71).

[15] Gardi A, Sabatini R, Ramasamy S. Multi-objective optimisation of aircraft flight trajectories in the ATM and avionics context. Progress in Aerospace Sciences. 2016;83:1–36.

[16] European Comission, the FAA and the project AIRE. Atlantic Interoperability Initiative to Reduce Emission (AIRE). Summary results 2010/2011. SESAR Joint Undertaking; 2012. http://www.sesarju.eu/news-press/documents-reports [Retrieved 01/09/2012].

[17] Soler M, Olivares A, Staffetti E. Hybrid optimal control approach to commercial aircraft trajectory optimization. Journal of Guidance, Control, and Dynamics. 2010;33(3):985–991.

[18] Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming. 2006;106(1):25–57.

[19] Eurcontrol. Concpet Document for the Base of aircraft Data (BADA) family 4. Eurocontrol Experimental Center; EECX Technical/Scientific Report No. 12/11/22-57. 2012.

[20] Bellman R. Dynamic Programming. Princeton University Press; 1957.

[21] Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF. The Mathematical Theory of Optimal Processes. Interscience Publishers; 1962.

[22] Yan H, Fahroo F, Ross IM. Accuracy and optimality of direct transcription methods. Advances in the Astronautical Sciences. 2000;105:1613–1630.

[23] Betts JT. Survey of numerical methods for trajectory optimization. Journal of Guidance, Control, and Dynamics. 1998;21(2):193–207.

[24] Trélat E. Optimal control and applications to aerospace: some results and challenges. Journal of Optimization Theory and Applications. 2012;154(3):713–758.

[25] Rao AV. A survey of numerical methods for optimal control. Advances in the Astronautical Sciences. 2009;135(1):497–528.

[26] Kraft D. On converting optimal control problems into nonlinear programming problems. Computational Mathematical Programming. 1985;15:261–280.

[27] Bock HG, Plitt KJ. A multiple shooting algorithm for direct solution of optimal control problems. In: In Proceedings of the 9th International Federation of Automatic Control World Congress. Budapest, Hungary. Pergamon Press; 1984. p. 242–247.

[28] Tsang T, Himmelblau D, Edgar T. Optimal control via collocation and non-linear programming. International Journal of Control. 1975;21(5):763–768.

[29] Ross IM, Karpenko M. A review of pseudospectral optimal control: from theory to flight. Annual Reviews in Control. 2012;36(2):182–197.

[30] Williams P. Hermite-Legendre-Gauss-Lobatto direct transcription methods in trajectory optimization. Advances in the Astronautical Sciences. 2005;120, Part I:465–484.

[31] Hargraves CR, Paris SW. Direct trajectory optimization using nonlinear programming and collocation. Journal of Guidance, Control, and Dynamics. 1987;10(4):338–342.

[32] von Stryk O. Numerical solution of optimal control problems by direct collocation. In: Bulirsch R, Miele A, Stoer J, editors. Optimal Control - Calculus of Variations, Optimal Control Theory, and Numerical Methods. vol. 111 of International Series of Numerical Mathematics. Birkhäuser; 1993. p. 129–143.

[33] Gonzalez-Arribas D, Sanjurjo-Rivo M, Soler M. Optimization of Path-Constrained Systems using Pseudospectral Methods applied to Aircraft Trajectory Planning. In: Workshop on Advanced Control and Navigation for Autonomous Aeroespace Vehicles. IFAC; 2015. .

[34] Garg D, Patterson M, Hager WW, Rao AV, Benson DA, Huntington GT. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. Automatica. 2010;46(11):1843–1851.

[35] Nocedal J, Wright SJ. Numerical Optimization. Springer Verlag; 1999.

[36] Nuic A. User Manual for the Base of Aircraft Data (BADA) Revision 3.6; 2005.
http://www.eurocontrol.int/eec/public [Retrieved 01/09/2009].