HAL
archives-ouvertes.fr

Optimal control approaches for aircraft conflict avoidance using speed regulation : a numerical study

Loïc Cellier, Sonia Cafieri, Frederic Messine

▶ **To cite this version:**

Loïc Cellier, Sonia Cafieri, Frederic Messine. Optimal control approaches for aircraft conflict avoidance using speed regulation : a numerical study. ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management, Jul 2013, Toulouse, France. hal-00867935

HAL Id: hal-00867935

https://hal-enac.archives-ouvertes.fr/hal-00867935

Submitted on 1 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Control Approaches for Aircraft Conflict Avoidance using Speed Regulation: a Numerical Study

Loïc CELLIER [a], Sonia CAFIERI [a] and Frédéric MESSINE [b]
[a] *ÉNAC - University of Toulouse, F-*31055 *Toulouse, France.*
Email: {loic.cellier, sonia.cafieri}@enac.fr
[b] *ÉNSÉÉIHT - University of Toulouse, F-*31071 *Toulouse, France.*
Email: frederic.messine@n7.fr

Abstract.
In this paper a numerical study is provided to solve the aircraft conflict avoidance problem through velocity regulation maneuvers. Starting from optimal control-based model and approaches in which aircraft accelerations are the controls, and by applying the direct shooting technique, we propose to study two different large-scale nonlinear optimization problems. In order to compare different possibilities of implementation, two environments (AMPL and MATLAB) and deterministic local optimization solvers are used. Numerical results are discussed. They show that the considered problem is really difficult to solve to global optimality, as different local minima are found using different methods.

Keywords. air traffic management; conflict avoidance; speed regulation; optimal control; Pontryagin's maximum principle; interior point-based solvers; numerical study.

1. Introduction

Aircraft conflict avoidance is crucial in air traffic management, and it is even more challenging as the air traffic is continuously increasing. The problem is to keep a given minimum safety distance for aircraft along their trajectories. In order to ensure safety within the air traffic, minimum distances between aircraft have to be respected. Aircraft are said in *conflict* if they are too close to each other. The associated problem is called *aircraft conflict avoidance*. The actual norm of the separation between two aircraft is defined as follows: 1000ft vertically and 5NM horizontally (with the units: 1NM (nautical mile) = $1,852$m; 1ft (feet) = 0.3048m). In this context, various approaches to the aircraft conflict avoidance problem have been provided. A survey is given by *Kuchar and Yang* in [9]. The problem of aircraft conflict avoidance is largely investigated in terms of aircraft trajectory deviations using altitude or heading changes. Recently, speed regulation appeared to be more and more relevant as a maneuver for solving the problem. In 2008, the European ERASMUS project (*Bonini et al.* [1]) proposed to use small speed changes to solve aircraft conflicts. Based on velocity variations, new models and solution approaches, gener-

ally coming from mixed-integer programming, have been developed (see, *e.g.*, *Pallottino et al.* [10], *Rey et al.* [12], *Cafieri* [2,3]).

In this paper, we focus on numerical ways for solving the aircraft conflict avoidance problem formulated as an optimal control problem. In Section 2, we present the optimal control problem based on velocity changes that we address in this work. By defining a zone where aircraft are potentially in conflict and by using the necessary conditions of the Pontryagin maximum principle, we reformulate the optimal control problem into another equivalent one. In Section 3, we apply the direct shooting method to provide two equivalent NLP (Nonlinear Programming) large-scale optimization problems. In Section 4, a numerical study on the solution of the NLP problems is done using two different modeling languages: AMPL and MATLAB. Section 5 concludes the paper.

2. Optimal control models

The aircraft conflict avoidance problem that we address in this paper has the following assumptions: (i) en-route flights (cruise phase); (ii) tactical phase (few time before potential conflicts); (iii) planar cartesian positions of the aircraft; (iv) no wind.

In the following optimal control problem ($\mathscr{P}$) based on velocity variations, we choose to minimize a quadratic-energy cost function. The controls are the accelerations of each aircraft i and they are denoted by u_i. Let I denotes the set $\{1,\cdots,n\}$, where n is the number of aircraft evolving through a time window $[t_0,t_f]$. The state variables are the velocity v_i and the (planar) position x_i of the aircraft; $x_i(t) \in \mathbb{R}^2$ represents the planar cartesian coordinates. Let $d_i \in \mathbb{R}^2$ denotes the fixed planar direction of the aircraft i. The state variables are subject to dynamical laws yielding to differential equations on the velocity and the position; this defines the system of state equations of our model. In this optimal control problem, the initial positions and velocities of the aircraft are known as well as the final velocities (the final positions of the aircraft are free). One of the main difficulties of this problem is given by the separation constraints. These ones are concave depending on the positions of the aircraft.

$$
(\mathscr{P}) \begin{cases} \min\limits_{u} \quad \sum\limits_{i=1}^{n} \int_{t_0}^{t_f} u_i^2(t)dt \\ s.t. \\ \qquad \dot{v}_i(t) = u_i(t) & \forall t \in [t_0,t_f], \forall i \in I \\ \qquad \dot{x}_i(t) = v_i(t)d_i & \forall t \in [t_0,t_f], \forall i \in I \\ \qquad \underline{u_i} \leq u_i(t) \leq \overline{u_i} & \forall t \in [t_0,t_f], \forall i \in I \\ \qquad \underline{v_i} \leq v_i(t) \leq \overline{v_i} & \forall t \in [t_0,t_f], \forall i \in I \\ \qquad x_i(t_0) = x_i^0 \qquad v_i(t_0) = v_i^0 & \forall i \in I \\ \qquad v_i(t_f) = v_i^f & \forall i \in I \\ \qquad \| x_i(t) - x_j(t) \|^2 \geq D^2 & \forall t \in [t_0,t_f], \forall i < j,\ (i,j) \in I^2 \end{cases}
$$

To solve this problem, we can apply direct shooting methods on $(\mathscr{P})$. In *Cellier et al.* [4], we observed that we can decompose the problem $(\mathscr{P})$ into subproblems by considering different '*zones*' with respect to the separation constraints. Thus, a temporal '*zone*' denoted by $[\underline{t_Z}, \overline{t_Z}] \subseteq [t_0, t_f]$ is defined. In this zone, the aircraft are potentially in conflict while outside the zone the separation constraints are proved to be always satisfied. Hence, by previously identifying the time zone $[\underline{t_Z}, \overline{t_Z}]$, we can split the optimal control problem into two parts, respectively from t_0 to $\overline{t_Z}$ and from $\overline{t_Z}$ to t_f. By using the necessary conditions of Pontryagin's Maximum Principle (*Pontryagin et al.* [11]), we proved in *Cellier et al.* [4] that an analytical solution can be provided for all time t in $[\overline{t_Z}, t_f]$. Thus, the problem $(\mathscr{P})$ can be reformulated into an equivalent optimal control problem as follows:

$$
(\mathscr{P}_Z) \begin{cases} \min\limits_{u} \quad \sum\limits_{i=1}^{n} \int_{t_0}^{\overline{t_Z}} u_i^2(t)dt + \dfrac{(v_i^f - v_i(\overline{t_Z}))^2}{t_f - \overline{t_Z}} & \\ s.t. & \\ \qquad \dot{v}_i(t) = u_i(t) & \forall t \in [t_0, \overline{t_Z}], \forall i \in I \\ \qquad \dot{x}_i(t) = v_i(t) d_i & \forall t \in [t_0, \overline{t_Z}], \forall i \in I \\ \qquad \underline{u_i} \leq u_i(t) \leq \overline{u_i} & \forall t \in [t_0, \overline{t_Z}], \forall i \in I \\ \qquad \underline{v_i} \leq v_i(t) \leq \overline{v_i} & \forall t \in [t_0, \overline{t_Z}], \forall i \in I \\ \qquad \underline{u_i} \leq \dfrac{v_i^f - v_i(\overline{t_Z})}{t_f - \overline{t_Z}} \leq \overline{u_i} & \forall i \in I \\ \qquad x_i(t_0) = x_i^0 \qquad v_i(t_0) = v_i^0 & \forall i \in I \\ \qquad \| x_i(t) - x_j(t) \|^2 \ \geq \ D^2 & \forall t \in [t_0, \overline{t_Z}], \forall i < j, \ (i,j) \in I^2 \end{cases}
$$

Notice that the fixed final velocity v_i^f appears in the objective function of $(\mathscr{P}_Z)$, so it is not necessary to keep it as a final condition as in problem $(\mathscr{P})$. Problem $(\mathscr{P}_Z)$ only depends on the time window $[t_0, \overline{t_Z}]$. For all time t in $[\overline{t_Z}, t_f]$, the solution is given by: $u_i(t) = \frac{v_i^f - v_i(\overline{t_Z})}{t_f - \overline{t_Z}}$, $v_i(t) = \frac{v_i^f - v_i(\overline{t_Z})}{t_f - \overline{t_Z}} \, t + v_i^0$ and $x_i(t) = \frac{d_i}{2} \frac{v_i^f - v_i(\overline{t_Z})}{t_f - \overline{t_Z}} \, t^2 + v_i^0 d_i \, t + x_i^0$, see *Cellier et al.* [4].

To solve the problem $(\mathscr{P}_Z)$, we apply a direct shooting method based on time discretization. In the following section we discuss different possible formulations.

3. Applying the direct shooting method

By applying the direct shooting method to the optimal control problem $(\mathscr{P}_Z)$, the ordinary differential equations are computationally treated by discretizing the variables appearing in the equations with respect to the time. Numerical integrators (for example, Euler-type integrators) are used to approximate the differential equations. In *Cellier et al.* [4], we proposed to use two discretization steps, one tight enough to check if separation constraints are respected (detection step), and another one to compute the value of

control variables, this step (resolution step) is larger than the previous one. Let ρ be the ratio between detection and resolution steps (*e.g.,* if the detection (resp. resolution) step is $15''$ (resp. $1'$), ρ corresponds to 4).

By discretizing the time in $[t_0, \overline{t_Z}]$ into N_Z steps ($N_Z = \lceil \frac{\overline{t_Z}}{h} \rceil$), and then by discretizing the controls u_i and the state variables v_i, x_i (for all aircraft i), we obtain the following NLP problem:

$$
(P) \begin{cases}
\min\limits_{(x,v,U)\in\mathbb{R}^{2nN_Z.nN_Z.nm_Z}} & h \sum\limits_{i\in I} \left(\rho \sum\limits_{l\in M_Z} (U_i^{(l)})^2 + (\rho-1)\left(U_i^{(m_Z-1)}\right)^2 + \left(\dfrac{v_i^f - v_i^{(N_Z)}}{t_f - N_Z h} \right)^2 \right) & \\
s.t. & & \\
& v_i^{(k+1)} = \mathrm{NUM}_{v_i}(U_i^{(\lfloor \frac{k}{\rho} \rfloor)}) & \forall k \in K, \forall i \in I \\
& x_i^{(k+1)} = \mathrm{NUM}_{x_i}(v_i^{(k)}) & \forall k \in K, \forall i \in I \\
& \underline{u_i} \leq U_i^{(l)} \leq \overline{u_i} & \forall l \in \overline{M_Z}, \forall i \in I \\
& \underline{u_i} \leq \dfrac{v_i^f - v_i^{N_Z}}{t_f - N_Z \times h} \leq \overline{u_i} & \forall i \in I \\
& \underline{v_i} \leq v_i^{(k)} \leq \overline{v_i} & \forall k \in \overline{K}, \forall i \in I \\
& x_i^{(0)} = x_i^0 \qquad v_i^{(0)} = v_i^0 & \forall i \in I \\
& \| x_i^{(k)} - x_j^{(k)} \|^2 \geq D^2 & \forall k \in \{\lfloor \frac{t_Z}{h} \rfloor, \ldots, \lceil \frac{\overline{t_Z}}{h} \rceil\}, \forall i < j, (i,j) \in I^2
\end{cases}
$$

where NUM is a numerical integrator depending on the state variables. The state variables v_i and x_i are computed using a discretization step h while the control variables U_i are computed using a discretization step ρh; $U_i^{(\lfloor \frac{k}{\rho} \rfloor)}$, $v_i^{(k)}$ and $x_i^{(k)}$ denotes values of the acceleration, velocity and respectively position at time $t_k = kh$. Variables are indexed on sets $K = \{0, \ldots, N_Z - 1\}$, $\bar{K} = \{0, \ldots, N_Z\}$, $M_Z = \{0, \ldots, m_Z - 2\}$ and $\overline{M_Z} = \{0, \ldots, m_Z - 1\}$ with $m_Z = \frac{N_Z}{\rho}$.

Problem (P) has $nN_Z\left(3 + \frac{1}{\rho}\right)$ variables and $3n(N_Z - 1)$ equality constraints and $\frac{n(n-1)}{2}\left(\lceil \frac{\overline{t_Z}}{h} \rceil - \lfloor \frac{t_Z}{h} \rfloor\right)$ inequality constraints (separation constraints) and $2n(m_Z + N_Z + 1)$ bound constraints.

Notice that the NLP only depends on the discretized controls $U_i^{(l)}$, as the discretized state varibles can be calculated from the equality constraints. Hence, we can provide a more compact NLP formulation as follows:

$$
(P_c)\begin{cases}
\min\limits_{U\in\mathbb{R}^{nm_Z}} \quad h\sum\limits_{i\in I}\left(\rho\sum\limits_{l\in M_Z}(U_i^{(l)})^2+(\rho-1)\left(U_i^{(m_Z-1)}\right)^2+\left(\dfrac{v_i^f-\mathrm{NUM}_{v_i}(U_i^{(\lfloor\frac{N_Z}{\rho}\rfloor)})}{t_f-N_Z h}\right)^2\right)\\
s.t.\\
\quad \underline{u_i}\le U_i^{(l)}\le\overline{u_i} & \forall l\in\overline{M_Z},\forall i\in I\\
\quad \underline{u_i}\le\dfrac{v_i^f-\mathrm{NUM}_{v_i}(U_i^{(\lfloor\frac{N_Z}{\rho}\rfloor)})}{t_f-N_Z\times h}\le\overline{u_i} & \forall i\in I\\
\quad \underline{v_i}\le\mathrm{NUM}_{v_i}(U_i^{(\lfloor\frac{N_Z}{\rho}\rfloor)})\le\overline{v_i} & \forall i\in I\\
\quad x_i^{(0)}=x_i^0\qquad v_i^{(0)}=v_i^0 & \forall i\in I\\
\quad \|\,\mathrm{NUM}_{x_i}(\mathrm{NUM}_{v_i}(U_i^{(\lfloor\frac{k-1}{\rho}\rfloor)}))-\mathrm{NUM}_{x_j}(\mathrm{NUM}_{v_j}(U_j^{(\lfloor\frac{k-1}{\rho}\rfloor)}))\,\|^2\ \ge\ D^2, & \\
\qquad\qquad \forall k\in\{\lfloor\frac{t_Z}{h}\rfloor,\ldots,\lceil\frac{\overline{t_Z}}{h}\rceil\},\forall i<j,(i,j)\in I^2
\end{cases}
$$

Problem (P_c) has nm_Z variables and $4n+\frac{n(n-1)}{2}\left(1+\lceil\frac{\overline{t_Z}}{h}\rceil-\lfloor\frac{t_Z}{h}\rfloor\right)$ inequality constraints (separation constraints) and $2nm_Z$ bound constraints. Thus, the more compact problem (P_c) has a smaller number of variables and constraints. On the other hand, in (P_c) a number of mathematical expressions (arising in the objective and the constraints) depend implicitly on the control variables.

4. Numerical implementations and experiments

We implement the two formulations (P) and (P_c) in two different computational environments, respectively MATLAB and AMPL [6]. We solve the NLP problems using the following solvers:

- MATLAB/`fmincon` selecting the 'sqp' solver (based on a sequential quadratic programming method);
- MATLAB/`fmincon` selecting the 'interior-point' solver;
- AMPL/`SNOPT` (based on an active set method, see [8]);
- AMPL/`IPOPT` (based on an interior point method, see [13]).

AMPL is a declarative modeling language for mathematical programming, therefore it does not allow us to implement the compact formulation (P_c). On the contrary, using MATLAB we can implement both formulations (P) and (P_c).

Computational experiments are carried out using six different possibilities: the two solvers within AMPL applied to (P) and the two solvers within MATLAB applied to (P) and (P_c).

Data problems were generated with the following characteristics: the trajectory paths are straight; the horizontal separation norm is 5NM; the initial velocity for each air-

craft i corresponds to $v_i^0 = 447$NM/h; the velocities are bounded based on the ERASMUS directives [7] by a small range: $[0.94\ v_i^0,\ 1.03\ v_i^0]$; the accelerations are bounded, based on Eurocontrol's Base of Aircraft Data (BADA) [5], $\overline{u_i} = -\underline{u_i} = 4000$NM/h^2; the problem-configuration time window corresponds to $[t_0, t_f]$ with $t_0 = 0$ and $t_f = 1$h (except for 'pb7', where $t_f = 45'$); terminal conditions are to return to the initial velocities (i.e, $v_i(t_f) = v_i^f = v_i^0 = 447$NM/h).

The number of aircraft n, the number of *potential conflicts*, the *zone* duration percentage with respect to the whole time window, the number of variables and the numbers of constraints for the two formulations (P) and (P_c) are reported in Table 1.

Table 1. Configuration data: number of aircraft, number of *potential conflicts*, *zone* duration purcentage with respect to the whole time window, number of variables and number of constraints for the two formulations .

ID	number of aircraft	number of conflicts	*zone* time percentage	(P) form. var.	(P) form. constr.	(P_c) form. var.	(P_c) form. constr.
pb1	7	3	15.77	2821	5593	217	2555
pb2	8	4	15.77	3536	7368	272	3560
pb3	9	6	21.99	4095	8820	315	4410
pb4	6	3	4.56	2574	4641	198	1869
pb5	8	4	4.56	3432	6340	264	2644
pb6	5	5	26.97	3120	5910	240	2550
pb7	2	1	6.08	546	936	42	348

In our numerical experiments, for all the instances the detection/resolution steps are $h = 15''$ and $\rho h = 1'$ ($\rho = 4$). Notice that the acceleration controls are piecewise constant. For all the instances, the *starting points* are deduced from the controls equal to 0 (for (P) all the variables velocity and position are computed from the null control).
The numerical tests are performed on two PC computers with 2.53GHz/4*GB* for the solvers using AMPL and with 3.2GHz for solvers using MATLAB.

We first performed tests using the two MATLAB/`fmincon` algorithms 'sqp' and 'interior-point', applied on two formulations (P) and (P_c). We remark that:

- The two algorithms applied on formulation (P) are unable to provide a solution within the time limit $10'$. Thus, for MATLAB solvers, the use of formulation (P_c) appears to be more robust.
- 'sqp' algorithm is also inefficient on formulation (P_c). Only one instance is solved: 'pb7' (the smallest one with only two aircraft). In this case, 'sqp' provides an answer (in 19 iterations) with an objective value of 1835.4 within 54.351 seconds. Notice that this local minimum is better than the one provided by the 'interior-point' algorithm of MATLAB but it is still greater than the one obtained by IPOPT (see Table 2 discussed below).

Hence, only the solutions corresponding to the MATLAB/`fmincon`-'interior-point' algorithm applied to formulation (P_c) are considered in the following.

In Table 2, we provide numerical results obtained using the AMPL environment and SNOPT and IPOPT solvers on formulation (P), and MATLAB with the `fmincon`-'interior-point' routine on formulation (P_c) only. In the two first columns, results are re-

Table 2. Comparison of numerical results: (P) with AMPL/SNOPT, (P) with AMPL/IPOPT, (P_c) with MATLAB/fmincon-'interior-point' algorithm.

	(P) with AMPL/SNOPT			(P) with AMPL/IPOPT			(P_c) with MATLAB/fmincon		
	nb_it.	obj_value	time	nb_it.	obj_value	time	nb_it.	obj_value	time
pb1	3100	4339.58	14.477	276	4339.58	5.413	55	5824.4	43.415
pb2	4056	5180.19	19.048	245	5180.19	10.621	–	–	$> 10'$
pb3	–	–	$> 10'$	556	39742.2	52.869	–	–	$> 10'$
pb4	2804	2882.6	14.726	1175	2882.6	56.474	48	3576.9	22.979
pb5	5384	3843.47	65.193	1565	3843.47	116.189	45	4395.2	41.668
pb6	3924	4553.92	9.126	394	8282.67	19.701	130	22235	127.765
pb7	–	–	$> 10'$	91	1631.47	0.484	100	3152.6	3.947

ported for two different solvers SNOPT and IPOPT which use the same formulation (P), the same environment AMPL and the same computer. Hence, these results are directly comparable. We remark that:

- IPOPT always provide a local solution while SNOPT fails on two cases 'pb3' and 'pb7'.
- All the local solutions correspond, except for 'pb6' where SNOPT finds a better local minimum than IPOPT.
- When the local minima found are the same, IPOPT is faster than SNOPT on two instances 'pb1' and 'pb2', and it is the contrary on the two instances 'pb4' and 'pb5'. This may depend on the size of the zone: 15.77% for 'pb1' and 'pb2' and 4.56% for 'pb4' and 'pb5'.
- As expected for an active-set solver compared to an interior point solver, the number of iterations performed by SNOPT to find a local solution is much more larger than the one of IPOPT, even though the iterations of SNOPT are faster.

According to these numerical results, IPOPT appears to be more robust than SNOPT. It seems however quite difficult to establish which solver is faster than the other. Notice that the IPOPT presolver reduced considerably the size of the problems: *e.g.,* on problem 'pb3', we have 4095 variables and 8820 constraints (see Table 1), and they are reduced by the IPOPT presolver to 2946 variables and 4575 constraints.

Concerning the numerical results provided in Table 2 by the MATLAB/fmincon-'interior-point' algorithm applied on (P_c), we remark that:

- All the local solutions are greater than those provided by IPOPT and SNOPT.
- The number of iterations is in general quite low. This implies that one iteration of the MATLAB interior point solver is more expensive than an iteration of the interior point solver in the AMPL environment.
- MATLAB/fmincon-'interior-point' algorithm is unable to provide a solution within the time limit $10'$ for the two most difficult instances, even using the compact formulation (P_c).
- The CPU-times are generally greater than those provided by IPOPT and SNOPT; although the CPU-times are difficult to compare because we used two different computers.

Therefore, the solutions provided by MATLAB appear to be less interesting compared to those provided via the AMPL environment on the addressed problem.

We notice that the local minima of (P) are also local minima of (P_c); this is easily proved by taking as a starting point of the 'interior-point' and 'sqp' algorithms of MATLAB the local minima coming from the use of the IPOPT.
We also remark that the different solutions provided by the distinct applied optimization solvers show that the addressed large-scale NLP problems exhibit several local minima and are computationally difficult to solve to global optimality.

5. Conclusion

We discussed about the numerical solution of an optimal control model to address the aircraft conflict avoidance problem with velocity regulation. By decomposing the optimal control problem into zones and by applying the direct shooting method, two equivalent but distinct large-scale NLP problems were generated. We then use different solvers and two computational environments, MATLAB and AMPL, to solve these NLP problems. Numerical results show clearly that on formulation (P) the solvers used in the AMPL environment are the most efficient. Moreover, the IPOPT solver using AMPL appears to be the most robust. The numerical study discussed in this paper also proved that the addressed NLP problems are difficult to solve to global optimality, as different approaches generally yield to different local minima.

References

[1] D. Bonini, C. Dupré, and G. Granger. How ERASMUS can support an increase in capacity in 2020. In *Proceedings of CCCT: the 7th International Conference on Computing, Communications and Control Technologies*, 2009.

[2] S. Cafieri. A mixed-integer optimization model for air traffic deconfliction. In *Proceedings of GOW: the Global Optimization Workshop*, 2012.

[3] S. Cafieri and N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization, DOI 10.1007/s10898-013-0070-1*, 2013.

[4] L. Cellier, S. Cafieri, and F. Messine. Hybridizing direct and indirect optimal control approaches for aircraft conflict avoidance. In *Proceedings of ADVCOMP: the 6th International Conference on Advanced Engineering Computing and Applications in Sciences*, 2012.

[5] EEC. User manual for the base of aircraft data. Technical report, EUROCONTROL Experimental Centre, 2008.

[6] R. Fourer, D. Gay, and B. Kernighan. *AMPL: a modeling language for mathematical programming*. Thomson/Brooks/Cole, 2003.

[7] G. Gawinowski, J.-L. Garcia, R. Guerreau, R. Weber, and M. Brochard. ERASMUS: a new path for 4d trajectory-based enablers to reduce the traffic complexity. In 26^{th} *Digital Avionics System Conference, Dallas, USA*, 2007.

[8] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[9] J. Kuchar and L. Yang. A review of conflict detection and resolution modeling methods. *IEEE Trans. on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[10] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):3–11, 2002.

[11] L. Pontryagin, V. Boltyanski, R. Gamkrelidze, and E. Michtchenko. *Théorie mathématique des processus optimaux*. Editions Mir, Moscou, 1974.

[12] D. Rey, C. Rapine, R. Fondacci, and N.-E. E. Faouzi. Potential air conflicts minimization through speed regulation. *Transportation Research Board*, 2012.

[13] A. Wächter and L. Biegler. On the implementation of primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–27, 2006.