

# A Constrained Inverse Modeling Approach for Trajectory Optimization

Reiko Mueller\*, Gertjan Looye†

*Deutsches Zentrum für Luft- und Raumfahrt, Münchner Str. 20, 82234 Wessling, Germany*

In aircraft trajectory optimization, modeling almost always relies on the usage of standard (forward) three- or more degrees of freedom equations of motion for the aircraft. The usual practice for simulating the trajectories is either to continuously trim the aircraft along the desired flight path and speed profile, and/or to use reduced point mass models that are defined in the vertical plane only. Alternative approaches became prominent in recent years, whereas in this paper the inverse modeling technique and an important extension to it is considered. First, inverse models have several advantages for trajectory optimization, the most important ones being the faster computation time and easy generation of start solutions, compared to standard forward models. In this work it is combined with speed- and flightpath- control loops to cancel the position error introduced during model integration. Still, an important problem remains: saturation limits and dynamics of the control systems used in the actual forward model are normally not considered when inverting the model equations, therefore losing the information of the limitations of the aircraft. In the presented case, the inversion becomes invalid, when the aircraft thrust is outside of the limits due to the commands of the inverse model inputs. This problem is addressed in the model by a pseudo-control-hedging (PCH) approach for altitude and velocity commands. By this means the inverse model inputs are automatically altered depending on the error between commanded and available thrust and the generated trajectory is rendered flyable.

## I. Introduction

In the trajectory optimization field several approaches exist to generate and optimize aircraft flightpaths with respect to a certain criterion. For example highly dynamic trajectories for maneuverable aircraft like fighters or unmanned aerial vehicles (UAVs) require a model representation that depicts all nonlinear effects for the trajectory optimization. Models yielding the full six degree-of-freedom rigid body equations or even more complex elastic models are the best-suited choice there. On the other hand, for large transport aircraft which are strongly constrained by prescribed flightplans and operational requirements, simpler approaches like 2-DOF modeling of the movement in the vertical plane may suffice.

In this work, the focus regarding optimization lies on the design of trajectories which are optimal with respect to several objectives at a time. Since passenger aircraft routes are considered, these can be of economical (e.g. fuel costs) or ecological (e.g. CO<sub>2</sub> emissions) nature. The user should also be able to choose from a set of optimal solutions with different criteria weighting, which leads to a multiobjective optimization problem. In typical simulations concerning air traffic management (ATM) operations, the standard is to use aircraft performance models based on the Eurocontrol Base of Aircraft Data<sup>1</sup> (BADA). These models have two degrees of freedom (true airspeed and altitude), and therefore approximate the aircraft's motion in the vertical plane. The rationale behind the neglect of the horizontal plane is that the time, in which the aircraft operates outside a steady-state wings-level flight, is relatively short.

Previous studies showed that an inverse modeling approach enhances the accuracy of results while still keeping the simulation times low.<sup>2</sup> In most cases however, the actuator limits and dynamics are not considered in the inversion process, since they would introduce further difficulties in generating the model inputs. These include

---

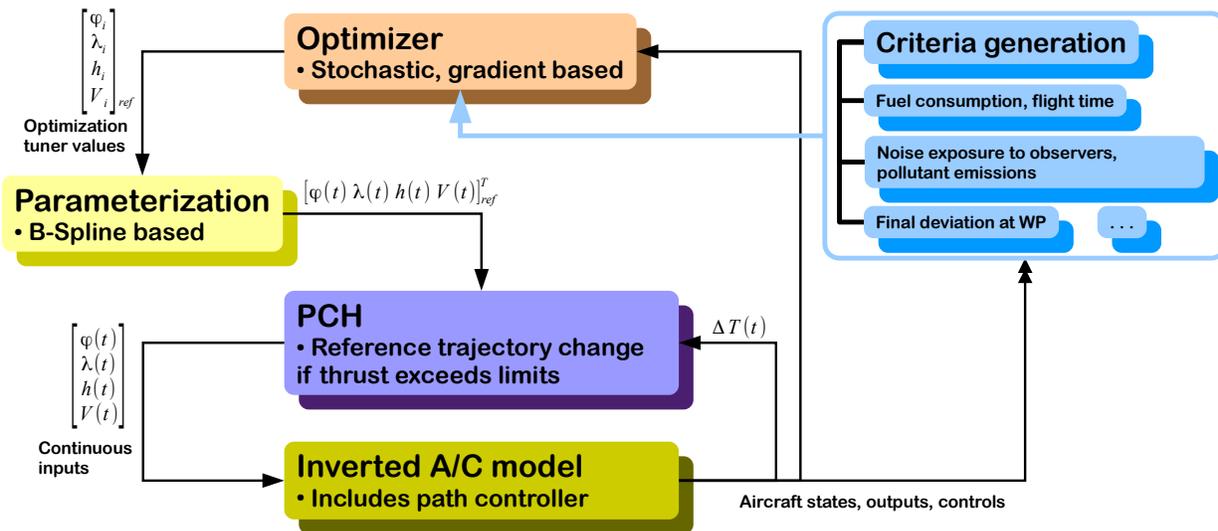
\*Research assistant, Robotics and Mechatronics Center, Institute of System Dynamics and Control, Department of Aircraft Systems Dynamics

†Senior researcher, Robotics and Mechatronics Center, Institute of System Dynamics and Control, Department of Aircraft Systems Dynamics

additional differentiations of the input functions to enable inversion of the actuator dynamics accompanied by numerical problems when integrating the whole system. If the inputs to the system are not chosen carefully, the model calculates infeasible commands for the aircraft which in turn yields unrealistic results. To prevent this, a pseudo-control-hedging-like approach is introduced in this work in order to mainly bound the control inputs for the vertical flight plane, which in turn makes the trajectories flyable. This is the main contribution of this work.

A process scheme with the layout of the trajectory optimization is shown in figure 1, the different blocks are explained in the first part of the paper: In section II, the optimization problem formulation is explained, which also includes a discussion on the parameterization and criteria calculation (sections II.A, II.B). The used models and their interconnections are discussed in section III, with the aircraft model, path controller, as well as the PCH-module (sections III.C, III.B, III.D).

The second part consists of results obtained with the PCH-module in comparison to the plain inverse formulation as well as a single- and multiobjective optimization results for a typical airport approach procedure in section IV. The final section V provides conclusions to the presented results.



**Figure 1:** Trajectory optimization problem with pseudo control hedging block

## II. Trajectory optimization

The problem is discretized as a direct single shooting optimal control problem with the optimization criterion

$$\min_u J(x, u, p) \quad (1)$$

with objective functions  $J$ , design parameters  $p$ , system states  $x$  and controls  $u$ , and is subject to the constraints

$$\dot{x}(t) = f(t, x(t), u(t), p) \quad (\text{Model dynamics}), \quad (2)$$

$$r_{eq} = 0 \quad (\text{Equality constraints}), \quad (3)$$

$$r_{iq} \geq 0 \quad (\text{Inequality constraints}). \quad (4)$$

The goal is to find a continuous control function  $u^*$ , that minimizes  $J$  and fulfills all constraints. Since this problem exhibits infinite dimensionality, the control function  $u$  must be parameterized. This is realized by a B-Spline approximation (see subsection II.A). It reduces the problem to the dimension of a parameter set for the B-Splines in the control function.

The optimization is realized using the optimization tool MOPS (Multi Objective Parameter Synthesis) which is developed and maintained in the DLR-RMC's Institute of System Dynamics and Control.<sup>3</sup> The tool itself

includes multiple solvers (gradient-based, stochastic), worst-case-search, Monte-Carlo analysis, visualisation functions, optional GUI-driven setup functionality, and builds upon the Matlab<sup>®</sup> programming language. The presented examples represent single- and multiobjective optimizations for which different solvers were used, a derivative-free search algorithm (PSEARCH) for the first and a population based evolutionary algorithm (NSGA2) for the latter case. The evolutionary or genetic algorithm is able to generate a solution population which can be Pareto optimal. Pareto optimality for multiobjective optimization problems is characterized by the fact that for a set of Pareto optimal solutions one criterion value cannot be decreased (minimized) without increasing one or more other criteria.

## II.A. Parametrization

The parametrization is done via B-Splines where optimizer parameters correspond to the control points of the respective splines. A B-Spline is defined as a piecewise polynomial of degree  $p$

$$C(u) = \sum_{i=0}^n N_{i,p}(u) \cdot \mathbf{P}_i \quad (5)$$

with parameter  $u$ , control points  $\mathbf{P}_i$ , Basis-functions  $N_{i,p}$  and the spline at the current parameter value  $C(u)$ . The basis functions can be iteratively calculated by e.g. the Cox-De-Boor Algorithm<sup>4,5</sup> and are defined as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } k_i \leq u < k_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$N_{i,p}(u) = \frac{u - k_i}{k_{i+p} - k_i} N_{i,p-1}(u) + \frac{k_{i+p+1} - u}{k_{i+p+1} - k_{i+1}} N_{i+1,p-1}(u) \quad (7)$$

where  $k_i$  are the so-called spline-knots which characterize the actual parameterization of the spline. A record containing the waypoint positions in ellipsoidal coordinates needs to be generated (this is either done by the user or a flight planning tool), and the desired or required airspeeds have to be given at every waypoint as well. In the practical implementation the optimization parameters or tuners are represented as the respective spline's control points.

Currently four variables can be tuners for the optimization and are stored in the record (At the moment the configuration changes of the high-lift system of the aircraft are not included as control inputs to avoid integer valued tuners. This will be addressed in the next version of the setup):

- Altitude of the waypoints
- Prescribed  $V_{TAS}$  at the waypoints
- Allowed lateral deviation, this is defined as a translation deviation to the left or right of the trajectory at the waypoint. The maximum deviation points in the direction of the bisecting line of the two legs intersecting at the waypoint
- Time when the undercarriage is lowered or raised (e.g. two binary tuners for a city to city trajectory). This is mainly important for optimizations regarding noise emission with specialized tools (e.g.<sup>6</sup>).

## II.B. Criteria and constraints

Several scenario specific criteria are available for calculation, the total flighttime  $t_f$ , overall fuel consumption  $m_{F,f}$ , mass of emitted climate gases like carbon-monooxides (CO), nitric-oxides (NO<sub>x</sub>), hydrocarbons (HC) and soot, as well as noise related variables, like area enclosed in a certain dB-level contour and different noise metrics for measuring stations at the airports.

The masses of the climate gases are computed from the emission indices (EI) which are defined as mass of pollutant in kg per kg consumed fuel

$$m_{\text{emission}} = \int_0^{t_f} \dot{m}_F(t) \cdot \text{EI}_{\text{emission}}(t) dt \quad (8)$$

The emission indexes are determined by interpolation in a dataset that generates continuous results similar to those of the Advanced Emission Model (AEM) and the Boeing Fuel Flow Method 2.<sup>7,8</sup> The total masses of the respective gases emitted during the flight are calculated by equation (8) and can be set as optimization criteria.

Noise emission calculation is based on the model described in the ECAC Doc.29 Report.<sup>9</sup> One difference in the implementation is the direct integration of the sound exposure levels (SEL) in parallel to the integration of the aircraft model equations, whereas in the report a sum approximation is defined. In short, the noise-power-distance-tables of the Doc.29 model assume an infinitely long flight path of an aircraft to calculate the first value for the sound pressure level  $L_{E\infty}(T, d)$ . Then several correction terms are introduced, e.g. for the finite length of a segment  $\Delta_F$ , for lateral attenuation  $\Lambda(\beta, l)$ , installation of the engines on the aircraft  $\Delta_I(\phi)$  and duration correction  $\Delta_V$ . The resulting equation for corrected SEL depending on thrust  $T$ , slant distance  $d$ , distance on ground  $l$ , observer depression-angle  $\phi$  and elevation angle  $\beta$  is

$$L_{E,\text{seg}} = L_{E\infty}(T, d) + \Delta_V + \Delta_I(\phi) - \Lambda(\beta, l) + \Delta_F \quad (9)$$

and the total SEL for one observer is calculated as

$$L_E = 10 \cdot \log \left( \sum_{\text{seg}=1}^{n_{\text{seg}}} 10^{L_{E,\text{seg}}/10} \right) \quad (10)$$

Since one finite segment is just an approximation to a continuous process, the last equation can also be defined as the mentioned integral expression

$$L_E = 10 \cdot \log \left( \int_{t=0}^{t_f} 10^{L_{E,\text{seg}}/10} dt \right) \quad (11)$$

These sound exposure levels then form either individual criteria for single observer positions for example to model noise enforcement points at airports. Another option is to create an observer grid, on which contour lines of sound exposure levels can be determined. The area enclosed in these contours may then serve as a scalar minimization criterion.

In the definition of constraints, an explicit path constraint  $r_{\text{path}}$ , and additional acceleration and rate constraints were omitted<sup>a</sup>. As well, there is no time of arrival (TA) constraint for the final waypoint, but the parameterization module (see III.A) exactly calculates the complete time vector for the trajectory, therefore TA constraints could easily be introduced as equality constraints to the problem. The simulation is terminated at this final time value and the distance between the last position of the aircraft and the final waypoint is output as a possible performance measure.

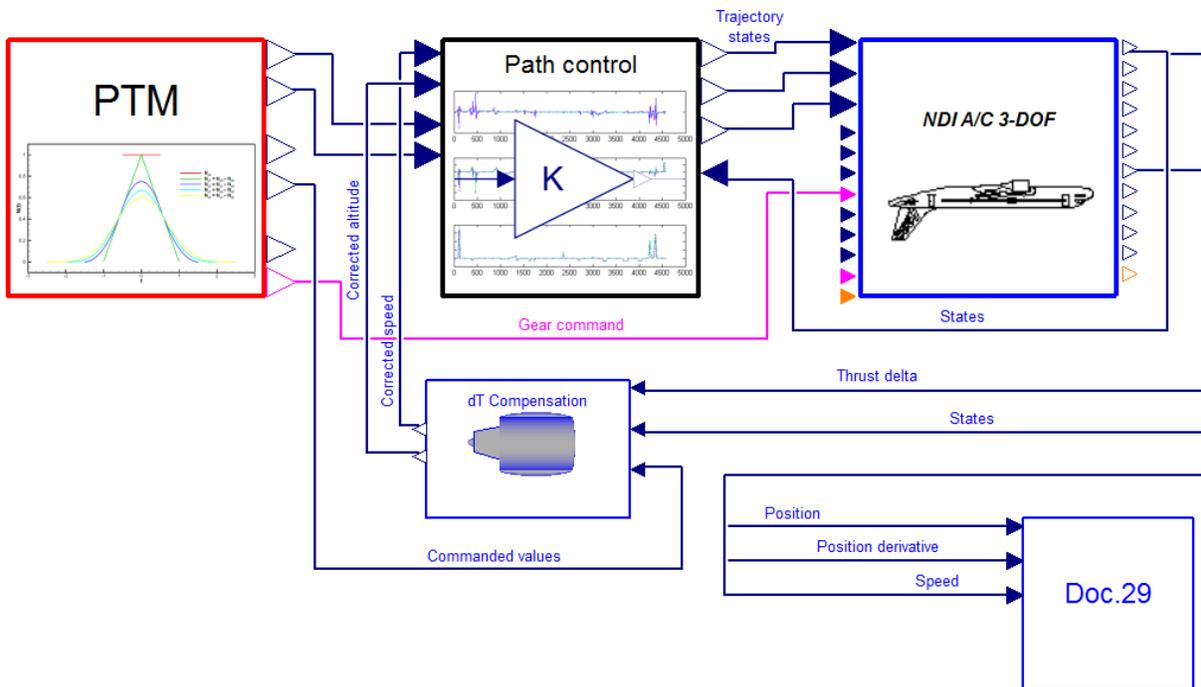
### III. Models

The so-called wrapper model works as an integrated model to be used in the optimization, and links the four sub-models parameterization module (PTM), path controller, inverse aircraft model and the newly developed PCH-thrust correction module. These are described in the following sections. During the optimization process, the wrapper obtains the optimizer parameters through a trajectory definition record, which is described in section II.A. Out of these values the parameterization module generates time dependent input functions that serve as commanded inputs for the PCH-module (sec.III.D) and trajectory controller (section III.B). These two models finally generate commanded inputs to the inverse aircraft model (sec.III.C) whose outputs are fed back to the controller and PCH in order to cancel out possible numerical errors during integration and to ensure flyability. The principal layout of the wrapper model can be seen in figure 2.

#### III.A. Parameterization module

The parameterization module uses the B-spline definition explained in section II to generate the commanded input values for the inverse aircraft, which are the trajectory state's derivatives  $\left[ \dot{V}_{TAS}, \dot{\gamma}, \dot{\chi} \right]$ . Due to the provision of the derivatives, some drift occurs in the states during integration of the model. This is cancelled by the trajectory following controller that is described in the next section.

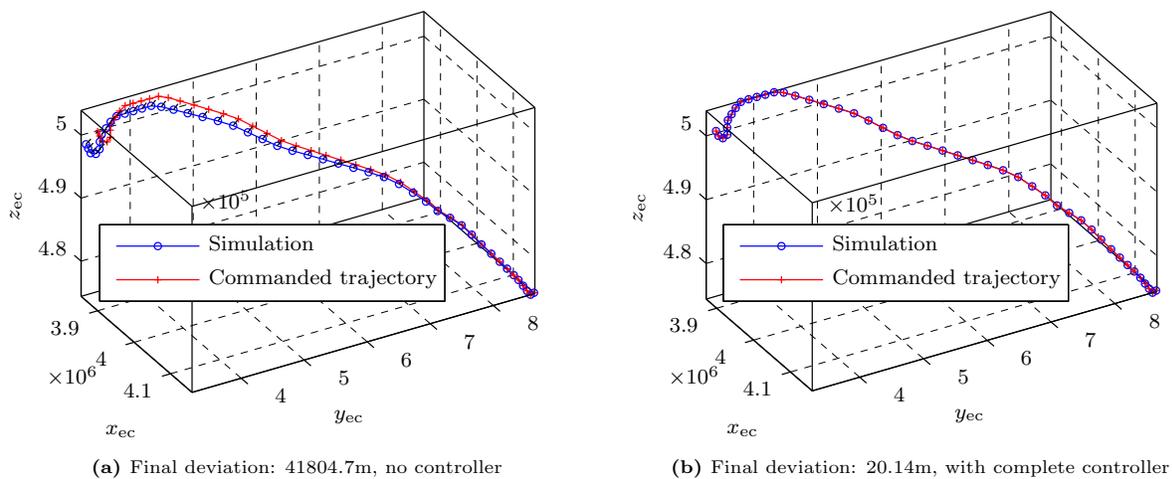
<sup>a</sup>An exception is the model without the PCH-module used for the comparison in section IV.B.



**Figure 2:** Wrapper model consisting of parameterization module (PTM), path controller (Path control), aircraft model (NDI A/C 3-DOF), noise model (Doc.29) and PCH module (dT Compensation)

### III.B. Trajectory following controller

In order to cancel out possible deviations in the flown trajectory originating from numerical error or coarse discretization, a 4-D trajectory following controller has been integrated in the global wrapper model. In figure 3 two simulation runs of a flight between two cities (Munich and Amsterdam) with and without the path controller are shown.



**Figure 3:** Trajectory deviation resulting from numerical error, in ECEF coordinates

The commanded values are calculated at every timestep using the parameterization module, and the flown trajectory values are fed back from the aircraft model described in section III.C. In detail, the aircraft's



are divided in the kinematic differential equations

$$\dot{x} = V \cdot \cos \gamma \cdot \cos \chi, \quad (14)$$

$$\dot{y} = V \cdot \cos \gamma \cdot \sin \chi, \quad (15)$$

$$\dot{h} = V \cdot \sin \gamma, \quad (16)$$

and the force equations

$$\dot{V} = (T - D)/m - g \cdot \sin \gamma, \quad (17)$$

$$\dot{\gamma} = (L \cdot \cos \mu)/(m \cdot V) - (g \cdot \cos \gamma)/V, \quad (18)$$

$$\dot{\chi} = (L \cdot \sin \mu)/(m \cdot V \cdot \cos \gamma). \quad (19)$$

Along with an aerodynamic and engine model, and taking into account the local air density  $\rho$ , the corresponding thrust, lift and drag are given by

$$T = f(\delta_T, V, h), \quad L = \bar{q} \cdot S \cdot c_A, \quad D = \bar{q} \cdot S \cdot c_W, \quad \bar{q} = \frac{\rho}{2} V^2, \quad (20)$$

where  $c_A$ ,  $c_W$  are the lift- and drag coefficients respectively. The model is then re-cast using model-inversion, which is also known from feedback-linearisation, nonlinear dynamic inversion and differential flatness. These theoretical approaches have been subject of multiple studies and are also described in various publications and textbooks, for a detailed description see e.g. the book<sup>10</sup> and the initial differential flatness paper.<sup>11</sup>

In short, model inversion is based upon the principle that a nonlinear dynamic system

$$\dot{x} = f(x) + g(x)u, \quad (21)$$

with the output function

$$y = h(x), \quad (22)$$

is invertible, if the states  $x$  and the controls  $u$  can be expressed by the outputs  $y$  and a finite number of their derivatives  $y^{(\beta)}$ ,  $\beta \in \mathbb{N}$ .

$$x = \Psi_1(y, \dot{y}, \dots, y^{(\beta)}) \quad (23)$$

$$u = \Psi_2(y, \dot{y}, \dots, y^{(\beta+1)}) \quad (24)$$

Another requirement is that the number of outputs equals the number of inputs ( $\dim y = \dim u$ ), so that the outputs are differentially independent, or they do not fulfil differential equations of the following form

$$\Phi(y, \dot{y}, \dots, y^{(\beta)}) = 0. \quad (25)$$

In aircraft applications, there are often situations where  $\dim y \neq \dim u$ , e.g. if there are more control surfaces than outputs. In these cases a so-called control allocation algorithm can be employed to render the model invertible.<sup>10</sup>

The practical approach to invert the model is to repeatedly differentiate the output function until the control  $u$  appears explicitly.

This is done by means of Lie-Derivatives which represent the gradient of the output function with respect to the system's trajectory  $x(t)$

$$\dot{y} = \frac{dy}{dt} = \frac{\partial h(x)}{\partial x} \frac{dx}{dt} = \nabla h(x) \dot{x} = L_f h(x). \quad (26)$$

The amount of differentiations necessary is denoted by the relative degree  $r$ , the outputs' corresponding derivative is

$$y^{(r)} = L_f h(x) + L_g L_f^{(r-1)} h(x) u = \nu, \quad (27)$$

which is the so called pseudo-control  $\nu$ . By using a nonlinear transformation function dependent of  $x$  and  $\nu$  the original control  $u$  can be stated as

$$u = \frac{1}{L_g L_f^{(r-1)} h(x)} (\nu - L_f^{(r)} h(x)). \quad (28)$$

The procedure of generating inverse models is highly automated in the Modelica<sup>®</sup>-modeling language<sup>12</sup> and the Dymola<sup>®</sup>-tool, which makes it easier to derive inverted versions from existing models. The equation-based modelling in Modelica is also beneficial to this process since users can define their models based on equations and two-directional connectors. The object-oriented and modular capabilities finally help the user to rapidly change certain parts of the model and to quickly build up new models from scratch. An example for an integrated model for a solar airplane developed in this language is given in reference.<sup>13</sup>

### III.D. Thrust compensation

The last block in the wrapper model forms the novelty of the described approach for trajectory optimization and picks up the idea to constrain the control  $\nu$  by the pseudo-control-hedging methodology. Pseudo-control-hedging is due to Johnson and Calise<sup>14</sup> and has been successfully applied in the fields of nonlinear and adaptive control with nonlinear dynamic inversion controllers.<sup>15</sup> It has also been part of control laws for real flight tests as described in the paper.<sup>16</sup>

As stated in equation (27), the relative degree determines the number of differentiations necessary to obtain an equation which includes the former control inputs, in the presented case the throttle input  $\delta_T$ . In the inversion process, saturation blocks cannot be handled since they vanish when linearizing the system. Actuator dynamics which are commonly modeled as first order lags introduce another state integrator into the system. When trying to invert these dynamics as well, the relative degree of the inversion also has to increase. In the presented example, the derivative of acceleration (jerk) would need to be provided by the parameterization module, which in turn causes additional numerical drift in the integration (see section III.B and figure 3) and possibly oscillations in the controls. Furthermore, the spline degree would need to be increased as well as the number of control points in order to model the trajectory as a sequence of straight flight legs with interconnecting arcs.

The engine model interpolates thrust tables dependent on Mach number, throttle setting and altitude. Furthermore the values are extrapolated if the inputs are out of bounds and a variable for limit thrust  $T_{\text{lim}}$  is output next to the demand thrust  $T_c$ . This difference is taken as input to the module in order to calculate new values for commanded speed  $V_c$  and altitude  $h_c$ :

$$\Delta T = T_c - T_{\text{lim}}. \quad (29)$$

The force equation in the aircraft's longitudinal direction reads

$$m\ddot{x} = T + L \sin \alpha - D - G \sin \Theta. \quad (30)$$

Assuming level flight with lift equal to weight, it reduces to

$$m\ddot{x} = T - D - G \sin \gamma = (T_0 + \Delta T) - (D_0 + \Delta D) - G(\gamma_0 + \Delta \gamma). \quad (31)$$

Linearization about  $(T_0, D_0, \gamma_0)$  yields

$$m\ddot{x} = \Delta T - \Delta D - G\Delta \gamma. \quad (32)$$

Assuming that the drag is not changing, this equation represents the influence of velocity and pitch angle or altitude to thrust in level, trimmed flight.

$$\Delta T = m\ddot{x} + G\Delta \gamma = mk_v(V_c - V) + Gk_\gamma(\gamma_c - \gamma) \quad (33)$$

It can finally be separated in two new reference signals, which form the new inputs for the path controller (see figure 2). The modified velocity equals

$$V_{\text{lim}} = -\frac{\Delta T}{m \cdot k_v} \cdot a_v + V_c, \quad (34)$$

and the new altitude is given by

$$h_{\text{lim}} = -\frac{\Delta T \cdot V_c}{m \cdot g \cdot k_h} \cdot a_h + h_c, \quad (35)$$

with the weighting factors  $a_v$  and  $a_h$  specifying the fraction of the thrust difference  $\Delta T$  each channel has to compensate, where  $a_v + a_h = 1$ . These factors can for example be used to model a speed priority ( $a_v = 0$ ,  $a_h = 1$ ) or a pitch angle / altitude priority ( $a_v = 1$ ,  $a_h = 0$ ) during capture of the reference trajectory. The corrected  $V$  and  $h$  commands for a complete trajectory between two cities with a 50 % distribution on the weighting can be seen in figure 5.

## IV. Results

In order to test and validate the combined PCH and inverse modeling approach, several simulations and optimizations were conducted. In the first example, two simulations were performed, one with activated PCH and one without, and the results were compared with the focus on the change of the reference inputs. The second example attempts to examine the influence of the pseudo-control-hedging on the result of a single-objective optimization. The result is compared to a non-PCH optimization, which ensures flyability with a number of inequality constraints. In order to show the benefits of the presented approach especially for evolutionary optimization methods, which are not very well suited for (in-)equality constraints, the last example is a multiobjective trajectory optimization with six criteria that are evaluated with respect to Pareto optimality. All results have been obtained with the following configuration:

- Intel Core 2 Duo Workstation 3.0 GHz, 4 GB RAM, Windows 7 enterprise 64bit operating system
- MOPS version 6.2, Matlab 2011b, Dymola<sup>®</sup>2012 FD01, Modelica 3.2, FlightDynLib 4.0
- DAE-solver Dassl for model integration (relative tolerance  $1e^{-4}$ ), NSGA2 algorithm for multiobjective optimization, Pattern-Search algorithm for single-objective optimization

### IV.A. Comparison inverse model with and without thrust compensation

In order to show the practical relevance of the explained approach, a simulated standard flight between two cities (Munich to Amsterdam)<sup>b</sup> serves as a test case to show the differences between a non-modified inverse model formulation and the addition of the thrust correction described in section III.D. The flight is defined as a set of waypoints with given altitudes and true airspeed values at the waypoints. Out of these, the parameterization module generates a smooth spline trajectory which returns a set of  $[\varphi \ \lambda \ h \ V_{TAS}]$  values at each timestep  $t$ . The computed thrust demanded by the inverse formulation, the limit thrust of one engine and the corrected thrust value are shown in figure 5a. The differences in the time series originate from certain parts of the trajectory definition where high demands are posed upon the aircraft with respect to acceleration / deceleration and climb- / descent- rates. Figure 5 was obtained when using the thrust correction, that keeps the calculated thrust on or in the close vicinity of the limit thrust.

In figure 5b and 5c the altered reference signals coming out of the thrust module are shown. As was expected, the module reduces the commanded speed and altitude when the thrust demand is too high for the engine. In the case of demand thrust below idle thrust, the module increases the commanded speed and the climb rate / commanded altitude. Deviations in the trajectory are however minor and it can be observed, that due to the ATD-tracking and the 3D tracking of the flightpath performed by the trajectory controller, the aircraft still arrives at the final waypoint at the precomputed time.

A good example is the time interval from  $t \approx 3100s$  to  $t \approx 3300s$ , when the commanded thrust becomes negative to satisfy the high descent rate. It then rises again quickly to follow the trajectory. The corrected thrust on the other hand is reduced to idle thrust, and forces the aircraft to fly faster and higher. It therefore captures the precomputed trajectory later than the inverse formulation, but does not violate any limits and has a smoother shape.

### IV.B. Optimality comparison

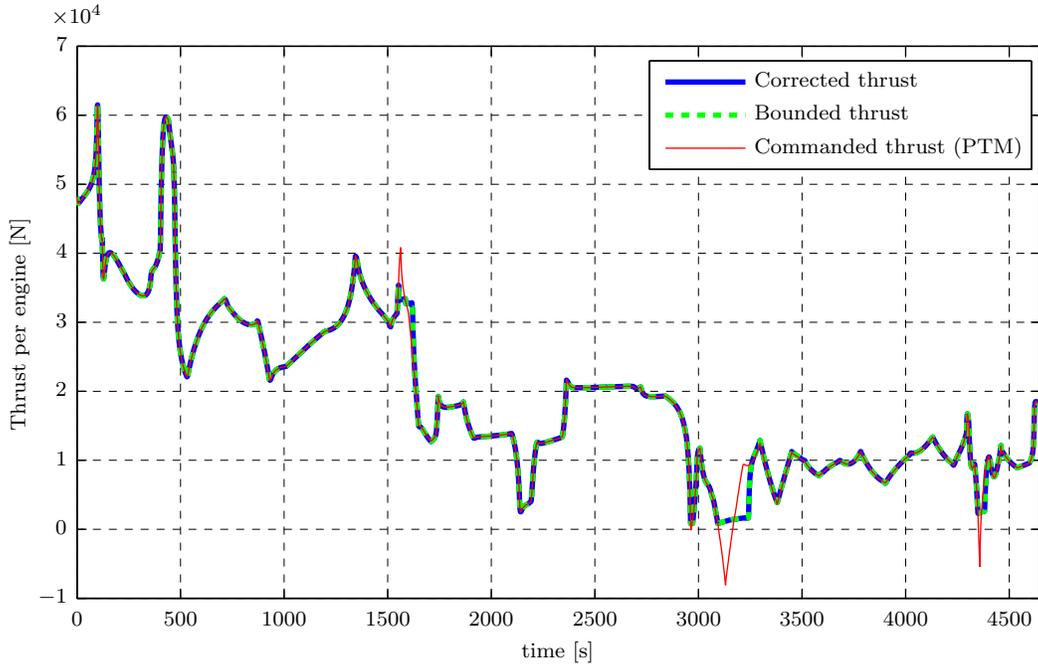
As was shown in the previous sections, the thrust correction module can directly modify the vertical reference trajectory that is given either by the user or the optimization from outside of the model. In the latter case, it is obvious that the PCH-module alters the behaviour of the optimization model when there is a difference between achievable and commanded thrust. In the unfavorable case of the optimization generating a reference trajectory that causes the PCH-module to constantly alter the reference trajectory so that it is still flyable, the optimizer could get wrong results with respect to optimality of this trajectory.

In a first step to examine the influence of the PCH to the quality of the optimized trajectories, two single objective optimizations were performed, one with and one without the module. The flyability of the non-PCH model was established using a path constraint for the thrust deviation

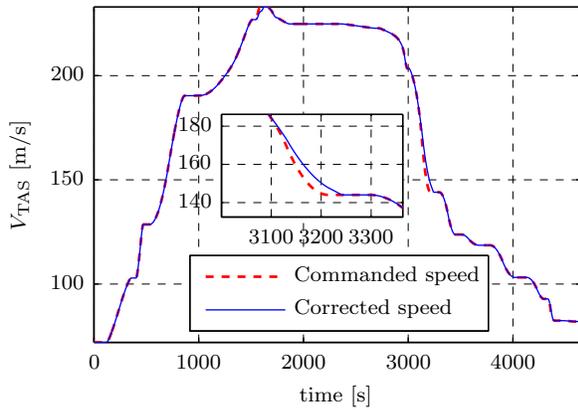
$$r_{\text{Thrust}}(x, p, t) = T_{\text{ref}} - T_{\text{bounded}} \leq 1 \quad (36)$$

---

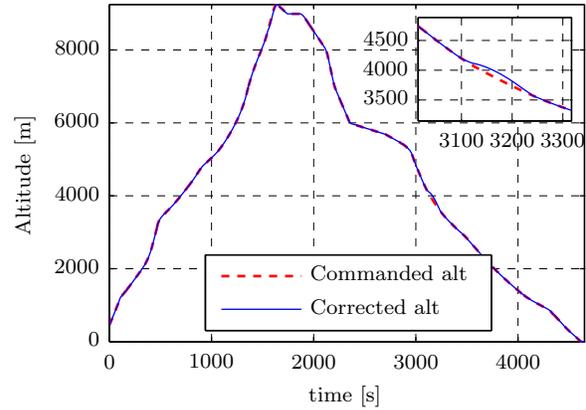
<sup>b</sup>The trajectory in ECEF-coordinates can be viewed in figure 3.



(a) Thrust comparison for simulations with and without PCH



(b) Corrected speed commands



(c) Corrected altitude commands

**Figure 5:** Differences between commanded values from the parameterization module and PCH-corrected values

sampled across the optimization time horizon resulting in a total of 360 inequality constraints. Since the baseline trajectory has a total flight duration of roughly 1800 s, the path constraints are enforced at about every 5 s (this interval is also reducing during the optimization since  $t_f$  is minimized). The flight scenario is the RENLO-4M standard terminal arrival route (STAR) to Munich Airport with altitudes and speeds at five waypoints as optimization variables and total flight time as criterion for the optimization<sup>c</sup>. The standard values and allowed deviations are shown in table 1 and the two optimal trajectories are shown in figure 6, along with their thrust time series.

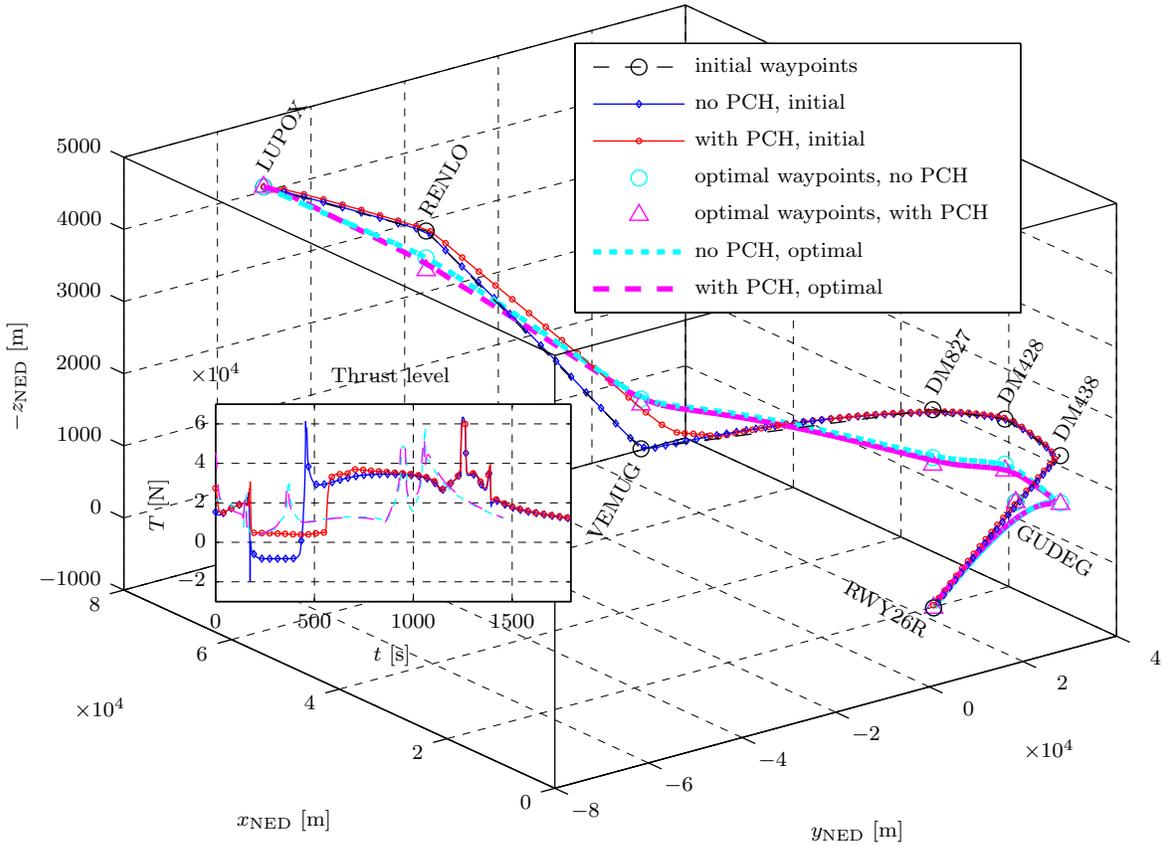
As expected, the non-PCH model experiences a negative thrust command which is provoked by requesting a fast descent and high airspeed in the baseline trajectory<sup>d</sup>. This can be seen when comparing the two initial

<sup>c</sup>Note that the lateral deviation tuners listed in table 1 are not considered for this setup.

<sup>d</sup>Note that due to the transformation from WGS (ellipsoidal) to NED (flat-earth)-coordinates, the altitudes  $h_{NED}$  do not coincide with the altitude values in table 1.

**Table 1:** Waypoint data for RENLO 4M arrival to Munich airport

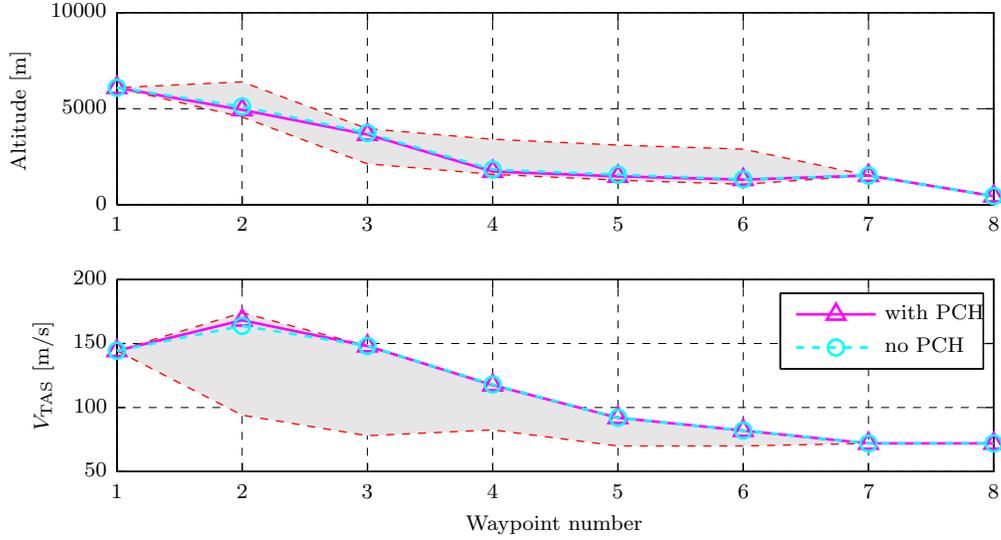
| Waypoint  | Speed      | Min-/Max            | Altitude  | Min-/Max          | Lateral deviation | Min-/Max |
|-----------|------------|---------------------|-----------|-------------------|-------------------|----------|
| LUPOX     | 144.33 m/s | -/-                 | 6095.00 m | -/-               | 0 m               | -/-      |
| RENLO     | 124.04 m/s | 94.04/174.04 m/s    | 5486.40 m | 4572.00/6400.80 m | 0 m               | ±9270 m  |
| VEMUG     | 108.03 m/s | 78.03/148.03 m/s    | 3047.00 m | 2133.60/3962.40 m | 0 m               | ±6490 m  |
| DM827     | 87.56 m/s  | 82.56/117.56 m/s    | 2499.35 m | 1585.00/3413.8 m  | 0 m               | ±3710 m  |
| DM428     | 72.02 m/s  | 70.02/92.02 m/s     | 2194.56 m | 1280.20/3109.00 m | 0 m               | ±1850 m  |
| DM438     | 72.02 m/s  | 70.0222/82.0222 m/s | 1981.20 m | 1066.80/2895.60 m | 0 m               | -/-      |
| GUDEG FAF | 72.02 m/s  | -/-                 | 1524.00 m | -/-               | 0 m               | -/-      |
| RWY26R    | 72.02 m/s  | -/-                 | 453.24 m  | -/-               | 0 m               | -/-      |



**Figure 6:** Initial and optimized trajectories for flight time minimization problem, relatively large differences are present in the altitude and thrust(speed) after the optimization.

trajectories (red and blue) and their corresponding thrust time series: while the non-PCH version follows the waypoint polygon (black) closely, the PCH version is altered (shallower descent and larger turn radius) to ensure flyability and to avoid negative thrust. The larger turn radius is due to the altitude- and speed-dependent roll angle limiter implemented in the path controller (see III.B).

In the non-PCH version, the optimizer has to fulfill the inequality constraints which are active in the descent phase. In the final solution, the negative thrust is eliminated by satisfying the constraints, and the two optimizations yield very similar results with a minimal flight time of 1453.4s for the PCH model and 1458s



**Figure 7:** Final tuner values for the (non-) PCH optimizations

for the non-PCH version. Also when looking at the optimal tuner values and the two resulting trajectories, no larger differences are present (see figures 6, 7). This result indicates that the PCH-controller has only a minor effect on the optimizer when determining the optimal solution for this problem. However, since this is only a single test which may not be representative, more extensive studies have to be performed to make a profound statement on the influence of the PCH-module on the optimization.

#### IV.C. Multiobjective optimization

In this section a multiobjective optimization case is considered, with the approach defined in section IV.B and using the PCH wrapper model. Again, altitudes and  $V_{TAS}$ -values are the optimization variables, with the addition of the lateral deviations at the waypoints which are listed in table 1 (see section II.A for a description on the three tuner types). Since the influence of all environmentally related criteria is of interest in this case, a Pareto optimization using the NSGA2<sup>17</sup> optimizer was set up, taking into account the criteria fuel consumption ( $m_{F,f}$ ), flight time ( $t_f$ ), masses of carbon-monoxide, hydrocarbons, nitric oxides ( $m_{CO}, m_{HC}, m_{NO_x}$ ) and area enclosed in the 50dB noise contour ( $a_{50dB}$ ). Several optimization parameters and results are listed in table 2.

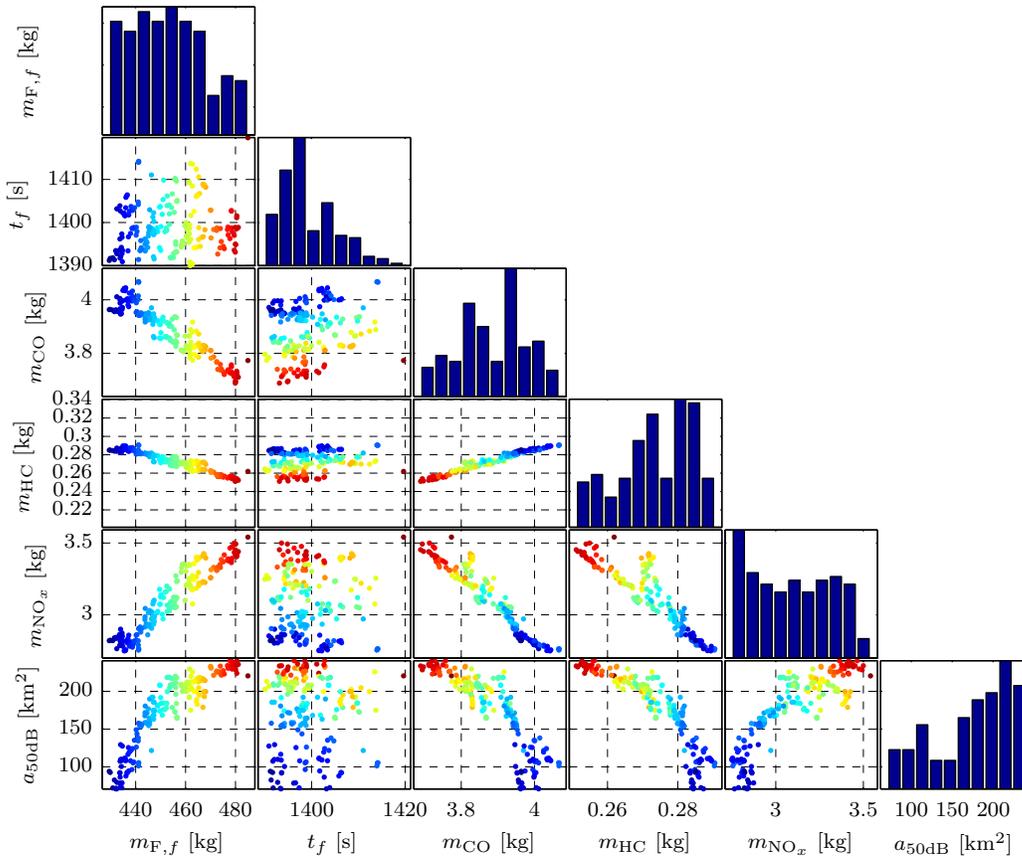
**Table 2:** Optimization parameters and results

|                               |       |                                     |           |
|-------------------------------|-------|-------------------------------------|-----------|
| Algorithm                     | NSGA2 | Population size                     | 32        |
| Generations                   | 100   | Maximum function evaluations        | 5000      |
| Average model evaluation time | 40 s  | Total optimization time (single PC) | 35h 45min |

In figure 8 the points contributing to the respective finally reached Pareto front approximation after 100 generations are shown. The coloring is defined by the solution range of the first active criterion, in this case the fuel consumption  $m_{F,f}$ . In this sense, all red points refer to the solutions with high fuel consumption, all blue points refer to those with low fuel consumption and so on.

Interestingly, the flight time does not clearly oppose the fuel consumption here, which might be due to the rather strict constraints of the problem, and/or the trajectory is too short to obtain large differences in flight time. There is also no evident conflict in the noise,  $NO_x$ , HC and CO emissions w.r.t. flighttime. This indicates that all of these criteria are minimized when the flight time is minimized as well, and is confirmed when inspecting the Pareto fronts progress plots from generation 1 to 100 (not shown here).

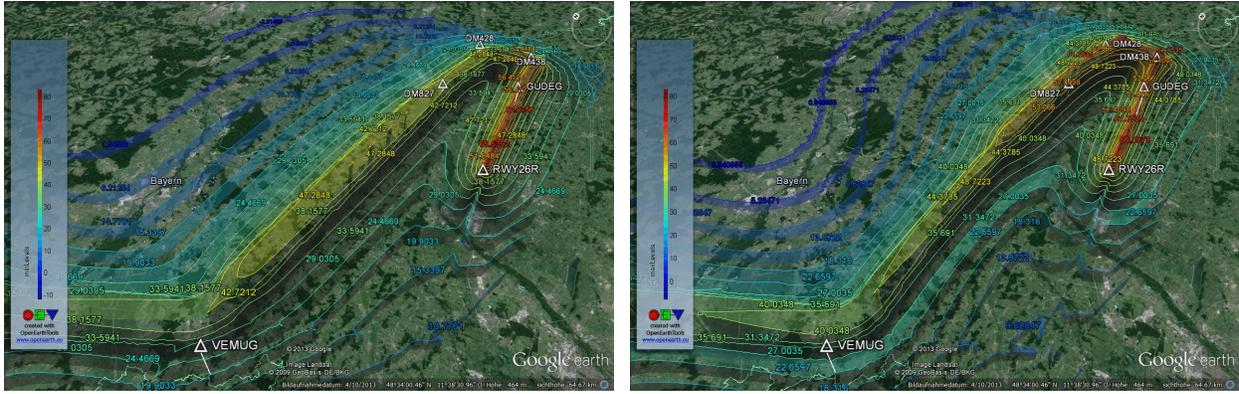
Other criteria however show a behaviour which is in line with physical expectations, e.g.  $m_{CO}$  vs.  $m_{NO_x}$ , where  $NO_x$  is mainly produced at high temperatures in the combustion chamber (high speeds causes high



**Figure 8:** Pareto front approximations for the six-objective optimization, the plots on the diagonal axis represent the histograms for the criterion values in the last generation

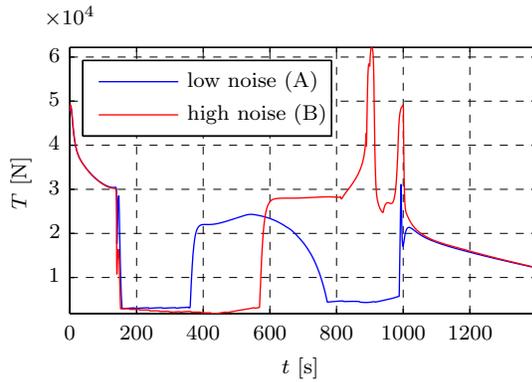
fuel consumption, which relates to the red points) and CO at lower temperatures when the combustion is incomplete and carbon-based pollutants are emitted (lower speeds causes low fuel consumption, which is related to the blue points). This is also confirmed by the concurrent Pareto front of carbonmonoxides and hydrocarbons which also precipitate stronger in low temperature combustion. When looking at the values for the 50 dB noise area, the noise pollution seems to increase with higher and decrease with lower fuel consumption. This is also evident since high fuel consumption implies higher airspeeds and therefore a higher noise emission from the engines.

Two Pareto optimal solutions for low (A) and high noise (B) are shown in figure 9a and 9b. The according thrust, altitude and velocity plots (figures 9c, 9d) indicate that (B) descends longer and has idle thrust in the beginning of the trajectory where the aircraft is still far away from the observers. Furthermore at  $t \approx 900$  s a large thrust increase is commanded which is due to the climb initiated at this time. The second thrust surge is common to both solutions and can be explained by the last turn at waypoint DM438. In contrast to (B), solution (A) keeps the distance to the noise observers as high as possible by enlarging the altitude. The thrust is decreased to idle thrust when the aircraft is closer to the ground and by using the potential energy from the higher altitude, the aircraft can pass the final two turns with fewer thrust corrections. The resulting difference between the 50dB noise areas is as high as  $a_{50dB,high} - a_{50dB,low} = (230.77 - 157.45) \text{ km}^2 = 73.32 \text{ km}^2$ . With the exception of the high computational effort necessary for this approach, this example shows the advantage of Pareto optimization that enables the user to examine and choose from a large set of potentially very diverse solutions, depending on his criteria focus.

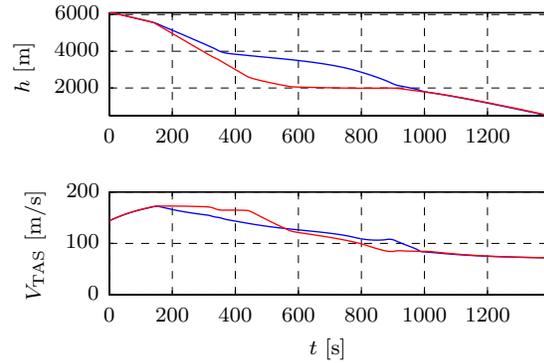


(a) Low noise trajectory (A)

(b) High noise trajectory (B)



(c) Thrust per engine



(d) Altitude and velocity

**Figure 9:** Two Pareto optimal trajectories with low- and high noise generation (noise contour plots generated with Google Earth<sup>®</sup> and the visualization and conversion routines<sup>18, 19</sup>)

## V. Conclusion

The presented method enables the consideration of thrust limits in a dynamic inversion model by feeding back the deviation between commanded and limit thrust to the reference model. This approach was inspired by the method of pseudo control hedging, and adapted for usage in a trajectory optimization setup. It was shown that the trajectory following capability of the original setup is not compromised by the new approach but automatically avoids the generation of unflyable trajectories during runtime. The controller was tested in a scenario of a passenger aircraft flying on a standard segmented trajectory from city to city and showed the ability to handle out-of-limit thrust demands efficiently. The effects of the PCH-approach on the optimization result was examined in a first comparison where path constraints were used to ensure flyability. Furthermore, a multiobjective trajectory optimization setup depicting the airport approach phase under the consideration of environmental related criteria was investigated.

When looking at the results, it can be stated that the 4-D trajectory tracking using an inverse model is possible using this approach, if the predefined flightplan is not inherently infeasible (e.g. demanding too high  $V_{TAS}$  values which causes a reaction in the controller, so that the final waypoint is impossible to reach in time). The approach also showed its practicality for multiobjective optimization with stochastic solvers, in which extreme parameter combinations can occur. Despite the shortcomings like long computation times, the user gains more insight into the problem with this algorithms, since Pareto front plots can directly show the mutual dependences of the different criteria, and give quantitative estimations on the benefits as well as a large set of best compromise solutions to choose from.

## VI. Outlook

Further research will be directed towards investigation of other bounded variables apart from thrust, for example the roll angle, and possibly higher dynamic limits for use in the PCH-module and the inverse model. For the trajectory optimization, the influence of the PCH on optimization results will be investigated in a more detailed and larger scale, to create better assurance in this regard. As well, handling of integer-valued tuners will be implemented in the optimization model to allow for the consideration of aircraft configuration changes.

## References

- <sup>1</sup>EUROCONTROL, “User manual for the base of aircraft data (BADA) revision 3.10,” Technical report, EUROCONTROL, 2005.
- <sup>2</sup>Mueller, R., “Multi-Objective Optimization of an Aircraft Trajectory between Cities using an Inverse Model Approach,” *AIAA Modeling and Simulation Technologies Conference, Minneapolis, MN*, 2012.
- <sup>3</sup>Joos, H.-D., Bals, J., Looye, G., Schnepfer, K., and Varga, A., “A multi-objective optimisation based software environment for control systems design,” *Proc. of 2002 IEEE International Conference on Control Applications and International Symposium on Computer Aided Control Systems Design*, IEEE, Glasgow, 2002, pp. 161–176.
- <sup>4</sup>De Boor, C. R., “A practical guide to splines,” *Springer-Verlag New York, Inc.*, 1978.
- <sup>5</sup>Piegl, L. and Tiller, W., “The NURBS book,” *Springer-Verlag New York, Inc.*, 1997.
- <sup>6</sup>Bertsch, L., Guerin, S., and Pott-Polenske, M., “The Parametric Aircraft Noise Analysis Module - status overview and recent applications,” *AIAA/CEAS Aeroacoustics Conference, Portland, OR*, 2011.
- <sup>7</sup>“Advanced Emission Model (AEM3), V.1.5: Validation Report,” Tech. rep.
- <sup>8</sup>DuBois, D. and Paynter, G., “Fuel Flow Method2 for Estimating Aircraft Emissions,” Technical report, Society of Automotive Engineers (SAE), 2006.
- <sup>9</sup>“ECAC.CEAC Doc 29 3rd Edition, Vol.2: Technical Guide,” Tech. rep.
- <sup>10</sup>Isidori, A., “Nonlinear Control Systems Third Edition,” *Springer-Verlag London, Inc.*, 2001.
- <sup>11</sup>Fliess, M., Levine, J., Martin, P., and Rouchon, P., “Flatness and defect of nonlinear systems: Introductory theory and examples,” *International Journal of Control*, Vol. 61, No. 6, 1995, pp. 1327–1361.
- <sup>12</sup>“Modelica Language Specification Version 3.2,” Technical report, Modelica Association, 2010.
- <sup>13</sup>Klöckner, A., Leitner, M., Schlabe, D., and Looye, G., “Integrated Modelling of an Unmanned High-Altitude Solar-Powered Aircraft for Control Law Design Analysis,” *Advances in Aerospace Guidance Navigation and Control – Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control*, edited by Q. Chu, B. Mulder, D. Choukroun, E.-J. van Kampen, C. de Visser, and G. Looye, Springer Berlin Heidelberg, 2013, pp. 535–548.
- <sup>14</sup>Johnson, E. and Calise, A., “Pseudo-Control Hedging: A New Method For Adaptive Control,” *Advances in Navigation Guidance and Control Technology Workshop*, 2000.
- <sup>15</sup>Holzapfel, F., “Nichtlineare adaptive Regelung eines unbemannten Fluggeraetes,” *Dissertation Technische Universität München*, 2004.
- <sup>16</sup>Lombaerts, T. and Looye, G., “Design and Flight Testing of Nonlinear Flight Control Laws,” *to be submitted to AIAA Journal of Guidance, Control and Dynamics*, 2013.
- <sup>17</sup>Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, April 2002, pp. 182–197.
- <sup>18</sup>Deltares and Delft University of Technology, “OpenEarth Toolbox,” 2013.
- <sup>19</sup>De Oliveira, R. F., “KML Toolbox v2.6,” 2013.