

Clean Sky – AWACs

Final Report

Table of Contents

1	Executive Summary	2
2	Summary Description of Project Context and Objectives	3
3	Description of the main S&T results/foregrounds	5
3.1	Analysis of Optimisation Environment	5
3.1.1	Problem sparsity	6
3.1.2	Certification aspects	6
3.2	New solver interfaces	6
3.3	Conservative iteration mode	6
3.4	Robustness to erroneous inputs	7
3.5	Direct multiple shooting	7
3.6	Grid refinement	8
3.7	Structure exploitation	9
3.8	Multiobjective Optimisation	9
3.9	Testing	10
3.9.1	Goals of testing	10
3.9.2	Mathematical testing vs. aircraft related testing	10
3.9.3	Test Results	11
4	Potential impact	13
4.1	Technical impact	13
4.2	Strategic impact	13

1 Executive Summary

Achieving optimal performance of technical processes is increasingly difficult, as the number of parameters that influence their performance continues to grow, driven by overall technological progress. The high complexity typically prohibits manual optimisation and instead requires automatic optimisation procedures by mathematical software, such as the European NLP solver WORHP (We Optimize Really Huge Problems). It is specifically designed to solve large-scale nonlinear optimisation problems with several hundred thousands or even millions of optimisation parameters.

The purpose of the EU CleanSky project “AWACs” was to adapt WORHP to aviation objectives and constraints with the aim to increase its robustness and thus to enable new technologies in the overall field of aerospace engineering. Among others, the project tackled the following particular issues:

- Improve performance, i.e. have the solver produce optimal solutions in less computational time. This is relevant when the solver is run on resource-constrained hardware, such as the so-called Electronic Flight Bags (EFBs), which are essentially laptops or tablet computers used by pilots to plan their flight.
- Improve robustness, i.e. prevent breakdowns of the optimisation process, e.g. if the computer model of the aircraft suffers from technical problems in certain states that are evaluated by the solver.
- Ensure the solver is interruptible, so an application can stop it to work with a suboptimal solution, rather than waiting for an optimal one; this is of interest if time budgets are limited.
- Handle multiple objectives, so-called cost functions, which describe how “good” a particular solution is. Aircraft operation (like virtually every application) has various, partially competing objectives, such as fuel consumption, noise generation, overall flight time or generation of contrails. Being able to handle them in a comprehensive way can help aircraft operators achieve better solutions.
- Validate that the solver is able to solve realistic aviation problems, and document its performance.

The project achieved its core objectives: WORHP satisfies all functional and technical requirements and is able to solve 11 of the 12 evaluated aircraft trajectory optimisation problems, more than the other considered solvers Ipopt (9 of 12) and SNOPT (7 of 12). Performance testing demonstrated a reduction in required CPU time of 60%, and an overall success rate close to 96% for the standard optimisation test set “CUTEst”.

The work in this project was shared between a consortium of four partners: Steinbeis Innovationszentrum für Optimierung, Steuerung und Regelung at Universität Bremen, the Operations Research group at the University of Southampton, Steinbeis Innovationszentrum für Optimierung, Steuerung und Regelung at Universität der Bundeswehr München and the Institute of Flight System Dynamics at the Technische Universität München. The Topic Managers role with respect to this project covers statement of the problem, feedback on deliverables and integration of the results into CleanSky SGO activities.

2 Summary Description of Project Context and Objectives

The air transport industry is paying a lot of attention to growing public concern about the environmental issues of air pollution, noise and climate change. Although today air transport only produces 2% of man-made CO₂ emissions, this is expected to increase to 3% by 2050 with the continuous and steady growth of traffic. The Clean Sky project as a whole was set up to bring significant step changes regarding the environmental impact of aviation.

The AWACs activity is established in the “Systems for Green Operations” (SGO) section of the programme, which is aimed at operating existing hardware in a greener way. The AWACs output is supposed to be employed in the aircraft trajectory management, i.e. to steer the airplane such that overall fuel consumption and emissions (noise, CO₂, contrails) are reduced.

Since the actual control of an airplane is highly safety-critical, it is managed and controlled by an extensively tested and certified flight management system (FMS) and a flight control computer (FCS), being in charge of computing a continuous reference trajectory and ensuring that the aircraft follows it. This whole system is subject to strict regulation by national, European and worldwide air traffic authorities. However, as the FMS requires pre-flight input from the pilot, for instance on climb rates, airspeeds etc., a restricted form of trajectory optimisation is able to achieve improvements towards the project’s overall aims by helping the pilot choose these degrees of freedom in an optimised way.

The intended solution is to outfit any suitable computational platform, such as an Electronic Flight Bag (EFB), which is essentially a laptop or tablet computer used by pilots in the cockpit helping them to do their work, with software based on mathematical optimisation. This way the EFB will compute and suggest optimal choices for the pilot to feed into the FMS for each particular flight, potentially depending on weather conditions, airplane load, traffic and other considerations. As here the pilot always remains “at the controls”, the solutions created by that software of course need to be correct and trustworthy but do not have to be available in any case and within a fixed amount of time, leading to significantly lower regulatory and certification constraints.

Tools for numerical trajectory optimisation have existed for several decades, based on developments of numerical analysis researchers in applied mathematics, but only recent technological advances in the last one or two decades have enabled the solution of optimisation problems with millions of decision variables. These advances theoretically enabled tackling highly complex real-world problems, such as aircraft control, by methods of mathematical optimisation. The specific goal of the AWACs project is therefore to enhance the existing optimisation solver WORHP to meet the special demands of aircraft trajectory optimisation, and to validate these enhancements using high-fidelity aircraft models in realistic test scenarios.

To this end, following core development tasks were addressed:

- **Conservative iterations:** Since the WORHP computations are interruptible, it would be practical to ensure that each trial step of the solver is feasible, i.e. corresponds to an actually flyable, albeit suboptimal trajectory for the aircraft. Since it is impossible to guarantee this for arbitrary problems and models, the goal is to increase the ratio of feasible iterates.
- **Robustness to errors:** High-fidelity aircraft models are so complex that they may break down for unphysical input parameter settings, but optimisation solvers tend to evaluate models at

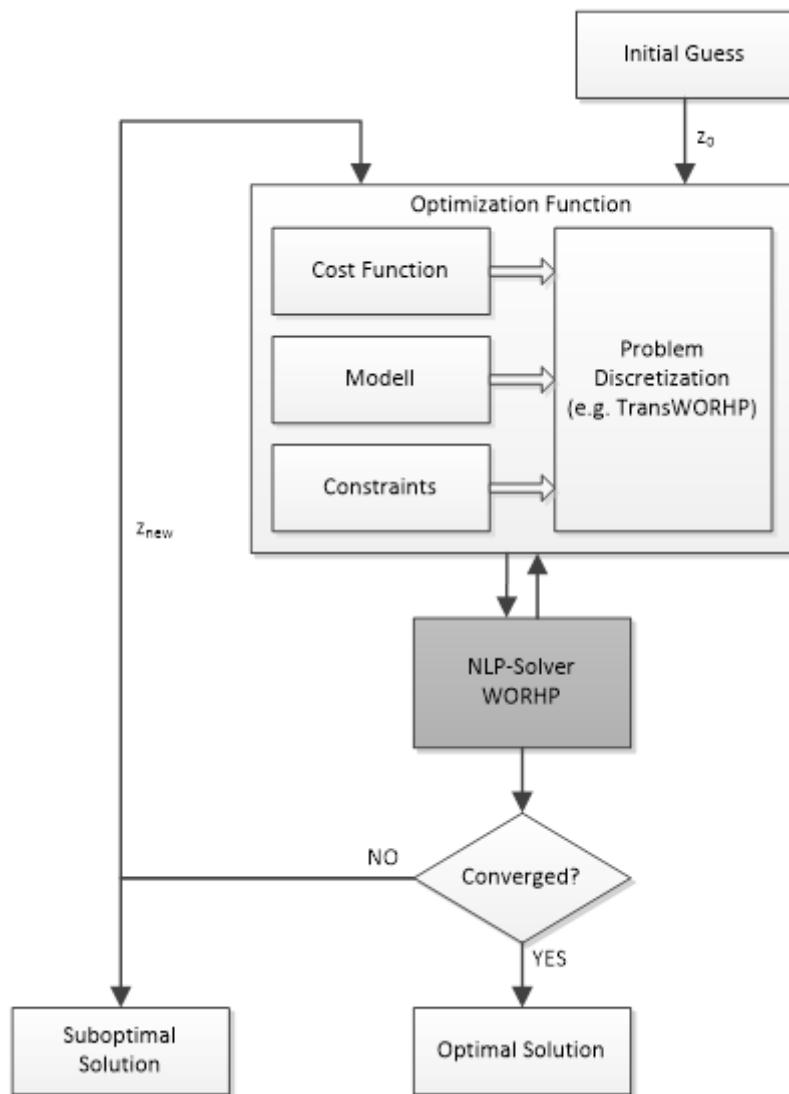
such unphysical trial points during the calculations. The goal is to harden the solver against such model breakdowns by activating heuristics to recover gracefully from these situations, and carry on optimising instead of terminating unsuccessfully.

- Multi-Objective module: Seeing as real-world optimisation problems rarely have a single scalar cost function only, it is desirable to enhance the solver such that it is able to solve problems with multiple cost functions without requiring additional infrastructure. The goal of this task is therefore to develop a multi-objective module that calculates a so-called Pareto-front, whose points are optimal solutions for different mutual combinations of the objective functions.
- Filter: WORHP uses a mathematical 2D-Filter method to determine step sizes for the iterative calculations performed. The goal of this task is to determine whether additional criteria can help to speed up the solver by finding “better” step sizes, and possibly extend the filter from single to multiple objectives, as a natural extension.
- Structure Exploitation: WORHP relies on sparse linear algebra solvers to compute its search direction. These are highly complex and sophisticated pieces of software, but generic, so they are suitable for a very wide range of problems. Since the problem structure of trajectory optimisation problems is known, a tailored linear solver for problems of this particular structure should be able to outperform the generic solvers, both in terms of runtime and memory requirements. The goal was to implement such a specialised solver for a range of selected typical problem structures, depending on the discretisation scheme that is used to mathematically model the trajectory optimisation problem.
- Multiple shooting: As a companion tool to WORHP, TransWORHP can be used to transform infinite-dimensional trajectory optimisation problems into finite-dimensional (but potentially large) optimisation problems that can then be solved by WORHP. TransWORHP currently uses a so-called full discretisation approach, designed for exploiting WORHP’s strengths on large-scale problems. Given that realistic aircraft trajectory optimisation problems are (in parts) highly constrained, and to serve as an additional test platform for WORHP development, the alternative discretisation approach by multiple shooting was implemented and tested.
- Grid Refinement: Discretisation schemes consider trajectories at a finite number of points. If this grid of points is too coarse, important or highly dynamic features of the model cannot be represented to sufficient precision, resulting in suboptimal or even wrong solutions, depending on the post-processing. Automatic grid refinement detects these issues and refines the grid where necessary. When starting with a very coarse grid, and iteratively refining it, it should be possible to achieve an overall speedup and lower errors.

3 Description of the main S&T results/foregrounds

3.1 Analysis of Optimisation Environment

In WP2, the overall structure of the solution algorithm for aircraft trajectory optimisation problems has been analyzed. An overview of this structure is depicted in the figure below.



The process used to solve aircraft trajectory optimisation problems is an iterative process that alternately calls the Optimisation Function and the NLP-Solver. The Optimisation Function contains the call to the model dynamics, the cost and the constraint functions that are organized by the discretisation scheme used. This discretisation scheme – also called transcription scheme – can be of various types. In any iteration of the NLP solver it checks whether the algorithm already reached the given tolerances and thus converged or not. In case convergence has been achieved, the optimal solution has been determined and can be handed back to the user. Otherwise, either a suboptimal solution is delivered to the user or the next iteration is performed.

3.1.1 Problem sparsity

As WORHP is particularly tailored to solving sparse problems (problems containing many structural independencies), the sparsity of the aerospace problems has been examined within WP2 of this project.

The aircraft models normally used in aircraft trajectory optimisation problems have great similarities in their structure – at least if standard coordinate systems are chosen for modelling. In all of these models using a local position coordinate format all states are independent of the lateral aircraft motion, as the position of the aircraft in these local coordinate system does not affect the aircraft behavior. This independence leads to a special inherent sparsity pattern of the equations of motion that can be used within the NLP solver as it affects the overall sparsity of the discretized problem. Anyway, the discretisation scheme used affects problem sparsity most.

3.1.2 Certification aspects

Besides the structural analysis described above, in WP2 of AWACs, the applicability of aircraft certification regulations to WORHP has been investigated. It can be seen from the regulations in CS25 and DO-178C and their supplements that a qualification of WORHP was not possible during this project as neither a specific application nor a target hardware has been selected. Moreover, WORHP is already mostly implemented while DO-178C and all other qualification documents specify processes required to follow during the development of a new software designed for the use in aircraft electronics. This makes it almost impossible to apply the regulations directly to WORHP leading to a massive amount of redevelopment for qualification. Consequently, WORHP may be used in less critical applications like Electronic Flight Bags or other tools for pilots.

3.2 New solver interfaces

The technical infrastructure for testing within this project is based on Matlab. Before the activities WORHP offered a basic Matlab interface with some lacks in regard of efficiency and flexibility. Therefore UHB and TUM came to the conclusion that an improved interface is required.

A more flexible and efficient Matlab interface was implemented. The new interface enables the user to interact with the solver via a reverse communication loop, which allows a high degree of flexibility especially for specialised application cases. Furthermore consistent with WORHP's other interfaces the same sparse matrix format is used and memory between Matlab and the solver is shared leading to reduced memory requirements.

TUM implemented a wrapper function to connect their MATLAB based optimisation framework to the new MATLAB version of WORHP. Thus, it is possible to use the full capabilities of TUM's optimisation framework. Furthermore, comparison between different solvers, such as SNOPT, Ipopt and WORHP, was enabled.

3.3 Conservative iteration mode

During the optimisation progress of WORHP the theory does not provide any guarantees for feasible iterates, i.e. flyable trajectories, and most runs even show the contrary, only the final result gets feasible. Especially users in aircraft trajectory optimisation may be interested in flyable trajectories at any time during the optimisation as for example in an on board optimisation scenario the time available for trajectory calculation may be limited.

The conservative iteration feature uses parametric sensitivity analysis, a field of mathematical theory mainly used for post optimality analysis, in order to generate more feasible iterates.

This operational mode is realised via application of post optimality correction on sub problem level during the optimisation progress. The violation of the constraints is regarded as a perturbation of the problem and the variables are corrected in order to compensate this perturbation, which means to fulfill the violated constraints. Convergence for this kind of real time correction has been shown for post optimality correction algorithms and the same procedure is used within the conservative iteration mode.

The implementation is completely automated, the user must only decide whether to use the feature always, never or just to maintain already reached feasibility. Furthermore the algorithm detects situations where the feature is expected to improve convergence and autonomously decides when to terminate the conservative iteration mode again. Therefore no further detailed theoretical background is required from the user.

In trajectory optimisation different objectives like fuel consumption or noise level minimisation can be formulated. However one may be interested to switch from one objective to another. The underlying optimisation maintains quite similar as only the objective function changes while the constraints are not modified. The conservative iterations mode can be used to exploit the supposed vicinity of the different solutions to those problem formulations. Testing on academic mathematical examples using a homotopy approach show the capability of the operational mode. Normal optimisation runs generated around 30% feasible iterates the conservative iteration mode produced around 52% feasible iterates while reducing the overall necessary iterations by 16%.

The feature was demonstrated to improve overall convergence on average NLP problems, and to better retain feasibility, when the problem is modified. In resource-constrained environments like EFBs, this will increase the number of cases where a flyable (possibly suboptimal) trajectory can be computed in a given time frame.

3.4 Robustness to erroneous inputs

Problematic model behaviour like infinite or not-a-number function or derivative values usually cause an optimisation algorithm to break down. As depending on implementation singularities in aerospace models may lead to such cases, WORHP has to be safeguarded against these.

Several handlers for erroneous inputs were implemented in the solver.

The concurrent testing proved the success of the implemented extension, where the solver can defend against semi-random insertions of such values.

The increased robustness of the solver is able to prevent unsuccessful terminations due to technical problems of the dynamic model. Again regarding EFBs, this will increase the likelihood to gracefully recover from breakdowns of the aircraft model, and produce useful results.

3.5 Direct multiple shooting

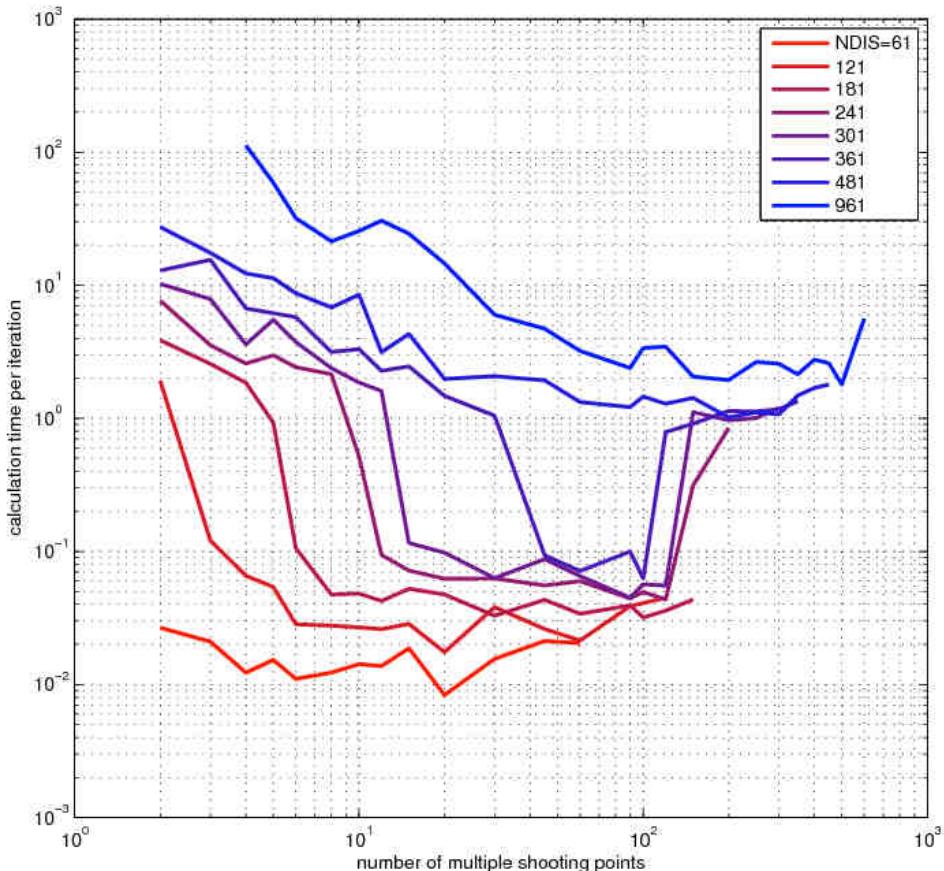
So far TransWORHP used only full discretisation to generate an NLP problem custom tailored for the sparse solve WORHP.

Within this project the transcription method TransWORHP was extended by direct multiple shooting methods.

The direct multiple shooting implementation reuses the TransWORHP interface to easily switch between discretisation methods. For direct multiple shooting, the user provides a free number of multiple shooting nodes and selects an explicit Runge-Kutta-scheme. The optimisation variables consist of the controls at every discretisation point, and the states at every multiple shooting node.

The sparse matrix structures, which are can efficiently be exploited by WORHP, consist of multiple blocks for Jacobian and Hessian and are generated automatically for direct multiple shooting. For the Jacobian a more detailed (repeating) pattern will be extracted from user defined ode structures. UHB analysed various ratios of shooting points and discretisation to the full discretisation.

The numerical results showed a breakeven point between full discretisation and single shooting, which can be reached using multiple shooting, allowing the transcription to run highly efficient.



An optimal control problem in TransWORHP can consist of several freely linkable phases to allow optimisation of staged problems, e.g. in mission planning. Each phase uses its own discretisation method (full discretisation or direct multiple shooting) to integrate the ode system.

3.6 Grid refinement

After a solution was found within the full discretisation mode of TransWORHP the solution is not only available at the discretisation points, but also between adjacent points using spline interpolation. This can be used to estimate the local discretisation error for the state dynamics between

discretisation points. If discretisation points are added automatically to critical intervals before a new optimisation the local discretisation error can be reduced while keeping the number of optimisation variables small.

In a similar way, by analysing the adjoint variables the switching structure of controls and states of the optimal solution can be refined.

In the current implementation the user can adjust several parameters for the iterative refinement process.

3.7 Structure exploitation

The structure of discretized constrained optimal control problems was analyzed for different discretisation schemes like the explicit Euler scheme, the implicit Euler scheme, and the Trapezoidal scheme. Based on this structure, a routine was implemented that approximates the Jacobian and Hessian matrix particularly for discretized optimal control problems. A condensing method was implemented with the aim to solve large-scale and sparse linear equation systems arising in discretized optimal control problems more efficiently. The condensing algorithm was tested on sample applications with regard to its dependence on the number of states, controls, and grid points, but showed a very poor runtime performance compared to the direct sparse LU factorization method MA97. Further tests have been conducted to identify common problems of MA97 and to identify crucial parameters of MA97. Common issues are singular KKT matrices or matrices with large elements. Recovery strategies for such cases are in place and work properly. The pivot tolerance parameter of MA97 was found to have a notable influence on the performance of MA97. A recovery strategy using this pivot tolerance parameter was tested and showed good performance. Finally, some effort was invested to be able to use WORHP in a multithreaded application. The multithreading capability of WORHP has been successfully tested and a very good speedup could be achieved.

3.8 Multiobjective Optimisation

Many problems in aircraft trajectory optimisation contain multiple conflicting objectives. Examples include minimizing noise footprint, whilst simultaneously minimizing NOx emissions, and likewise simultaneously minimizing travel time, fuel expenditure, or various other emissions. Such problems are inherently hard, and modern optimisation solvers still lack the automatic capability to solve such problems.

The standard way to attack such problems is to compute trade-off information between various criteria to be optimised simultaneously, and display such trade-off curves to relevant decision makers. This strategy implies transforming the given problem (with several objectives) into an infinite number of (classical) problems with a single objective, and then solving a carefully selected sample of these. An extension to WORHP, the multiobjective WORHP extension (M-WORHP) is now able to use various different strategies to transfer a multiobjective problem with several objectives into an infinite number of classical optimisation problems, and to use an automatic sampling strategy that provides provably good results for problems with two or three criteria, and that performs well in practice for problems with a higher number of criteria. This implementation is fully parallel, i.e. it makes automatic use of multicore systems by solving the various classical problems sampled by different cores, thereby achieving an almost perfect speedup.

3.9 Testing

The testing of WORHP has been performed in different modes: In parallel to the development process, concurrent testing has been performed (from 01/2014 - 04/2014 and from 07/2014 - 12/2014). Additionally, to get a more detailed picture of the current status, from 05/2014 - 06/2014 the Intermediate Test Campaign took place. By the end of the project, in 01/2015, the Final Test Campaign was performed. This test campaign was very similar to the Intermediate Testing and the results were used to evaluate the projects success.

The concurrent tests mainly focused on general testing of functionality and robustness. Besides the CUTEst test set, some of the optimal control test cases have been used there.

In the Intermediate and Final Test Campaigns, all tests have been performed, including comparative testing, that enabled a comparison of performance of WORHP to the competing optimisation algorithms IPOPT and SNOPT. Additionally, exhaustive feature testing was done giving a full picture of the goals achieved at that time and the items still to be tackled.

3.9.1 Goals of testing

The goal of testing on the one hand was the evaluation of the features to be added to WORHP during the project and on the other hand the assessment of the overall performance of WORHP also compared to other solvers. The features especially tested were:

- The conservative Iteration mode
- Multiple Objectives
- Robustness to errors

Besides, the overall performance of an optimisation algorithm is best measured by the time and the number of iterations the algorithm needed to converge to the optimal solution. These have also been the two main criteria for the comparative performance testing of WORHP.

3.9.2 Mathematical testing vs. aircraft related testing

To test the basic mathematical performance of WORHP the CUTEst test set has been used. CUTEst is a comprehensive testing and benchmarking environment for NLP solvers based on over 1000 individual test problems.

The aircraft related test cases were made up from real application scenarios of future electronic flight bag software to be able to realistically represent the possible usage scenarios of WORHP. The aircraft model used in the test cases aims at representing a civil transport aircraft. It is mainly composed of:

- An aerodynamic model
- A propulsion/fuel consumption model
- A flight envelope model

The aircraft dynamics have been modeled using a point mass simulation model with parameters taken from the Base of Aircraft Data (BADA) and the Aircraft Noise and Performance (ANP) database,

both published by EUROCONTROL. For simplification, a non-rotating earth is assumed, while the equations for both a flat and a round WGS-84 earth model were implemented. Besides that full 3-dimensional model, in some of the test cases a simplified pure longitudinal model has been used.

The optimisation criteria for the test cases were fuel consumption and noise. For fuel consumption an overall mass balance was used, where the mass change of the aircraft has to be minimized. For minimum noise trajectories, the criteria to be minimized is the “Number of Awakenings” (NA), i.e. the number of people likely to be awakened due to a single flyover.

Besides the aircraft model and the cost functions several constraints have to be fulfilled. Additionally to the aircraft flight envelope inequality constraints, these are the following equality path constraints:

- Calibrated airspeed
- Mach number
- Specify excess power sharing ratio

In total, 12 aircraft related scenarios plus two standard optimal control problems have been used for performance testing in this project.

3.9.3 Test Results

All requirements have been successfully tested. Consequently, WORHP fulfills all requirements put up in this project. The following table gives a detailed overview of the requirements tested and the results achieved.

Functional Requirements	Description	Status
F-Req-01	Problem formulations, multi-objective formulations	Completely fulfilled
F-Req-02	Deterministic, gradient based approach	Completely fulfilled
F-Req-03	Exploit sparsity, solve huge problems	Completely fulfilled
F-Req-04	Default to dense problem	Completely fulfilled
F-Req-05	Nonlinear, nonconvex constraints	Completely fulfilled
F-Req-06	Initial guess for primal variables	Completely fulfilled
F-Req-07	Initial guess for dual variables	Completely fulfilled
F-Req-08	User specified gradient function	Completely fulfilled
F-Req-09	User specified Jacobian function	Completely fulfilled
F-Req-10	Interruptible solver	Completely fulfilled
F-Req-11	Conservative iteration mode	Completely fulfilled
F-Req-12	Robustness against erroneous inputs	Completely fulfilled
F-Req-13	Output of a solution for primal and dual variables	Completely fulfilled
F-Req-14	Indicate termination criterion	Completely fulfilled
Technical Requirements	Description	Status
T-Req-01	Established programming language	Completely fulfilled
T-Req-02	Interfaces with C/C++, Fortran	Completely fulfilled
T-Req-03	Interfaces of different complexity	Completely fulfilled

T-Req-04	MATLAB interface	Completely fulfilled
T-Req-05	Compatible with Windows, Linux, Mac	Completely fulfilled
T-Req-06	32bit and 64bit modes	Completely fulfilled
T-Req-07	No specific hardware / software required	Completely fulfilled

Turning to performance testing, the following table lists the results for the testing based on the CUTEst test set. Five of the 1149 problems were excluded, since they produced timeouts for any setting or modification, and therefore skewed the timing results. The tests were run on *titan*, UHB's 24-core Xeon E5-4617 compute server. The results for WORHP version V1, delivered in the middle of the project are compared to version V2, the final version of the project.

	V1		V2	
Total	1144		1144	
Successful	1053	92.0%	1095	95.7%
Optimal	998	87.3%	1074	93.9%
Acceptable	55	4.8%	21	1.8%
Unsuccessful	91	8.0%	49	4.2%

It can be clearly seen that Version V2 has notably improved: In comparison to V1, unsuccessful terminations were reduced by nearly 50%, acceptables by over 60%. Timing results mirror this success, showing a CPU time reduction of 60% and an even higher reduction of wall time:

	V1	V2
wall time	1h:54m	0h:40m
user time (CPU time)	20h:32m	8h: 8m

The results achieved in the aircraft related optimal control test cases have been very good, too. Summarizing all optimal control performance tests, WORHP is able to solve 11 of the 12 test cases considered (2 cases where skipped altogether), whereas IPOPT can only solve 9 and SNOPT 7. Besides, the test results show that WORHP also became more robust during the project. When comparing the 6 test cases that can be solved by all solvers the following cumulated runtimes result:

Solver	Cumulated runtime
WORHP	627 s
IPOPT	325 s
SNOPT	1392 s

These runtimes show that overall, IPOPT is still the fastest solver for the frame used here, but cannot solve as many problems as WORHP can, while SNOPT performs relatively poor. Anyway, these performance numbers have to be read with care as two single test cases affect the sum a lot. Overall, the tests show that WORHP is a very competitive solver for sparse constrained optimisation problems that furthermore was very much improved during this project.

4 Potential impact

We can generally state that the immediate impact of this project is hard to observe and quantify, as mathematical optimisation is a mid-to-long-term enabler for other, more immediate and visible technological advances. Its impact is therefore appropriately assessed both from the technical improvements made available from using mathematical optimisation, and from a strategic European point of view.

4.1 Technical impact

The project aims at lowering the bars for using a mathematical optimisation solver, a so-called NLP solver, onboard of a commercial airplane, in particular on Electronic Flight Bags, or on the ground in the frame of flight services. Having an efficient and robust NLP solver onboard enables high fidelity aircraft trajectory optimisation, which in turn enables a wide variety of improvements in air traffic, such as a reduction in fuel consumption and correlated emissions, as well as a reduction in noise perceived by people living in the surroundings of airports.

Nowadays, flight management systems available to pilots have to use several assumptions to be able to calculate fuel and emission minimal flight paths for the aircraft. As these assumptions are not always valid the results sometimes are imprecise and the aircraft cannot use their full potential of savings. When using a high performance robust optimisation algorithm either directly in the aircraft's equipment (e.g. the flight management system) or in the pilots additional tools (e.g. in an Electronic Flight Bag - the devices a pilot uses to perform additional calculations for the flight) the potential can be used to further reduce pollution and noise.

On the one hand, the reductions in emissions and noise are desired by the people living around an airport and due to ecological reasons. On the other hand, a reduction in fuel consumption helps European airlines reduce their operating costs and therefore defend their positions in this highly competitive market. Despite its current low, the oil price is expected to rise again in the medium and long term, which will cause fuel efficiency to play an important role for the economic efficiency of an airline.

4.2 Strategic impact

We assume that the project will have a two-fold strategic impact: Raising acceptance in the flight management community and increasing European capabilities and independence of US products in optimisation.

Concerning acceptance, we hope that the solver improvements will raise acceptance and usability of mathematical optimisation in and by the European aerospace community; while it seems that optimisation is a tool that is well-known and routinely used by some, the community as a whole is extremely safety-conscious and therefore conservative (and with reason). The robustness, conservative iteration and interruptibility extensions made to WORHP should be particularly useful for building higher acceptance, since these were identified as critical issues of existing NLP solvers, such as Ipopt.

Concerning capabilities, all major NLP solvers, i.e. commercially established or widely used by academia and industry, are non-European products; notable examples are SNOPT (Stanford University), KNITRO (roots at Northwestern University), and Ipopt (Carnegie Mellon University and

IBM), two of which were also considered as comparison baseline in the testing campaign. WORHP is initially the result of a strategic ESA initiative to reduce the dependence of the European space community on US software for mission analysis and design optimisation and to build up European competence and capabilities for industrial-strength nonlinear optimisation software sufficiently to sustain itself. WORHP has since reached sufficient maturity to be considered for this project. Whereas optimisation has long been an essential requirement for space mission analysis, this is not yet the case for aircraft operation, though we are convinced that technological progress will continue (also thanks to applicable optimisation software) until it becomes essential to maintain the achieved technological standard. We therefore strongly believe that sustaining and extending European capabilities in mathematical optimisation is of strategic interest to the European Union and its high-technology industry, and that this project has supported this interest. The project is furthermore expected to enrich at least two forthcoming dissertations at Bremen University and the Technical University in Munich.