Flight Control Law Testing Using Optimal Control and Postoptimal Sensitivity Analysis

Johannes Diepolder, Saurabh Saboo, Venkata Sravan Akkinapalli, Stefan Raab, Jiannan Zhang, Pranav Bhardwaj, Michael Krenmayr, Benedikt Grüter and Florian Holzapfel

1 Introduction

For the clearance of flight control laws, it is required to show that the aircraft cannot be driven to an unsafe states, especially considering inherent modeling uncertainties and disturbances like wind gusts. Current industry practice is to use mostly gridding or monte carlo methods for testing closed loop systems. These methods are straightforward to use, but often computationally expensive, due to the fact that numerous simulations have to be performed to clear the flight control law. Additionally, they do not refine the search space around worst cases, and may miss solutions in between grid points. In essence, these methods evaluate multiple scenarios to show that for all possible combinations of uncertain parameters or maneuvers, safe aircraft operation is ensured. Contrary, optimization based clearance techniques try to find worst cases by means of optimization algorithms. These methods do not investigate the whole solution space for all possible scenarios, but try to find worst cases for which the respective criteria is violated. In the past, considerable effort was made to develop suitable optimization algorithms for this kind of task.

Reference [12] investigates the use of a novel global optimization technique based on dividing rectangles for the nonlinear clearance criteria of a hypersonic re-entry vehicle. The results showed that this approach has potential for significantly enhancing the confidence for the clearance task. In [6], the authors compare probabilistic

J. Diepolder (✉) · S. Saboo · V.S. Akkinapalli · S. Raab · J. Zhang · P. Bhardwaj · M. Krenmayr · B. Grüter · F. Holzapfel
Institute for Flight System Dynamics, TU Munich, Boltzmannstraße 15, 80747 Garching bei München, Germany
e-mail: johannes.diepolder@tum.de

S. Saboo
e-mail: ssaboo@live.de

F. Holzapfel
e-mail: florian.holzapfel@tum.de

B. Dołęga et al. (eds.), *Advances in Aerospace Guidance, Navigation and Control*, https://doi.org/10.1007/978-3-319-65283-2_2

monte carlo methods, and a non-dominated sorting genetic algorithm for multi-criteria optimization for a nonlinear AIRBUS aircraft simulation model.

Reference [8] tests the functionality of a maneuver load limiter using the clonk stability criterion. The authors used a genetic algorithm and adaptive simulated annealing for the worst case parameter search. Reference [13] augments *Differential Evolution* and genetic algorithms by a local optimization method for the clearance task. This results in a hybrid optimization scheme with improved convergence and computational efficiency. The method is illustrated using a six-degree of freedom fighter aircraft model (ADMIRE). In [16] local and global optimization methods are investigated to enhance current industry practice for the clearance task. Especially, the class of evolutionary algorithms like genetic algorithms, evolutionary strategies and *Differential Evolution* showed high potential in finding worst case pilot inputs for criteria such as the maximum angle of attack exceeding (alpha protection) or high velocity/Mach Number (high speed protection). One of the main outcomes from this study is a workflow for the optimization based clearance, which complements existing techniques (Gridding/Monte Carlo) by global and local optimization methods and sensitivity analysis. In this paper, we propose to enhance this workflow by the use of *Optimal Control Theory* and *Postoptimal Sensitivity Analysis*. The paper is organized as follows. First we present the novel methodology and the proposed workflow. Then we give a short introduction to *Differential Evolution*, *Optimal Control Theory* and *Postoptimal Sensitivity Analysis*. To illustrate the methodology, we investigate the height loss for a preliminary control design of a tilt rotor vehicle, from hover-mode to forward flight.

2 Optimal Control Based Clearance of Flight Control Laws

The workflow proposed in [16] for the optimization based clearance of flight control laws consists of four steps. First, a monte carlo analysis is performed to find worst cases for the respective criteria in the flight envelope. These worst cases are then used in a second step to initialize global optimization runs, to explore the search space surrounding those worst cases. The results of global optimization are then refined by local optimization methods, such as the quasi-Newton method in combination with cyclic coordinate descent. The last step consists of performing sensitivity analysis for these refined solutions e.g. by parameter perturbation and re-simulation. Reference [5] motivates the use of optimal control theory for the determination of worst case pilot inputs and wind gusts. The authors investigated an aircraft model, linearized around a horizontal steady state flight condition in the longitudinal and lateral plane. Using a direct collocation scheme implemented in the MATLAB software FALCON.m, the authors were able to obtain meaningful results for different criteria, such as the maximum angle of attack exceeding in the longitudinal plane and the maximum angle of sideslip exceeding in the lateral plane. These results motivate further investigation of using *Optimal Control Theory* for flight control law testing. For the solution of optimal control problems using direct transcription methods (e.g.

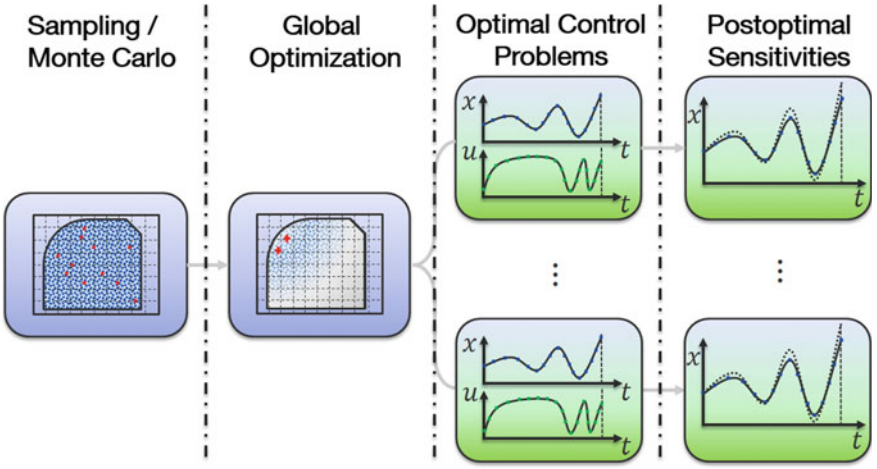

Fig. 1 Novel workflow for optimal control based clearance of flight control laws

shooting or collocation) in combination with gradient based optimization, it is crucial to provide good initial guesses. Therefore, we propose to enhance the workflow from [16] by replacing the last two steps namely local optimization and sensitivity analysis, by *Optimal Control Methods* and *Postoptimal Sensitivity Analysis* (Fig. 1).

3 Theoretical Background

3.1 Global Optimization: Differential Evolution

Differential Evolution (DE) is a global optimization algorithm, which has increasingly gained attention due to its simple structure on one side, and its effectiveness for a wide range of optimization problems on the other [14]. For the current application, we use the classic DE implementation "*DE/rand/1/bin*", for which the base vector is selected *randomly*, *one* weighted difference vector is added to it and the parameter crossover from the mutant vector follows a *binomial* distribution. The basic algorithm is depicted in Algorithm 1. Here, we define the population of generation $g = 1...g_{max}$ as $\mathbf{P}_z^g$. This population consists of N_p parameter vectors $\mathbf{z}_i^g$ with $i = 0, 1, ..., N_p - 1$. Each of the parameter vectors has N_z real valued elements $z_{i,j}^g$ for $j = 0, 1, ..., N_z - 1$. Similarly, we define the mutant vector population $\mathbf{P}_m^g$ and the trial vector population $\mathbf{P}_t^g$:

$$\mathbf{P}_z^g = \bigcup_{i \in 0,1,...,N_p-1} \mathbf{z}_i^g, \mathbf{z}_i^g = [z_{i,0}^g, z_{i,1}^g, ..., z_{i,N_z-1}^g]^T \tag{1}$$

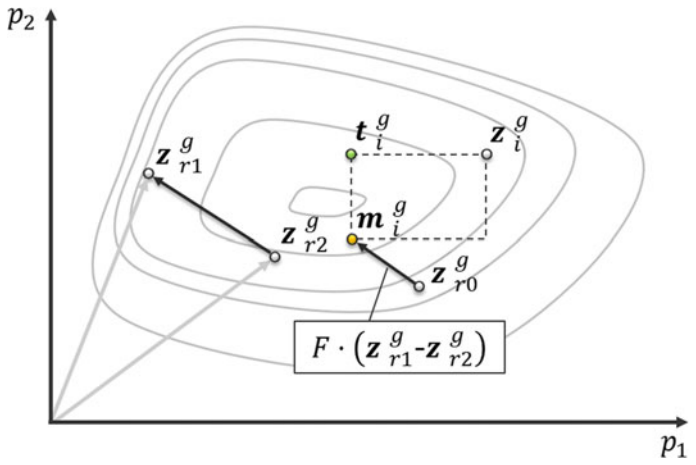

Fig. 2 Illustration of mutation and crossover for differential evolution in a two dimensional parameter space

$$\mathbf{P}_m^g = \bigcup_{i \in 0,1,\ldots,N_p-1} \mathbf{m}_i^g, \mathbf{m}_i^g = [m_{i,0}^g, m_{i,1}^g, \ldots, m_{i,N_z-1}^g]^T \tag{2}$$

$$\mathbf{P}_t^g = \bigcup_{i \in 0,1,\ldots,N_p-1} \mathbf{t}_i^g, \mathbf{t}_i^g = [t_{i,0}^g, t_{i,1}^g, \ldots, t_{i,N_z-1}^g]^T \tag{3}$$

At first, the initial population is generated by randomly assigning values between the upper bounds $\mathbf{z}_{ub}$ and lower bounds $\mathbf{z}_{lb}$:

$$z_{i,j}^0 = rand(0, 1) \cdot (z_{ub,j} - z_{lb,j}) \tag{4}$$

To evolve the existing generation, we first create the mutant population $\mathbf{P}_m^g$. For each mutant vector, we randomly pick three parameter vectors $\mathbf{z}_{r0}^g$, $\mathbf{z}_{r1}^g$ and $\mathbf{z}_{r2}^g$, and add the weighted difference vector of $\mathbf{z}_{r1}^g$ and $\mathbf{z}_{r2}^g$ with weight $F \in [0, 1]$ to the base vector $\mathbf{z}_{r0}^g$:

$$\mathbf{m}_i^g = \mathbf{z}_{r0}^g + F \cdot (\mathbf{z}_{r1}^g - \mathbf{z}_{r2}^g) \tag{5}$$

From the mutant vector $\mathbf{m}_i^g$, the trial vector $\mathbf{t}_i^g$ is constructed via crossover of the elements, with crossover probability $p_c \in [0, 1]$:

$$t_{i,j}^g = \begin{cases} m_{i,j}^g & if(rand(0, 1) \leq p_c) \\ z_{i,j}^g & else \end{cases} \tag{6}$$

Figure 2 illustrates the generation of a trial vector in a two dimensional parameter space. The grey lines in Fig. 2 represent the isolines of the cost function J.

In the selection phase, the trial vectors $\mathbf{t}_i^g$ compete against the target vectors $\mathbf{z}_i^g$ from the current population, by comparing their objective function values $J(\mathbf{t}_i^g)$ and $J(\mathbf{z}_i^g)$. The vectors with the lower cost values win and become part of the next generation:

$$\mathbf{z}_i^{g+1} = \begin{cases} \mathbf{t}_i^g & if\, J(\mathbf{t}_i^g) \leq J(\mathbf{z}_i^g) \\ \mathbf{z}_i^g & else \end{cases} \tag{7}$$

Algorithm 1 Differential Evolution - *DE/rand/1/bin*

1: $F \leftarrow [0, 1]$, $p_c \leftarrow [0, 1]$,$g \leftarrow 0$
2: **for all** $i \in 0, 1, ..., N_p - 1$, $j \in 0, 1, ..., N_z - 1$ **do**
3: $z_{i,j}^0 = rand(0, 1) \cdot (z_{ub,j} - z_{lb,j})$
4: **end for**
5: **while** $g \leq g_{max}$ **do**
6: $g \leftarrow g + 1$

Mutation

7: **for all** $i \in 0, 1, ..., N_p - 1$ **do**
8: **Select** $\mathbf{z}_{r0}^g, \mathbf{z}_{r1}^g, \mathbf{z}_{r2}^g$
9: $\mathbf{m}_i^g = \mathbf{z}_{r0}^g + F \cdot (\mathbf{z}_{r1}^g - \mathbf{z}_{r2}^g)$
10: **end for**

Crossover

11: **for all** $i \in 0, 1, ..., N_p - 1$, $j \in 0, 1, ..., N_z - 1$ **do**
12: $t_{i,j}^g = \begin{cases} m_{i,j}^g & if(rand(0, 1) \leq p_c) \\ z_{i,j}^g & else \end{cases}$
13: **end for**

Selection

14: **for all** $i \in 0, 1, ..., N_p - 1$ **do**
15: $\mathbf{z}_i^{g+1} = \begin{cases} \mathbf{t}_i^g & if\, J(\mathbf{t}_i^g) \leq J(\mathbf{z}_i^g) \\ \mathbf{z}_i^g & else \end{cases}$
16: **end for**
17: **end while**

Please note that for the workflow proposed in Sect. 2, we modify the steps 2–4 in Algorithm 1, and use *Latin Hypercube Sampling* (LHS) to initialize the first generation. By doing so, we ensure that the algorithm is "warmstarted" with the worst cases found by LHS, and explore the region around these worst cases using DE.

3.2 Optimal Control Theory

3.2.1 General Problem Statement

Optimal Control Theory [1–3] seeks to solve the following problem: Find the optimal control $\mathbf{u}^*(t)$ which minimizes the Bolza cost function J

$$J = e(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t. \tag{8}$$

subject to the dynamic equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{9}$$

the initial and final boundary conditions at the initial time t_0 and the final time t_f

$$\psi(\mathbf{x}(t_0), \mathbf{x}(t_f)) = \mathbf{0} \tag{10}$$

and the equality and inequality path constraints

$$\mathbf{C}_{P_{eq}}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0} \tag{11}$$

$$\mathbf{C}_{P_{ineq}}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \tag{12}$$

which are defined on the time interval $t \in [t_0, t_f]$. The general case of the Bolza cost function includes the Mayer cost $e(\mathbf{x}(t_f), t_f)$, which only depends on the state values at the end of the time horizon and the Lagrange cost $L(\mathbf{x}(t), \mathbf{u}(t), t)$, which is integrated from the starting time t_0 to the terminal time t_f. Note, that the above formulation also allows for the introduction of the parameter p by introducing an additional state x_{n+1} with:

$$\dot{x}_{n+1} = 0 \tag{13}$$

and the initial condition

$$x_{n+1}(t_0) = p \tag{14}$$

3.2.2 Solution Strategies

Most of the solution strategies for optimal control problems can be classified into two basic approaches. The first one is named the indirect approach and relies on deriving necessary conditions (*Minimum Principle*) and numerically solving a resulting boundary value problem. To do so, it is useful to introduce the Hamiltonian H by adjoining the right hand side of the dynamic equation (9), to the Lagrange cost function L, using the costates $\boldsymbol{\lambda}(t)$ and the multiplier l_0:

$$H = \boldsymbol{\lambda}(t) \cdot \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + l_0 \cdot L(\mathbf{x}(t), \mathbf{u}(t), t) \tag{15}$$

The *Minimum Principle* states that for the optimal control u^*, the Hamiltonian H has to be stationary:

$$H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}^*(t), t) = \min_{\mathbf{u}} H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}(t), t) \tag{16}$$

and additionally the Euler Lagrange Equation:

$$\dot{\boldsymbol{\lambda}}^T(t) = -\nabla_{\mathbf{x}} H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}^*(t), t) \tag{17}$$

as well as the transversality conditions

$$\boldsymbol{\lambda}(t_0) = -\nabla_{\mathbf{x}_0}[l_0 \cdot e(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f)) + \boldsymbol{\sigma}^T \cdot \psi(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f))]^T \tag{18}$$

$$\boldsymbol{\lambda}(t_f) = -\nabla_{\mathbf{x}_f}[l_0 \cdot e(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f)) + \boldsymbol{\sigma}^T \cdot \psi(\mathbf{x}^*(t_0), \mathbf{x}^*(t_f))]^T \tag{19}$$

have to be satisfied for $(l_0, \boldsymbol{\lambda}^T, \boldsymbol{\sigma}^T) \neq \mathbf{0}$. The solution of the necessary conditions usually requires the knowledge of the switching structure of the problem, and in most real world applications with complex dynamics or constraints, the solution using indirect methods is very difficult and often only possible for heavily simplified models.

The second approach for solving optimal control problems is called the direct approach and transcribes the infinite dimensional optimal control problem into a finite dimensional parameter optimization problem. There exist various methods for the transcription e.g. single shooting, multiple shooting or collocation. All of these methods discretize the dynamic equation in time, and introduce additional equality constraints, which ensure that the dynamic equation is satisfied. The main difference between these transcription schemes is how the dynamics are discretized on the time grid $\mathbf{t} = [t_0, t_1 \ldots t_f]$.

In case of single shooting using a Euler discretization scheme

$$\mathbf{x}_{i+1} = \mathbf{x}_i + (t_{i+1} - t_i) \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i) \tag{20}$$

we only introduce the initial state $\mathbf{x}_0$ and the controls $\mathbf{u}_{0\ldots f}$ additional to the n_p parameters $\mathbf{p}_{1\ldots n_p}$ to obtain the optimization parameter vector $\mathbf{z}_{SS}$:

$$\mathbf{z}_{SS} = [\mathbf{p}_{1\ldots n_p}, \mathbf{x}_0, \mathbf{u}_{0\ldots f}]^T \tag{21}$$

In order to improve numerical stability it is often necessary to introduce n_s additional shooting-nodes $\mathbf{x}_{s_{0\ldots n_s-1}}$ at distinct time points t_i at which the dynamic equality constraints

$$\mathbf{C}_{dyn,j} = \mathbf{x}_{s_j} - \mathbf{x}(t_i) = 0,\ j = 0...n_s - 1 \tag{22}$$

have to be fulfilled. This method is called multiple shooting and yields the following parameter vector:

$$\mathbf{z}_{MS} = [\mathbf{p}_{1...n_p}, \mathbf{x}_{s_{0...n_s-1}}, \mathbf{u}_{0...f}]^T \tag{23}$$

A third option is to use collocation methods e.g. the trapezoidal collocation scheme for which we introduce one state and control per time discretization point. The dynamic constraints $\mathbf{C}_{dyn,i}, i = 0...f - 1$ in this case become:

$$\mathbf{C}_{dyn,i} = \mathbf{x}_{i+1} - \mathbf{x}_i - \frac{1}{2} \cdot (\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i) + \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}, t_{i+1})) \cdot (t_{i+1} - t_i) \tag{24}$$

and the parameter vector contains the parameters $p_{1...n_p}$ and the fully discretized states and controls $\mathbf{x}_{0...f}, \mathbf{u}_{0...f}$:

$$\mathbf{z}_{Col} = [\mathbf{p}_{1...n_p}, \mathbf{x}_{0...f}, \mathbf{u}_{0...f}]^T \tag{25}$$

In case of collocation methods the resulting optimization problem exhibits a strongly decoupled and hence sparse structure at the cost of a greatly increased number of optimization variables and constraints.

Additional to the dynamic constraints $\mathbf{C}_{dyn}$ the equality and inequality constraints (11) and (12) are discretized on the time grid $\mathbf{t}$:

$$\mathbf{C}_{P_{eq},i}(x_i, u_i) = 0 \tag{26}$$

$$\mathbf{C}_{P_{ineq},i}(x_i, u_i) \leq 0 \tag{27}$$

The residual vector $\mathbf{F}$ holds the cost function (8), the dynamic constraints (22) or (24) and the discretized inequality and equality constraints (26), (27) to be satisfied:

$$\mathbf{F} = [\psi(\mathbf{x}(t_0), \mathbf{x}(t_f)), \mathbf{C}_{P_{eq},0...f}, \mathbf{C}_{P_{ineq},0...f}, \mathbf{C}_{dyn,0...n_s-1/f-1}]^T \tag{28}$$

Both, the parameter vector $\mathbf{z}$ and the residual vector $\mathbf{F}$ are required to satisfy box constraints with the respective upper and lower bounds $\mathbf{z}_{ub}$, $\mathbf{z}_{lb}$,$\mathbf{F}_{ub}$, $\mathbf{F}_{lb}$:

$$\mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub} \tag{29}$$

$$\mathbf{F}_{lb} \leq \mathbf{F} \leq \mathbf{F}_{ub} \tag{30}$$

3.3 Post-optimal Sensitivity Analysis

Postoptimal Sensitivity Analysis [4, 7] provides the sensitivities of the optimal solution with respect to parameters $\boldsymbol{\omega}$. The derivation of the local sensitivities is based on the Fiacco equation, which is obtained by applying the implicit function theorem to the Karush–Kuhn–Tucker (KKT) conditions. The resulting sensitivity equation for the parameter vector $\mathbf{z}$, and the Lagrange-multipliers $\boldsymbol{\lambda}$, for the constraints reads:

$$\begin{pmatrix} \frac{d\mathbf{z}}{d\boldsymbol{\omega}} \\ \frac{d\boldsymbol{\lambda}}{d\boldsymbol{\omega}} \end{pmatrix} = \begin{bmatrix} \nabla_{\mathbf{zz}}\mathscr{L} & \nabla_{\mathbf{z}}\mathbf{C}_a{}^T \\ \nabla_{\mathbf{z}}\mathbf{C}_a & \mathbf{0} \end{bmatrix}^{-1} \cdot \begin{pmatrix} \nabla_{\mathbf{z}\boldsymbol{\omega}}\mathscr{L} \\ \nabla_{\boldsymbol{\omega}}\mathbf{C}_a \end{pmatrix} \tag{31}$$

In (31), $\mathbf{C}_a$ are the active equality constraints and $\mathscr{L}$ is the Lagrangian defined as:

$$\mathscr{L}(\mathbf{z}, \boldsymbol{\lambda}; \boldsymbol{\omega}) = J(\mathbf{z}; \boldsymbol{\omega}) + \boldsymbol{\lambda}^T \cdot \mathbf{C}_a(\mathbf{z}; \boldsymbol{\omega}) \tag{32}$$

This equation provides a first order approximation for perturbations $\Delta\boldsymbol{\omega}$, to obtain updates for the parameter vector and the Lagrange multipliers:

$$\hat{\mathbf{z}} = \mathbf{z} + \frac{d\mathbf{z}}{d\boldsymbol{\omega}} \cdot \Delta\boldsymbol{\omega} \tag{33}$$

$$\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda} + \frac{d\boldsymbol{\lambda}}{d\boldsymbol{\omega}} \cdot \Delta\boldsymbol{\omega} \tag{34}$$

Furthermore, the first and second order sensitivities of the objective function can be computed as follows [4]:

$$\frac{dJ}{d\boldsymbol{\omega}} = \nabla_{\mathbf{z}} J \frac{d\mathbf{z}}{d\boldsymbol{\omega}} + \nabla_{\boldsymbol{\omega}} J \tag{35}$$

$$\frac{d^2 J}{d\boldsymbol{\omega}^2} = \left(\frac{d\mathbf{z}}{d\boldsymbol{\omega}}\right)^T \nabla_{\mathbf{zz}}\mathscr{L} \frac{d\mathbf{z}}{d\boldsymbol{\omega}} + 2\left(\nabla_{\boldsymbol{\omega}\mathbf{z}}\mathscr{L} \frac{d\mathbf{z}}{d\boldsymbol{\omega}}\right)^T + \nabla_{\boldsymbol{\omega}\boldsymbol{\omega}}\mathscr{L} \tag{36}$$

These sensitivities are used to study the effect of parameters on the objective function of the optimization problem. In particular, they provide sensitivities of second order for the worst cases, found by solving the optimal control problem for the respective clearance criteria.

4 Case Study: Testing of the Preliminary Design of a Tilt-Rotor

4.1 Plant - Model Equations

For the case study a 2-D model of a vertical take-off and landing vehicle with tilt rotor mechanism is considered. The reduced model has three degrees of freedom where the translational motion is described in x_B and z_B direction and the rotation is considered around the pitch axis y_B. This aerial vehicle possesses seven control effectors: two rotors on the fuselage, two rotors on the wings with tilt mechanisms and symmetrical elevators at the tail. The propulsion, gravitational and aerodynamic forces acting on the airframe are described in the following.

4.1.1 Propulsion Forces and Moment

Thrust and drag forces from the propeller are taken as follows:

$$T = C_T \omega^2 \rho \frac{d^4 \pi^2}{4} \tag{37}$$

$$D = C_H \omega^2 \rho \frac{d^4 \pi^2}{4} V_A^R \tag{38}$$

Here V_A^R is the aerodynamic velocity at the aircraft reference point R, and C_H and C_T are drag and thrust constants of the propeller, ω is the motor rotational speed, d is the diameter of the rotor and ρ is the atmospheric density. The resulting propulsion force $(\mathbf{F}_P^P)_P$ at the propulsion point P in the propulsion frame P reads:

$$(\mathbf{F}_P^P)_P = \begin{bmatrix} -D \\ -T \end{bmatrix}_P \tag{39}$$

Using the tilt rotor angle δ, the transformation $\mathbf{M}_{PB}$ from body frame B to propulsion frame P is defined as follows:

$$\mathbf{M}_{PB} = \begin{bmatrix} cos(\delta) & sin(\delta) \\ -sin(\delta) & cos(\delta) \end{bmatrix} \tag{40}$$

Note, that the tilt angle δ is zero for rotors on fuselage. Using this transformation we obtain the propulsion forces and moments at the reference point R in the body fixed frame B:

$$(\mathbf{F}_P^R)_B = \mathbf{M}_{PB}^T \cdot (\mathbf{F}_P^P)_P \tag{41}$$

$$(\mathbf{M}_P^R)_B = \mathbf{r}^{RP} \times (\mathbf{F}_P^P)_B \tag{42}$$

4.1.2 Gravitational Forces and Moments

The gravitational forces and moments resulting from aircraft mass m and acceleration g is transformed from the North-East-Down frame O to the body fixed frame B using the Euler angle θ around the pitch axis:

$$(\mathbf{F}_G^R)_B = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ m \cdot g \end{bmatrix}_O \tag{43}$$

$$(\mathbf{M}_G^R)_B = \mathbf{r}^{RG} \times (\mathbf{F}_G^R)_B \tag{44}$$

Here $\mathbf{r}^{RG}$ is the distance from the reference point R to the center of gravity G.

4.1.3 Aerodynamic Forces and Moments

For the aerodynamic forces and moments, the wing and tail section are considered separately. The aerodynamic angle of attack for the wing α_w is determined as follows:

$$\alpha_w = \text{atan2}\left(\frac{(w_A^G)_B^E - (\omega_y^{IB})_B \cdot (x^{RW})_B}{(u_A^G)_B^E + (\omega_y^{IB})_B \cdot (z^{RW})_B} \right) + \alpha_0 + i \tag{45}$$

Here $(u_A^G)_B^E$ and $(w_A^G)_B^E$ are aerodynamic velocities at the center of gravity G in x_B and z_B direction of the body frame B. $(\omega_y^{IB})_B$ is the angular velocity around the pitch axis, $(x^{RW})_B$ and $(z^{RW})_B$ are the distances from the wing reference point W to the reference point R, α_0 is the angle of attack at zero velocities and i is the inclination angle. The transformation matrix from body axis to wing axis is determined as:

$$\mathbf{M}_{WB} = \begin{bmatrix} \cos(\alpha_w) & -\sin(\alpha_w) \\ \sin(\alpha_w) & \cos(\alpha_w) \end{bmatrix} \tag{46}$$

The aerodynamic forces for the wing $(\mathbf{F}_W^W)_W$ originating from the lift and drag forces L_W and D_W at the wing reference point W in wing frame W are calculated as follows:

$$L_W = \frac{1}{2} \cdot \rho V_A^{R^2} S_W C_{L\alpha,W} \alpha_w \tag{47}$$

$$D_W = \frac{1}{2} \cdot \rho V_A^{R^2} S_W C_{D\alpha,W} \alpha_w \tag{48}$$

$$(\mathbf{F}_W^W)_W = \begin{bmatrix} -D_W \\ -L_W \end{bmatrix}_W \tag{49}$$

Here $C_{L\alpha,W}$ and $C_{D\alpha,W}$ are the wing aerodynamic coefficients, α_W is the angle of attack of the wing and S_W is the wing reference area. The aerodynamic forces and moments from the wing at the reference point R denoted in the body frame B are determined as:

$$(\mathbf{F}_W^R)_B = \mathbf{M}_{WB}^T \cdot (\mathbf{F}_W^W)_W \tag{50}$$

$$(\mathbf{M}_W^R)_B = \mathbf{r}^{RW} \times (\mathbf{F}_W^W)_B \tag{51}$$

with the distance from the reference point R to the wing reference point W being $\mathbf{r}^{RW}$. The whole tail section of the aircraft is allowed to move, and hence acts as elevators. Similar to (45) the angle of attack of the tail α_t is calculated as follows:

$$\alpha_t = \text{atan2}\left(\frac{(w_A^R)_B^E - (\omega_y^{IB})_B \cdot (x^{Rt})_B}{(u_A^R)_B^E + (\omega_y^{IB})_B \cdot (z^{Rt})_B}\right) \tag{52}$$

and the transformation matrix from body axis B to tail axis t is determined as

$$\mathbf{M}_{tB} = \begin{bmatrix} \cos(\alpha_t) & -\sin(\alpha_t) \\ \sin(\alpha_t) & \cos(\alpha_t) \end{bmatrix} \tag{53}$$

The aerodynamic forces $(\mathbf{F}_t^t)_t$ for the tail at the tail reference point t in tail frame t are computed from the lift and drag forces L_t and D_t using the aerodynamic coefficients $C_{L\alpha,t}$ and $C_{D\alpha,t}$, the reference area S_t, the angle of attack at the tail α_t and the elevator deflection η:

$$L_t = \frac{1}{2} \cdot \rho V_A^{R^2} S C_{L\alpha,t}(\alpha_t + \eta) \tag{54}$$

$$D_t = \frac{1}{2} \cdot \rho V_A^{R^2} S C_{D\alpha,t}(\alpha_t + \eta) \tag{55}$$

$$(\mathbf{F}_t^t)_t = \begin{bmatrix} -D_t \\ -L_t \end{bmatrix}_t \tag{56}$$

The aerodynamic forces and moments from the tail t at reference point R in body frame B can then be determined as:

$$(\mathbf{F}_t^R)_B = \mathbf{M}_{tB}^T \cdot (\mathbf{F}_t^t)_t \tag{57}$$

$$(\mathbf{M}_t^R)_B = \mathbf{r}^{Rt} \times (\mathbf{F}_t^t)_B \tag{58}$$

with the distance from the reference point R to the tail reference point r being $\mathbf{r}^{Rt}$.

4.1.4 Total Forces and Moment

The forces and moments are summed together to obtain the total forces $(\mathbf{F}_T^R)_B$ and the moment $(M_T^R)_B$ in the body frame B and acting on the airframe at reference point R:

$$(\mathbf{F}_T^R)_B = (\mathbf{F}_P^R)_B + (\mathbf{F}_G^R)_B + (\mathbf{F}_W^R)_B + (\mathbf{F}_t^R)_B \tag{59}$$

$$(M_T^R)_B = (M_P^R)_B + (M_G^R)_B + (M_W^R)_B + (M_t^R)_B \tag{60}$$

4.1.5 Equations of Motion

The state derivatives are finally determined from following equations of motions with the aircraft mass m and the moment of inertia around the pitch axis I_{yy}:

- Position:

$$\begin{bmatrix} \dot{x} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ sin(\theta) & -cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} (u_K^R)_B^E \\ (w_K^R)_B^E \end{bmatrix} \tag{61}$$

- Attitude:

$$\dot{\theta} = (\omega_y^{IB})_B^B \tag{62}$$

- Translation:

$$\begin{bmatrix} (\dot{u}_K^R)_B^{EB} \\ (\dot{w}_K^R)_B^{EB} \end{bmatrix} = \begin{bmatrix} -(\omega_y^{IB})_B \cdot (w_K^R)_B^{EB} \\ (\omega_y^{IB})_B \cdot (u_K^R)_B^{EB} \end{bmatrix} + \frac{(\mathbf{F}_T^R)_B}{m} \tag{63}$$

- Rotation:

$$(\dot{\omega}_y^{IB})_B^B = (I_{yy})^{-1}(M_T^R)_B \tag{64}$$

Furthermore we consider PT2 transfer functions between the commanded values δ_c and the effector reactions δ

$$G(s) = \frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{65}$$

This is done for each of the control effectors, namely the elevator η, the tilt angles for the rotors at the wing tips δ_1 and δ_2, the two propellers at the left and right wing tips ω_1 and ω_2 and the propellers at the tail section ω_3 and ω_4. In order to introduce a smooth wind disturbance we additionally introduce a first order filter

$$(\dot{\mathbf{V}}_W^R)_B^{EB} = \frac{(\mathbf{V}_W^R)_{B,cmd}^E - (\mathbf{V}_W^R)_B^E}{\tau_w} \tag{66}$$

for the wind components of $(\mathbf{V}_W^R)_B^E$: the velocities $(u_W^R)_B^E$ and $(w_W^R)_B^E$ relative to the earth E, acting on the reference point R and denoted in the body fixed frame B.

The resulting state space model for the plant thus consists of 22 states comprised of the rigid body equations of motions, the second order actuator transfer functions and the first order filters for the wind disturbance.

4.2 Incremental NDI Control Structure

4.2.1 Governing Equations

The baseline control law implements the incremental nonlinear dynamic inversion technique [15]. Consider an aircraft dynamic model,

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u}) \tag{67}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{68}$$

using Taylor expansion,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \tag{69}$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \left.\frac{\partial [\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u})]}{\partial \mathbf{x}}\right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \tag{70}$$

and differentiating equation (68):

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \dot{\mathbf{x}} \tag{71}$$

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \dot{\mathbf{x}}_0 + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left.\frac{\partial [\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u})]}{\partial \mathbf{x}}\right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \tag{72}$$

yields:

$$\dot{\mathbf{y}} - \dot{\mathbf{y}}_0 = \delta \dot{\mathbf{y}} = \mathbf{A}(\mathbf{x}_0, \mathbf{u}_0) \delta \mathbf{x} + \mathbf{B}(\mathbf{x}_0, \mathbf{u}_0) \delta \mathbf{u} \tag{73}$$

The control effectors directly controls accelerations, while the states are integrations of accelerations. For high control frequencies, the increments of states are much smaller than the increments of inputs. Therefore, Eq. (73) can be further simplified by eliminating the term $\mathbf{A}(\mathbf{x}_0, \mathbf{u}_0) \delta \mathbf{x}$:

$$\delta \dot{\mathbf{y}} = \mathbf{B}(\mathbf{x}_0, \mathbf{u}_0) \delta \mathbf{u} \tag{74}$$

4.2.2 Controller Structure

The pilot commands velocities in x_O and z_O direction of the NED frame, and the pitch angle θ. A linear reference model is designed to force the controlled variables to follow a reference trajectory. A relative degree 1 reference model is implemented for the velocities, and a relative degree 2 reference model is used for the pitch angle. Furthermore error controllers are implemented to account for model uncertainties. The reference model and error controller states add six additional states to the 22 plant states, which results in a total number of 28 states for the closed loop system. The on-board plant model, which is used to determine the control effectiveness B consists of the complete set of aerodynamic, propulsion and the gravitation model from Sect. 4.1. Current state accelerations are estimated using equations of motion in this on-board plant model and the estimated reaction $\boldsymbol{\nu}_{est}$ and the control effectiveness matrix $\mathbf{B}$ are given as outputs. The desired pseudo-command $\Delta\boldsymbol{\nu}_{des}$ is then calculated as

$$\Delta\boldsymbol{\nu}_{des} = \boldsymbol{\nu}_{ref} - \boldsymbol{\nu}_{est} \tag{75}$$

In this preliminary case study the required control effector deflection $\Delta\mathbf{u}_{cmd}$ is determined using weighted-pseudo inverse with weighting matrix Λ_u:

$$\Delta\mathbf{u}_{cmd} = \Lambda_u(\mathbf{B}\Lambda_u)^{+}\Delta\boldsymbol{\nu}_{des} \tag{76}$$

Finally $\Delta\mathbf{u}_{cmd}$ is added to $\mathbf{u}_{i-1}$ from the previous time step, and subsequently fed to the plant.

4.3 Problem Formulation and Results

4.3.1 General Problem Formulation

The problem we solve using the proposed workflow from Sect. 2, is to test for height loss during the transition phase from hovering mode to forward flight. The cost function J to be minimized is the maximum height loss over the time horizon $t \in [t_0, t_f]$:

$$\min_{\mathbf{z}} J(\mathbf{z}) = \min_{\mathbf{z}} \min_{t\in[t_0,t_f]} h(t) \tag{77}$$

The simulation is performed by Euler's method using a step size of $\Delta t = 0.001\,\mathrm{s}$ with an initial time of $t_f = 0\,\mathrm{s}$ and a final time of $t_f = 10\,\mathrm{s}$. This time span is sufficient to allow the vehicle to perform the complete transition. Additionally, we set the initial condition for hover flight with zero velocities in 10 m height. Throughout the optimization, the commanded velocity u_{cmd} is $5\frac{\mathrm{m}}{\mathrm{s}}$ and the other control inputs are zero:

$$\begin{pmatrix} u_{cmd} \\ w_{cmd} \\ \theta_{cmd} \end{pmatrix} = \begin{pmatrix} 5\frac{\mathrm{m}}{\mathrm{s}} \\ 0 \\ 0 \end{pmatrix} \tag{78}$$

4.3.2 Step 1: Latin Hypercube Sampling

In the first step, *Latin Hypercube Sampling* is performed with 250 samples using the MATLAB function *lhsdesign()*. For the wind gust parametrization we introduce the timepoints for the wind commands T_u and T_w and the command values for the wind gusts $u_{W,cmd}$ and $w_{W,cmd}$

$$t_0 \leq T_u \leq t_f \tag{79}$$

$$t_0 \leq T_w \leq t_f \tag{80}$$

$$u_{W,cmd,lb} \leq u_{W,cmd} \leq u_{W,cmd,ub} \tag{81}$$

$$w_{W,cmd,lb} \leq w_{W,cmd} \leq w_{W,cmd,ub} \tag{82}$$

with $u_{W,cmd,lb} = w_{W,cmd,lb} = -2\frac{\mathrm{m}}{\mathrm{s}}$ and $u_{W,cmd,ub} = w_{W,cmd,ub} = +2\frac{\mathrm{m}}{\mathrm{s}}$ and the time constant $\tau_W = 3\,\mathrm{s}$ for the filter in (66).

Additionally, we add two parameters for the uncertainties in mass Δm and moment of inertia ΔI_{yy}:

$$\Delta m_{min} \leq \Delta m \leq \Delta m_{max} \tag{83}$$

$$\Delta I_{yy_{min}} \leq \Delta I_{yy} \leq \Delta I_{yy_{max}} \tag{84}$$

These parameters have the bounds $\Delta m_{min} = \Delta I_{yy_{min}} = 90\%$ and $\Delta m_{max} = \Delta I_{yy_{max}} = 110\%$ and are used to pertube the nominal values $I_{yy_{nom}}$ and m_{nom} as follows:

$$m = \Delta m \cdot m_{nom} \tag{85}$$

$$I_{yy} = \Delta I_{yy} \cdot I_{yy_{nom}} \tag{86}$$

Please note, that the bounds of the wind disturbance and parameter uncertainties are chosen very carefully here as the control structure is still under development.

4.3.3 Step 2: Differential Evolution

The following global optimization is initialized by the 25 worst cases found by LHS. In case of *Differential Evolution*, we now use five parameters for each wind gust command $u_{cmd,1,\ldots5}$, $w_{cmd,1,\ldots5}$ on a fixed equidistant grid for the control inputs. To translate the step inputs from LHS with step times T_u and T_w on the DE grid, we take the closest value on the equidistant time grid (see Fig. 3).

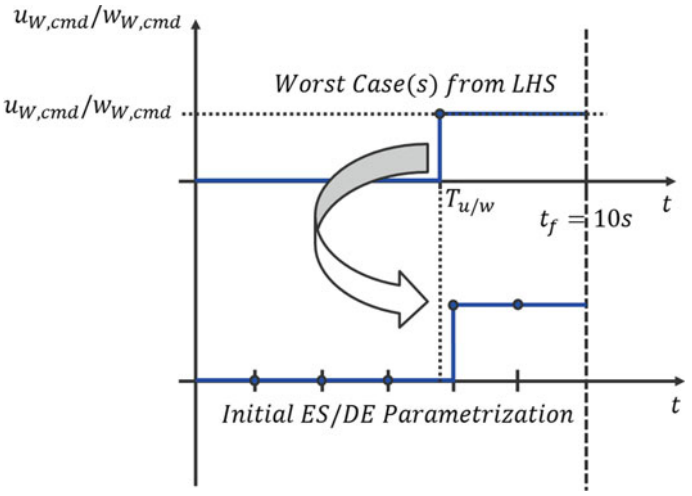

Fig. 3 Translation of the worst case wind command input from LHS to DE

For the classic DE algorithm "*DE/rand/1/bin*" implemented in the solver *DeMat* [14], we set the population size to $N_p = 25$, the weighting factor to $F = 0.8$, the crossover probability to $p_c = 0.7$ and the maximum number of generations to $g_{max} = 50$.

4.3.4 Step 3: Optimal Control

The initial guess for the subsequent *Optimal Control* (OC) problem is the best solution from the DE optimization. In the example presented here the minimum height found from global optimization was at the final time t_f. The cost function is now altered to maximize the height loss at this particular time point.

Now we refine the discretization even further using 40 discretization points for each of the wind gust commands u_{cmd} and w_{cmd}. For this particular problem we found it sufficient to use single shooting due to the short time frame of the optimization. For other problems, especially those who exhibit highly sensitive parameters or long time intervals it may be required to introduce additional shooting nodes to lower the sensitivities and thus enhance convergence.

The solver used here is the SQP-solver SNOPT [9] which requires the gradient of the problem. We compute the gradient by the complex step method as described in [11]. This method exploits the fact that using a complex step $x + ih$ for a function $f(z)$ with sufficiently small step size h we can approximate the derivative of this function up to machine precision by dividing the imaginary part of $f(z + ih)$ and by h. For the scalar case the equation reads:

Table 1 Summary of worst cases

Step	Height loss (m)	Δm (%)	ΔI_{yy} (%)
Step 1: LHS	0.42	106.45	93.47
Step 2: DE	0.46	109.80	96.56
Step 3: OC	0.58	110.00	90.00

$$\frac{\partial f}{\partial z} \approx \frac{\Im[f(z+ih)]}{h} \tag{87}$$

On the one hand, the main drawbacks of this method are the fact that the complex evaluation is computationally more expensive and that some of the functions, such as $<, >$, *min()*, *max()* and *abs()* to name a few have to be redefined in order to evaluate the model properly. On the other hand the derivatives are extremely accurate (in the order of the machine precision) which reduces the "numerical noise" introduced from finite difference approximations. Additionally, in MATLAB and SIMULINK most of the built-in functions can be used both for complex and real algebra such that the additional modeling requirements are greatly reduced.

4.3.5 Step 4: Postoptimal Sensitivities

For the *Postoptimal Sensitivities* we need to compute the Hessian of the Lagrangian (see (31)) with respect to the optimal parameter vector $\mathbf{z}_{opt}$. Similarly we again use the complex step method from [10] to compute the second derivatives.

The first and second order sensitivities $\frac{dJ}{d\boldsymbol{\omega}}$ and $\frac{d^2J}{d\boldsymbol{\omega}^2}$ of the cost function (77) w.r.t. $\boldsymbol{\omega} = [\Delta m, \Delta I_{yy}]^T$ are computed using (35) and (36) which allows us to approximate the change in the cost function ΔJ (worst case) when changing the parameters by $\Delta\boldsymbol{\omega}$:

$$\Delta J = \frac{dJ}{d\boldsymbol{\omega}} \cdot \Delta\boldsymbol{\omega} + \Delta\boldsymbol{\omega}^T \cdot \frac{d^2J}{d\boldsymbol{\omega}^2} \cdot \Delta\boldsymbol{\omega} \tag{88}$$

4.3.6 Results

The comparison between the baseline height history without uncertainties and the height trajectories for the different steps (LHS, DE, OC) for the transition phase is depicted in Fig. 4.

The worst case height loss and the corresponding parameter combinations are summarized in Table 1.

The *Postoptimal Sensitivities* for $\boldsymbol{\omega} = [\Delta m, \Delta I_{yy}]^T$ in this example revealed that for the worst case the uncertainty in the mass Δm has significant influence on the optimal solution, but the uncertainty in the moment of inertia ΔI_{yy} is almost neglectable.

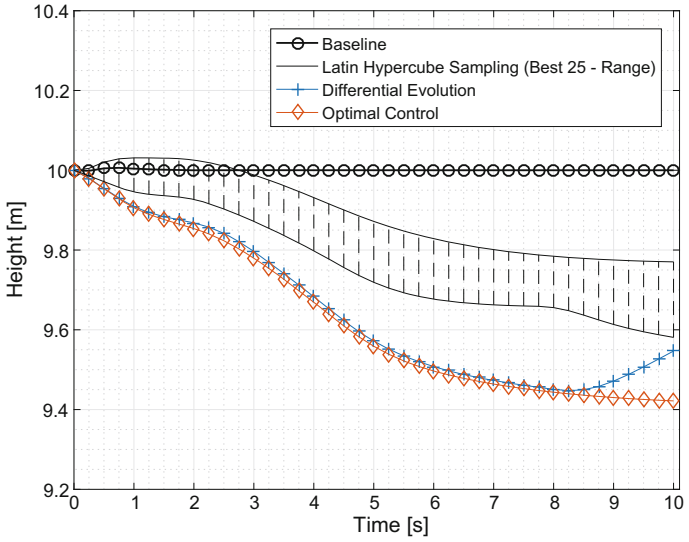

Fig. 4 Comparison of the height loss during transition from hover-mode to forward-flight

$$\frac{dJ}{d\boldsymbol{\omega}} = [-0.98, 6.52 \cdot 10^{-6}]^T \tag{89}$$

$$\frac{d^2J}{d\boldsymbol{\omega}^2} = \begin{bmatrix} -0.34 & -2.43 \cdot 10^{-5} \\ -2.43 \cdot 10^{-5} & -4.76 \cdot 10^{-7} \end{bmatrix} \tag{90}$$

This can be explained by the fact that the influence of I_{yy} is mostly seen in dynamic maneuvers where the attitude of the vehicle changes. In the example presented here, the controller tries to keep the attitude angle at zero as required by $\theta_{cmd} = 0$ and only accelerates by tilting the rotors at the wings forward. This means that for the most part of the transition phase the attitude remains approximately constant zero which results in an acceleration maneuver without attitude change. For this maneuver I_{yy} has almost no influence whereas the uncertainty in mass significantly influences the acceleration and height loss for the worst case found.

Furthermore, as the parameters are subject to optimization we would expect a zero first order sensitivity $\frac{dJ}{d\boldsymbol{\omega}}$ at the optimal point, which is not necessarily the case for the second order sensitivity $\frac{d^2J}{d\boldsymbol{\omega}^2}$. In this specific case study the parameter values converged to the boundary of the admissible parameter set. Therefore, the first order sensitivity $\frac{dJ}{d\boldsymbol{\omega}}$ is non-zero and provides the sensitivity of the cost function value with respect to a change in the boundary value.

5 Conclusions

In the present paper, we investigated a novel workflow for the clearance of flight control laws with continuous control or disturbance inputs using *Optimal Control* Theory and *Postoptimal Sensitivity Analysis*. For this methodology, sampling and a global optimization method are used to obtain good initial guesses for the application of direct *Optimal Control* Methods. Additionally, first and second order approximations for the objective function (clearance criteria) were revised which can be derived from *Postoptimal Sensitivity Analysis*. The workflow is illustrated in a case study for the preliminary design of an INDI controller for a tilt rotor system. In the example presented here the maximum height loss during the transition phase of this tilt-rotor system with respect to uncertainties in mass, moment of inertia and wind gusts was investigated. The results gave meaningful insights concerning the robustness of the controller and allowed to determine the sensitivities of the solution with respect to the uncertainties. Future work will be devoted towards the extension for more realistic models and additional clearance criteria.

Acknowledgements The authors wish to thank Matthias Bittner and Patrick Piprek for the helpful discussions and useful comments. This work was partly supported by the DFG grant HO4190/8-1.

References

1. Ben-Asher JZ (2010) Optimal control theory with aerospace applications. AIAA education series, American Institute of Aeronautics and Astronautics
2. Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming. Advances in design and control. Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), 2nd edn. Philadelphia
3. Bryson AE, Ho YC (1975) Applied optimal control: optimization, estimation, and control. Taylor and Francis, New York
4. Büskens C (1998) Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustands-Beschränkungen: Münster (Westfalen), Univ., Diss
5. Diepolder J, Gabrys A, Schatz S, Bittner M, Grüter B, Holzapfel F, Ben-Asher JZ (eds) (2016) Flight control law clearance using worst-case inputs (To be published)
6. Fernandes de Oliveira R, Puyou G (2011) On the use of optimization for flight control laws clearance: a practical approach. 18th IFAC World Congress 44(1):9881–9886
7. Fiacco AV (1983) Introduction to sensitivity and stability analysis in nonlinear programming, vol 165. Mathematics in science and engineering. New York, Academic Press
8. Forssell LS, Åke Hydén (eds) (2003) Flight control system validation using global nonlinear optimisation algorithms
9. Gill PE, Murray W, Saunders MA (2005) SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev 47(1):99–131
10. Lai KL (2007) Extensions of the first and second complex-step derivative approximations. J Comput Appl Math
11. Martins JRRA (2003) The complex-step derivative approximation. ACM Trans Math Softw 29(3):245–262
12. Menon PP, Bates DG, Postlethwaite I, Marcos A, Fern V, Ez Bennani S Worst-case analysis of flight control laws for re-entry vehicles

13. Menon PP, Kim J, Bates DG, Postlethwaite I (2006) Clearance of nonlinear flight control laws using hybrid evolutionary optimization. IEEE Trans Evol Comput 10(6):689–699
14. Price KV, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Natural computing series. Springer, Berlin
15. Sieberling S, Chu QP, Mulder JA (2010) Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. J Guid Control Dyn 33(6):1732–1742
16. Varga A, Hansson A, Puyou G (2012) Optimization based clearance of flight control laws: a civil aircraft application. Lecture notes in control and information sciences. Springer, Berlin

http://www.springer.com/978-3-319-65282-5

Advances in Aerospace Guidance, Navigation and Control
Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, April 2017
Dołęga, B.; Głębocki, R.; Kordos, D.; Żugaj, M. (Eds.)
2018, XII, 751 p. 392 illus., 292 illus. in color.,
Hardcover
ISBN: 978-3-319-65282-5