PSEUDO-INVERSE SIMULATION OF PULL-UP MANEUVERS AT LOW AND HIGH SPEEDS BY MEANS OF OPTIMIZATION

Jakob Thiemeier[1]	jakob.thiemeier@koptergroup.com
Ferdinando Montemari[1]	ferdinando.montemari@koptergroup.com
Francesco Fresta[1]	francesco.fresta@koptergroup.com
Luca Litterst[2]	luca.litterst@koptergroup.de
Christian Spieß[2]	christian.spiess@koptergroup.de

[1]Kopter Group AG, Wetzikon ZH, Switzerland
[2]Kopter Germany GmbH, Munich, Germany

Abstract

In the course of the certification process of a helicopter, applying EASA certification specifications and means of compliance, a loads survey is required which is then used for the design process or the stresses verification. Reasonable load cases can be derived from typical maneuvers, e.g. pull-ups, characterized by a phase of positive pitch rate and increased load factor. The paper at hand presents how an optimization of pull-up maneuvers simulated with the flight mechanics code Flightlab can be utilized to acquire desired load cases. The maneuvers are "flown" by prescribing control inputs for longitudinal cyclic and collective, whereas lateral cyclic and pedal a handled by a control system. The simplified input signals for longitudinal and collective controls are parameterized and then altered by an optimization framework, called Maneuver Optimization Tool (MOPT) and written in Python, in order to achieve given maneuver characteristics, e.g. a target pitch rate and main rotor torque. The simulation framework, consisting of Flightlab and the mentioned Python optimization, is introduced, and results are shown for low-speed flight at $v_l = 70\,\mathrm{kts}$ indicated and at never-exceed speed v_{NE}. In addition, four different combinations of low/high helicopter mass, and front/aft center of gravity location are considered.

Nondisclosure Notes

In order to protect intellectual property owned by Kopter, results are generally masked by removing the scales. However, this does not prevent the proper evaluation of the results which is rather focused on the optimization than on the helicopter itself.

Nomenclature

a	Linear penalty gain factor
$d_{1,2,3}$	Step height of control signal
H	Altitude
m	Mass
N_z	Vertical load factor
p	Constraint penalty
Q_{MR}	Main rotor torque
t	Time
$t_{1,2,3}$	Duration of step signals $d_{1,2,3}$
t_r	Duration of ramp times
t_{tot}	Total duration of maneuver
v_{IAS}	Indicated air speed
v_l	Flight speed (low)
v_{NE}	Never-exceed speed
x	Long. coordinate, positive back
z	Vertical coordinate, positive up
δ_{Ex}	Exceedance of constraint
Θ	Airframe pitch attitude
$\dot{\Theta}$	Airframe pitch rate
$\ddot{\Theta}$	Airframe pitch acceleration
Θ_0	Collective control angle
Θ_C	Lateral cyclic angle
Θ_S	Longitudinal cyclic angle
Θ_{TR}	Tail rotor collective angle
Φ	Airframe roll attitude
Ψ	Airframe yaw attitude

CG	Center of gravity
DOF	Degree of freedom
ISA	International Standard Atmosphere
MCP	Maximum continuous power
MOPT	Maneuver Optimization Tool

1 INTRODUCTION

In the process of getting to the certification of a rotorcraft, reasonable approximations for sizing and fatigue loads have to be found. This is done in close accordance to the EASA certification specifications and acceptable means of compliance. In the case of the AW09 helicopter, currently under development at Kopter, EASA CS-27 is applicable. Reasonable load cases can be derived by modelling typical maneuvers, e.g. a pull-up maneuver, characterized by an increased pitch rate and load factor. In addition, the torque and thrust of the main rotor will change along the maneuver. The resulting rotor and airframe loads can then be passed to either the design process or the stresses verification of already existing structures.

As a fast means for helicopter simulations, comprehensive analysis codes are widely used across the helicopter industry. They provide sufficiently good results in a huge range of applications, combining aerodynamics, structural dynamics and flight mechanics on a lower level of fidelity compared to full Finite Element Methods or Computational Fluid Dynamics, but in considerably shorter time.

Simulating a transient maneuver to provide realistic load conditions has an inherently high level of complexity as there is involved a high number of unknown variables. The time histories of the control inputs which achieve the desired maneuver specifications is not known a priori, and neither is the exact flight trajectory. Searching by trial and error for a set of control inputs that lead to a maneuver satisfying certain target values can be both boring and time-consuming. Therefore, an automated approach potentially increases efficiency in terms of time and accuracy of the resulting maneuver.

In the paper at hand, a framework consisting of the comprehensive analysis code Flightlab and a Python-based optimization is introduced. This framework performs a pseudo-inverse simulation which aims at deriving proper control inputs for given maneuver specifications. The term *pseudo-inverse simulation* is used since the original inverse simulation derives the control input history from direct differentiation or integration of a given flight trajectory known a priori, see Thomson[1, 2]. In contrast, the approach at hand optimizes a simplified control input signal in order to achieve a limited number of maneuver criteria without caring about an exact trajectory. Therefore, this procedure is considered rather curve fitting by optimization than real inverse simulation, although it aims at a comparable goal. This also applies to the approaches used by Kalkan and Tosun[3, 4] who also used optimized control inputs to generate a desired maneuver. In addition, Guglieri and Mariano[5] utilized a genetic optimization in order to match given maneuver trajectories closely.

The emphasis of this introduction is on how to simplify this optimization problem in order to integrate it reasonably into the work flow of the load assessment. It aims at finding an approach that can be applied to a preferably large range of parameters, e.g. helicopter mass, center of gravity (CG), flight speeds and atmospheric conditions.

2 COMPUTATIONAL SETUP

The computational setup consists of a Python framework which Flightlab is embedded in. On the one hand, Flightlab provides trim conditions and transient maneuver simulations, and on the other hand, the Python framework changes the Flightlab configuration files, controls the Flightlab runs, organizes the results, and runs the optimizer on the available variables. After providing a working Flightlab setup which must already feature variables appropriate for the optimization, the entire optimization is controlled by a single user input file processed by the Python framework.

2.1 Maneuver Optimization Tool (MOPT)

An optimization is employed in order to find a set of values for a given number of variables defining a simplified time history of selected control inputs. The optimization is driven by given objectives by which a maneuver can be characterized, and constraints that restrict the scope of possible maneuvers.

The framework is realized by a Python script that handles all necessary tasks: (1) Modify the Flightlab configuration files; (2) Run Flightlab; (3) Organize the results of the single Flightlab

runs; (4) Evaluate and plot the results; (5) Run an optimizer based on the Flightlab results.

The program logic is shown in Figure 1. An initial trim is performed prior to each maneuver optimization. This trim serves as initial condition for the subsequent transient maneuver simulation of each optimization iteration. Inside the indicated optimization loop, the available variables of the transient maneuver are altered by the optimizer, and the maneuver is run multiple times. The optimization itself is defined by objectives that refer to certain maneuver characteristics, e.g. a certain pitch rate, and by variables that alter the maneuver, e.g. control inputs. Furthermore, constraints may be defined, concerning maneuver characteristics that must not be exceeded. The optimization loop is executed until the altered maneuver satisfies the optimization objectives. Eventually, the final maneuver is stored and evaluated automatically.

The optimization loop employs different optimization methods provided by the Python package SciPy. Thanks to the modular structure of the Maneuver Optimization Tool, the optimization method can be swapped easily or multiple methods can be employed in a series.

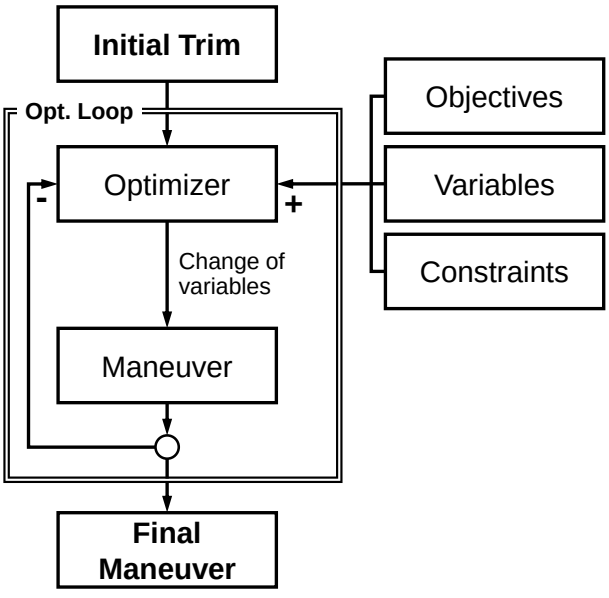

Figure 1: Program logic of implemented optimization framework.

The optimization can be applied to any maneuver as long as necessary control variables are defined that can be used to generate a desired maneuver.

2.2 Flightlab Helicopter Model, Trim and Transient Maneuver

The underlying Flightlab model was built by Kopter and represents the current prototype of the AW09 helicopter. The model features five main rotor blades and a ten-bladed shrouded tail rotor, both treated rigidly. The aerodynamic representation of the airframe consists of fuselage, skids, tail boom, horizontal and vertical stabilizer surfaces.

Four different helicopter configurations were analyzed in terms of mass and longitudinal CG position, see Figure 2. The configurations selected roughly build the corners of the weight-CG envelope of the helicopter. They feature low mass/front CG, low mass/aft CG, high mass/aft CG, and high mass/front CG.

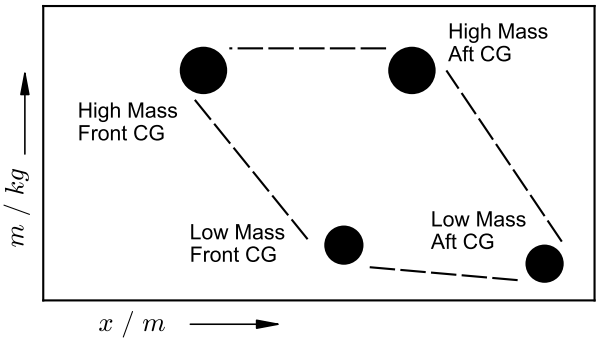

Figure 2: Masses and center of gravity longitudinal location. Size of bullets proportional to mass. Weight-CG envelope roughly indicated.

For the optimization, Flightlab provides both an initial trim and transient maneuvers over a period of several seconds. The trim is derived using six degrees of freedom (DOFs), namely Θ_0, Θ_C, Θ_S, $\Theta_{0,TR}$, Θ and Φ, in order to achieve zero translatory and rotatory accelerations. In case the system power required exceeds the maximum continuous power (MCP), the sink rate is added to the set of DOFs, whereas the MCP is made a trim objective. This leads to a 7-DOF trim and to a descent at MCP.

The transient maneuver always starts from a corresponding trim state that is not disturbed for an initial second before a time history of control inputs is imposed, comparable to an actual pilot. In case of a longitudinal maneuver like a pull-up, longitudinal cyclic Θ_S and collective Θ_0 are prescribed, whereas lateral cyclic Θ_C and pedal Θ_{TR} are automatically operated by a controller system that was integrated into the Flightlab model by Kopter Germany. It is designed to keep roll attitude and sideslip angle at bay.

In order to accommodate all chosen control inputs, an overall maneuver duration of $t_{tot} = 4\,\mathrm{s}$ is used.

The simulations are conducted with the atmospheric conditions at sea level ISA.

3 OPTIMIZATION SETUP

An optimization approach is used in order to find suitable parameters for a parameterized time history of longitudinal and collective control input. Simplifying and parameterizing the control input reduces complexity and, therefore, run time of the optimization. It aims at generating a maneuver that features a period of increased positive load factor during a phase of pulling up, see Figure 3. In addition, the load condition during this pull-up phase is supposed to get as close as possible to conditions that can be used for sizing purposes. On the other hand, certain maximum values must not be exceeded and serve as constraints for the optimization.

Therefore, an optimization used for the approach at hand is defined by the variables that may be altered, the objectives that shall be achieved and the constraints that must not be exceeded. The complexity of the optimization is basically determined by the number of variables, whereas the number of objectives and constraints affect how flexible the optimizer may search for an optimal solution.

Objectives and constraints may be adapted to a specific problem in question. They are monitored either over the entire duration of the maneuver or only for selected periods. However, an objective is considered satisfied if it is achieved for at least one single time step of the monitored period.

3.1 Maneuver Definition Including a Pull-Up Phase

In the literature, multiple terms for maneuvers exist, all of which feature a phase of pulling up. Prasad[6] mentions a distinction between pull-up, pop-up and hurdle-hop maneuvers in the course of his introduction of an inverse simulation of maneuvers flown by a UH-60A and a Westland Lynx. The respective initial trajectories of all mentioned maneuvers are very similar, though, compare Figure 3. However, for the present paper, the highlighted concept of a pull-up maneuver is used, whereas the exact trajectory is not known a priori.

General criterion for such maneuvers is a limitation of the flight trajectory to the vertical plane, ensuring a mainly longitudinal character of the maneuver and preventing a significant change in roll and sideslip angles.

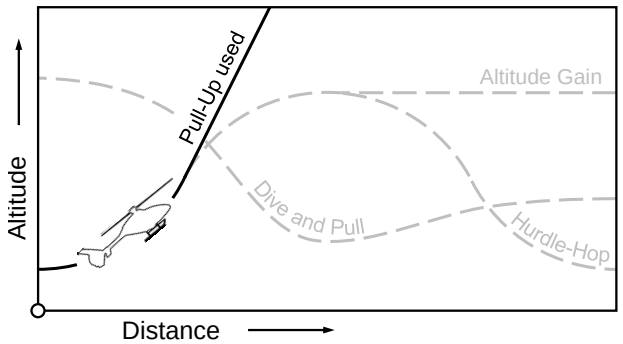

Figure 3: Sketched trajectories of maneuvers featuring a pull-up phase.

3.2 Variables, Objectives and Constraints of the Optimization

In the case of the pull-up maneuver, the time history of longitudinal and collective control input are defined by three parameters $d_{1,2,3}$ each of which define the height of a step signal, see Figure 4. The duration of the step signals are predefined with $t_1 = t_2 = 1.1\,\mathrm{s}$ and $t_3 = 1.75\,\mathrm{s}$, while the ramp-up and down times are kept constant at $t_r = 0.2\,\mathrm{s}$. These durations are the re-

sult of a small preliminary analysis which aimed at getting the most general approach, applicable to a wide range of flight conditions and weight-CG configurations.

Step signal d_1 refers to the initial pull on the stick, initiating the pitch-up of the helicopter's nose. Step signal d_2 is dedicated to a pull on the collective in order to drive the main rotor torque Q, whereas step signal d_3 is a second step signal on the longitudinal control, but pushing on the stick, in order to reduce the high pitch attitude after the actual pull-up phase. This is done to ensure that the helicopter may reach an uncritical attitude afterwards, since a resulting maneuver that could not be recovered would not be relevant for certification.

3.3 Approach Used: Serial Optimization

Optimizing all three input variables at once would be straightforward and technically feasible. However, this approach was successful only in a very few low-speed flight cases. For more information on the failure of that approach see Section 3.5. The most effective approach, for both high and low speeds in the present investigation, does not optimize all variables at once, but each variable in an own stage. As a consequence of this serial optimization, each stage is less complex and can be controlled more specifically. The subsequent stages are built upon the optimization results of previous stages. Figure 4 exemplarily shows the evolution of the control step signals during three consecutive stages, indicating the differences in control values compared to the trim values. Starting from the trim values ($\Delta\Theta_S = \Delta\Theta_0 = 0$), Stage 1 optimizes the step signal d_1 only, which refers to the initial pull on the stick. The result for that first optimization is assumed and not changed anymore for the following stages. This procedure is also applied to the step signals d_2 and d_3. Stage 3 eventually shows the final result of the entire optimization procedure.

Since the entire optimization is split in stages, each stage can be controlled separately. As mentioned earlier, an optimization is defined by variables, objectives and constraints. Therefore, each single stage can have its own objectives and constraints. Table 1 gives an overview of the optimization definitions for each stage.

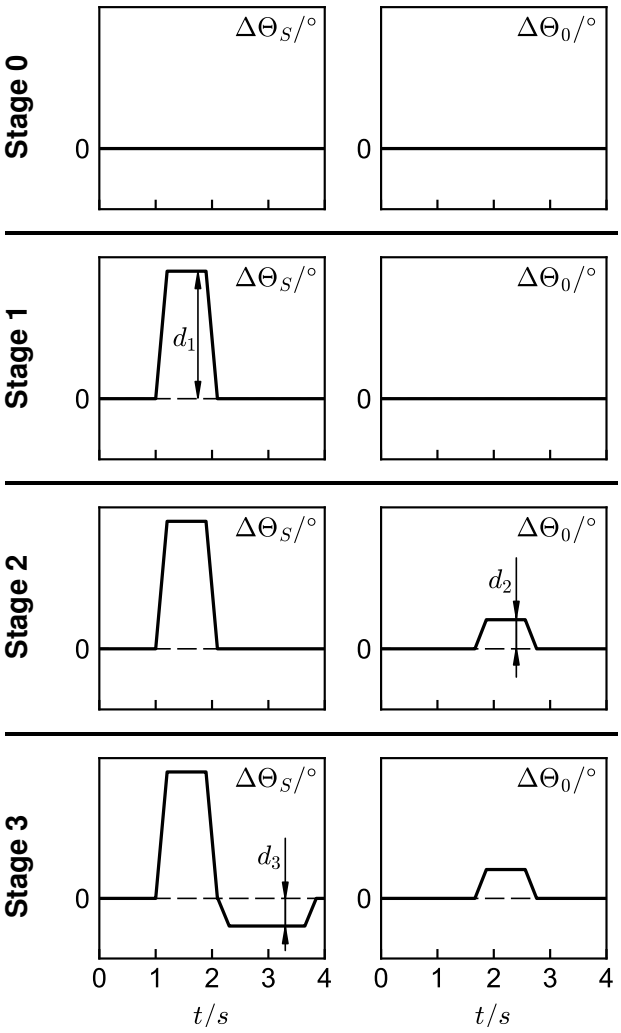

Figure 4: Serial optimization approach with consecutive stages. Deltas of control values $\Delta\Theta_S$ and $\Delta\Theta_0$ refer to the respective trim values. Same scale in all graphs.

Opt. Stage	Variable	Objective	Constraints
1	d_1	$\dot{\Theta}$	$\dot{\Theta}$, $N_z - \ddot{\Theta}$, Q_{max}
2	d_2	Q_{max}	
3	d_3	Θ	

Table 1: Variables, objectives and constraints of the optimization approach.

Stage 1 aims at reaching a given pitch rate $\dot{\Theta}$, using the step height d_1 of the longitudinal command, as the pitch rate mainly determines the aggressiveness of the maneuver. Three con-

straints must be respected. First, the pitch rate $\dot{\Theta}$ is not only an objective, but also a constraint. As the values for the pitch rate objective and constraint are equal in the present case, the optimizer shall reach the given pitch rate, but must not exceed it. In addition, the objective can be defined either for the entire duration of the maneuver or for a specific period of time. For the present optimization, a target period of $1.8\,\mathrm{s} \leq t_{\dot{\Theta}} \leq 2.2\,\mathrm{s}$ was found to give good results.

Second, an $N_z - \ddot{\Theta}$ envelope is defined, determined by load factor and pitch acceleration. This envelope considers the combined effect of linear and angular accelerations on the loads helicopter parts are exposed to during a maneuver. Being set as a constraint, the $N_z - \ddot{\Theta}$ envelope shall contain the entire maneuver.

Third, the maximum main rotor torque Q_{max} is used as a constraint. While every stage features its own dedicated objective, the three constraints mentioned above apply to each of the three stages.

Stage 2 aims at reaching the maximum main rotor torque Q_{max} by employing a collective command with the step height d_2 in order to generate a demanding maneuver in terms of tail rotor thrust and tail boom bending moment. The Q_{max} objective is defined for a target time of $2.0\,\mathrm{s} \leq t_Q \leq 3.0\,\mathrm{s}$.

Stage 3 is applied in order to recover a reasonable pitch attitude after the actual pull-up phase. Therefore, a low pitch attitude Θ close to horizontal is set as objective during a target period of $3.9\,\mathrm{s} \leq t_{\Theta} \leq 3.95\,\mathrm{s}$ at the very end of the maneuver duration.

Clearly, such a detailed definition of single optimization problems requires already some specific knowledge of the maneuver. However, this knowledge has to be gained only once per maneuver type, and can then be applied in a more universal way to parameter variations of single flight cases.

Also, the number and order of step signals and involved controls currently chosen do not represent an ultimate solution. These properties may be adapted if requirements of the maneuver change or entirely different maneuvers shall be computed.

3.4 Constraint Definition

Constraints shall not be exceeded for an optimized maneuver. Therefore, a simple but flexible constraint implementation was integrated into the MOPT. The exceedance of a constraint shall cause an immense cost for the optimizer which makes a solution less optimal. This is realized by computing a quadratic penalty for the exceedance that is then added to the residual of the optimization sample in question. The penalty p is computed by a linear gain factor a and the exceedance δ_{Ex} as follows:

$$p = a \cdot \delta_{Ex}^2 \tag{1}$$

The gain factor can be set for each constraint individually via the user input. In case a quadratic penalty was not aggressive enough, the power of 2 could be increased very easily in the code.

3.5 Challenges for the Optimization

One of the biggest challenges for the optimization seems to be the highly non-linear character of the problem as the helicopter is considered as a whole system. Control inputs at the very beginning of a maneuver may cause strong perturbations at later points in time that are not proportional to the inputs which is especially a problem when multiple variables are optimized at the same time. Therefore, gradient-based optimization methods seem not to be suitable and were already tested, revealing difficulties due to their local quality. In contrast, global optimization methods show a better reliability in terms of finding a good solution, but suffer from a higher computational effort and are still struggling with the numerous local optima, obscuring the global optimum.

The present optimization approach is meant to be applied to a quite huge test matrix in terms of helicopter masses, CG locations, flight speeds, and atmospheric conditions. Ideally, one optimization setup should cover all relevant test conditions without manual effort required. However, this is a challenge since the response of the helicopter during a maneuver can significantly change depending on the test conditions. For example, the high-speed regime is much trickier to handle than the low-speed regime since small input changes have a much

bigger impact and non-linear behavior occurs much faster.

3.6 Optimization Strategies Tested

In order to overcome the challenges mentioned above, different optimization strategies were tested, before choosing the approach introduced in Section 3.3. This includes different control input patterns, different optimization methods and splitting the optimization of multiple variables into separate stages (serial optimization).

3.6.1 Control Input Pattern

Different patterns of control input signals were tested. The number of longitudinal step signals was varied, starting from a single one (just pulling once), and increasing to two (pull and recover) and three (push - pull - recover). An initial push on the stick can lead to an advantage in terms of pitch attitude margin as the pull-up phase starts from a lower pitch attitude. For example, a higher pitch acceleration $\ddot{\Theta}$ can develop, if desired, before a critical pitch angle Θ is reached.

Eventually each additional variable increases complexity and, therefore, two step signals seem to be a reasonable compromise.

3.6.2 Optimization Method

Local (gradient-based) and global optimizer algorithms were investigated. While local methods seem suitable for only a very limited range of test conditions due to their local quality, global methods are more promising, but require a higher computational effort and may not be as accurate as gradient-based methods. Currently, the most promising approach is nesting global and local methods. The result of an initial global search can then be refined by applying a local method, using the result of the global search as starting point.

The Python package SciPy features various implementations of both global and local methods. They can be replaced easily inside the Python framework used for the paper at hand since it was written to provide a simple interface for the optimizer, allowing a quick change. Currently, the global *Dual Annealing* method in combination with the *Least Squares* method is used.

3.6.3 Serial Optimization

Besides a full optimization of all given variables at the same time, also a serial optimization approach may be reasonable. This serial approach was used for the present paper. Its basic idea is the division of the optimization into consecutive stages of chronological order, see Section 3.3. This makes use of the fact that single stages show a less complex optimization behavior, and that control inputs only have a very limited impact on maneuver periods prior the actual input. As a consequence, earlier stages are almost not affected by later ones which offers the opportunity to build a desired maneuver chronologically period by period.

4 OPTIMIZATION RESULTS

In order to demonstrate the functionality of the Maneuver Optimization Tool, this section shows the results for a total of eight different pull-up cases. Four different combinations of low/high mass and front/aft center of gravity location are computed for a lower speed of $v_l = 70\,\mathrm{kts}$ indicated and for never-exceed speed v_{NE}. These eight cases can be considered to be located on the edges of the relevant parameter space. Therefore, these cases are assumed to be the most critical ones. However, a variation of atmospheric conditions is neglected.

4.1 Objectives

For the pull-up maneuver at hand, the main objectives are the pitch rate $\dot{\Theta}$, determining the aggressiveness of the maneuver, and the maximum main rotor torque Q_{max}, enhancing the maneuver's severity.

Figure 5 shows the evolution of the pitch rate $\dot{\Theta}$ over the entire duration of the maneuver. The first second does not show a change since the control input does not start before $t = 1.0\,\mathrm{s}$. However, there is a slight drift for the high-speed cases caused by small inaccuracies of the trim, as trimming is more challenging at high speeds and imbalances develop quickly.

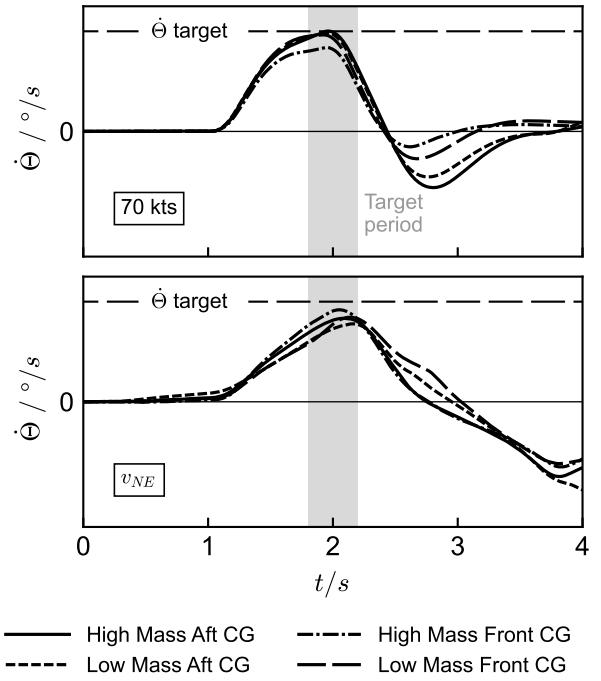

Figure 5: Airframe pitch rate $\dot{\Theta}$. Same scale in both graphs.

The actual target period is indicated by the grey colored area, where the optimizer tries to reach the $\dot{\Theta}$ objective. For 70 kts, the objective is matched very good, except for the case with high mass and front CG. Further, for v_{NE}, the objective is not reached as closely, but with a tolerance of around 20 %. This is caused by the second stage of optimization: After the $\dot{\Theta}$ objective is matched during the first stage, the collective step signal alleviates the $\dot{\Theta}$ peak, as the collective command overlaps the $\dot{\Theta}$ target period. The collective step height d_2 is slightly negative (reduced collective) and leads to an unintended decrease in pitch rate, but keeps the main rotor torque inside its constraint, see lower Figure 6.

The behavior of the main rotor torque Q_{MR} in Figure 6 differs very much between 70 kts and v_{NE}. The torque level of the trims at v_{NE} is much higher as the power required at that speed is much higher compared to 70 kts. In addition, at 70 kts the maximum torque Q_{max} is not reached by far, although the collective step d_2 is used by the optimization. Here, the use of collective is restricted by the $N_z - \ddot{\Theta}$ envelope, as the negative pitch acceleration reaches high values, see Figure 9. This is amplified by the collective as the collective tends to increase the nose-up pitch attitude which leads to a faster drop of the nose afterwards (negative $\ddot{\Theta}$).

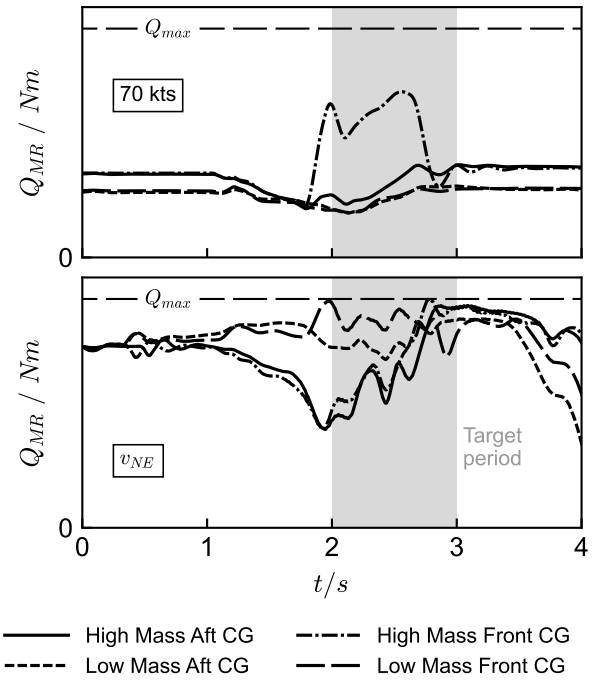

Figure 6: Main rotor torque Q_{MR}. Same scale in both graphs.

However, since the torque level is already close to or already above Q_{max} at v_{NE}, only slight collective corrections d_2, that also may be negative to reduce Q_{MR}, are required to reach Q_{max}. As a consequence, Q_{max} is reached for all high-speed cases.

4.2 Optimized Variables

The input step signals d_1, d_2, and d_3 (compare Figure 4) are the variables used by the optimization. Figure 7 shows the final longitudinal input step signals of all cases. At 70 kts the initial pull on the stick (d_1) needs a change in longitudinal cyclic approximately three times higher compared to the recovery push on the stick (d_3). Both weight configurations featuring the front CG even end up without a recovery push at all. This is caused again by the $N_z - \ddot{\Theta}$ constraint that limits the negative pitch acceleration $\ddot{\Theta}$, compare upper Figure 9. Especially the low-speed cases with front CG tend to develop highly negative pitch accelerations after reaching their peak of pitch attitude Θ, and when the

stick position changes from pulling to pushing. As a consequence of only a very moderate or even no recovery control input, the pitch attitude of the low speed cases remains quite high at the end of the maneuver.

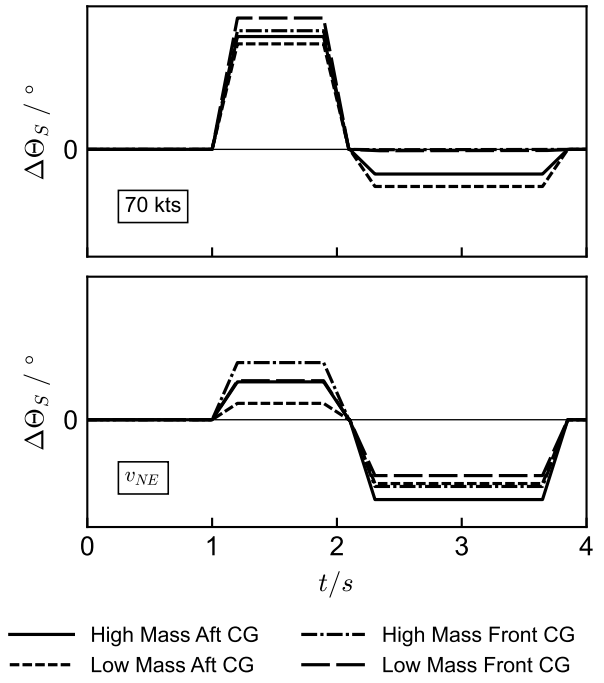

Figure 7: Change in longitudinal control input $\Delta\Theta_S$. Input step signals d_1 and d_3. Same scale in both graphs.

In contrast, at v_{NE}, the initial pull on the stick is always significantly smaller compared to 70 kts as control sensitivity increases with speed due to the higher dynamic pressure. Further, the recovery push is more emphasized as there is still a margin regarding the limit of negative pitch acceleration $\ddot{\Theta}$. As a result, the recovery phase is more effective at v_{NE} and the pitch attitude is almost back to horizontal at the end of the maneuver.

The input step signal d_2 utilizes the collective control in order to adjust the main rotor torque Q_{MR}. At 70 kts all values are positive with a certain bandwidth, see Figure 8. However, as mentioned earlier, the objective of Q_{max} is not reached, see Figure 6. The usage of collective is restricted by a highly negative pitch acceleration after reaching the peak pitch attitude as it tends to increase that negative pitch acceleration. High-mass configurations allow for a stronger collective input as their moment of inertia is higher, and therefore, the pitch response is less pronounced.

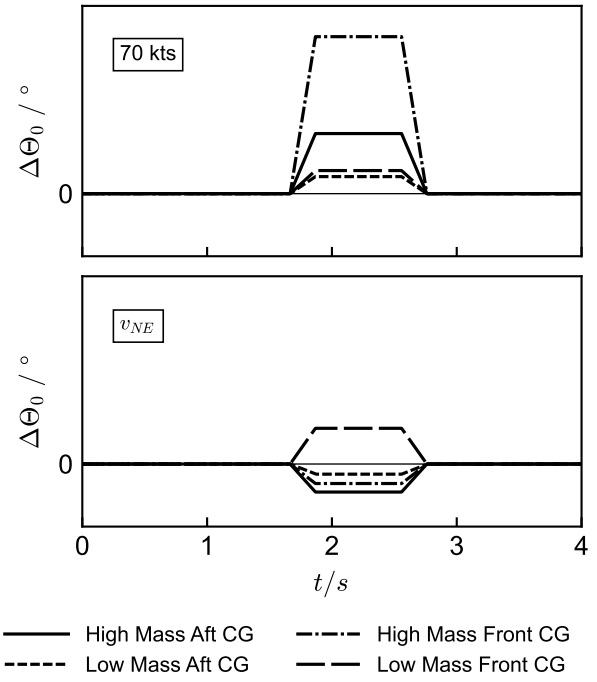

Figure 8: Change in collective control input $\Delta\Theta_0$. Input step signal d_2. Same scale in both graphs.

In contrast, the change in collective input required is mostly negative at v_{NE}, except for low mass front CG, as the torque would already exceed Q_{max} at some points otherwise. Of course the general torque level is anyway already higher, and therefore closer to Q_{max}, at high speeds due to the higher power requirement.

4.3 Constraints and Limits

Figures 5 and 6 already demonstrate the adherence of the maneuvers to the given $\dot{\Theta}$ and Q_{max} constraints.

In addition, Figure 9 shows the $N_z - \ddot{\Theta}$ envelope constraint which takes load factor and pitch acceleration into account. For both low and high speeds, the envelope is respected. The low-speed pull-up maneuvers are mostly stretched between negative and positive pitch accelerations at moderate load factors. Especially the

limit of negative pitch acceleration is reached when the pull on the stick is revoked.

In contrast, at v_{NE}, the maneuvers rather cover a bigger range of low and high load factors which is caused by the bigger impact of changes in rotor angle of attack at high dynamic pressure.

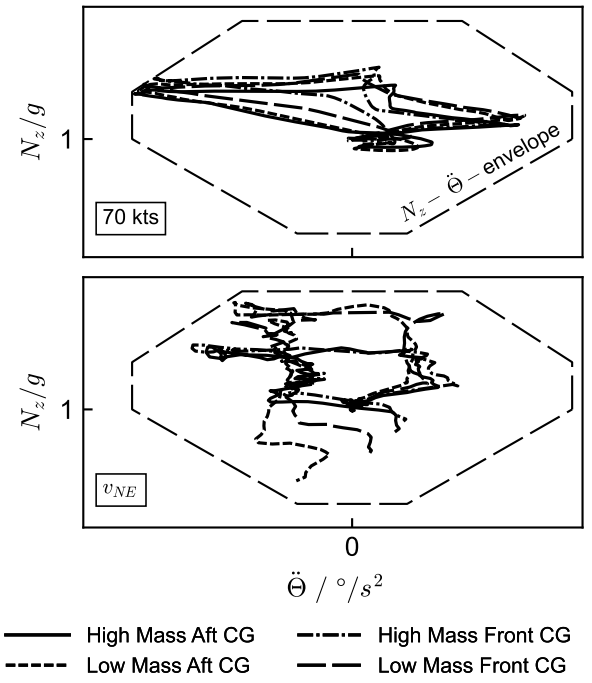

Figure 9: Load factor N_z and pitch acceleration $\ddot{\Theta}$ with envelope constraint.

While only the longitudinal cyclic Θ_S and collective Θ_0 are prescribed by the optimization, lateral cyclic Θ_C and pedal Θ_{TR} are handled automatically by a control system integrated into the Flightlab model. It is employed to keep roll Φ and yaw Ψ attitude at bay. Currently, there are no optimization constraints regarding roll and yaw attitude, but these angles are monitored in order to ensure a pull-up maneuver that is mainly taking place in the vertical plane without lateral deviations too high.

The left-hand column in Figure 10 indicates the roll deviation $\Delta\Phi$ compared to the trim value. For both speeds, roll characteristics are similar with a maximum value around $t = 3\,\mathrm{s}$. Furthermore, the peak yaw deviation $\Delta\Psi$ occurs at $t = 2\,\mathrm{s}$ and $2.5\,\mathrm{s}$, respectively, as can be seen on the right-hand side of Figure 10.

Both deviations stay inside the acceptable limits indicated, which are set to a value considerably lower than 10 ° ($|\Delta\Phi_{lim}| = |\Delta\Psi_{lim}| \ll 10\,°$).

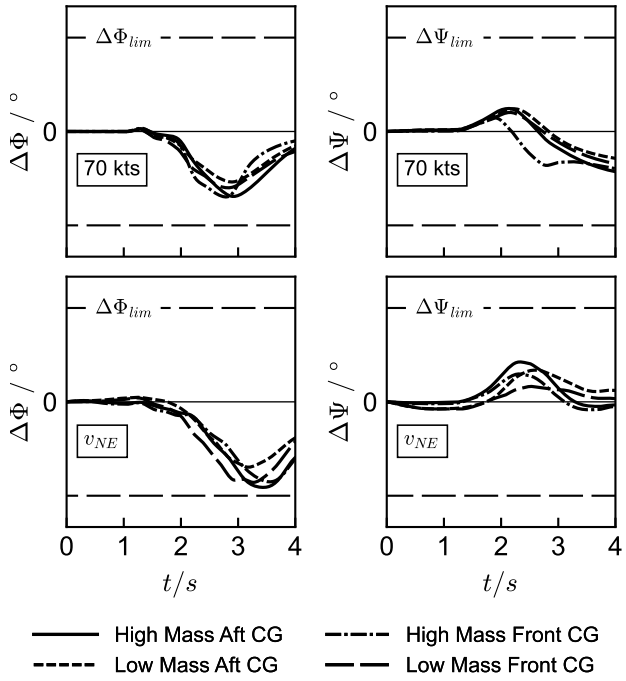

Figure 10: Change in airframe roll $\Delta\Phi$ and yaw $\Delta\Psi$ attitude. Same scale in all graphs.

As a consequence, these lateral deviations are considered very moderate, and the maneuvers are limited to the vertical plane.

4.4 Flight Trajectory

During the pull-up maneuver the helicopter nose is lifted and the flight trajectory is bent upward. Figure 11 demonstrates the altitude change caused by the maneuvers. While the trajectory is horizontal at 70 kts prior to the pull-up phase, there is a sink rate at v_{NE}. This sink rate is required by the initial trim since MCP would otherwise be exceeded in horizontal flight at that speed. Especially the cases with low masses follow a steeper trajectory after the pull-up phase.

Even though the recovery push on the stick (d_3) reduces the pitch attitude of the helicopter, it is not able to stop the initiated climb before the end of the simulated time at $t = 4\,\mathrm{s}$.

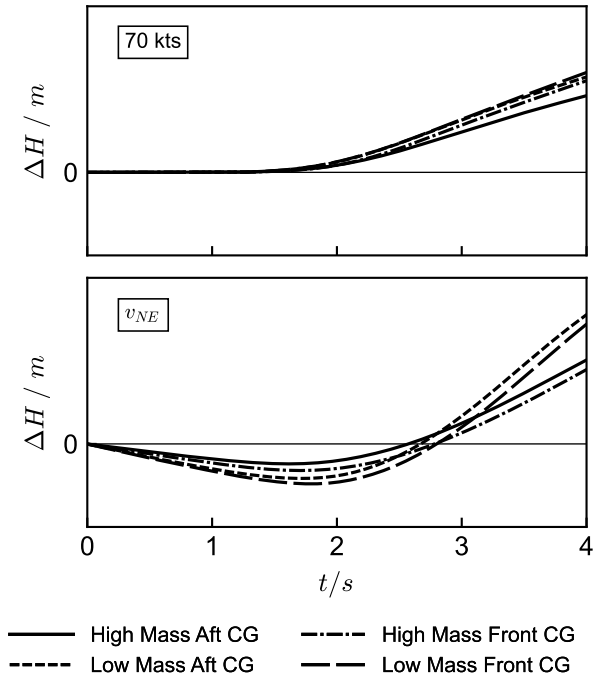

Figure 11: Altitude change ΔH. Same scale in both graphs.

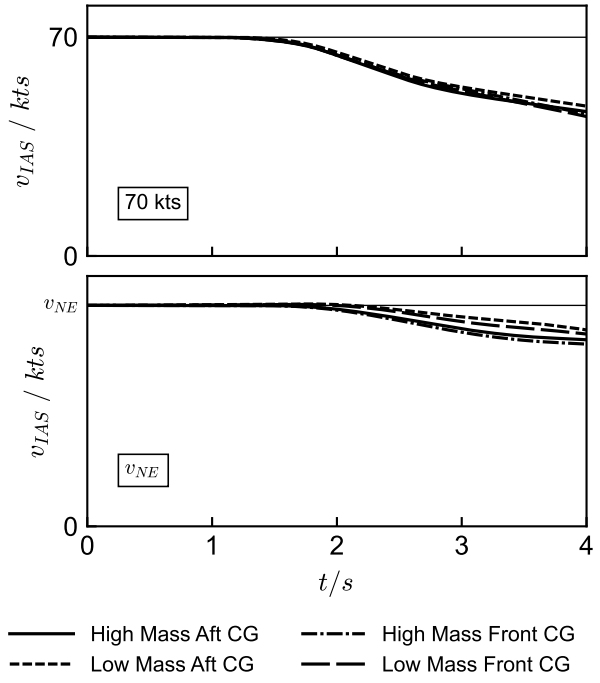

Figure 12: Change in indicated air speed v_{IAS}. Scale differs between graphs.

As a result of the initiated climb, the flight speed decreases when kinetic energy is converted into potential energy, see Figure 12. Bleeding off the flight speed clearly impacts the control sensitivities and the behavior of the helicopter at later stages of the maneuver. Especially the low-speed cases suffer from that effect in the recovery phase as they lose around one third of their speed, which also supports the tendency of highly negative pitch accelerations, affecting the optimization procedure. In contrast, at v_{NE}, the relative speed loss is much lower (≈15 %) which ensures a better control authority in the recovery phase, also giving the optimizer a better margin to work with.

5 CONCLUSIONS

An optimization framework for helicopters called Maneuver Optimization Tool (MOPT) is introduced, coupling a Python tool with Flightlab and employing SciPy optimization methods. A serial optimization of pull-up maneuvers in multiple stages is demonstrated for 70 kts and v_{NE}. In addition, four combinations of low/high mass and front/aft CG location are used for this demonstration. The optimization is set up with three variables, three objectives ($\dot{\Theta}$, Q_{max}, Θ) and three constraints ($\dot{\Theta}$, Q_{max}, $N_z - \ddot{\Theta}$ envelope):

- The pitch rate objective $\dot{\Theta}$ is matched well for 70 kts, but influenced by subsequent optimization stages in the case of v_{NE}.
- The maximum main rotor torque Q_{max} is not matched for 70 kts, as the $N_z - \ddot{\Theta}$ envelope constraint prevents it due to highly negative pitch accelerations $\ddot{\Theta}$ during the recovery phase of the maneuver. In contrast, at v_{NE}, the Q_{max} objective is matched well.
- In the recovery phase, high pitch attitudes Θ are maintained by the low-speed cases since stronger pushing is again prevented by the $N_z - \ddot{\Theta}$ envelope constraint. However, at high speed, an almost horizontal pitch attitude can be recovered easily.
- During the optimization procedure, all constraints are respected.
- Lateral attitude deviations are kept moderate by the control system implemented in the Flightlab model.

Based on the experiences gained so far, the MOPT will be applied to further maneuver types, e.g. a power-off pull-up, a yaw entry or a coordinated turn entry.

REFERENCES

[1] Thomson, D. G. and Bradley, R., "The Principles and Practical Application of Helicopter Inverse Simulation," *Simulation Practice and Theory*, vol. 6, no. 1, pp. 47–70, 1998.

[2] Thomson, D. G. and Bradley, R., "Inverse Simulation as a Tool for Flight Dynamics Research—Principles and Applications," *Progress in Aerospace Sciences*, vol. 42, no. 3, pp. 174–210, 2006.

[3] Kalkan, U. and Tosun, F., "Optimization Based Inverse Simulation Method For Helicopter Pull Up Maneuver," in *Proceedings of the 45th European Rotorcraft Forum, Warsaw, Poland*, 2019.

[4] Tosun, F., Kalkan, U., Tıraş, H., and Yaman, Y., "Generation of Helicopter Pull-Up Maneuver with Harmony Search Optimization Method," in *8th Asian/Australian Rotorcraft Forum, Ankara, Turkey*, 2019.

[5] Guglieri, G. and Mariano, V., "Optimal Inverse Simulation Of Helicopter Maneuvers," in *Communications to SIMAI Congress*, vol. 3, 2009.

[6] Prasad, R., "Development and Validation of Inverse Flight Dynamics Simulation for Helicopter Maneuver," in *4th Asian/Australian Rotorcraft Forum. India, Institute of Science*, 2015.