OPTIMIZATION BASED INVERSE SIMULATION METHOD FOR HELICOPTER PULL UP MANEUVER

Uğur KALKAN, ugur.kalkan@tai.com.tr, Turkish Aerospace Inc. (Turkey)
Fatih TOSUN, fatih.tosun@tai.com.tr, Turkish Aerospace Inc. (Turkey)

Abstract

The aim of this work is to perform helicopter Pull Up maneuver with inverse simulation methodology. This method is based on the combination of Broyden-Fletcher-Goldfarb-Shanno (BFGS) and line search optimization algorithms. By using the inverse simulation method, unsteady maneuvers can be performed automatically by changing the control inputs of the helicopter. In addition, maneuverability, agility and performance limits can also be determined. In the conventional methods, trial-error is being performed in order to achieve the maneuver. However, inverse simulation method not only saves time due to trial-error but also increases the accuracy of the aimed maneuver. In the inverse simulation numerical optimization problem, pilot control inputs are defined as the design variables. By changing the pilot inputs, objective function which defines the target maneuver is minimized. The algorithm is used to perform a Pull Up maneuver at maximum achievable load factor for a large category rotorcraft within the available engine power and rotor control limits. Flightlab® software is used for the flight simulations. In mathematical model, blades are modelled as rigid and flow is modelled as uniform inflow. Nonlinear transient Pull Up simulation have been performed and maximum 3.5g is obtained.

NOTATIONS

$a_{x_{body}}$ $a_{y_{body}}$ $a_{z_{body}}$	Total acceleration components along x, y, z body axes
$\dot{u}$ $\dot{v}$ $\dot{w}$	Acceleration components along x, y, z body axes
u v w	Velocity components along x, y, z body axes
p q r	Roll, pitch and yaw rates
$n_{z_{body}}$	Load factor along z body axes
cg	Center of Gravity
BFGS	Broyden-Fletcher-Goldfarb-Shanno
θ_0	Main rotor collective pitch angle
θ_{1s}	Main rotor longitudinal cyclic pitch angle
∇f	Gradient of function f
B_k	Hessian Matrix at kth iteration
α	Step size
x_k	Design vector at kth iteration
g	Earth gravity 9.81 m/s^2

---

1. INTRODUCTION

For the transient helicopter manoeuvres, pilot control inputs which are collective, longitudinal cyclic, lateral cyclic and pedal vary with time. Each different pilot input results in a different manoeuvre. Therefore, performing a desired manoeuvre requires trial-error if the process is done manually. The manual process both takes time and decreases the accuracy of the manoeuvre. Therefore, there is a need of methodology to determine automatically the pilot control inputs for the desired manoeuvre. Obtaining the pilot control inputs for the desired transient manoeuvre is called inverse simulation method. It is an automatic method to find pilot control inputs, no trial-error is required to find the control inputs. However, solution highly depends on objective function thus it should be constructed carefully.

Some of the very first attempts about the inverse simulation method is done at 1930s by R.T. Jones [1]. He studied the equations of motions for the gust analysis. Also, Thomson and Brandley developed nonlinear six degree of freedom inverse simulation algorithm for rotorcraft namely “HELINV” and studied the hurdle-hop and quick hop manoeuvres [2]. It is shown that the tool developed is an effective and useful for unsteady helicopter manoeuvres. These methods are the traditional inverse simulation methods and fixed prescribed trajectory is tried to be achieved. Celi [3] proposed a new methodology for the inverse simulation. This methodology operates on a family of possible trajectories instead of a single trajectory. It selects the best one and there may exist multiple pilot

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019

inputs combinations for the desired manoeuvre. Slalom manoeuvre's trajectory is defined indirectly in the objective function and is minimized by using the gradient based BFGS algorithm. In addition, Guglieri and Mariano [4] studied the same slalom manoeuvre by using genetic algorithm. One advantage of the genetic algorithm is that it can find the global minimum. On the other hand, gradient-based methods can find the local minimum but they are faster than the genetic algorithms because, there is no search direction in the genetic algorithms. Lots of analyses are needed for the solution and many of the analyses have high value of objective function due to the random inputs.

In this paper, inverse simulation method is used to obtain the helicopter maximum available load factor for Pull Up manoeuvre. Maximum load factor capacity of the helicopter is tried to be found. Pull Up is one of the limit flight manoeuvre, which sizes the helicopter structure mainly in vertical direction. BFGS algorithm is used in the calculations. The objective function is constructed in such a way that the goal of the optimization algorithm is maximizing the load factor. In addition, while finding the maximum load factor, engine power, flap stop limits and theta limits are defined in the objective function. Therefore, exceedance of the engine power, flap stop limits are prevented. Also, logical region for the theta is defined.

According to Certification Specification for Large Rotorcraft [5] (CS 29.337), the rotorcraft must be designed for a limit manoeuvring load factor ranging from 3.5g and -1.0g. For the positive limit load factor, if the helicopter design load factor is less than 3.5g, it must be shown by analysis or flight test that probability of exceeding the design limit load factor is extremely remote. Helicopters can attain limit load factor via symmetrical Pull Up manoeuvre. This manoeuvre sizes the airframe structure. The schematic representation of the symmetrical Pull Up manoeuvre is shown in Figure 1.

Acceleration of the helicopter cg in body z axis is derived by using Coriolis Transport Theorem as shown in Equation (1) and (2). Since symmetrical Pull Up is considered, roll rate (p) and sideward velocity (v) is close to zero and can be eliminated from the equation (2). Therefore, Equation (3) gives the $a_{z_{body}}$ for symmetrical Pull Up manoeuvre. By using the acceleration, load factor is calculated as given in Equation (4). According to the Equation (4), maximum load factor can be achieved by minimizing the $\dot{w}$ and maximizing the $u*q$ as Seyhan et al. discussed [6].

$$(1) \qquad \begin{bmatrix} a_{x_{body}} \\ a_{y_{body}} \\ a_{z_{body}} \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} x \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$where \begin{bmatrix} u \\ v \\ w \end{bmatrix} : body\ velocities, \qquad \begin{bmatrix} p \\ q \\ r \end{bmatrix} : body\ angular\ rates$

$$(2) \qquad a_{z_{body}} = \dot{w} - u*q + p*v$$

$a_{z_{body}}: acceleration\ in\ body\ z\ direction(\ +ive\ downward)$

$$(3) \qquad a_{z_{body}} = \dot{w} - u*q$$

$$(4) \qquad n_{z_{body}} = 1 - \frac{\dot{w} - u*q}{g}$$

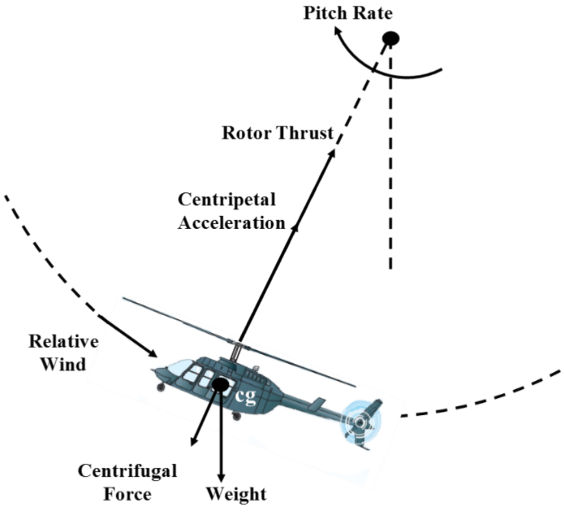

Figure 1. Schematic Representation of Pull Up Maneuver [8]

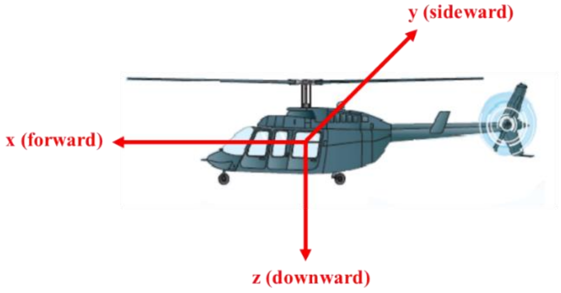

Figure 2. Helicopter Body Axis System

For the large category rotorcraft, 3.5g is aimed to be achieved. In the literature there is no unique definition of Pull Up Manoeuvre. There are some methods performing dynamic trim and simulation for the Pull Up by just changing the longitudinal cyclic [6]. In addition, simulation of Pull Up without loss of altitude for the purpose of terrain avoidance manoeuvre is achieved by both changing collective and longitudinal cyclic [7]. In this paper, transient analysis for Pull Up manoeuvre is achieved by both changing the longitudinal cyclic and collective to maximize the load factor. Pitch controller is closed whereas roll and yaw controllers are opened for the transient analysis. In addition, constraints which are engine available maximum power, flap stop limits and maximum pitch angle are defined in the objective function. The manoeuvre is performed without exceeding these limits. Defining the manoeuvre with an objective function is a critical issue. If the manoeuvre is not properly defined in the objective function, the optimization algorithm may fail to perform desired manoeuvre.

2. OPTIMIZATION METHODOLOGY

For the inverse simulation problem, script based on the BFGS and line search optimization algorithms is developed. Collective and longitudinal cyclic at defined time points are considered as the design variables. Adding a single time point to the problem increases the design vector by 2. Therefore, for n time point there is a total 2n number of design variables.

$$X^T = \{\Theta_0(t_1), \Theta_0(t_2), \dots \Theta_0(t_n), \Theta_{1s}(t_1), \Theta_{1s}(t_2), \dots \Theta_{1s}(t_n)\}$$

Design variables are updated in the BFGS-line search algorithms in order to achieve the desired manoeuvre. In the BFGS, the search direction is obtained. By using this direction, line search obtains the minimum objective on the search direction. The algorithm used is given in the following steps [9].

1. Obtain direction p_k by solving,

$$p_k = B_k^{-1} * [-\nabla f(x_k)\ ]$$

2. Perform line search to find acceptable stepsize α_k in the direction found,

$$\alpha_k = argmin\, f(x_k + \alpha * p_k)$$

3. Set $s_k = \alpha_k * p_k$ and update

$$x_{k+1} = x_k + s_k$$

4. $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

5. $B_{k+1} = B_k + \frac{y_k * y_k^T}{y_k^T * s_k} - \frac{B_k * s_k * s_k^T * B_k}{s_k^T * B_k * s_k}$

At the start of the algorithm, initial condition x_0 is the user defined collective and longitudinal cyclic. Also, initial approximation of Hessian Matrix (B) is the identity matrix. In addition, Hessian Matrix is obtained numerically at step 5 and gradient is calculated by Newton's Central Difference Method at step 4. During the analysis, backward difference is also used in the calculation of gradient. However, it is seen that central difference gives more accurate and fast results as expected. Although central difference requires 2 times more Flightlab analysis than the backward difference, total time in order to achieve the converged result is less in the central difference. This is mainly due to the accuracy of the central difference. It gives more accurate search direction than the backward difference. Therefore, in the analysis central difference is used in the calculation of gradient vector. After finding the search direction, p_k is scaled such that its maximum value is 1. Therefore, maximum of the search direction is equal to the value of α. Line search algorithm is constructed such that the $x_k + \alpha * p_k$ is split into 8 equal segments. For each of these points, objective function is evaluated. For the minimum of the objective, neighbour is identified and again these segment is divided into equal 8 segments. Minimum of the objective is the output of the linesearch. If the x_{lower} or x_{upper} has the minimum value then it has only one neighbour. This time segment between the neighbour and x_{lower}/x_{upper} is split into 8 equal segments. The split operation is done twice in each line search. As a result, minimum objective between x_{lower} and x_{upper} is found.

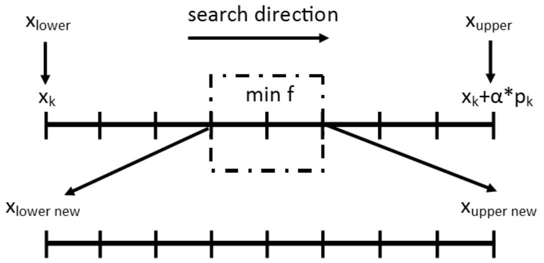

Figure 3. Schematic Representation of Line Search Algorithm

The constraints are defined based on the helicopter design limits and the Pull Up manoeuvre definition. As the design limits, maximum blade flapping angle, engine maximum and minimum available power and helicopter pitch attitude are considered. These are defined at constraints 1 to 3. For the maximum load factor, integration is performed as given in constraint 4. Maximum loaf factor is tried to be achieved between the 4-5 seconds. Since helicopter can not achieve 5g load factor and the maximum available load factor is tried to be found, area between the 5g and load factor of the helicopter is used as objective.

$$g_1 = \begin{cases} -\log\left(\theta_{\max flap\,lim} - \max\left(\theta_{flap}\right)\right) - \log\left(\theta_{\min flap\,lim} + \min\left(\theta_{flap}\right)\right), \\ \quad \text{for } \max\left(\theta_{flap}\right) < \theta_{\max flap\,lim} \ \ and \ \min\left(\theta_{flap}\right) > \theta_{\min flap\,lim} \\ \quad Infinity, \quad otherwise \end{cases}$$

$$g_2 = \begin{cases} -\log\left(\theta_{\max H/C} - \max\left(\theta_{H/C}\right)\right) - \log\left(\theta_{\min H/C} + \min\left(\theta_{H/C}\right)\right), \\ \quad \text{for } \max\left(\theta_{H/C}\right) < \theta_{\max H/C} \ \ and \ \min\left(\theta_{H/C}\right) > \theta_{\min H/C} \\ \quad Infinity, \quad otherwise \end{cases}$$

$$g_3 = \begin{cases} -\log\left(HP_{engine\,max} - \max\left(HP_{engine}\right)\right) - \log\left(\min\left(HP_{engine}\right)\right), \\ \quad \max\left(HP_{engine}\right) < HP_{engine\,max} \;\; and \;\; \min\left(HP_{engine}\right) > 0 \\ \qquad Infinity, \quad otherwise \end{cases}$$

$$g_4 = \int_{t=4}^{t=5} abs(Nz - 5)^2 * dt$$

Objective function to be minimized in the optimization algorithm is given as follows.

$$f_{objective} = g1 * p1 + g2 * p2 + g3 * p3 + g4 * p4$$

where p_1, p_2, p_3 and p_4 are the penalties applied. Constraints from 1-3 are written using the logarithmic barrier function. A sample logarithmic function is shown in Figure 4. It can be seen that at the boundaries of the limit barrier function increases, which results in increase in the objective. In addition, if the barrier function limit is exceeded objective goes to infinity. Therefore, constrained variables stay within the limits defined during the optimization. Exceedance of the limit/barrier is prohibited and is not allowed.

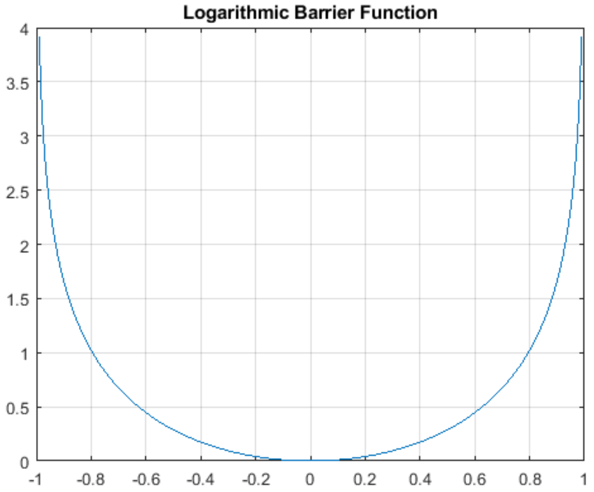

Figure 4. Sample Logarithmic Barrier Function

3. SENSITIVITY ANALYSIS

As described in the methodology section, there are some parameters used in the analyses. For example, epsilon used in the Newton’s Central Difference highly effects the solution. Also, penalty values used in the objective functions change the results. In addition, maximum step size used in the line search algorithm may lead to algorithm to be stuck into local minimum which is the general problem of gradient based optimization algorithms. In this section, some sensitivity analysis for these variables are conducted. Note that starting point is the 100 Knot forward flight trim in the Pull Up maximum N_z problem.

It can be seen from the Figure 5 that epsilon has significant effect on both convergence and analysis time. All other optimization parameters are kept constant, only epsilon is changed. It can be concluded that eps 0.1 is the most efficient value among the others. About 30 iterations it converges the solution and it takes 7 hours to complete the analysis. Also, note that there is 4 completed iterations for eps=1 and 2 completed iterations for eps=5. The solution did not go further. This is because of the barrier function used. During the iterations some of the parameters defined (power, theta and flapping) becomes to be near the boundary. Using high value of epsilon during the perturbations pushes these parameters outside the boundary defined in barrier function. As a result, objective values goes to infinity and this makes gradient infinity. Therefore, using epsilon values greater than 1 is not useful for this optimization problem. Note that zero collective-longitudinal cyclic is defined as initial condition and barrier function penalties are defined as 10 but the load factor penalty is defined as 100 in the epsilon sensitivity analysis.

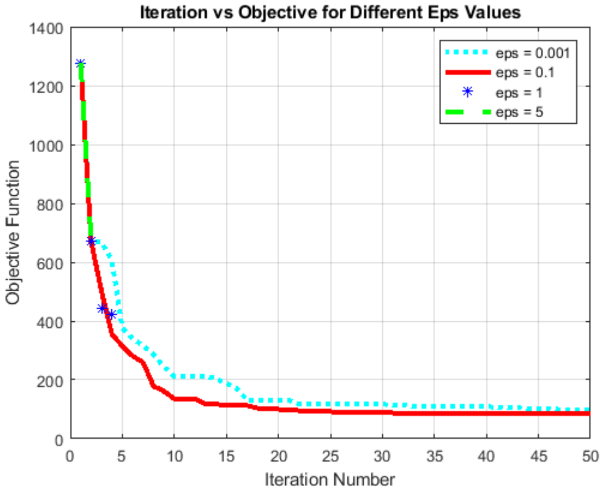

Figure 5. Effect of Epsilon on the Results/Convergence

After setting epsilon as 0.1, sensitivity analysis is done for the maximum step size used in line search algorithm. Unit of the step size is the percentage of pilot control input. To illustrate, maximum step size of 10 means that design variable having the maximum gradient value is scaled to 10 in the first iteration of the line search algorithm. Comparison is done for different maximum step sizes of 5, 10 and 20. It can be seen from the Figure 6 that step size of 5 is faster compared to step sizes of 10 and 20. It converged the solution at the 20th iteration. Also, note that the convergence values of the results are all same, only the convergence speed changes. Considering the run time, step size of 5 is used in the following analysis.

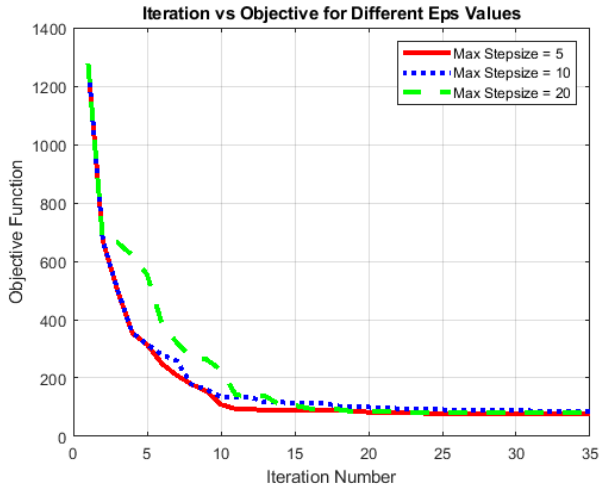

Figure 6. Effect of Max Stepsize (used in Line Search) on the Results

Using epsilon as 0.1 and max step size as 5, different analysis are performed for different penalty values of the barrier function. Since different penalty values are considered, objective function value is different at converged solutions. Therefore, objective function is normalized between 0-1. Objective vs iteration graph only gives idea about the convergence speed. According to Figure 7, almost every analysis converged the solution about 20 iterations. Although converge speeds are same, maximum N_z values may be different. Figure 8 shows that maximum load factor is highest for the barrier function penalty value of 5. Note that there is decreasing trend of maximum load factor with the increasing barrier function penalty values. This is expected because g_1, g_2 and g_3 are kept away from the limiting values by increasing the barrier penalties. As the penalties decrease, variables defined in g_1, g_2 and g_3 are close to the barrier. As can be seen from Figure 8 maximum load factor of 3.5g is achieved.

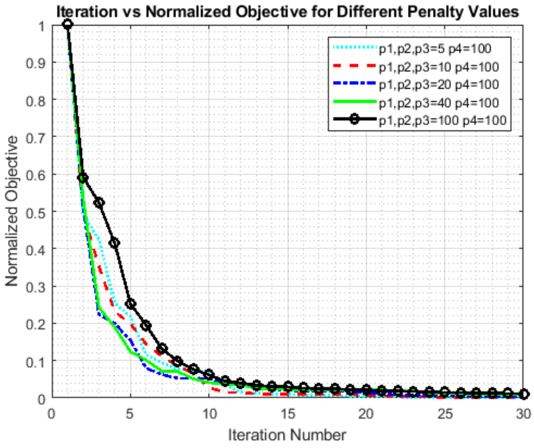

Figure 7. Effect of Barrier Function Penalties on the Results

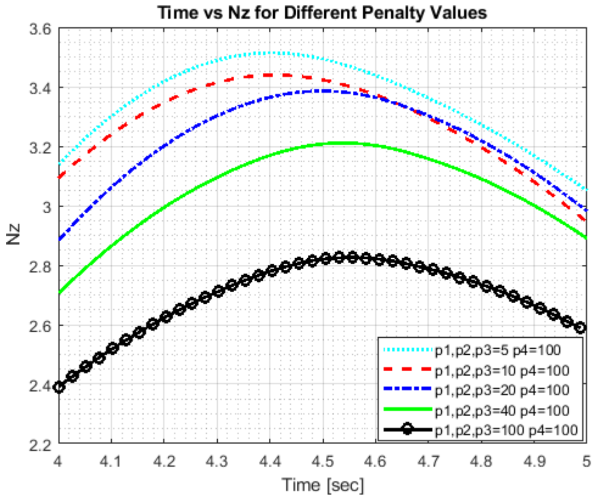

Figure 8. Maximum Load Factor Achieved for Different Barrier Function Penalty Values

For the maximum load factor, optimization algorithm variables are set as given in Table 1. However, initial condition may be effect the solution. In order to see the effect of initial condition, analysis with different initial conditions are performed. At each design points random input is generated as can be seen from Figure 9. Using a random data for the initial condition leads convergence problems. For random pilot input 2 and 3, the analysis converged as can be seen from Figure 10. However, random input of 5 is not a good initial condition such that objective is infinity for the first iteration. It could not even iterate the solution. In addition, with random pilot inputs of 1, 4 solution did not converged. Because random initial condition lead the barrier function variables close to the limits, perturbation with selected maximum step size pushed these variables outside the limits. Therefore, convergence problem occurred. From the converged results as given in Figure 11, it can be seen that all the different initial conditions results in almost same maximum load factor. Therefore, it can be concluded that maximum load factor achieved is same for the different initial conditions if the solution is converged. It may indicate that global minimum is obtained. In addition, it is logical to use zero initial condition for Pull Up problem because of the convergence problem.

Table 1. Optimization Algorithm Variables for Maximum Available Load Factor

Parameters	Values
ε	0.1
Max Step size	5
p_1, p_2, p_3	5
p_4	100

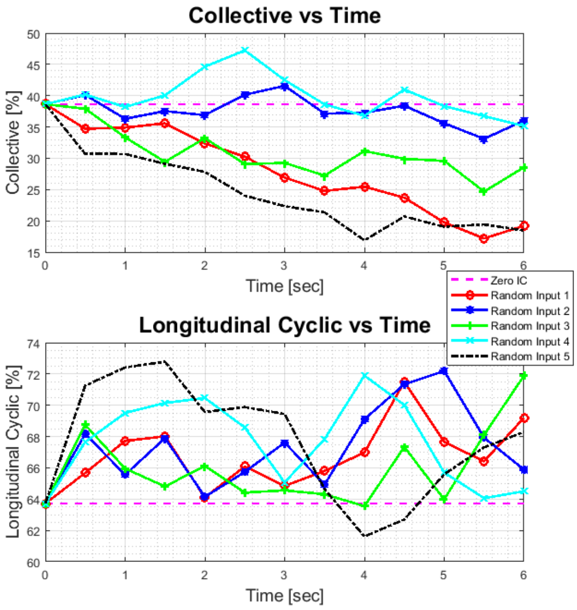

Figure 9. Randomly Generated Pilot Control Inputs

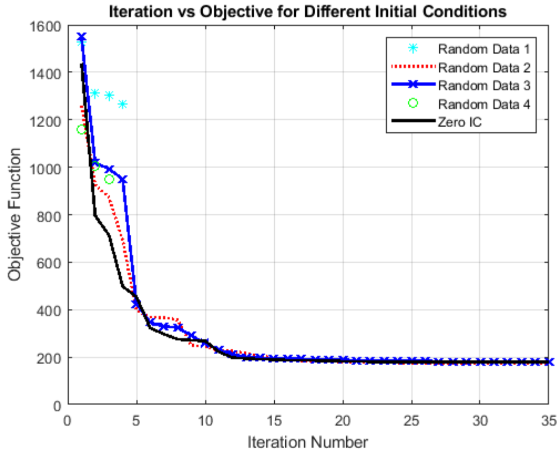

Figure 10. Iteration vs Objective for Different Initial Conditions

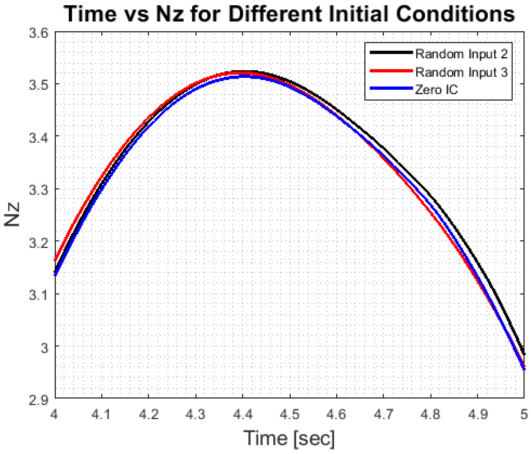

Figure 11. Load Factors for Different Pilot Initial Conditions

4. RESULTS AND DISCUSSION

Pull Up manoeuvre is started from the 100 Knot forward flight trim condition. 6 seconds transient analysis is performed. Between the 4-5 seconds maximum load factor is aimed. After achieving the maximum load factor, 1 second is left to see the trend of the helicopter attitudes whether it diverges or trends to decrease. In addition, no disturbance is given to the control inputs for the initial condition.

Flight path for the Pull Up problem is not defined in the objective function. Between the 4th and 5th seconds maximum load factor is tried to be achieved. In order to achieve the maximum N_z, flight path of the helicopter formed the circle shape with a certain radius as can be seen from the Figure 12. The shape of flight path is similar to Figure 1 as described earlier. In addition, maximum load factor of 3.5g is achieved at the deepest point of the cavity.

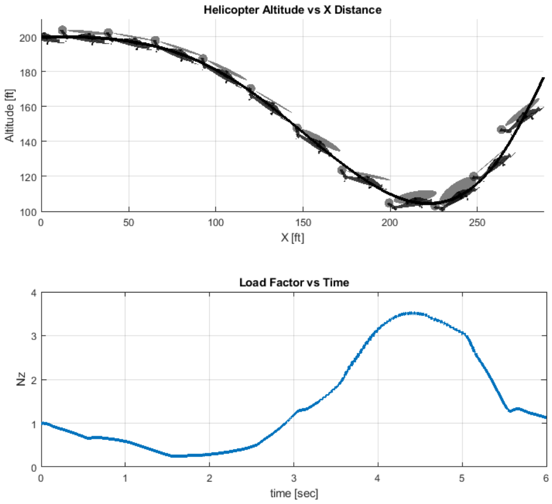

Figure 12. Helicopter Altitude and Nz for Pull Up Manoeuvre

According to the Figure 13, at the start of the manoeuvre longitudinal cyclic is decreased while maintaining the collective almost constant. This yields in altitude decrease and pitching down the helicopter, beginning of concave down flight path. Then, both collective and longitudinal cyclic is increased and concavity of altitude is achieved. Maximum load factor of 3.5g is obtained almost at the deepest altitude location. Also, note that load factor does not have sharp peak. Almost 1 second, between 4th and 5th seconds, it has g value greater than 3.

Assuming symmetrical manoeuvre, $u * q - \dot{w}$ should be maximum in order to achieve maximum load factor as described in the Equation (4). However, there is a difference between the actual N_z and resultant load factor due to Equation (4) as can be seen from Figure 13. This is because of the asymmetrical behaviour of the helicopter during Pull Up. While deriving Equation (4), it is assumed that p*v is zero due to symmetrical Pull Up. However, during the manoeuvre helicopter tends to roll and sideward velocity v increases as can be seen from Figure 13. This is because of the gyroscopic effect of main rotor which has tendency to maintain steady direction of its axis of rotation. Pitching up the helicopter rolls helicopter to the right. Roll and sideward motion is tried to be kept constant by using the attitude controllers in the mathematical model. However, it is not sufficient to keep the roll and sideward velocity v constant because there is an abrupt change in both collective and longitudinal cyclic. As a result, using the inverse simulation method, transient analysis is performed for Pull Up manoeuvre and losses due to symmetric assumptions are eliminated. In addition, total power has a decreasing trend up to 4.5 sec due to the increase of pitch angle and steady value of collective. Since air is coming from the downward as theta increases, power decreases. However, after 4.5 second, power increases even if theta increases because collective is pulled. At the start of manoeuvre, oscillation of main rotor flapping is small. After pulling back the longitudinal cyclic flapping oscillations becomes increasing and almost reaches to the barrier function limits. To conclude, Pull Up manoeuvre for maximum load factor is performed without exceeding the power, flap stop and theta limits.

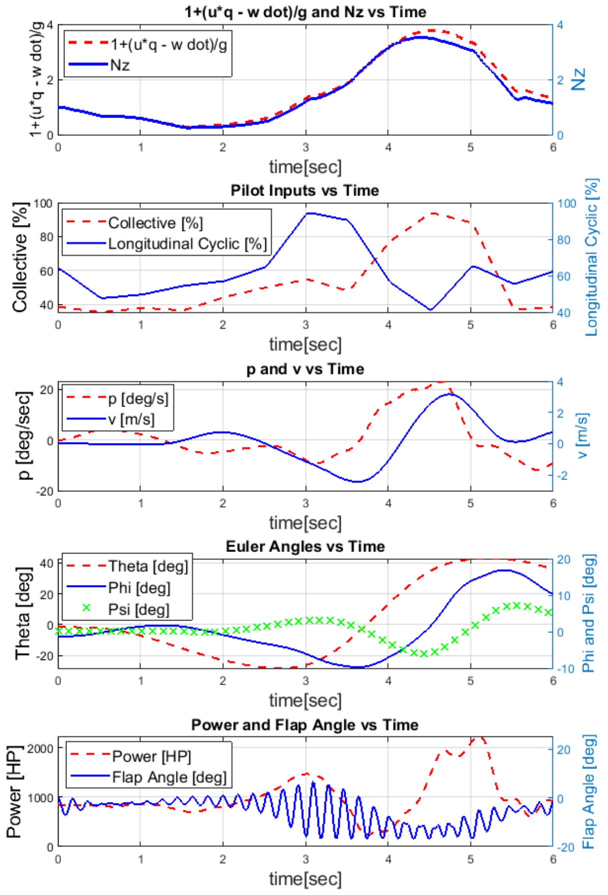

Figure 13. Resultant Pull Up Manouevre

5. SUMMARY AND CONCLUSIONS

This paper presented the details of transient Pull Up manoeuvre performed with inverse simulation methodology. By using this method, pilot control inputs of collective and longitudinal cyclic are automatically determined while achieving the maximum load factor. It is impossible to obtain maximum load factor by using trial-error method because there are several combinations of the pilot control inputs. Also, lots of engineering effort is eliminated to perform a transient manoeuvre. In addition, dynamics of Pull Up manoeuvre is investigated. Flightlab® software is used for the mathematical modelling and blades are modelled as rigid, flow is modelled as uniform inflow. In addition, constant RPM 100% is used.

Some sensitivity analysis is performed for the optimization problem. Convergence and results highly depends on initial conditions, maximum step size used in line search, epsilon used in gradient calculation and penalty values of the objectives. Therefore, these values are changed and results are investigated. It is seen that initial condition may result in convergence problems. However, if a convergence problem is not occurred maximum available load factor is same for different initial conditions. This may point to the global minimum although BFGS method is gradient based and it can stuck into local minimum. Maximum step size also effected the convergence. During the line search, higher values of maximum step size pushed the variables used in barrier function outside the limits. Epsilon both affected the convergence and run time. Finally, penalty values used effected the maximum available load factor because it determines the variable distances to the barrier values. It actually acts like a spring, increasing the penalty of the barrier functions eliminates convergence problem but it leads to low value of load factor.

For the maximum load factor Pull Up manoeuvre, 3.5g is achieved by using optimization based inverse simulation method. Flight path is not defined in the problem objective and pilot control inputs are automatically determined. While achieving the maximum load factor, limits of theta, power and flap stop is not exceeded.

ACKNOWLEDGEMENT

The author is grateful to his colleagues Fatih Mutlu Karadal at Turkish Aerospace for his all advice and technical discussions.

REFERENCES

[1] R.T. Jones "A Simplified Application of the Method of Operators to the Calculation of Disturbed Motions of an Airplane", NACA Report 560, 1936

[2] Thomson D.G. and Bradley R. "Development and Verification of an Algorithm for Helicopter Inverse Simulations" Vertica Vol. 14, No. 2, pp185-200, 1990

[3] Celi R. "Optimization-Based Inverse Simulation of a Helicopter Slalom Maneuver", Journal of Guidance, Control and Dynamics, Vol. 23, No.2, March-April 2000

[4] Guglieri G. and Mariano V. "Optimal Inverse Simulation of Helicopter Maneuvers" Communications to SIMAI Congress Vol. 3, 2009, pp261-272

[5] European Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Large Rotorcraft CS 29*. Amendment 4, 30 November 2016.

[6] Seyhan G., Bilen M. E., Şahin M., Gürak D., Yaman Y. *A Convergence Study for a Longitudinal Maneuver by Using Various Tools*. AHS 72nd Annual Forum, Florida, USA, 17-19 May 2016

[7] Raghavendra P., Abhiskek. *Development and Validation of Inverse Flight Dynamics Simulation for Helicopter Maneuver*. ARF 4th Forum, India, 16-18 November 2015

[8] Leishman, J. G., "*Principles of Helicopter Aerodynamics*", Cambridge University Press, New York, NY, 2006.

[9] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (1983)*