Communications to SIMAI Congress,
ISSN 1827-9015, Vol. 3 (2009) 261 (12pp)
DOI: 10.1685/CSC09261

Optimal Inverse Simulation Of Helicopter Maneuvers

Giorgio Guglieri[1], Valeria Mariano[1]

[1] *Dipartimento di Ingegneria Aeronautica e Spaziale, Politecnico di Torino, Corso Duca degli Abruzzi 24 10129 - Torino - Italy*
giorgio.guglieri@polito.it
valeria.mariano@polito.it

Abstract

An inverse simulation methodology is applied for the optimization of the pirouette and the slalom maneuvers described in the ADS-33 specifications. The optimization process is performed using a genetic algorithm: defined the desired trajectory, the algorithm provides the correct pilot inputs necessary to reproduce the maneuver. The algorithm is based on the maximization of an objective function defined appropriately.

Keywords: `rotorcraft dynamics, ADS-33, maneuvers simulation`

1. Introduction.

During the development of the project of a rotorcraft, it must be verified that the machine is able to meet the requirements included in the ADS-33 specifications [1], concerning the assessment of the flying and handling qualities, the safety, the certification.
The ADS-33 provides the ranges of the time and the frequency analysis parameters with respect to which the flying and handling qualities of the helicopter can be defined. Besides, several maneuvers, the Mission Task Elements and the requirements related to their execution are presented.
The MTEs are usually performed in flight tests: the pilot ratings procedure then follows and through the Cooper-Harper scale the handling qualities of the machine are assessed. However, the MTEs can be also reproduced through the analytical simulation, a proceeding that has revealed to be useful for the assessment of the flying and handing qualities and the analysis of the helicopter performances, stability and maneuvrability.
The inverse simulation is a suitable method to be employed for the analytical reproduction of maneuvers, thus also for the MTEs included in the ADS-33 specifications. Once the trajectory of the maneuver has been defined, the inverse simulation procedure provides the commands necessary

Received 13/02/2009, in final form 15/06/2009
Published 31/07/2009

Licensed under the Creative Commons Attribution Noncommercial No Derivatives

to perform the maneuver, that is, the time history of the appropriate pilot inputs.
In the present work it has been chosen to deal with the inverse simulation problem in terms of an optimal control problem: this approach consists in defining a suitable cost function related to the error between the prescribed trajectory and the one resulting from the simulation, and in minimizing it through the adoption of a numerical algorithm.
In this case, the Genetic Algorithm developed by D.L.Carroll [2] has been employed to solve the optimization problem. The same algorithm was previously employed to other aerospace applications [3] [4] [5] [6].

2. Maneuvers

The inverse simulation procedure has been carried on for the reproduction of two of the several MTEs presented in the ADS-33 specification, that are the slalom and the pirouette maneuvers.
The slalom maneuver (Fig.1) consists of a first stationary flight along the course, followed by a series of turns; the helicopter has to complete the maneuver in straight flight again on the centerline.

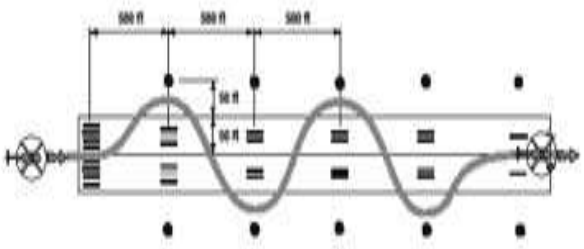

Fig. 1. The slalom maneuver

The pirouette maneuver (Fig.2) starts from a hover condition on a point of a reference circonference of 100 ft radius; the rotorcraft must then perform a lateral translation around the circonference, pointing the nose to the centre of the circle. The maneuver has to be performed in both directions. As it can be read in the specifications, the ADS-33 doesn't describe the MTEs defining a specific trajectory, it just indicates some constraints on geometric and dynamic variables, like time, speed, altitude, X and Y position components.

DOI: 10.1685/CSC09261

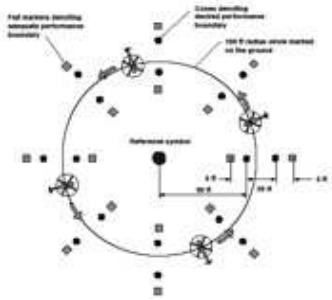

Fig. 2. The pirouette maneuver

The slalom trajectory has been defined by the following relations:

$$\text{For} \quad t \le t_{late} \qquad \begin{aligned} y_{ref} &= 0 \\ x_{ref} &= x_{ref} + v \cdot \Delta t \end{aligned}$$

$$\text{For} \quad t > t_{late} \qquad \begin{aligned} dy &= x_{late} \cdot \sin \tfrac{v(t-t_{late})}{dist}\pi - x_{late} \cdot \sin \tfrac{v(t-t_{late}-\Delta t)}{dist}\pi \\ ds &= v \cdot \Delta t \\ dx_{ref} &= ds \cdot \cos(a\sin(\tfrac{dy}{ds})) \\ x_{ref} &= x_{ref} + dx_{ref} \\ y_{ref} &= x_{late} \cdot \sin \tfrac{v(t-t_{late})}{dist}\pi \end{aligned}$$

where t_{late} is the duration of the initial straight flight, Δt is the step time used by the Runge-Kutta numerical algorithm employed for the integration of the equations of motion, x_{late} is the maximum lateral displacement admitted with respect to the centerline, *dist* is the distance between two close centerline crossing points, v is the speed at which the trim condition has been calculated. Finally, x_{ref}, y_{ref} and z_{ref} are the position components that define the desired trajectory, with respect to which the difference of the flown trajectory is computed. The maneuver has been defined in a different way with respect to the $ADS-33$, such that the helicopter performs six turns instead of four, without returning to the centerline at the end of the task.

The pirouette maneuver is described as:

$$z_{late} = 0 \qquad y_{ref} = dist \cdot \sin(\omega t) \qquad x_{ref} = -dist \cdot \cos(\omega t)$$

$$\varphi_{ref} = 0 \qquad \psi_{ref} = -\omega t$$

where *dist* is the circle radius, ω is the angular velocity and φ_{ref} , ψ_{ref} , x_{ref} , y_{ref} and z_{ref} are respectively the reference values of roll and heading angles, position X and Y, altitude Z.

3. Rotorcraft mathematical model

In the present work a linear model of the BO105 rotorcraft has been used. For the simulation of the pirouette and the slalom maneuvers, the linearized models in hover and forward flight at 60 knots have been adopted respectively [7]. The model used is very simple, including just the fusolage dynamics, thus the state and the control vectors are:

$$x = (u \quad w \quad q \quad \theta \quad v \quad p \quad r \quad \varphi \quad \psi)$$

$$u = (\delta c \quad \delta b \quad \delta a \quad \delta p)$$

The machine shows to be unstable in both the flight conditions, thus it has been chosen to stabilize the machine, through the implemetation of a SAS, employing the LQR optimal control technique. The optimal gain matrix K, having dimension (4x9), have been computed; it makes the rotorcraft to exhibit a stable behaviour in both the considered flight conditions.

4. The Genetic Algorithm

The Genetic Algorithms [8] [9] [10] are search methods based on the principles of natural selection and survival of the fittest that govern the evolution of natural population; when used as function optimizers, although these algorithms have several other kind of applications, the GAs provide, simulating the evolution process, the solution of the optimization problem. The implementation of the genetic algorithm starts with the definition of a prescribed number of individuals: they constitute the population and to each of them the defined design variables are allocated. These parameters are coded in binary strings of bits, called chromosomes, and they are the effective solution of the problem, so it can be said that each individual represents a possible solution; to each individual the value of a fitness function is assigned, indicating how good the solution related to that individual is. If an optimization problem is taken into account, the fitness function is a performance index or a profit or cost function that must be minimized or maximized, according to the type of problem, and that the user must define in an appropriate way.
For the application of the genetic algorithm, several parameters must be defined by the user:

- number of bits for the coding procedure;
- *npopsiz*: size of the population, that is number of individuals of each population;
- *maxgen*: maximum number of generations;

DOI: 10.1685/CSC09261

- *nparam*: number of design variables that consitute the solution;
- *parmax, parmin*: maximum and minimum allowed values of the design variables;
- *nchild*: number of child per pair of parents;
- *ielite*: elitism flag (if elitism is activated, the best individual is automatically duplicated in the next generation);
- *iunifrm*: type of crossover;
- *pcross*: crossover probability (usually between 0.6 and 0.7 for single-point crossover and equal to 0.5 for uniform crossover);
- *pmutate*: jump mutation probability (usually 1/population size);
- *icreep*: creep mutation flag;
- *pcreep*: creep mutation probability (usually equal to: (n.chromosome/n.parameter)/population size);
- *iniche*: niching flag. Niching is usually allowed to prevent that the algorithm converges too fast on a local maximum or minimum.
- *nposibl*: array integer number indicating the minimum variation allowed for each parameter.

5. Integrator and genetic algorithm implementation

In the present work the inverse simulation procedure has been adopted for the reproduction of two maneuvers described in the ADS-33 specification; the inverse simulation allows to obtain as outputs the pilot commands necessary to perform the required maneuver, receiving as input the desired trajectory.
The matter has been approached as an optimal control problem, thus the search of the appropriate pilot inputs is related to the search of the absolute minimum or maximum of a suitable cost function, that in turn is defined in terms of the error between the desired and the actual trajectories. The maneuvers, both the slalom and the pirouette, are divided into time steps: for each time step and only for the first generation, the genetic algorithm starts providing randomly the values of the design variables, that in the actual case are the four pilot inputs, that is, collective, longitudinal and lateral cyclic, pedal.
These parameters are given as inputs to the algorithm used for the integration of the equations of motion; the system to be integrated is made of 9 equations for the description of the rotorcraft dynamics and 3 equations for the navigation. Once the integrator has solved the system, the outputs that it provides, that is, the state vector and the position variables x y z, are used in the evaluation phase, for the computation of the fitness function.
This procedure, from the random generation of the parameters to the eval-

uation stage, is executed for each individual of the population. The first generation is thus completed and the recombination and mutation phases can then follow for the creation of the next generation.
For each member of this new generation, the fitness function is evaluated and recombination and mutation start again.
When the maximum number of generations defined by the user is reached, the ultimate parameters that the genetic algorithm has created should be the optimal pilot inputs for the execution of the first time step of the maneuver, depending on the choice of the fitness function and all the other algorithm parameters.
The entire process is repeated for each time step: the integration of the equations of motion through the Runge-Kutta algorithm and the evaluation of the fitness function are performed on steps of amplitude equal to 1 second, but actually the maneuver is divided into steps time of 0.1 seconds; thus, once the pilot inputs have been determined at the end of a generic step,

$$\Delta t_k = [t_k, t_{k+1}], \qquad t_{k+1} = t_k + 1sec$$

the entire procedure won't be applied to the step:

$$\Delta t_{k+1} = [t_{k+1}, t_{k+2}], \qquad t_{k+2} = t_{k+1} + 1sec$$

but to the one defined as:

$$\Delta t_{k+1} = [t_k + 0.1sec, t_{k+1} + 0.1sec]$$

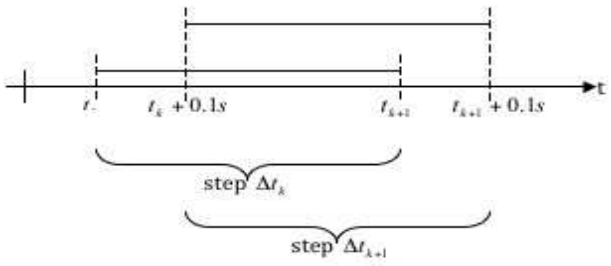

Fig. 3. Step time sequence used by the optimization algorithm.

Both the state and the control vectors determined at the end of the procedure in $t = t_{k+1}$ are given as inputs for the procedure that will be carried on in the step $\Delta t_{k+1} = [t_k + 0.1sec, t_{k+1} + 0.1sec]$. At the end,

DOI: 10.1685/CSC09261

the time history of the optimal pilot inputs that minimize the difference between the desired and the simulated maneuver is obtained.

6. Results

The adoption of the genetic algorithm requires the definition by the user of a suitable fitness function on the basis of the problem that has to be solved and the choice of appropriate values of the GA algorithm, the most important being the number of individuals in a population, the number of generations and the probability of crossover and mutation operators.

6.1. *I. Slalom maneuver*

The fitness function chosen has the following form:

$$f = a_0\left(f + a_1\left(1 - \left(\frac{x - x_{ref}}{b_1}\right)^2\right) + a_2\left(1 - \left(\frac{y - y_{ref}}{b_2}\right)^2\right) + \right.$$
$$+a_3\left(1 - \left(\frac{z - z_{ref}}{b_3}\right)^2\right) + a_4\left(1 - \left(\frac{v - v_{eff}}{b_4}\right)^2\right) +$$
$$\left.+a_5\left(1 - \left(\frac{\varphi - \varphi_{ref}}{b_5}\right)^2\right) + a_6\left(1 - \left(\frac{\psi - \psi_{ref}}{b_6}\right)^2\right)\right)$$

It is the sum of parabolas, each one representing the error between the reference value and the actual value of a problem variable:

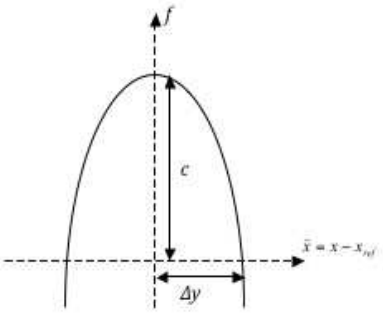

Fig. 4. The fitness function.

The parabolas have the vertex on the f axis, so that:

$$f = c\left(1 - \left(\frac{\hat{x}}{\Delta y}\right)^2\right)$$

where c, called a_i in the Fortran code, is the maximum value of the fitness function, a sort of weight parameter, representing the importance of a design variable with respect to the others, and Δy, b_i in the code, is the error corresponding to a zero fitness function, that gives an indication on how rapidly the fitness function can converge to its maximum value c. Having defined the fitness function in this way, its maximum value must be search by the genetic algorithm to solve the optimization problem. The a_i and b_i factors chosen are:

$$a_0 = a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = 1$$

$$b_1 = b_2 = b_3 = b_4 = 2.5$$

$$b_5 = b_6 = \pi/2$$

Several attempts have been made to find the best fitting between the desired maneuver and the simulated one, varying the algorithm parameters. The following values have shown to be the most suitable:

Table 1. Setting of the genetic algorithm parameters.

Parameter	Value	Parameter	Value
npopsiz	5	ielite	1
nparam	4	nchild	1
maxgen	2500	parmin	-5; -40; -40; -40
iunifrm	1	parmax	40; 40; 40; 40
pcross	0.5	iniche	1
pmutate	0.03	nopsibl	2^{15}
icreep	0		

As shown in (Fig.5), a good matching between the defined trajectory and the one obtained through the analytical simulation has been achieved, also demonstrated in (Fig.6) and (Fig.7), relative to the separated x and y component. The discrepancy on the altitude (Fig.8) remains small too. In (Fig.10) the attitude angles show a time history consistent with the type of maneuver, being the pitch angle almost constant to zero, while the roll and the heading angles following the turns sequence. The same observation can be made about the pilot inputs ((Fig.11) and (Fig.12)), the solution of the optimization problem, that, apart from the lateral cyclic that clearly follows the turns, show small oscillation about the zero value. As it is shown in (Fig.9), the fitness function is positive just in the first seconds of the maneuver, corresponding to the first straight flight along the centerline, while it becomes negative as the helicopter performs the sequence of turns, with increasing magnitude as the maneuver proceeds. The fitness function trend reflects the increasing, even if small, discrepancy between the desired trajectory and the simulated one, as it can be observed

DOI: 10.1685/CSC09261

in (Fig.5) and especially in (Fig.7). This problem could be partially solved by splitting the maneuver sequence and executing the optimization on each interval instead of the entire maneuver duration. In any case the maneuver is reproduced with adequate tolerance. The difficulty that comes out in the reproduction of the maneuver is caused by several factors: the adoption of the linear mathematical model of the rotorcraft, a constant parameters model, the aggressive nature of the slalom maneuver, and finally the fact that the SAS has been build just to ensure the stability of the machine and not specifically for the execution of the slalom.

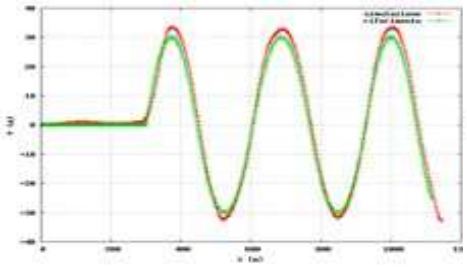

Fig. 5. Slalom: trajectories.

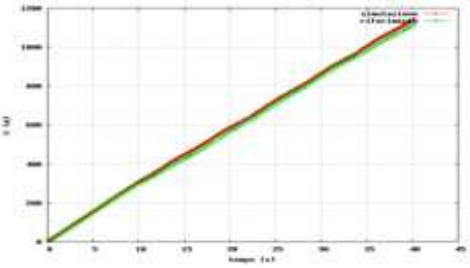

Fig. 6. Slalom: x position.

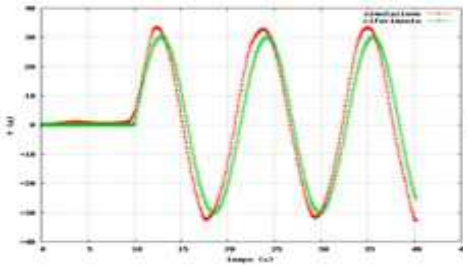

Fig. 7. Slalom: y position.

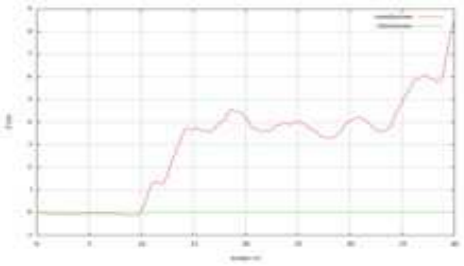

Fig. 8. Slalom: z position.

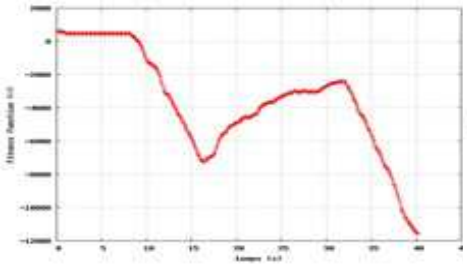

Fig. 9. Slalom: fitness function.

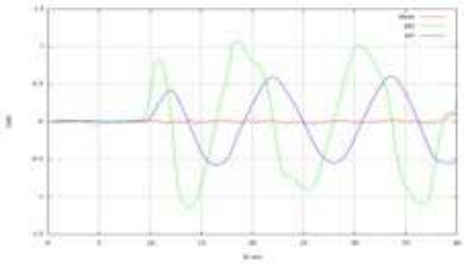

Fig. 10. Slalom: attitudes.

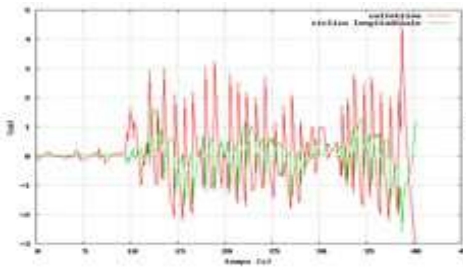

Fig. 11. Collective and long. cyclic.

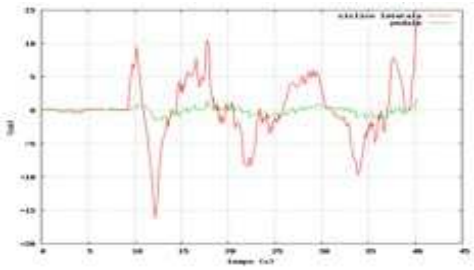

Fig. 12. Slalom: Lat. cyclic and pedal.

6.2. *II. Pirouette maneuver*

The same fitness function and the a_i and b_i factors chosen for the slalom maneuver have been adopted for the pirouette:

$$f = a_0\left(f + a_1\left(1 - \left(\frac{x - x_{ref}}{b_1}\right)^2\right) + a_2\left(1 - \left(\frac{y - y_{ref}}{b_2}\right)^2\right) + \right.$$
$$+a_3\left(1 - \left(\frac{z - z_{ref}}{b_3}\right)^2\right) + a_4\left(1 - \left(\frac{v - \omega \cdot dist}{b_4}\right)^2\right) +$$
$$\left. +a_5\left(1 - \left(\frac{\varphi - \varphi_{ref}}{b_5}\right)^2\right) + a_6\left(1 - \left(\frac{\psi - \psi_{ref}}{b_6}\right)^2\right)\right)$$

The genetic algorithm parameters are also the same. The simulation results exhibit a precise reproduction of the maneuver, as it can be seen in (Fig.13), (Fig.14) and (Fig.15). The time histories of the attitude angles (Fig.18) and the speed components (Fig.16) are also significant: the heading angle varies linearly from 0 deg to 315 deg, following the maneuver, while θ and φ remain almost equal to zero; the lateral speed v is constant to the reference value imposed by the specification, 8 knots (4.11 m/s). Finally, the pilot inputs determined by the genetic algorithm show reasonable trends (Fig.19)(Fig.20): pedal command, and consequently collective, increase in magnitude as the maneuver goes on, in agreement with the heading angle trend; the lateral cyclic does not exhibit wide changes, due to the fact that a constant lateral speed must be hold, while the longitudinal cyclic oscillates about the zero value.

DOI: 10.1685/CSC09261

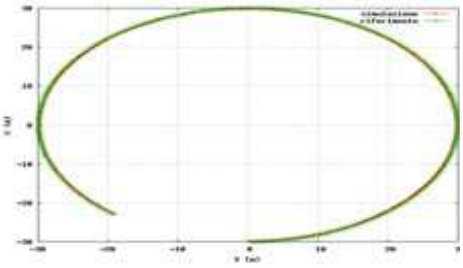

Fig. 13. Pirouette: trajectories.

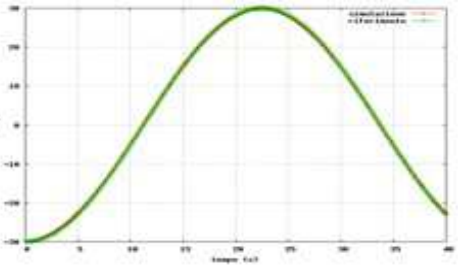

Fig. 14. Pirouette: x position.

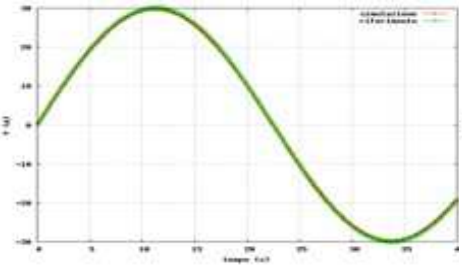

Fig. 15. Pirouette: y position.

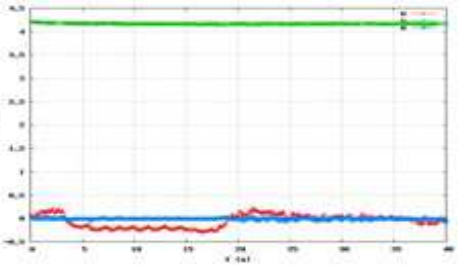

Fig. 16. Pirouette: speed components.

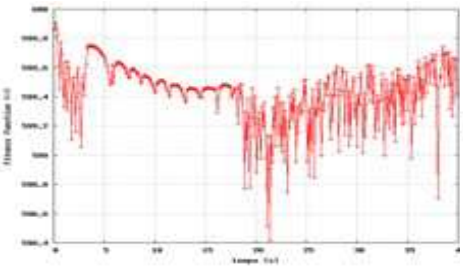

Fig. 17. Pirouette: fitness function.

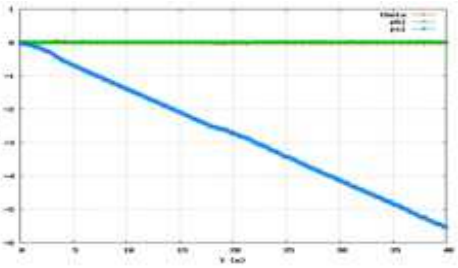

Fig. 18. Pirouette: attitudes.

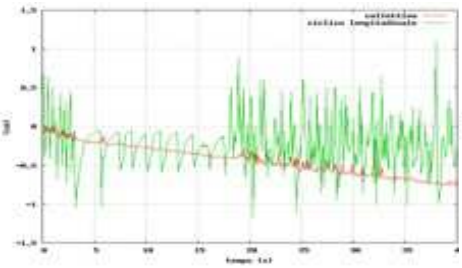

Fig. 19. Pirouette: collective and long. cyclic.

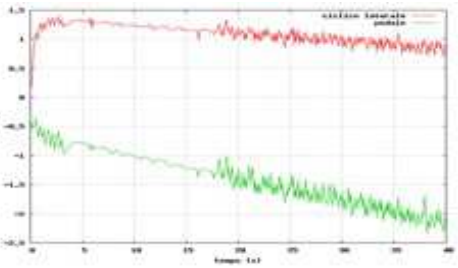

Fig. 20. Pirouette: lat. cyclic and pedal.

7. Concluding remarks

The inverse simulation procedure has been adopted to achieve the reproduction through the analytical simulation of rotorcraft maneuvers. The slalom and pirouette Mission Task Elements described in the ADS-33 have been chosen for this application, although they have been slightly modified with respect to those presented in the specifications. The work has been tackled as an optimization problem and a genetic algorithm has been used as numeric solver. The procedure has provided good results: the simulation of the maneuvers have shown precise tracking of the desired trajectories with reasonable pilot control inputs time histories.
Future work could be addressed to modify the definition of the slalom and pirouette maneuvers according to the ADS-33 specifications and to extend the procedure used to the reproduction of the others MTEs. Further improvement in the precision of the maneuvers reproduction could be achieved by employing a genetic algorithm that uses a real encoding instead of the binary one adopted in the present work.

REFERENCES

1. ADS-33E, Aeronautical Design Standard, Performance Specification and Handling Qualities Requirements for Military Rotorcraft,US Army, 2000.
2. D. L. Carroll, Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers, *Development in Theoretical and Applied Mechanics*, Vol.XVIII, H. Wilson ed.
3. R. Celi, Optimization-Based Inverse Simulation of a Helicopter Slalom Maneuver, *Journal of Guidance, Control, and Dynamics*, **vol.23, No.2**, 2000, pp. 289–297.
4. R. Fantinutto, G. Guglieri, and F. Quagliotti. Flight Control System Design and Optimisation with a Genetic Algorithm, *Aerospace Science and Technology*, **vol.9**, 2005, pp. 73–80.
5. G. Guglieri, B. Pralio and F. Quagliotti. Flight Control System Design for a Micro Aerial Vehicle, *Aircraft Engineering and Aerospace Technology*, **vol.78**, 2006, pp. 87–97.
6. G. Guglieri, F. Quagliotti and G. Speciale. Optimal Trajectory Tracking for an Autonomous UAV, *Automatic Control in Aerospace (on line)*, **vol.1**, 2008, pp. 1–9.
7. R.K. Heffley, A Compilation and Analysis of Helicopter Handling Qualities Data, Volume One: Data Compilation, NASA Contractor Report 3145, Aug. 1979.
8. D.E. Goldberg, Algorithms in Search, Optimization and Machine Learning, Addison Wesley Reading, USA, 1989.
9. D. Whitley, A Genetic Algorithm Tutorial, Technical Report CS-93-103, Colorado State University, 1993.
10. F. Busetti, Genetic Agorithms Overview, w.geocities.com/francorbusetti/gaweb.pdf