

Genetic-Inverse Simulation Algorithm for a Model Helicopter Pilot

Sunny Orike

Department of Electrical/Computer Engineering
Rivers State University of Science and Technology
Port Harcourt, Nigeria
sdorike@gmail.com

Abstract

This paper presents a methodology for optimizing a model helicopter pilot using inverse simulation technique and genetic algorithms. The inverse simulation generates the control inputs required for a desired set of motion outputs, while the genetic algorithm (GA) generates feasible solutions to the inverse problem. The goal of the work is to define tasks for the helicopter and have the model pilot find control settings that carry out those tasks. The pilot was formulated as a multi-objective optimization problem with four objectives imposed as penalties. While minimizing the pilot's work load, the optimization approach targets the highest of four penalties imposed.

Keywords: Algorithm, Fitness, Optimization, Simulation.

INTRODUCTION

The helicopter pilot's inverse simulation technique calculates the pilot's control input required for a particular trajectory the helicopter will fly [1]. This contrasts the conventional approach which involves developing a computational model of the helicopter and calculating its response to a set of pilot inputs [2]. The technique simulates the way a human pilot will fly a helicopter from one steady state to another. The main requirement of this work is to incorporate human-like behavior in the development of a simulated non-human piloted helicopter flight.

Algorithms that could be used to generate feasible solutions to the inverse simulation technique, in this case the helicopter's trajectory, will no doubt help in calculating the pilot's control settings. Embarking on a work of this nature is both a challenging as well as interesting task in artificial intelligence. This is because the helicopter has multiple inputs and multiple controls, the controls are cross-coupled, they have multiple effects and produce motion in various directions and the possibility of imposing constraints in both velocity and acceleration.

Helicopter Maneuver and Pilot's Control

The main distinguishing features of helicopters over other aircrafts are low speed and hover flights. These they do relative to earth's fixed axis. A maneuver is usually composed of two parts, the positional coordinates and the heading angle [3]. The positional coordinates (x_e, y_e, z_e) describe the motion of the helicopter's center of gravity relative to the earth-fixed axes. The heading angle (ψ) describes the nose direction of the helicopter.

The helicopter maneuver in this work is defined in terms of motion specified relative to the earth-fixed axis. The following maneuvers are identified in the literature: Roll [2], acceleration/deceleration, side-steps [3], pop-up [4], turn [5], bob-up, hurdle-hop [6], etc. The current work focuses on acceleration/deceleration maneuver. Deciding on which controls to apply to perform the given maneuver is the basic task of the pilot.

A typical helicopter pilot consists of four modules or controls: collective, lateral cyclic, longitudinal cyclic and pedal. The collective and cyclic controls control the main rotor. The collective increases the rotor thrust and this could bring about vertical flight. The cyclic control allows both forward-backward (longitudinal

cyclic) as well as left-right turning (lateral cyclic) motions. The pedal provides anti-torque to the main rotor, turn co-ordination and turns about a vertical axis, while in a hover [7]. The interaction of these controls constitutes the biggest challenge to the pilot, as an adjustment in one of the controls requires a corresponding adjustment in others.

The Helicopter Model

The helicopter model is represented by the state space equations which are in their original form. The general mathematical model of the motion is given by [8] as:

$$\dot{\underline{x}} = A \underline{x} + B \underline{u} \quad (1)$$

Where \underline{x} , \underline{u} and $\dot{\underline{x}}$ are the state vector, control vector and output vector respectively, and $\dot{\underline{x}}$ is the time derivative of \underline{x} , A is 9x9 linear coefficients of system matrix and B is 9x4 linear coefficients of controls matrix of the helicopter. The elements of the state and control vectors are:

$$\underline{x} = [u, v, w, p, q, r, \phi, \theta, \varphi, u_e, v_e, w_e, x, y, z] \quad (2)$$

$$\underline{u} = [u_1, u_2, u_3, u_4] \quad (3)$$

Where u, v and w are the linear velocities, relative to the body axes; p, q, r , the angular velocities; ϕ, θ, φ , the Euler angles for roll, pitch, yaw, respectively; u_e, v_e, w_e , the velocity components in the earth axes; x, y, z the linear distances over the ground all in the x, y, z directions; u_1 is the collective control; u_2 and u_3 are the lateral and longitudinal controls respectively; u_4 is the pedal control [2].

The structure of a helicopter pilot model is based on three piloting functions, in which a piloting task could be divided into: navigation, guidance and stabilization [3] [9]. All pilot decisions are made at the navigation level. It involves the knowledge of current position and desired future trajectory. The guidance level achieves the velocity and/or position commands set by navigation level and sets appropriate commands for attitude needed in the stabilization level through which control positions are generated [3]. The helicopter is guided to the required trajectory, where the pilot determines the parameters needed to achieve the new trajectory. For stabilization, the pilot makes use of the control stick, for he knows the response of the helicopter to a particular stick input. The helicopter is stabilized around the required attitude.

Genetic Inverse Simulation for Helicopter Flights

Several approaches to the solutions of inverse simulation problems exist in literature. One approach models the helicopter flight as an optimal control problem, achieved through minimizing the difference between the desired and the achieved flight trajectories using a gradient method [1] [10]. This is achieved through two categories of algorithms: differentiation and integration inverse methods. The former uses numerical differentiation to evaluate the time derivations of the states and computes controls directly from the differential equations, while the latter uses Newton's iteration method to calculate the controls step by step. The helicopter's trajectory is discretised, and for a given step with known initial controls, the equations of motion are integrated with estimated controls at the end of the step. The error (difference between actual and desired trajectory), is calculated, and the controls at the end of the step are adjusted using Newton-Raphson technique to reduce the errors to zero [1].

A. Mathematical Computation of the Inverse Simulation Technique

The input-output relationship for the helicopter could be represented as a transformation from a vector u_k , a vector containing the four control inputs in the helicopter model to a vector y_k , a vector containing the resulting vehicle trajectory, where k is the time step:

$$y_k = G[u_k] \quad (4)$$

G is the mapping or transformation function of the algorithm. The desired vehicle trajectory is specified, and could be represented as a vector, y_k^D , with the desired output vector representing the desired vehicle response time histories. From (4):

$$G[u_k] - y_k^D = 0 \quad (5)$$

Employing Newton's method to solve (5), an error vector $F(u_k)$ is defined as:

$$F(u_{k,n}) = G(u_{k,n}) - y_k^D \quad (6)$$

Where n is the iteration number. As n increases, the error vector approaches zero. The Newton's solution to this problem could be written in the form:

$$u_{k,n+1} = u_{k,n} - J[G(u_{k,n})]^{-1} \cdot F(u_{k,n}) \quad (7)$$

J is the Jacobian matrix computed as:

$$J[G(u_{k,n})]_{i,j} = \frac{\partial G_i}{\partial (u_{k,n})_j} \quad (8)$$

$$Y(u)_{i,j} = \frac{\partial y_{i,k+1}}{\partial u_j} \quad (9)$$

A pseudo code for this technique is given as:

Read initial states $x(0)$, inputs $u(0)$, $k = 1$, $n = 0$

While k is not equal to maximum time

Define desired trajectory at present time: y_k^D

Obtain a trajectory output y_k with u_k as input

Compute error as: $y_k - y_k^D$

Obtain Jacobian, J

Compute $u_{k,n+1} = u_{k,n} - J^{-1} \times \text{error}$

Run simulation to obtain the next value of the trajectory output

Evaluate the error function

If error criteria are not met,

Increase n by 1 and compute error

Else

Record $x_{k,n}$ and $u_{k,n}$

Update state variables

Increase k by 1

Stop

End.

This computation generates both the helicopter's trajectory and the required pilot controls needed to fly the trajectory. The impact of inverse simulation is felt in the areas of getting comparative information on pilot workload, handling qualities and helicopter performance.

The Problems with Conventional Inverse Simulation

All dynamic systems, in which the helicopter forms a part, are generally subject to the system model:

$$\dot{x} = f(x, u) \quad (10)$$

And a measurement model:

$$y = C(x) \quad (11)$$

Where: $x \in \mathbb{R}^m$ (State vector);

$u \in \mathbb{R}^m$ (Control vector);

$y \in \mathbb{R}^m$ (Output vector).

The aim of inverse simulation is to find the required control vector. A use is made of initial guesses of the control vector for the forward simulation. The output vector obtained from the simulation is compared with the desired output. The guessed controls are modified based on the errors, and the algorithm is repeated until the simulation results converge to the desired trajectory. One of the problems with conventional inverse simulation for the helicopter pilot is that the process cannot be handled adequately by the mathematical approach.

The system's input-output relationship can be represented as a transformation from a vector \underline{u} in some input space to a vector \underline{y} in an output space, with h as the mapping function:

$$\underline{y}(t) = h \{ \underline{u}(t) \} \quad (12)$$

The whole inverse simulation problem begins with the definition of desired output, \underline{y}^D over a given time interval. The input and output variables for the linear system model are defined as:

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) \quad (13)$$

$$\underline{y}(t) = C\underline{x}(t) \quad (14)$$

In this case, $\underline{u}(t)$ is the variable to be determined, and from (14):

$$\underline{x}(t) = C^{-1} \underline{y}(t) \quad (15)$$

The entire inverse problem begins with the need to specify a desired output (trajectory), therefore, (15) will be correctly written in the form:

$$\underline{x}(t) = C^{-1} \underline{y}^D(t) \quad (16)$$

From (13):

$$\underline{u}(t) = B^{-1} \{ \dot{\underline{x}}(t) - A\underline{x}(t) \} \quad (17)$$

The above solution is obtained at each time step and hence the necessity of calculating the derivatives of the intermediate state variables in iterative fashion at each step using numerical differentiation. This is a very cumbersome task. Moreover, B in this instance is a 9×4 matrix coefficient of $\underline{u}(t)$, and inverting a non square matrix is impossible.

Another problem is error analysis [11]. Generally, the models of all dynamic control systems, are first or second order systems consisting of state variables, input and output variables. In this work, a relationship exists between the outputs and the inputs. The input parameters (controls) depend on the error computed

The maneuver started and stopped in hover. This means that the helicopter is at rest with zero velocity and acceleration at the end of the maneuver. A fourth penalty was imposed on the velocity and acceleration, achieved by summing the squares of velocity and acceleration, given by:

$$p_3 = u_{final}^2 + \dot{u}_{final}^2 \quad (21)$$

While achieving optimality of the model pilot, the following fitness function was made use of.

$$f = \frac{1}{1 + \max(p_1, p_2, p_3, kp_{controls})} \quad (22)$$

The approach investigates the contributions/effects of each penalty towards the overall fitness. Similar to lexicographic approach [12], it returns the highest of the penalties in each run of the algorithm. K is a normalizing constant, which adjusts the value of $p_{controls}$, ensuring that all penalties occupy similar numerical ranges, to reduce oscillation effects in controls.

The following four selections methods were investigated in the parameters tuning using both genetic exploration and exploitation: tournament selection, best selection, fitness proportionate selection and greedy selection [13] [14], from where the choice of fitness proportionate selection was made, with the highest fitness value. Details of genetic algorithm, and parameters tuning were made in earlier and similar work presented in [15].

To prevent dominating effects of one penalty over the others, tolerance factors k_i were introduced, aimed at ensuring that all the penalties were of similar magnitude.

$$f = \frac{1}{1 + \max(k_{i1}p_1, k_{i2}p_2, k_{i3}p_3, k_{i4}p_{controls})} \quad (23)$$

With these tolerance values for each of the penalties, a GA was built for (23). A total of 20,000 fitness evaluations were made, averaged over 30 runs in each case. The controls that produced the highest fitness value were used to test the model.

Results and Discussion

The inverse simulation technique generates controls in an experimental situation. The benefits of developing systems that will carry out inverse simulation are to simulate real maneuvers and evaluate the resulting controls in terms of deviation from the reference trajectory, proximity to the target, and effects on pilot workload.

A comparison was made of the model pilot with a Standard Inverse Simulation (SIS) technique of a smooth trajectory for the straight line task. This comparison is achieved by means of graphs of control movements, distance, linear velocity, angular velocity and pitch attitude. It was found that there is a strong similarity in longitudinal cyclic control in the nature of the shape of longitudinal cyclic (qualitative) and amplitude (quantitative). Fig. 3 shows the graph of longitudinal cyclic controls of the family of five solutions alongside that of SIS which is closely related to pitch attitude.

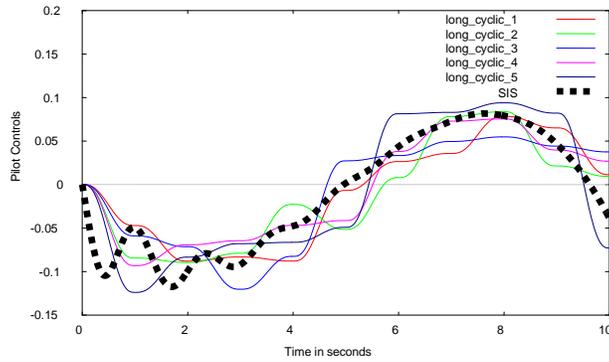


Fig. 3. Longitudinal control movements

Clearly the five test solutions of the model pilot and the SIS show similarity in their trend. The helicopter is initially pushed nose down to accelerate. It is then pulled up after 4 seconds to adjust to the increasing velocity and in the deceleration phase, the helicopter is moved nose up to a maximum pitch of about 5 degrees, and later pushed over into a near hover position. The SIS curve seems to be a best fit curve that can be drawn to represent the five test solutions curves.

Figs. 4 and 5 show the graphs of the x -axis distances and velocities of the helicopter. There is also a strong similarity in longitudinal distance and velocity of the straight line maneuver in the nature of the shape of the distances and velocities (qualitative) and amplitude (quantitative) between the five test solutions and SIS. They all follow the same trend and were able to successfully fly to the target. A single best fit trajectory can be drawn to accurately represent all the trajectories shown in the graph of Fig. 4.

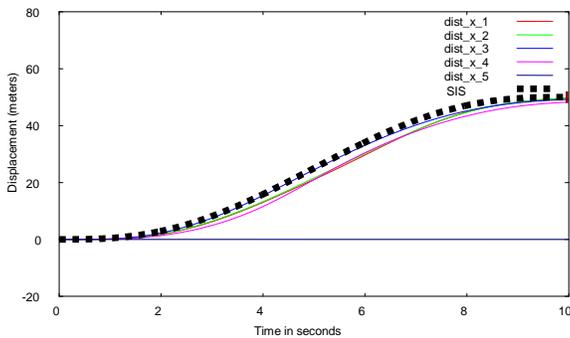


Fig. 4. x -directional axis distances

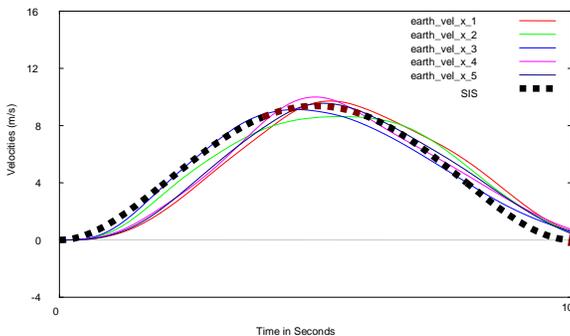


Fig. 5. x -directional axis velocity

Figs. 6 and 7 show the graphs of the y-directional angular velocity and pitch attitude of the helicopter respectively. They all follow the same trend both in the nature and amplitude of the graphs. There is however an observable difference in the curve between the SIS and five test solutions of the model pilot at the last second (Fig. 6). This is as a result of the end condition placed on the pilot controls, causing all of them to converge to zero. Also, though both curves follow the same pattern, there is a slight difference observed in the first, fifth and sixth seconds. These are implementation effects, due to differences in the helicopter modeling techniques.

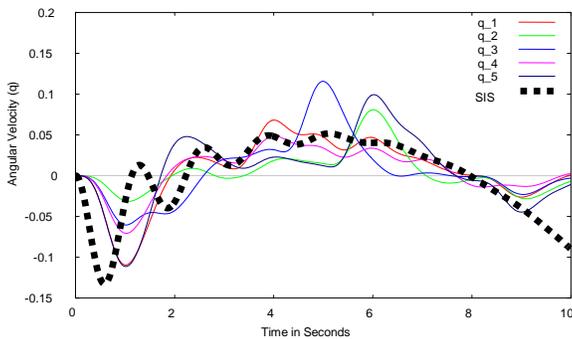


Fig. 6. x – directional axis angular velocity

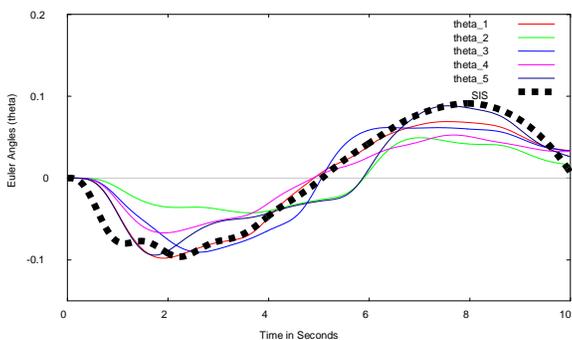


Fig. 7. Pitch attitudes

Conclusion

The work described an optimization approach for a model helicopter pilot combining inverse simulation technique and genetic algorithms, generating feasible solutions as control inputs required for a desired set of motion outputs. The approach targets the contributions of each penalty towards the overall fitness, and returns the biggest of the four imposed penalties in each run. The results show that the maneuver was well flown by the model pilot. Therefore, a combination of inverse simulation and genetic algorithm has great potentials in solving multi-objective optimization problems, exhibiting human-like behavior.

References

- R. Celi, "Optimization-Based Inverse Simulation of a Helicopter Slalom Maneuver". *Journal of Guidance, Control and Dynamics*, vol. 23, no. 2, pp. 289-297, 2000.
- D. G. Thomson and R. Bradley, "The Principles and Practical Applications of Helicopter Inverse Simulation," *Simulation Practice and Theory*, vol. 6, no. 1, pp. 47-70, 1998.

- J. V. Vorst, "A Pilot Model for Helicopter Maneuvers," NLR-TP-98448 Report, National Aerospace Laboratory NLR, presented at 24th European Rotorcraft Forum, Marseilles, France, Sept. 15-17 1998.
- Temeris, M. S. 2003. Automating the Cockpit: Constructing an Autonomous Human-like Flight Bot in a Simulated Environment. Technical Report, Delft University of Technology, Netherlands.
- D. Solinger, "Creating a Dogfight Agent," Technical Report DKS05-01 / ICE 10, Data and Knowledge Systems Group, Faculty of Information Technology and Systems, Delft University of Technology, 2005, version 1.0.
- G. Kowaleczko, "Simulations and Reconstructions of Aircraft Flights and Accidents," Flight Test: Sharing Knowledge and Experience, vol. 22, pp. 1-21, 2005.
- C. Phillips, C.L. Karr and G. Walker, "Helicopter Flight Control with Fuzzy Logic and Genetic Algorithms". *Engineering Applications of Artificial Intelligence*, vol. 9, no. 2, pp. 175-184, 1996.
- T.G.B. Amaral, M.M. Crisostomo, V.F. Pires, "Helicopter Motion Control Using a General Regression Neural Network," in *Computational Intelligence in Control*, M. Mohammadian, R. A. Sarker, X. Yao, Eds. London: Idea Group Publishing, 2003, pp. 41-68.
- G. D. Padfield and M. D. White, "Measuring Simulation Fidelity through an Adaptive Pilot Model," *Aerospace Science and Technology*, vol. 9, no. 5, pp. 400-408, 2005.
- A. Bressan, "Optimal Control Problems on Specified Domains". *American Institute of Mathematical Sciences*, vol. 2, no. 2, pp. 313-331, 2007.
- Whaley, M. S. 1991. Development and Evaluation of an Inverse Simulation Technique for Studying Helicopter Maneuverability and Agility. US Army Aviation Systems Command Technical Report 90-A-008, Moffett Field, CA
- C. A. Coello-Coello, "20 Years of Evolutionary Multi-Objective Optimization: What has been done and what remains to be done," in *Computational Intelligence: Principles and Practice*, G.Y. Yen, D.B. Fogel, Eds., IEEE Computational Intelligence Society, 2006, pp. 73-88.
- Luke *et al.*, 2008. A Java-based Evolutionary Computation and Genetic Programming Research System. Version 20. ECLab, George Mason University.
- Mitchell, M., *An Introduction to Genetic Algorithms*. Cambridge: MIT Press, 1998.
- S. Orike, "Multi-Objective Optimization for a Helicopter Pilot using Genetic Algorithms," in *Proc of the IEEE International Conference on Adaptive Science and Technology*, Accra, Ghana, 14-16 Dec, 2009

Author's Profile:

Sunny Orike is a Lecturer in the Department of Electrical/Computer Engineering, Rivers State University of Science and Technology, Port Harcourt, Nigeria. He holds a B.Tech in Computer Engineering from Rivers State University of Science and Technology, Port Harcourt, Nigeria; an MSc in Computing from Robert Gordon University, Aberdeen, United Kingdom; and currently a PhD student in Heriot-Watt University, Edinburgh, United Kingdom. His research area is Genetic and Evolutionary Algorithms. He is a member of several professional bodies, including: Nigerian Society of Engineers, Nigeria Computer Society, British Computer Society, Association of Computing Machinery and Institute of Electrical Electronics Engineers.