DESIGN OF A LINEAR QUADRATIC OPTIMAL CONTROL FOR AIRCRAFT FLIGHT CONTROL BY GENETIC ALGORITHM

J. Aranda[1], J.M. de la Cruz[2], M. Parrilla[1], P. Ruipérez[1]

[1]*Departamento de Informática y Automática. Facultad de Ciencias U.N.E.D. C) Senda del Rey s/n. 28040-Madrid. Spain. E-mail:* jaranda@dia.uned.es

[2]*Departamento de Arquitectura de Computadores y Automática. Facultad de C. Físicas. U Complutense. Ciudad Universitaria. 28040-Madrid. Spain. Email:* jmcruz@dia.ucm.es

Abstract: In this paper we have investigated the use of genetic algorithm for the selection of weighting matrices of performance index for the linear quadratic control design. We can easily consider in the fitness index of the genetic algorithm different design specifications and their verifications in different operations conditions. As well as a measure of robustness $\sigma(S+T)$ evaluated at the input and at the output for linearized models obtained for different parameters. This technique is applied to design a control for the longitudinal dynamic of an aircraft. The robustness and specifications are evaluated for different values of mass, airspeed, centre of gravity, and transport delay.

KEYWORDS: optimal control, genetic algorithm, aircraft control

1. INTRODUCTION

One of the greatest difficulty to apply new and classical control methods is the effort to learn, to implement and to apply the method, and the problematical that is to re-design with a non-familiar method. The Linear Quadratic method is considered to be relatively easy to use, but requires a good understanding of optimal control theory. Obtaining the desired results by selection of weighting functions will probably provide the greatest difficulty for a new user. Some iterations are necessary (Anderson and Moore, 1992). So, Frangos and Yavin (1992) have proposed an iterative procedure for automating the design of linear optimal control systems, the method was applied to the design of a decoupled lateral control systems for an RPV. Stevens *et al.* (1992) gave an approach for design servo compensators of a desirable structure that is based on LQ output-feedback techniques, with this approach a few parameters must be tuned during the interactive design process. Blight *et al.* (1994) have several applications of multivariable control theory for aircraft control law developments, but the problem is to adjust the penalty.

After those results, we have investigated other possible design methods, that we can apply to other techniques, and with less necessary effort to application and re-design. So we investigated the use of genetic algorithms to design different controllers. Genetic algorithms were explored by Krishnakumar and Goldberg (1992) and Goldberg (1998) as a technique for solving aerospace-related control system optimization problems. They used a genetic algorithm to design a lateral autopilot and a windshear controller. Genetic algorithms were used to optimize a standard linear quadratic regulator problem. In this case, the fitness function for the genetic algorithm was the performance index of the LQR, which was selected to maintain heading and roll attitude. The genetic algorithm was used as an optimization technique to solve the LQR optimization problem.

We want to investigate possible techniques to make us easier to include new design specifications, without being a new problem with different performance index. It is only necessary to choose other weighting matrices, which are selected by a genetic algorithm. So we have applied genetic algorithms techniques to selection of weighting matrices of performance index for the linear quadratic control design (application which is showed in this paper). This technique was extended to the robust selection of the eigenvalues and eigenvectors for the eigenstructure assignment design (which was used, without genetic algorithm, with very good results, de la Cruz *et al*. 1997, Magni *et al*. 1997 ch. 18). The results are very promising since with direct transcriptions and intuitively we can obtain the weighting matrices for the linear quadratic control algorithm. In the next sections we explain the application of genetic algorithm to solve the design of a linear quadratic optimal control for aircraft control, how the weighting matrices are selected and how the design specifications into the algorithms are considered (Davis, 1991; Trebi-Ollennu and White, 1997).

2. LINEAR QUADRATIC OPTIMAL CONTROL

The fundamentals of linear quadratic optimal control theory can be found in the special issue on the LQG problem (see IEEE Trans. on Automatic Control vol. AC-16), since then, many books have been written on this subject (Anderson and Moore, 1992; Lewis, 1986; Mosca, 1995). Many applications have been performed in the aeronautical field, see among others (Stevens and Lewis, 1992; Magni *et al.*, 1997 chapters 4 and 28), a list of aeronautics applications can be found in (Delgado *et al.* 1997).

This control technique allows the designer to take into account both requirements on the amplitude of the control inputs and the settling time of the state variables; moreover, when considering infinite horizon optimization and provided that the weighting matrices are suitable chosen, an important feature of LQ control is that the resulting closed-loop system exhibits very good guaranteed multivariable stability margins. When the complete state is not available for measurement and some or all of the measures are affected by noise, the Kalman optimal filtering theory can be used to design an observer of the state variables; however, the robustness margins are no longer guaranteed in the presence of an observer. If sensor noise is absent or one does not care about it, it is possible to use the degree of freedom on the design of the observer to recover the LQ robustness margins; this is the celebrated Loop Transfer Recovery (LTR) technique (Stein and Athans, 1987), which, however, can be applied only when the plant under considerations is minimum phase. Linear quadratic optimal control performs a trade-off between control amplitudes and setting times; this trade-off is strongly influenced by the choice of the weighting matrices Q and R. Large values of R with respect to Q will result in weak control amplitudes and a slow regulation of the state variables; conversely we have stronger control amplitudes and a faster regulation. The design cycle is usually composed of iterative steps (Amato *et al*., 1997).

3. GENETIC ALGORITHM

Genetic algorithms were originally developed by John Holland (1975) and have been growing since then. This field is changing rapidly and different researches have implemented them in different ways (Davis, 1991; Goldberg, 1989). Genetic algorithms were invented to mimic some of the processes observed in natural evolution. Holland thought that, appropriately incorporated in a computer algorithm, they might yield a technique for solving difficult problems in the way that nature has done through evolution.

Evolution is a process that operates on chromosomes. The process of reproduction is the point at which evolution takes place. Mutations may cause the chromosomes of children to be different from those of their parents, and recombination processes may create quite different chromosomes in the children combining material from chromosomes of two parents. Processes of natural selection cause those chromosomes that encode successful structures to reproduce more often than those that do not. The first difficulty is the encoding solutions to the problems on chromosomes, and the definition of an evaluation function that returns a measurement of the worth of any chromosome in the context of the problem. The technique for encoding the possible solutions may vary from problem to problem and from genetic algorithm to genetic algorithm. In some works the encoding is carried out using bit strings, but other codification can be used too (Goldberg, 1989). With regard to our case of linear quadratic control design, the weighting matrices are usually diagonals. So chromosomes are comprised for number in floating point, which are the elements of main diagonals of weighting matrices.

The evaluation function is the link between the genetic algorithm and the problem to be solved. An evaluation function takes a chromosome as input and returns a number or list of numbers that is a measured of the performance or fitness of chromosome on the problem to be solved. The return values are named fitness index. The evaluation function plays the same role in genetic algorithm than the environment plays in natural evolution. The iteration of an individual with its environment provides a measure of its fitness, and the iteration of a chromosome with an evaluation function provides a measure of fitness that the genetic algorithm uses when carrying out reproduction.

The procedural flow of our genetic algorithm is outlined as follows:

1) The initial generation of chromosomes (elements of main diagonals of weighting matrices) is randomly chosen.
2) From chromosomes the weighting matrices are obtained and by an analytical method the controller gains are calculated.
3) Simulation is conducted and fitness index is evaluated for each gain set.
4) Create new chromosomes by applying selection, crossover and mutations.
5) Delete members of the population to make room for the new chromosomes.
6) Evaluate the new chromosomes: obtain the weighting matrices, calculate the controller gains and evaluate the fitness index for each gain set. Insert the new chromosome into the population
7) If time is up, stop and return the best set gains; if not go to 4.

The purpose of parent selection is to give more reproductive chances, on the whole, to the fittest members of population. There are many ways to do this. One commonly used technique is the roulette wheel parent selection (Davis, 1991): a) sum the fitnesses of all the members of the population (call the result total fitness), b) generate a random number *n* between 0 and total fitness, and c) return the first member of the population whose fitness, added to the fitnesses of the preceding members of the population, is greater than or equal to *n*. Although this selection technique is random, each parent's chance of being selected is directly proportional to its fitness. On balance, over a number of generations this algorithm will drive out the least fit members and contribute to the spread of the genetic material in the fittest members of the population. The mutation is a procedure or operator carried out by reproduction phase. It is simply an occasional random alteration. When mutation is applied some chromosomes elements are randomly replaced. The mutation has an associated probability parameter that is typically quite low. The mutation procedure helps in avoiding the possibility of mistaking a local minimum for a global minimum. When mutation is sparingly used with reproduction and crossover, it improves the global nature of the genetic algorithm search. Another process that alters chromosomes during reproduction and may be at least as important as mutation is the crossover.

Crossover occurs when two parents exchange parts of their corresponding chromosomes. In a genetic algorithm, crossover recombines the genetic material in two parent chromosomes to make two children. A simple crossover follows reproduction in three steps. First, the newly reproduced chromosomes are paired together at random. Second, an integer position *n* along every pair of chromosomes is selected uniformly at random. Finally, based on a probability of crossover, the paired chromosomes undergo crossing over at the integer position *n* along the chromosome. This result in new pairs of chromosomes that are created by swapping all of the characters 1 and *n* inclusively. Although the crossover procedure is a randomised event, when combined with reproduction it becomes an effective means of exchanging information and combining portions of good quality solutions. Crossover is a very important component of a genetic algorithm. Many genetic algorithm practitioners believe that if we delete the crossover operators from a genetic algorithm the result is not a genetic algorithm. In fact, the use of a crossover operator distinguishes genetic algorithms from other optimization techniques. Dynamic programming, for instance, maintains populations of individuals and applies mutation-like operations to them, preserving the best ones.

4. PROBLEM DESCRIPTION

The problem formulation corresponds to the RCAM design problem (Lambretchs *et al.*, 1997) proposed for the GARTEUR Action Group FM(AG08). In this paper, we design a linear quadratic control to verify the specifications of the longitudinal dynamics for the RCAM model. The linear longitudinal model (around the nominal conditions: air speed V=80 m/s, altitude h=1000m, mass=120.000 K, centre of gravity cgx=0.23, cgz=0, transport delay δ=0) is:

$$\begin{pmatrix} \dot{q} \\ \dot{\theta} \\ \dot{u}_B \\ \dot{w}_B \\ \dot{X}_T \\ \dot{X}_{TH} \end{pmatrix} = \begin{pmatrix} -0.9825 & 0 & -0.0007 & -0.0161 & -2.4379 & 0.5825 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -2.1927 & -9.7758 & -0.0325 & 0.0743 & 0.1836 & 19.6200 \\ 77.3571 & -0.7674 & -0.2265 & -0.6683 & -6.4785 & 0 \\ 0 & 0 & 0 & 0 & -6.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.6667 \end{pmatrix} \begin{pmatrix} q \\ \theta \\ u_B \\ w_B \\ X_T \\ X_{TH} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 6.6667 & 0 \\ 0 & 0.6667 \end{pmatrix} \begin{pmatrix} d_T \\ d_{TH} \end{pmatrix}$$

$$\begin{pmatrix} q \\ n_x \\ n_z \\ w_V \\ V_A \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0075 & 0 & -0.0033 & 0.0076 & 0.0187 & 2 \\ -0.2661 & 0 & -0.0231 & -0.0681 & -0.6604 & 0 \\ 0 & -79.8667 & -0.0283 & 0.9996 & 0 & 0 \\ 0 & 0 & 0.9996 & 0.0290 & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ \theta \\ u_B \\ w_B \\ X_T \\ X_{TH} \end{pmatrix}$$

where the states are: pitch rate q, pitch angle θ, x component of inertial velocity in body axis u_B, z component of inertial velocity in body axis w_B, the state corresponding to the first order tailplane model X_T and the state corresponding to the first order engine model X_{TH}. This last state is the sum of the individual engine commands. The inputs are the tailplane deflection δ_T and the throttle position δ_{TH}. The measurements are: pitch rate q, horizontal load factor n_x, vertical load factor n_z, z component of inertial velocity in the vehicle-carried vertical frame w_V, and air speed V_A. The control system is showed in Figure 1.

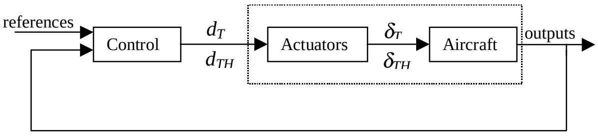

Fig. 1. Flight control system

The design specifications (Lambretchs *et al.*, 1997) are summarised as follows:

- The control system should be able to track:
  - speed commands V_{Ac} with a rise time t_r<12s, a settling time t_s<45s and overshoot M_p<5%.
  - flight path angle command with t_r<5s, t_s<20s y M_p<5%. But γ is neither available as an output nor as a reference signal. To cope with such a problem we use the relation $sin(\gamma) = -\frac{W_V}{V}$, where V is the total inertial velocity. We are thus lead to interpret the specifications in terms of commands in W_V.
- Ride quality criteria: under normal conditions, the vertical acceleration (n_z) should be minimised; it should be less than ± 0.05g. And during a 30-degree turn less than ±0.2g.
- Control activity criteria:
  - Throttle limits (saturation): $0.5\frac{\pi}{180}\,\text{rad} \le \delta_{TH} \le 0.5\frac{\pi}{180}\,\text{rad}$
  - Rate limits for throttle movement are: rising slew rate = $1.6\frac{\pi}{180}\,\text{rad/s}$, falling slew rate = $-1.6\frac{\pi}{180}\,\text{rad/s}$
  - Saturation of tailplane deflection: $-25\frac{\pi}{180} \le \delta_T \le 10\frac{\pi}{180}\,\text{rad}$
  - Rate limits for tailplane deflection: $-15\frac{\pi}{180} \le \dot{\delta}_T \le 15\frac{\pi}{180}\,\text{rad/s}$

5. DESIGN

The control structure is shown in Figure 2. The controller has two parts: a static gain acting on the states measured and a stating gain acting on the integral of the errors (compensator block in Figure 2) in the commanded variables w_V and V_A to eliminate steady state errors.

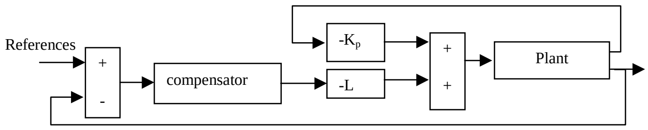

Fig. 2. Longitudinal autopilot control scheme

The gains are selected to minimize a quadratic performance index, whose weighting matrices are chosen by a genetic algorithm to verify the design specifications (performance, robustness, ride quality and control activity). The genetic algorithm (described before) was implemented as MATLAB functions.

The evaluation of the fitness of new chromosomes is the key to obtain good results. In our case, the evaluation function contains the specifications and criteria of design. The evaluation function returns a fitness index, which is a quantitative measure of the fitness of chromosome to the problem solution. We considered criteria about transient response characteristic to command signals and cross coupling constraints for W_V and V_A (overshot Mp, output negative evolution, rise time t_r, stationary error e_s), control saturation and control rate limits

Furthermore, we can applied the evaluation function to other linear models around different operation conditions, and penalise chromosomes whose gains may make the system unstable. Each element in the evaluation function is normalised, so all elements have the same *weight* on the fitness index. Weight constants are calculated to normalisation. So the product between each constant and maximum value of each element is one.

The fitness index is a weight sum of all those specifications terms. The fitness index may include values of those specifications terms in other operation conditions To consider, a measure of robustness σ(S+T) is evaluated at the input and at the output for linearized models obtained for different values of parameters. The sum of σ(S+T) for each model and condition is included in the fitness index.

The crossover rate is around 40% of the population and the mutation rate is around 0.8%. These parameters can be change. As well as the size of the population and the number of generations to achieve the optimum. We have studied which is the best size of the population as well as the best number of generations for this kind of problem. Figure 3 shows the optimum fitness versus the size of population, and Figure 4 shows the optimum fitness versus the number of generations. We can see that it is enough with a number of generations around 35 and the size of the population with around 50 chromosomes. The weighting matrices Q and R for the better fitness index are:

$$Q = \mathrm{diag}\{3*10^{0} \quad 3*10^{21} \quad 2.4*10^{17} \quad 4.8*10^{6} \quad 9*10^{5} \quad 4.7*10^{9} \quad 2.4*10^{17} \quad 2.4*10^{16}\}$$

$$R = \mathrm{diag}\{3.7*10^{20} \quad 6.6*10^{20}\}$$

For these weighting matrices, the gains K_p and L (Figure 2) obtained with the Linear Quadratic Optimal control are:

$$K_p = \begin{bmatrix} -1.5852 & -6.3884 & -0.0031 & 0.0398 & 0.4427 & -0.3360 \\ 0.0781 & 0.6411 & 0.0375 & -0.0054 & -0.0188 & 0.8120 \end{bmatrix}$$

$$L = \begin{bmatrix} -0.02421 & -0.002501 \\ 0.005922 & -0.005732 \end{bmatrix}$$

Figures 5 and 6 show the responses when a controller is used with those gains. The responses satisfy all the requirements. Figure 5 shows the response to a step of 13m/s in commanded airspeed V_A. The cross coupling to W_V is little, the maximum deviation in flight path angle is smaller than 0.2deg. There is an overshoot and the rise time and the settling time are shorter than specified. The vertical acceleration n_z remains smaller than the specified 0.05g. Figure 6 shows the response to a step in commanded vertical velocity W_V corresponding to –3deg in commanded flight path angle. The overshoot, the rise time, the settling time and the cross coupling are verified too. The vertical acceleration n_z surpasses the specified in normal flight but it is smaller than the one specified for the manoeuvre. Tables 1 and 2 show the gain and phase margins obtained from the sensitivity functions for different models. Very good stability margins are obtained.

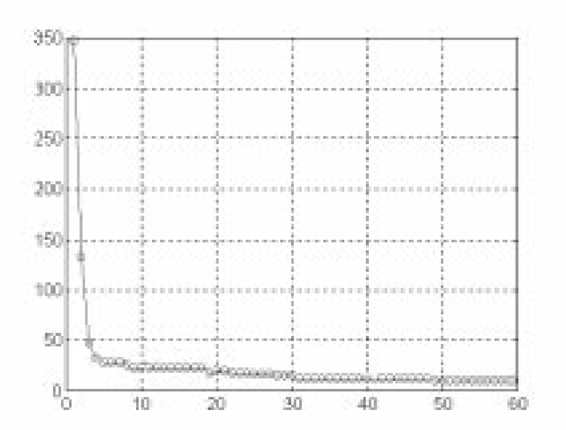

Fig. 3. Optimum fitness versus population size

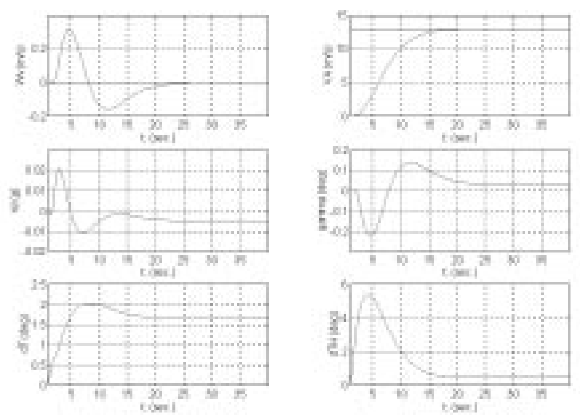

Fig. 5. Response to a step in V_A

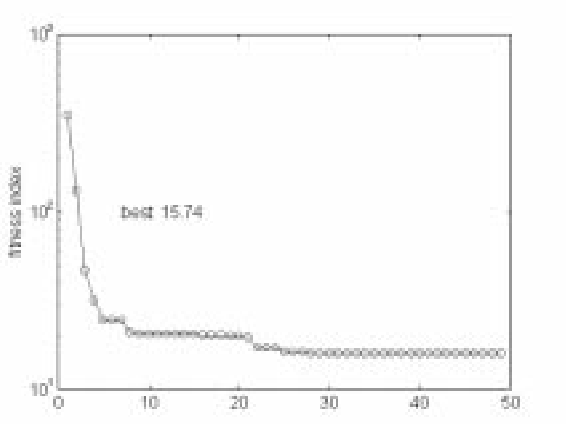

Fig. 4. Optimum fitness versus number of generations

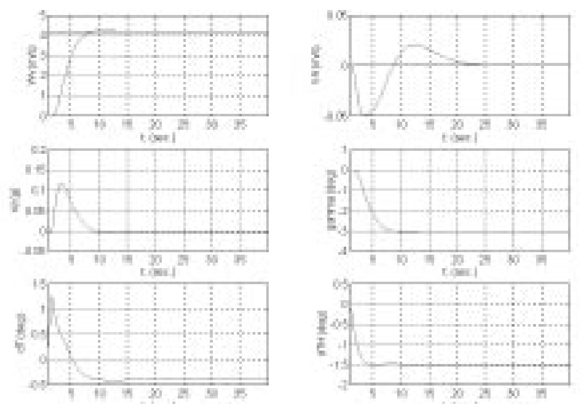

Fig. 6. Response to a step in W_V.

Table 1. Stability margins at the inputs

	Linear model for Nominal parameters		Linear model for Minimum parameters		Linear model for Maximum parameters	
Function	Gain margin	Phase margin	Gain margin	Phase margin	Gain margin	Phase margin
S	[-6.0, 38.1]	±59.2	[-6.0, 38.2]	±59.2	[-5.9, 36.4]	±59.0
T	[-60.8, 6.0]	±59.9	[-28.0, 5.8]	±57.4	[-46.9, 6.0]	±59.7
S+T	[-38.1, 38.1]	±88.6	[-30.0, 30.0]	±86.37	[-36.4, 36.4]	±88.3

Table 2. Stability margins at the outputs

	Linear model for Nominal parameters		Linear model for Minimum parameters		Linear model for Maximum parameters	
Function	Gain margin	Phase margin	Gain margin	Phase margin	Gain margin	Phase margin
S	[-6.0, 38.0]	±59.2	[-6.0, 38.7]	±59.2	[-5.9, 36.3]	±60.0
T	[-60.4, 6.0]	±60.0	[-62.8, 6.0]	±59.9	[-67.0, 6.0]	±60.0
S+T	[-38.1, 38.1]	±88.6	[-30.0, 30.0]	±86.4	[-36.4, 36.4]	±88.3

6. CONCLUSION

Though the linear quadratic method is considered to be relatively easy to use, requires a good understanding of optimal control theory, and the selection of correct weight matrices provide the greatest difficulty. In this paper we have showed that by a genetic algorithm we can easily obtain the correct weight matrices to applied the linear quadratic algorithm. Furthermore, we can easily consider in the fitness index of the genetic algorithm different design specifications and their verifications in different operations conditions. As well as we consider a measure of robustness σ(S+T) evaluated at the input and at the output for linearized models obtained for different values of parameters. In summary, with direct transcriptions and intuitively we can easily obtain the weighting matrices for the linear quadratic algorithm. This technique was extended to the robust selection of the eigenvalues and eigenvectors for the eigenstructure assignment design. So their selection, that traditionally was made by iterative and heuristic methods, was made easily.

ACKNOWLEDGEMENT

Part of this work has been performed under project CICYT TAP97-0607-C03.

REFERENCES

Special issue on the LQG problem, February 1971. *IEEE Transaction on Automatic Control*, **VOL AC-16**.

A.Amato, M.Mattei and S.Scala (1997). *HIRM design challenge presentation document: the LQ methods and eigenstructure assignment approach*. Technical Publication TP-088-26, Group for Aeronautical Research and Technology in Europe GARTEUR-FM(AG08).

B.D.O. Anderson and J.B. Moore (1992). *Linear Optimal Control*. Prenticel Hall.

J.D: Blight, R.L. Dailey and D. Gangsaas (1994). Practical control law design for aircraft using multivariable techniques. In: *Int. J. Control*. **Vol. 59 no. 1**.

J.M. de la Cruz, P. Ruipérez and J. Aranda (1997). *RCAM design challenge presentation document: an eigenstructure assignment approach*. Technical Publication TP-088-22, GARTEUR-FM(AG08).

L. Davis (1991). *Handbook of Genetic Algorithm*. Van Nostrand Reinhold.

I. Delgado et al. (1997). *Robust control techniques tutorial document*. Technical Publication TP-088-7, Group fo Aeronautical Research and Technology in Europe GARTEUR-FM(AG08).

C. Frangos and Y. Yavin. (1992). Design Methodology for Linear Optimal Control Systems. In: *J. Guidance*, **vol. 15, no. 5**. Sept.-Oct.

D.E. Goldberg. (1989) *Genetic Algorithm in Search, Optimization, and Machine Learning*. Addison Wesley.

D.E.. Goldberg. (1998). *A Meditation on the Applicatrion of Genetic Algorithms*. IlliGAL Report No. 98003. Feburary 1998. Departament of General Engineering. University of Illinois.

J.H. Holland. (1975) *Adaptation in Natural and Artificial Systems*. ANN Arbor: The University of Michigan Press.

K. Krishnakumar and D.E. Goldberg. (1992). Control System Optimization Using Genetic Algorithms. In: *Journal of Guidance, Control, and Dynamics*. **Vol. 15, No. 3**, May-Jun.

P.F. Lambrechts et al. (1997) *Robust flight control design chalenge problem formulation and manual: the research civil aircraft model (RCAM)*. Technical Publication TP-088-3, GARTEUR-FM(AG08).

F.L. Lewis. (1986). *Optimal Control*. Wiley-Interscience Publication.

J.F.Magni, S. Bennani and J. Terlow (Eds). (1997). *Robust Flight Control. A Design Challenge*. Springer Verlag.

E. Mosca. (1995). *Optimal, Predictive and Adaptive Control*. Prentice Hall.

G. Stein and M. Athans. (1987). The LQG/LTR procedure for multivariable feedback control design. In: *IEEE Transactions on Automatic Control*, **Vol. 33, no. 2**. Pp. 105-114.

B.L. Stevens and F.L. Lewis. (1992). *Aircraft Control and Simulation*. Wiley-Interscience.

B.L. Stevens, F.L. Lewis and F. Al-Sunni. (1992). Aircraft Flight Controls Design Using Output Feedback. In: *Journal of Guidance, Control, and Dynamics*. **Vol. 15, No. 1**. Jan.-Feb.

A. Trebi-Ollennu and B.A. White. (1997). Multiobjetive fuzzy genetic algorithm optimization approach to nonlinear control system design. In: *IEE Proc.-Control Theory Appl*., **Vol. 144, No. 2**. March.