Research Article

Flight Control Laws Verification Using Continuous Genetic Algorithms

A. Al-Asasfeh,¹ N. Hamdan,² and Z. Abo-Hammour³

¹ *Department of Mechanical Engineering, The University of Jordan, P.O. Box 962069, Sport City, Amman 11196, Jordan*

² *Mechanical Engineering Department, King Faisal University, Al Hofuf, Saudi Arabia*

³ *Department of Mechatronics Engineering, The University of Jordan, Amman, Jordan*

Correspondence should be addressed to N. Hamdan; naderhamdan@hotmail.com

Received 14 December 2012; Accepted 7 January 2013

Academic Editors: A. Hamzaoui and K. Watanabe

Copyright © 2013 A. Al-Asasfeh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work is concerned with the application of a continuous genetic algorithm (CGA) to solve the nonlinear optimization problem that results from the clearance process of nonlinear flight control laws. The CGA is used to generate a pilot command signal that governs the aircraft performance around certain points in the flight envelope about which the aircraft dynamics were trimmed. The performance of the aircraft model due to pitch and roll pilot commands is analyzed to find the worst combination that leads to a nonallowable load factor. The motivations for using the CGA to solve this type of optimization problem are due to the fact that the pilot command signals are smooth and correlated, which are difficult to generate using the conventional genetic algorithm (GA). Also the CGA has the advantage over the conventional GA method in being able to generate smooth solutions without the loss of significant information in the presence of a rate limiter in the controller design and the time delay in response to the actuators. Simulation results are presented which show superior convergence performance using the CGA compared with conventional genetic algorithms.

1. Introduction

A validation and verification (clearance) process of flight control laws is required to prove and guarantee that the aircraft response is safe and stable for any possible failure case such as engine or actuator failures. In addition, the clearance process must take into account possible variations of flight parameters (e.g., large variations in mass, inertia, center of gravity positions, highly nonlinear aerodynamics, aerodynamic tolerances, and air data system tolerances). Also it is required to prove that pilot commands will not drive the aircraft response to critical operating points. It is noted that the aircraft flight quality and performance requirement are specified in the form of sets of stability performance and handling requirement criteria [3]. Consideration of all of these requirements makes the clearance process computationally complex, time consuming, and extremely expensive [2]. Therefore, the approach commonly used by investigator is to clear each performance/handling criterion individually. For a given performance/handling criterion, the clearance

process requires finding for all possible configurations and for all combinations of parameter variations, and uncertainties, the worst-case scenario that violates the specified criterion in a specified flight envelope [3].

To date, very little research has been reported in the literature on the flight control law (FCL) clearance for time-varying pilot command inputs. Skoogh et al. [4] used a parameterization of pilot command inputs where the pilot signals are represented as low-order piecewise polynomials. Menon et al. [5] employed a sequence of discretized pilot command inputs and a sequence of pilot command inputs called Clonk. The resulting nonlinear optimization problems by using the above-mentioned signals were solved using the genetic algorithm and the differential evolution algorithm. These two approaches apply many restrictions on the shape of the pilot command inputs. Due to the presence of a rate limiter in the controller design and the time delay in response of the actuators, it is suitable to use smooth and correlated commands. Other investigators treated the pilot command inputs as predetermined step inputs and

FIGURE 1: Block diagram for a typical genetic algorithm.

FIGURE 2: The initialization functions used in the continuous genetic algorithm: (a) modified Gauss function and (b) modified Sinc function.

evaluated the clearance criteria based on evaluation of system parameter uncertainties.

The continuous genetic algorithm (CGA), explained in the next section, is one of the stochastic population methods for global optimization which operates on a population of solution vectors [6]. It is an efficient method for the solution of optimization problems in which the parameters to be optimized are correlated with each other or the smoothness of the solution curve should be achieved. The advantage of this approach is the ability to find the global optimal solution for nonlinear optimal problem having multiple solutions such as the clearance process found in flight control validation. The advantages of using this method, in comparison with other available techniques, to solve optimal control problems have been demonstrated in several linearized model of control problems such as chemical reactors, tracking systems, and multidegree-of-freedom manipulator [7]. Based on the previous work in applying this technique to solve the above-mentioned optimal control problems [8], the present work will explore applying this technique to the analysis of

the nonlinear flight control laws clearance process. To the authors knowledge, the application of this technique to flight control laws clearance process has not been addressed in the open literature.

Military aircrafts are usually, due to performance reasons, naturally unstable, and therefore a flight control system which provides a required artificial stability is essential for their operation. Because the performance of the aircraft is dependent on the controller, and this performance affects the safety of the pilot and reliability of the aircraft structure and components, it must be proven to the aviation authorities that a controller performance will be correctly acceptable throughout a specified flight envelope in all possible failure conditions and in the presence of all possible system parameter variations [1, 2].

2. Continuous Genetic Algorithm

2.1. General Description. Genetic algorithms (GAs) are adaptive heuristic search algorithms premised on the evolutionary

FIGURE 3: Crossover process: (a) first parent, (b) second parent, (c) crossover function, (d) first child, and (e) second child.

ideas of natural selection and genetics [9]. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of the survival of the fittest. As such, these algorithms represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 1960s [9], genetic algorithms have been widely studied, experimented, and applied in many fields in the engineering world. Not only do GAs provide an alternative method to solving problem, but also outperform other traditional methods in most of the problems considered. Many of the real-world problems

have proven difficult for traditional methods but ideal for GAs.

The GA procedural steps are illustrated in Figure 1. An initial population is created containing a predefined number of individuals (or solutions), incorporating the variable information. Each individual has an associated fitness measure, typically representing an objective value. The concept that the fittest (or best) individuals in a population will produce fitter offspring is then implemented in order to reproduce the next population. Selected individuals are chosen for crossover at each generation, with an appropriate mutation factor to randomly modify the genes of an individual in order to develop the new population. The result is another

FIGURE 4: Mutation process: (a) offspring, (b) mutation function, and (c) offspring after mutation.

FIGURE 5: Principal layout of the control surface configuration.

set of individuals based on the original population leading to subsequent populations with better (min. or max.) individual fitness. Therefore, the algorithm identifies the individuals with the high fitness values, and those with lower fitness will naturally get discarded from the population [9, 10].

The CGA was shown to be an efficient method for the solution of optimization problems in which the parameters to be optimized are correlated with each other or the smoothness of the solution curve should be achieved. As indicated above, this procedure has been successfully applied in the motion planning of robot manipulators in the field of robotics [6], in the numerical solution of boundary value problems in

TABLE 1: The selected operating points in flight envelope.

Point no.	Speed (Mach)	Altitude (m)
1	0.3	1000
2	0.6	6000
3	0.8	3000
4	1	5000
5	1.2	3000
6	1.2	6000

TABLE 2: Maximum load factor (n_{zmax}) results obtained using a modified Gauss function and the population size of 64 individuals.

Mach	Alt.	Run 1		Run 2		Run 3		Run 4		Max value
		Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	
0.3	1000	158	10.5582	189	10.4209	319	10.5693	138	10.4263	10.5693
0.6	6000	368	10.4260	133	10.4495	169	10.6645	256	10.3020	10.6645
0.8	3000	213	10.6005	301	10.5697	257	10.4085	179	10.5354	10.6005
1	5000	202	10.5085	154	10.2681	320	10.4363	176	10.4696	10.5085
1.2	3000	189	10.4260	401	10.4495	188	10.6645	147	10.3020	10.6645
1.2	6000	198	10.4260	180	10.4495	152	10.6645	137	10.3020	10.6645

TABLE 3: Same as Table 2, but using modified Sinc function.

Mach	Alt.	Run 1		Run 2		Run 3		Run 4		Max value
		Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	
0.3	1000	160	10.4049	295	10.4432	313	10.4736	171	10.4206	10.4736
0.6	6000	142	10.4843	275	10.5461	250	10.4613	177	10.4852	10.5461
0.8	3000	269	10.4258	401	10.5404	195	10.6252	137	10.6064	10.6252
1	5000	155	10.4702	225	10.4961	143	10.3488	246	10.6615	10.6615
1.2	3000	215	10.5534	262	10.4596	245	10.8290	214	10.4406	10.8290
1.2	6000	194	10.6149	184	10.4595	209	10.3111	310	10.5633	10.6149

TABLE 4: Same as Table 2, but for a population size of 128 individuals.

Mach	Alt.	Run 1		Run 2		Run 3		Run 4		Max value
		Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	
0.3	1000	143	10.6429	183	10.3990	200	10.7130	208	10.6184	10.7130
0.6	6000	139	10.6506	381	10.8285	207	10.6122	195	10.5630	10.8285
0.8	3000	285	10.7417	293	10.6393	228	10.6686	256	10.9005	10.9005
1	5000	180	10.8291	202	10.6537	196	10.8093	239	10.8391	10.8391
1.2	3000	189	10.6367	401	10.5831	188	10.8135	147	10.7866	10.8135
1.2	6000	152	10.4815	168	10.6388	247	10.8049	190	10.5147	10.8049

the field of applied mathematics [7], and as optimal control problem solver for both linear and nonlinear systems [8].

Before going to the detailed description of the CGA, the conditions concerning the continuous functions that can be used in such algorithms should be clearly stated. In relation to the initialization function, any smooth function that is close enough to the expected solution curve can be used. It is to be noted that the closer the initialization function to the final solution, the faster the convergence speed since the coarse tuning stage of CGA in this case will be bypassed and the algorithm will jump to the fine tuning stage. In the case that there is no prior information about the expected solution curves, then any smooth function can be used. Furthermore, a mixture of functions can be used which is beneficial in this case as it allows one to obtain a diverse initial population. The effect of the initial population usually dies after few tens of generations and the convergence speed after that is governed by the selection mechanism, crossover and mutation operators [8]. The crossover function, as shown later on, is within the range $[0, 1]$ such that the offspring solution curve will start with the solution curve of the first parent and gradually change its values until they

reach the solution curve of the second parent at the other end. The mutation function may be any continuous function. However, both the crossover and mutation functions should satisfy any problem-specific constraints if such constraints exist.

2.2. CGA Steps. The continuous genetic algorithm consists of the following main steps.

(1) *Initialization.* In this phase, an initial population comprising of N_p smooth individuals is randomly generated. In this work, two smooth functions are used for initializing the population: the modified Gaussian function and the modified Sinc function (Figures 2(a) and 2(b)). The modified Gaussian function $P_j(h, i)$ is defined as follows:

$$P_j(h, i) = u_{\text{left}}(h) + \frac{u_{\text{right}}(h) - u_{\text{left}}(h)}{N_t - 1} (i - 1) + A * \exp \left[\frac{-(i - \mu)^2}{2\sigma^2} \right], \quad (1)$$

TABLE 5: Maximum load factor ($n_{z\max}$) results obtained using a modified Sinc function and the population size of 128 individuals.

Mach	Alt.	Run 1		Run 2		Run 3		Run 4		Max value
		Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	Iter. no.	n_z	
0.3	1000	275	10.6571	299	10.5299	204	10.5714	401	10.7866	10.5092
0.6	6000	255	10.4802	237	10.7215	216	10.8129	210	10.5034	10.5174
0.8	3000	189	10.3804	401	10.5431	188	10.7358	147	10.3020	10.6630
1	5000	401	10.6885	223	10.6571	211	10.4816	294	10.5633	10.5110
1.2	3000	122	10.5953	148	10.7304	278	10.8423	136	10.5147	10.5034
1.2	6000	235	10.6007	204	10.7166	280	10.6554	381	10.7092	10.7092

FIGURE 6: The selected operating points in flight envelope.

FIGURE 7: Maximum load factor ($n_{z\max}$) versus the iteration (generation). Result using CGA with population sizes of 64 and 128 and both initialization functions: modified Gauss function (G) and modified Sinc function (S). Aircraft model trimmed at 1000 m altitude and 0.3 Mach speed.

FIGURE 8: Same as Figure 7, but the aircraft model trimmed at 3000 m altitude and 0.8 Mach speed.

FIGURE 9: Same as Figure 7, but the aircraft model trimmed at 3000 m altitude and 1.2 Mach speed.

and the modified Sinc function $P_j(h, i)$ is given by the following equation:

$$P_j(h, i) = u_{\text{left}}(h) + \frac{u_{\text{right}}(h) - u_{\text{left}}(h)}{N_t - 1} (i - 1) + \text{sinc}(x_s - i) * i, \quad (2)$$

where $1 \leq h \leq N_c$, $1 \leq i \leq N_t$, $P_j(h, i)$ is the i th node value of the h th curve for the j th parent, $u_{\text{left}}(h)$ is the value of the leftmost variable of the h th curve, $u_{\text{right}}(h)$ is the value of the rightmost variable of the h th curve, $u_{\text{left}}(h)$ and $u_{\text{right}}(h)$ are randomly generated subject to the problem constraints, N_t is the total number of nodes along each

solution curve, A represents a random number within the range $[-3R(h), 3R(h)]$, with $R(h) = |u_{\text{right}}(h) - u_{\text{left}}(h)|$, μ is a random number within the range $[1, N_t]$, and σ is a random number within the range $[1, N_t/6]$, $x_s = f(x_v, N_t)$, index series at which to interpolate, and x_v is random number between 1 and 5.

(2) *Evaluation.* The fitness (F) is calculated for each individual in the population. In the present clearance process, F is given by

$$F = \max(\xi), \quad (3)$$

where ξ is the angle of attack or the load factor.

FIGURE 10: Same as Figure 7, but the aircraft model trimmed at 5000 m altitude and 1.0 Mach speed.

FIGURE 11: Same as Figure 7, but the aircraft model trimmed at 6000 m altitude and 0.6 Mach speed.

(3) *Selection*. In this step, individuals from the current population are selected to mate according to their relative fitness so that best individuals receive more copies in subsequent generations.

(4) *Crossover*. Crossover is the way through which information is shared among the population. The crossover process combines the features of two parent individuals, say j and k , to form two children individuals, say L and $L + 1$, as given by the following equations:

$$C_L(h, i) = W(h, i) P_j(h, i) + [1 - W(h, i)] P_k(h, i)$$

$$C_{L+1}(h, i) = [1 - W(h, i)] P_j(h, i) + W(h, i) P_k(h, i)$$

$$W(h, i) = 0.5 \left[1 + \tanh \left(\frac{i - \mu}{\sigma} \right) \right] \quad (4)$$

for all $1 \leq h \leq N_c$, $1 \leq i \leq N_t$, where P_j and P_k represent the two parents chosen from the mating pool, C_L and C_{L+1} are the two children obtained through crossover process, W represents the crossover weighting function within the range $[0, 1]$, μ is a random number within the range $[1, N_t]$, and σ is a random number within the range $[1, N_t/6]$.

The crossover operator in CGA is applied in the same way as in conventional GA except that smooth continuous representation of the individuals is used in the CGA while in

FIGURE 12: Same as Figure 7, but the aircraft model trimmed at 6000 m altitude and 1.2 Mach speed.

the GA binary code representation for individuals is used. In the CGA, pairs of individuals are crossed with probability P_{ci} . Within the pair of parents that should undergo crossover process, individual curves are crossed with probability P_{cc} . That is, the h th smooth curve of the first parent is crossed with the h th smooth curve of the second parent with P_{cc} probability. If P_{ci} value is set to 0.5 and P_{cc} value is set to 0.5, then one pair of parents between two pairs is likely to be crossed, and within that pair, $N_c/2$ of the curves are likely to be crossed. Figure 3 shows the crossover process between two random solution curves. It is clear that new information is incorporated in the children while maintaining the smoothness of the resulting solution curves.

(5) *Mutation*. The function of mutation is used to introduce occasional perturbations to the variables to maintain the diversity within the population. The mutation process in CGA is governed by the following formulas:

$$C_j^m(h, i) = C_j(h, i) + d \times M(h, i),$$

$$M(h, i) = \exp \left[\frac{-(i - \mu)^2}{2\sigma^2} \right], \quad (5)$$

where C_j represents the j th child produced through the crossover process, C_j^m is the mutated j th child, M is the Gaussian mutation function, d represents a random number within the range $[-\text{range}(h), \text{range}(h)]$, with $\text{range}(h)$ representing the difference between the minimum and maximum values of the h th smooth curve of child C_j , and μ and σ are as given in the crossover process.

In the mutation process, each individual child undergoes mutation with probability P_{mi} . However, for each child that should undergo a mutation process, individual curves are

mutated with probability P_{mc} . If P_{mi} value is set to 0.5 and P_{mc} value is set to 0.5, then one child out of two children is likely to be mutated, and within that child, $N_c/2$ of the solution curves are likely to be mutated. Figure 4 shows the mutation process in a solution curve of a certain child. As in the crossover process, some new information is incorporated in the children while maintaining the smoothness of the resulting solution curves.

(6) *Replacement*. In this step, the parent population is totally or partially replaced by the offspring population depending on the replacement scheme used. This completes the “life cycle” of population.

(7) *Termination*. CGA is terminated when some convergence criterion is met. Possible convergence criteria are the fitness of the best individual so far found exceeds a threshold value, the maximum number of generations is reached, or the progress limit (the improvement in the fitness value of the best member of the population over a specified number of generations being less than some predefined threshold) is reached. After terminating the algorithm, the optimal solution of the problem is the best individual so far found.

To summarize the evolution process in CGA, an individual is a candidate solution of the required solution curves; that is, each individual consists of N_c solution curves, each consisting of N_t variables. This results in a two-dimensional array of the size $N_c \times N_t$. The population undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed with probability P_{ci} . Within that pair of parents, individual solution curves are crossed over with probability P_{cc} . This process results in an offspring generation where every individual child undergoes mutation with probability P_{mi} . Within that child, individual solution curves are mutated with probability P_{mc} . After that, the next generation is produced according to the replacement strategy applied.

This process is repeated till the convergence criterion is met where the N_c solution curves of the best individual are the required solution curves. The final goal of discovering the required solution curves is translated into finding the fittest individual in genetic terms. It is to be noted that the two functions used in the initialization phase of the CGA will smoothly oscillate between the two ends with a maximum number of single oscillations. If the final solution curves have more smooth oscillations than one oscillation, then this will be done during the crossover and mutation mechanisms throughout the evolution process. This is actually done by those two operators during the run of the CGA while solving a problem. However, the evaluation step in the CGA will automatically decide whether they have rejected or accepted modification because of their fitness function value.

In addition to the previous operator, the elitism operator is introduced to enhance the performance of the algorithm. The preservation of the best solution or solutions and moving it or them to the next generation is vital to the effectiveness of GA [11, 12]. Elitism is utilized to ensure that the fitness of the best candidate solution in the current population is larger than or equal to that in the previous population.

FIGURE 13: Pitch and roll pilot command obtained from CGA represents the pilot command that generates the maximum load factor ($n_{z \max}$) value. The aircraft model trimmed at 1000 m altitude and 0.3 Mach speed.

FIGURE 14: Same as Figure 13, but the aircraft model trimmed at 6000 m altitude and 0.6 Mach speed.

FIGURE 15: Same as Figure 13, but the aircraft model trimmed at 3000 m altitude and 0.8 Mach speed.

FIGURE 16: Same as Figure 13, but the aircraft model trimmed at 5000 m altitude and 1.0 Mach speed.

FIGURE 17: Same as Figure 13, but the aircraft model trimmed at 3000 m altitude and 1.2 Mach speed.

FIGURE 18: Same as Figure 13, but the aircraft model trimmed at 6000 m altitude and 1.2 Mach speed.

FIGURE 19: The convergence performance of CGA, flight envelope points corresponding to number of iterations with load factor achieve 97.5% of maximum value of it.

TABLE 6: The results using a modified Gauss function and the population size of 64 individuals.

Mach	Alt.	Run 1		Run 2		Run 3		Run 4		Max value
		Iter. no.	AOA	Iter. no.	AOA	Iter. no.	AOA	Iter. no.	AOA	
0.3	1000	208	20.39536	214	19.86801	171	19.35071	209	20.75794	20.75794
0.6	6000	140	19.39938	187	19.80721	148	18.83285	147	19.34686	19.80721
0.8	3000	221	20.04925	282	20.60796	308	21.66616	172	18.59303	21.66616
1	5000	179	20.00234	201	19.7998	150	19.12577	182	20.94341	20.94341
1.2	3000	276	17.59301	185	19.37977	162	20.19032	271	20.91565	20.91565
1.2	6000	212	21.90015	229	15.17864	208	20.39536	208	20.39536	21.90015

TABLE 7: Same as Table 6, but using a modified Sinc function.

Mach	Alt.	Run 1		Run 2		Run 3		Run 4		Max value
		Iter. no.	AOA	Iter. no.	AOA	Iter. no.	AOA	Iter. no.	AOA	
0.3	1000	182	19.30103	170	19.04308	251	20.60302	166	19.00672	20.60302
0.6	6000	210	20.24026	207	20.67776	262	20.89506	208	19.15628	20.89506
0.8	3000	175	13.64315	239	21.17564	210	20.24026	207	20.67776	21.17564
1	5000	210	20.24026	207	20.67776	262	20.89506	209	18.92149	20.89506
1.2	3000	200	21.0805	364	14.64127	192	19.00408	255	12.94459	21.0805
1.2	6000	450	20.90201	257	12.77573	178	19.91921	181	15.52381	20.90201

In other words, it guarantees that the best fitness in the population is a monotonically nondecreasing function. The above procedure will be closely followed in this work to solve the clearance problem using the ADMIRE aircraft model which is described next.

3. ADMIRE Aircraft Model

The aircraft model used in this work is the ADMIRE (aerodata Model in a Research Environment). It is a nonlinear, six-degree-of-freedom simulation model developed by the Swedish Aeronautical Research Institute using aerodata obtained from a generic single-seat, single-engine fighter aircraft with a delta canard configuration. This model is augmented with a flight control system and includes engine dynamics and actuator models and number of uncertain parameters [13]. In state space, this model is represented

by a set of twelve first-order coupled nonlinear differential equations [13]:

$$\dot{x}(t) = f(x(t), u(t), \Delta), \quad (6)$$

$$y(t) = h(x(t), u(t)), \quad (7)$$

where $x(t)$ is the state vector with twelve components (velocity, angle of attack, sideslip angle, angular rate vector, attitude angles, position, etc.), Δ represent uncertain parameters, $y(t)$ is the output vector, $u(t)$ is the control input vector, whose components are left and right canard deflection angles, left and right inboard/outboard elevator deflection angles, leading edge flap deflection angle, rudder deflection angle, and vertical and horizontal thrust vectoring (see Figure 5 for illustration). The control input is given by

$$u(t) = g(x(t), y_{\text{ref}}(t)), \quad (8)$$

FIGURE 20: Maximum angle of attack versus the iteration (generation). Result using CGA with population size of 64 and both initialization functions: modified Gauss function (G) and modified Sinc function (S). Aircraft model trimmed at 1000 m altitude and 0.3 Mach speed.

FIGURE 21: Same as Figure 20, but the aircraft model trimmed at 3000 m altitude and 0.8 Mach speed.

where $g(\cdot, \cdot)$ is a standard flight control law, which is set by the ADMIRE model, and $y_{\text{ref}}(t)$ is the reference demand that consists of the pilot inputs such as pitch and roll stick demands. Equations (6), (7), and (8) represent the closed loop dynamics of the aircraft with the flight control law in the loop.

The ADMIRE is augmented with a flight control system, FCS, in order to provide stability and sufficient handling qualities within the operational envelope. The FCS contains a longitudinal and a lateral part. The function of the longitudinal controller is pitch rate control q_{com} below Mach number 0.58 and load factor control $n_{z\text{com}}$ above Mach number 0.62. A blending function is used in the region in between in order

to switch between the two different modes. The longitudinal controller also contains a speed control $V_{T\text{com}}$. The lateral controller enables the pilot to perform roll control around the velocity vector of the aircraft $P_{\omega\text{com}}$ and to control the sideslip angle β_{com} [13].

The augmented ADMIRE operational flight envelope is defined up to 1.2 Mach and 6000 meter altitude [13]. The longitudinal control law is the gain scheduled over the whole flight envelope with respect to Mach and altitude variations and is designed to ensure robust stability and handling qualities over the entire flight envelope. The model also contains rate limiting and saturation blocks as well as

FIGURE 22: Same as Figure 20, but the aircraft model trimmed at 3000 m altitude and 1.2 Mach speed.

FIGURE 23: Same as Figure 20, but the aircraft model trimmed at 5000 m altitude and 1.0 Mach speed.

nonlinear stick shaping elements in its forward path; more details are found in [13].

4. Results

The CGA as described in Section 2 was used in this work to generate the pilot input signals for both pitch and roll pilot commands. These two command inputs were used during the simulation of ADMIRE model to find the aircraft response. The load factor was extracted from the aircraft response and is used as fitness function in the GCA. The CGA algorithm coupled with ADMIRE aircraft model described in Section 3 was implemented and programmed using the MATLAB platform.

The initial setting of the CGA-related parameters are as follows: the population size was set to 64 and 128 individuals, the crossover probability was set to 0.5, and the mutation probability was set also to 0.5. During the iteration, the initialization of new population was taking place and shared the offspring population by 12.5% of population size. The passed offspring population to the next generation from crossover process was 25% of population size. The rank-based selection strategy was used where the rank-based ratio was set to 10%.

Immigration operator was applied when the improvement in the fitness value of the best individual of the population over 20 generation is less than $1e-6$. The CGA was stopped when one of the following conditions is met. First,

FIGURE 24: Same as Figure 20, but the aircraft model trimmed at 6000 m altitude and 0.6 Mach speed.

FIGURE 25: Same as Figure 20, but the aircraft model trimmed at 6000 m altitude and 1.2 Mach speed.

the number of immigration processes taking place is more than 100 times and second, when the generation number exceeds 400. Due to the stochastic nature of CGA, a total of four runs were performed for every point in the flight envelope.

Table 1 shows the selected points in the flight envelope used in the present investigation. Four of these points are located on the flight envelope boundary and two are inside it. The selection of these points is similar to that used by other investigators [3–5]. The aircraft model was trimmed to the selected altitude and speed value after that the pilot command inputs generated from the CGA algorithm was applied to aircraft model for 10 seconds.

The simulation was performed using two initialization functions which are the modified Gauss function and the modified Sinc function. For both of these functions the population size is 64 and 128 individuals. Tables 2, 3, 4, and 5 show the results obtained from this simulation. The number of iterations (generation) required to obtain the maximum load factor value are presented in Tables 2–5 for each selected point in the flight envelope, Figure 6. Figures 7, 8, 9, 10, 11, and 12 show the convergence performance of CGA for the selected points in the flight envelope. Each of these figures shows the convergence performance for the cases that results in using the two initialization modified Gauss and modified Sinc functions and two deferent values for the population

FIGURE 26: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum angle of attack value. The aircraft model trimmed at 1000 m altitude and 0.3 Mach speed.

FIGURE 27: Same as Figure 26, but the aircraft model trimmed at 6000 m altitude and 0.6 Mach speed.

FIGURE 28: Same as Figure 26, but the aircraft model trimmed at 3000 m altitude and 0.8 Mach speed.

FIGURE 29: Same as Figure 26, but the aircraft model trimmed at 5000 m altitude and 1.0 Mach speed.

FIGURE 30: Same as Figure 26, but the aircraft model trimmed at 3000 m altitude and 1.2 Mach speed.

FIGURE 31: Same as Figure 26, but the aircraft model trimmed at 6000 m altitude and 1.2 Mach speed.

sizes: 64 and 128 individuals. Pitch and roll pilot commands obtained from CGA which represent the pilot command that generate the maximum load factor ($n_{z \max}$) value are shown in Figures 13, 14, 15, 16, 17, and 18. These results show that the maximum value for the load factor is about 10.9 g obtained at 3000 m attitude and 0.8 Mach speed. This means that the load factor criterion exceeds the limit ($-3\text{ g}-9\text{ g}$) specified by the clearance process [3]. This result agrees with that obtained by other investigators using other optimization methods [1–5].

The previously mentioned results show that the CGA has a significant convergence speed to the maximum value of the clearance criteria. For example, Figure 19 shows the number of iterations (generations) in the above simulation results in Figures 7–12 for the clearance criteria to reach 97.5% of the maximum value. As can be seen from this figure, the majority of the simulation iterations reached 97.5% of the maximum value in about 20 iterations or less.

Tables 6 and 7 and Figures 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, and 31 show the simulation results when considering the angle of attack as fitness function in CGA. These results show that the maximum value for the angle of attack is 21.9 obtained at 6000 m altitude and 1.2 Mach speed. This means that the angle of attack criteria is within the limit ($-10^\circ-26^\circ$) specified by the clearance process [3]. This result also agrees with that obtained by other investigators using other optimization methods [1–5].

5. Discussion and Conclusions

In this work, the CGA was used as a global optimization problem solver to study the clearance process of flight control laws. The CGA was used to find the worst-case pilot inputs found based on the load factor and angle of attack exceedance criteria, for a given speed and altitude, that might stall the aircraft or exceed its structural stress limit. The results presented in this work agreed well with those obtained by other investigators using other optimization methods. In addition, the results in this investigation show that the CGA has a significant convergence speed to the maximum value of the considered clearance criteria for angle of attack and load factor.

Several researches used conventional genetic algorithms and other optimization methods to find the worst-case combination but with some restriction on the pilot command signal such as parameterizing the pilot signals and then solving the resulting optimization problem. But here in CGA, there is no any restriction on the shape of the pilot command signal.

The parallel implementation of the CGA is currently under consideration by the authors to reduce the computational burden of genetic algorithms.

References

- [1] M. Selier, C. Fielding, U. Korte, and R. Luckner, "New analysis techniques for clearance of flight control laws," in *AIAA Guidance, Navigation and Control Conference*, pp. 11–14, Austin, Tex, USA, August 2003.
- [2] P. P. Menon, J. Kim, D. G. Bates, and I. Postlethwaite, "Improved clearance of flight control laws using hybrid optimisation," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 676–681, Singapore, December 2004.
- [3] C. Fielding, A. Varga, S. Bannani, and M. Selier, Eds., *Advanced Techniques for Clearance of Flight Control Laws*, Springer, 2002.
- [4] D. Skoogh, P. Eliasson, F. Berefelt, R. Amiree, D. Tourde, and L. Forssell, "Clearance of flight control laws for time varying pilot input signals," in *Proceedings of the 6th IFAC Symposium on Robust Control Design*, pp. 16–18, Haifa, Israel, June 2009.
- [5] P. P. Menon, J. Kim, D. G. Bates, and I. Postlethwaite, "Computation of worst-case pilots for clearance of flight control laws," in *Proceedings of the 16th Triennial World Congress*, Prague, Czech Republic, 2005.
- [6] Z. S. Abo-Hammour, N. M. Mirza, S. M. Mirza, and M. Arif, "Cartesian path generation of robot manipulators using continuous genetic algorithms," *Robotics and Autonomous Systems*, vol. 41, no. 4, pp. 179–223, 2002.
- [7] Z. S. Abo-Hammour, M. Yusuf, N. M. Mirza, S. M. Mirza, M. Arif, and J. Khurshid, "Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 8, pp. 1219–1242, 2004.
- [8] Z. S. Abo-Hammour, A. G. Asasfeh, A. M. Al-Smadi, and O. M. K. Alsmadi, "A novel continuous genetic algorithm for the solution of optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 414–432, 2011.
- [9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [10] D. Sarkar and J. M. Modak, "Genetic algorithms with filters for optimal control problems in fed-batch bioreactors," *Bioprocess and Biosystems Engineering*, vol. 26, no. 5, pp. 295–306, 2004.
- [11] Z. Michalewicz, C. Z. Janikow, and J. B. Krawczyk, "A modified genetic algorithm for optimal control problems," *Computers and Mathematics with Applications*, vol. 23, no. 12, pp. 83–94, 1992.
- [12] Z. Michalewicz, J. B. Krawczyk, M. Kazemi, and C. Z. Janikow, "Genetic algorithms and optimal control problems," in *Proceedings of the 29th IEEE Conference on Decision and Control*, pp. 1664–1666, December 1990.
- [13] L. S. Forssell, G. Hovmark, A. Hyden, and F. Johansson, "The aero-data model in a research environment (ADMIRE) for flight control robustness evaluation," GARTUER/TP-119-7, 2001.