

A DARWINIAN APPROACH TO CONTROL - STRUCTURE DESIGN*

David C. Zimmerman
University of Florida
231 Aerospace Building
Gainesville, Florida 32611

GENETIC ALGORITHMS

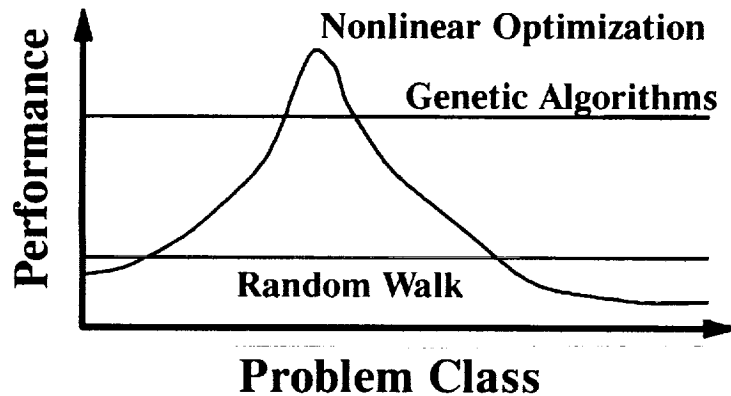
- Based on Darwin's "Survival of the Fittest" Theories
- Shows Great Potential for
 - Multi-Modal Objective Functions
 - Discrete and/or Continuous Design Variables
 - Discontinuous Design Space
- Works With a Coding of the Design Variables,
Not the Design Variables Themselves
- Searches From a Population of Designs,
Not a Single Design Point
- Uses Payoff (Objective Function) Information,
Not Gradient Information
- Uses Probabilistic Transition Rules,
Not Deterministic Rules

Genetic algorithms (GA's), as introduced by Holland (1975), are one form of directed random search. The form of direction is based on Darwin's "survival of the fittest" theories. GA's are radically different from the more traditional design optimization techniques. GA's work with a coding of the design variables, as opposed to working with the design variables directly. The search is conducted from a population of designs (i.e., from a large number of points in the design space), unlike the traditional algorithms which search from a single design point. The GA requires only objective function information, as opposed to gradient or other auxiliary information. Finally, the GA is based on probabilistic transition rules, as opposed to deterministic rules. These features allow the GA to attack problems with local-global minima, discontinuous design spaces and mixed variable problems, all in a single, consistent framework.

* Research Sponsored in part by the NASA CSI Program Office,
NASA Grant Number NAG-1-1017.

GA's – ALGORITHM OVERVIEW

- Initial Population of Designs Created – Random or Heuristic
- Initial Population Allowed to “*Evolve*” Over Generations
- Conjecture – Evolution is the Best Compromise Between Determinism and Chance
- Motivation – GA's are Robust Over a Wide Range of Problems



In GA's, a finite number of candidate solutions or designs are randomly or heuristically generated to create an initial population of designs. This initial population is then allowed to evolve over generations to produce new, and hopefully better, designs. The basic conjecture behind GA's is that evolution is the best compromise between determinism and chance. The basic motivation behind the development of GA's is that they are robust problem solvers for a wide class of problems. However, it should be noted that they are not as efficient as nonlinear optimization techniques over the class of problems which are ideally suited for nonlinear optimization; namely continuous design variables with a continuous differentiable unimodal design space.

GENETIC ALGORITHM MODULES

- Design Variables *Coded* as a q-Bit Binary Number
 - Continuous Variables Like A/D Converter
 - Discrete Variables Have Unique Binary Strings
 - A Population Member is Just a String of Design Variables

- GA *Evaluation*- Level of *Fitness* Assigned to Each Member
 - Fitness Chosen to be Related to Objective Function
 - GA's Maximize Fitness

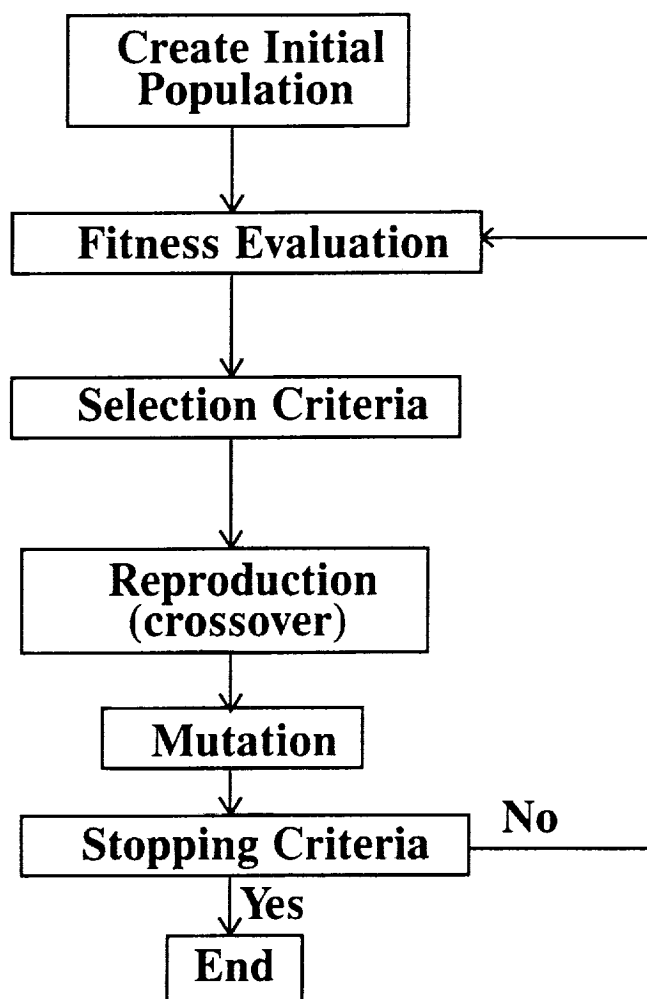
- GA *Selection* - Determination of Which Individuals in Current Population Chosen to be Parents
 - Biased Towards More Fit Members
 - Proportional Bias $p(\text{member}_i) = \frac{\text{fitness}_i}{\sum_{j=1}^{n_{pop}} \text{fitness}_j}$

- GA *Crossover* - Transfer of Design Information From Parents to Prodigy

- GA *Mutation* - Low Probability Random Switch of Bits
 - Retain Design Information Over Entire Design Space
 - Aides Search For Global Optimal Solution

Each design variable is *coded* as a q-bit binary number. A continuous design variable is approximated by 2^q discrete numbers between lower and upper bounds set for the design variable. Discrete variables would each be assigned an unique binary string. A population member is obtained by concatenating all design variables to obtain a single string of ones and zeros. *Evaluation* is the process of assigning a fitness measure to each member of the current population. Because GA's attempt to maximize the fitness of each member, an objective function which is to be minimized must be converted into an equivalent maximization problem. *Selection* is biased towards the most fit members of the population. Therefore, designs which are better as viewed from the fitness function, and therefore the objective function, are more likely to be chosen as parents. *Crossover* is the process in which design information is transferred to the prodigy from the parents. Many crossover operators (1-point, 2-point, uniform) have been investigated. *Mutation* is a low probability random operation which may perturb the design represented by the prodigy. The operator works on a bit-by-bit basis and is governed by the probability of mutation, p_m . At each bit, a biased coin toss is used to determine whether the bit should be logically "NOTed". The mutation operator is used to retain design information over the entire domain of the design space during the evolutionary process.

GENETIC ALGORITHMS – SUMMARY



In the implementation of the GA shown above, the prodigies are produced until the number of prodigies created is equal to n_{pop} , the population size. At that point, the current population of parents are discarded and the prodigies are in turn made parents which are capable of producing the next generation of prodigies. Thus, the production of n_{pop} prodigies can be viewed as the completion of one generation cycle in the evolutionary process. During this procedure, it is possible that both the fitness of the most fit member and the average population fitness can be temporarily reduced during the evolutionary process. To overcome this, the concept of a steady-state GA was implemented. In a steady-state GA (SSGA), the fitness of the children after they have been mutated is evaluated. These fitness values are then compared to the fitness of the two least fit parents in the current population. If the mutated child's fitness is higher than the least fit member in the population, the child will replace that member and will instantly become a candidate parent. To keep intact the concept of a generation, a generation is defined to be complete when the number of children produced, but not necessarily accepted into the population, is equal to n_{pop} .

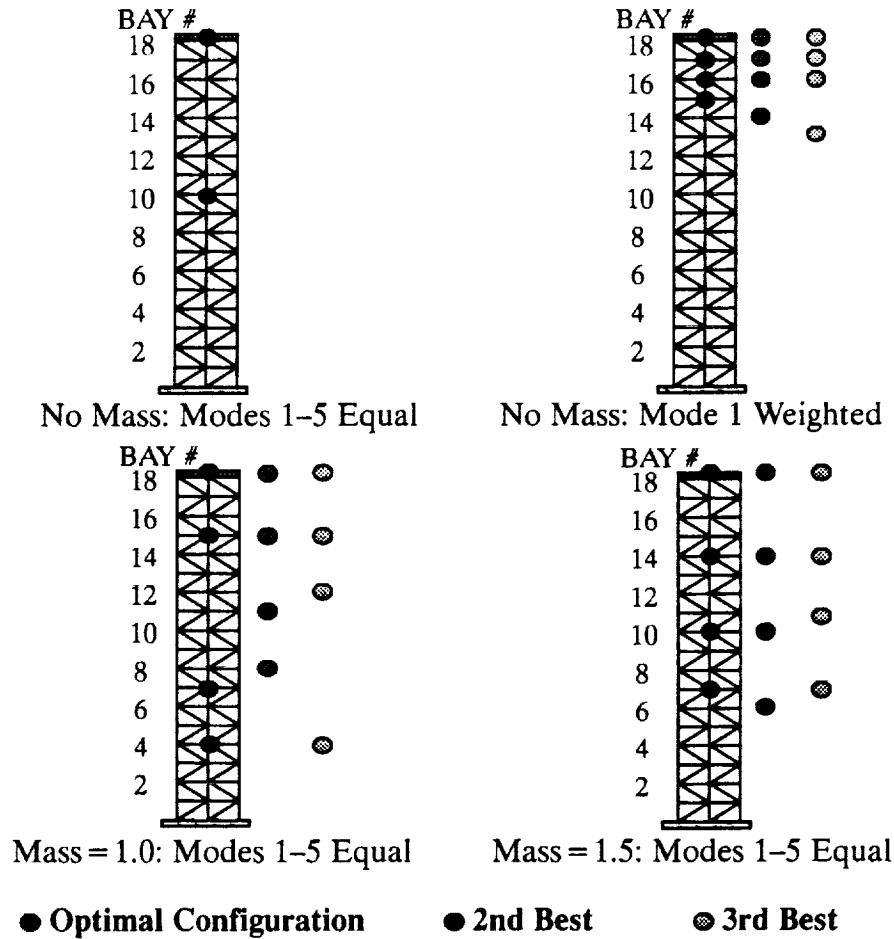
GA EXAMPLE - ACTUATOR PLACEMENT FOR MINI-MAST (discrete design problem)

- **PROBLEM** – Given “N” Candidate Actuator Locations and a Maximum of “K” Actuators, Each of Mass M, Determine the Optimal Configuration
- **Actuator Placement**
 - Criterion Representing Desirability of Configuration
 - Simple Method of Evaluation
 - Algorithm for Cycling Through Configurations – GA’s
- **Criterion: Energy Optimal Degree of Controllability (Longman)** – maximize the “size” of the state space that can be returned to origin in prescribed time and energy
- **FITNESS = EDOC** – (soft penalty function)
 - Soft Penalty Function Penalizes Configurations Which Have More Than Allowable Number of Actuators
- **In Addition to Identifying Optimal Configuration, “Nearly” Optimal Configurations Also Found**

Fundamental to the problem of actuator placement are: (i) the definition of an appropriate criterion representing the desirability of actuator configurations, (ii) the development of a computationally efficient method for the evaluation of this criterion, and (iii) the development of algorithms to cycle through possible candidate actuator configurations. To date, the greatest amount of work has focussed on problems (i) and (ii). The approach taken for problem (iii) by most researchers has been an exhaustive search. That is, given n candidate locations to place m actuators, $m < n$, evaluate the effectiveness criteria for all configurations. The numbers aspect (i.e. place m or less actuators) has rarely been investigated. In this demonstration of the GA, the energy degree of controllability (EDOC) developed by Longman (1989) is used as an actuator configuration effectiveness measure. The effects of actuator mass are incorporated into the EDOC. The energy degree of controllability (EDOC) is related to the size of the region in the state space that can be returned to the origin in a prescribed amount of time T using less than a prescribed amount of energy e . The larger a given actuator configuration’s EDOC is, the greater its control authority. The optimal actuator configuration is that which maximizes the EDOC. Therefore, the objective function used for evaluation is taken as $J = EDOC - W(n_{act} - n_{actmax})\mu(n_{act} - n_{actmax})$ where W is an arbitrary weight function, n_{act} is the number of actuators, n_{actmax} is the maximum allowable number of actuators, and μ is the unit step function. The second term is essentially a soft penalty function which reduces the objective function for a given actuator configuration only if the configuration has more actuators than the maximum allowable. Actuator configurations which have less than the maximum allowable are not penalized by this term. Therefore, the optimal number of actuators is also determined. It is possible in this problem that the optimal number of actuators is less than the maximum allowable because of actuator mass effects. Details of this work are presented in Zimmerman (1991).*

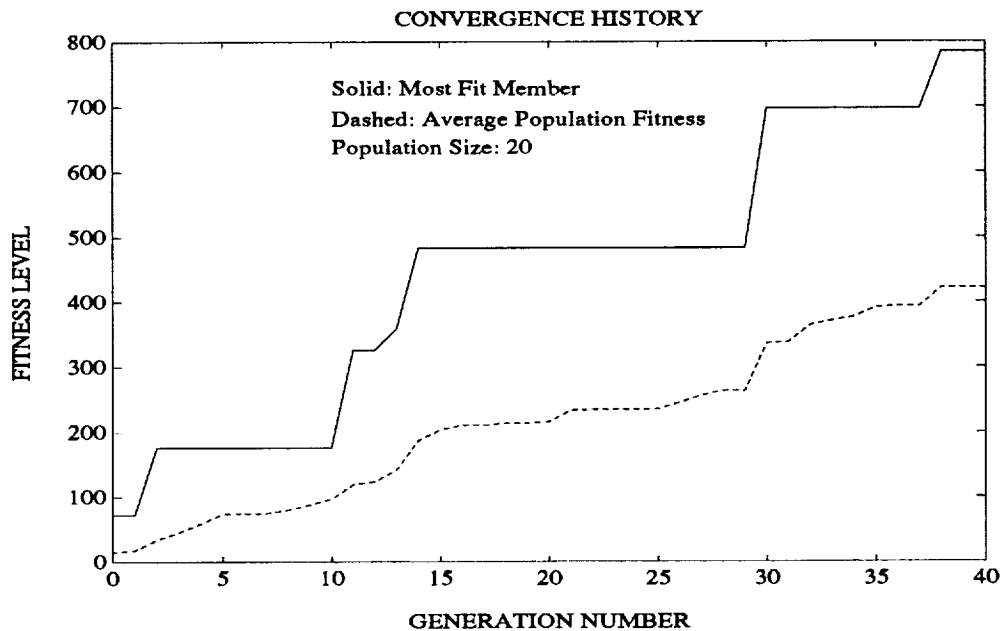
*Zimmerman, D.C. (1991). “A Darwinian Approach to the Actuator Number and Placement Problem with Nonnegligible Actuator Mass.” to appear, *Journal of Mechanical Systems and Signals*.

ACTUATOR PLACEMENT – RESULTS



Four test cases were run. In each figure, the optimal actuator configuration is shown pictorially with the solid ovals. In addition to identifying the optimal configuration, the final generation of designs also provides valuable information concerning other “nearly optimal” solutions. This is of significance in that (i) insight into the optimization process can be gained and (ii) it allows for human judgement to factor in other criteria not embodied in the objective function in comparing the “nearly optimal” designs to the “true optimal” design as dictated by the fitness function. These “nearly optimal” designs are indicated to the right of each figure. In the top-left case, the optimal configuration for placing two massless actuators was determined with equal mode 1-5 weighting (171 possible combinations). It should be noted that the results correspond to the actual Mini-Mast configuration. To increase the possible number of combinations, the remaining problems looked at placing four or less actuators (4047 possible combinations). The top-right case was for no actuator mass and control of only mode one deemed important. The GA results are consistent with physical intuition. The bottom two cases demonstrate the effects of actuator mass on the placement problem. For actuator mass normalized to unity (mass = 1), the optimal configurations are shown in the bottom-left figure. For an increase in actuator mass of 50%, the optimal actuator configurations are shown in the bottom-right figure. Comparing these two figures demonstrates the obvious importance of including actuator mass in any placement algorithm. All GA results presented above were validated by exhaustive search. This was possible due to the size of the factorial problem investigated. The results showed that the final GA population included a minimum of five of the top seven actuator configurations (including the optimal) for each case.

CONVERGENCE HISTORY

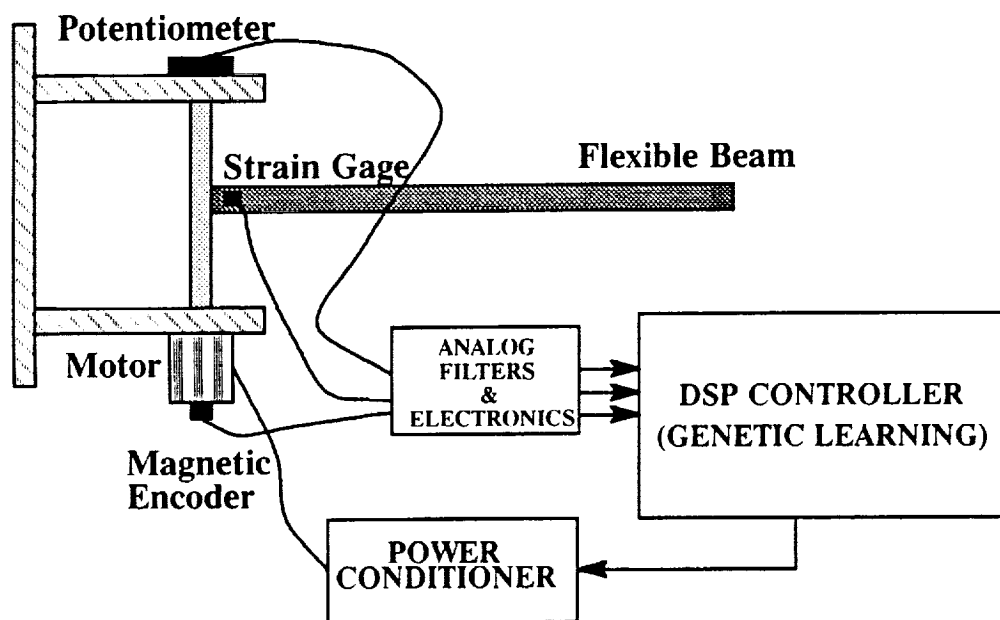


- **Average Population Increase With Each Generation - Characteristic of Steady-State GA**
- **Population Size = 20, Generations = 40, Therefore 800 Function Evaluations (4047 possible combinations)**

The above figure shows the convergence history of the GA for the case of placing 4 actuators with no actuator mass. The GA identified the optimal solution after 38 generations, although the algorithm was run for a total of 40 generations. With a population size of 20 members, the GA required 800 function evaluations to arrive at the optimal solution (exhaustive search would require 4047). At a given generation number, the maximum fitness value represents the most fit member in the population, whereas the average fitness is the mean fitness of the entire population. It can be seen that the average fitness increases with each new generation, which is a property of the SSGA used. In a study of a large combinatorial problem not shown (906,192 possible solutions, optimal solution known), the GA was able to determine the optimal solution in less than 2500 function evaluations. Although no optimization algorithm can guarantee convergence to the global optimal solution, experience with the GA has shown that GA's are a powerful tool to improve CSI designs.

GENETIC ALGORITHM LEARNING CONTROL

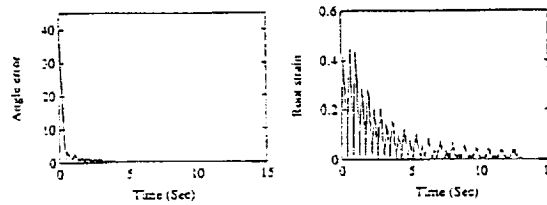
- Utilize Genetic Principles to Evolve Controller Making Use of On-Line Experimental Measures of Fitness
- Focus Application - Single Link Large Angle Slewing
- Weighted Fitness Function - Strain and Angle Error



In this application of GA's, a Genetic Algorithm Learning Control (GALC) formulation is investigated (Layton and Zimmerman, 1992). In learning control, the control law is adapted from information gained by repeating the desired operation. In the GALC, various controller forms (i.e., parameterized control laws) are formulated. The evolutionary principles of Genetics are then utilized to not only select the optimal control law parameters, but also to select the optimal control law form. For this particular test case, the desired maneuver is a rest-to-rest 45 degree slew. Available sensor information included angle, angle rate, and beam root strain. In simulation studies, the optimal control law form was determined (as well as the optimal control parameters). Experimentally, the control law form was fixed with the GALC varying the control parameters. Fitness information was obtained experimentally by integrating the angle error (square difference of the desired and actual angle) and the square of the root strain signal. The objective of the GALC was to minimize a weighted integral of angle error and root strain. All processing was done digitally using a DSP controller with an approximate update rate of 33kHz.

GENETIC LEARNING

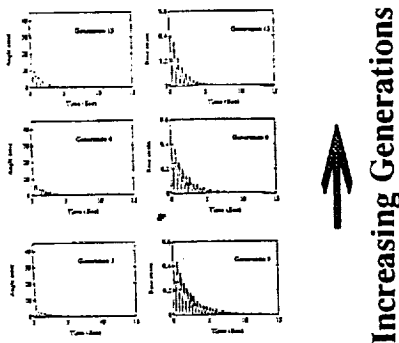
● Heavy Weighting on Angle Error



● Heavy Weighting on Strain Error



● Relative Weighting Learning

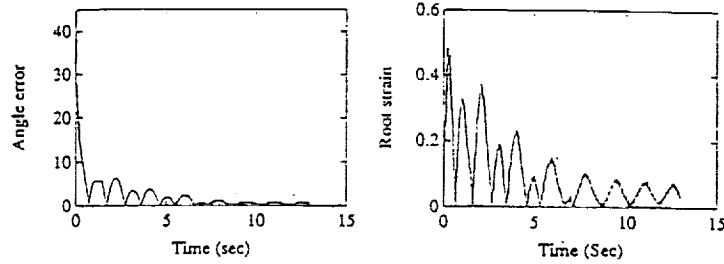


Three experimental case studies were investigated. In all cases, the initial population was selected randomly and was not biased with any knowledge of the beam, actuator, or sensor dynamics. In other words, there was no need to develop a system model as far as the experiment was concerned. Fitness functions were developed using experimental sensor signals. The repeatability of these calculations was nominally 8% error. Thus, issues of noisy function evaluation were addressed. In the first experiment, the fitness function was weighted such that there was no penalty on the strain signal. The results are in agreement with physical reasoning: the motor slews as quickly as possible to reduce the angle error regardless of the strain signal. The second experiment is just the opposite of the first; no penalty associated with the angle error. Again, the result of learning control agrees with physical reasoning: slew the beam slowly to minimize the strain signal. Finally, the third experiment provided for approximate equal weighting (in a voltage sense) of the angle and strain signal. The waterfall plot shows the progressive learning of the controller. It should be noted that the cost function also included a time cubed weighting factor within each integral. This effect clearly is demonstrated in the above figures.

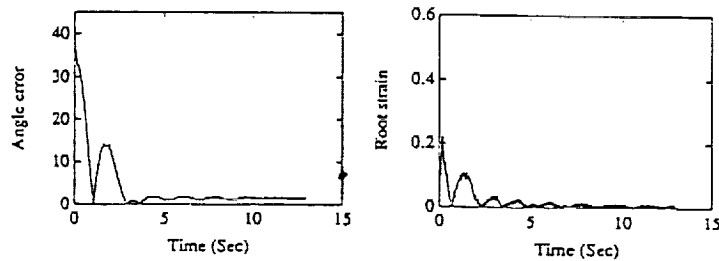
GENETIC LEARNING FOR CHANGED SYSTEM

- **Mass Added to End of Flexible Beam (76g)**
(Mass of Flexible Beam - 112g)

- **Previous Optimal Control on Perturbed System**



- **Maxfit Member After Five Generations**



In this experiment, a 76g mass was added at the tip of the beam. The mass of just the flexible beam was 112g. Obviously, the tip mass greatly influenced the system dynamics. The top graph shows the angle error and root strain time histories when the control optimized for the previous system (i.e. no tip mass) is used to maneuver the beam with tip mass. In comparing this figure with the bottom figure of the previous slide, it is obvious that performance has been seriously degraded. The second figure of this slide shows the angle error and root strain time histories after five generations of learning. It is obvious that the GALC has adapted the control law to better match the "new" system dynamics. It would be expected in this case that the strain signal would have a larger RMS level than the angle error signal, and thus "equal" weighting between angle error and strain is no longer achieved with the same weighting values. In the time history shown above, the weighting values were kept the same as in the previous case. This causes the angle error to remain at a non-zero value as time increases. If the weight on angle error is increased, the angle error would go to zero in the steady-state.

SUMMARY

- **Genetic Algorithms Represent a New Class of Optimization Tools Which Are Applicable To Many CSI Design Problems**
- **Genetic Can Handle Discontinuous Design Spaces and Both Discrete and Continuous Design Variables**
- **Demonstrate Quick Convergence to Near Optimal Solution, Then Slows Down (hybrid solution techniques possible)**
- **Because GA's Require Function Evaluations, Instead of Gradient Information, Well-Suited For Noisy Experimental Function Evaluations**
- **Demonstrated For Both Actuator Placement and Learning Control, But Other Applications Tested Include**
 - **Truss Configuration and Sizing**
 - **Constrained Layer Damping Treatment Placement**
 - **Actuator Placement with Simultaneous Control Design**

In this work, the use of Genetic Algorithms (GA's) in solving various CSI design problems was presented. The basic principles of GA's were addressed as well as the motivation of applying GA's to CSI design problems. Two case studies were presented. The first problem involved actuator number and placement, a discrete design problem. The focus structure was the NASA Mini-Mast. The results indicate the promise of GA's in solving large order combinatorial problems. The second problem addressed the development of a Genetic Algorithm Learning Control technique. Experimental results for the slewing of a flexible beam demonstrated the learning ability of the controller. Most importantly, the control law was able to adapt even in the worst case of no prior knowledge of system dynamics. The adaption was driven making use of experimental measures of performance. Of course, prior knowledge of system dynamics can be used to bias the initial GA population to enhance GA learning. In this case, GA learning would compensate for analytical modelling errors (including unmodelled effects).

REFERENCES

Holland, J.H., (1975), *Adaption in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan.

Longman, R.W., and Horta, L.G., (1989), "Actuator Placement by Degree of Controllability Including the Effect of Actuator Mass," Proceedings of the 7th VPI&SU/AIAA Symposium on the Dynamics and Control of Large Structures, Blacksburg, Virginia.

Layton, D.S. and Zimmerman, D.C. (1992), "Genetic Algorithm Learning Control of Large Angle Slewing Maneuvers," AeMES TR-92-03-01, Department of Aerospace Engineering, Mechanics & Engineering Science, University of Florida.

SESSION III

Chairman: Robert A. Laskin
Jet Propulsion Laboratory
Pasadena, California

Co-Chairman: Richard Martin
Phillips Laboratory
Edwards AFB, California

