DSC-Vol 44, Transportation Systems – 1992
ASME 1992

GENETIC PROGRAMMING:
A NEW PARADIGM FOR CONTROL AND ANALYSIS

Richard J. Hampo
Ford Motor Company Research Laboratory
Vehicle Electronics Department
Dearborn, Michigan

Abstract

Genetic Programming (GP) is a method of discovering computer programs which solve problems. It combines ideas about genetics and evolution based on fitness with computer programming to come up with a very powerful yet still general problem solving technique.

Genetic Programming has several useful features. The most important of these is the ability to 'discover' explicit solutions to problems. Unlike neural networks which learn to do a job but can be difficult to extract information from, GP not only learns to do the job, but can offer a simple representation of that answer, from which insight into the problem may be gained.

Background

Genetic programming was derived by John Koza from the field of genetic algorithms. Koza's work is described in [3], [4], and [5]. John Holland describes genetic algorithms in his 1975 book "Adaptation in Natural and Artificial Systems", in which the main new ideas are the use of fitness proportionate mating and reproduction to generate new members in a population of individuals. Each of these individuals is a potential solution to the problem at hand. This problem is encoded into strings of binary digits. Each member of an initial population of strings is tested for fitness on the given problem after it is decoded. The strings are then crossed with another such member of the population to produce a new member. This new member contains string parts from each parent. The more fit individuals are chosen to mate more frequently, so this type of reproduction is fitness proportionate. As a result, the population as a whole improves its fitness as the generations progress. Genetic programming overcomes some of the limitations inherent in the genetic algorithms. Essentially, in genetic algorithms, the representation scheme used to encode the problems into bit strings artificially limits the size and shape of the possible solutions. With GP, the representation used is much more natural to the problem at hand. In addition, the size and shape of the possible solutions is not as limited as it is with genetic algorithms.

Basics of Genetic Programming

Genetic programming operates on a data set of candidate solutions to the problem at

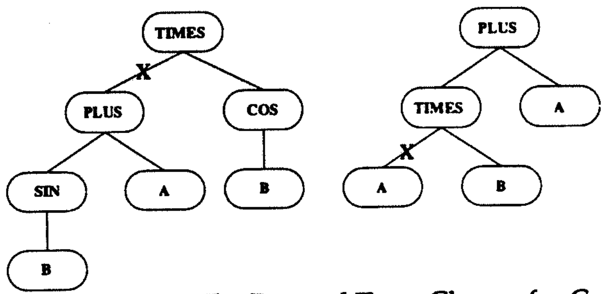

Figure 2: The Parental Trees Chosen for Crossover

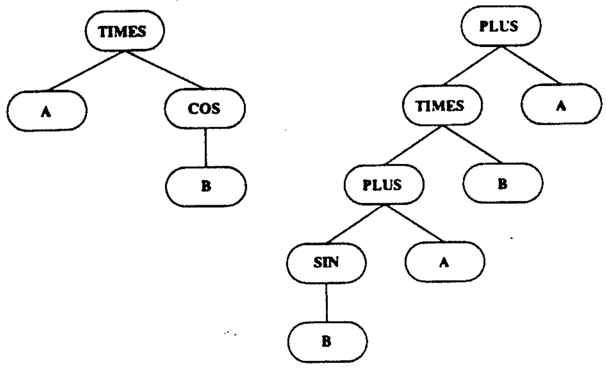

Figure 3: The Resultant Trees After Crossover

3. Two parents are needed for crossover, so another is chosen in the same manner as the first. Crossover points are then chosen (randomly) which can be any point in the trees. Two bold X's mark the points chosen for crossover. The portions of the trees below the crossover point are then swapped. These two new individuals are inserted into the new generation. Note that the parents are not removed from the old population, only copied. Thus reproduction takes place with reselection - i.e. one individual can be parent to more than one child.

Replication is simpler than crossover. One parent is chosen from the old generation (proportional to fitness, of course) and is copied into the new generation.

Mutation is performed least often. This operation consists of choosing a parent, choosing a mutation point, in the same manner as one chooses a crossover point, and then inserting a randomly generated tree at the mutation point. Mutation can help to re-introduce variety in a population that has become too homogeneous.

These reproduction operations are performed until the new generation is complete. The old generation is then discarded (as far as the algorithm is concerned). One may, however, choose to retain the best individual in memory for comparison as the process continues.

After reproduction, the algorithm simply loops back to the fitness calculation. These calculations are performed again, this time using the new population. This process is repeated until some termination condition is met. Typically, some maximum is imposed on the number of generations, or some measure of perfection is defined for the fitness function. This termination criteria is application dependent.

Example 1: Two-Dimensional Function Approximation

A real-life problem that involves a two dimensional curve (surface) fitting is defined as follows. An electric motor and electronic drive combination is used as the prime mover in an application. This drive system has to operate over wide speed and load ranges. The efficiency of the drive is important because it determines how much power needs to be delivered to the system, and also cooling requirements. One would like to take the power consumption solutions produced. The best solution from each of the runs had raw fitnesses that were different by at most 15 percent. The best solutions varied in complexity, but most of them were approximately quadratic in the LOAD parameter and ignored the SPD parameter completely. The difference between the overall best solution (one that used both SPD and LOAD) and the best solution using only LOAD was very small. The quadratic dependance on LOAD makes sense since most of the loss was expected to be resistive loss due to the current flowing in the motor, which is a simple quadratic. Figure 5 shows a graph of one of the best solutions found. This solution simplifies to:

$$1.16 \cdot LOAD^2 - .066 \cdot LOAD + .000934.$$

Figure 6 shows the error surface for this solution. The best solution found which only uses the LOAD parameter is

```
(Limiter[(-0.068224)+(LOAD),0.947698])* ((LOAD)*(Exp[0.195231]))
```

This solution's fitness is about 5 percent better than that of the quadratic solution shown in figure 5. The overall best solution found was

```
(((Div[((((Limiter[(-0.095502)*(0.812365),Div[0.791595,SPD]])-(Div [(SPD)*(SPD)
,Cos[Pi*(LOAD)]]])+(((Sin[Pi*(SPD)])+(Div[LOAD,LOAD])) +(SPD)))*(LOAD))+(LOAD),Div[Div[LOAD,(0.298916)*
(Limiter[LOAD,-0.015224])] ,(Limiter[LOAD,-0.015224])-(Cos[Pi*(((-0.079271)*(-0.441625))+(Sin[Pi*(SPD)]))])]]
)*((SPD)-(Limiter[(SPD)-(LOAD),(Div[0.888507,(SPD)-(SPD)])-(
SPD)]))))+(LOAD))*(((0.298916)*(Limiter[LOAD,Cos[Pi*(-0.341589)]]))+((Limiter[-0
.079207,Div[Div[((LOAD)-(((Cos[Pi*(LOAD)])+(SPD))- (((LOAD)+(LOAD))*(Div[0.992653,SPD]))))+((-0.027312)+
(LOAD)),Div[Sin[Pi*(Div[Limiter[LOAD,Limiter[LOAD,LOAD]],(LOAD)*(SPD)])]
,(0.044022)+(Limiter[LOAD,Cos[Pi*(-0.341589)]])]],0.298916]])+(LOAD)))
```

which is very complex, having 12 levels and 112 operators. This solution's fitness is only about 8 percent better than the best quadratic solution shown in figure 5. Thus we conclude that a the quadratic

$$1.16 \cdot LOAD^2 - .066 \cdot LOAD + .000934.$$

is a good, concise approximation to the experimental data.

Example 2: Active Suspension Optimal Control

Active suspensions (AS) are in the research and advanced development stages now at most of the automotive companies, including Ford. AS is expected to be a popular option in the years to come because it offers ride and handling benefits not available from any other type of suspension system. Essentially, an AS system is differentiated from other systems by the presence of an active element, one which is capable of generating force or motion under command between the wheel and body. A picture of the model used for the simulation is shown in figure 7. For more extensive discussion of the active suspension, the reader should see [8-14].

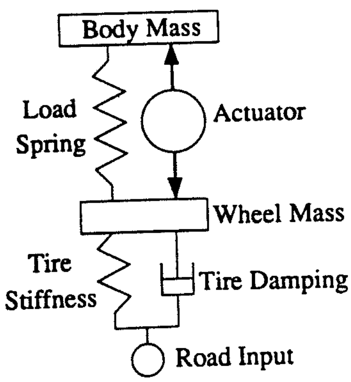

Figure 7: Model of Active Suspension

Results

As with the previously mentioned mapping problem, many runs were made, changing random number seeds, the size of the population, and the set of operators used. The simulation used to determine fitness was a drive over two seconds of a particular road. This road had been measured and its vertical irregularities have been stored in a computer file. The model of the suspension was a four state model, with some nonlinear terms. The best individuals discovered were then tested controlling a more complete model, including sensor and actuator dynamics, run over several different road surfaces. For comparison purposes, the same simulations were performed with a model for the passive suspension. The model parameters for both the active and passive systems (masses , spring rates, etc.) were obtained from an actual vehicle. A table showing some of the results achieved is shown in figure 9. The rating number column simply lists the value of the fitness functions described above (weighted sum of RMS body acceleration, RMS tire deflection, and RMS body velocity) for each test case. The notation *** in a particular case means that the simulation was not completed due to the de-stabilizing effect of the particular controller. In a few cases, the system was stable for a few seconds, then went unstable when a large bump was encountered. This was particularly common for controllers containing cubic and quartic terms in a measured variable, like BV^4 .

Some of the best genetic controllers are listed in figure 8. The base-line controller is described in [1] and [6]. This controller was developed heuristically, and tuned via simulation.. The figures 10 through 12 compare the response of passive, base-line, and genetic controllers when subjected to the Southfield Road input. The passive system allows a great deal of body motion as compared to the active systems, as expected. This is a good reason for using an active system. However, the trade-off is in power consumption and also increased tire deflection. Excessive tire deflection is undesirable because it erodes the roadholding ability of the tires. However, when comparing passive and the best active systems on smoother roads, there is only a very small increase in the RMS tire deflection while the RMS body acceleration can be reduced by more than an order of magnitude[1]. Looking at the table of results indicates that the best controllers found (numbers 8 and 9) are strictly linear. This is an interesting result since nonlinear dynamics were included in the plant and actuator models. However, it appears that the nonlinearities are well enough behaved that a linear controller can work well.

Future Work

A wealth of linear optimal control theory exists, and one investigation not yet undertaken is to develop a linear optimal control law based on a linearization of the plant and the same cost (fitness) function that was used in the genetic programming. A comparison could then

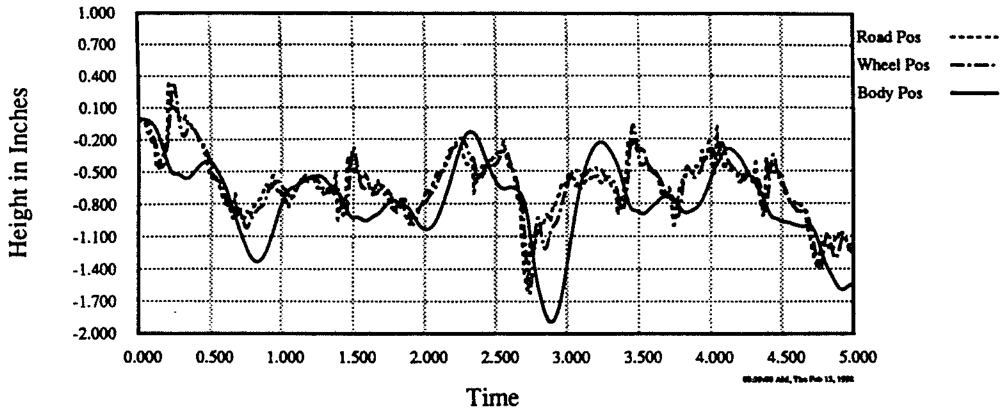

Figure 10: Passive Suspension

Bibliography:

[1] Hampo, R. J. "The Genetic Programming Paradigm: A New Tool for Analysis and Control" Ford Research Technical Report, March 1992

[2] Holland, J. H. "Adaptation in Natural and Artificial Systems," Univ. of Michigan Press, 1975

[3] Koza, J. R."Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems," Stanford Univ. Computer Science Dept. Technical Report # STAN-CS-90-1314, June 1990

[4] Koza, J. R. "Genetic Programming" MIT Press, (to be published 1992)

[5] Koza, J. R. "Evolving a Computer Program to Generate Random Numbers Using the Genetic Programming Paradigm," Proceedings of the Fourth International Conference on genetic Algorithms, San Mateo California, 1991, pages 37-44

[6] Hampo, R. J. "Neural Network Control of an Active Suspension System", Masters Thesis, Univ. of Michigan, May 1990

[7] Hampo, R. J., Marko, K. A., "Neural Network Architectures for Active Suspension Control", Proceedings of the IJCNN-91, July 1991, pp II-765 -- II-770

[8] Yue, C., Butsuen, T., and Hedrick, J. K., "Alternative Control Laws for Active Suspensions," Journal of Dynamic Systems, Measurement, and Control, Vol. 111, June 1989, pp. 288 – 296.

[9] Hrovat, D. Margolis, D. L., and Hubbard, M., "An Approach Toward the Optimal Semi-Active Suspension," Journal of Dynamic Systems, Measurement, and Control, Vol. 110, September 1988, pp. 288 -- 296

[10] Thompson, A. G., "Optimal and Suboptimal Linear Active Suspensions for Road Vehicles," Vehicle System Dynamics, 13(1984), pp.61 -- 72

[11] Karnopp, D. "Two Contrasting Versions of the Optimal Active Suspension," Journal of Dynamic Systems, Measurement, and Control, Vol. 108, September 1986, pp. 264 -- 268

[12] Chalasani, R. M., "Ride Performance Potential of Active Suspension Systems -- Part I: Simplified Analysis Based on a Quarter Car Model," Proceedings of the ASME Symposium on Simulation of Ground Vehicles, Dec, 1986

[13] Chalasani, R. M., and Alexandridis, A. A., "Ride Performance Potential of Active Suspension Systems – Part II: Comprehensive Analysis Based on a Full-Car Model," General Motors research Publication GMR-5313, 1986

[14] Thompson, A. G., "Design of Active Suspensions," Proceedings of the Institute of Mechanical Engineers 1970-71, Vol. 185, pp. 553 -- 563