MAE 345
Robotics and Intelligent Systems

Assignment #5
due: December 1, 2011

1) Suppose that you have 10 strings with the following probabilities of selection in a genetic algorithm: 0.1, 0.2, 0.05, 0.15, 0.11, 0.07, 0.04, 0.12, 0.08, 0.08. Write a computer program that simulates roulette wheel selection for these 10 strings. Spin the wheel 1,000 times, and keep track of the number of selections for each string, comparing these results to the expected number of selections.

2) Create a procedure that receives four binary strings and a crossing site value, performs simple crossover, and returns four offspring strings. Test the program by crossing the following strings of length 8, arranged as random pairs: {10101011}, {01010100}, {11111001}, {11010111}. Use several arbitrary crossover points to validate the algorithm.

3) Figure 1 shows the response of a linear, time-invariant second-order system (*e.g.*, a lightly damped robot arm) to an initial condition on its rate:

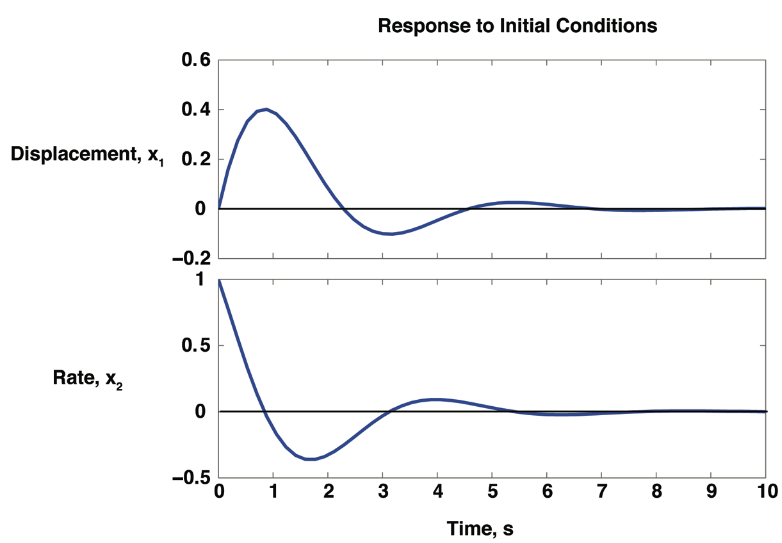

Figure 1. Initial-condition response of a second-order system.

This response is tabulated below and is separately available as an Excel spreadsheet. The system's unforced differential equation is

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (1$$

The object is to find the values of ζ and ω_n that best describe the unknown dynamic system using a genetic algorithm.

a) Define a cost function, J, that penalizes the error between the given history of the state, $\mathbf{x}_{\text{nominal}}(t_k)$ in (0:0.25:10), and the history of a test value of the state, $\mathbf{x}(t_k)$ in (0:0.25:10), that you generate by simulating eq. 1 with the test values of ζ and ω_n. A quadratic cost function is a good choice:

$$J = \frac{1}{2}\sum_{0}^{40}\left[\mathbf{x}_k - \mathbf{x}_{nom_k}\right]^T\left[\mathbf{x}_k - \mathbf{x}_{nom_k}\right] \qquad (2$$

b) We wish to minimize the cost function; however, the genetic algorithm normally maximizes a fitness function. Therefore, transform J to a positive-definite function that is maximized when $J = 0$, *e.g.*,

$$Fitness = e^{-J} \qquad (3$$

c) Write your GA script. Here's some guidance:
i) Define Gene 1 to be ζ and Gene 2 to be ω_n. Given Fig. 1, what are reasonable ranges for the two genes?
ii) Assign 12 bits to each gene; hence, the resolution of each estimate is 1 in 4096, and the chromosome is 24-bits long. Given the ranges of values in (i), what is the value of the least-significant bit for each gene?
iii) Choose an even number, n, of chromosomes (*e.g.*, 32 or 64) for the GA.
iv) Use a random number generator (with uniform distribution) to generate a starting set of chromosomes.
v) Compute J for each chromosome and J_{total} for the entire set. What must you do to compute each J?
vi) Use roulette-wheel selection to generate a new set of n chromosomes.
vii) Choose $n/2$ pairs of chromosomes, and perform crossover at random locations on each chromosome.
viii) Flip a bit at a random location on one randomly chosen chromosome (*i.e.*, mutate the chromosome) once every 5 generations. Does this make a difference in convergence?
ix) Using the new set of chromosomes, repeat the process beginning at (v) until your chromosome set converges to a steady value of J_{total}. How many generations does this require?
x) Choose the most fit chromosome to define your estimates of ζ and ω_n.

d) What are the best-estimate values of ζ and ω_n?

e) Plot J_{total} and J_{best} vs. chromosome generation.

f) Repeat the process assuming that there is a Gaussian random error with 30% standard deviation, σ, in the cost function residual $\left[\mathbf{x}_k - \mathbf{x}_{nom_k}\right]$. Compute the effect by using a random number generator to increase the cost at step (v),

$$J = \frac{1}{2}\sum_{0}^{40}\left\{\left[\mathbf{x}_k - \mathbf{x}_{nom_k}\right]^T\left[\mathbf{x}_k - \mathbf{x}_{nom_k}\right]\left[1+\left(\sigma \times \text{randn}\right)_k\right]^2\right\}$$

, and perform the GA as before. ***randn*** generates a random number with zero mean and a standard deviation of one.

g) What are the best-estimate values of ζ and ω_n?

4) Find the values of ζ and ω_n by minimizing the noise-free cost using MATLAB's ***fminsearch***. Compare these results, including computing time and convergence, with the GA results.

Table 1. Numerical values of response tabulated at 0.25-s interval.

Time, s	x_1	x_2	Time, s	x_1	x_2
0.00	0.000	1.000	5.75	0.023	-0.015
0.25	0.211	0.684	6.00	0.018	-0.022
0.50	0.342	0.367	6.25	0.013	-0.023
0.75	0.398	0.089	6.50	0.007	-0.022
1.00	0.392	-0.128	6.75	0.002	-0.018
1.25	0.340	-0.273	7.00	-0.002	-0.013
1.50	0.261	-0.348	7.25	-0.005	-0.008
1.75	0.171	-0.362	7.50	-0.006	-0.003
2.00	0.084	-0.329	7.75	-0.007	0.001
2.25	0.009	-0.264	8.00	-0.006	0.004
2.50	-0.047	-0.185	8.25	-0.005	0.005
2.75	-0.083	-0.104	8.50	-0.003	0.006
3.00	-0.100	-0.032	8.75	-0.002	0.006
3.25	-0.100	0.026	9.00	-0.001	0.005
3.50	-0.089	0.065	9.25	0.000	0.004
3.75	-0.069	0.087	9.50	0.001	0.002
4.00	-0.047	0.092	9.75	0.002	0.001
4.25	-0.024	0.085	10.00	0.002	0.000
4.50	-0.005	0.070			
4.75	0.010	0.050			
5.00	0.020	0.029			
5.25	0.025	0.010			
5.50	0.026	-0.005			