Proceedings of the 34th
Conference on Decision & Control
New Orleans, LA - December 1995

THE USE OF A GENETIC ALGORITHM IN POWER PLANT CONTROL SYSTEM DESIGN

Robert Dimeo and Kwang Y. Lee

Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802
kyl@ecl.psu.edu

Abstract

The use of a genetic algorithm is studied in designing control systems for a boiler-turbine power plant, with specifics in developping a proportional-integral (PI) controller and a state feedback controller for a non-linear multi-input/multi-output (MIMO) plant model. The plant model is presented along with a discussion of the inherent difficulties in such controller development. A sketch of the genetic algorithm (GA) is presented and its strategy as a method of control system design is discussed.

1. Introduction

The robust control of power plants has recieved a great deal of attention lately [1]. Automation is deemed a necessary condition for safe operation which minimizes material fatigue, the number of staff, and enables efficient plant management [2]. The challenge in controller design for these plants exists because they are typically non-linear and multi-variable with multiple control objectives. While conventional controls such as PID compensators yield an acceptable response, they do not have the flexibility necessary to provide a good performance over a wide region of operation. Application of modern optimal control techniques yields system performances that are optimal at only one operating point.

Recent applications of robust control to power plant models have yielded quite favorable results [1]. These robust controllers perform very well over a wide range of operation. However, these robust control design methodologies require frequency response information in addition to a linearized model of the plant. A goal of modern intelligent control design is to obtain a controller based on input/output information *only*. The genetic algorithm (GA) is a method that holds promise for such a control system design.

The GA is a search technique based on the mechanics of natural genetics and survival-of-the-fittest. Touted as an efficient and general method of searching a complex space [3], the GA has had success in many areas, such as the traveling salesman problem [4], optimal control of an aircraft autopilot system [5], multivariate curve-fitting, and game-playing [4]. A number of these achievements suggest the potential utility of the GA as a method for control system design.

2. The Power Plant Model

The power plant model used is the one developed by Bell and Åström [6] for the boiler-turbine plant P16/G16 at the Sydvenska Kraft AB Plant in Malmö, Sweden. The boiler is oil-fired and the rated power is 160 MW. The model is a third-order, non-linear MIMO system with hard constraints and rate limits imposed on the actuators. It has fuel flow, control valve position, and feedwater flow as control inputs, and drum pressure, power output, and drum water level deviation as outputs. Although the model is of low order, it is capable of illustrating some of the complex dynamics associated with the real plant. The dynamics for the system are

$$\frac{dp}{dt} = -0.0018 u_2 p^{\frac{9}{8}} + 0.9 u_1 - 0.15 u_3 \tag{1a}$$

$$\frac{dP_o}{dt} = (0.073 u_2 - 0.016) p^{\frac{9}{8}} - 0.1 P_o \tag{1b}$$

$$\frac{d\rho_f}{dt} = \frac{\left(141 u_3 - \left(1.1 u_2 - 0.19\right) p\right)}{85} \tag{1c}$$

where p=drum pressure (kg/cm^2), P_o=power output (MW), and ρ_f=fluid density (kg/m^3). The normalized inputs to the system are u_1=fuel flow valve position, u_2=steam control valve position, and u_3=feedwater flow valve position. The following limitations are imposed on the valves:

0-7803-2685-7/95 $4.00 © 1995 IEEE

$$\left|\frac{du_1}{dt}\right| \le 0.007/\sec \tag{2a}$$

$$-2/\sec \le \frac{du_2}{dt} \le 0.02/\sec \tag{2b}$$

$$\left|\frac{du_3}{dt}\right| \le 0.05/\sec \tag{2c}$$

and all valve position variables are constrained to lie in the interval [0,1]. The outputs to the system are P (drum pressure), P_o (output power), and X_w (drum water level in meters). Pressure and power output are just the first two state variables, whereas water level is found through the auxiliary relationships

$$X_w = 0.05\left(0.13073\rho_f + 100\alpha_{cs} + \frac{q_e}{9} - 67.975\right) \tag{3a}$$

$$\alpha_{cs} = \frac{\left(1 - 0.001538\rho_f\right)\left(0.8p - 25.6\right)}{\rho_f\left(1.0394 - 0.0012304p\right)} \tag{3b}$$

$$q_e = \left(0.854u_2 - 0.147\right)p + 45.59u_1 - 2.514u_3 - 2.096 \tag{3c}$$

where q_e is the evaporation rate (kg/s) and α_{cs} is the steam quality.

At a load level of 66.65 MW, pressure of 108 kg/cm^2, and fluid density of 428 kg/m^3, the nominal inputs are found to be $u^o = [0.34 \quad 0.69 \quad 0.436]^T$. From these nominal values, a linearized model is obtained from a truncated Taylor series expansion of the non-linear equations:

$$\frac{d\bar{x}}{dt} = A\bar{x} + B\bar{u} \tag{4a}$$

$$\bar{y} = C\bar{x} + D\bar{u} \tag{4b}$$

where $\bar{x} = x - x^o$, $\bar{y} = y - y^o$, and $\bar{u} = u - u^o$. The linear system matrices are found to be

$$A = \begin{pmatrix} -2.509\times10^{-3} & 0 & 0 \\ 6.94\times10^{-2} & -0.1 & 0 \\ -6.69\times10^{-3} & 0 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.9 & -0.349 & -0.15 \\ 0 & 14.155 & 0 \\ 0 & -1.389 & 1.659 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 6.34\times10^{-3} & 0 & 4.71\times10^{-3} \end{pmatrix}, \text{ and}$$

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.253 & 0.512 & -0.014 \end{pmatrix}.$$

3. The GA Control System Design

The GA used in this paper is very similar to the algorithm that can be found in the standard literature on the topic [3], [7], [8], [9], also known as the *simple genetic algorithm.* We used the three-operator GA with only minor deviations from the original.

In this scheme, an initial population of binary strings is created randomly. Each of these strings represents one possible solution to the search problem. Next the solution strings are converted into their decimal equivalents and each candidate solution is tested in its environment. The fitness of each candidate is evaluated through some appropriate measure. The algorithm is driven towards maximizing this fitness measure. Application of the GA to an optimal control problem entails minimizing the quadratic cost function of the input and states. After the fitness of the entire population has been determined, it must be determined whether or not the termination criterion has been satisfied. This criterion can be any number of things. One possibility is to stop the algorithm at some finite number of generations and designate the result as the best fit from the population. Another possibility is to test if the average fitness of the population exceeds some fraction of the best fit in the population. If the criterion is not satisfied then we continue with the three genetic operators: reproduction, crossover, and mutation.

Fitness-proportionate *reproduction* is effected throught the simulated spin of a weighted roulette wheel. The roulette wheel is biased with the fitnesses of each of the solution candidates. The wheel is spun N times where N is the number of strings in the population. This operation yields a new population of strings that reflect the fitnesses of the previous generation's fit candidates. The next operation, *crossover*, is performed on two strings at a time that are selected from the population at random. Crossover involves choosing a random position in the two strings and swapping the bits that occur after this position. The final genetic operator in the algorithm is *mutation.* Mutation is performed sparingly, typically every 100-1000 bit transfers from crossover, and it involves selecting a string at random as well as a bit position at random and changing it from a 1 to a 0 or vice-versa. After mutation, the new generation is complete and the procedure begins again with fitness evaluation of the population.

In a control system design using the GA, the parameters that are represented as binary strings are the relevant control parameters. In the design of the proportional-integral (PI) control system, the parameters are the 12 proportional, integral, and cross-coupling gains in the coupled controller illustrated in Fig. 1. In the linear quadratic regulator (LQR) controller the parameters are the 9 state feedback gains in the standard LQR configuration. In each of the control designs the quadratic performance index is selected as

$$J = \int_{t_0}^{tf} \left((y - y^{ref})^T Q (y - y^{ref}) + u^T R u\right) dt \qquad (5)$$

PI Controller Design

In the coupled PI controller design the gains are trained in 5 stages. The first stage consists of training the *proportional gains* in the PI controller only and leaving the others fixed at zero. The initial gains are selected at random between some coarse upper and lower bounds and tuned through each genetic iteration. Once some pre-specified convergence criterion has been achieved, the best fit triplet of proportional gains is designated as the result for the stage. Stage 2 extends the strings in the population to include the next three *integral gains*. The bounds on the first three values in the string (the proportional gains) are between ± 25% of the result from the previous stage. This constrains the first three gains to be around their previous best value, and yet allows fine-tuning with the introduction of the new integral gains which are set initially to random values. Again, the genetic algorithm is executed until some convergence criterion has been attained. This procedure is repeated in the same manner with the string lengthened by two *cross-coupling gains* which represent the coupling of the first input error to the other two. This is done two more times to train the final four coupling gains.

Each stage of training entails an alternating reference demand change in pressure and power. After the system has been trained to perform well with one reference change, it is then trained to perform well with a different reference demand change. This is done until some convergence criterion has been attained. By training the controller gains in this manner the gains are tuned to some average performance between two high-performance results. The purpose of this method is to design a controller that performs well at different operating points.

The goal of the controller is to track step demands in power and pressure. To achieve this the following performance index is selected to be minimized:

$$J = J_0 + J_{ss},$$

where J_o is defined by (5). In addition, we penalize the steady-state error through computation of

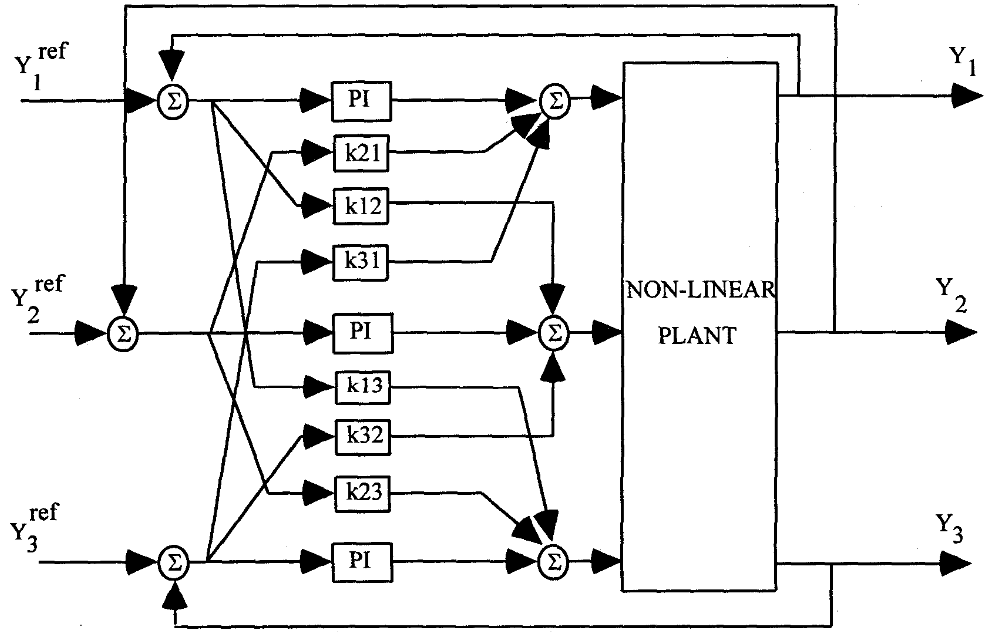

Fig. 1. Coupled PI controller structure.

$$J_{ss} = \sum_{i=1}^{3} e_i(t_f)$$

where

$$e_i(t) = y_i(t) - y_i^{ref}(t).$$

The weighting matrices are selected as $Q = I_{3\times 3}$,

$$R = \begin{pmatrix} 724 & 0 & 0 \\ 0 & 141 & 0 \\ 0 & 0 & 5 \end{pmatrix}.$$

The values of these matrices can be determined from the relative importance of the variables and from the relationships of the outputs to the inputs in the steady-state. Since the order of the system has been increased to six by including the integral action in the controller, we weigh the steady-state error heavily in the hopes of achieving perfect steady-state tracking.

The GA parameters are selected for the training cycle as:

Population size	64
Crossover rate	0.9
Mutation rate	0.001
Parameter resolution	9 bits per substring

Each training cycle is executed until either 30 generations have elapsed or 95% convergence has been achieved, and the fitness function used in a training cycle is $10^7/J$.

LQR Controller Design

The goal of the genetic algorithm is to determine the matrix gains in the feedback path to ensure tracking of the reference signal over a wide operating range. The controller structure is simply a state feedback matrix in the feedback path and a feedforward gain matrix to ensure tracking. This feedforward matrix is found using the transfer function from the input to the output. To obtain reasonable tracking over a wide operating range, the GA trains the gains of the feedback matrix with a demand that covers the nominal as well as large load changes.

The genetic algorithm is implemented with a number of different genetic parameters (i.e., crossover rate, mutation rate, population number, etc.). In an effort to improve convergence, the individual matrix elements' resolutions are all increased at a specified generation of the run. At this specified time, all of the substrings are lengthened by two bits. The old matrix element values are retained but at a greater precision. We select this lengthening of the bits to occur one-third of the way through the optimization run. We choose the following parameters:

Population size	100
Crossover rate	0.8
Mutation rate	0.2
Terminating generation	30
Matrix element interval	[-1,1]
Initial resolution	15 bits per substring
Resolution after 10 gen.	17 bits per substring

In addition to these parameters, the initial population is seeded with 3 members from a previous run's "best fit" population with the remainder of the population constructed at random. This is another attempt to obtain quicker convergence.

In the GA optimization runs, the fitness function is selected as $100/(1+J)$ where J is identical to (5). In addition, the weighting matrices are identical to those in the PI controller design.

4. GA Design Results

The performance of each of the control systems is tested with the following inputs to the system,

$$y_{1ref}(t) = 108 + (120 - 108) \times 1^{+}(t - 200)$$

$$y_{2ref}(t) = 66.65 + (120 - 66.65) \times 1^{+}(t - 600)$$

$$y_{3ref}(t) = 0.$$

This reference change in pressure and power represents a relatively large demand change in both variables and illustrates the merits and weaknesses of each design.

GA/PI Controller Results

The performance of the system with the GA-designed coupled PI controller is illustrated in Figs. 2-4. Table I contains the controller gains at each stage of training. The drum pressure is very oscillatory with a large overshoot. This response settles out with a steady-state error of 0.175. The power response is good during the first change in pressure at 200 seconds. As can be seen, the change has little influence on power. When the demand in power steps up to 120 MW the power output overshoots by about 5 MW for a duration of less than 200 seconds. This overshoot can be attributed to the saturation of the control valve for that period of time (not shown). Again, the change in power has little effect on pressure at 600 seconds. The steady-state error of the power is 0.0004 MW. The drum water level deviation is acceptable during both transitions in pressure and power. The largest deviation of the level occurs during the power demand change at 600 seconds when the level exceeds 0.15 meters from the nominal level. The steady-state error for the level is 0.000243 meters.

Table I. Gains at each training stage.

	I	II	III	IV	V
Kp1	11.3883	8.9675	10.2678	11.7566	11.1185
Kp2	0.0793	0.0009	0.0007	0.0005	0.004
Kp3	1.2051	0.9579	1.0423	1.1341	1.1631
Ki1	-	0.0041	0.0038	0.0036	0.0033
Ki2	-	0.0191	0.0149	0.0117	0.0093
Ki3	-	0.0269	0.0238	0.0211	0.0186
K12	-	-	0.0405	0.0386	0.0292
K13	-	-	0.0979	0.1214	0.1344
K21	-	-	-	0.0381	0.0468
K23	-	-	-	0.0950	0.0875
K31	-	-	-	-	0.0842
K32	-	-	-	-	0.0699

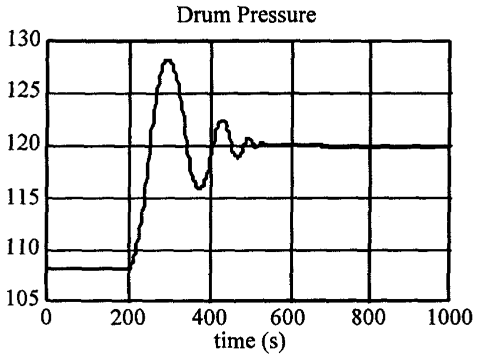

Fig. 2. GA/PI pressure response.

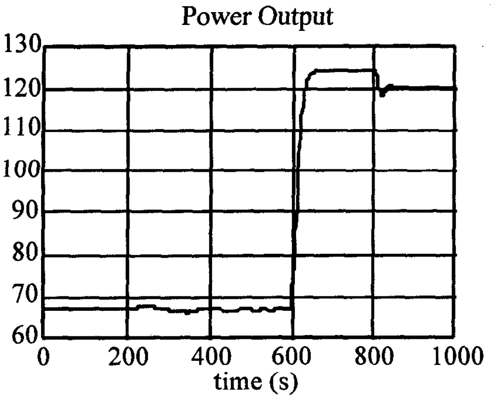

Fig. 3. GA/PI power response.

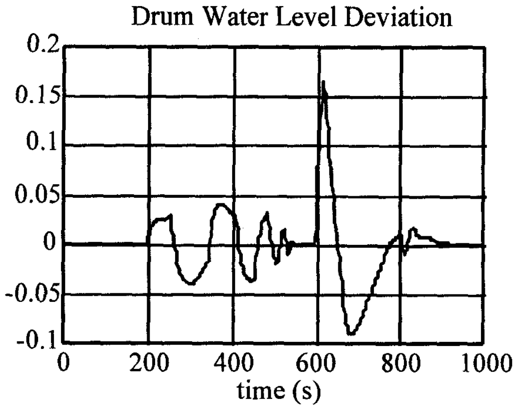

Fig. 4. GA/PI level response.

GA/LQR Controller Results

The performance of the system with the GA-designed LQR controller is shown in Figs. 5-7. The result of the GA/LQR design is the state feedback matrix,

$$K = \begin{pmatrix} 0.0354 & 0.2236 & -0.1451 \\ -0.0430 & 0.0588 & 0.0039 \\ 0.1609 & -0.6316 & 0.4275 \end{pmatrix}.$$

It should be noted that, to ensure steady-state tracking, that a feedforward gain matrix is added to the system. This is obtained by calculating the system transfer function and finding a suitable matrix, F, so that the system tracks step demands in the steady-state. The values of the matrix are found to be

$$F = \begin{pmatrix} 0.0341 & 0.0018 & 0.6567 \\ -0.0051 & 0.0903 & -1.3537 \\ -0.4234 & -0.4088 & 94.3416 \end{pmatrix}.$$

The pressure tracks the initial demand change well and there is a 5 kg/cm^2 pressure bump when the power demand is stepped up. The steady-state error in pressure is 0.252 kg/cm^2. The power response shows a large drop (25 MW) when the pressure demand is stepped up at 200 seconds. The power tracks the step demand at 600 seconds well achieving a steady-state error of 0.68 MW. The drum water level deviation undergoes relatively large deflections at each demand change in pressure and power. The largest of these is a level drop of 0.25 meters. The steady-state error in level is 0.0425 meters, due in large part from the deviation from the nominal operating point.

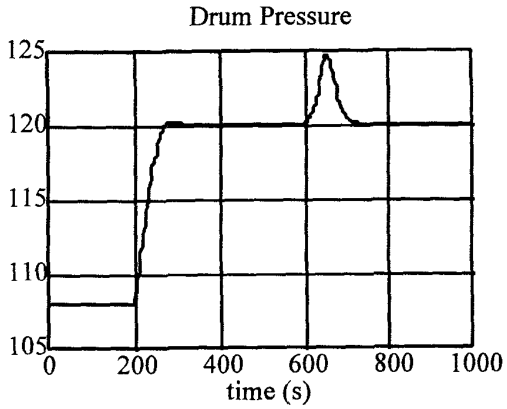

Fig. 5. GA/LQR pressure response.

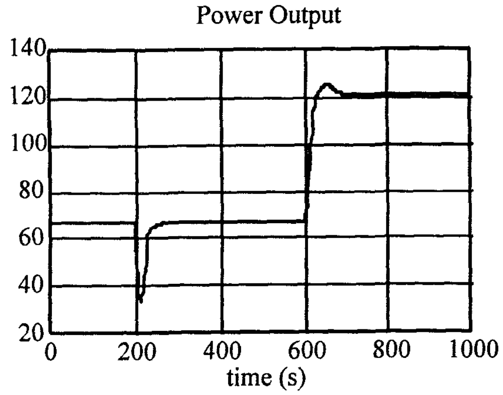

Fig. 6. GA/LQR power response.

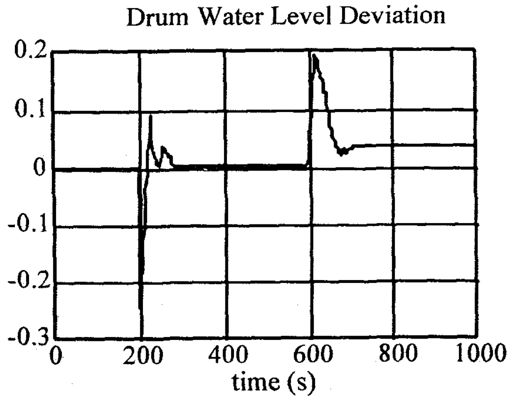

Fig. 7. GA/LQR level response.

5. Conclusions

In this paper, a control system design methodology for a non-linear MIMO boiler-turbine plant was presented. A brief summary of the GA was presented and its application to power plant control system design was discussed. The results of the GA design of a coupled PI controller and output feedback controller were presented. It was found that the GA/PI control system achieved good steady-state tracking but oscillations due to the integral action were prevalent. The GA/LQR control system performed well but at the cost of small but finite steady-state error.

6. Acknowledgements

This work is supported in part by a NSF grant, EID-9212132, Research and Curriculum Development for Power Plant Intelligent Distributed Control, and NSF/EPRI joint project, ECS-9216504/RP8030-4, Experimental Development of Power Reactor Intelligent Control.

7. References

[1] A. Ben-Abdennour, K.Y. Lee, and R.M. Edwards, "Multivariable robust control of a power plant deaerator," *IEEE Transactions on Energy Conversion*, 8, pp. 123-129, March 1993.

[2] I.G. Klefenz, *Automatic Control of Steam Power Plants, 3rd Edition*, Bibliographisches Institut, 1986.

[3] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, & Machine Learning*, Addison-Wesley, 1989.

[4] R.M. Brady, "Optimization strategies gleaned from biological evolution," *Nature*, 317, pp. 804-806, 1985.

[5] K. Krishnakumar and D.E. Goldberg, "Control system optimization using genetic algorithms," *J. Guidance, Control, and Dynamics*, 15, pp. 735-740, 1992.

[6] R.D. Bell and K.J. Åström, "Dynamic models for boiler- turbine-alternator units: Data logs and parameter estimation for a 160 MW unit," Report TFRT-3192, Lund Inst. of Tech., Sweden, 1987.

[7] John H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.

[8] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

[9] L. Davis (Ed.), *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, 1987.