Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003

AN OVERVIEW OF GENETIC ALGORITHMS APPLIED TO CONTROL ENGINEERING PROBLEMS

QING WANG[1], PIETER SPRONCK[2], RUDOLF TRACHT[3]

[1]Control and Simulation Center, Harbin Institute of Technology, 150001, Harbin, P. R. China
[2]IKAT Institute, Maastricht University, The Netherlands
[3]Essen University, Germany
E-MAIL: qingwang@hit.edu.cn

Abstract:

Genetic Algorithms (GAs) are the most widely known evolutionary search algorithms. While they are regularly applied to control engineering problems, currently they are not a standard tool in the control engineer's toolbox. This may in part be the result of the fact that few general overview of the application of GAs to control engineering problems yet exists, and the fact that they are usually reported on at conferences of computer scientists, not of control engineers. This paper attempts to alleviate that omission by presenting an overview of recent applications of GAs in the field of control engineering.

Keywords:

Genetic algorithms; evolutionary computation; control engineering.

1. Introduction

Genetic Algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics. They were invented in 1975 by John Holland of the University of Michigan[1]. After David Goldberg gave a basic framework of GAs in his popular book "Genetic Algorithms in Search, Optimization & Machine Learning"[2], they have received considerable and increasing interest. GAs are applied in many different areas, such as signal processing, game playing, robotics, image segmentation, scheduling and control engineering. Evolutionary techniques related to GAs, such as Evolutionary Programming (EP)[3], Evolution Strategies (ES)[4] and Genetic Programming (GP)[5], are similar in their process and strategies and vary mainly in implementation details. In recent years the boundaries between these different evolutionary approaches have broken down to some extent, with researchers combining aspects of the various algorithms.

Papers on the application of GAs in control engineering can be found in various conference proceedings and journals. They cover a wide range of forms of control, including PID control, optimal control, adaptive control, robust control and system identification. General surveys of GAs in control engineering are, however, rare[6,7]. This paper aims at introducing recent applications of GAs in control to researchers in the field of control engineering. The important characteristics of GAs and their relevance to problems in control engineering are considered and future applications in this field are prospected.

2. Characteristics of GAs

GAs are search and optimisation techniques inspired by two biological principles, namely the process of "natural selection" and the mechanics of "natural genetics". Contrary to regular search algorithms, GAs manipulate not just one potential solution to a problem, but a collection of potential solutions, called a population. The potential solutions in the population, called "individuals" or "chromosomes", are encoded representations of all the parameters of the solution. Each chromosome is awarded a fitness rating that indicates how successful this particular solution is compared to the other chromosomes in the population. To evolve chromosomes that encode better solutions, the GA employs so-called "genetic operators", such as crossover and mutation, to create new chromosomes from the existing ones in the population, by either merging two or more parent chromosomes or by modifying an existing chromosome. The selection mechanism for parent chromosomes takes the fitness of the parents into account, ensuring that the better solutions have a higher chance to procreate and donate their beneficial characteristics to their offspring. Newly generated individuals in time replace the existing ones. Through this process after a while the population will converge to a "best" solution.

Essentially a GA is a random search mechanism, but its inherent randomness is guided towards better performance through the selection mechanism. Due to this inherent randomness, GAs usually are resource intensive, and it is not guaranteed that the optimum solution will be derived, not even a mediocre one. On the other hand, GAs are universally applicable, because they need only a good fitness function to work, which is a requirement for any optimisation technique[8]. Therefore, the application of GAs is most suitable for problems for which no good dedicated solution mechanism exists.

3. Application of GAs in Control Engineering

Control system design must take into account a number of performance issues, such as system stability, the static and

0-7803-7865-2/03/$17.00 ©2003 IEEE

dynamic index, and system robustness. Each of these issues strongly depends on the controller structure and parameters. However, this dependence usually cannot be expressed in a mathematical formula. Additionally, often a trade-off has to be made among conflicting performance issues.

Obviously the lack of a systematic and intuitive approach to select values for a large number of control parameters is a big obstacle when attempting to obtain a satisfactory control system. To solve these problems by GAs, we can encode the structure and parameters of the controllers into a chromosome, and define a fitness measure as a function over the performance demands, thus formulating the design problem as the minimisation of an objective function with respect to the controller parameters. Since GAs only need a fitness function to guide the optimisation process, they can be employed to execute this search. The creative combination of a variety of pre-existing control methodologies and GAs can result in a powerful tool that is able to address real engineering control problems. The remainder of this section will focus on the use of GAs for specific control problems.

3.1. Multiobjective Control

Many real world problems involve multiple objectives that must simultaneously be achieved. A suitable optimal solution meeting all the objectives usually is hard to find since the objectives often are in conflict. In general the solution to a Multiobjective Optimisation (MO) problem is not one single point, but a family of non-dominated, alternative points, known as the Pareto-optimal set[9], which describes the trade-off among contradicting objectives. The Pareto front yields a set of candidate solutions, from which we pick the desired one under different trade-off conditions. GAs are a suitable technique for solving MO problems, since GAs can search for multiple solutions in parallel, producing a family of possible solutions to a problem.

GAs have the ability to handle complex problems involving discontinuities, non-differentiability, and multi-modality. A Pareto-optimal set can be identified by a collection of different individuals generated by the evolution process[10]. One of the first approaches to utilise the concept of Pareto optimality in GAs was Fonseca and Fleming's multiobjective genetic algorithm (MOGA), which is applicable to control engineering problems[6]. In a later paper they proposed a unified decision making framework for MO problems encompassing multiple constraints[9]. As a demonstration of the proposed method they gave the optimisation of the low-pressure spool speed governor of a gas turbine engine.

A new framework for multiobjective fuzzy GA optimization was proposed by Trebi-Ollennu and White[11]. They used a GA to select free control parameters of an input-output linearising controller with sliding mode control for the depth control system of a remotely-operated underwater vehicle. The relative importance of the objective functions was assessed by using a new membership weighting strategy.

3.2. PID Control

A multivariable adaptive digital tracking PID controller was presented by Zuo[12]. He used a GA to tune the controller on-line, so that the defined performance index of closed-loop systems was minimised and the desired behaviour of closed-loop systems was achieved. The controller was applied to the attitude control and momentum management system for a space station. The claimed results were remarkable in stability robustness and setpoint tracking behaviour with respect to the large moment-of-inertia variations from 200 to 400%.

Chen and Cheng[13] presented a procedure to tune PID parameters to achieve mixed $H2/H\infty$ optimal performance consisting of the following three steps. (1) Based on the Routh-Hurwitz criterion the stability domain of the three PID parameter space, which guarantees the stability of the closed loop system, was specified. (2) The subset of the stability domain in the PID parameter space in step one was then specified so that the H_∞ constraint was satisfied. (3) The design problem in the subset domain of the H_∞ constraint domain given in step 2 was redefined as the search for one point which minimises the H_2 tracking performance. This is generally considered to be a highly nonlinear minimisation problem, in which many local minima may exist. They therefore used a GA for the minimal point search.

3.3. Optimal Control

Robandi, Nishimori *et al.*[14] proposed a method to search for elements of the matrices Q and R with a GA and applied the method to a complex power control system for the case where various small load disturbances exist. Simulation results showed the method gave a new alternative procedure in time-varying feedback control to improve the stability performances.

3.4. Robust Control

H_∞ control

In the case of H_∞ control, Chen[15] designed a robust controller for the Permanent Magnet Linear Synchronous Motor which allowed mass variation of the moving part ranging from 0 to 100 percent of a nominal load. To minimise the error between the actual response and the reference, the controller parameters were optimised by a GA. Their simulation and experimental results both showed that the system could achieve robust performance under such a large load variation. In the case of the H_∞ loop-shaping design procedure, Tang *et al.*[16] incorporated GAs to search the shaping function space in order to find a suitable robust controller and close-loop performance.

Chen and Cheng[17] used a GA to design a controller directly. They implemented a structure-specified H_∞ controller for systems with parameter variations and disturbance uncertainties designed by a GA from a suboptimal point of view. First an admissible domain of controller parameters was determined according to the Routh-Hurwitz stability criterion. The design problem was then reduced to finding an optimal parameter vector by use of a GA in this admissible domain such that the $H\infty$ performance restriction was achieved.

Eigen-structure assignment

Patton and Liu[18] combined Eigen-structure assignment and gradient-based optimisation with a GA. The sensitivity and the complementary sensitivity functions of the closed-loop were taken as the robust control performance index. The GA handled the performance optimisation. The controller was developed for a lateral flight control system. In a simulation the resulting controller was determined to be a preferred solution.

Lyapunov's direct-control

Ge and Lee[19] used GAs to design a Lyapunov's direct-controller for a single-link flexible robot system, taking into account both system stability and desired performance. The objective was to drive the tip of the flexible beam to a predefined position as fast as possible with minimal overshoot and oscillation. To achieve a good trade-off between the joint motion speed and the tip deflection size, the feedback gains of the controller were tuned by a GA optimisation process. Further investigation was done into the control of a more complex plant, namely a flexible spacecraft with one flexible appendage[20].

LQG Control

Combining LQG design with a GA, Mei and Goodall[21] presented an effective solution for the active steering of railway vehicles. Using the LQG approach, the system stability was guaranteed. Then, using a GA to search for the best values for the weighting factor matrix, the control design was concentrated to get the optimal performance while compromising between the curving of the rail, the fast travelling speed, the stability of the vehicles and the comfort of the passengers.

Stochastic Robustness Control

Considering the fact that many deterministic worst-case robust analyses and syntheses were unduly conservative, while the resulting controllers usually needed a very high control effort, Marrison & Stengel[22] and Wang & Stengel[23] have studied the probabilistic robustness method in combination with a GA. The goal of their design was to find the optimal controller parameters that minimised the stochastic robustness cost function, which was formalised by combining the probabilities of different design requirements with certain trade-offs. The probabilities were estimated by Monte Carlo Evaluation (MCE)[24], which was a practical and flexible approach to the problem. The discrepancy between the results of the MCE and the true values resulted in apparent "noise" in the evaluation of the cost function and the fact that the cost function was not convex. To alleviate these problems, a GA was used to search for the control design parameters. Their results showed excellent stability and performance robustness.

Stability Robustness Analysis

Fadali and Zhang[25] reduced stability robustness analysis for linear, time-invariant, discrete-time systems to a search for determining whether the root of closed-loop characteristic polynomials is located inside the unit circle. They solved it by applying a GA. In their implementation the coefficients of the closed-loop characteristic polynomials could be interval, affine, multilinear or even exponentially dependent on the uncertain plant parameters.

Sliding mode control

One of the main underlying problems associated with SMC is the lack of an optimal and systematic way of feedback gain selection, which becomes more serious for large numbers of feedback gains. One solution was proposed by Al-Hamouz *et al.*[26] by formulating the feedback gain selection as an optimisation problem after which a GA can be applied to perform the optimisation process. The application of the proposed method to the load frequency control problem of a power system revealed that both the dynamic system performance and the control effort improved dramatically.

Another major problem associated with SMC is "chattering" due to imperfections in switching devices and delays. An adaptive fuzzy SMC with GA-based continuous-type reaching law was presented by Su *et al.*[27] for a class of non-linear plants. Where a GA was used to optimise the parameters of the reaching law, the undesirable chattering phenomenon was effectively suppressed provided the size of the boundary layer of the sliding control was chosen large enough. Moreover, the reaching dynamics could be significantly improved during the reaching phase.

3.5. Intelligent Control

The control of complex dynamical systems, such as nonlinear, time-varying, environmentally uncertain, and imprecisely defined systems is still a challenging problem. Solutions for most of these complex systems can only be obtained through the accumulation of information from system responses and control experts, and then using this information to dynamically generate an acceptable solution. Among others, neural network control (NNC) and fuzzy logic control (FLC) methods have proven to be effective for such complex systems[28].

Fuzzy Logic Control (FLC)

In order to efficiently design a controller while assuring high performance, the fusion of FLC and GA is steadily growing, mainly to optimise fuzzy rules and/or fuzzy membership functions[29]. Tarng *et al.*[30] proposed an automatic synthesis of membership functions based on a GA to control non-linear and time-varying tuning processes. The seven linguistic sets in the membership function base and the scaling factors of input and output were encoded as the chromosome. The summation of the square-root error was used as fitness function. The effectiveness of the technique was shown by a computer simulation and by experimental verification.

Herrera *et al.*[31] presented a three stages fuzzy rule learning process based on a GA. The process consisted of the following three elements: (1) a fuzzy rule genetic generating process based on a rule learning iterative approach; (2) a genetic process for combining the generated rules with experts rules and removing the redundant ones; and (3) a genetic tuning process to adjust the membership functions of the rules. The inverted pendulum control problem was successfully used as a test case.

Shieh[32] proposed a stability criterion and a robust controller for continuous uncertain systems with state time-varying delay. In order to achieve enhanced performance, an FLC with a small number of rules and membership functions, which were automatically adjusted on-line by a GA, was induced to the robust controller. The chromosome consisted of a rulebase table and input-output membership function encoded as a binary string. As fitness function the system performance index was used.

Neural network control (NNC)

The search for a successful ANN for a specific problem is basically a search for the global minimum in the space of errors generated by all possible ANNs working on a set of training samples for this problem. This set of training samples usually consists of a collection of plant states with corresponding control inputs. However, it is not always possible to construct such a set (for instance because the plant has inner states which are not visible to the controller), in which case the error can be defined as the result on a simulation run[33]. The error space is noisy, since small changes in the weights of a network may affect the error significantly, and multimodal, since for most mappings many different ANNs exist which represent them. GAs are particularly suited to handle the problem of ANN optimisation.

GAs have been used to evolve ANNs in three main ways: (1) for network architecture design, including the number of hidden layers, the number of nodes within the layers and connectivity; (2) for determination of the connection weights; and (3) for selection of the ANN parameters, such as the learning rate and momentum coefficient. For control problems architecture design and weight determination can often be combined in a GA with good results[34], which is a boon most regular ANN training methods do not have. Yao[35] gave a fairly complete overview of the use of GAs to evolve ANNs in general.

Stanley *et al.*[36] presented a method, which they called NeuroEvolution of Augmenting Topologies (NEAT), that evolved an ANN topology in addition to the connection weights. To efficiently evolve the network, their method comprised three principles: (1) starting the evolution from a minimal structure and growing it only when necessary; (2) designing a genetic representation (using historical markings to line up genes with the same origin) that allowed disparate topologies to merge in a meaningful way; and (3) separating each new structure into a different species so that it is protected from interference from other species and has time to optimise its structure before it has to compete with other niches in the population. The efficacy of NEAT was demonstrated on the benchmark double pole balancing control problem.

Neuro-fuzzy Control

Neuro-fuzzy systems combine the learning capability of neural networks with the knowledge representation of fuzzy logic. Typically, the fuzzy model is transferred into a neural network-like architecture, which then is trained by some learning method.

Seng *et. al.*[37] proposed a method for tuning a neuro-fuzzy-logic controller using a GA. All of the parameters of the controller, i.e. the width and centre of the membership functions, and the weights of the ANN were tuned simultaneously. Dynamic crossover and mutation probabilistic rates were also applied for faster convergence of the GA evolution. The method was applied to a liquid-level control system with non-linear dynamics, in real time, and then compared with a conventional FLC and a PID controller in terms of step response, load disturbance and changes in plant dynamics. It was observed that the proposed method showed considerable robustness and advantages. They also applied their method to an unstable and non-minimum phase plant, and an automated car parking system[38].

4. System Identification

There are many good traditional methods for system identification, such as least-squares and maximum-likelihood, but most of these are for linear or linear-in-the-parameters non-linear systems, and based upon the assumption of a smooth search space. The model-determination therefore often fails in the search for a global optimum if the search space is not differentiable or linear-in-parameters. Furthermore, these traditional methods still suffer from various problems, such as the facts that (1) initial information on the system parameters is needed for convergence; (2) estimated parameters may be biased if the noise is correlated; and (3) they cannot easily be applied to non-linear systems. Techniques for the selection of structure and for non-linear-in-the-parameters identification are still an open issue[7].

GAs can be applied to continuous- and discrete-time system, both on-line and off-line and both time domain and frequency domain systems, and can directly identify physical parameters or poles and zeroes of the system. A thorough study (including all issues mentioned above) was made by Kristinsson and Dumont[39] for linear systems. Their simulation results showed that the algorithm was robust and was able to converge towards the actual value of the parameters.

For the case of poles and zeroes identification, one example was given by Reeves[40] based on a hybrid GA. The parameters were encoded as radii and angles of poles/zeroes with ranges of [0, 0.99] and [0, 2π] respectively to keep the system stable and at a minimum phase. To prevent premature convergence which often affects incremental Genetic Algorithms, they combined the classical Golden Section method with their GA. They first used a very crude precision and allowed the GA to converge, after which they reduced the search range of the parameters to get a high resolution. The hybrid method was applied to an unknown system (gas engine) identification problem. They found the results outperformed the traditional least squares methods.

For structure identification, Luh and Wu[41] developed a GA-based non-linear autoregressive with exogenous inputs system identification (GANARXSI) algorithm to identify non-linear systems. They applied this to both non-linear continuous-time and discrete-time systems with reasonable accuracy. To improve the convergence rate, they proposed a truncation mutation operator. An identified non-linear coupled liquid-level system was used to evaluate the performance of the algorithm. They found the results to be a practical technique for identification for non-linear systems.

Billings and Mao[42] discussed details of non-linear rational model for the simultaneous structure detection and parameter identification using a GA. Compared with other approaches, the proposed algorithm had two advantages. Firstly, the algorithm did not require a linear-in-the-parameters regression equation and, as a consequence, severe noise problems were avoided. Secondly, the algorithm provided near-optimal global parameter estimation. The simulation results illustrated that the algorithm worked well on systems with modest system structure and parameter identification, but could fail for larger systems.

5. Online Adaptive Identification and Control

GA based controllers have the ability to adapt to a time-varying environment (changes in plant or disturbances from outside) and may be able to maintain good closed-loop system performance. However, the stochastic and time-intensive nature of GAs presents a serious problem for on-line real-time applications, with respect to the determination of a correct control action between limited sample times. To solve this problem, the following two issues must be considered. Firstly, as far as the plant is concerned, only those that allow a relatively long sample time (long enough for the GA to complete the convergence process) can be used for GA-based identification and control. Secondly, the time needed for fitness evaluation of candidate solutions should be short, and the active control scheme should be ensured at each generation.

Specific GA methods for online optimisation have been developed. One example is the Incremental GA (IGA)[43], in which only one chromosome from a population is evaluated each time interval, while the other chromosomes in the population are evaluated in successive time intervals. Linkens and Nyongesa's IGAs[44] also evaluated one chromosome at each sample time, but the fitness of the remainder of the population was estimated based on this one evaluation. A final example is the microGA, in which simply a very small population is used.

Lennon and Passino[45] developed a general genetic adaptive controller (GGAC). They used genetic adaptive identification to estimate the parameters of the model that was used in the fitness function for the direct genetic adaptive controller. The GGAC identified the plant model and tried to tune the controller at the same time, so that if the estimates were inaccurate, good control could still be achieved. Since GAs are stochastic processes, it is possible that good controllers will not be found and thus degrade the system performance. One method they used to alleviate this problem was to seed the population of the GAs with some individuals that remain unchanged in every generation. These fixed controllers were distributed throughout the control parameter space to ensure that a reasonably good controller was always present in the population. Based on those fixed controllers the GA could find an acceptable chromosome quickly and then search nearby solutions to find better ones. In their simulations they used 25 fixed controllers and 75 controllers to be adapted by regular GAs.

The number of real-time adaptive control experiments with GAs is still very limited. Ahmad *et al.*[46] investigated the online GA tuning of a PI controller for a heating system with both a time-invariant plant model and a time-varying plant model. In the time-varying case the model was estimated each time step using a recursive least square (RLS) parameter estimator. The objective of their experiments was to achieve the desired temperature as quickly as possible with minimal overshoot. The model was run between the sampling intervals of the experiment to obtain and evaluate the cost function for each pair of gains generated by the GA. The population size was restricted to 60 to reduce computational time.

Another real-time implementation[47] used an adaptive sliding-mode position controller based on real-time GAs for an induction motor servo drive. First, an adaptive SMC with an integral-operation switching surface was investigated, in which a simple adaptive algorithm was utilized to estimate the boundaries of uncertainties. The adaptation gain in the adaptive algorithm was tuned on-line by a real-time GA in order to prevent sluggish or chattering responses due to a large external load disturbance. The population size was 20 and the number of generations 10. The simulation and experimental results clearly showed robust control performance of the adaptive controller based on a real-time GA both in the tracking and the load regulation.

6. Conclusion

Many successful applications of GAs for controller design indicate that GAs can be a powerful tool in the hands of a control engineer. In particular the fact that GAs require nothing more than a fitness measure to work and pose no restrictions to the problem at hand, gives them an edge over most regular methods in dealing with non-linear systems and uncertainty. We therefore conclude that control engineers should consider the use of GAs when they are faced with a control problem and the regular techniques cannot handle very well ,provided their application can accept the resource intensive nature of GAs.

References

[1] Holland, J.J. Adaptation in Natural and Artificial Systems, University of Michigan Press. 1975

[2] Goldberg, D.E. Genetic Algorithms in Search, Optimization and Machine Learning, Addison- Wesley. 1989

[3] Fogel, C.J, A.J. Owens and M.J. Walsh. Artificial Intelligen- ce Through Simulated Evolution, Wiley Publishing, New York. 1966

[4] Rechenberg, L. Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Frommann-Holzboog, Stuttgart. 1973

[5] Koza, J.R. Evolving a Computer Program to Generate Rand- om Numbers Using the Genetic Programming Paradigm, Proc. 4th Int. Conf. On Genetic Algorithms, Morgan Kaufmann, La Jolla, CA.: 37-44, 1991

[6] Man, K.F., Tang, K.S. Genetic algorithms for control and signal processing. 23rd Int. Conf. on Industrial Electronics, Control and Instrumentation, Vol. 4: 1541 –1555. 1997

[7] Fleming, P.J., R.C. Purhouse. Genetic Algorithms in Control Systems Engineering, Research Report No. 789, University of Sheffield Sheffield, UK,2001

[8] Michalewicz, Z., D. B. Fogel. How To Solve It: Modern Heuristics, Springer-Verlag. 2000

[9] Fonseca, C.M, P.J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algori-thms-Parts I&II, IEEE Trans. on SMC-Part A. 28(1): 26-47, 1998

[10] Lu, H. M., Gary G. Yen. Multiobjective optimization design via Genetic algorithm, Proc. of the 2001 IEEE Int. Conf. on control applications: 1190-1195.

[11] Trebi-Ollennu and B.A. White. Multiobjective fuzzy genetic algorithm optimization approach to nonlinear control system design, IEE Proc.-Conrol Theory Appl., 144(2): 137-142. 1997

[12] Zuo, W. Multivariable Adaptive Control for a Space Station Using GAs, IEE Proc.-Control Theory Appl. 142(2): 81-87, 1995

[13] Chen, B. S, Cheng, Y. M. A Genetic Approach to Mixed Optimal H2/H∞ PID Control, IEEE Control Systems Magazine, 1995(10):51-60

[14] Robandi, I, K. Nishimori. Optimal feedback control design using genetic algorithm in multimachine power system, Electrical Power and Energy Systems 23 (2001): 263-271.

[15] Chen Y.R., N.C. Cheung. H-inf robust control of permanent magnet linear synchronous motor in high-performance motion system with large parametric uncertainty. Power Electronics Specialists Conference. IEEE 33rd Annual. Vol. 2: 535 -539, 2002

[16] Tang, K.S., K.F. Man, et al. Structure Genetic Algorithm for Robust H infinity Control Systems Design, IEEE Trans. on Industrial Electronics, 43(5): 575-581,1996

[17] Chen, B. S. and Cheng, Y. M. A Structure-Specified H-Infinity Optimal Control Design for Practical Applications: A Genetic Approach, IEEE Trans. on Control Systems Technology, 6(6): 707-718,1998

[18] Patton, R.J. and G.P. Liu. Robust Control Design via Eigenstructure Assignment, Genetic Algorithms and Gradient-based Optimisation, IEE Proc.- Control Theory Appl. 141(3),1994

[19] Ge, S. S., Lee T. H, Zhu, G. Genetic Algorithm Tuning of Lyapunov-Lased Controllers: An Application to a Single Link Flexible Robot System. IEEE Trans. on Industrial Electronics, 43(5): 567-573

[20] Ge, S.S., T.H. Lee. Robust Controller Design with Genetic Algorithm for Flexible Spacecraft, Proc. of the 2001 Congress on Evolutionary Computation: 1033-1039, Korea.27-30 May.

[21] Mei, T.X. and Goodall, R.M. LQG and GA solutions for active steering of railway vehicles, IEE Proc.-Control Theory and Appl. 147(1): 111-117, 2000.

[22] Marrison, C.I., and R.F. Stengel. Robust Control System Design Using Random Search and Genetic Algorithms, IEEE Trans. Automat Control, Vol. 42: 835-839,1997

[23] Q. Wang, R. F. Stengel, Robust control of nonlinear systems with parametric uncertainty, Automatica 38 (2002): 1591-1599

[24] Kalos M H and Whitlock P A. Monte Carlo methods.New York:Wiley, 1986

[25] Fadali, M.S., Y. Zhang. Robust Stability Analysis of Discrete-Time System Using Genetic Algorithms, IEEE Trans. on Sys. Man and Cyber. 29(5): 503-508. 1999

[26] Al-Hamouz and Al-Duwaish, A new load frequency variable structure controller using genetic algorithms, Electric Power Systems Research 55: 1–6, 2000

[27] J. P. Su , T. M. Chen, et al. Adaptive fuzzy sliding mode control with GA-based reaching laws Fuzzy Sets and Systems 120: 45－158

[28] Linkens, D. A., Nyongesa, H. O. Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic control applications, IEE Proc.-Control Theory and Appl. 143(4): 367-386, 1996.

[29] Cordon, O., F. Herrera, et al. Recent advances in genetic fuzzy systems, Information sciences, 136(2001): 1-5

[30] Tarng, Y.S., C.Y. Nian. Automatic synthesis of membership functions for the force control of turning operations, Journal of Materials Processing Technology 65(1997): 80-87.

[31] Herrera, F., M. Lozano. A learning process for fuzzy control rules using genetic algorithms, Fuzzy sets and systems, 100(1998): 143-185

[32] Shieh, C. S. Genetic fuzzy control for time-varying delayed uncertain systems with robust stability, Applied Mathematics and Computation 131(2002): 39-58.

[33] Spronck, P.H.M. and E.J.H. Kerckhoffs (1997). Using Genetic Algorithms to Design Neural Reinforcement Controllers for Simulated Plants. Proceedings of the 11th European Simulation Conference (ESM '97) (eds. A. Kaylan and A. Lehmann): 292-299.

[34] Spronck, P, I. Sprinkhuizen-Kuyper, et al. Evolutionary Learning of a Box-pushing Controller. Computational Intelligence in Control (eds. Masoud Mohammadian, Ruhul Amin Sarker and Xin Yao), pp. 104-121. Idea Group Publishing. 2003

[35] X. Yao. Evoling artifical neural networks, Proc. of the IEEE, 87(9): 1423-1447,1999

[36] K. O. Stanley, R. Miikkulainen. Efficient Evolution of Neural Network Topologies, IEEE. Proc. of the 2002 Congress on Evolutionary Computation , Hawaii

[37] Seng, T.L., M. Khalid. Tuning of a neuro-fuzzy controller by genetic algorithms with an application to a coupled-tank liquid-level control system, Engineering Applications of Artificial Intelligence 11 (1998): 517-529.

[38] Seng, T.L., M. Khalid and R, Yusof. Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm, IEEE Trans. on SMC. 1999

[39] Kristinsson, K. and Guy A. Dumont. System Identification and Control Using GAs, IEEE Trans. on SMC. 22(5): 1033-1047,1992

[40] Reeves C R, Dai P. A Hybird GA for System Identification. Proc. of the Int. Symposia on Intelligent Industrial Automation and Soft Computing: B278-283, 1996

[41] Luh, G. C., C. Y. Wu. NonLinear System Identification using GAs, Proc. of the I MECH E Part I. J. of Systems & Control Engineerings, Vol. 213: 105-118, 1999

[42] Billings, S. A., Mao, K. Z. Structure detection for nonlinear rational models using genetic algorithms, Int. Journal of Systems Science, 29(3): 223-231,1998

[43] Fogarty, T.C. An incremental genetic algorithm for real time optimisation, Proc. of the 1989 IEEE Conf. on SMC: 321-326,1989

[44] Linkens, D.A., H.O. Nyongesa. GAs for fuzzy control Part2: Online system development and application, IEE Proc.- Control Theory Appl. 142(9): 177-185, 1995

[45] Lennon, W. K., Passino, K. M. Intelligent Control for Brake Systems, IEEE Trans. on Control Systems Technology 7(2), 1999

[46] Ahmad M, Zhang, L, and Readle J C, On-line genetic algorithm tuning of a PI controller for a heating system, GALESIA'97 – Genetic Algorithms in Engineering Systems: Innovations and Applications: 510-515, 1997

[47] Lin F. J., Chou W. D. An induction motor servo drive using sliding-mode controller with genetic algorithm. Electric Power Systems Research 64 (2): 93 – 108, 2003