ISSN 1746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 1, No. 2, 2005, *pp*. 75-80

Nonstandard optimal control by utilizing genetic algorithms

Jianmin He +, Xiaoping Hu and Hongsheng Lue

School of Economics and Management, Southeast University, Nanjing, 210096, P.R. China

(*Received September 8 2005, accepted October20 2005*)

Abstract. Genetic Algorithms (GAs) have been successful in global optimization problems, optimal control, pattern recognition, resource allocation and others. The use of GAs was introduced for the optimal control of discrete time system. After transforming optimal control problem into unconstrained optimization one with respect to control variables, we utilized GAs to solve this optimization problem, and obtained optimal control strategies. Finally, the optimal trajectory was gained according to state transition equations. Simulation example was illustrated for tracking problem. The result shows that GAs can solve N-stage optimal control problem well, and the advantages of using this method are the following: the needed computation resource is not very sensitive to the size of N; functions in the problem needn't to be differentiable; elements of admissible control set can be continuous, discrete, integer, and mixed.

Keywords: optimal control, genetic algorithm, optimization

1. Introduction

Most of the applications of artificial intelligence (AI) methods to control-system design are formulated either as a parameter optimization task or control structure learning task.

Among the many model-based approaches, the N-stage optimal control problem has been studied in combination with several structures for the neural controller. For example, Nguyen and Widrow (1990) applied backpropagation to the problem of backing up a trailer-truck based on a neural network emulator for the plant. In Saerens (1993), a full static state feedback controller was studied in the context of optimal control, thereby relating the Lagrange multiplier sequence to the backpropagation algorithm. Parisini and Zoppoli (1994) assumed a linear structure preserving principle for the state-tracking problem with application to control of a space robot. In Suykens (1996), the problem of swinging up of an inverted pendulum and double inverted pendulum with stabilization at the endpoint was formulated as a parametric optimization problem in the unknown weights of feedforward or recurrent neural controllers. Suykens (2001) discussed N-stage optimal control by least squares support vector machines (LS-SVM).The problem is formulated in such a way that it incorporates the N-stage optimal control problem as well as a least squares support vector machine approach for mapping the state space into the action space. In the optimal control method by LS-SVM's, the N-stage optimal control problem and the optimization problem related to the LS-SVM controller are incorporated within one problem formulation. A main difference between standard neural network approaches (Bishop (2002) MLP, RBF) and LS-SVM control is that in the former one solves a parametric optimization problem in the unknown interconnection weights while in the latter also the state vector sequence is part of the unknown parameter vector in the optimization problem. However, standard methodologies suffer from problems like the choice of the number of hidden units needed in order to accomplish a given control task. More specifically, in the case of RBF networks one has a curse of dimensionality when one defines a regular grid for the centers (hidden units) in state space. In the LS-SVM control case, the centers will follow from the optimal trajectory that one seeks. In the aforementioned literatures, their common methods lie in the fact that the state space is explicitly mapped into the action space by AI, and original control problem becomes an unconstrained optimization one with respect to state variables. This optimization problem is high-dimension nonlinear one related to N-stage.

The GAs, which is studied within the realm of AI, has been utilized for optimal control by Kundu (1996).

+ Corresponding author. Tel.:86-13914756422.

E-mail address: nj.jian@public1.ptt.js.cn , hxpnj@163.com and mylucklv@163.com .

In this literature, continuous-time optimal control problem was researched by using maximum principle to gain optimal necessary condition. Then, the original object function was modified according to this condition, and the modified object function was optimized by GAs.

In this paper, we discuss nonstandard N-stage optimal control of discrete time system by GAs. In general, optimal control problem can be solved by using maximum principle or dynamic programming method. However, the object function must be separable and all relevant functions should also be differential for utilizing those methods. Nevertheless, not all conditions of separability and differentiability may be satisfied in practice. We refer to the optimal control problem not satisfying both separability and differentiability as the so-called nonstandard optimal control problem. Considering that the final goal is to achieve the optimal strategy and optimal trajectory of optimal control problem, we needn't explicitly give map from state variables to control variables, and solve the problem only in the action space, which differs from the mentioned method done in the state space. After transforming optimal control problem into unconstrained optimization one with respect to control variables, we make use of GAs to solve the optimization problem, and achieve optimal control strategies. At last, the optimal trajectory is obtained based on state transition equations. Simulation example is illustrated for tracking problem.

This paper is organized as follows. In Section 2 we review GAs. Section 3 discusses optimal control by GAs. Simulation example for tracking problem is present in Section 4, followed by conclusions in section 5.

2. Genetic Algorithms

GAs are a potent instrument for optimization and have been applied with success over a broad range of fields. In reality, it is not proper to speak of a single technique but of a family of methods inspired by a single basic idea. The GAs, developed by Holland in the 1960s and 1970s, are based on the fundamental concept of Darwin-type natural evolution. Genetic selection and reproduction are the two fundamental processes which underpin the evolution of the species and its adaptation to the outside world. Selection identifies which elements of a population survive to reproduce, a fact which leads to the recombination of genes. A third process, genetic mutation, introduces further changes which intervene more rarely on genes. Hence, the mechanism of reproduction with the recombination of genes determines a faster evolutionary process than mutation by itself.

The gene selection process is based on the principle of the individual's adequacy to the needs imposed by the outside world: it is not the strongest or the tallest or the fastest who survives but the fittest overall. Nevertheless, implicit in the concept of evolution is the idea of the progressive improvement of the species with respect to previous generations. Between two successive generations, there are small changes, but the favorable ones cumulate themselves so that, through many generations, they produce big changes.

GAs reiterate the principles of natural evolution in a stylised way, applied to a problem, they do not work on one particular solution but on populations of solutions, which evolve until they converge at levels regarded as optimal. Evolution takes place with the selection of the best individuals of a population and their reproduction with recombination of their genes. The objective of GAs is to generate successive populations of solutions whose roots (genes) are selected from the most promising (highest accurate) solutions of the previous population. GAs combine random search procedures with highly effective exploration techniques.

The fundamental operators of GAs are:

(a) the selection and reproduction of better individuals;

(b) genetic recombination (crossovers);

(c) the random mutation of individuals' genes.

Selection is a fundamental operation which is performed with the aid of a function for the evaluation of the fitness of individuals. The straightforward duplication of selected individuals, reproducing them identical to those of the previous population, entails no benefit in terms of the exploration of solution spaces. In the course of reproduction, the operation of genetic recombination (crossover) is thus introduced: the genes of two individuals selected for reproduction are combined to cause the population to evolve and to permit the exploration of new portions of space of the possible solutions (the exploration sets out from the most promising points that have already been identified).

The recombination of genes can be accompanied by the operation of mutation, which has a very low probability, so as not to destroy entirely the genetic heritage accumulated through previous selections. Performed in this way, mutation makes it possible to enrich the variety of individuals present in the population, preventing them from tending to be too uniform, hence losing the capacity to evolve. In setting up a GA procedure, there are two essential points: the genetic representation of solutions to the problem and the definition of the evaluation function (fitness). The evaluation function is a way to measure the individual performances in the population. The fitness of each individual is merely its classification rate's performance.

Since GAs operates on symbolic strings, the solutions we intend to find have to be represented by a code that can be manipulated by the algorithm. This genetic coding may take on a variety of forms: in general, the binary alphabet, made up of sequences of 0 and 1 is used, but there are cases in which coding with other symbols facilitates the use of genetic strings.

The fitness function is designed to evaluate the performance of the individuals (ex: functions) who make up the populations, transforming the fitness of the solutions proposed by the GAs into numeric values based on their performance.

The typical steps into which a GA procedure proceeds are:

(a) the initial population of individuals (or genomes) is generated randomly (by random numbers, binary strings of zeroes ad ones are generated, all with the same length);

(b) for every individual, fitness is calculated in relation to the problem that has to be solved (i.e., the goodness of the hypothetical solutions);

(c) the degree of homogeneity of the entire population's fitness is calculated (bias);

(d) individuals are ordered on the basis of their fitness and those suitable to generate the subsequent population are selected;

(e) the successive population is generated on the basis of reproduction of new individuals, starting from the ones selected in the previous population;

(f) in the new population, the sequence is repeated from point (b) onwards.

The procedures (b)-(f) are repeated until the actual population converges towards increasingly homogeneous individuals; in the extreme hypothesis of the individuals all being equal, we reach a stable situation in which there is no further evolution by crossovers, and the only possibilities for new exploration are limited to rare genetic modifications. In general, in order to stop the process, we are satisfied with achieving a certain level of uniformity (or bias).

The analysis of the population with genetic operators allows us to obtain a larger amount of information than would be possible with a straightforward examination of individuals taken one by one (collective analysis adds information to independent individual analyses). The significant advantage of Gas involves the exploration of a high number of portions of the solution space with great efficiency, concentrating samples in the most promising regions to develop knowledge about them. A population's similarity in terms of string is an important item of information for the continuation of the search strategy with future populations.

3. N-stage optimal control problem by using GAs

In the N-stage optimal control problem one aims at solving the following problem

$$\min F_N\left(x_k,u_k\right)=G\left(x_{N+1}\right)+C\left(x_1,\cdots,x_N,u_1,\cdots,u_N\right) \tag{1}$$

subject to the system dynamics

$$x_{k+1}=f\left(x_k,u_k\right),u_k\in A,k=1,\cdots,N\left(x_1\ given\right) \tag{2}$$

where $G(\square)$ is positive definite function. $C(\square)$ may be separable, for example:

$$\min F_N\left(x_k,u_k\right)=G\left(x_{N+1}\right)+C\left(x_1,\cdots,x_N,u_1,\cdots,u_N\right)=G\left(x_{N+1}\right)+\sum_{k=1}^{N}h_k\left(x_k,u_k\right) \tag{3}$$

in which h_k is positive definite function. Atypical choice is the quadratic cost function

$$h_k\left(x_k,u_k\right)=x_k^TQ_kx_k+u_k^TR_ku_k\ ,G\left(x_{N+1}\right)=x_{N+1}^TQ_{N+1}x_{N+1}$$

where Q_k,R_k is semi-positive definite matrix. $x_k\in R^n$ denotes the state vector, $u_k\in A\subset R^m$ where A is admissible control set.

Denotes the control strategy from 1 to N stage by the vector $u=\left(u_1,\cdots,u_N\right)$, and let vector $x=(x_1,\cdots,x_{N+1})$ be state variables. Known u_1 and x_1, x_2 can be gained by using(2), the rest may be deduced by analogy iterative method, x_k (where $k=2,\cdots,N+1$) is achieved for giving control variables u_{k-1} at the $k-1$ stage. In other words, state vector x can be expressed by the function of control vector u.

$$x_2=f(x_1,u_1)$$
$$x_3=f(x_2,u_2)=f(f(x_1,u_1),u_2)$$
$$\cdots$$
$$x_{N+1}=\underbrace{f(f(\cdots f}_{N}(x_1,u_1)\cdots,u_{N-1}),u_N)$$

For notion simplification, let $x_k=f_k(u,x_1)$. So (1) becomes the optimization problem about u:

$$\min F_N(u)=G(f_{N+1})+C(f_1,\cdots,f_N,u_1,\cdots,u_N) \tag{4}$$

Let $u^*\in\arg\min F_N(u)$, in terms of (2), x^* can be achieved. For (x^*,u^*) satisfy(2) and make (1) be minimum at the point (x^*,u^*), the pair (x^*,u^*) is respectively the optimal trajectory and optimal control strategy of original problem described by (1) and (2).

In view of control variables $u_k\in A\subset R^m$, let

$$u_k=(u_{k,1},\cdots,u_{k,m})^T$$

then u is a $m\times N$ matrix. For tackling convenience of utilizing GAs to solve (4), it is necessary to sort order of elements in matrix u as follows:

$$u'=(u_1^T,\cdots,u_k^T,\cdots,u_N^T)=(u_{1,1},\cdots,u_{1,m},\cdots,u_{k,1},\cdots,u_{k,m},\cdots,u_{N,1},\cdots,u_{N,m})$$

then, (4)can be rewritten as

$$\min F_N(u')=G(f_{N+1})+C(f_1,\cdots,f_N,u_1,\cdots,u_N) \tag{5}$$

Accordingly, (5)can be optimized by using the float-encode GAs.

Denote

$$u'^*\in\arg\min_{u'}F_N(u')$$

and let

$$u_k^*=(u'^*_{1+(k-1)m},\cdots,u'^*_{km})^T$$

According to (2), we have

$$x^*_{k+1}=f(x^*_k,u^*_k),x_1^*=x_1.$$

If one is interested in obtaining the explicit optimal control law, parametric or non-parametric methods can be used to fit the optimal control law by utilizing the data (x^*,u^*).

4. Simulation example

Here we illustrate GAs optimal control method of section 3 on an example reported in Suykens (2001). Given the system dynamic equations

$$\begin{cases} x_{1,k+1}=0.1x_{1,k}+2\dfrac{u_k+x_{2,k}}{1+(u_k+x_{2,k})^2} \\ x_{2,k+1}=0.1x_{2,k}+u_k\left(2+\dfrac{u_k^2}{1+x_{1,k}^2+x_{2,k}^2}\right) \end{cases} \tag{6}$$

we consider a state vector tracking problem with (3)-type cost function as follows

$$h_k(x_k, u_k) = (x_k - x_k^r)^T Q_k (x_k - x_k^r) + u_k^T R_k u_k$$
$$G(x_{N+1}) = (x_{N+1} - x_{N+1}^r)^T Q_{N+1} (x_{N+1} - x_{N+1}^r)$$

where

$$Q_k = Q = diag\{1, 0.001\}, k = 1, \cdots, N+1; R_i = R = 1, i = 1, \cdots, N$$

and x_k^r is the reference trajectory to be tracked. From the elements of Q, we aim at tracking the first element of state variable. Let

$$x_k^r = \left(\sin\left(2\pi k/N\right), \cos\left(2\pi k/N\right)\right)^T$$

with $k = 1, \cdots, N$ and $N = 60$. The given initial state is $x_1 = (0, 0)^T$. For illustrating the excellent advantages of solving the mentioned nonstandard optimal control problem, let admissible control set be continuous and discrete (in this simulation, we use integer set as admissible control set) respectively.

We make use of the GAs toolbox in Matlab 7.0. The primary parameters in GAs are described in table1, and other parameters are defaults.

Table1. primary parameters in GAs

	continuous admissible control set	discrete admissible control set
PopulationType	doubleVector	bitstring
PopulationSize	1000	100
Generations	10000	10000
StallGenLimit	1000	1000
StallTimeLimit	60	60
HybridFcn	fminsearch	patternsearch

Simulation results are shown for the method described in section 3. Fig.1 illustrates the case of continuous admissible control set, where the left panel shows the simulation results for the first variable of the optimal trajectory and reference trajectory, and the right one shows the simulation results for the optimal control policy; while Fig.2 illustrate the case of discrete admissible control set.

In the same way, the GAs shows a good generalization performance by tackling the tracking problem with respect to other initial states. In the case of continuous admissible control set, optimal trajectory track reference trajectory better than in the case of continuous admissible control set.

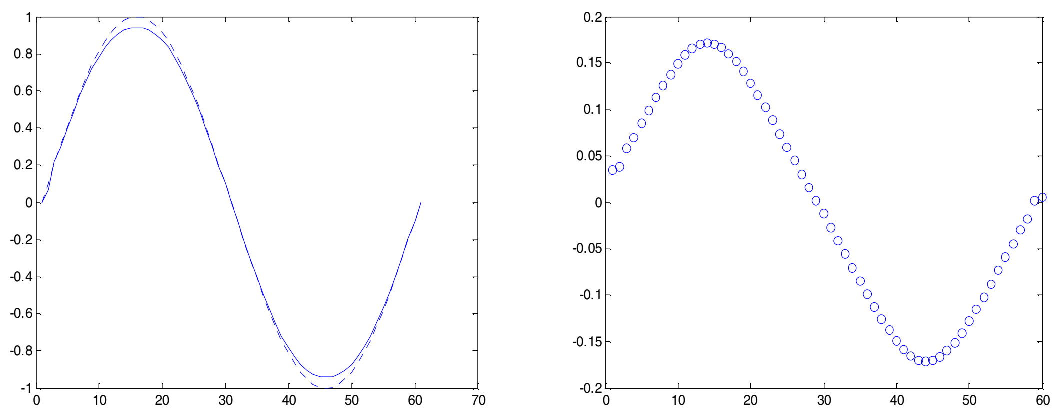

Fig.1 optimal trajectory (full line), reference trajectory(dashed line) and optimal policy in the continuous case

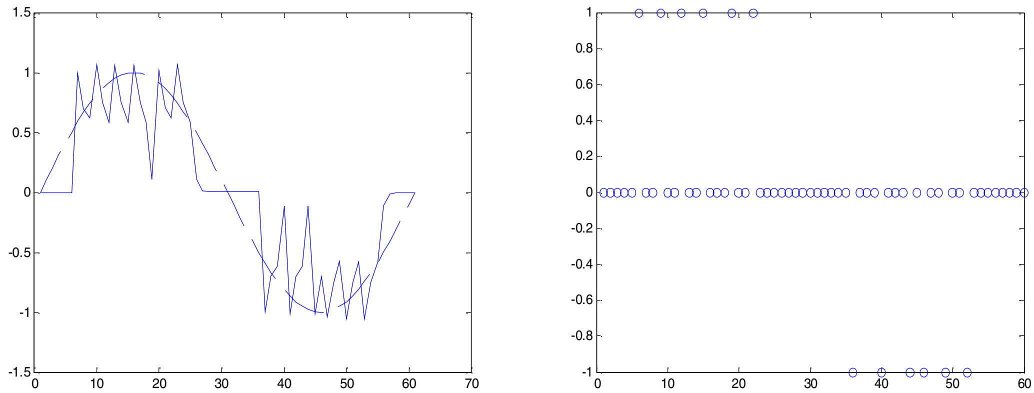

Fig.2 optimal trajectory (full line), reference trajectory (dashed line) and optimal policy in the discrete case

5. Conclusions

In the paper, we introduced the use of GAs for solving optimal control problems. The N-stage optimal control problem has been formulated for a nonlinear optimization one with respect to control variables. This optimization problem has been solved by the method of GAs. GAs have a distinguishing feature that make them very attractive when considering nonlinear optimization: they can be used to solve all classes of nonlinear programming models, i.e. continuous, integer, discrete, and mixed models. They can be especially effective when solving highly nonlinear models. On the other hand they can also be used to provide quick, near global-optimal solution. Since the final goal is to achieve the optimal strategy and optimal trajectory, we needn't explicitly give map from state variables to control variables, and solve the problem only in the action space, such differs from the proposed method working in the state space. Simulation example illustrated for tracking control problem shows that GAs may solve N-stage optimal control problem well. The advantages of using GAs are the following: the needed computation resource is not very sensitive to the size of N; functions in the problem needn't to be differentiable; elements of admissible control set can be continuous, integer, discrete, and mixed.

6. References

[1] D. Nguyen, B. Widrow, *Neural networks for self-learning control system*, IEEE Control System Magazine, 10(1990) No. 3, pp. 28-23.

[2] M. Saerens, et al, *Neural controllers based on back propagation algorithm*, In M.M. Gupta & N.K. Sinha, IEEE Press book on intelligent control: theory and practice, New York: IEEE Press, 1993.

[3] T. Parisini, R. Zoppoli, *Neural networks for feedback feed forward nonlinear control systems*, IEEE Transactions on Neural Networks, 5(1994)3, pp. 436-449.

[4] J. A. K.suykens, et al, *Artificial neural networks for modeling and control of non-linear systems*, Boston: Kluwer Academic, 1996.

[5] J. A. K.suykens, et al, *Optimal control by least squares support vector machines*, Neutral Networks, (2001)14, pp. 23-35.

[6] C.M. Bishop, *Neural networks for pattern recognition*, Oxford: Oxford University Press, 2002.

[7] S. Kundu, S. Kawata, *Genetic algorithms for optimal feedback control design*, Engineering application of Artificial Intelligence, l9(1996)4, pp 403-411.