Paper:

Genetic Algorithm On Line Controller for the Flexible Inverted Pendulum Problem

Elmer P. Dadios*, **Patrick S. Fernandez****, **and David J. Williams*****

*Department of Manufacturing Engineering and Management
2401 Taft Avenue, De La Salle University, Manila 1004, Philippines
E-mail: dadiose@dlsu.edu.ph
**National Power Corporation
Diliman, Quezon City 1004, Philippines
***Loughborough University
Loughborough Leicestershire, LE11 3TU, U.K.
E-mail: D.J.Williams@lboro.ac.uk

[Received January 8, 2005; accepted August 25, 2005]

This paper presents a real time controller for a highly non-linear system. The Flexible Pole-Cart Balancing Problem (FPCBP) is used as the test case to investigate the learning capability of Genetic Algorithm (GA) in physical application. The controller developed is initially trained using a set of data taken from on line dynamics of the flexible pole cart balancing system. Based from the physical data of the system, the weights W1 to W6 are optimized by the genetic algorithm in order to determine the correct value of the force applied to the cart. The trained GA-based controller then controls the physical Flexible Pole-Cart Balancing system for infinite time. Analysis on the behavior of the GA model developed is presented. Results of the physical experiments show that the controller developed is accurate, adaptive and robust.

Keywords: Genetic Algorithm (GA), learning controllers, flexible inverted pendulum problem

1. Introduction

Genetic Algorithm (GA) is a biologically inspired class of algorithms based loosely on Darwinian principles of biological evolution [1–6]. This was proposed by John Holland in 1975 as general-purpose optimization methods and has been successfully applied to search, optimization and machine learning tasks [7]. In the area of non-linear optimization, Genetic Algorithm had shown greater potential especially in cases wherein the optimization space is large and the data is randomly diverse [9–13]. The standard model of the GA information leading to the problem approximate solution is represented by string length called chromosomes [8].

This research investigates the learning behavior of Genetic Algorithm and its potentials in real time non-linear control. Analysis on the effects of various parameters used in the GA for its learning process is discussed. On line controller to solve the flexible pole-cart balancing problem is used as benchmark for the proposed system. The gene of each chromosome contains the parameters that have to be optimized in order to control the flexible pole cart balancing problem with the following information: displacement of the cart, velocity of the cart, angle of the pole, angular velocity of the pole, deflection of the pole, deflection velocity of the pole, and the applied forces to the cart. A full description of the dynamics of the flexible pole-cart balancing problem can be found in [14, 15].

2. Genetic Algorithm Learning Process for the Flexible Pole Cart Balancing Problem

As has been shown in the equations presented in [15], the Flexible Pole Cart Balancing Problem (FPCBP) is an advanced version of the inverted pendulum problem. The effect of additional cart status variables (D = pole deflection and Dd = pole deflection velocity) makes the system highly nonlinear and very unstable. Based from the GA linearization capability, the FPCBP can be represented by a linear equation taken into consideration the effect of pole and cart dynamic parameters given below.

$$Fc = W1*X+W2*Xd+W3*A+W4*Ad+W5*D+W6*Dd \quad (1)$$

where: X = cart position
Xd = cart velocity
A = pole angle
Ad = pole angular velocity
D = pole deflection
Dd = pole deflection velocity
Fc = computed applied force
W1 to W6 = weight factors for each FPCBP variables

The values of weight factors W1 to W6 are the one to be optimized by the GA learning process using the information acquired from the physical flexible pole cart balancing system. In order to attain fast, accurate and efficient optimization, several GA tuning parameters are

Table 1. Sample lists of training data.

Set	X (cm)	Xd (cm/s)	A (deg)	Ad (deg/s)	D (cm)	Dd (cm/s)	Fa (volts)
G1	15.651	3.461	2.044	−29.353	−4.105	−3.16	3.35
2	16.917	13.638	−1.536	−49.114	−4.143	−0.762	−1.65
3	17.006	8.155	1.686	5.251	−3.938	1.884	9.75
…	…	…	…	…	…	…	…
1100	8.675	−7.908	2.51	12.258	−2.189	−12.369	7.15

taken into consideration. These include 1) Population size, 2) Crossover rate, 3) Mutation rate, and 4) Elitism. These parameters were investigated and analyzed until a good combination is attained (see section 3).

Though the linear solution presented above seems simple, determining the values of the parameters W1 to W6 is very difficult task. We can investigate the equations and arbitrarily assign values to these parameters taking into consideration their logical relationship with the applied force. However, with six values to manipulate, this will take huge amount of time.

The GA-based learning process for determining the parameter values W1 to W6 is shown in **Fig.1**. This is essentially a search space that uses eq.(1) as the Objective Function (OF) and eq.(2) as the Fitness Function (FF) which is equivalent to NF below.

$$\mathrm{NF} = \mathrm{MF} - \sum_{x=1}^{n} \left(\mathrm{abs}(\mathrm{Fa}_x - \mathrm{Fc}_x)\right) \quad . \ . \ . \ . \ . \ . \ . \ . \quad (2)$$

where: Fa = actual applied (or training) force
n = number of training data
MF = maximum fitness
NF = net fitness

The MF is set to 1000000 or 1E6 based on the maximum cumulative error for 1100 training data.

For each chromosome in the population, the value of the force is evaluated in every training data set using eq.(1). The resulting force value is compared with the force value on the training data shown in **Table 1** and the fitness is evaluated using eq.(2). Once the highest possible value of the fitness function is attained, the values of W1 to W6 are taken for the use of control program.

3. Analysis of Genetic Algorithm Learning Process

In this research two chromosome representations were investigated namely: binary and real number representations. The binary string chromosome representation gave no promising results because the conversion process (binary to exact numeric physical values and vice versa) takes so much time and round off errors are propagated. Also, besides being inaccurate and non-convergent, the program can only handle a small amount of training data. The maximum fitness was set to 1E6 based on the maximum total error for 1100 data. The fine tuning parameters (population size, crossover rate, mutation rate, and elitism) were investigated in each run to find out which

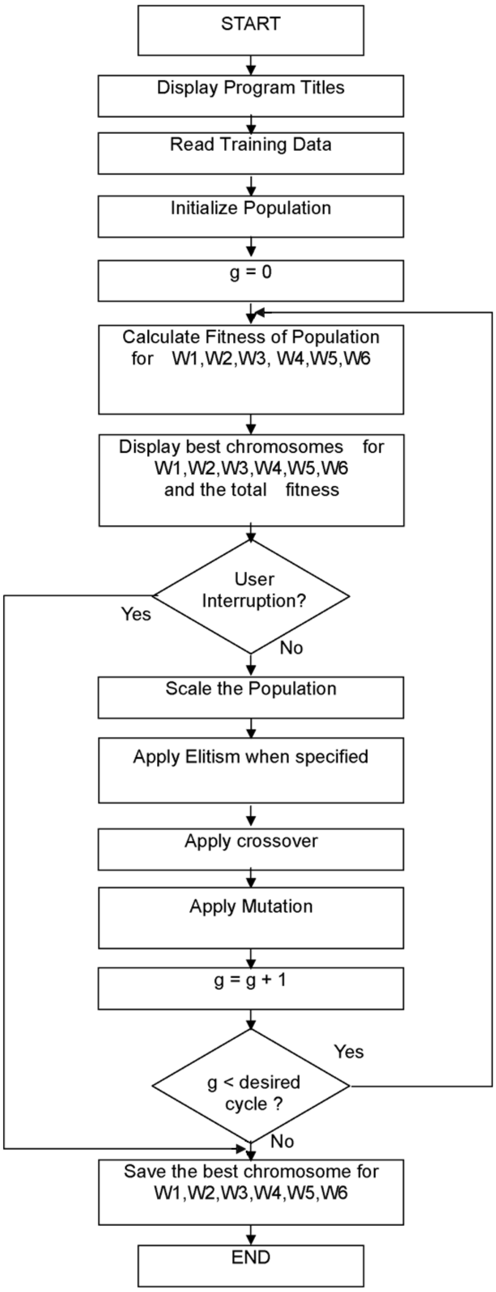

Fig. 1. Flowchart of the GA learning process for FPCPB.

combination yields the fastest and most accurate optimization. The resulting values for W1 to W6 are based on the best results obtained from these experiments.

3.1. Population Size

Different runs were conducted to investigate the effect of the population size. It is observed in **Fig.2** that the convergence is fast and smooth as the population size increases. For the population of 50, the fitness abruptly increases from 98% to 99.75% within 3 generations. This increase becomes less abrupt for population = 100 which increases from 98% to 99.75% within 7 generations. The convergence curve is even smoother for population = 150,