Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

17$^{\text{th}}$ International Conference on Process Control 2009

Hotel Baník, Štrbské Pleso, Slovakia, June 9 – 12, 2009

ISBN 978-80-227-3081-5

`http://www.kirp.chtf.stuba.sk/pc09`

Editors: M. Fikar and M. Kvasnica

Sekaj, I.: The Use of Matlab Parallel Computing Toolbox for Genetic Algorithm-Based MIMO Controller Design, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 17th International Conference on Process Control '09*, Štrbské Pleso, Slovakia, 277–280, 2009.

Full paper online: `http://www.kirp.chtf.stuba.sk/pc09/data/abstracts/098.html`

THE USE OF MATLAB PARALLEL COMPUTING TOOLBOX FOR GENETIC ALGORITHM-BASED MIMO CONTROLLER DESIGN

S. Kajan*, I. Sekaj, M. Oravec*****

Institute of Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava
*e-mail : * slavomir.kajan@stuba.sk,** ivan.sekaj@stuba.sk,*
**** michal.oravec@stuba.sk*

Abstract: The paper describes the use of Matlab's Parallel Computing Toolbox for parallel genetic algorithm-based design of a MIMO controller. Parallel genetic algorithms (PGA) represents a stochastic optimization approach which is computed in more co-operating and interconnected computation nodes in a parallel mode. Each node of the PGA can be located on the same processor, on more processors or on more computers respectively. This approach is able to solve very complex search/optimization/design tasks, but on the other hand, it requires a high computation power. From that reason the distribution of computation complexity to more processors under Matlab environment is presented.

Keywords: Matlab Parallel Computing toolbox, parallel genetic algorithms, MIMO controller design

1 INTRODUCTION

Genetic algorithm (GA) is a powerful stochastic optimization approach, which mimics the evolution in the living nature with the aim to solve practical problems. It is from its basic a parallel algorithm, because it is searching the solution concurrently in a group of potential solutions (a population). In case of solving complex design problems it sometimes happen, that the current solution stagnates in a local optimum (suboptimal solution). Furthermore the computation times of solving complex problems are long. Such drawbacks can be eliminated using a next level of parallelism in - parallel GA (PGA). Such a parallelism is able to increase the information diversity in the population and to redirect the solution process to the global optimum. The drawback of the required high computational effort of PGA's can be eliminated by distribution of the algorithm to more processors/computers. In Matlab for that reason the Parallel Computing toolbox may be used.

2 THE PRINCIPLE OF PGA

The basic principle of the PGA is the parallel run of more GA's, which are performed on isolated subpopulations - nodes (Cantú-Paz, E., 1995, Sekaj, I., 2004). Each subpopulation is handled using its own GA. These subpopulations in defined time periods (number of generations) or in some asynchronous events exchange information between some nodes according defined interconnections. That is, some individuals from some subpopulations are copied into other subpopulations. This process is called migration. The migration management is defined in the migration architecture and migration time schedule. Various migration architectures are depicted in Fig.1. Each circle represents a node with its own GA.

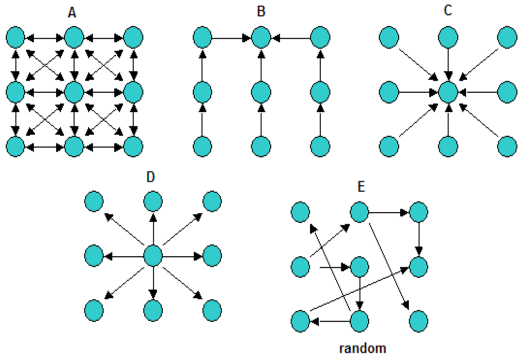

Fig. 1. Various migration architectures of PGA

3 MIMO CONTROLLER PARAMETER DESIGN USING PGA

Consider a MIMO controller design, which is depicted in fig. 2. The control scheme consists from the controlled system with two inputs and two outputs, two PID controllers and two correctors. Let us consider the following performance index

$$J = \sum_{k=1}^{N} \left(|e_{1k}| + |e_{2k}| + \alpha |dy_{1k}| + \alpha |dy_{2k}| \right) \qquad (1)$$

where e_1, e_2 are control errors, dy_1, dy_2 are derivatives of the controlled variables and α is a weight constant. The use of output derivatives can damp the output oscillations.

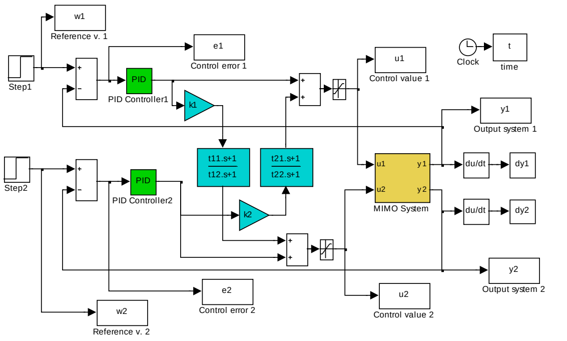

Fig. 2. The MIMO control loop simulation scheme in Matlab/Simulink

The aim of the design is the search for optimal controller parameters from the defined parameter space, which minimize the performance index (1). The evaluation of the cost function consists of two steps. The first step is the computer simulation of the closed-loop time-response, and the second one is the performance index evaluation.

3.1 *Genetic algorithm*

Genetic algorithms are described in e.g. (Goldberg, D.E., 1989, Man, K.F, 2001, Sekaj, I., 2005) and others. Each chromosome represents a potential solution, which is a linear string of numbers, whose items (genes) represent in our case the designed controller parameters. Because the controller parameters are real-number variables, real-coded chromosomes have been used. The chromosome representation in this case can be in form

$$\mathbf{ch} = \{ P_1, I_1, D_1, P_2, I_2, D_2, k_1, t_{11}, t_{12}, k_2, t_{21}, t_{22} \}$$

where the first 6 genes are parameters of the two PID controllers and the rest genes are the gains and time constants of the two lead-lag feed-forward correctors.

A general scheme of a GA can be described by following steps:

1. Initialization of the population.

2. Evaluation of the cost function for each chromosome (simulation, performance index evaluation).

3. Parent chromosomes selection

4. Crossover and mutation of parents

5. Completion of the new population from the new children and selected members of the old population. Jump to the step 2.

3.2 *The PGA realization*

For the PGA demonstration three subpopulations running on three PC's have been used. They are marked S1-Slave1, S2-Slave2 and M-Master. In the Master node the migration and the computation management is realized. The used migration connections are shown in fig. 3.

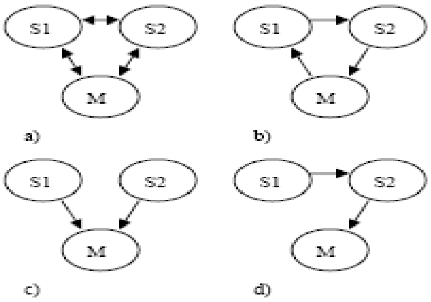

Fig. 3. Migration schemes with tree subpopulations

For the parallel computing implementation the Matlab Parallel Computing Toolbox with the Matlab Distributed Computing Engine (MDCE) are needed. In the fig.4 the block scheme of the computation management is depicted (The Mathworks. 2006).

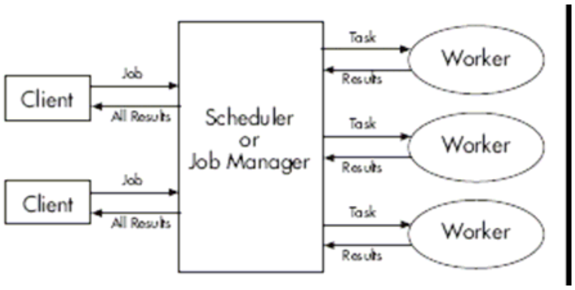

Fig. 4. The block scheme of distributed computing sessions using PCT and MDCE

The parallel computation can be described in following steps:

1. On the client PC required computation tasks are created using PCT.

2. The job manager will provide the task management on particular workstations where the Matlab MDCE must be installed.

3. After finishing of the jobs the manager will collect all responses from workstations and send it to the client.

Based on this the parallel genetic algorithm can be realized as follows:

1. Creation of the function - genetic algorithm, which input is the initial population in form of a real matrix (rows are chromosomes, columns are genes) and the output is the population after N generations of the GA run.

2. On the PC client the initialization of 3 independent populations is performed.

3. The parallel run of the 3 GA's is performed until the required number of generations is computed.

4. Migrations between subpopulations according the defined migration architecture are performed.

5 CASE STUDY

Next the results of MIMO controller design (Fig.2) using the proposed PGA structure are demonstrated. All migration architectures have been tested. In fig.5 graphs of the cost function (1) evolution wtih the migration scheme fig.3c are shown. The migration period was 5 generations.

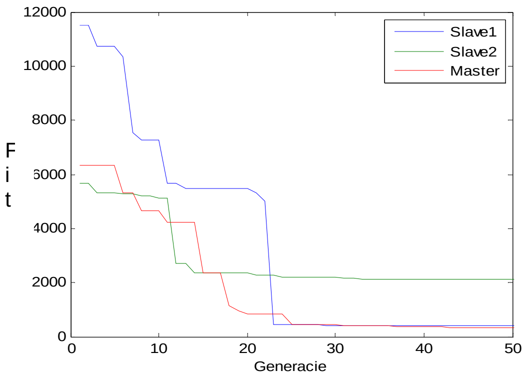

Fig. 5. Cost function evolution during the controller design procedure for all 3 subpopulations.

The final solution is the best individual of the population in the node Master after performing defined number of populations. The obtained results are follows:

P_1=53.899, I_1=78.2064, D_1=1.8854, P_2=24.7023, I_2=19.4204, D_2=17.2952, k_1=-0.064, t_{11}=29.235, t_{12}=89.2842, k_2=-0.1227, t_{21}=19.8595, t_{22}=54.4186

The appropriate time-responses are shown in fig.6 and fig.7.

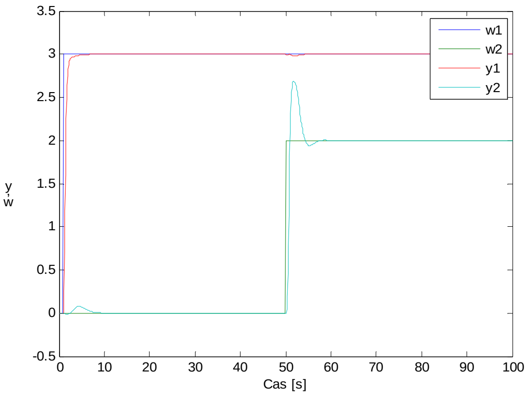

Fig. 6. Time-responses of the reference variables w_1, w_2 and controlled variables y_1, y_2

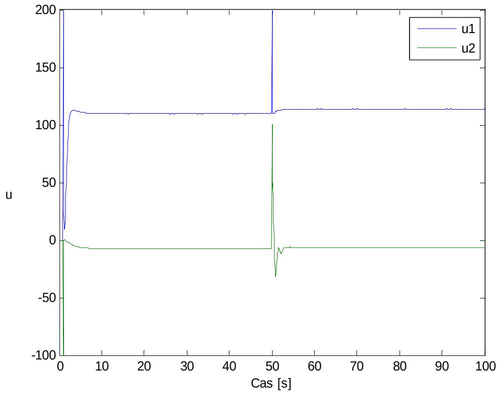

Fig. 7. Time-response of the control values u_1, u_2

6 CONCLUSIONS

The aim of this paper was to demonstrate the application of parallel genetic algorithms running on a more computer configuration (cluster) using Parallel Computing toolbox of Matlab. The solved problem was an optimal parameter design of a MIMO control system. Using this means it is possible to solve complex search/optimization tasks from control engineering area like non-linear controller design, robotics etc., but also from other technical and non-technical application areas. The condition of using this approach is the possibility of the cost function evaluation in each point of the searched space. This requires the existence of a computer model of the solved problem. Using the PGA it is possible to reduce the computation time from hours to minutes or form days to hours respectively. Furthermore the PGA allows eliminate the premature convergence of the solution process to a local optimum (to suboptimal solution).

ACKNOWLEDGMENTS

The work has been supported by the grants agency VEGA no. 1/0544/09 and no. 1/0690/09. This support is very gratefully acknowledged.

7 REFERENCES

Cantú-Paz, E. (1995). A summary of research on parallel genetic algorithms, *IlliGAL Report No. 95007, Illinois Genetic Algorithms Laboratory,* University of Illinois at Urbana Champaign

Cantú-Paz, E. (2001). Migration polices, selection pressure, and parallel evolutionary algorithm*s*, In: Journal of heuristics 7(4), 2001, pp. 311-334.

Dorf, R.C. (1990). Modern Control Systems. Addison-Wesley publishing Company, 5th edition

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addisson-Wesley.

Man, K.F. Tang, K. S. Kwong, S. (2001). Genetic Algorithms, Concepts and Deign. Springer

Sekaj, I. (2003). Genetic Algorithm Based Controller Design", In: *2nd IFAC conference Control System Design'03*, Bratislava

Sekaj, I. (2004). Robust Parallel Genetic Algorithms with Re-Initialization, In: *PPSN VIII*, September 18-22, Birmingham

Sekaj, I. (2005). Evolučné výpočty a ich využitie v praxi. Iris Bratislava

The Mathworks. (2006). *MATLAB Distributed Computing Engine*, System Administrator's Guide, Natick