Controller adaptation using dynamic evolutionary algorithm

Piotr Kaczyński[1] and Paweł D. Domański[2]

[1] Warsaw University of Technology, Institute of Control and Computation Engineering, Warsaw, Poland, email: P.Kaczynski@elka.pw.edu.pl
[2] Warsaw University of Technology, Institute of Control and Computation Engineering, Warsaw, Poland, email: P.Domanski@ia.pw.edu.pl

Abstract. In this paper we present an algorithmic solution to PID controller parameters tuning task. Firstly, the brief test model and a simple example on how to tune PID parameters using genetic algorithm are presented. This functionality is then tested in more complicated situation of a non-stationary linear object. Adaptivity of such a solution is considered along with some results.

1 Introduction

The possibilities of advanced control are of very strong practical importance. The automated update of the controller parameters is the crucial issue. The design and definition of the controller structure requires high level knowledge and experience, while with the structure set up the parameters can be tuned adaptively. The paper presents the results of the application of the genetic algorithm based approach to on-line parameter adaptation. The main objective was to satisfy real time requirements enabling controller real-time operation in the non-stationary conditions.

The presented approach is different from simple PID tuning using genetic algorithm. Firstly, evolutionary optimization algorithm works in time. Its epochs changes together with the sampling period of the controller. It runs as long as the controller operates. Thus it forms an on-line adaptation mechanism for the controller. It is a non-stationary system.

The paper is organized in two parts. First the idea of the application of the evolutionary optimization to the one-shot tuning of the controller is introduced. This is a well known task but it is then followed by on-line extension while the optimization runs in real-time. The concept is presented with an example of the PID controller. But the application to other regulators is straightforward and does not require additional specific modifications.

2 Parameter estimation

The main goal is to design the estimation algorithm, which would automatically tune the controller without user intervention. The extension of the algorithm between different control strategies should also be easy. It appeared that the modified evolutionary strategy can help to solve that problem. The optimized strategies for succession and mutation satisfy the the need for on-line adaptation scheme.

2.1 Controller

The simplest control strategy was chosen for the verification of the proposed approach. It has only three parameters: the gain K, integration time - T_i and derivation time - T_d. Tests with PID controller give the opportunity for many comparative tests and additionally enable easier practical verification due to the large application base. As for process, linear models were used with step changes of their parameters.

2.2 Evolutionary algorithm

The evolutionary algorithm is used as the optimization and adaptation mechanism. Most of the parameters are standard except for succession and mutation which were designed to satisfy specific requirements of the used structure. The characteristics of the algorithm is described in detail below.

Individual coding The proper choice of the coding of the solution domain within the chromosome is the most important factor in the algorithm design. Thus the genes of the chromosome reflects the values of the parameters being optimized. In the considered example it is quite natural to have three genes in chromosomes, each one being responsible for one controller parameter. Thus each individual is in the form of the three element vector

$$X = [K\ T_d\ T_i]^T \tag{1}$$

The chromosome variables are being modified to maximize the selected cost function. All the variables are constrained as below

$$\begin{array}{ccccc} K^{\min} & \leq & K & \leq & K^{\max} \\ T_d^{\min} & \leq & T_d & \leq & T_d^{\max} \\ T_i^{\min} & \leq & T_i & \leq & T_i^{\max} \end{array} \tag{2}$$

where $T_i^{\min} > 0$. These constraints are the only algorithm parameters that need to be defined by the user. Only rough estimate of the information about the process is thus required.

Cost function Taking into consideration that the process is described by some transfer function the performance index is defined as an error between simulated output of the controller-process control loop. For sake of the realism the additional factor limiting the variations of the control signal is applied. For the SISO systems we have the following criterion:

$$F(X) = -\left(\int_0^{t_h} (y^z(t) - y(t,X))^2 dt\ + 0.1 \int_0^{t_h} u^2(t,X) dt \right)$$

where $y^z(t)$ is the desired process output (setpoint), $y(t,X)$ is the real process output and $u(t,X)$ is the control signal. The overall functional depends on the chromosome defined in (1). The process is simulated over some horizon t_h (in our case 20 sec.) with the setpoint defined by the sinusoidal function

$$y^z(t) = \sin(t) \tag{3}$$

Due to the evolutionary algorithms convention the optimization task is to maximize, and thus there is a negative sign in the performance index. The evolutionary algorithm brings one additional advantage. We can consider non convex cost functions with non-linear constraints. It offers more flexibility in the algorithm design phase.

Genetic operators The decision about the genetic operators is needed to allow proper and fast searching capabilities of the optimization algorithm.

As the crossover, simple averaging algorithm is used. The two individual crossover is applied. The genotype of the resulting individual is defined as

$$\overline{X} = \xi X_1 + (1 - \xi) X_2 \tag{4}$$

where ξ is the stochastic variable with the flat distribution over the $[0, 1]$ domain. The resulting individual is always inside the feasible region because this region is convex. The crossover is performed with the probability p_c (in our case set to 0.7).

The mutation is the second genetic operator. It plays a very important role in the presented algorithm. It is responsible for the adaptation mechanism. If the mutation is too 'small' it might cause the grouping of the solutions close to one of the maximum, disabling further searching. Simultaneously it can slow down the full process, which is forbidden - the algorithm has to operate in real time. Its value must somehow reflects the adaptation speed.

The mutation strategy is quite simple. The new chromosome is formed as the sum of the actual chromosome and a random value with normal distribution, exponentially decaying with the epoch number, as in the formula below

$$\begin{bmatrix} \overline{K} \\ \overline{T_d} \\ \overline{T_i} \end{bmatrix} = \begin{bmatrix} K \\ T_d \\ T_i \end{bmatrix} + \begin{bmatrix} \sigma_m^K \zeta_1 \\ \sigma_m^{T_d} \zeta_2 \\ \sigma_m^{T_i} \zeta_3 \end{bmatrix} \cdot 0.99^{epoch} \tag{5}$$

Each of the genes has different standard deviation σ_m, while ζ_i denotes standard stochastic variables with variance 1 and expected value 0. It should be noted that standard deviations play the most important role being responsible for the adaptation. If mutation moves the individual outside of the feasible region, it is moved to the nearest point lying on the border of the feasible region.

Succession and reproduction Succession and reproduction plays also important role in the behavior of the algorithm. They have impact on the speed of the variance decrease of the selected genes in population, due to larger or smaller selectivity.

As the main succession strategy the full method has been chosen. All new individuals appeared after crossover and reproduction are accepted to the new population. Tournament reproduction is used, so to the new population comes an individual with the highest performance out of q individuals chosen with uniform probability. The parameter q enables selection sensitivity to be controlled.

2.3 Results

The algorithm described above was used to check if the real-time adaptation of the PID controller could be achieved. Two exemplary processes were tested. The first one is relatively slow but stable, while the second one is faster and unstable.

$$G_1(s) = \frac{1.68}{9.78s^2 + 7.17s + 1} \tag{6}$$

$$G_2(s) = \frac{s^2 + 8s + 7}{s^3 + 11s^2 - 62s - 720} \tag{7}$$

Several simulations were performed to find the best parameter setup for the desired goal of PID controller on-line tuning.

For the first process the tournament size was set to $q = 2$. The cubic constraints are as below:

$$\begin{bmatrix} K^{\min} & K^{\max} \\ T_d^{\min} & T_d^{\max} \\ T_i^{\min} & T_i^{\max} \end{bmatrix} = \begin{bmatrix} 0 & 30 \\ 0 & 30 \\ 0.1 & 300 \end{bmatrix}$$

The standard deviation σ vector was set according to the variation of each of the genes in such a way that, with probability 0.99, the mutation should be smaller then 0.1 of the parameter variation range.

$$\sigma_m = \begin{bmatrix} \sigma_m^K \\ \sigma_m^{T_d} \\ \sigma_m^{T_i} \end{bmatrix} = \frac{1}{30} \begin{bmatrix} K^{\max} - K^{\min} \\ T_d^{\max} - T_d^{\min} \\ T_i^{\max} - T_i^{\min} \end{bmatrix} \tag{8}$$

Typical run of the algorithm is presented on figure 1. Figure 1(a) presents a diagram of the performance index of the most feasible individual in the population. The graph is typical, with the most rapid changes on the beginning of the optimization process. The frequency and stepsize of changes decrease with time. Figure 1(b) presents relative variance (variance divide by the square of the range) of the selected genes in all individuals in all epochs. One can notice that the selection sensitivity is quite strong, with decreasing variance staying on the relatively small values during the simulation period.

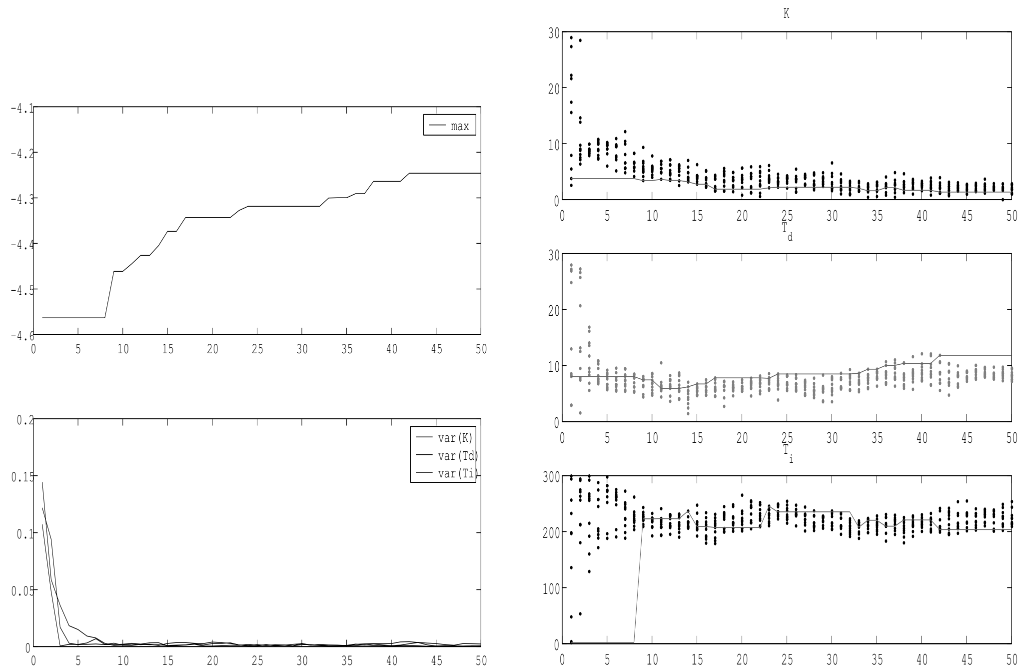
It is quite interesting to observe the changes in the values of the controller parameters in population. The graphs on figure 1(b) present the PID parameters. It can be seen that with time, the populations are more dense. The solid line presents the best individual.

As the result of the optimization the best parameters (the best individual) were found and are presented below.

$$\hat{X} = \begin{bmatrix} 1.36 \\ 11.82 \\ 203 \end{bmatrix}, \quad F(\hat{X}) = -4.24$$

The reference simulations are sketched on figure 2.

Similar experiments were performed for the second testing process. It appeared that due to the better 'controllability' of the process the part of the cost function responsible for the the control signal variation is practically switched off. The results prove that



(a) Best individual and population variance

(b) Chromosome values

Figure 1. Typical run of the algorithm (first object)

feature. New parameters ranges were chosen as:

$$\begin{bmatrix} K^{\min} & K^{\max} \\ T_d^{\min} & T_d^{\max} \\ T_i^{\min} & T_i^{\max} \end{bmatrix} = \begin{bmatrix} 0 & 300 \\ 0 & 300 \\ 0.1 & 3000 \end{bmatrix}$$

Mutation variance remains unchanged (still being some predefined range percentage). It is worth noticing that performance index in the instability situations reached very high values working as the soft constraint. Similarly to the previous example the algorithm operation is sketched on figure 3(a). The characteristic rapid increase of the cost function in the beginning of the optimization process can also be observed as for the selection mechanism effect. Lower number of step changes can be caused by the harder optimization task. The controller parameter changes are presented on figure 3(b).

It can be clearly seen that the controller gain is very close to its upper constraint. Thus the control signal constraint in the performance index is not so important. The same effect is clearly seen on the diagram presenting comparison of simulated process versus setpoint (figure 4). Both signals matches each other. The best found individual is as follows:

$$\hat{X} = \begin{bmatrix} 300 \\ 62.78 \\ 1870 \end{bmatrix}, \quad F(\hat{X}) = -4.24$$

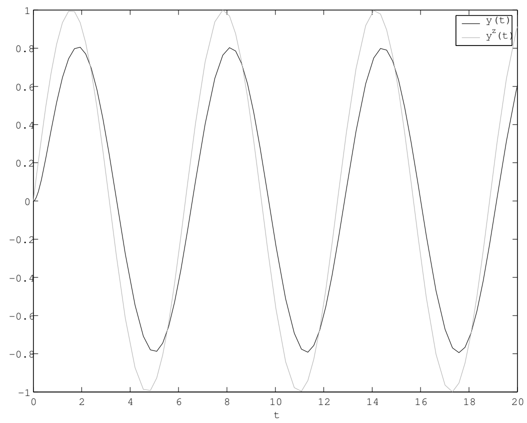


Figure 2. Controlled object simulation

The above experiments shows that evolutionary algorithm can be used for the tuning of the PID controller. The literature shows the same.

Now the problem of the controller on-line adaptation will be investigated. The evolutionary algorithm working as the tuner will be changed to match real-time operation and the adaptation task.

3 Adaptation algorithm

The analysis of dynamic evolutionary algorithm used for the controller tuning is investigated using the same examples as above: PID controller with an exemplary process. To check the adaptation, the process will be switching between two transfer functions:

$$G_3(s) = \frac{8.4}{9.78s^2 + 7.16s + 1} \tag{9}$$

and

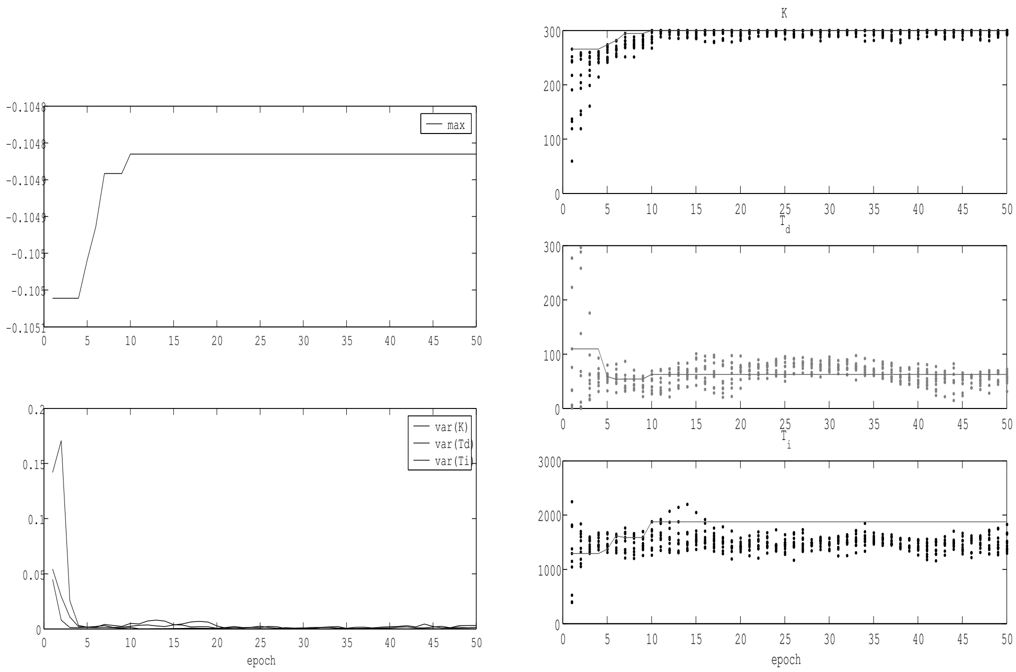
$$\widetilde{G}_3(s) = \frac{2.1}{9.78s^2 + 7.16s + 1} \tag{10}$$

The main parameter being changed is the process gain. This effect is often observed in practice. The dynamic evolutionary algorithm operates together with the controller. The optimization is being performed in real time. Each sampling period is equal to one epoch of the evolutionary algorithm.

3.1 Modified evolutionary algorithm

The optimization algorithm has to be modified to meet new 'dynamic' objectives. First, mutation operator was modified. The exponentially decaying dependent on the epoch number is switched off.

Additionally succession operator was modified. To increase algorithm exploration capabilities, simple succession was exchanged with a new one. Some of the individuals are deleted from the population and are exchanged with totally new ones initiated from



(a) Best individual and population variance

(b) Chromosome values

Figure 3. Typical run of the algorithm (second object)

scratch. Individuals for elimination are chosen in a way similar to tournament. So $\widetilde{q}$ elements are chosen and the worst of them is eliminated. The rest of the parameters remains unchanged.

3.2 Results

The algorithm was tested in several runs to find out the best parameter values.

$$\begin{bmatrix} K^{\min} & K^{\max} \\ T_d^{\min} & T_d^{\max} \\ T_i^{\min} & T_i^{\max} \end{bmatrix} = \begin{bmatrix} 0 & 30 \\ 0 & 30 \\ 0.1 & 300 \end{bmatrix}$$

To increase selective capabilities of the reproduction the tournament size was increased to $q = 4$. The process changes in the 50th sampling period. A typical optimization process is sketched in figure 5.

The adaptation effect can be clearly seen. Especially when we observe changes of the controller parameters.

It must be noted that the diagram presents individuals just before succession and thus the new initiated individuals are not seen. Their effect can be observed after 50th sampling period. The variance of the parameter values increases.

Also the impact of the mutation variance on the adaptation speed and the tuning quality was investigated. Figure 6 shows the performance index value for the best in-

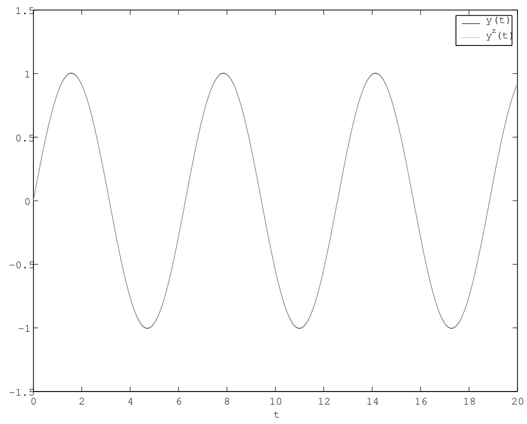
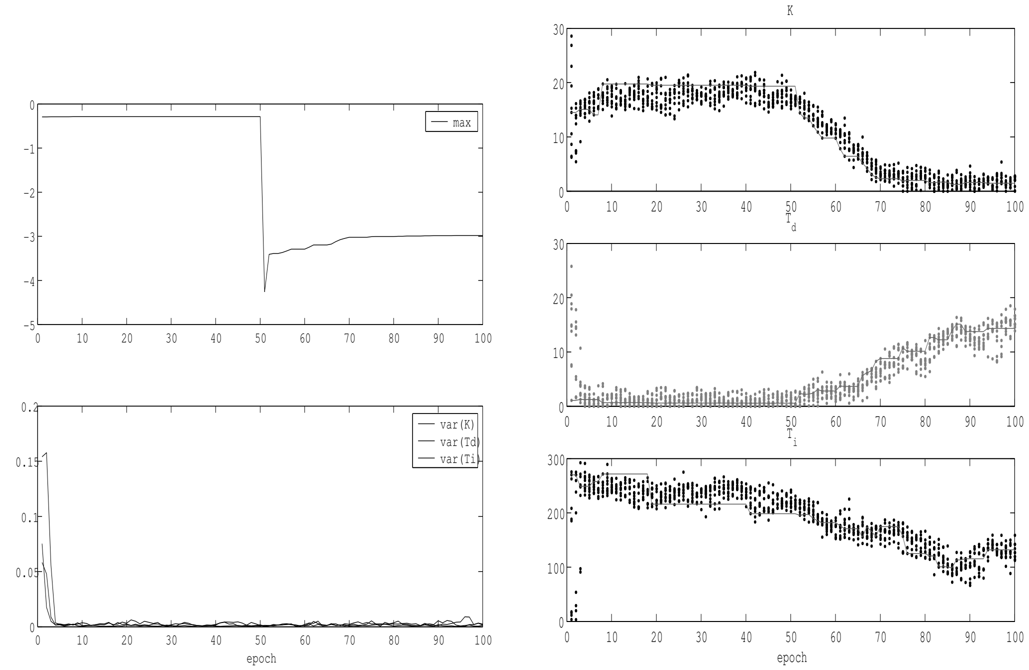


Figure 4. Controlled object simulation (second object)



(a) Best individual and population variance

(b) Chromosome values

Figure 5. Adaptive run of the algorithm

dividual found for two cases. The parameter before stochastic element added to the genotype is equal to $\frac{1}{30}$ or $\frac{1}{20}$.

In the second case the algorithm after the process change is much faster and subop-

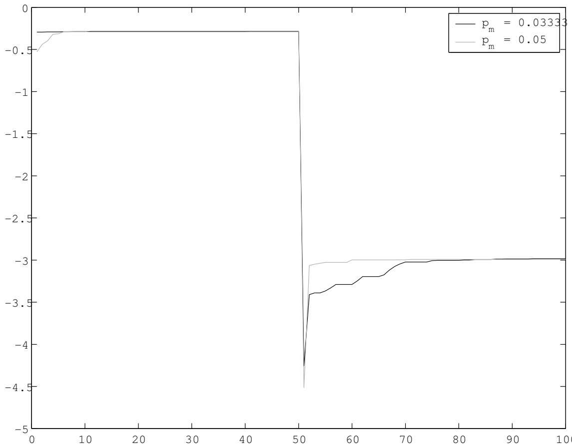


Figure 6. Performance comparison for different mutation parameter

timal parameters sets are reached in a lower number of epochs (sampling periods).

4 Conclusions

The paper presents the new approach to the controller tuning. The dynamic evolutionary algorithm is applied. The algorithm operates in real time, simultaneously with the sampling periods of the control loop.

First conclusion is that such an algorithm can work successfully. Second is that the optimization algorithm can dynamically operate with time. The algorithm also has very large potential. The same approach can be used for any type of the algorithm. Furthermore, we have full flexibility in choosing the cost function.

Bibliography

[1] J. Arabas. *Lectures from Evolutionary Algorithms.* WNT, 2001.

[2] Z. Bubnicki. *Algorithms and Control Theory.* PWN, 2002.

[3] Ph. deLarminat and Y. Thomas. *Automatic Control - Linear Systems.* WNT, 1983.

[4] T. Kaczorek. *Systems and Control Theory.* PWN, 1974.

[5] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness.* Prentice-Hall Advanced Reference Series, 1994.