TUNING OF PID ROBUST CONTROLLERS BASED ON ISE CRITERION MINIMIZATION

Cătălin N. Calistru, and Paul V. Georgescu
"*Gheorghe Asachi*" Technical University of Iaşi

Abstract – *PID control is the most widely used controller type in industry. Usually, the design engineer must tune PID parameters according to specific needs. The ability of obtaining a certain stationary and transient regime for the closed-loop system imposes a particular choice of the system structure and controller parameters, in concordance with performance accomplishment. Since a control problem may be seen as an optimization constraint problem the solution represents optimal values (so called tuning) for the controller parameters k_R, k_I, k_D. The control system robustness usually should be considered. The objective function is integral squared error (ISE) criterion subject to stability and robustness constraints. The paper presents two tuning strategies: symbolic tuning (ST), $\mathcal{H}_2/\mathcal{H}_\infty$ robust genetic tuning (RGT) and illustrative simulation results.*

Keywords – *PID control, tuning, symbolic tuning, robust control, robust genetic tuning*

1. INTRODUCTION

The increasing complexity of the modern control systems has emphasized the idea of applying new approaches in order to solve different control engineering problems. PID control is the most widely used controller type in industry and the usually design engineer must tune PID parameters according to specific needs [1]-[7], [12]-[21].

Three major factors in the PID controller tuning must be known: the plant, the controller type and performances of the control loop. The ability of obtaining a certain stationary and transient regime for the closed-loop system imposes a particular choice of the structure and controller parameters, in concordance with performance accomplishment. The controller tuning means therefore in finding optimum for the controller parameters $\boldsymbol{k_R}$, $\boldsymbol{k_I}$, $\boldsymbol{k_D}$.[

The first section of this paper deals with the ***symbolic-tuning*** procedure. Symbolic tuning of PID controllers assumes that the control system is stable and the controller structure is already chosen. It means that the controller parameters may vary inside a well define domain-the stability domain. However stability is not enough from the performance point of view and the designer imposes a certain integral type criterion that must be minimized. The most common are ISE (integral of squared error), IAE (integral of absolute error), ITSE (integral of time weighted squared error) etc. In this paper ISE criterion ($ISE = \int_0^\infty e^2(t)dt$) will be considered. ***Mathematica*** uses symbolic expressions to provide a general representation of mathematical and other structures. In this order, the generality of symbolic expressions allows ***Mathematica*** lead to the symbolic expression for the eligible controller parameters. An illustrative example show the effectiveness of the ***symbolic tuning (ST)*** approach.

One step ahead treated in the next section is to consider the problem of robustness for the systems under parameter perturbations.

Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control design approaches are useful for robust performance for systems under parameter perturbation and uncertain disturbances. However, the conventional output feedback designs of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal control are very complicated and not easily implemented for practical industrial applications. One solution is to consider an intelligent tuning approach via **GA** (genetic algorithms) so called ***robust genetic tuning (RGT)***

Genetic algorithms are optimization and machine learning algorithms initially inspired from the processes of natural selection and evolutionary genetics. In this section, the proposed algorithm will bridge the gap between the theoretical mixed optimal $\mathcal{H}_2/\mathcal{H}_\infty$ control and classical PID industrial control. The proposed $\mathcal{H}_2/\mathcal{H}_\infty$ control design consists in finding an internally stabilizing PID controller that minimizes an $\mathcal{H}_2$ integral performance index subject to an inequality constrained on the $\mathcal{H}_\infty$ norm of the closed loop transfer function. That means to solve two problems: stability robustness constraint and external disturbance attenuation constraint. The problem can be interpreted as a problem of optimal tracking performance ($\mathcal{H}_2$) subject to a robust stability constraint ($\mathcal{H}_\infty$) (or external disturbance attenuation constraint). The algorithm of robust genetic tuning is detailed and some examples are presented. Finally, a brief section of conclusions and future directions is presented.

2. SYMBOLIC TUNING

Let consider the PID classical control system .The plant to be controlled is *G(s)* and the PID controller is of the classical type *C*(*s*), *r* is reference, *e*=*r*-*y* is error, *A(s)*, *B(s)* Hurwitz polynomials with coefficients a_i, b_j. Then the controller transfer function is

$$C(s)=k_R+\frac{k_I}{s}+k_D s \tag{1}$$

We have in view to tune the controller parameters – to find a global optimum for the triplet k_R, k_I, k_D.
Therefore the error transfer function is:

$$e(s)=\frac{r(s)}{1+G(s)C(s)}=\frac{B(s)}{A(s)}=\frac{\sum_{j=0}^{m} b_j s^j}{\sum_{i=0}^{n} a_i s^i} \tag{2}$$

ISE criterion may be rewritten using Parseval formula:

$$J=\int_0^{\infty} e^2(t)\,dt=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty} e(-s)e(s)ds=$$
$$=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{r(-s)r(s)ds}{\left[1+G(-s)C(-s)\right]\left[1+G(s)C(s)\right]}=$$
$$=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{B(s)B(-s)}{A(s)A(-s)}ds$$

From Krasovskii-Pospelov formulae J may be explicited:

$$J=\frac{B_0^{/}\Delta_0+B_1^{/}\Delta_1+\cdots+B_{r-1}^{/}\Delta_{r-1}-2b_0^{/}b_1\Delta}{2a_0^2\Delta}$$
$$=\frac{\sum_{j=0}^{r-1}B_j^{/}\Delta_j-2b_0^{/}b_1\Delta}{2a_0^2\Delta} \tag{3}$$
$$b_0^{/}=b_{r-1}\left(-\frac{a_0}{a_r}\right)$$
$$b_1^{/}=b_{r-1}\left(\frac{b_0}{b_{r-1}}-\frac{a_1}{a_r}\right),$$
$$b_2^{/}=b_{r-1}\left(\frac{b_1}{b_{r-1}}-\frac{a_2}{a_r}\right),$$
$$\vdots$$
$$b_{r-1}^{/}=b_{r-1}\left(\frac{b_{r-2}}{b_{r-1}}-\frac{a_{r-1}}{a_r}\right)$$
$$B_0^{/}=b_0^{/2},\quad B_1^{/}=b_1^{/2}-2b_0^{/}b_2^{/},$$
$$B_2^{/}=b_2^{/2}-2b_1^{/}b_3^{/}+2b_0^{/}b_4^{/},$$
$$\vdots$$
$$B_{r-1}^{/}=b_{r-1}^{/\,2}$$

$$\Delta=\begin{bmatrix} a_0 & -a_2 & a_4 & -a_6 & \cdots & 0\\ 0 & a_1 & -a_3 & a_5 & \cdots & 0\\ 0 & -a_0 & a_2 & -a_4 & \cdots & 0\\ 0 & 0 & -a_1 & a_3 & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \cdots & a_{r-1}\end{bmatrix} \tag{4}$$

and Δ_i are obtained from Δ replacing the column $i+1$ with $\begin{bmatrix}a_1 & a_0 & 0 & \cdots & 0\end{bmatrix}^T$, $i=\overline{0,r-1}$.

Definition *Symbolic tuning* is the minimization of *J* with respect to controller parameters and stability constraints [8]-[10], [12], [15], [16], [21].

Having in view the iterative form (3) using Mathematica facilities one obtains the general symbolic solution for the controller parameters, $\boldsymbol{k_R}$, $\boldsymbol{k_I}$, $\boldsymbol{k_D}$.[15], [16].
Example Let us consider a second order plant, a PI controller. *Mathematica* lines are the following:

```
Gp[s]=1/ (a2 s^2+a1 s+a0)
Gr[s]=kr (1+1/ (s Ti))
e[s]=Together[1/(s (1+Gp[s] Gr[s]))]
n=CoefficientList[Numerator[e[s]],s]
n1={0,n[[1]],n[[2]],n[[3]]}
d1=CoefficientList[Denominator[e[s]],s
]
c=Cancel[Table[n1[[4]](n1[[i]]/n1[[4]]
-d1[[i]]/d1[[4]]),{i,1,3}]]
b={(c[[1]])^2,(c[[2]])^2-2     c[[1]]
c[[3]],(c[[3]])^2}
delta=Det[{{d1[[1]],-
d1[[3]],0},{0,d1[[2]],-d1[[4]]},{0,-
d1[[1]],d1[[3]]}}]
delta0=Det[{{d1[[2]],-
d1[[3]],0},{d1[[1]],d1[[2]],-
d1[[4]]},{0,-d1[[1]],d1[[3]]}}]
delta1=Det[{{d1[[1]],d1[[2]],0},{0,d1[
[1]],-d1[[4]]},{0,0,d1[[3]]}}]
delta2=Det[{{d1[[1]],-
d1[[3]],d1[[2]]},{0,d1[[2]],d1[[1]]},{
0,-d1[[1]],0}}]
SYMTUNE=(b[[1]] delta0+b[[2]]
delta1+b[[3]] delta2-
      2 c[[1]] n[[2]] delta)/(2
d1[[1]]^2 delta)
```

In fact, SYMTUNE is the objective functions supposed to be constrained by closed-loop stability conditions. What is remarkable in using *Mathematica* is the fact that *Mathematica* uses symbolic expressions to provide a general representation of mathematical and other structures. So one can find *symbolic solutions* for rather complicated problems such as differential and integral equations or constrained optimization problems. In the PI example presented above $\boldsymbol{J=J(k_R,\ k_I)}$ is the integral criteria ISE (Integral Squared Error Type) subject to the stability constraints. After some simple operations with *Lists* of coefficients (in *e* expression) and some determinants (Δ), using the clipping procedure the optimum point was obtained. In Fig 1 is depicted the three dimensional index (we have only two controller parameters to be optimized-tuned) and the clipping procedure – contour plot for the optimum isolation.

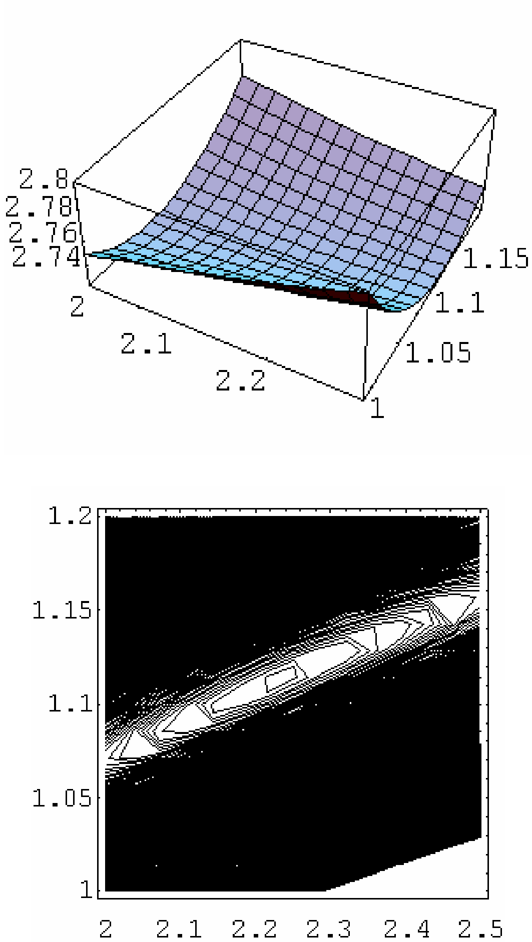

Fig.1 - ST approach for a PI controller.

Above, for simplicity $a_2 = a_1 = a_0 = 1$ and after the numerical simulation the optimum tuning parameters are $\boldsymbol{k_{Ropt}}$=1,172434, $\boldsymbol{k_{Iopt}}$=2,21326.

3. ROBUST $\mathcal{H}_2/\mathcal{H}_\infty$ CONTROL APPROACH

Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control design approaches are useful for robust performance for systems under parameter perturbation and uncertain disturbance. However, the conventional output feedback designs of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal control are very complicated and not easily implemented for practical industrial applications.

The proposed $\mathcal{H}_2/\mathcal{H}_\infty$ control design consists in finding an internally stabilizing PID controller that minimizes an $\mathcal{H}_2$ integral performance index subject to an inequality constrained on the $\mathcal{H}_\infty$ norm of the closed loop transfer function. There are two problems to be solved: stability robustness and external disturbance attenuation. The problem can be interpreted as a problem of optimal tracking performance subject to a robust stability constraint (or external disturbance attenuation constraint). The design procedure proposed for the PID robust tuning in order to achieve the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal performance follow the steps: 1.in the *first step* based on *Routh-Hurwitz* criterion, the stability domain of the three PID parameter space, which guarantees the stability of the closed loop is specified; 2.in the *second step*, the subset of the stability domain in the PID parameter space corresponding to step 1 is specified so that $\mathcal{H}_\infty$ constraint mentioned above is satisfied; 3.in the *third step* the design problems becomes, in the subset domain of the $\mathcal{H}_\infty$ constraint domain mentioned in step 2, how to obtain one point which minimizes the $\mathcal{H}_2$ tracking performance. This is generally considered to be a highly nonlinear minimization problem, in which many local minima may exist. A local minimum can be reached via genetic algorithms (GA). GA are parallel, global search techniques that emulate natural genetic operators. Because a *GA* simultaneously evaluates many points in the parameter space, *it is more likely to converge to the global solution*. It does not need to assume that the search space is differentiable or continuous, and can also iterate several times on each datum received. Global optimization can be achieved via a number of genetic operators, e.g., reproduction, mutation, and crossover. GAs are more suitable to the iterative PID $\mathcal{H}_2/\mathcal{H}_\infty$ control design for the following reasons: the search space is large; the performance surface does not require a differentiability assumption with respect to changes in PID parameters (therefore, the gradient-based searching algorithms that depend on the existence of the derivatives is inefficient); the likely fit terms are less likely to be destroyed under a genetic operator, thereby often leading to faster convergence.

Let consider the PID control system in Fig.2. The plant $G(s)$ to be controlled undergoes perturbation $\Delta G(s)$ and the PID controller, is of the classical type (1)

$$C(s) = k_R + \frac{k_I}{s} + k_D s$$

where the plant perturbation $\Delta G(s)$ is assumed to be stable but uncertain.

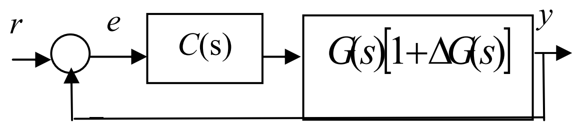

Fig.2 - PID control system with plant perturbation.

Suppose $\Delta G(s)$ is bounded according to the relation:

$$|\Delta G(j\omega)| \le |\xi(j\omega)|, \quad \forall \omega \in [0, \infty) \tag{5}$$

where the function $\xi(s)$ is stable and known.

The robust stability reveals that if a controller $C(s)$ is chosen so that nominal closed loop system (free of $\Delta G(s)$) in Fig.2 is asymptotically stable, and the following inequality holds,

$$\left\| \frac{G(s)C(s)\xi(s)}{1 + G(s)C(s)} \right\|_\infty \le 1. \tag{6}$$

then the closed loop system in Fig.1 is also asymptotically stable under plant perturbation (5), where the $\mathcal{H}_\infty$ norm in (8) is defined as:

$$\|G(s)\|_\infty = \sup_{\omega \in [0,\infty)} |G(j\omega| \tag{7}$$

i.e., the maximum peak of the spectral density of $G(s)$.
However, often robust stability alone is not enough in control system design. Optimal tracking performance is also appealing in much practical control engineering applications. Therefore, the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control problem is formulated as follows

$$\min \int_0^\infty e^2(t)\, dt \tag{8}$$

for the nominal closed loop system in Fig. 1, subject to the robust stability constraint (6), where $e(t)$ is the tracking error. That means, under the constraint (6), the error energy (*integral squared error criterion*) (8) must be as small as possible. From the above analysis the control design problem involve how to specify a PID controller to achieve the optimal tracking (8) subject to the robust stability constraint (6).
Coming back to (5), for *m=1, 2, 3* (usually in practical applications the plant maximum order is 3), following the methodology explained in previous section, one obtains via *Mathematica*

$$\begin{aligned} J_1 &= \frac{b_0^2}{2a_0a_1} \\ J_2 &= \frac{b_1^2a_0 + b_0^2a_2}{2a_0a_1a_2} \\ J_3 &= \frac{b_2^2a_0a_1 + (b_1^2 - 2b_0b_2)a_0a_3 + b_0^2a_2a_3}{2a_0a_3(-a_0a_3 + a_1a_2)} \end{aligned} \tag{9}$$

where $a_i, b_i,\ i = \overline{0,3}$ depend only on the plant and controller parameters.
Then, the robust performance in (6) must be of the following form:

$$J_m = \min J_m(k_R, k_I, k_D) \tag{10}$$

From the definition (4), the constraint in (3) can be expressed by:

$$\left\| \frac{G(s)C(s)\xi(s)}{1+G(s)C(s)} \right\|_\infty =$$
$$= \sup_{\omega\in[0,\infty)} \sqrt{\frac{G(-j\omega)G(j\omega)C(-j\omega)C(j\omega)\xi(-j\omega)\xi(j\omega)}{[1+G(-j\omega)C(-j\omega)][1+G(j\omega)C(j\omega]}} \le 1.$$

or:

$$\left\| \frac{G(s)C(s)\xi(s)}{1+G(s)C(s)} \right\|_\infty = \sup_{\omega\in[0,\infty)} \sqrt{\frac{p(\omega)}{q(\omega)}} = \sqrt{\sup_{\omega\in[0,\infty)} \frac{p(\omega)}{q(\omega)}} =$$
$$= \sqrt{\sup_{\omega\in[0,\infty)} \Theta(\omega)} \le 1.$$

where $p(\omega)$ and $q(\omega)$ are some appropriate polynomials of ω.
The physical meaning of the above relation is that *if the largest peak of* $\Theta(\omega)=p(\omega)/q(\omega)$ *is less than 1, then the system in Fig.2 is stable under plant perturbation*.
Generally speaking, to scan $\omega\in[0,\infty)$ to find the peaks of $\Theta(\omega)$ is not an easy task. Actually, the peaks of $\Theta(\omega)$ occur at the points which must satisfy the following equation:

$$\begin{aligned} &\frac{d\Theta(\omega)}{d\omega} = 0 \Leftrightarrow \\ &\Leftrightarrow p(\omega)\frac{dq(\omega)}{d\omega} - q(\omega)\frac{dp(\omega)}{d\omega} = 0 \Leftrightarrow \\ &\Leftrightarrow \prod_{i=1}^{n} (\omega - \alpha_i) = 0 \end{aligned} \tag{11}$$

Therefore, only the real roots α_i of the above equation need to be found. So, the robust stability constraint in (6) is equivalent with:

$$\sqrt{\max_{\alpha_i} \frac{p(\alpha_i)}{q(\alpha_i)}} \le 1 \tag{12}$$

The design procedure is, therefore, a minimization problem (8) under the inequality constraint (12). The algorithm follows the steps:
Step 1**:** Given a plant $G(s)$, PID controller and the enveloping function ξ(s) of the plant perturbation.
Step 2**:** Specify the parameter domain Δ of (k_R, k_I, k_D) to guarantee the stability of the nominal closed loop system via the Routh-Hurwitz criterion, where:

$$\Delta := \{(k_R, k_I, k_D) \subset R^3\} \tag{13}$$

Step 3**:** Compute a set of parameters in Δ from *GA* and compute α_i, $i=1,n$ from (11).
Step 4**:** Check if the relation (12) is fulfilled.
Step 5: Compute J_m (10) in order to obtain robust performance. Then repeat the procedure *Step3* to *Step 5* until a suitable parameter set is obtained.
Definition. The above algorithm described by Step1-Step5 is called ***robust genetic tuning (RGT).***
Genetic Algorithms are parameter optimization methods following the examples of the natural behavior of living species. A number of points in the search space are considered as a population of living creatures inside an artificial world. Basic Darwinistic propagation in such world enables the fittest to survive. Although some random effects play an important role, the behavior is not pure coincidental, since the historical information inside the individuals are being conserved as much as possible in view of the common goal, i.e. *survival of the fittest*. GA have been developed by John Holland during the late 60's but the 1989 David Goldberg's book "*Genetic Algorithms in Search, Optimization and Machine Learning*" bring the GAs into the spotlight and make them as one of the more promising emergent architectures for optimization in an overwhelming number of application fields. Important aspects of GAs are universality and robustness, because species are to be believed to survive in many different and hostile environments. As David Goldberg states *"...where robust performance is desired (and where is it not) nature does it better; the secrets of adaptation and survival are best learned from a careful study of a biological example. Yet we do not accept the genetic algorithm method by appeal to this beauty-of-nature argument alone. Genetic Algorithms are theoretically*

and empirically proven to provide robust search in complex spaces[11].GAs are powerful search algorithms based on the mechanics of natural selection and natural genetics. The algorithms work with a *population of strings*, searching many peaks in parallel as opposed to a single point; use probabilistic transition rules instead of deterministic rules; use objective function information instead of derivatives or other auxiliary knowledge. GAs are inherently parallel, because they simultaneously evaluate many points in the search space. Considering many points in the search space they have a reduced chance of converging to the local optimum and would be *more likely to converge to the global optimum*. GAs require only information concerning the quality of the solution produced by each parameter set (objective function evaluation). This differs from many optimization approaches, which require derivatives information, or, worse yet, complete knowledge of the problem structure and parameters. *Since genetic algorithms do not require such problem specific information, they are more flexible than more search methods*. A genetic algorithm is an iterative procedure, which maintains a constant size population of candidate solutions. During each iteration step, or generation, three genetic operators (*reproduction*, *crossover* and *mutation*) are performing to generate new populations (offsprings), and the chromosomes of these new populations are evaluated via the value of fitness which is related to some cost functions. On the basis of these genetic operators and evaluation, the better new populations of candidate solution are formed. It is shown in the *SCHEMA THEOREM*, that the genetic search algorithm will converge from *the viewpoint of schema*. With the above descriptions, the procedure of a simple genetic algorithm is given as follows:

1. *Generate randomly a population of binary strings*.
2. *Calculate the fitness for each string in the population*.
3. *Create offspring strings by simple GA operators*.
4. *Evaluate the new strings and calculate the fitness for each string*.
5. *If the search goal is achieved, or an allowable generation is attained, stop and return*.

GA are working with a population of binary strings, not with the parameters themselves. For example, with the binary coding method, The PID parameters set would be coded as binary strings, of 0's and 1's with different length. The designer in the search space specifies the choice of a certain length. In the binary coding, the bit length B_i and the corresponding resolution *the relation relates Ri*:

$$R_i = \frac{M_i - m_i}{2^{B_i} - 1} \tag{14}$$

where M_i and m_i are the upper and the lower of the parameter k_i. As a direct result, the PID parameter set (k_R, k_I, k_D), can be transformed into binary string (chromosome), with the length:

$$L = \sum_i B_i \tag{15}$$

The decoding procedure is the reverse procedure of coding.

In this paper, the *fitness* and *cost function* is obviously defined with the relation:

$$E(k_R, k_I, k_D) = J_m(k_R, k_I, k_D) \tag{16}$$

where the triplet $(k_R, k_I, k_D) \in \Delta$. The *fitness* value is a reward based on the performance of the possible solution represented by the string, or it can be thought of *as how well a PID controller can be tuned according to the string to actually minimize the tracking error*. The better the solution encoded by a string (chromosome), the higher the fitness. To minimize the quality index in (16) is equivalent to getting a maximum fitness value in the genetic searching algorithm. A chromosome that has lower quadratic index should be assigning a larger fitness value. Then the genetic algorithm tries to generate better offsprings to improve the fitness. Therefore, a better PID controller could be obtained via better fitness in genetic algorithms. There are quite a number of approaches to perform this mapping known as *fitness techniques*. In this paper is proposed the technique so-called *windowing*. Now, let us shortly describe the operation with the three basic operators.

Reproduction. Reproduction is based on the principle of survival of the better fitness. The fitness of the *i*th string F_i is assign to each individual string in the population where higher F_i means as shown better fitness. These strings with large fitness would have a large number of copies in the new generation.

Crossover. By the second operator, the strings exchange information via probabilistic decisions. Crossover provides *a mechanism for strings to mix and match their desirable qualities through a random process*.

Mutation. The third operator, *mutation*, enhances an ability of genetic algorithms to find a near-optimal solution. Mutation is the occasional alternation of a value at a particular string position. In the case of binary coding, the mutation operator simply flips the state of a bit from *0* to *1* and vice versa. Mutation should be used sparingly because it is a random search operator. As said above the convergence of a genetic search algorithm is discussed from the viewpoint of *schema*.

Example. Let us consider the control system shown in Fig. 2., a *PD* controller would be given to achieve the mixed H_2/H_∞ optimal tracking under the bounded plant perturbation.

$$C(s) = k_R + k_D s, \quad G(s) = \frac{1}{s^2}, \quad |\Delta G(s)| \le \left| \frac{0.1}{s^2 + 0.1s + 10} \right|.$$

Suppose the input command is a unit step, then:

$$e(s) = \frac{s}{s^2 + k_D s + k_R} = \frac{B(s)}{A(s)}$$

The cost function is:

$$E(k_R, k_I, k_D) = J_2(k_R, k_I, k_D) = \frac{b_1^2 a_0 + b_0^2 a_2}{2a_0 a_1 a_2} = \frac{k_R}{2k_R k_D} = \frac{1}{2k_D}$$

The linear relation gives the relation between fitness function and cost function. The robust stability constraint lead to the relation (11) with appropriate polynomials for $\Theta(\omega)$. The genetic algorithm begins by randomly generating a population of 1.000 chromosomes. *After 10 generation, proper controller parameter can be obtained*. Thus, the obtained values for controller parameters are $k_R=100$ and $k_D=30$.

4. CONCLUSIONS

This paper presents non-conventional PID intelligent control methodologies so called ***symbolic tuning (ST)*** and ***robust genetic tuning (RGT)***. The common denominator is ISE criterion and the main goal is the robustness of the control loop. Examples show clearly the effectiveness of the proposed robust approaches. GA are more suitable to the iterative PID $\mathcal{H}_2/\mathcal{H}_\infty$ control design for the following reasons: the search space is large; the performance surface does not require a differentiability assumption with respect to changes in PID parameters. Therefore, the gradient-based searching algorithms that depend on the existence of the derivatives are inefficient. The likely fit terms are less likely to be destroyed under a genetic operator, thereby often leading to faster convergence. The algorithm proposed has a methodological advantage over the traditional PID tuning methods. The design becomes fully systematic, saving design time, reducing the need for non-explicit knowledge.

REFERENCES

[1] A. Kaya., T.J. Scheib, *Tuning of PID controllers of different structures*, Control Engineering, **7**, pp. 62-65, 1988.

[2] C.N. Calistru, *A symbolic optimization approach for tuning of PID controllers*, Proceedings of the 4th IEEE Conference on Control Applications, CCA'95, Albany, New York, pp.174-175, 1995.

[3] J.G. Ziegler, N.B. Nichols, *Optimum setting for PID controllers*, Transactions of ASME, **64**, pp. 759-768, 1942.

[4] K.L. Chien, A.J. Hrones, J.B. Reswick, *On the automatic control of generalized passive systems*, Transactions of ASME, **74**, pp. 175-185, 1952.

[5] K.J. Aström, T. Hägglund, *Automatic tuning of simple regulators with specifications on phase and amplitude margins,* Automatica, **20,** pp. 645-651, 1984.

[6] K.J. Aström, T. Hägglund, *Automatic tuning of PID regulators*, ISA., Research Triangle Park, 1988.

[7] D.W. Clarke, *PID algorithms and their computer implementation*, Transactions of Instrumentation Measurement and Control, **6**, pp. 305-316, 1984.

[8] C.N. Calistru, I. Olah, L. Mastacan, *A Method for tuning PI controllers in mechanical and metallurgical processes*, Proceedings of TCEPCEMMME'94 Workshop, Miskolc, pp. 88-93, 1994.

[9] C. Lazăr, C.N. Calistru, S. Ifrim, *Mini-toolbox for tuning PID controllers using MATHEMATICA facilities*, Proceedings of WAAC'93, pp. 115-119, Prague, 1993.

[10] C.N Calistru, *A Symbolic approach to tuning PID controllers via MATHEMATICA software,* International Journal on System Analysis Modelling and Simulation, Gordon & Breach Publishing House, New Jersey, **18-19**, pp. 341-344, 1995.

[11] D.E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Reading, Addison Wesley, 1989.

[12] C.N. Calistru, *Solving control engineering problems using symbolic calculus facilities,* Proceedings of 15th IMACS World Congress, Berlin, pp. 789-791, 1997.

[13] D.P. Atherton, P. Borne, *Concise Encyclopedia of Modelling and Simulation*, Pergamon Press, New York, 1992.

[14] P. De Larminat *La commande robuste: un tour d'horizon*, APII, **25**, no.3,1991.

[15] S. Wolfram, *Mathematica-A system for Doing Mathematics by Computer,* Addison Wesley, 1994.

[16] M.L. Abbel, J.P. Braselton, *Mathematica by Example*, Academic Press, New York, 1992.

[17 F.L. Lewis, *Applied Optimal Control & Estimation,* Prentice Hall, 1992.

[18] C.N. Calistru, *Increasing Robustness of Control Systems Using Integral Criteria and Alternate Feedback*, (in Romanian) MatrixRom, Bucureşti, 2004.

[19] C.N. Calistru, *Robust control via genetic algorithms and integral criteria minimization*, WSEAS Transactions on Systems, Issue **2**, Vol. **3**, pp. 894-900, 2004.

[20] C.N. Calistru, C.G. Haba, C.C. Pruteanu, *Intelligent tuning of PID robust controllers*, Proceedings of the 5th International Conference on Microelectronics and Computer Science ICMCS-2007, Chişinău, pp.209-218, 2007.

[21] C.N. Calistru *Intelligent PID robust control based on integral criteria*, Proceedings of the 11th WSEAS International Conference on Automatic Control, Modelling and Simulation (ACMOS '09), pp.534-540, Istanbul, 2009.

Dr. Cătălin Nicolae Calistru teaches Computer Networks and Computational Logics at the Faculty of Automatic Control and Computer Engineering but his interests cover various fields as Control Theory, Robust Control, Intelligent Control, Nonlinear Control, Applied Mathematics. Prof. Calistru was invited professor and lectured at the Charles University in Prague, Universita degli Studi di Firenze, won Excellency Diploma at IEEE TTTC AQTR'04 Cluj. He is author of more than 90 scientific papers, 3 books., member of the IASTED Technical Control Committee (Calgary, Canada) member of SRAIT, IFAC, WSEAS, NAUN, IASTED and editor of the *International Journal of Mathematics and Computers in Simulation (USA*).

Dr. Paul Georgescu was born in Braila, Romania, in 1971. He holds a D. Sc. in Pure Mathematics awarded by Hiroshima University, Japan, in 2000 and a Ph. D. in Applied Mathematics awarded by Central European University, Hungary in 2008. His research interests include control theory and its applications to the modelling of real-life systems. Dr. Georgescu authored more than 20 scientific papers and a book and is an editor of *Journal of Biological Systems*.