Robust Control via Genetic Algorithms and Integral Criteria Minimization

CĂTĂLIN NICOLAE CALISTRU

Department of Automatic Control & Industrial Informatics
"*Gheorghe Asachi*"Technical University of Iaşi
Str. Prof. D. Mangeron 53A, RO-6600, Iaşi
ROMANIA

Abstract: PID controllers have found extensive industrial applications for several decades. Recently, mixed H_2/H_∞ optimal control problems have received a great deal of attention from the viewpoint of theoretical design. Mixed H_2/H_∞ control design approaches are useful for robust performance for systems under parameter perturbation and uncertain disturbance. In this paper, a design procedure is proposed to tune PID controller parameters to achieve mixed H_2/H_∞ optimal performance. Since the criterion that must be optimized is of integral type, and the constraints are imposed by the robust control, the problem to be solved is a highly nonlinear minimization problem, in which many local minima may exist. Genetic algorithms are parallel, global search techniques that emulate natural genetic operators. Because a genetic algorithm simultaneously evaluates many points in the parameter space, *it is more likely to converge to the global solution. MATHEMATICA* is powerful symbolic calculus software, which also will be used in the paper to illustrate better the results. An illustrative example confirms that good performance can be achieved by the proposed method.

Keywords: PID control, optimization, robust control, genetic algorithms, symbolic computation

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. The PID control is the most widely used controller type in industry and the design engineer must tune PID parameters according to specific needs. Three major factors in the PID controller tuning must be known: the plant, the controller type and the performance criterion of the control loop [1]. Within the framework of the classical theory, one of the factors very important in tuning PID controllers, the performance criteria, may be divided into two main groups*: time domain criteria* and *frequency domain criteria*. In this respect, the designer must simultaneously meet more ore less hard to satisfy the design limits [2]. Starting from these, in the PID control the following tuning methods are typically used: Ziegler and Nichols (1942), Chien, Hrones and Reswick (1952), Clarke (1984), Kaya and Scheib (1988), Aström and Hägglund (1984, 1988), [1], [3], [4], [5], [6], [7]. The performance mentioned above are characterized by one obvious disadvantage: the designer must more or less try to satisfy the design limits. 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 performance indexes are specified by design. Because in most of the cases the achievement of good quality performance indexes lead to contradictory solutions and the trial to "box in" the system response within limits of the types as: a: zero steady-state error, good overshoot and overshoot time, rise time, system time response, gain and phase margin etc. the only way out seems to be via *trial and error* synthesis methods. The system error *e* is the signal that is most likely to influence the mentioned performances [8], [9], [10]. As the duration of error also must play a role, one finds that the most meaningful of these performance criteria have the integral form. Mixed H_2/H_∞ control design approaches are useful for robust performance for systems under parameter perturbation and uncertain disturbance. However, the conventional output feedback designs of mixed H_2/H_∞ optimal control are very complicated and not easily implemented for practical industrial applications. Genetic algorithms (mnemonic ***GAs***) are optimization and machine learning algorithms initially inspired from the processes of natural selection and evolutionary genetics. In this paper, the proposed algorithm will bridge the gap between the theoretical mixed optimal H_2/H_∞ control and classical PID industrial control. *The proposed H_2/H_∞ control design consists in finding an internally stabilizing PID controller that minimizes an H_2 integral performance index subject to an inequality constrained on the H_∞ norm of the closed loop transfer function*. That means the solving of the two problems: stability robustness constraint and external disturbance attenuation constraint. 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 off line PID tuning in order to achieve the mixed H_2/H_∞ 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 H_∞ constraint mentioned above is satisfied

3.in the *third step* the design problems becomes, in the subset domain of the H_∞ constraint domain mentioned in step 2, how to obtain one point which minimizes the 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 ***GAs.***

Genetic algorithms are parallel, global search techniques that emulate natural genetic operators [11]. 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 H_2/H_∞ 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. *MATHEMATICA* is powerful symbolic calculus software, which also will be used in the paper to illustrate better the results. [12]. *MATHEMATICA* offers a reach extensible environment for engineering applied mathematics. Some illustrative examples obtain by simulation confirm that good performance can be achieved by the proposed method.

2. PID Robust Control Design

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

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

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

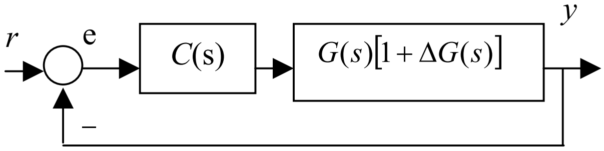

Fig.1. *PID control system with plant perturbation*

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

$$|\Delta G(j\omega)| \le |\xi(j\omega)|, \quad \forall \omega \in [0, \infty) \tag{2}$$

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.1 is asymptotically stable, and the following inequality holds,

$$\left\| \frac{G(s)C(s)\xi(s)}{1+G(s)C(s)} \right\|_\infty \le 1. \tag{3}$$

then the closed loop system in *Fig.1* is also asymptotically stable under plant perturbation (2), where the H_∞ norm in (3) is defined as:

$$\|G(s)\|_\infty = \sup_{\omega\in[0,\infty)} |G(j\omega| \tag{4}$$

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 H_2/H_∞ control problem is formulated as follows:

$$\min\int_0^\infty e^2(t)\,dt \tag{5}$$

for the nominal closed loop system in Fig. 1, subject to the robust stability constraint (3), where $e(t)$ is the tracking error. That means, under the constraint (3), the error energy (*integral squared error criterion*) (5) 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 (5) subject to the robust stability constraint (3).

In the nominal case, the tracking error signal is given by:

$$e(s)=\frac{r(s)}{1+G(s)C(s)}=\frac{B(s)}{A(s)} \tag{6}$$

Using *Parseval's theorem*, one obtains the expression of the integral criterion:

$$\begin{aligned}J&=\min\int_0^\infty e^2(t)\,dt=\min\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}e(-s)e(s)ds=\\&\min\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{r(-s)r(s)ds}{[1+G(-s)C(-s)][1+G(s)C(s)]}=\\&=\min\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{B(s)B(-s)}{A(s)A(-s)}ds,\end{aligned} \tag{7}$$

where $B(s)$ and $A(s)$ are Hurwitz polynomials of s with appropriate degree. The minimization problem in the equation (7) can be solved with the aid of the residue theorem. Let

$$A(s)=\sum_{k=0}^{m}a_k s^k;\quad B(s)=\sum_{k=0}^{m-1}b_k s^k\,. \tag{8}$$

Then using *MATHEMATICA* facilities, and Krasovschii-Pospelov formulae one obtains the general form of J_{m}:

$$J_m=\frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{\sum\limits_{k=0}^{m-1}b_k s^k\sum\limits_{k=0}^{m-1}b_k(-s)^k}{\sum\limits_{k=0}^{m}a_k s^k\sum\limits_{k=0}^{m}a_k(-s)^k}ds \tag{9}$$

For *m=1, 2, 3* (usually in practical applications) one obtains [2]:

$$\begin{aligned}&J_1=\frac{b_0^2}{2a_0a_1};\quad J_2=\frac{b_1^2a_0+b_0^2a_2}{2a_0a_1a_2};\\&J_3=\frac{b_2^2a_0a_1+(b_1^2-2b_0b_2)a_0a_3+b_0^2a_2a_3}{2a_0a_3(-a_0a_3+a_1a_2)}\end{aligned} \tag{10}$$

where $a_i, b_i,\ i=\overline{0,3}$ depend on the plant and controller parameters.

Then, based on the residue theorem, the robust performance in (9) must be of the following form:

$$J_m=\min J_m(k_R,k_I,k_D) \tag{11}$$

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

$$\begin{aligned}&\left\|\frac{G(s)C(s)\xi(s)}{1+G(s)C(s)}\right\|_\infty=\\&\sup_{\omega\in[0,\infty)}\sqrt{\frac{G(-j\omega)G(j\omega)C(-j\omega)C(j\omega)\xi(-j\omega)\xi(j\omega)}{[1+G(-j\omega)C(-j\omega)][1+G(j\omega)C(j\omega]}}\le 1.\end{aligned} \tag{12}$$

or:

$$\begin{aligned}&\left\|\frac{G(s)C(s)\xi(s)}{1+G(s)C(s)}\right\|_\infty=\sup_{\omega\in[0,\infty)}\sqrt{\frac{p(\omega)}{q(\omega)}}=\sqrt{\sup_{\omega\in[0,\infty)}\frac{p(\omega)}{q(\omega)}}=\\&=\sqrt{\sup_{\omega\in[0,\infty)}\Theta(\omega)}\le 1.\end{aligned} \tag{13}$$

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. 1 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:

$$\frac{d\Theta(\omega)}{d\omega}=0\Leftrightarrow p(\omega)\frac{dq(\omega)}{d\omega}-q(\omega)\frac{dp(\omega)}{d\omega}=0\Leftrightarrow\prod_{i=1}^{n}(\omega-\alpha_i)=0 \tag{14}$$

Therefore, only the real roots α_i of the above equation need to be found. So, the robust stability constraint in (3) is equivalent with:

$$\sqrt{\max_{\alpha_i}\frac{p(\alpha_i)}{q(\alpha_i)}}\le 1 \tag{15}$$

The design procedure is, therefore, a minimization problem (11) under the inequality constraint (15). The algorithm follows the steps:

Step 1: Given a plant $G(s)$, PID controller and the enveloping function $\xi(s)$ 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:

$$\Delta := \{(k_R, k_I, k_D) \subset R^3\} \tag{16}$$

Step 3: Compute a set of parameters in Δ from ***GA*** and compute α_i, *i=1,n* from (14).

Step 4: Check if the relation (15) is fulfilled.

Step 5: Compute J_m (11) in order to obtain robust performance. Then repeat the procedure ***Step3*** to ***Step 5*** until a suitable parameter set is obtained.

3. *GA* for mixed robust control

Genetic Algorithms ***(GAs)*** 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.* GAs 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* [11]., 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.

Genetic algorithms 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*:

$$R_i = \frac{M_i - m_i}{2^{B_i} - 1} \tag{17}$$

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:

$$L = \sum_i B_i \qquad (18)$$

The decoding procedure is the reverse procedure of coding.

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

$$E(k_R, k_I, k_D) = J_m(k_R, k_I, k_D) \qquad (19)$$

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 (19) 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 obtain 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* [11], as described in Fig. 3.,

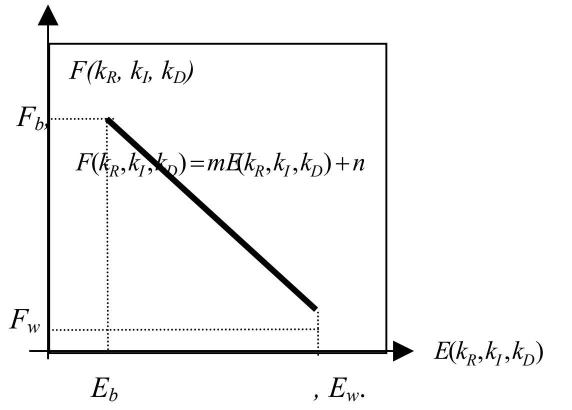

Fig. 3. The relation between the cost function and fitness function F

where *m* and *n* are computed in each generation according with F_b, F_w, E_b, E_w.

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*.[11]

4. Design example. Conclusions

In order to illustrate the effectiveness of the proposed approach the following example with numerical simulation is given:

Example. Let us consider the control system shown in Fig. 1. A PD controller would be given to achieve the mixed H_2/H_∞ optimal tracking under the bounded plant perturbation. So,

$$C(s)=k_R+k_D s, \quad G(s)=\frac{1}{s^2}, \quad \left|\Delta G(s)\right| \le \left|\frac{0.1}{s^2+0.1s+10}\right|. \qquad (20)$$

Suppose the input command is a unit step, then:

$$e(s) = \frac{s}{s^2 + k_D s + k_R} = \frac{B(s)}{A(s)} \qquad (21)$$

The cost function is:

$$E(k_R,k_I,k_D)=J_2(k_R,k_I,k_D)=\frac{b_1^2 a_0 + b_0^2 a_2}{2a_0 a_1 a_2}=\frac{k_R}{2k_R k_D}=\frac{1}{2k_D} \qquad (22)$$

The linear relation gives the relation between fitness function and cost function. The robust stability constraint lead to the relation (13) 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.* The obtained values for controller parameters are $k_R=100$ and $k_D=30$.

In *Fig. 4*, is represented the step response of the control system.

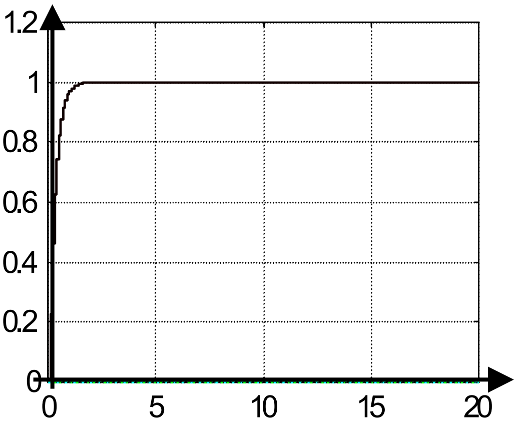

Fig. 4. The step response of the control system

In conclusion, the presented example shows clearly the effectiveness of the proposed robust approach. GAs are more suitable to the iterative PID H_2/H_∞ 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.

References:

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

[2]Calistru, CN., "A symbolic optimization approach for tuning of PID controllers", *Proc. of . the 4th IEEE Conference on Control Applications, CCA'95, Albany, New York*, pp. 174-177, 1995.

[3]Ziegler, JG., and Nichols, NB., "Optimum setting for PID controllers", *Trans. ASME*, **64**, pp. 759-768, 1942.

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

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

[6]Aström, K.J., and Hägglund, T., "Automatic tuning of PID regulators", *I.S.A., Research Triangle Park*, 1988

[7]Clarke, DW., "PID algorithms and their computer implementation", *Trans. Inst. Meas. Control*, **6**, pp.305-316, 1984.

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

[9]Lazăr C., Calistru, CN., and Ifrim, S., "Minitoolbox for tuning PID controllers using *MATHEMATICA* facilities", *Proc. of WAAC'93*, pp.115-119, Prague, 1993.

[10]Calistru, CN., "A Symbolic approach to tuning PID controllers via *MATHEMATICA* software*",Int. Journal on* ***SAMS*** *Gordon & Breach Publishing House*, **18-19**, pp. 341-344, 1995.

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

[12]Calistru, CN., "Solving control engineering problems using symbolic calculus facilities", *Proc. of 15th IMACS World Congress*, Berlin, 1997

[13]Atherton, DP., Borne P., *Concise encyclopedia of modelling and simulation*, Pergamon Press, New York, 1992.

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

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

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

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