Genetic Multivariable PID Controller Based on IMC

Sh. Kermanshachi
Department of Electrical Engineering
Azad University, Science & Research Branch
Tehran, Iran
Sh.kermanshachi@gmail.com

N. Sadati
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
Sadati@sina.sharif.edu

***Abstract*—A new approach for PID tuning, based on GA (Genetic algorithm) and Internal Model Control (IMC) technique, is presented in this paper. PID tuning is based on using Method. The IMC technique reduces the number of parameters that must be tuned for a multivariable system using PID controller. The algorithm uses GA for optimal determination of IMC variables. Simulation results present the good performance of the proposed method.**

I. INTRODUCTION

Nowadays, industry widely utilizes PID controllers due to their familiarity and simplicity of structure. Advantages of a PID controllers include simplicity, robustness, but it cannot effectively control a complicated or fast running system. Since the response of a plant using PID controller depends only on the gains P, I and D, the subject of many of papers are related to their tuning.

In some papers, the tuning of PID parameters are based on an exact form of the process, expressed by a transfer function [1]-[3]. Although multivariable PID control schemes have been researched extensively for over 30 years, with various design methodologies [4],[5], still the PID controller design for multi-input multi-output (MIMO) systems is a much more complicated problem compared to the SISO case. Apart from the fact that MIMO PID controllers have many more parameters than SISO PID controllers, the loop interaction (or coupling) is the more challenging problem. This makes it difficult for designer to design each loop independently, as the tuning of controller variables of one loop will affect the performance of the others and may even destabilize the entire system [6],[7].

By developing fully cross-coupled PID controllers that inherently compensates the interactions, satisfactory responses can be obtained.

Wang et al. [8], introduced a fully cross-coupled multivariable PID controller for processes with small dead times. The PID controller parameters can be selected to give the best least squares fit to the desired diagonal close loop frequency response over a prespecified rang of frequencies.

The IMC concept was developed by and its concept was generalized so that the transfer function in the feedback form of IMC could be expanded in Maclaurin series form about zero, while dropping all terms higher than second order in the Laplace domain [9]-[11].

This paper presents a Genetic multivariable PID controller based on IMC. By using IMC method, decoupled controllers are achieved and the number of parameters that must be tuned for a multivariable system could be reduces to the number of loops. So the main goal of this paper is regulation of IMC parameters, which determines PID controller coefficients for a multivariable system. Due to the main benefits of GA that is a powerful search optimization technique, based on the mechanics of natural selection and natural genetics, this approach is used for the regulation of IMC parameters. By defining appropriate fitness function in genetic algorithm, satisfactory results are obtained.

II. DESIGNING MULTIVARIABLE CONTROLLER

Block diagram of proposed controller is shown in Fig. 1. At first, the PID controller parameters are tuned by IMC technique and then, by using genetic algorithm, the parameters of IMC are determined optimally.

A. PID Tunning Based on IMC

In this section, the IMC technique for PID controller tuning is explained. For multivariable systems, the IMC structure reduces the classical feedback structure with the controller C(s) which is the conjunction of IMC controller and the process model , as shown in Fig. 2 and is described below.

$$\mathbf{C}(s)=\mathbf{Q}(s)\left[\mathbf{I}-\mathbf{M}(s)\mathbf{Q}(s)\right]^{-1} \tag{1}$$

where $\mathbf{Q}(s)$ is the IMC controller, $\mathbf{M}(s)$ is the process model and the other terms are defined as follows:

$$\mathbf{Q}(s)=\mathbf{M}_{-}^{-1}(s)\mathbf{F}(s) \tag{2}$$

$$\mathbf{M}(s)=\mathbf{M}_{+}(s)\mathbf{M}_{-}(s) \tag{3}$$

$\mathbf{M}_{-}(s)$ is the portion of the process that needs to be inverted.

$\mathbf{M}_{+}(s)=\mathbf{M}(s)\mathbf{M}_{-}^{-1}(s)$; $\mathbf{M}_{+}(0)=\mathbf{I}$,

$\mathbf{F}(s)=$ is a diagonal matrix of filters in the form of $\mathrm{Diag}\left\{1/(\varepsilon_i s+1)^r\right\};\mathbf{F}(0)=\mathbf{I}$, where ε_i is the filter time constant corresponding to the *i* th output. Now (1) can be rewritten as:

$$\mathbf{C}(s)=s^{-1}\mathbf{f}(s) \tag{4}$$

where $\mathbf{f}(s)=s\mathbf{M}_-^{-1}(s)\mathbf{F}(s)\left[\mathbf{I}-\mathbf{M}_+(s)\mathbf{F}(s)\right]^{-1}$

$\mathbf{f}(s)$ can now be expanded in a Maclaurin series expansion form as shown below

$$\begin{aligned}\mathbf{C}(s)&=s^{-1}\left[\mathbf{f}(0)+s\mathbf{f}'(0)+s^2\mathbf{f}''(0)/2+...\right]\\&=\mathbf{K}+\mathbf{K}_I s^{-1}+\mathbf{K}_D s+...\end{aligned} \tag{5}$$

where

$$\mathbf{K}=\mathbf{f}'(0) \tag{6.a}$$

$$\mathbf{K}_I=\mathbf{f}(0) \tag{6.b}$$

$$\mathbf{K}_D=\mathbf{f}''(0)/2 \tag{6.c}$$

Now in order to calculate the PID controller parameters, as described in (6), let

$$\begin{aligned}\mathbf{D}(s)&=s^{-1}\left[\mathbf{I}-\mathbf{M}_+(s)\mathbf{F}(s)\right]\mathbf{F}^{-1}(s)\\&=s^{-1}\left[\mathbf{F}^{-1}(s)-\mathbf{M}_+(s)\right]\end{aligned} \tag{7}$$

In consideration of (4) we now have

$$\mathbf{f}(s)=\mathbf{M}_-^{-1}(s)\mathbf{D}^{-1}(s) \tag{8}$$

$$\mathbf{f}'(s)=\left[\mathbf{M}_-^{-1}(s)\right]'\mathbf{D}^{-1}(s)+\mathbf{M}_-^{-1}(s)\left[\mathbf{D}^{-1}(s)\right]' \tag{9}$$

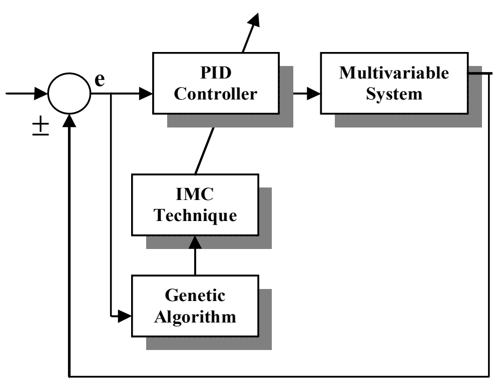

Fig. 1. Block diagram of the proposed controller.

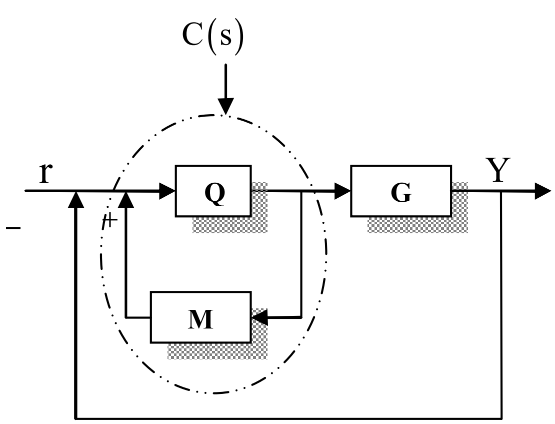

Fig.2. Block diagram of the IMC scheme.

$$\begin{aligned}\mathbf{f}''(s)=&\left[\mathbf{M}_-^{-1}(s)\right]''\mathbf{D}^{-1}(s)+2\left[\mathbf{M}_-^{-1}(s)\right]'\left[\mathbf{D}^{-1}(s)\right]'\\&+\mathbf{M}_-^{-1}(s)\left[\mathbf{D}^{-1}(s)\right]''\end{aligned} \tag{10}$$

$$\mathbf{D}(s)\big|_{s=0}=\left[\mathbf{F}^{-1}(s)\right]'-\mathbf{M}_+'(s)\big|_{s=0} \tag{11.a}$$

$$\mathbf{D}'(s)\big|_{s=0}=\left.\left\{\left[\mathbf{F}^{-1}(s)\right]''-\mathbf{M}_+''(s)\big|_{s=0}\right\}\right|_{s=0}/2 \tag{11.b}$$

$$\mathbf{D}''(s)\big|_{s=0}=\left.\left\{\left[\mathbf{F}^{-1}(s)\right]'''-\mathbf{M}_+'''(s)\big|_{s=0}\right\}\right|_{s=0}/3 \tag{11.c}$$

$$\left[\mathbf{D}^{-1}(s)\right]'\big|_{s=0}=-\mathbf{D}^{-1}(s)\mathbf{D}'(s)\mathbf{D}^{-1}(s)\big|_{s=0} \tag{11.d}$$

$$\begin{aligned}\left[\mathbf{D}^{-1}(s)\right]''\big|_{s=0}=&-\left[\mathbf{D}^{-1}(s)\right]'\mathbf{D}'(s)\mathbf{D}^{-1}(s)\\&-\mathbf{D}^{-1}(s)\mathbf{D}''(s)\mathbf{D}^{-1}(s)\\&-\mathbf{D}^{-1}(s)\mathbf{D}'(s)\left[\mathbf{D}^{-1}(s)\right]'\Big|_{s=0}\end{aligned} \tag{11.e}$$

$$\left[\mathbf{M}^{-1}(s)\right]'\big|_{s=0}=-\mathbf{M}_-^{-1}(s)\mathbf{M}_-'(s)\mathbf{M}_-^{-1}(s)\big|_{s=0} \tag{11.f}$$

$$\begin{aligned}\left[\mathbf{M}_-^{-1}(s)\right]''\big|_{s=0}=&-\left[\mathbf{M}_-^{-1}(s)\right]'\mathbf{M}_-'(s)\mathbf{M}_-^{-1}(s)\\&-\mathbf{M}_-^{-1}(s)\mathbf{M}_-''(s)\mathbf{M}_-^{-1}(s)\\&-\mathbf{M}_-^{-1}(s)\mathbf{M}_-'(s)\left[\mathbf{M}_-^{-1}(s)\right]'\Big|_{s=0}\end{aligned} \tag{11.g}$$

Also in consideration of (5), since the differentiation amplifies the noise excessively, the actual analog PID controller can be chosen as

$$\mathbf{C}(s)=\mathbf{K}+\mathbf{K}_I s^{-1}+\mathbf{K}_D\left(s/(\alpha s+1)\right) \tag{12}$$

where α is a constant, given as

$$\alpha=\max\left\{\left(\mathbf{K}^{-1}\mathbf{K}_D(i,j)\right)/20\right\}$$

So the PID controller parameters can now be obtained as

$$\mathbf{K}=\left[\mathbf{M}_-^{-1}(s)\right]'\mathbf{D}^{-1}(s)+\mathbf{M}_-^{-1}(s)\left[\mathbf{D}^{-1}(s)\right]'\Big|_{s=0} \tag{13}$$

$$\mathbf{K}_I=\mathbf{M}_-^{-1}(s)\mathbf{D}^{-1}(s)\big|_{s=0} \tag{14}$$

$$\begin{aligned}\mathbf{K}_D=&\left\{\left[\mathbf{M}_-^{-1}(s)\right]''\mathbf{D}^{-1}(s)+2\left[\mathbf{M}_-^{-1}(s)\right]'\left[\mathbf{D}^{-1}(s)\right]'\right.\\&\left.\left.+\mathbf{M}_-^{-1}(s)\left[\mathbf{D}^{-1}(s)\right]''\right\}\right|_{s=0}/2\end{aligned} \tag{15}$$

As it is clear, just by tuning the diagonal elements of matrix $\mathbf{F}(s)$, the PID controller parameters could be determined. The outputs of GA, for a 2 x 2 multivariable system, are ε_1 and ε_2 which are the elements of matrix $\mathbf{F}(s)$.

III. PID Controller Optimization By Genetic Algorithm

Standard genetic or genetic searching algorithms that are used for numerical parameter optimization, are based on the principle of evolutionary genetic and the natural selection process. A general genetic algorithm contains, usually, the three steps: selection, crossover and mutation. These steps are responsible for the "global" search minimization function without testing all the solutions.

Selection corresponds to keeping the best members of the population for the next generation to preserve the individuals with good performance (elite individuals) in fitness function. Crossover originates new members for the population, by a process of mixing genetic information from both parents. Depending on the selected parents, the growing of the fitness of the population is higher or lower. Among many other solutions, the parent selection can be done with the roulette method, tournament or random and elitist [10]. Mutation is a process by which a percentage of the genes are selected in a random fashion and are changed.

The genetic algorithm evaluates a population and generates a new one iteratively, with each successive population referred to as a generation. Given an initial population $P(0)$, the GA generates a new generation $P(t)$ based on the previous generation $P(t-1)$ as follows:

```
Initialize P(t) ⟹ P(0)    : P(t) Population at the time t
Evaluate P(0)
While ( not terminated-cindition) do
begin
   t ← t+1        :Increment generation select
   P(t) from P(t-1)
   recombine P(t)  : apply genetic operators
   (crossover, mutation)
   evaluate P(t)
end
end.
```

In this paper the GA is coded in MATLAB environment. The programs uses static values for maximum number of generations (maxgen=20), probability of crossover (pc=0.7), and probability of mutation (pm=0.05).

The initial population is randomly generated. Also the population size (psize=20), is selected. The fitness function that must be minimized in our case, can be considered as follows:

$$F(t) = \alpha_1 \int_{t_0}^{t_{final}} \|\mathbf{e}(t)\| dt + \alpha_2 \left(\sum_{i=1}^{n} \text{Over Shoot}(i) \right) \quad (21)$$

in which $\mathbf{e}(t) = [e_1(t) \ e_2(t)]$ is the tracking error vector. Overshoot (i) means the overshoot corresponding to the loop i, and α_1 and α_2 are the weights that regulate the effectiveness of each part of function. Defined fitness function is combined of two terms; the first term indicates the integral of tracking error, which by its minimization, the settling time and steady state error is minimized. The second term minimizes the overshoot of response.

IV. Simulation and Results

The performance evaluation of the proposed controller can be investigated by one example derived from [11], which represents a distillation column system (Fig. 3) .defined by the equation 22. Liquid mixture that is going to be processed is known as the feed. The feed tray divides the column into a top (enriching or rectification) section and a bottom (stripping) section. and $T_i(s)$ is the temperature of the ith tray. Heat is supplied to the reboiler to generate vapour and S(s) represents the steam valve to control the heat. The vapour moves up the column, and as it exits the top of the unit, it is cooled down by a condenser. The condensed liquid is stored in a holding vessel known as the reflux drum. Some of this liquid is then recycled back to the top of the column. This is called the reflux, where R(s) is the valve to control the reflux flow. In this paper the model relating to temperature tray $T_i(s)$ and control valves R(s) and S(s) is being used for simulation.

$$\begin{bmatrix} T_{17}(s) \\ T_4(s) \end{bmatrix} = \begin{bmatrix} \frac{-2.1e^{-s}}{8.5s+1} & \frac{1.2e^{-s}}{7.05s+1} \\ \frac{-2.75e^{-1.8s}}{8.25s+1} & \frac{4.28e^{-1.05s}}{9s+1} \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix} \quad (22)$$

T_4 and T_{17} denote the temperature on the 4th and 17th trays from the bottom of the column, respectively. We also choose $\mathbf{M}_+ = \text{diag}\{e^{-s} \ \ e^{-1.05s}\}$.

Simulation can be done for two cases. First, with Pc = 0.7 (Crossover Probability) and second, with Pc = 0.9. Fig. 4 shows the fitness values variation for these two cases, using genetic algorithm. The outputs of GA, after training process, for Pc = 0.7 are $\varepsilon_1 = 2.3$ and $\varepsilon_2 = 1.35$, and for Pc = 0.9 are $\varepsilon_1 = 2.18$ and $\varepsilon_2 = 1.26$. As it is clear from fig. 4, in case Pc = 0.9, the fitness value converges faster. Figs. 5- 8, show the system outputs for these two cases.

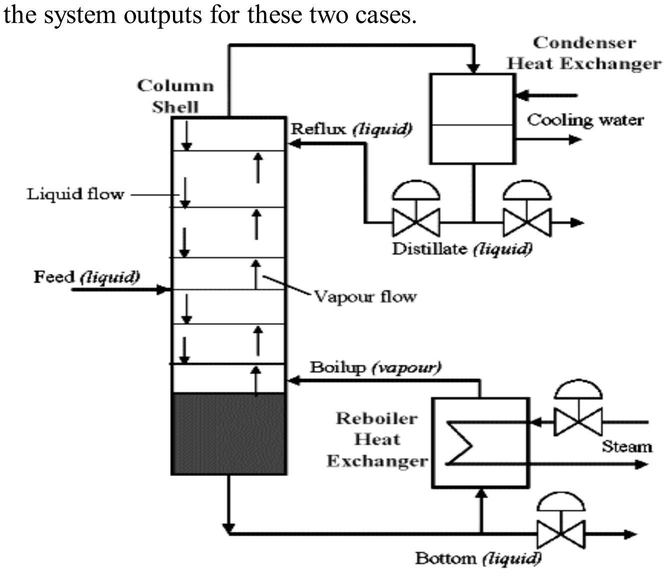

Fig. 3. Structure of a distillation column.

V. Conclusion

In this paper, Genetic algorithm in conjunction IMC technique are used for PID controller tuning, in multivariable applications. IMC technique reduces the number of parameters that must be tuned. Genetic algorithm provides successful parameter optimization for this purpose. It would optimally determines the IMC parameters. The performance of the proposed approach is evaluated by simulation of one application example.

REFERENCES

[1] R. Krohling and j. Rey, “Design of optimal disturbance rejection PID controllers using Genetic algorithms,” IEEE Trans. Evolutionary Computation. vol. 5, no. 1, Feb. 2001.

[2] T. F. Chwee, “Self-tuning PID controllers for dead time process,” IEEE Trans., vol. 35, no. **1,** pp. 119-125, 1988.

[3] C. Lin and H. Y Jan, “Multiobjective PID control for a linear brushless DC motor: an evolutionary approach,” *IEE Proc.-Power Appl.* , vol. 149, no. 6, pp. 397406, Nov. 2002.

[4] S. Sung and I.Lee, ”Limitations and countermeasures of PID Ind. Eng. Res,35, 2596-2610, 1996.

[5] Q. G. Wang and C. C. Hang, "Single-loop controller design via IMC principles,", Automatica, 37, pp.2041-2048, 2001.

[6] F. Zheng, Q. Wang, and T.H. Lee, ”On the design of multivariable PID controllers via LMI approach,” Automatica, 38, pp. 517–526, 2002.

[7] A. Isaksson and T. Hgglund, “Special issue on PID Control,” IEE Proc. Control Theory Appl., 2002, 149, (1), pp. 1–81.

[8] Q. Wang and C. Hang,"A frequency response approach to auto-tuning multivariable PID Controllers," in 13th Triennial World Congress, San Francisco, USA, 1996.

[9] C. B. Brosilow, "Incorporating process uncertainty and analytic design of PID controller into undergraduate process control," computers in engineering education, June, 1996.

[10] Z. Hímer," Fuzzy control of combustion with Genetic learning automata," in Proceedings 16th European Simulation Symposium, 2004.

[11] D. Seborg, “Process Dynamics and Control,” John Wiley, 1989.

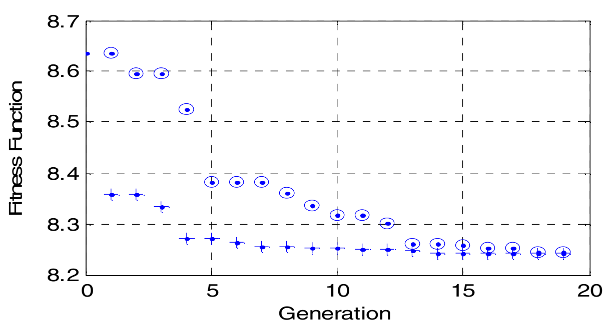

Fig. 4. Fitness value variations by the generations of genetic algorithm

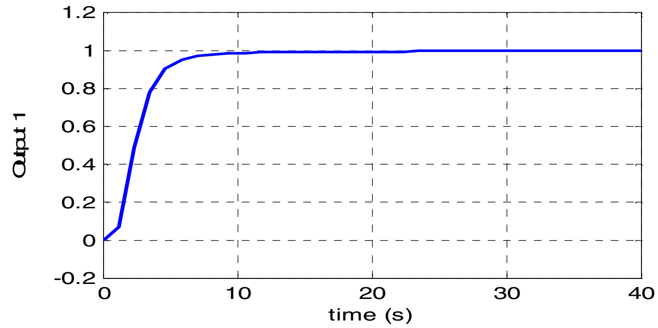

Fig. 5. Output 1 in response to unit step function, with Pc=0.7

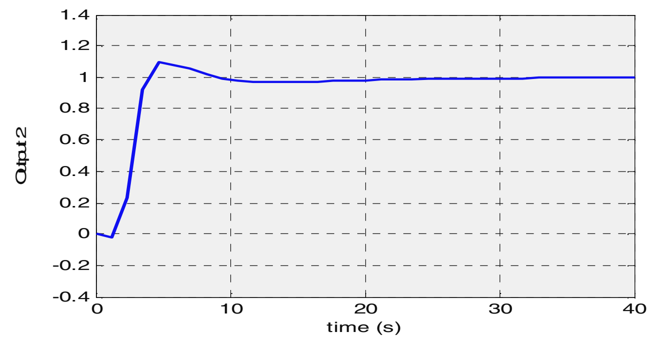

Fig. 6. Output 2 in response to unit step function, with Pc =0.7

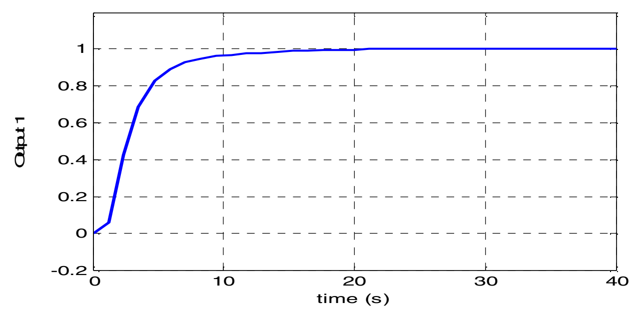

Fig. 7. Output 1 in response to unit step function, with Pc = 0.9

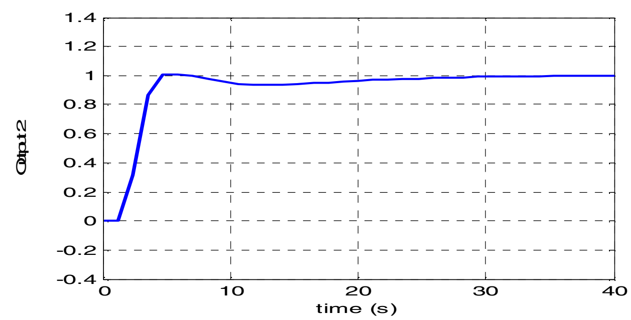

Fig. 8. Output 2 in response to unit step function, with Pc = 0.9