# TWO DEGREE OF FREEDOM PID CONTROLLER DESIGN USING GENETIC ALGORITHMS

DANIEL CZARKOWSKI†, TOM O'MAHONY‡

† Polish Register of Shipping, Electrical and Automation Department,
al. gen. J. Hallera 126, 80-416 Gdańsk, POLAND, d.czarkowski@prs.pl
‡ Cork Institute of Technology, Department of Electronic Engineering,
Advanced Control Group, Cork, IRELAND, tomahony@cit.ie

**Abstract.** This paper proposes three new robust controller design strategies for a 2-DOF PID controller. In each design, performance is optimised subject to constraints on robustness where in the first design this is achieved by specifying a desired minimum value for the gain and phase margins. In the second design the constraint is placed on the modulus margin and in the third case a desired upper bound on the maximum value of the input sensitivity function is considered. Each of these optimisation problems are solved using a canonical genetic algorithm. These three designs are comprehensively compared in simulation using a set of benchmark processes. The most consuming part of the GA optimsation is the evaluation of the fitness function, hence a novel solution is proposed that reduce the execution time by up to 60%.

**Key Words.** Genetic Algorithms, two degree of freedom PID controller, sensitivity functions, gain and phase margin.

## 1. INTRODUCTION

Since the early 1990's, Genetic Algorithms (GAs) have been used to determine PID controller parameters. Initially, simple cost functions, e.g. integral of squared error, were specified [10, 16, 17, 18]. More recently the versatility of the GA approach has been exploited and more complex cost functions incorporating robustness considerations have been proposed. In general, the robustness issue has been addressed through the use of (i) gain and phase margin criteria and (ii) a specification on a closed-loop sensitivity function.

Relatively few GA–based designs for PID controllers using gain and phase margins exist, and those that have been proposed, e.g. in [11] use simple PID controller structures and relatively simple cost functions. In this paper, a more useful PID algorithm, based on a two degree-of-freedom (2-DOF) structure is studied and the cost function incorporates metrics that measure servo performance, regulator performance as well as constraints on both the gain and phase margin. As such, this design is considered to be realistic and applicable to the general automation industry. In contrast to the use of gain and phase margin criteria a number of useful GA-based approaches using sensitivity functions have been proposed, e.g. [4, 6, 7, 9, 12, 15, 19, 21, 22], amongst others. Many of these approaches are broadly similar and involve solving a mixed $H_2/H_\infty$ problem where performance is specified as a 2-norm and robustness by an $\infty$-norm on some suitably chosen sensitivity function. In this paper an additional two cost functions will be considered. Like the gain and phase margin design these will both include metrics to assess the servo and regulator performance and robustness will be achieved by constraints on (i) the modulus margin, *MM,* as defined in [13] and (ii) the input sensitivity function. While *MM* being the inverse of the peak value of the sensitivity function, is a common and widely applied approach to controller design the application of the input sensitivity function is atypical but, in the authors' opinion, very practical.

The contribution of this paper is then to (i) present a number of novel controller design approaches and (ii) to compare the performance of these designs and determine whether any of these objective functions offer any real benefit to the automation industry by optimizing process units to, for example, maximize throughput, minimize waste, etc. The performance of these objective functions will be evaluated in simulation over a range of classical and a-typical transfer

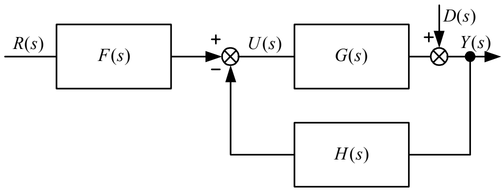


*Fig. 1 Block diagram of 2-DOF PID controller*

functions [1, 2] that are representative of the process industry. From this simulation study general conclusions will be drawn.

## 2. CONTROLLER DESCRIPTION

The ideal PID controller is given by equation (1)

$$U(s) = E(s)C(s) \tag{1}$$

where *U(s)* is the control signal, the *E(s)* is the control error and *C(s)* is defined by

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{2}$$

$K_p$, $K_i$ and $K_d$ are the controller parameters. Equation (2) is known as a single degree of freedom control law because the controller parameters may be optimised for servo performance or regulator performance but not both. Most industrial controllers are considerably more complex than the "textbook" control law of equation (2); a typical representation is given by equation (3).

$$U(s) = K_p\left(bR(s) - Y(s)\right) + \frac{K_i}{s}\left(R(s) - Y(s)\right) + \\ + \frac{sK_d\left(cR(s) - Y(s)\right)}{1 + \dfrac{sK_d}{K_p N}} \tag{3}$$

In equation (3), *R(s)* is the set-point, *Y(s)* is the process output, $N$ is the derivative filter, $b$ and $c$ are weightings that influence the set-point response. This equation may be restructured as

$$U(s) = R(s)F(s) - Y(s)H(s) \tag{4}$$

where $F(s)$ and $H(s)$ are defined by equations (5) and (6) respectively. This representation is illustrated in figure 1, where $G(s)$ is the transfer function representing the plant.

$$F(s) = K_p b + \frac{K_i}{s} + \frac{sK_d}{1 + \dfrac{sK_d}{K_p N}} c \tag{5}$$

$$H(s) = K_p + \frac{K_i}{s} + \frac{sK_d}{1 + \dfrac{sK_d}{K_p N}} \tag{6}$$

Clearly, the industrial PID structure corresponds to a 2-DOF control law, where the transfer function $H(s)$ may be chosen to yield optimal regulator performance while the transfer function $F(s)$ can be chosen to yield good servo performance. Note that $K_p$, $K_i$, $K_d$ and $N$ appear in both of these transfer functions. Thus, the 2-DOF PID controller is not a true 2-DOF controller. However, the parameters $b$ and $c$ may be utilised to enhance the servo performance and clearly do not affect the regulator performance. For the ideal PID controller $F(s)$ and $H(s)$ are equal, given by equation (1).

## 3. CONTROLLER DESIGN

The controller design problem consists of choosing suitable values for the controller coefficients $K_p$, $K_i$, $K_d$, $N$, $b$ and $c$. In this paper an optimization approach is adopted where these coefficients are determined by minimising the integral of the absolute value of the combined servo plus regulator error subject to robustness constraints. Three different types of robustness constraints are considered. In the first design the standard gain margin, $A_m$, and phase margin, $\phi_m$ are employed and the following constraints specified $A_m > 2$; $\phi_m > 45°$. The second design considers placing a constraint on the modulus margin. The Modulus Margin, *MM*, is defined in [13] as the radius of the circle centred on the critical point ($-1$, $i0$) and tangent to the Nyquist plot of the open-loop transfer function, $L(s)$. The inverse of the *MM* corresponds to the peak amplitude value of the sensitivity function $S(j\omega) = 1/(1 + L(j\omega))$ which is commonly used in, for example, $H_\infty$ control. Recommended practical value for the modulus margin is $MM \geq 0.5$ ($-6dB$) and this constraint was adopted in the second design. In the final design an alternative sensitivity function, called the input sensitivity function, denoted by $\mathcal{U}(s)$, and defined by

$$\mathcal{U}(s) = \frac{U(s)}{D(s)} = \frac{H(s)}{1 + G(s)H(s)} = \frac{H(s)}{1 + L(s)} \tag{7}$$

using the notation of figure 1. As the name suggests, this function defines the sensitivity of the input signal, $U(s)$, to deterministic disturbances, $D(s)$. The function may also be used to directly penalise the high-frequency gain of the feedback controller $H(s)$ and thus reduce the impact of high-frequency measurement noise on the closed-loop system. Since model-uncertainty tends to be predominantly high-frequency in nature, a suitably shaped $\mathcal{U}(j\omega)$ will also enhance the robustness of the closed-loop system. In common with the sensitivity function, $S(j\omega)$, the philosophy adopted with $\mathcal{U}(j\omega)$ was to constrain the peak value of this function. However, unlike $S(j\omega)$, recommendations for a suitable peak value do not exist. The problem was circumvented by examining the results obtained from the other two designs and using this knowledge to chose a suitable upper bound. In all three cases the objective function was minimised using a Genetic Algorithm (GA).

### 3.1 Genetic Algorithm

The GA approach is an intuitive and mature search and optimisation technique based on the principles of natural evolution and population genetics. Typically the GA starts with little or no knowledge of the correct solution and depends entirely on responses from an interacting environment and its evolution operators to arrive at good solutions. By dealing with several independent points, the GA samples the search space in parallel and hence is less susceptible to converging to a suboptimal solution. In this way the GA has been shown to be capable of locating high performance areas in complex domains without experiencing the difficulties associated with false optima, as may occur with gradient descent techniques. Thus the GA has been recognised as a powerful tool in many control applications.

The structure of the GA was broadly similar to that described in [5] and is popularly known as the simple or canonical GA. The controller parameters were encoded using Gray coding and an initial, randomly generated population of 100 individuals was used. At each iteration parents were probabilistically selected using stochastic universal sampling, while offsprings were generated using single point crossover (probability = 1) and a single bit mutation (probability = 0.0143). The algorithm terminated after 100 generations. Six parameters $K_p$, $K_i$, $K_d$, $b$, $c$ were tuned. Each of the gains was initially optimised over the range 0–20 and represented as 10 bit segments with a resolution of 0.01. This upper bound was carefully monitored and if the GA returned parameters close to the bound it was then increased. However, in accordance with the PID controller structure, the parameters $b$ and $c$ have a fixed range (0–1) and the remaining parameter $N$ was tuned over the range 0–31 and represented with 5 bits.

### 3.2 Objective functions

The critical component of any optimisation is the design of the objective function to be minimised. As mentioned previously, the objective function employed here incorporates robustness via constrains on 1) the gain and phase margins, 2) the modulus margin, and 3) the input sensitivity function. These constraints were implemented as follows. First consider the standard Gaussian function

$$y = e^{x} \tag{8}$$

If this function is modified by appropriate scaling (trial and error) it is possible to generate curves such as illustrated in figure 2. If the gain margin is less than 6*(dB)* then the contribution by this function to the overall cost is large, hence the cost function will be heavily weighted, regardless of the IAE; while if the gain margin is greater than 6*(dB)* the contribution is constant at unity. A similar idea was applied to the phase margin, the modulus margin and the maximum peak on the input sensitivity function. Mathematically these curves are represented by equations (9)-(12)

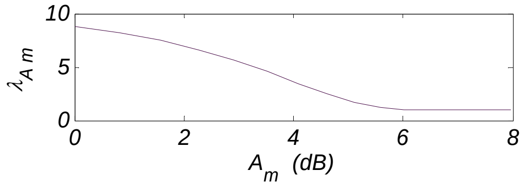


*Fig. 2 Constraining factor*

$$\lambda_{Am} = \begin{cases} 10 - 9e^{-2(\underline{Am} - Am)^2} & \forall \quad A_m < \underline{A_m} \\ 1 & \forall \quad A_m \geq \underline{A_m} \end{cases} \tag{9}$$

$$\lambda_{\phi m} = \begin{cases} 10 - 9e^{-0.0025(\underline{\phi m} - \phi m)^2} & \forall \quad \phi_m < \underline{\phi_m} \\ 1 & \forall \quad \phi_m \geq \underline{\phi_m} \end{cases} \tag{10}$$

$$\lambda_{MM} = \begin{cases} 10 - 9e^{-20(\underline{MM} - MM)^2} & \forall \quad MM < \underline{MM} \\ 1 & \forall \quad MM \geq MM \end{cases} \tag{11}$$

$$\lambda_{Mu} = \begin{cases} 10 - 9e^{-0.0333(\overline{Mu} - Mu)^2} & \forall \quad M_u > \overline{M_u} \\ 1 & \forall \quad M_u \leq \overline{M_u} \end{cases} \tag{12}$$

where $A_m$ is the measured gain margin, $\underline{A_m}$ is the specified minimum value, both expressed as absolute values, $\phi_m$ is the measured phase margin (in degrees) and $\underline{\phi_m}$ is the specified minimum value. *MM* represents measured modulus margin and $\underline{MM}$ a user specified minimum value, while the maximum peak on the input sensitivity function is denoted by $M_u$ and the specific maximum value is $\overline{M_u}$ .

The IAE of the servo and regulatory response is calculated using equation (13)

$$IAE = \sum_{k=0}^{t_1} |e(k)| + \sum_{k=t_1}^{t_2} |e(k)| \tag{13}$$

where *e(k)* is the control error and $t_1$ is time at which the disturbance, *D(s)*, is applied. Therefore the IAE servo is calculated over the period $0 < t < t_1$ , and the regulator is calculated over the period $t_1 \leq t \leq t_2$ . The various objective functions are then

$$J_{Am} = \min_{K_p, K_i, K_d, b, c, N} \{ IAE \cdot \lambda_{Am} \cdot \lambda_{\phi m} \} \tag{14}$$

$$J_{MM} = \min_{K_p, K_i, K_d, b, c, N} \{ IAE \cdot \lambda_{MM} \} \tag{15}$$

$$J_{Mu} = \min_{Kp, Ki, Kd, b, c, N} \{ IAE \cdot \lambda_{Mu} \} \tag{16}$$

Clearly, if the robustness criteria are satisfied $\lambda_{Am} = 1$ , $\lambda_{\phi m} = 1$ , $\lambda_{MM} = 1$ and $\lambda_{Mu} = 1$ equations (14)-(16) reduce to minimising the IAE. The criteria may then be interpreted as optimising performance subject to constraints on robustness.

### 3.3 GA with a look up table

The GA of section 3.1 was implemented in the MATLAB environment and the objective functions of section 3.2 were evaluated using MATLAB and Simulink. MATLAB was used to calculate the gain and phase margins, *MM* and $M_u$. If the process transfer function contained a time-delay the time-delay was replaced with a Padé approximation and this approximation was used in the calculation of *MM* and $M_u$. For accuracy reasons it was decided that Simulink would be used to generate the closed-loop signals *y(t), e(t)* as this avoided the problem of approximating time-delay. The cost, however, was increased computational time which varied significantly depending on the nature of the process under consideration.

A solution to speed the GA up is to implement a look up table. The most time consuming part is the evaluation of the fitness function, so a faster GA implies that the objective function must be executed quicker. As was mentioned before GAs are stochastic search techniques and a fitness function with the same controller parameters may be evaluated a number times in each population. The number of unnecessary computations increase with each generation as the algorithm converges. To avoid this, a matrix (look up table) is generated, where previously computed GA chromosomes are stored with the equivalent value of the objective function. The algorithm checks whether the fitness function was previously evaluated and, if so, the objective function is not executed; instead the solution is passed through. A disadvantage of this method is that the matrix grows with each member of the population that is evaluated. However, this is not a serious restriction as the MATLAB *find* function is relatively fast and this solution provides a faster off-line optimisation.

Consider controller parameter tuning problem of transfer function with time-delay, $G_2(s)$, for the second deisgn using the AMD Athlon 650Mhz processor. Without the look up table the typical optimisation takes 30 minutes and the population of 50 individuals requires, on average, 23.6*(sec)* to be evaluated. The time may vary depending on the value of $K_d$. If the look up table is implemented, individuals require approximately 13.5*(sec)* to be evaluated and the optimisation is performed in 20 minutes. This performance improvement is illustrated in figure 3, where the x-axis represents the number of generations and the y-axis shows the time which is required to evaluate a single generation. The dashed plot presents the GA solution without the look up table, while the solid line corresponds to the case where the look up table is used. Note that the initial population has been evaluated and this data has already been stored in the matrix. In this case 10% of the population is passed to another generation unchanged and accounts for the initial difference between algorithms. In this example, the use of the look-up table results in an average 60% reduction in the evaluation time even though the dimension of the final matrix was 2342 rows and 7 columns (6 tuning parameters and the objective function evaluation). The combination of the GA with a look up table significantly reduces the computation time required to tune the more complicated controllers such as the two degree of freedom PID controller.

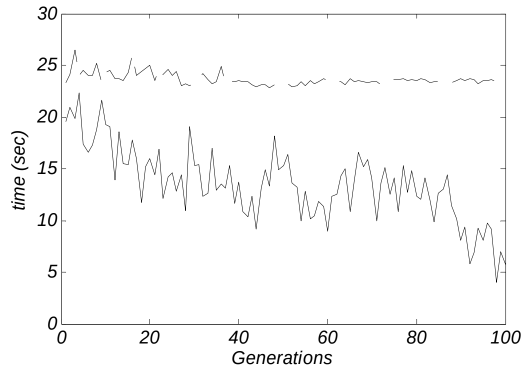


*Fig. 3 Time taken to evaluate the GA population; solid line using a look-up table, dashed line excluding the look-up table*

### 4. SIMULATION RESULTS

The 2-DOF PID was evaluated by implementing the controller design equations, (14)-(16) to the list of process models (17)-(27). These benchmark systems were proposed by Åström and Hägglund [1, 2] and are representative of typical industrial process transfer functions.

$$G_1(s) = \frac{1}{(s+1)^3} \tag{17}$$

$$G_2(s) = \frac{1}{(s+1)^3}e^{-15s} \tag{18}$$

$$G_3(s) = \frac{1}{(s+1)(1+0.2s)(1+0.04s)(1+0.008s)} \tag{19}$$

$$G_4(s) = \frac{1}{s(s+1)^2} \tag{20}$$

$$G_5(s) = \frac{1-2s}{(s+1)^3} \tag{21}$$

$$G_6(s) = \frac{9}{(s+1)(s^2+2s+9)} \tag{22}$$

$$G_7(s) = e^{-s} \tag{23}$$

$$G_8(s) = \frac{e^{-s}}{s} \tag{24}$$

$$G_9(s) = \frac{100}{(s+10)^2}\left(\frac{1}{s+1} + \frac{0.5}{s+0.05}\right) \tag{25}$$

$$G_{10}(s) = \frac{150}{(s+10)^2(s+1)} \tag{26}$$

$$G_{11}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)} \tag{27}$$

| | 1) $\underline{A_m} = 6dB$ $\underline{\varphi_m} = 45°$ | | | | | 2) $\underline{A_m} = 14dB$ $\underline{\varphi_m} = 45°$ | | | | | 3) $\underline{MM} = 6$ | | | | | 4) $\overline{M_u} = \text{var}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *IAE* | $A_m$ | $\varphi_m$ | *MM* | $M_u$ | *IAE* | $A_m$ | $\varphi_m$ | *MM* | $M_u$ | *IAE* | $A_m$ | $\varphi_m$ | *MM* | $M_u$ | *IAE* | $A_m$ | $\varphi_m$ | *MM* | $M_u$ |
| $G_1(s)$ | 1.633 | 14.66 | 45 | 0.56 | 30 | 1.647 | 14.59 | 45 | 0.56 | 32 | 1.707 | 20.61 | 45 | 0.60 | 37 | 2.351 | 11.63 | 42 | 0.52 | 18 |
| $G_2(s)$ | 40.25 | 5.72 | 62 | 0.50 | 45 | 51.43 | 9.84 | 63 | 0.69 | 34 | 41.38 | 7.15 | 62 | 0.59 | 41 | 39.53 | 1.03 | 65 | 0.13 | 34 |
| $G_3(s)$ | 0.2688 | 15.59 | 45 | 0.61 | 49 | 0.2608 | 17.03 | 45 | 0.63 | 51 | 0.3369 | 12.90 | 50 | 0.60 | 28 | 0.6563 | 9.29 | 23 | 0.35 | 26 |
| $G_4(s)$ | 3.156 | 14.67 | 30 | 0.44 | 39 | 4.887 | 14.77 | 45 | 0.55 | 29 | 3.997 | 22.24 | 41 | 0.59 | 32 | 3.406 | 5.18 | 16 | 0.25 | 25 |
| $G_5(s)$ | 9.713 | 5.88 | 55 | 0.48 | 22 | 13.09 | 14.03 | 57 | 0.62 | 13 | 10.79 | 8.15 | 58 | 0.60 | 23 | 10.12 | 2.73 | 45 | 0.27 | 16 |
| $G_6(s)$ | 1.147 | 13.32 | 45 | 0.46 | 28 | 1.15 | 14.21 | 45 | 0.47 | 30 | 1.376 | 17.03 | 66 | 0.60 | 26 | 1.49 | 6.48 | 42 | 0.38 | 17 |
| $G_7(s)$ | 3.298 | 8.31 | 58 | 0.66 | 9.1 | 3.116 | 8.02 | 59 | 0.67 | 10 | 2.837 | 5.55 | 60 | 0.61 | 20 | 3.017 | 6.21 | 57 | 0.64 | 9 |
| $G_8(s)$ | 4.717 | 3.70 | 39 | 0.48 | 47 | 8.899 | 5.14 | 60 | 0.51 | 38 | 5.144 | 5.51 | 33 | 0.60 | 58 | 4.043 | 2.08 | 27 | 0.36 | 37 |
| $G_9(s)$ | 0.2584 | ∞ | 74 | 0.99 | 30 | 0.2563 | ∞ | 107 | 1.00 | 39 | 0.1225 | ∞ | 96 | 1.00 | 38 | 0.1039 | ∞ | 42 | 0.66 | 18 |
| $G_{10}(s)$ | 0.3285 | 25.54 | 45 | 0.61 | 51 | 0.2967 | 14.33 | 45 | 0.57 | 34 | 0.2941 | 19.53 | 45 | 0.60 | 42 | 0.4707 | 7.30 | 26 | 0.36 | 24 |
| $G_{11}(s)$ | 0.6125 | ∞ | 67 | 0.94 | 64 | 0.5877 | ∞ | 48 | 0.81 | 48 | 0.8188 | -20.13 | 35 | 0.60 | 36 | 0.9228 | -13.66 | 65 | 0.92 | 55 |

*Tab. 1 The optimised IAE with robustness constraints*

The control signal was constrained to the range $-10 \leq u(k) \leq 10$. Robustness was achieved by considering requirements for the gain and phase margin. In the literature recommendations for an adequate gain margin vary from $A_m$ = 4.6 – 14*dB's* [8, 20, 23 and 3 on p. 126] while suggested practical values for the phase margin are $\varphi_m = 30° - 60°$. In [3, 13] it is suggested that the modulus margin lie between 0.5 (aggressive tuning) and 0.77 (robust tuning). In contrast, the author's are not aware of any specific recommendations for the input sensitivity function, other than the general requirement that the peak value be some suitably small number. Based on values returned from the three previous designs the smallest $M_u$ was taken into consideration, hence the value varied depending on the model

The simulation results are presented in table 1. Clearly, the design was successful in that the specified minimum gain and phase margins $\underline{A_m} = 6dB$ and $\underline{\varphi_m} = 45°$ were achieved for most models, the exceptions being $G_8(s)$ ($A_m$ = 3.7dB) and $G_4(s)$ ($\varphi_m = 30°$). The second design ($\underline{A_m} = 14dB$ and $\underline{\varphi_m} = 45°$) was not successful for $G_2(s)$, $G_7(s)$ and $G_8(s)$. However, the design specifications could be achieved if the penality function weighting was increased, i.e. $\lambda_{Am}$ of equation (14) was multiplied by 10. The third design ($\underline{MM} = 0.6$) was achieved for all models and the fourth design ($\overline{M_u} = \text{var}$) as well.

A selection of closed-loop responses are illustrated in figures 4 and 5 where the solid line represents the response of the first design, the dotted line of the second, the dashed line of the third and the fourth design is shown by dashdot line. In each of these figures the upper plot illustrates the closed-loop response to a unit step input occurring at $t$=0 and to a unit step disturbance. In figure 4 the disturbance was applied at $t$=10*(sec)* while in figure 5 the disturbance occurs at time $t$=15*(sec)*. Also shown are the corresponding control signals.

Consider the first model, $G_1(s)$, $M_u$ varies from 30 for the first design to 37 for the third one and the most sluggish response is for the fourth design ($M_u$=18). This design gives a reasonably fast response and the amount of energy required is smaller than for other designs. The closed-loop system with model $G_9(s)$ gives the fastest response using this design. The proposed method is very effective. However, a difficulty remains with selecting a desired maximum value of $M_u$.

In general, the gain margin is expressed as a positive number (in dB), nevertheless, there are two exceptions. Model $G_{11}(s)$, tuned using the sensitivity function approach, returned a *gain reduction margin* or *downward margin*, [14]. Therefore, it is possible that the gain margin, expressed in *dB*, could be negative and the closed-loop system remains stable.

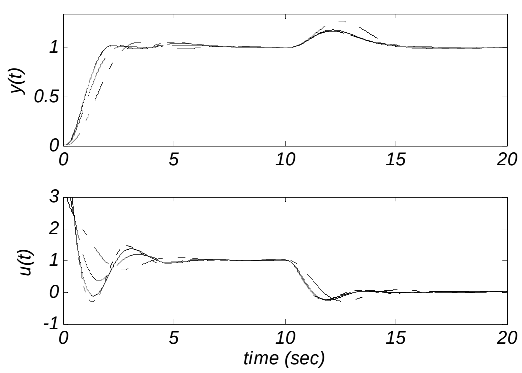


*Fig. 4 Set–point following for $G_1(s)$*

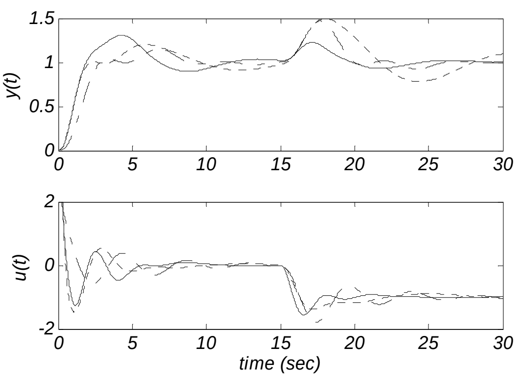


*Fig. 5 Set–point following for $G_4(s)$*

## 5. CONCLUSIONS

From figures 4, 5 and the results listed in table 1 it is evident that designs proposed in this paper are very effective. Despite the wide range of process dynamics: stable, inverse unstable, integrating, long time-delay, stable and consistent closed-loop performance was obtained for all of the systems examined and systems remained stable. If the integral of absolute error criteria are compared, it may be conclude that none of the proposed methods is significantly better. However, even though the responses from the third design are not as fast as from the first design, it can be summarised that this design gives slightly better results than the other counterparts.

This work, while providing some insight into the relative merits of the controllers is limited by the fact that perfect modelling was assumed. It is anticipated that if model uncertainty was incorporated and robust performance considered the results might vary.

## REFERENCES


1. Åström K. J., Panagopoulos H., and Hägglund T.: Design of PI controllers based on non-convex optimization, Automatica, vol. 34, no. 5, 1998, pp. 585–601.
2. Åström K. J. and Hägglund T.: Benchmark systems for PID control, in IFAC Workshop on Digital Control: Past Present and Future of PID Control, (Terrassa, Spain), 2000, pp. 165–166.
3. Åström K. J. and Hägglund T.: PID controllers: theory, design, and tuning, Instrument Society of America, 2nd ed., 1995.
4. Chen B.-S., Cheng Y.-M. and Lee C.-H.: A genetic approach to mixed $H_2/H_\infty$ optimal PID control, IEEE Control Systems Magazine, vol. 15, no. 5, 1995, pp. 51–60.
5. Chipperfield A., Fleming P., Pohlheim H. and Fonseca C.: Genetic Algorithm Toolbox user's guide, Department of Automatic Control and Systems Engineering, (University of Sheffield), 1.2 ed., 1994.
6. Czarkowski D. and O'Mahony T.: Intelligent controller design based on gain and phase margin specifications, in Irish Signals and Systems Conference, (Belfast, Northern Ireland, Queen's University), 2004, pp. 381–386.
7. Herreros A., Baeyens E., and Perán J. R.: MRCD: a genetic algorithm for multiobjective robust control design. Engineering Applications of Artificial Intelligence, vol. 15, no. 3–4, 2002, pp. 285–301.
8. Ho W. K., Hang C. C. and Cao L. S.: Tuning of PID controllers based on gain and phase margin specifications, Automatica, vol. 31, no. 3, 1995, pp. 497–502.
9. Ho W., Lim K. and Xu W.: Optimal gain and phase margin tuning for PID controllers, Automatica, vol. 34, no. 8, 1998, pp. 1009–1014.
10. Hwang W. and Thompson W.: An intelligent controller design based on genetic algorithms, in Proceedings of the 32nd IEEE Conference on Decision and Control, 1993, pp. 1266–1267.
11. Jones A., Ajlouni N. and Uzam M.: Online frequency domain identification and genetic tuning of PID controllers, in IEEE Conference on Emerging Technologies and Factory Automation, vol. 1, 1996, pp. 261–266.
12. Krohling R. and Rey J.: Design of optimal disturbance rejection PID controllers using genetic algorithms, IEEE Transactions on Evolutionary Computation, vol. 5, no. 1, 2001, pp. 78–82.
13. Landau I., Lozano R. and M'Saad M.: Adaptive control, Springer–Verlag London Limited, 1998.
14. Lee C.-H. and Teng C.-C.: Tuning of PID controllers for unstable process based on gain and phase margin specifications: a fuzzy neural approach, Fuzzy Sets and Systems, vol. 128, 2002, pp. 95–106.
15. Lieslehto J.: PID controller tuning using evolutionary programming, in American Control Conference, vol. 4, 2001, pp. 2828–2833.
16. Mitsukura Y., Yamamoto T. and Kaneda M.: A design of self-tuning PID controllers using a genetic algorithm, In American Control Conference, vol. 2, 1999, pp. 1361–1365.
17. O'Mahony T., Downing C. and Fatla K.: Genetic algorithms for PID parameter optimisation: minimising error criteria, in Proceedings of the Process Control and Instrumentation, (Glasgow, Scotland), 2000, pp. 148–153.
18. Porter B. and Jones A.: Genetic tuning of digital PID controllers, Electronics letters, vol. 28, no. 9, 1992, pp. 843–844.
19. Poulin E. and Pomerleau A.: Unified PID design method based on a maximum peak resonance specification, IEE Proceedings, Control Theory & Applications, vol. 144, no. 6, 1997, pp. 566–574.
20. Seborg D., Edgar T. and Mellichamp D.: Process Dynamics and Control, New York: John Wiley & Sons, 1989.
21. Shen, J.-C.: New tuning method for PID controller, in Conference on Control Applications, (Mexico City), 2001, pp. 459–464.
22. Shinskey F. G.: How good are our controllers in absolute performance and robustness?, Measurement and Control, vol. 23, 1990, pp. 114–121.
23. Wang Q.-G., Fung H.-W. and Zhang Y.: PID tuning with exact gain and phase margin, ISA Transactions, vol. 38, 1999, pp. 243–249.