Tuning of Robust PID Controller with Filter for SISO System Using Evolutionary Algorithms

RATHINAM Muniraj[1]*, MARIA SILUVAIRAJ Willjuice Iruthayarajan[2], RAMAVEERAPATHIRAN Arun[3]

[1] *(*Corresponding author)* Assistant Professor (Senior Grade), Department of EEE, National Engineering College, K.R.Nagar, Kovilpatti – 628 503, Tuticorin District, Tamil Nadu, India.
e-mail: munirajphd@gmail.com

[2]Professor & Head of the Department, Department of EEE, National Engineering College, K.R.Nagar, Kovilpatti – 628 503, Tuticorin District, Tamil Nadu, India.
e-mail:m.willjuice@gmail.com

[3]Associate Professor, Department of EIE, National Engineering College, K.R.Nagar, Kovilpatti – 628 503, Tuticorin District, Tamil Nadu, India.
e-mail: arun.pathiran@gmail.com

Abstract: This paper discusses the performance comparison of Evolutionary Algorithm techniques with a view to tuning the desired design parameters of robust PID controller with filter. The design parameters of robust PID controller such as proportional gain K_p, integral time constant T_i, derivative time constant T_d and filter time constant t_d/N for Single Input Single Output (SISO) system are computed using Real coded Genetic Algorithm (RGA), Differential Evolution (DE) algorithm and Particle Swarm Optimization (PSO) algorithm. The design parameters are optimized using the statistical measures in twenty independent simulation runs. The computed design parameters obtained using Evolutionary Algorithms are used to determine the performance specifications of the Robust PID controller. The performance specifications determined are robustness with respect to model uncertainties and disturbance attenuation, set point tracking, load disturbance rejection and control energy. The test systems used to ascertain the performance specifications are Phase Locked Loop (PLL) system with motor control and Magnetic Levitation system (MLS). For PLL system, the output response obtained using the computed design parameters by RGA algorithm has proved to be better than DE and PSO. For MLS system, the output response obtained using the computed design parameters by DE algorithm has been better than PSO and RGA.

Keywords: Single Input Single Output (SISO) system; Real coded Genetic Algorithm (RGA); Differential Evolution (DE); Particle Swarm Optimization (PSO).

1. Introduction

Most of the industrial controllers are still designed and utilized depending on PID control algorithms. The desirable features of PID controllers are their amiability, clear functionality, applicability and ease of use. The PID controllers perform several important functions like elimination of steady state offset through integral action and anticipation of deviation and generation of corrective signals through derivative action [28]. Most of the real world problems are vulnerable to external disturbances and the robustness property of PID controller provides an efficient means of solution. Robustness of a PID controller is defined as the system's ability to maintain its functionalities under conditions of varying internal or external parameters [9].

Minimum parameter tuning for control processes is a desirable feature of the PID controller. The three parameters to be tuned are K_p, K_i and K_d. In most cases PID controllers are used as PI controllers by switching off the derivative action. The usefulness of the robust PID controller to meet out the performance specifications of the given system mainly depends on the tuning of the controller .Since the system has nonlinear plant and various uncertainties the tuning of the robust PID controller becomes difficult. This leads to poor tuning. The reasons of the poor tuning is due to a) lack of knowledge among operators and commissioning personnel b) variety of PID structures that leads to error during tuning c) in some cases the nature of tuning method do not meet the requirement of the process involved [3]. Out of the installed PID controllers in industry, nearly one third controllers are tuned manually and remaining two third are tuned automatically. But the automatically tuned PID controllers are not effective [7].To overcome these limitations optimization algorithms are used for the tuning of robust PID controller. In the past many researchers have used different optimization algorithm for tuning the parameters of robust PID controller. But the parameter tuned to meet out the performance specifications of the system is different in each case. Many researchers have considered set

https://doi.org/10.24846/v26i3y201703

point tracking and load disturbance rejection has the performance specifications. Few have also taken in to account the robustness against model uncertainty has performance specification of the system [18].

Amal Moharam et al in his work proposed a hybrid DE-PSO algorithm based robust PID controller [19].Jimenez et al suggested a novel method of auto tuning of PID controller using genetic algorithm [13]. Yang et al tuned the controller parameters using cuckoo search algorithm [29]. Kiam et al modelled a PID control system and tuned the parameters of the control system using intelligent control techniques [14]. Sugie et al designed a fixed structure controller with H∞ norm and tuned the deign parameters using constrained PSO algorithm [24]. Chen et al designed a PID controller structure with mixed H_2 / H∞ norm and used GA algorithm to tune the design parameters for a SISO system [6].

Hultmann et al proposed a multi objective algorithm for tuning of PID controller [10]. Zhao et al tuned a robust PID controller using multi objective PSO [30]. Krohling et al designed a GA based optimal disturbance rejection PID controller for a servo motor system [16]. Kristiansson et al. proposed a structured PID controller for different plants and tuning method to find out the performance parameters [15]. Alberto Herreros et al presented a PID controller for a multi objective problem [2]. Qi Bing-Jin et al. tuned the PID controller using improved cuckoo search algorithm and observed that the overshoot and the transient time much smaller when compared to Ziegler–Nichols tuned PID controller [20]. Huang et al. suggested a robust PID controller for nonlinear system which is capable of self-tuning [25].

In this paper, Evolutionary algorithm is used to determine the optimal parameters for the robust optimal controller. Evolutionary Algorithms used are Particle Swarm Optimization (PSO), Real coded Genetic Algorithm (RGA) and Differential Evolution (DE). The main focus of this paper is to determine the design parameters of robust PID controller using evolutionary algorithm such that it meet out the system performance specifications like robustness with respect to model uncertainties and disturbance attenuation, set point tracking, load disturbance rejection and control energy. The performance of the tuned controller using evolutionary algorithm is tested on PLL and MLS system.

The main contribution in this paper is the formation of objective function by considering the robustness to model uncertainty, load disturbance rejection, set point tracking and control energy as performance specifications. Each performance specification is treated as a separate individual single objective function and a collective measure of all the performance specifications is also considered.

The other noted contribution of this paper is the inclusion of a derivative filter in the robust PID controller structure. The addition of this filter alters the position of zeros in the controller such that the stability of the system is enhanced [1].

The remaining part of the paper is organized as follows: Section 2 introduces the Robust PID controller structure. Section 3 describes the formulation of objective function which meets out the performance specifications such as. Section 4 portrays the implementation of the different Evolutionary algorithms for the problem ascertained. Section 5 details about the Test system used to evaluate the performance of the tuned robust PID controller. Section 6 gives a detailed view on results and discussion. Section 7 details the inferred result of the robust controller design problem as conclusion.

2. Robust PID Controller

2.1 Structure of PID Controller

A PID controller is otherwise considered as an extreme form of a phase lead-lag compensator with one pole at the origin and the other at infinity. A standard PID controller is also known as three term controller. The transfer function for standard PID controller is parallel form is given by

$$G(s) = K_p + K_I \frac{1}{s} + K_D s \tag{1}$$

In Ideal form it is given by

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \tag{2}$$

where

K_p = Proportional gain,

K_I = Integral gain,

K_D = Derivative gain,

T_i = Integral time constant,

T_d = Derivative time constant.

The ‘‘three term” of a PID controller are highlighted below [5]:

- The proportional term – providing an overall control action proportional to the error signal through the all pass gain factor.
- The integral term – reducing steady state errors through low frequency compensation by an integrator.
- The derivative term – improving transient response through high frequency compensation by a differentiator.

2.2 Structure of Robust PID Controller

Consider a control system with n_i inputs and n_o outputs as shown in Figure 1, where $P(s)$ is the plant transfer function, $\Delta P(s)$ is the change in plant, $G_c(s)$ is the controller transfer function , $r(t)$ is the reference input signal, $u(t)$ is the control input, $e(t)$ is the error signal, $d(t)$ is the external disturbance and $y(t)$ is the output of the system [17].

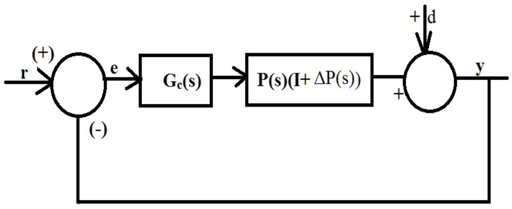

Figure 1. Control system with plant perturbation and external disturbance

The change in plant or plant perturbation is bounded by a known stable function matrix $W_1(s)$.

$$\bar{\sigma}(\Delta P(j\omega)) \leq \bar{\sigma}(W_1(j\omega)),\ \forall_\omega(0,\infty), \tag{3}$$

where $\bar{\sigma}(A)$ denotes the maximum singular value of a matrix A.

The controller G_c (s) is designed by considering an asymptotically stable nominal feedback control system (ΔP(S) = 0 and d(t) = 0) is, the robust stability performance satisfies the following inequality

$$\|W_1(s)S(s)\|_\infty < 1 \tag{4}$$

$$\|W_2(s)T(s)\|_\infty < 1 \tag{5}$$

where the closed loop system is also asymptotically stable with $\Delta P(s)$ and $d(t)$, where $W_1(s)$ is a stable weighting function matrix specified by the designers. $S(s)$ and $T(s) = 1 - S(s)$ are the sensitivity and complementary sensitivity functions of the system respectively.

$$S(s) = (1 + P(s)G_c(s))^{-1} \tag{6}$$

$$T(s) = P(s)G_c(s)\ \ (1 + P(s)G_c(s))^{-1} \tag{7}$$

H_∞ norm in (3) and (4) is defined as

$$\|A(s)\|_\infty = aX_\omega \bar{\sigma}(A(j\omega)) \tag{8}$$

2.3 Control structure of Robust PID controller with filter

The transfer function of Robust PID controller with filter is given as

$$K(s) = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \left(t_d/N\right)s}\right) \tag{9}$$

where

T_i = Integral time constant

T_d = Derivative time constant

t_d/N = Filter time constant

Mostly in PID controllers the derivative part is switched off since the tuning of the controller in the derivative section is difficult. Also the inherent amplification of the measurement noise if not filtered properly can cause damage to the actuator. Therefore there is a need to connect a derivative filter in the Robust PID controller design. The zeros of the PID controller without derivative filter are determined by solving equation 10. The zeros obtained are given by equation 11. The zeros of the PID controller with derivative filter are determined by solving equation 12. The zeros obtained are given by equation 13.

$$T_i T_d S^2 + T_i(s) + 1 = 0 \tag{10}$$

The zeros are

$$Z_{1,2} = \frac{\frac{1}{2} - T_i\sqrt{T_i^2 - 4T_i T_d}}{T_i T_d} \tag{11}$$

If the derivative filter is applied, the zeros of the controller are the solution of the equation.

$$T_i T_d\left(1+\frac{1}{N}\right)S^2+\left(T_i+\frac{T_d}{N}\right)S+1=0 \qquad (12)$$

The zeros are

$$\overline{Z_{1,2}}=\frac{\frac{1}{2}-T_iN-T_d\pm\sqrt{(T_iN-T_d)^2-4T_iT_dN^2}}{T_iT_d(1+N)} \qquad (13)$$

The usage of the filter along with the robust PID Controller significantly enhance the load disturbance rejection performance with a modest increase of a control energy i.e., addition of the filter reduces the control energy for increase in load disturbance rejection [8].

3. Problem Formulation

The problem formulation involves the tuning of the design parameters of robust PID controller to meet out the desired performance specifications. In this paper, the tuning of a robust PID controller has taken into account the following performance specifications such as robustness with respect to model uncertainties and disturbance attenuation, set point tracking, load disturbance rejection and control energy. These specifications are analysed and characterized in terms of norms of signals and systems. The improvement of these performance specifications is formulated as a single objective problem with mixed H_2 and H_∞ norm.

3.1 Robustness with respect to model uncertainties and disturbance attenuation

This is a crucial objective in the design of robust PID controller. Conventional methods like Ziegler Nichols and Cohen - Coon methods failed to produce robust controller designs with guaranteed robustness degree. Hence stability of the control system gets affected [31].

Minimize $J_1 = (J_a^2 + J_b^2)^{1/2}$ (14)

where J_1 = robustness with respect to model uncertainties and disturbance attenuation

J_a = model uncertainty in H_∞ norm

J_b = disturbance attenuation in H_∞ norm

$$J_1=\left|\ \|W_1(s)S(s)\|_\infty+\|W_2(s)T(s)\|_\infty\right|^{\frac{1}{2}} \qquad (15)$$

subjected to the constraints in eqns. (4) and (5).

3.2 Set point tracking

Set point tracking is expressed in terms of the integral performance index J_2 for a step change in set point that always a large initial error. These initial errors are weighted to give a stable output response. Here, the error is defined as the difference between the PV and a user-defined first order reference trajectory $y_r(t)$ connecting the actual PV and the set point. By specifying the time constant of the trajectory, the user can affect the speed of the response to set point changes. The relative importance of each specification depends on the specific control application. Set point changes are more important in other applications such as motion control systems [21].

$$J_2=\int_0^\infty\left|y(t)-r(t)\right|dt \qquad (16)$$

where $y(t)$ = output process variable

$r(t)$ = reference input

3.3 Load disturbance Rejection

In process control the important performance specification required for a PID controller is load disturbance rejection. It is expressed in terms of J_3 It is an integral performance measure of the output to unit step disturbance input when any other input is zero. There are no detailed assumptions made about the load disturbances except that they are low frequency [27].

$$J_3=\int_0^\infty\left|w(t)-y(t)\right|dt \qquad (17)$$

where $y(t)$ = output process variable

$w(t)$ = unit step disturbance

3.4 Measurement of control energy

The fourth performance measure determined by Robust PID controller is control energy (J_4) . It is energy required for the disturbance to settle down [27].

To evaluate control energy, compute the Total Variation (TV) of the input U(t), which is sum of all its moves up and down.

$$TV=\left|U(k+1)-U(k)\right| \qquad (18)$$

The Total Variation is a good measure of the smoothness of the signal and it should be small. A control system exhibits high degree of performance if it provides rapid and smooth response to input changes. The control energy is computed as the total variation of set point tracking and load disturbance rejection.

3.5 Objective function

In this paper, the performance specifications are treated as individual objective functions J_1, J_2, J_3 and J_4. These objective functions are treated as minimization function. The output response of the designed robust PID controller is obtained by the summation of individual functions J_1, J_2, J_3 and J_4. Hence the objective function of the problem under study has been formulated as

$$\text{Minimize } J_{all} = \{J_1 + J_2 + J_3 + J_4\}, \tag{19}$$

where

J_1 = Robustness with respect to model uncertainties and disturbance attenuation

J_2 = Set point tracking

J_3 = load disturbance rejection

J_4 = control energy

J_{all} = Performance of Robust PID controller

PID controller with equal weightage.(i.e. J_1, J_2, J_3 and J_4 have equal weightage as 0.25).

4. Tuning of design parameters of robust PID controller using Evolutionary Algorithms

Evolutionary algorithms are now widely used by the researchers to tune the robust PID controller to meet out the desired performance specifications of the system used. These algorithms are better than traditional analytical methods, since these algorithms make use of population of solutions and not a single point solution. Evolutionary algorithms search for the optimum solution in the search space and hence it has least possibility to converge into local optima. Many evolutionary algorithms have different methodologies to weed out poor off springs and generate fittest population. For solving global functional optimization problems, algorithms like RGA, PSO and DE have been developed. Using these algorithms the convergence rate of obtaining global optimum solution is increased. In this paper RGA, DE and PSO algorithms are used to find out the design parameters of robust PID controller with filter.

4.1 Real coded Genetic Algorithm with SBX crossover

Genetic Algorithm (GA) was introduced by John Henry Holland in the early 1970's. GA is inherently parallel, because it simultaneously evaluates many points in the parameter space and thus has reduced the probability of getting trapped in to local optimum. Real-number encoding is superior for optimization problems with respect to other binary, and grey coded GAs. GA helps to converge to the global optimum [26] with five components.

- Step 1: Representation of solutions according to the optimization problem.
- Step 2: Initialisation of population of solutions.
- Step 3: Evaluation of solution based on the fitness.
- Step 4: Select parents using Tournament selection for taking up reproduction.
- Step 5: Generate offspring using genetic operators (SBX crossover and polynomial mutation operators) by altering the genetic composition of children during reproduction.

They are explained as under.

4.1.1 Simulated Binary crossover

In SBX crossover, two offspring solutions are generated from two parents. Choose a random number $uu_i \varepsilon [0;1]$ and calculate β_{qi} as given below:

$$\beta_{qi} = \begin{cases} (2uu_i)^{\frac{1}{\eta_c+1}}, & uu_i \le 0.5 \\ \left(\dfrac{1}{2(1-uu_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{20}$$

where β_{qi} is spread factor and η_c is the crossover index.

Then compute the children, $X_i^{1,g+1}$ and $X_i^{2,g+1}$ using

$$\begin{aligned} X_i^{(1,g+1)} &= 0.5\left[(1+\beta_{qi})X_i^{(1,g)} + (1-\beta_{qi})X_i^{(2,g)}\right] \\ X_i^{(2,g+1)} &= 0.5\left[(1-\beta_{qi})X_i^{(1,g)} + (1+\beta_{qi})X_i^{(2,g)}\right] \end{aligned} \tag{21}$$

4.1.2 Polynomial Mutation

Offspring thus generated from crossover needs to undergo polynomial mutation operation. Like in the SBX operator, the probability distribution is a polynomial function, rather than a normal distribution. The new offspring is given as

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + \left(x_i^U - x_i^L\right)\bar{\delta}_i \tag{22}$$

where, factor 'δ_i' is found using

$$P(\delta) = 0.5(n_m+1)(1-|\delta|)^{n_m}$$

$$\bar{\delta}_i = \begin{cases} (2r_i)^{1/(n_m+1)} - 1, & \text{if} \quad r_i < 0.5 \\ 1 - 2\left[(1-r_i)\right]^{1/(n_m+1)}, & \text{if} \quad r_i \geq 0.5 \end{cases} \tag{23}$$

where, x_i^U and x_i^L are the upper and lower limit values; and η_m is the mutation index.

Step 6: Generate new population with replacement mechanism, wherein 80% of the fitter off springs and 20% of the best initial population become the individuals for the next generation. Procedure gets repeated until the stopping criterion is satisfied.

4.2 Particle Swarm Optimization (PSO)

PSO is a population-based search procedure in which individuals called particles fly and change their positions in a multidimensional search space with respect to time. During flight, each particle adjusts its position based on its own learning and the experience of neighbouring individuals, thus make use of the best location encountered by own self and its neighbours [12]. The swarm direction of a particle is defined by the set of individuals neighbouring the particle and its experience history.

The PSO algorithm requires the parameters followed below:

Maximum and minimum velocity that limit $v_i(t)$ within the range [Vmin, Vmax]

An inertial weight w_t, and maximum number of iterations

Two random numbers r_1 and r_2 which decide the influence of $p_i(t)$ and $g_i(t)$ on the velocity updating

Two multipliers c_1 and c_2

Subroutine: PSO update

Initialize randomly particle's position and velocity $x_i(0)$ and $v_i(0)$

while stopping condition is not met do

for all particles do

Perform fitness evaluation

Set iteration count, say, 'k'

Calculate p_i by identifying the best position of i^{th} individual at iteration k^{th} and g_i by finding the best position of the group until iteration 'k'.

Update the velocity using

$$v_i(t+1) = wv_i(t) + c_1 r_1 \left(p_i(t) - x_i(t)\right) + c_2 r_2 \left(g_i(t) - x_i(t)\right) \tag{24}$$

Calculate the movement of the swarm by updating the position of the particle using

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{25}$$

since each individual moves from the current position to the next one with the modified velocity

end for

Check with the maximum iteration count and if not satisfied, increase the iteration count and repeat the procedure from Step 3.

end while

Return: Optimum attractiveness

4.3 Differential Evolution

DE is an efficient population-based stochastic search technique for global optimization in many real problems. Differential Evolution is developed from Ken Price's attempts to find out solution for Chebychev Polynomial fitting Problem [23]. When Ken got an idea of using vector differences for perturbing the vector population, a breakthrough has come up. DE is considered better, since random direction is generated by simple vector subtraction and more variation in population leads to more search over solution space.

Step 1: Generate the initial population (Np) by

$$X_{i,G} = \left\{ x_{i,G}^1, \ldots, x_{i,G}^n \right\}, \tag{26}$$

where $i = 1,2,\ldots Np$

n = Dimension, which is chosen randomly and must bound the complete parameter space constrained by the specified minimum and maximum limits.

Step 2: Mutation Operation

At each generation, after initialization DE uses mutation to produce a mutant vector, $V_i G$ for each target vector, $X_i G$. The mutant vector can be produced by several mutation strategies. The basic mutation operation is given as

$$V_i = X_{r1} + F.(X_{r2} - X_{r3}) \quad (27)$$

where, r_1, r_2, r_3 are three random indexes between $i = 1,2,\ldots Np$, excluding, 'i'; F is a scaling factor.

Step 3: Crossover operation

Crossover operation is applied to each pair of the target vector $X_i G$ and its corresponding mutant vector $V_i G$ to generate a trial vector, $U_i G$ DE generally employs the binomial crossover defined by

$$u_{i,G}^{j} = \begin{cases} v_{i,G}^{j}, & \text{if } (\text{rand}_j [0,1) \le CR \text{ or } (j = j_{rand}) \\ x_{i,G}^{j}, & \text{otherwise} \end{cases} \quad (28)$$

$$j = 1, 2, \ldots., n$$

The crossover rate is a user-specified constant within the range [0,1), which controls the fraction of parameter values copied from the mutant vector. j_{rand} is a randomly chosen integer in the range [1,n].

The condition j = j_{rand} is introduced to make sure that the trial vector will vary from its corresponding target vector by at least one parameter. Otherwise, new parent vector would not be generated and the population will not vary.

Step 4: Selection

A selection phase is necessary to decide, if the trial vector enters the next generation population or not. In the selection phase each trial vector is compared to the corresponding target vector and the better vector will come into the population of the next generation.

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases}, \quad (29)$$

where $f(U_{i,G})$ is the fitness function value of the ith trial vector; and $f(X_{i,G})$ is the fitness function value of the i^{th} target vector.

The loss of the superior individuals in the subsequent iteration is prevented by the selection process, since the fitter individuals replace the inferior individuals. This process continues until the maximum function evaluations or generations are reached.

5. Test Systems

In order to validate the performance of the robust PID controller with filter, two SISO test systems are considered.

5.1 Test System 1

The transfer function of the test system 1 i.e., phase locked loop motor speed control system [4] is given as

$$P(s) = \frac{68.76}{S(1 + 0.05s)} \quad (30)$$

Also the system suffers from an external disturbance d(t) = 0.1 e $^{-0.1t}$ sin (0.8πt) and the plant perturbation transfer function is given as

$$\Delta P(s) = \frac{0.6}{s^2 + 0.2s + 8} \quad (31)$$

The plant perturbation transfer function is bounded between two weighting function namely $W_1(s)$ and $W_2(s)$

$$W_1(s) = \frac{0.6}{s^2 + 0.2s + 8} \quad (32)$$

$$W_2(s) = \frac{0.5s + 0.05}{s^2 + 0.2s + 6.3265} \quad (33)$$

The controller transfer function is given by

$$K(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \left(\frac{T_d}{N}\right) s} \right) \quad (34)$$

5.2 Test System 2

Consider the Magnetic Levitation system as test system 2 [11,22] and its linearized model about an equilibrium point of y = 0.018 is given as

$$P(s) = \frac{7.147}{(s - 22.55)(s + 20.9)(s + 13.99)} \quad (35)$$

The PID controller transfer function is given by

$$K(s;x) = 10^{x_1}\left(1 + \frac{1}{10^{x_2}s} + \frac{10^{x_2}s}{1+10^{(x_3-x_4)}s}\right) \quad (36)$$

where $X = (x_1, x_2, x_3, x_4)^T$ denotes the design parameter vector.

Suppose the initial search space of the design parameter vector is given by

$$D := \left\{x \in R^4 \mid (2,-1,-1,2)^T \le x \le (4,1,1,3)^T\right\} \quad (37)$$

The design parameter x which satisfies the following multiple H_∞ constraints given in eqns. (4) and (5) are determined. The plant perturbation is unknown in fact, but bounded by the following known stable function.

$$W_1(s) = 5/(s+0.1) \quad (38)$$

$$W_2(s) = 4.3867\times10^{-7}(s+0.066)(s+31.4)(s+88)\left(10^4/(s+10^4)\right)^3 \quad (39)$$

6. Simulation Results

The performance of the proposed evolutionary algorithm tuned Robust PID controller is validated using two different test systems. The two different test systems used are phase locked loop motor speed control system and simple magnetic levitation system. The evolutionary algorithms used for the determination of best design parameters are RGA, DE and PSO. The coding is run using MATLAB 7.10.0 software on Pentium 4 PC 2.16 GHZ with 2 GB RAM.

Since the solutions obtained using evolutionary algorithm is probabilistic in nature, many trials with independent population initializations are used to check the constant performance of algorithms. As the number of design parameters is four, the population size is fixed as 40. In this work 20 trials are conducted and feval max is fixed as 4000.

6.1 Simulation results for Test System 1

The design parameters like proportional gain K_p, integral time constant T_i, derivative time constant T_d and filter time constant t_d/N obtained for the PLL motor speed control system using RGA, PSO and DE are given in Table 1.

Table 1. Design Parameters obtained for PLL system using different optimization algorithms

Sl No	Algorithm	Design parameters			
		K_p	T_i	T_d	t_d/N
1.	RGA	0.1196	0.3405	0.6559	4.8694
2.	DE	0.1464	0.5723	0.4733	3.8388
3.	PSO	0.1152	0.3273	0.6736	4.9647

The convergence characteristics obtained using PSO, DE and RGA is shown in Figure 2. From the convergence characteristics it is clear that RGA converges faster than PSO and DE.

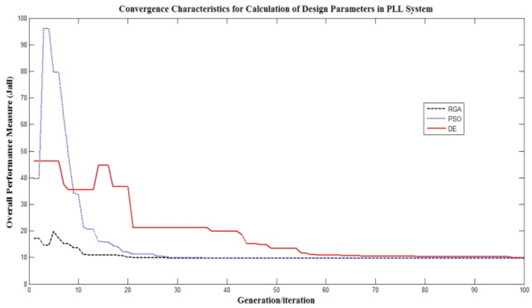

Figure 2. Convergence characteristics for PLL system

The overall response (J_{all}) of the system with plant perturbation $\Delta P(s)$ and the performance specification parameters like robustness with respect to model uncertainties (J_1), set point tracking (J_2), load disturbance rejection (J_3) and control energy (J_4) using DE, PSO and RGA tuned controller is shown in Table 2.

Table 2. Performance parameter specifications of robust PID controller obtained using RGA, DE and PSO for PLL System

Sl. No	Algorithm	J_1	J_2	J_3	J_4	J_{all}
1.	RGA	1.1180	0.1470	4.8107	3.5297	9.6054
2.	DE	1.0895	0.1332	5.1249	3.6019	9.9495
3.	PSO	1.1123	0.1442	4.7416	3.6224	9.6206

The output response of the PLL system obtained using the design parameters computed by RGA, DE and PSO is shown in Figure 3.

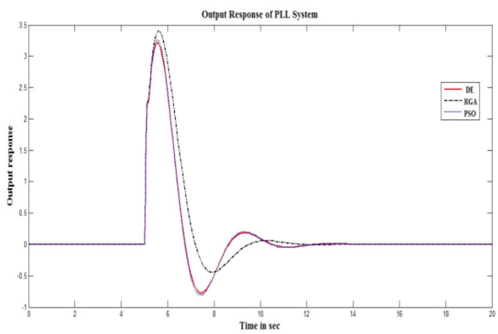

Figure 3. Output response of PLL system

From the overall (J_{all}) value and the output response, it is clear that RGA algorithm is better than DE and PSO to meet out the system specifications i.e., RGA tuned robust PID controller performs better with plant perturbation and model uncertainties for PLL system.

6.2 Simulation results for Test System 2

The design parameters like proportional gain K_p, integral time constant T_i, derivative time constant T_d and filter time constant t_d/N obtained for the magnetic levitation system using RGA, PSO and DE are given in Table 3.

Table 3. Design Parameters obtained for MLS system using different optimization algorithms

Sl No	Algorithm	Design parameters			
		K_p	**T_i**	**T_d**	**t_d/N**
1.	RGA	3.2452	-0.9991	-0.7023	2.2815
2.	PSO	3.2467	-1.000	-0.7047	2.2771
3.	DE	3.2477	-0.9764	-0.7155	2.4099

The convergence characteristics obtained using RGA, PSO and DE is shown in Figure 4. From the convergence characteristics it is clear that DE has the lowest fitness characteristics than RGA and PSO.

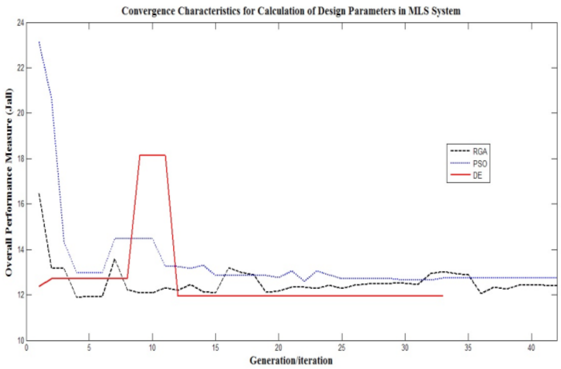

Figure 4. Convergence characteristics for MLS system

In Magnetic Levitation System the PID controller structure is given by eqn. (36), where $x=(x_1, x_2, x_3, x_4)^T$ denotes the design parameter vector. Here $K_p=10^{x1}$, $T_i=10^{x2}$, $T_d=10^{x3}$ and $t_d/N=10^{(x3-x4)}$. The modified transfer function obtained by substituting the computed design parameters using RGA, PSO and DE for robust PID controller with filter is given in Table 4.

Table 4. Modified Transfer function obtained for robust PID controller with filter using RGA, PSO and DE

S1 No	Algorithm	Transfer function of robust PID with filter obtained using the design parameters
1.	RGA	K(s)=1758.7*(1+(1/0.1002s)+(0.1985s/(1+(1/0380*10-3)s))
2.	PSO	K(s)=1764.8*(1+(1/0.1s)+(0.1973s/(1+(1.0427*10-3)s))
3.	DE	K(s)=1768.9*(1+(1/0.1056s)+(0.1925s/(1+(0.7492*10-3)s))

The overall response (J_{all}) of the system with plant perturbation $\Delta P(s)$ and the performance specification parameters like robustness with respect to model uncertainties (J_1), set point tracking (J_2), load disturbance rejection (J_3) and control energy (J_4) using DE, PSO and RGA tuned controller is shown in Table 5.

Table 5. Performance parameter specifications of robust PID controller obtained using RGA, DE and PSO for MLS System

S1 No	Algorithm	J_1	J_2	J_3	J_4	J_{all}
1.	RGA	1.3428	0.3518	0.000205	10.8478	12.5426
2.	PSO	1.3401	0.3552	0.000208	11.0520	12.7475
3.	DE	1.5270	0.3429	0.000211	10.0615	11.9316

The output response of the MLS system using the design parameters computed by RGA, DE and PSO is shown in Figure 5.

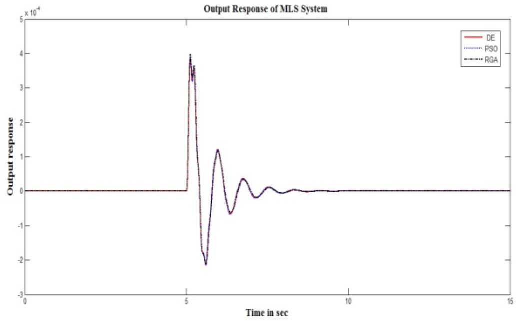

Figure 5. Output response of MLS system

From the overall (J_{all}) value and the output response, it is clear that DE algorithm is better than RGA and PSO to meet out the system specifications i.e., DE tuned robust PID controller performs better with plant perturbation and model uncertainties for MLS system.

7. Conclusion

The performance comparison of Evolutionary Algorithm techniques to tune the desired design parameters of robust PID controller with filter is discussed in this paper. The design parameters of robust PID controller with filter structure for Single Input Single Output (SISO) system is computed using Real coded Genetic Algorithm (RGA), Differential Evolution (DE) algorithm and Particle Swarm Optimization (PSO) algorithm. Simulation results show that the overall performance of the robust PID controller obtained by the computed design parameters using RGA is better than PSO and DE for PLL system and for MLS system DE tuned robust PID controller performance is better than RGA and PSO.

REFERENCES

1. Aidan O Dwyer, (2006), Handbook of PI and PID controller tuning rules, *Imperial College Press*, London.
2. Alberto Herreros, Enrique Baeyens, & Jose R. Peran, (2002), Design of PID – Type controllers

using Multi objective GA, *ISA Transactions*, 41, pp. 457-472.

3. Astrom, K. & Hagglund, T. (1995), PID Controllers: Theory, Design, and Tuning, *2nd Edition. Instrument Society of America, Research Triangle Park,* NC.
4. Bor – Sen Chen & Yu – Min Cheng, (1998), A structure – specified H_∞ Optimal control design for practical applications: A genetic approach, *IEEE Transactions on Control Systems Technology,* 6(6).
5. Cheng-Ching yu, (2006), Auto Tuning of PID controllers – A relay feedback approach, *Springer*, 2nd Edition.
6. Chen, Yu Min Cheng, & Ching Hsiang Lee, (Oct.1995), A Genetic Approach to mixed H_2/ H_∞ optimal PID control, *IEEE control systems,* pp. 51-59.
7. Coelho, L.S. & Pessôa, M.W. (2011), A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map", *Expert Systems with Applications,* 38(11),13694–13701.
8. Ding yu Xue, Yang Quan-Chen, Derek, P. & Atherton, (2002), Linear feedback control – Analysis and Design with MATLAB, *Springer Verlag.*
9. Felix, M.A. & Wagner, A. (2008), Robustness and evolution: Concepts, insights and challenges from a developmental model system, *Heredity*, 100(2), 132–140.
10. Helonvicente Hultmann, Ayala & Leandro dos Santos Coelho, (Aug.2012), Tuning of PID controller based on a multi objective genetic algorithm applied to a robotic manipulator, *Expert systems with applications,* 39(10), 8968 – 8974.
11. Ichiro Maruta,Tae Hyoung Kim,Toshiharu (2009), Fixed-structure H_∞ controller synthesis: A meta-heuristic approach using simple constrained particle swarm optimization, *Automatica 45, 53-559.*
12. James Kennedy & Russell Eberhart (1995), Particle Swarm Optimization, *Proceedings of International Conference on Neural Networks,* Perth, Australia.
13. Jimenez, (Dec.2015), An auto-tuning PID control system based on genetic algorithms to provide delay guarantees in Passive Optical Networks, *Expert Systems with Applications*, 42(23), 9211-9220.
14. Kiam Heon Ang, G. & Chong, Yun Li, (2005), PID control system analysis, design and technology, *Published in IEEE Transactions on Control Systems Technology,* 13(4), pp. 559 – 576.
15. Kristiansson, B. Lennartson, (2002) "Robust and optimal tuning of PI and PID controllers", *Proceedings in IEEE Control Theory Applications*, vol. 149, no. 1, pp. 17-25.
16. Krohling, (1997), Design of PID controller for disturbance rejection: A genetic Optimization approach, *2nd International Conference on Genetic Algorithms in Engineering Systems*, pp. 498-503.
17. Ming Hao Hung, Li Shun Shu, Shinn Jang Ho, Shiow Fen Hwang & Shinn Ying Ho, (Mar.2008), A Novel Intelligent Multi objective Simulated Annealing for designing robust PID controllers, *IEEE Transactions on Systems, MAN and Cybernetics –Part A, Systems and Humans,* 38(2).
18. Moharam Mostafa A. El-Hosseini & Hesham A. Ali, (2015), Design of optimal PID controller using NSGA II Algorithm and Level Diagram, *Studies in Informatics and Control*, 24(3), 301-308.
19. Moharam,Mostafa A.EI- Hosseini, Hesham A. Ali,(2016),Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challangers",Appied Soft Computing, 38(C), 727-737.
20. Qi Bing–Jin, Lin Feng–Qi, Beiyan–Jiang & Qi–Wang, (2014), Novel improved Cuckoo search for PID controller Design, *Transactions of the Institute of Measurement and control*, 37(6), 721-731.
21. Rainer Dittmar, A. Shabroz Gill, Harpreet Singh & Mark–Darby, (2012), Robust Optimization – Based Multi-loop PID controller tuning: A new tool and its industrial application, *Control Engineering Practice,* 20, 355-370.
22. Sreepriya, S. Baskar S. & Willjuice Iruthayarajan M. (2010), Covariance matrix adopted evolutionary strategy based design of robust optimal fixed structure controller, *Proceedings of Second International Conference on Computing, Communication and Networking Technologies.*
23. Storn, R. & Price, K. (1997), Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global optimization,* 11(4), 341-359.
24. Tae-Hyoung Kim, Ichiro Maruta, & Toshiharu Sugie (2007), Robust PID controller tuning based on the constrained Particle Swarm optimization, *Automatica*, 44(4), 1104-1110.
25. Tan, K. K. Huang, S. & Ferdous, R. (2002), Robust self-tuning PID controller for non linear systems, *Journal of process control*, 12(7), 753-761.
26. Thomas Back & Hans-paul schwefel, (1993), An overview of Evolutionary Algorithms for parameter optimization, *Evolutionary computation*, 1(1), 1-23.
27. Vijayakumar, V. Rao, V.S.R. & Chidambaram, M. (2012), Centralized PI controllers for interacting multi variable processes, *ISA Transactions*, 51, 400-409.
28. Wang, L. Barnes, T.J.D. & Cluett, W.R. (1995), New frequency-domain design method for PID controllers," *Proceedings of Institute of Electrical Engineers in Control Theory Applications*, 142(4), pp. 265–271.
29. Yang, X.S. & Deb, S. (2010), Engineering optimisation by cuckoo search, *International Journal of Mathematical Modeling and Numerical Optimisation*, 1(4), 330–343.
30. Zhao, Y. & Liu, H.L. (2013), Multiobjective particle swarm optimization based on population decomposition, *Lecture Notes in Computer Science,* 8206, Springer, Berlin, Heidelberg.
31. Zhao, S. Z. Willjuice Iruthayarajan, M. S. Baskar, S. & Suganthan, P.N (2011), Multi- Objective Robust PID controller tuning using Two Lbests Multi – objective Particle Swarm Optimization", *Information sciences,* 181, 3323-3335.