International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 5, October 2016, pp. 2033~2040
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i5.10633

Robustness Study of Fractional Order PID Controller Optimized by Particle Swarm Optimization in AVR System

N. Ramesh Raju[1], P. Linga Reddy[2]
[1] Department of Electrical and Electronics Engineering, Siddharth Institute of Engineering & Technology, India
[2] Department of Electrical and Electronics Engineering, K L University, India

Article Info

Article history:

Received Mar 26, 2016
Revised Jul 14, 2016
Accepted Jul 30, 2016

Keyword:

AVR System
Genetic Algorithm
Particle Swarm Optimization
PID Controllers
$PI^{\lambda}D^{\mu}$

ABSTRACT

In this paper a novel design method for determining fractional order PID ($PI^{\lambda}D^{\mu}$) controller parameters of an AVR system using particle swarm optimization algorithm is presented. This paper presents how to employ the particle swarm optimization to seek efficiently the optimal parameters of $PI^{\lambda}D^{\mu}$ controller. The robustness study is made for this controller against parameter variation of AVR system. This work has been simulated in MATLAB environment with FOMCON (Fractional Order Modeling and Control) tool box.The proposed PSO$PI^{\lambda}D^{\mu}$ controller has superior performance and robust compared to GA tuned $PI^{\lambda}D^{\mu}$ controller. The results are also compared with PSO tuned PID controller.

Corresponding Author:

N. Ramesh Raju,
Department of Electrical and Electronics Engineering,
Siddharth Institute of Engineering & Technology,
Narayanavanam Road, Near Tirupathi -Chennai Highway, Siddharth Nagar, Puttur, Chittoor,
AndhraPradesh 517583.
Email: nandi_rameshraju@yahoo.com

1. INTRODUCTION

PID controllers are simple and well known for process control applications and also for simple feedback control mechanisms.The biggest challenge lies in tuning the parameters of the PID controller. Traditional tuning methods such as Ziegler and Nichols are available but do not give optimum performance [1]. AI techniques such as fuzzy logic , neural networks neural-fuzzy logic have been widely used to proper tuning of PID controller parameters[2]-[4]. Tuning of PID controller parameters in off line made easy with the advent of heuristic optimization techniques [5]-[7]. Genetic Algorithms and Particle swarm Optimization are very popularly used random search heuristic optimization techniques for tuning PID controller parameters [6],[8]. These techniques have very high probability to achieve global optimum solution.

Since last decade the fractional order PID controllers are widely accepted in place of integer order PID controller [9],[10]. It has been proved that fractional order PID controller has superior performance compared to integer order PID controller [9]. The problem with fractional order PID controller is that it has five parameters need to be tuned,where as integer order PID controller has three parameters need to be tuned. This makes one step tuning of fractional order PID controller difficult compared to integer order PID controller. In this paper particle swarm optimization is used to tune the $PI^{\lambda}D^{\mu}$ controller in MATLAB/FOMCON environment. In this paper robustness study is made by comparing the results obtained through genetic algorithm based $PI^{\lambda}D^{\mu}$ controller.

2. RESEARCH METHOD

2.1. Automatic Voltage Regulator (AVR) System

Basic purpose of an AVR system is to control the terminal voltage of a synchronous generator by manipulating excitation. An AVR system basically contains four components namely Amplifier, Exciter, Synchronous generator and Sensor. One of the main cause of change in terminal voltage is reactive power consumption of the load. This change in voltage can be compensated by controlling the excitation to the generator. The block diagram of AVR system after linearization is presented in Figure 1.

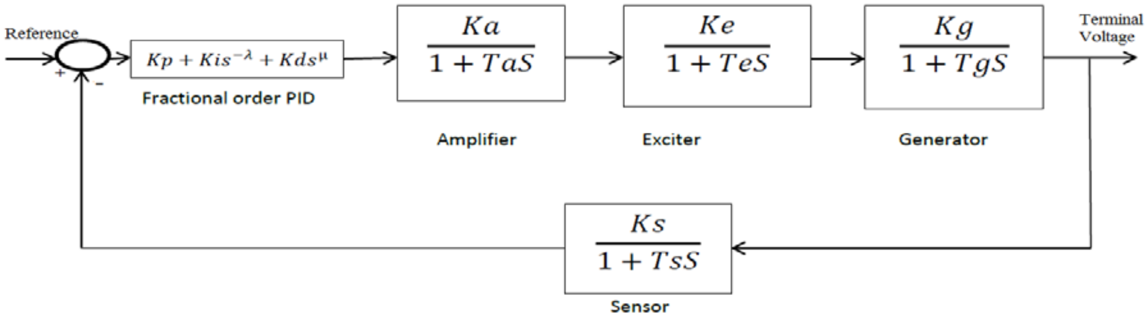

Figure 1. Block diagram of AVR system

The parameters of the generator of an AVR system depends on the load and the parameters of the remaining components depends on their design. The nominal range of the parameters are as follows [6].

10< Ka <40; 0.02s < Ta < 1s
1< Ke <10; 0.4s < Te < 1s
Kg and Tg depends on the load 0.7< Kg<1; 1s<Tg< 2s
0.01s < Ts < 0.06s and Ks =1

2.2. Particle Swarm Optimization

Particle swarm optimization method was introduced by Kennedy and Eberhart in1995. It is evolutionary optimization technique and stochastic method, developed by observing the social movement of swarms such as fish schooling and bird flocking. This method is robust in solving problems featuring nonlinearity, non differentiability, multiple optima and high dimensionality. It has stable convergence characteristics with good computational efficiency and easily implementable. Unlike other evolutionary methods where the evolutionary operators manipulate the particle, each particle in PSO flies in the search space with velocity which is dynamically adjusted according to its own flying experience and flying experience of its companions'.

At the beginning PSO algorithm introduces 'N' number of particles randomly. The objective function value is obtained for each particle. Then based on the flying velocity of the particle and its group the new population of particles are generated for next generation in seeking still better solution. The best value obtained by the particle so far is called pbest and the best value obtained among all the particles is called gbest. Each particle in the group updates their velocity based on the pbest and gbest as given in equation (1) and (2).

Let us assume jth particle is represented as $x_j = (x_{j,1}, x_{j,2}, \ldots\ldots x_{j,n})$ in n dimensional space. The previous best position of the jth particle is recorded as $pbest_j = (pbest_{j,1}, pbest_{j,2}, \ldots\ldots pbest_{j,n})$. The best particle among the group is represented by gbestg. The velocity of the particle j is represented as $v_j = (v_{j,1,} v_{j,2} \ldots\ldots v_{j,n})$. The calculation of modified velocity and position of each particle using velocity and distance through pbestj,g to gbestg is done as shown in the following formulas:

$$V_{j,n}(t+1) = w.v_{j,n}(t) + c_1*rand()*(pbest_{j,n}-x_{j,n}(t)) + c_2*rand()*(gbest_g-x_{j,n}(t)) \quad (1)$$

$$x_{j,n}(t+1)= x_{j,n}(t)+v_{j,n}(t+1) \quad (2)$$

$$j= 1, 2 ,\ldots\ldots\ldots\ldots\ldots,N$$

n=1, 2 ,…………….,M

Where,
Nnumber of particles in a group
M number of members in a particle
t generation number
$v_{j,n}(t)$ velocity of particle j at generation t
w inertia weight factor
c_1, c_2 acceleration constant
rand() Random number between 0 and 1
$x_{j,n}(t)$ current position of particle j at generation t
$pbest_j$ pbest of particle j
$gbest_g$ gbest of the group

The updated particles are the population for next generation and continue the above procedure up to the specified number of generations. The better solution is obtained at each subsequent generation.

2.3. Fractional Order systems
2.3.1. Fractional Order Calculus

Some of the practical systems could be well described using fractional order differential equations rather than integer order differential equations, in 1695, L'Hopital coined the word fractional order calculus [11]. Since then Euler, Laplace, Fourier, Able, Riemann, and Lurel worked on this. The research on the fractional order calculus is accelerated from 1884. The basi operator in the fractional order calculus is differintegral. This name has come because a single operator represents the fractional order derivative and fractional order integrator. The differintegral is represented as following:

$$ {}_aD_t^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dt^{r}} R(\alpha) > 0 \\ 1 \quad R(\alpha) = 0 \\ \int_a^t (d\tau)^{-\alpha} R(\alpha) < 0 \end{cases} \quad (3) $$

Where 'a' and 't' are the limits of the operator. The operator 'α' is the order of the operation and belongs to R (any rational number) but 'α' could also be a complex number [12]. Two definitions used for the general fractional differintegral are the Grunwald-Letnikov (GL) definition and the Riemann-Louville (RL) definition [13],[14]. The GL is given here:

$$ {}_aD_t^{\alpha} f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{j=0}^{\frac{t-a}{h}} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (4) $$

The fractional differintegral defined by RL is

$$ {}_aD_t^{r} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{(\alpha-n+1)}} d\tau \quad (5) $$

for (n−1 <α <n) and □(.) is the Gamma function.

2.3.2. Fractional Order PID Controllers

For last one-decade fractional order PID controllers are became very popular among researchers, because its robust performance and fast response. The fractional order PID controller transfer function G(s) is defined as in (6). Figure 2 gives the graphical presentation of PID controller.

$$ G(s) = Kp + Kis^{-\lambda} + Kds^{\mu} \quad (6) $$

Where,
Kp→Proportional gain
Ki → Integral gain
Kd→ Derivative gain
λ → order of the integrator
μ→ order of the differentiator

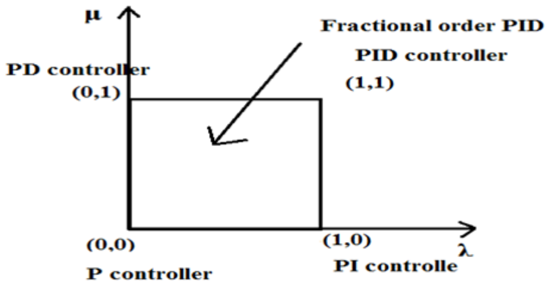

Figure 2. Graphical representation of fractional order PID controller

The fractional order PID controller needs tuning of above five parameters appropriately to make the system performance optimum.

2.4. Performance Evaluation of $PI^{\lambda}D^{\mu}$ Controller

For each particle in the population, which represents five parameters of the controller, the AVR system is simulated. Based on the system response the time domain parameters of the response such as Peak overshoot (Mp), Rise time (t_r), Settling time (t_s) and Stedy state error (Ess) are evaluated. These four time domain parameters depends on five parameters of the controller (Kp,Ki,Kd,ʎ,μ).The systemis good when these four parameters are minimum. The performance criterion (O) of the $PI^{\lambda}D^{\mu}$ controller is defined as in (7).

$$O(Kp,Ki,Kd,ʎ,\mu)= \beta(Mp+Ess)+(1-\beta)(t_s+t_r) \quad (7)$$

Since the objective function (O) has to be minimised, the fitness function(f) is defined as in (8)

$$f= 1/O \quad (8)$$

More importance is given to reduce peak overshoot and best value for β is found to be 0.92.

3. RESULTS AND ANALYSIS

The proposed robust analysis of fractional order PID controller optimized by PSO in AVR system is carried out through MATLAB/SIMULINK in combination with fractional order systems tool box FOMCON. The PSO algorithm is implemented through MATLAB code. The search space for the PSO-PIʎDμ is defined as follows:

The controller parameters search space is chosen as

$3<Kp<8,\ 0.5<Ki<2,\ 0.5<Kd<1.5, 0<\lambda<1.5,\ 0<\mu<1.5$

The velocity limits for each parameter and limits of inertial factor is chosen as

$-1.2<Vmin(Kp)<1.2,\ -0.35<Vmin(Ki)<0.35,\ -0.2<Vmin(Kd)<0.2$
$-0.15<Vmin(\lambda)<0.15,\ 0.15<Vmin(\mu)<0.15,\ 0.4<w<0.9$

When AVR system shown in Figure 1 is optimized through PSO-$PI^{\lambda}D^{\mu}$ and GA-$PI^{\lambda}D^{\mu}$ for the system parameters shown in Table 1, the convergence of the objective function for 100 generations is presented in Figure 3. It shows that the convergence in the case of PSO-$PI^{\lambda}D^{\mu}$ is superior than GA-$PI^{\lambda}D^{\mu}$. The optimum controller parameters and time response is shown in Table 2 and Figure 4. Table2 also presents the performance comparison for controller parameters tuned by RGAPID and PSOPID. Among all these methods the PSO-$PI^{\lambda}D^{\mu}$ controller gives better time response.

Table 1. Parameters ofAVR system

Ka	Ta	Ke	Te	Kg	Tg	Ks	Ts
10	0.1	1	0.4	1	1	1	0.01

Table 2. Performance comparison of PSO-PI$^{\lambda}$D$^{\mu}$with other tuning methods

Method	Kp	Ki	Kd	Λ	μ	Ts(s)	Tr(s)	Osh(%)
RGAPID	0.6820	0.2660	0.1790	1	1	1.2682	1.0668	4.00
PSOPID	0.6570	0.5389	0.2458	1	1	0.4025	0.2767	1.16
GAPI$^{\lambda}$D$^{\mu}$	5.6471	1.2471	0.6667	1.1471	1.4824	0.3000	0.0900	4.63
PSOPI$^{\lambda}$D$^{\mu}$	4.0143	0.8963	0.5014	1.3349	1.4154	0.2000	0.13	2.69

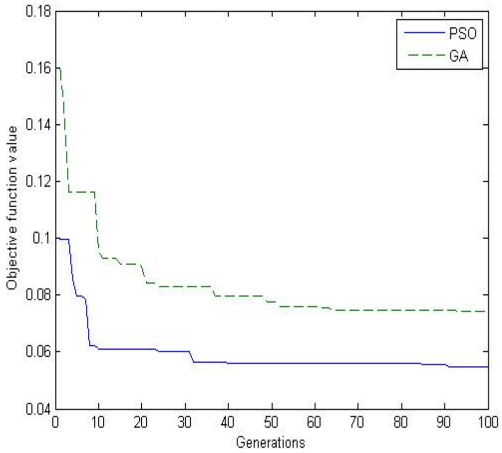

Figure 3. Convergence of objective function over 100 generations

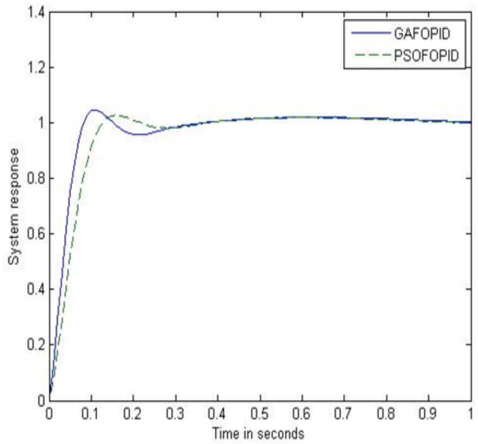

Figure 4. Time response of GAPI$^{\lambda}$D$^{\mu}$ and PSOPI$^{\lambda}$D$^{\mu}$ for Kg=1 and Tg=1

By observing the time response in Figure 4, the settling time and peak overshoot is less in the case of PSOPI$^{\lambda}$D$^{\mu}$ compared to GAPI$^{\lambda}$D$^{\mu}$,PSOPID and RGAPID controllers. The rise time is marginally high compared to other controllers.

To study the robustness of the AVR system for PSOPI$^{\lambda}$D$^{\mu}$ controller, the optimum controller parameters are obtained for 24 combinations of quantized values of generator parameters in their variation range. Kg and Tgis quantized as {0.7, 0.8, 0.9,1}, {1, 1.2, 1.4, 1.6, 1.8, 2}respectively. The obtained optimum controller parameters and time response parameters are shown in Table 3, Figure 5 and Figure 6.

Table 3. Optimum controller parameters and time response parameters tuned by PSOPI$^{\lambda}$D$^{\mu}$

S.No.	Generator Parameters		Controller Parameters					Time response parameters		
	Kg	τg	Kp	Ki	Kd	λ	μ	Ts(sec)	Tr(sec)	Mp(%)
1	0.7	1	4.0710	1.0384	0.6099	1.0849	1.3569	0.17	0.18	1.4
2	0.8	1	4.3582	1.8241	0.6206	1.3993	1.3760	0.22	0.14	3.23
3	0.9	1	6.5388	1.1746	0.7393	1.0300	1.500	0.30	0.08	4.16
4	0.7	1.2	3.666	1.7325	0.7984	0.6470	1.2902	0.28	0.17	3.67
5	0.8	1.2	4.2216	1.0428	0.6581	1.3410	1.3445	0.23	0.17	2.21
6	0.9	1.2	3.9876	0.9057	0.5704	1.3173	1.3807	0.16	0.17	1.95
7	0.7	1.4	6.4880	0.6434	0.9432	0.8358	1.3947	0.21	0.14	2.43
8	0.8	1.4	4.9261	1.0149	0.8350	1.0264	1.3447	0.18	0.18	1.66
9	0.9	1.4	4.2029	0.6006	0.6550	1.1792	1.3645	0.17	0.17	1.60
10	1	1.4	3.0928	0.7700	0.5488	1.0379	1.3269	0.19	0.20	1.76
11	0.7	1.6	7.5407	0.8775	1.1104	1.1828	1.3978	0.21	0.14	2.95
12	0.8	1.6	5.4834	0.5086	0.8613	1.3337	1.337	0.16	0.17	1.74
13	0.9	1.6	3.0835	1.0773	0.6782	0.9281	1.2780	0.21	0.22	1.9
14	1	1.6	3.0362	0.5779	0.5932	1.1169	1.2971	0.21	0.22	1.23
15	0.7	1.8	4.8817	1.1221	0.9568	1.0393	1.3078	0.21	0.22	1.92
16	0.8	1.8	5.0936	1.1821	0.9861	1.0232	1.3260	0.25	0.17	2.59
17	1	1.8	4.5345	0.6019	0.7565	1.4763	1.3603	0.17	0.17	1.66
18	0.7	2	6.9420	0.7572	1.1710	1.2067	1.3611	0.17	0.18	1.99
19	0.8	2	5.7915	0.5640	0.9942	1.2597	1.3505	0.18	0.19	1.82
20	0.9	2	4.2913	0.9132	0.8873	0.9023	1.3092	0.19	0.20	1.82
21	1	2	5.7162	0.7704	0.9614	1.0908	1.3838	0.37	0.14	2.59
Average values of PI$^{\lambda}$D$^{\mu}$ (AVPI$^{\lambda}$D$^{\mu}$)			4.7641	0.9095	0.7820	1.1310	1.3552			

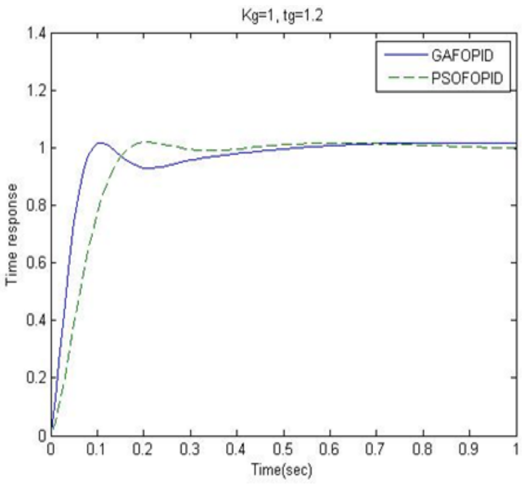

Figure 5. Time response of GAPI$^{\lambda}$D$^{\mu}$ and PSOPI$^{\lambda}$D$^{\mu}$ Kg=1 and Tg=1.2

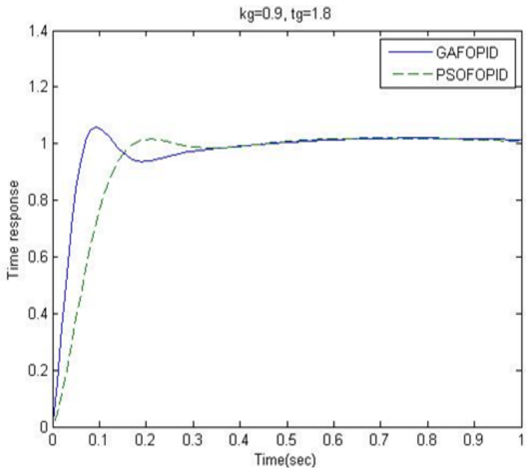

Figure 6. Time response of GAPI$^{\lambda}$D$^{\mu}$ and PSOPI$^{\lambda}$D$^{\mu}$ Kg=0.9 and Tg=1.8

The Table 3, Figure 5 and Figure 6 clearly shows that the time response of PSOPI$^{\lambda}$D$^{\mu}$ controller tuned against parameter variations achieves very good time response. The peak overshoot and settling time is less compared to GA tuned PI$^{\lambda}$D$^{\mu}$ controller i.e 4.63% and 0.3 seconds respectively (Table 2). However the rise time is marginally high which is normally not so important if the settling time is less.

In this Paper to compare the robustness of the PSOPI$^{\lambda}$D$^{\mu}$ controller, GAPI$^{\lambda}$D$^{\mu}$ controller has been considered. Similarto the PSOPI$^{\lambda}$D$^{\mu}$ controller, optimum controller parameters are obtained for parameter variation for GAPI$^{\lambda}$D$^{\mu}$ controller. The results are presented in Table 4 [15].

Table 4. Optimum controller parameters and time response parameters tuned by GAPI$^{\lambda}$D$^{\mu}$

S.No.	Generator Parameters		Controller Parameters					Time response parameters		
	Kg	τg	Kp	Ki	Kd	λ	μ	Ts(sec)	Tr(sec)	Mp(%)
1	0.7	1	6.6824	1.8118	0.9020	1.1529	1.5	0.37	0.09	2.17
2	0.8	1	6.6824	2.4118	0.9255	1.0941	1.5	0.34	0.08	5.00
3	0.9	1	4.8000	1.1765	0.7059	1.2882	1.5	0.44	0.09	1.44
4	1	1	5.5294	1.0235	0.7216	1.4471	1.5	0.35	0.08	4.88
5	0.7	1.2	7.3412	1.9412	1.1294	1.2471	1.5	0.42	0.09	2.50
6	0.8	1.2	7.3882	1.5647	1.0118	0.9765	1.5	0.34	0.08	3.70
7	0.9	1.2	6.0706	1.2706	0.8154	1.2882	1.5	0.36	0.10	1.99
8	1	1.2	4.9647	1.1647	0.7529	1.3294	1.5	0.42	0.10	1.85
9	0.7	1.4	7.6471	1.7176	1.2784	0.9882	1.5	0.45	0.09	1.85
10	0.8	1.4	7.7647	1.3765	1.0667	1.3588	1.5	0.35	0.10	2.40
11	0.9	1.4	6.0706	1.3882	0.9804	1.0059	1.5	0.42	0.09	2.00
12	1	1.4	5.7882	1.1882	0.9020	1.0412	1.5	0.41	0.09	2.40
13	0.7	1.6	9.3882	1.5059	1.4588	1.2471	1.5	0.40	0.09	2.50
14	0.8	1.6	8.6118	1.1647	1.2627	1.3588	1.5	0.37	0.09	2.70
15	0.9	1.6	5.4588	2.4824	1.1451	0.2000	1.5	0.40	0.09	2.27
16	1	1.6	7.0118	1.3882	1.0980	1.2118	1.5	0.39	0.08	3.97
17	0.7	1.8	8.4235	2.1176	1.6784	0.5647	1.5	0.47	0.09	2.10
18	0.8	1.8	9.0118	1.6588	1.5137	1.2706	1.5	0.42	0.08	3.45
19	0.9	1.8	8.0471	2.8588	1.5216	0.2741	1.5	0.34	0.07	5.90
20	1	1.8	7.9059	1.0353	1.2627	1.1000	1.5	0.38	0.08	4.50
21	0.7	2	9.4118	1.0471	1.7961	1.2765	1.5	0.51	0.09	1.59
22	0.8	2	8.8235	1.1882	1.5922	1.0765	1.5	0.46	0.09	2.32
23	0.9	2	8.1412	2.0706	1.4510	0.1706	1.5	0.35	0.09	3.60
24	1	2	9.3176	1.5294	1.4824	1.1000	1.5	0.35	0.07	5.62
Average values of PIλDμ (AVPI$^{\lambda}$D$^{\mu}$)			7.3451	1.5868	1.1856	1.0445	1.5			

Robustness study is made by taking the average controller parameters in both the cases. The performance of the both the systems against parameter variations with fixed average $PI^{\lambda}D^{\mu}$ controller parameters (calculated in Table 3 and Table 4) is presented in Table 5, Figure 7 and Figure 8.

Table 5. Time response for fixed $PI^{\lambda}D^{\mu}$ controller parameters against parameter variations

Kg	τg	Method	Kp	Ki	Kd	λ	μ	Ts(s)	Tr(s)	Osh (%)
		RGA PID	0.7246	0.3601	0.1643	1	1	1.19	0.81	2.16
0.77	1.50	AVGAPI$^{\lambda}$D$^{\mu}$	7.3451	1.5868	1.1856	1.0445	1.5	0.97	0.1	2.07
		AVGPSOPI$^{\lambda}$D$^{\mu}$	4.7641	0.9095	0.7820	1.1310	1.3552	0.18	0.19	1.94
		RGA PID	0.6598	0.2927	0.1743	1	1	1.11	0.94	0.29
0.79	1.15	AVPI$^{\lambda}$D$^{\mu}$	7.3451	1.5868	1.1856	1.0445	1.5	0.42	0.07	6.34
		AVGPSOPI$^{\lambda}$D$^{\mu}$	4.7641	0.9095	0.7820	1.1310	1.3552	0.39	0.13	4.59
		RGA PID	0.7379	0.2862	0.1643	1	1	0.90	0.84	6.33
0.85	1.30	AVPI$^{\lambda}$D$^{\mu}$	7.3451	1.5868	1.1856	1.0445	1.5	0.41	0.07	5.69
		AVGPSOPI$^{\lambda}$D$^{\mu}$	4.7641	0.9095	0.7820	1.1310	1.3552	0.37	0.14	4.27
		RGA PID	0.6321	0.3601	0.2643	1	1	1.80	1.07	0.06
0.75	1.67	AVPI$^{\lambda}$D$^{\mu}$	7.3451	1.5868	1.1856	1.0445	1.5	1.11	0.53	2.56
		AVGPSOPI$^{\lambda}$D$^{\mu}$	4.7641	0.9095	0.7820	1.1310	1.3552	0.9	0.45	2.55
		RGA PID	0.7080	0.3601	0.1652	1	1	0.77	0.77	1.96
0.99	1.45	AVPI$^{\lambda}$D$^{\mu}$	7.3451	1.5868	1.1856	1.0445	1.5	0.38	0.07	6.79
		AVGPSOPI$^{\lambda}$D$^{\mu}$	4.7641	0.9095	0.7820	1.1310	1.3552	0.35	0.13	5.48
		RGA PID	0.6030	0.3601	0.1757	1	1	1.41	0.99	0.04
0.99	1.96	AVPI$^{\lambda}$D$^{\mu}$	7.3451	1.5868	1.1856	1.0445	1.5	1.1	0.10	2.60
		AVGPSOPI$^{\lambda}$D$^{\mu}$	4.7641	0.9095	0.7820	1.1310	1.3552	0.92	0.19	2.82

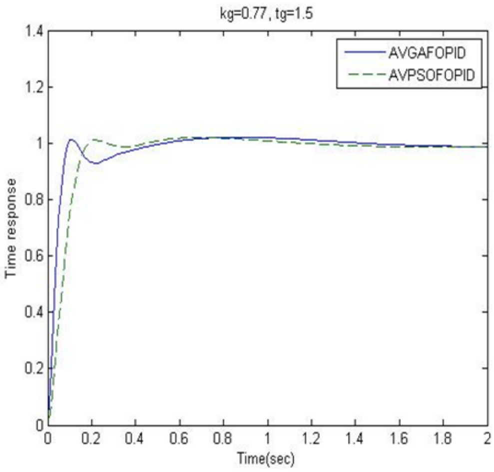

Figure 7. Time response of $GAPI^{\lambda}D^{\mu}$ and $PSOPI^{\lambda}D^{\mu}$ for fixed controller parameters (Kg=0.77,Tg=1.5)

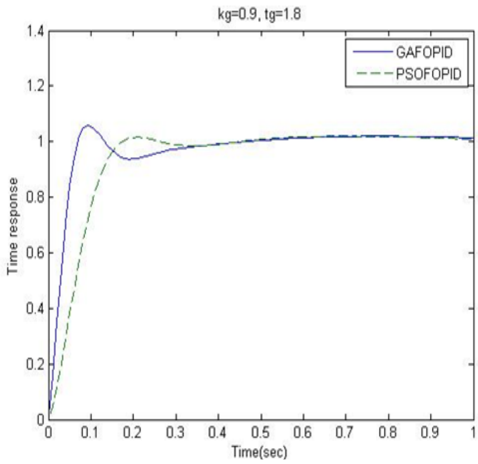

Figure 8. Time response of $GAPI^{\lambda}D^{\mu}$ and $PSOPI^{\lambda}D^{\mu}$ for fixed controller Parameters (Kg=0.9,Tg=1.8)

The results shown in Table 5, Figure 7 and Figure 8 clearly indicates, even when $PSOPI^{\lambda}D^{\mu}$ controller parameters are set fixed to average values, the time response of the AVR system is robust against parameter variations.

4. CONCLUSION

This work has been simulated in MATLAB/SIMULINK environment in combination with FOMCON tool box. The obtained results show that when fractional order PID controller is tuned with PSO algorithms, the results obtained is superior than when tuned with PSOPID and $GAPI^{\lambda}D^{\mu}$. Further robustness study shows that the AVR system is more robust with $PSOPI^{\lambda}D^{\mu}$ against parameter variations compared to $GAPI^{\lambda}D^{\mu}$ controller.

REFERENCES

[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol/issue: 64(8), pp. 759/768–759/768, 1942.
[2] A. Visioli, "Tuning of PID controllers with fuzzy logic," *2001 IEE Proceedings In Control Theory and Applications*, vol/issue: 148(1), pp. 1-8, 2001.
[3] T. L. Seng, *et al*., "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol/issue: 29(2), pp. 226-236, 1999.
[4] K. Yavarian, *et al.*, "Design of Intelligent PID controller for AVR System using an Adaptive Neuro Fuzzy Inference System," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 4(5), pp. 703-718, 2014.
[5] D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," *Addison-wisley*, 1989.
[6] Z. L Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," *IEEE Transactions on Energy Conversion*, vol/issue: 19(2), pp. 384-391, 2004.
[7] O. Ibrahim, *et al.*, "Performance Evaluation of three PID Controller Tuning Algorithm on a process plant," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 5(5), 2015.
[8] D. Devaraj and B. Selvabala, "Real-coded genetic algorithm and fuzzy logic approach for real- time tuning of proportional–integral–derivative controller in automatic voltage regulator system," *IET Generation, Transmission & Distribution*, vol/issue: 3(7), pp. 641-649, 2009.
[9] D. Maiti, *et al.*, "Tuning PID and PIλ Dμ controllers using the integral time absolute error criterion," *2008 IEEE InternationalConference on Information and Automation for Sustainability, Colombo (CL)*, pp. 457-462, 2008.
[10] A. Narang, *et al.*, "Tuning of fractional PI controllers for fractional order system models with and without time delays," *2010 American Control Conference Marriott Waterfront*, Baltimore, MD, USA June 30-July 02, 2010.
[11] Y. Chen, *et al*., "Fractional Order Control: A Tutorial," *2009 American Control Conference*, pp. 1397-1411, 2009.
[12] K. B. Oldham and J. Spanier, "The Fractional Calculus," *Academic Press*, New York, 1974.
[13] A. Oustaloup, *et al*., "Nanot: Frequencybandcomplex nonintegerdifferentiator: characterization and synthesis," *IEEE Transaction on Circuit and Systems - I: Fundamental Theory and Application*, vol/issue: 47(1), pp. 25-39, 2000.
[14] Podlubny, "Fractional Differential Equations," *Academic Press*, San Diego, 1999.
[15] N. R. Raju and P. L. Reddy, "Optimal Tuning of Fractional Order PID Controller for Automatic Voltage Regulator System through Genetic Algorithm," *International Journal of Engineering and Technology*, vol/issue: 8(2), pp. 922-927, 2016.

BIOGRAPHIES OF AUTHORS

Mr. N. Ramesh Raju was born in 1971. He received his M.Tech. from IIT Kharagpur in 1994. B.Tech. from Kaktiya University in 1993. At present he is working as Professor and Head EEE department at Siddharth Institute of Engineering & Technology, Puttur. He has 16 years of teaching experience and 4 years of industrial experience. Presently he is doing research on applications of Soft computing techniques in Control System.

Dr. P. Linga Reddy was born in 1939. He received his Ph. D. from IIT, Delhi in 1978. He received ME (PS) from Andhra University in 1965 and BE (Electrical Engineering) in 1962 from Andhra University. He was central government merit Scholar ship holder throughout his studies. He worked 33years in JNT University, Hyderabad in different capacities. He is having more than 46 years' experience in teaching profession. He is presently working as professor in K L University, Guntur, India. He published more than 30 papers in various National and International Journals and conferences. His research interest includes Power System Control System and Its application.