

Adaptive PSO based Tuning of PID Controller for an Automatic Voltage Regulator System

J. Femmy Nirmal

PG Student, Applied Electronics
St. Xavier's Catholic College of Engineering
Nagercoil, India
femmynirmal@gmail.com

D. Jeraldin Auxillia

Professor, Department of ECE
St. Xavier's Catholic College of Engineering
Nagercoil, India
jeraldin.auxillia@gmail.com

Abstract— In this paper, a Proportional-Integral-Derivative (PID) controller is designed for an Automatic Voltage Regulator (AVR) system, so that faster settling to rated voltage is ensured and the instability is avoided. AVR is a closed loop control system compensated with a PID controller. Particle Swarm Optimization (PSO) Algorithm, Simplified PSO (MOL) Algorithm, Adaptive PSO (APSO) Algorithm are used to tune the PID controller, to attain an optimal solution. MOL and APSO differ from the basic PSO in the velocity updation formula. MOL has the advantage of easy implementation and APSO has the advantage of faster convergence. Optimal control parameters are obtained by minimizing the objective function ITAE (Integral Time Absolute Error). Simulations are done to show the performance of PID controlled AVR system tuned using PSO, MOL, APSO algorithm and the results are compared. Time domain analysis and stability analysis are done using root locus and bode plots.

Index Terms— Particle Swarm Optimization, Many Optimizing Liaisons, Adaptive PSO, Local optima, Convergence.

I. INTRODUCTION

An Automatic Voltage Regulator is a device that is used to regulate the supply line voltage to a level that is safe for the equipments connected to it. AVR is mainly used in areas where the supply voltage is not stable and fluctuation of load occurs. The generator excitation system maintains generator voltage and controls the reactive power flow using an AVR [1]. The AVR system is a closed loop control system compensated with a PID or PSS controller. The Power System Stabilizer (PSS) controller is used for AVR system with high gain thyristor excitation system and has six tuning parameters. The Proportional Integral Derivative (PID) controller is used for AVR system with normal gain exciter and has three controller gains as the tuning parameters [4]. In addition to these two controllers, a Fractional Order PID (FOPID) controller, which is a generalization of the standard PID controller, can also be used for the AVR system. FOPID controller has five tuning parameters that include three controller gains, a derivative order and an integral order [7, 11]. Among these three controllers PID controllers are widely used due to its simple structure, easy implementation since it has only three tuning parameters and providing robust performance in wide range of operating conditions. Optimal tuning of PID control parameters are needed for the best performance of the system.

Previously various conventional tuning techniques such as Ziegler/ Nicholes tuning [14], Cohen/ Coon method [15], minimum variance method, gain phase margin methods were used. But these methods exhibited some demerits such as extensive rules to set the gains, difficulty to deal with nonlinear systems and complexity of control design [12]. Hence recently many evolutionary algorithms such as Genetic Algorithm (GA) [3,6,8],

Differential Evolution (DE) algorithm [10], Artificial Bee Colony (ABC) [9], Chaotic Optimization Algorithm [5], Chaotic Ant Swarm (CAS) Algorithm [11] and Particle Swarm Optimization (PSO) Algorithm [2,5] were used for tuning.

In 2004, Gaing compared the GA with Simulated Annealing and concluded that GA is faster due to its parallel search techniques, but has the disadvantage of premature convergence[3]. Coelho in 2009 proposed that the Chaotic Optimization Algorithm has the feature of easy implementation, short execution time and robust mechanism of escaping from local optima [5]. In 2012, Tang, Cui, Hua, Li and Yang used the CAS to tune PID controller and found that it has more chances to explore to global optimum in search space [11]. Gozde and Taplamacioglu in 2011 used ABC algorithm and proposed that ABC has triple search capability provided by separate artificial bee colonies [9]. In 2011, Panda proposed the DE Algorithm which is capable of handling non differentiable, nonlinear and multimodal objective function with few easily chosen control algorithms [10]. In 1995 Kennedy and Eberhart proposed a new algorithm that has root in bird flocking and swarming theory [2]. Among the entire evolutionary algorithm, PSO has the advantage that it requires only primitive mathematical operators and it is computationally inexpensive in terms of both memory requirement and speed. But it has the chance of getting trapped in local optima. To overcome this certain modifications were made in basic PSO.

In this study, variants of PSO such as Simplified PSO (MOL) and Adaptive PSO (APSO) that has the advantage of escaping from local optima are used to tune the control parameters of PID controller in the AVR system. Also MOL has the advantage of easy implementation and APSO has the advantage of faster convergence. MOL and APSO differ from the basic PSO in the velocity updation only.

II. AVR SYSTEM MODELLING

A simple AVR system comprises four main components, namely amplifier, exciter, generator, and sensor. For mathematical modeling and transfer function of the four components, these components must be linearized, which takes into account the major time constant and ignores the saturation or other nonlinearities [16,17]. The transfer function model of each component consists of a gain and a time constant and is given as

$$\text{Transfer function model of an amplifier is: } TF_A = \frac{k_a}{1 + T_a s} \quad (1)$$

$$\text{Transfer function model of an exciter is: } TF_E = \frac{k_e}{1 + T_e s} \quad (2)$$

$$\text{Transfer function model of a generator is: } TF_G = \frac{k_g}{1 + T_g s} \quad (3)$$

$$\text{Transfer function model of a sensor is: } TF_s = \frac{k_s}{1 + T_s s} \quad (4)$$

The typical parameter limits of the gain and time constants of the components and the parameter values used are given in the Table I. Using the mentioned parameter values, the transfer function model of an Automatic Voltage Regulator system is given in Fig. 1. Figure 2 shows the open loop response of the AVR system and its transfer function is

$$\frac{\Delta V_i(s)}{\Delta V_{ref}(s)} = \frac{0.1s + 10}{0.0004s^4 + 0.0454s^3 + 0.555s^2 + 1.51s + 11} \quad (5)$$

III. PID CONTROLLER DESIGN

A PID controller calculates the error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the three process control parameters. The three process control parameters include the proportional gain (k_p), integral gain (k_i) and derivative gain (k_d). The transfer function of PID controller Laplace domain is represented by,

$$\frac{u(t)}{e(t)} = k_p + \frac{k_i}{s} + k_d s \quad (6)$$

$$\frac{u(t)}{e(t)} = k_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (7)$$

where $u(t)$ is the control signal and $e(t)$ is the error signal.

In a PID controller, the proportional gain has the effect of reducing the rise time, the integral gain has the effect of eliminating the steady-state error and the derivative gain has the effect of increasing the stability of the system. These three control

TABLE I. PARAMETER LIMITS OF AVR SYSTEM [9]

	Parameter limits		Parameter values	
	Gain	Time constant	Gain	Time constant
Amplifier	$10.0 \leq k_a \leq 40.0$	$0.02 \leq T_a \leq 0.1$	$k_a = 10.0$	$T_a = 0.1$
Exciter	$1.0 \leq k_e \leq 10.0$	$0.4 \leq T_e \leq 1.0$	$k_e = 1.0$	$T_e = 0.4$
Generator	$0.7 \leq k_g \leq 1.0$	$1.0 \leq T_g \leq 2.0$	$k_g = 1.0$	$T_g = 1.0$
Sensor	$k_s = 1.0$	$0.001 \leq T_s \leq 0.06$	$k_s = 1.0$	$T_s = 0.01$

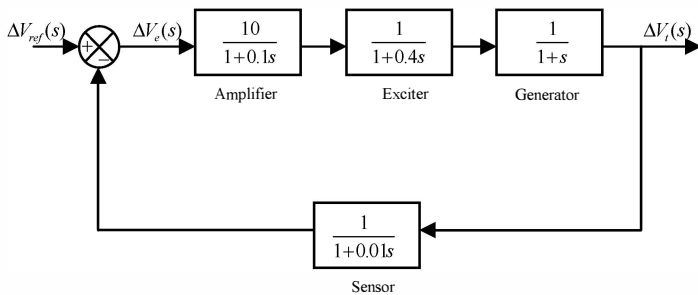


Fig. 1. Transfer function model of the AVR system without controller

parameters have to be tuned properly for the better performance of the PID controller.

A. PID Controller Tuning for AVR System

Tuning algorithms such as PSO, Simplified PSO (MOL) and Adaptive PSO (APSO) are used for tuning PID controller. Fig. 3 corresponds to the block diagram for PID controller tuning for an AVR system.

1) *Particle Swarm Optimization (PSO) Algorithm*: Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart in 1995, is one of the modern heuristic algorithms and a kind of evolutionary computation technique. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous nonlinear optimization problems [2].

In PSO algorithm, all the particles are placed at random position and are supposed to move randomly in a defined direction in the search space. Each particle's direction is then changed gradually to insist to move along the direction of its best previous positions to discover even a new better position with respect to some fitness measures [18]. Both the initial velocity and position of the particle are chosen randomly and updated iteratively using Eq. 8 and Eq. 9

$$V = wV + c_1 R_1 (p_b - X) + c_2 R_2 (g_b - X) \quad (8)$$

$$X = X + V \quad (9)$$

where V and X are the velocity and position of the particle. c_1 and c_2 are the accelerating coefficients, which are often set as $c_1 + c_2 \geq 4$. R_1 , R_2 are the random numbers between 0 and 1 and w is the inertia weight. p_b and g_b are the local best and global best respectively. Suitable selection of inertia weight w provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. The inertia weight controls the effect of previous velocities on the current velocity and is set according to Eq. 10.

$$w = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \right) \times \text{iter}_i \quad (10)$$

where w_{\max} is the maximum value of inertia weight, w_{\min} is the minimum value of inertia weight.

2) *Many Optimizing Liaisons (MOL) Algorithm*: MOL Algorithm is also called as the Simplified Particle Swarm Optimization algorithm [12]. In the MOL algorithm the swarm's best position p_b is eliminated by setting $c_1=0$ and thus Eq. 8 gets reduced as

$$V = wV + c_2 R_2 (g_b - X) \quad (11)$$

and $X = X + V$

In MOL the inertia weight w is set similar to PSO as in Eq. 10. This MOL algorithm simplifies the original PSO by randomly choosing the particle to update, instead of iterating over the entire swarm thus eliminating the particles best known position and making it easier to tune the behavioral parameters and is simple.

3) *Adaptive Particle Swarm Optimization (APSO) Algorithm*: Even though the variants of PSO improve the performance of PSO, the actual search process cannot be truly obtained without any feedback comparing the particle's fitness to

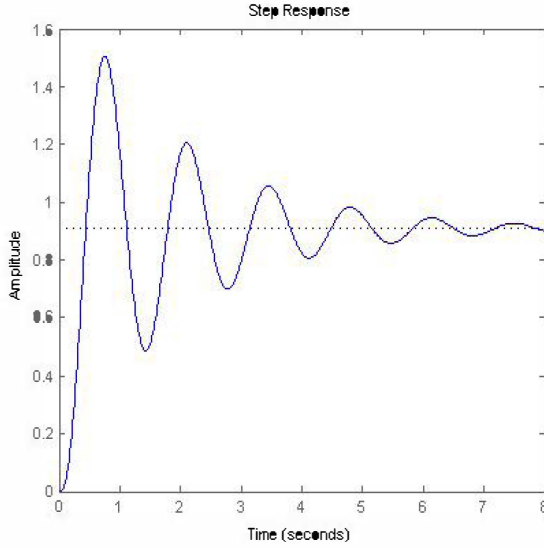


Fig. 2. Open loop response of AVR system

estimated optimal value, when the real optimal value is known in advance. If the global fitness is large, the particles are far away from the optimum point. Hence, a big velocity is needed to globally search the solution space and so w must be larger. Conversely, only small movements are needed and w can be set small [13]. Hence here the inertia weight is dynamically varied by considering the nearness of particles fitness with the optimal value, using a measure called adjacency index (AI) given as

$$AI_i^t = \frac{F(p_{bi}^t) - F_{KN}}{F(p_{bi}^t) - F_{KN}} - 1 \quad (12)$$

where $F(p_{bi})$ is the fitness of the previous position of i^{th} particle and F_{KN} is the known real optimal solution value. If AI_i is small, fitness of i^{th} particle is far away from the real optimal value and hence a large inertia weight is required. If AI_i is large, then the fitness of i^{th} particle is near the real optimal value and hence a small inertia weight is required. Based on this index, the inertia weight for every particle in t^{th} iteration can be dynamically calculated as,

$$w = \frac{1}{1 + e^{-(\alpha \times AI_i^t)}} \quad (13)$$

where α is a positive constant in the range (0,1] and it controls the decreasing speed of inertia weight. Here the velocity and position of the particle is updated using

$$V = wV + c_1 R_1(p_b - X) + c_2 R_2(g_b - X)$$

$$X = X + V$$

In APSO, as the inertia weight is varied dynamically according to the particles best fitness, it provides a better global and local exploration thus resulting in faster convergence.

IV. RESULTS AND DISCUSSION

With PID controller the closed loop transfer function of the AVR system is

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1k_d s^3 + (0.1k_p + k_d)s^2 + (0.1k_i + 10k_p)s + 10k_i}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + (1.51 + 10k_d)s^2 + (1 + 10k_p)s + 10k_i}$$

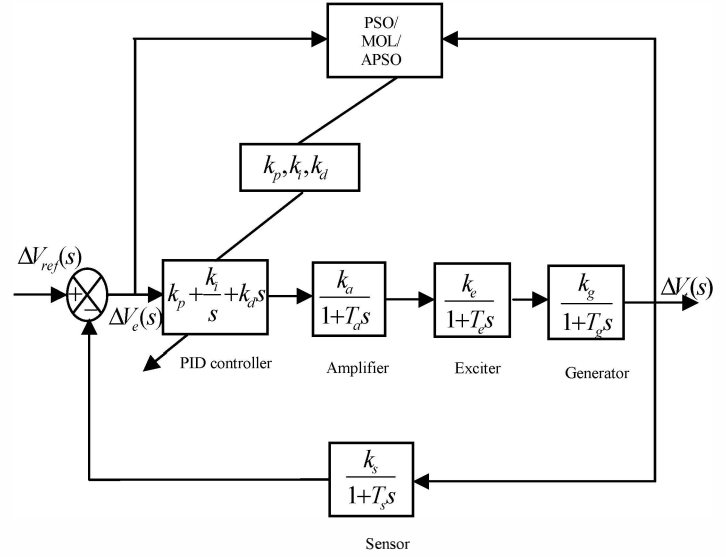


Fig. 3. Block diagram of PID controller tuning for an AVR system

PID controller gains are tuned using the conventional tuning method (Ziegler Nichole's method) and evolutionary algorithms such as PSO, MOL and APSO. For tuning the upper and lower bounds of the controller gains are set as (0.1-1). Velocity of the particle is limited to 10% of the dynamic range of the particle. The simulation parameters for various tuning algorithms are illustrated in Table II. Also the inertia weight w in PSO and MOL algorithms are set according to Eq. 10 which is dynamically varying in the range of 0.9 to 0.4. In APSO inertia weight w is set according to Eq. 13 and the real optimal value F_{KN} is set as zero [13].

For optimal tuning of the control parameters, minimizing the integral of time multiplied by absolute value of error (ITAE) given in Eq. 15 is considered as the objective function.

$$ITAE = \int_0^t |V_{ref} - V_t| dt \quad (15)$$

where V_{ref} and V_t are the reference and terminal voltage. The optimal controller gains obtained using different controller tuning methods are given in Table III and the transfer function of the system with optimal gains obtained using ZN, PSO, MOL, APSO are given in Eq. 16 – Eq. 19.

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.01469s^3 + 1.577s^2 + 10.998s + 19.8}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 2.979s^2 + 11.8s + 19.8} \quad (16)$$

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.0102s^3 + 1.052s^2 + 3.5s + 4.778}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 2.527s^2 + 4.452s + 4.778} \quad (17)$$

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.01572s^3 + 1.627s^2 + 5.567s + 4.418}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 3.082s^2 + 6.523s + 4.418} \quad (18)$$

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.0194s^3 + 1.995s^2 + 5.58s + 4.369}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 3.45s^2 + 6.536s + 4.369} \quad (19)$$

The terminal voltage curves obtained using these transfer functions is given in Fig. 4.

A. Transient Response Analysis:

Results obtained from the transient response analysis are listed

TABLE II. SIMULATION PARAMETERS FOR PSO, MOL, APSO ALGORITHMS

	Particle limit	Swarm Size	No. of iterations	Accelerating coefficients	Inertia weight
PSO	(0.1-1)	100	50	$c1=c2=2$	Linearly decreasing
MOL	(0.1-1)	100	50	$c1=0; c2=2$	Linearly decreasing
APSO	(0.1-1)	100	50	$c1=c2=2$	Dynamically varied according to adjacency index

in Table IV. Here the peak amplitude corresponds to the maximum deviation of the system voltage from its rated level during operation, and the settling time corresponds to the time taken by the system to settle back to the rated level from its deviation.

From the analysis it is observed that the peak amplitude obtained by tuning using APSO is 40% lesser than the PSO tuned system and 2% lesser than the MOL tuned system. Similarly the settling time of the system tuned using APSO is 199.6% lesser than the PSO tuned system and 63.6% lesser than the MOL tuned system.

B. Root Locus Analysis:

Figure 5 shows the root locus curve for closed loop AVR system tuned using ZN, PSO, MOL and APSO tuning algorithms. The closed loop poles and the damping ratios of the system tuned by ZN, PSO, MOL and APSO algorithms are given in Table V.

From the table it is clear that the system tuned using ZN, PSO, MOL and APSO algorithms remain stable, since all the poles lie on the left half of the s- plane. Also the damping ratio which is responsible for the faster settling is better for the AVR system tuned using APSO algorithm than ZN, PSO and MOL tuning algorithms.

C. Bode Analysis:

Bode analysis is used to analyze the frequency response of the control system. The magnitude and phase plot of the AVR system tuned using ZN, PSO, MOL and APSO algorithm are shown in Fig. 6. The gain margin, phase margin and bandwidth obtained from the Bode plots are depicted in Table VI. All the closed loop systems have positive phase margin. Maximum phase margin is obtained from APSO algorithm and this ensures better stability of the system.

TABLE III. OPTIMAL GAINS OBTAINED USING ZN, PSO, MOL, APSO TUNING TECHNIQUES

Parameters/ Tuning techniques	Ziegler/ Nichols method	PSO	MOL	APSO
k_p	1.08	0.3452	0.5523	0.5536
k_i	1.98	0.4778	0.4418	0.4369
k_d	0.1469	0.1017	0.1572	0.1940

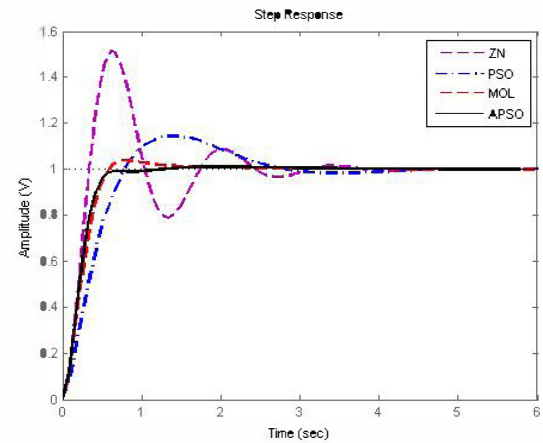


Fig. 4. Terminal voltage curve for the AVR system

TABLE IV. RESULTS OF TRANSIENT RESPONSE ANALYSIS

	Peak amplitude(V)	Settling time (sec)	Rise time (sec)	Peak time (sec)
Open loop system	1.51	6.99	0.261	0.75
ZN tuned system	1.52	2.95	0.232	0.604
PSO tuned system	1.14	2.56	0.536	1.364
MOL tuned system	1.03	1.2	0.372	0.778
APSO tuned system	1.01	0.564	0.346	1.98

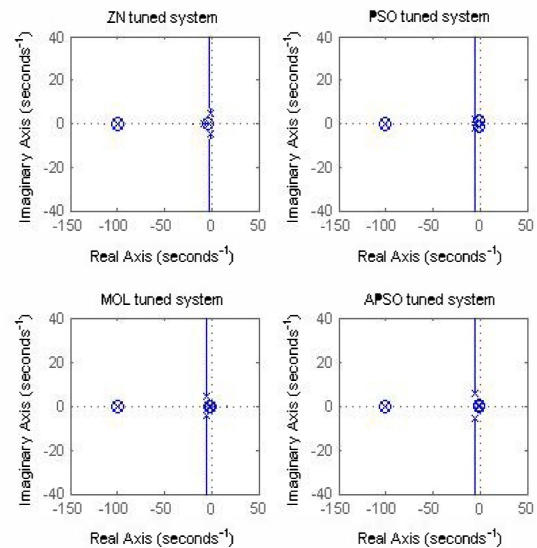


Fig. 5. Root locus curve of ZN, PSO, MOL, APSO algorithm tuned closed loop AVR system

TABLE V. CLOSED LOOP POLES AND DAMPING RATIO OF AVR SYSTEM TUNED USING ZN, PSO, MOL, APSO ALGORITHMS

System	Closed loop poles	Damping ratio
ZN tuned system	-99.2460	1
	-7.7420	1
	$-1.2827 \pm 4.4965i$	0.275
	-2.9465	1
PSO tuned system	-99.1355	1
	$-4.6722 \pm 2.0497i$	0.921
	$-1.0101 \pm 1.5140i$	0.783
MOL tuned system	-99.2908	1
	$-4.8939 \pm 3.9629i$	0.736
	-2.0586	1
	-1.3627	1
APSO tuned system	-99.3973	1
	$-5.2874 \pm 5.3381i$	0.698
	$-1.2639 \pm 0.5910i$	0.951

TABLE VI. BODE ANALYSIS

	Phase Margin (deg)	Bandwidth
ZN tuned system	51.3831	7.6975
PSO tuned system	122.6776	3.3637
MOL tuned system	151.5557	5.7334
APSO tuned system	166.9725	6.4765

TABLE VII. CONVERGENCE ANALYSIS

	PSO	MOL	APSO
Elapsed Time (sec)	169.85	167.46	165.02

D. Convergence Analysis:

The computational time of PSO, MOL and APSO algorithms are calculated to find the convergence rate of these algorithms. For measuring the computational time, a threshold of 10^{-4} is considered as the stopping condition. In addition to this a maximum iteration of 300 is also considered as the stopping condition for the cases where the threshold of 10^{-4} is not met. Then the computational time is calculated by running each algorithm 20 times and the average of the elapsed time is found.

Table VII shows the time required for each algorithm to complete its search process and converge to the global solution. From the table it is clear that APSO converges with less elapsed time compared to PSO and MOL algorithms. Thus it can be concluded that the APSO algorithm outperforms PSO and MOL algorithm in terms of accuracy and convergence speed. This faster convergence of APSO algorithm makes it more applicable for online tuning of the AVR system.

V. CONCLUSION

A PID controller tuned using APSO algorithm is designed for an Automatic Voltage Regulator system. To prove the tuning superiority of the APSO algorithm, PSO and MOL Algorithms are also used for tuning the PID controller. The optimal solutions are obtained and the performance of the PID controlled AVR system tuned using PSO, MOL and APSO algorithms are analyzed using root locus and bode plots in time and frequency domain. It is observed that the APSO algorithm has fast convergence and more accurate than PSO and MOL algorithms. From the simulation results, it is clear that the APSO algorithm outperforms PSO and MOL algorithms. APSO tuned system ensures faster settling than PSO and MOL tuned systems. Also the APSO tuned system has all its poles in the left half of s- plane and maximum phase margin, which shows that the APSO tuned system, is more stable.

REFERENCES

- [1] P. C. Sen (1987), "Power Electronics", Tata McGraw Hill Education.
- [2] J. Kennedy, R.C. Eberhart (1995), "Particles swarm optimization", in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, , pp. 1942–1948.
- [3] Z. L. Gaing (2004), "A particle swarm optimization approach for optimum design of PID controller in AVR system", IEEE transactions on energy conversion, vol. 19, no.2.
- [4] V. Mukherjee, S.P. Ghoshal (2007), "Comparison of intelligent fuzzy based AGC coordinated PID controlled and PSS controlled AVR system", Electric Power and Energy Systems 29 679–689.
- [5] L.D.S. Coelho (2009), "Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach", Chaos, Solitons and Fractals 39 (4) 1504–1514.

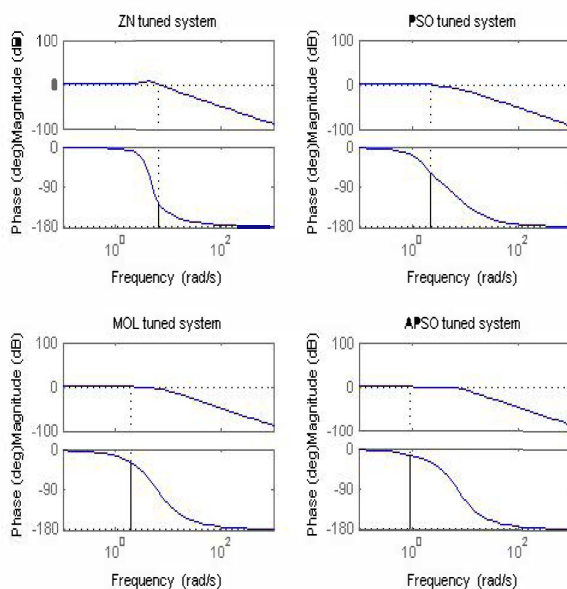


Fig. 6. Bode plot of system tuned using ZN, PSO, MOL, APSO algorithms

- [6] S. Panda, N.P. Padhy (2008), "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design", *Applied Soft Computing* 8 (4), 1418–1427.
- [7] M. Zamani, M. K. Ghartemani, N. Sadati, M. Parniani (2009), "Design of fractional order PID controller for AVR using particle swarm optimization", *Control Engineering Practice* 17(12) 1380–1387.
- [8] S. Panda (2010), "Application of non-dominated sorting genetic algorithm-II technique for optimal FACTS-based controller design", *Journal of the Franklin Institute* 347 (7) 1047–1064.
- [9] H. Gozde, M.C. Taplamacioglu (2011), "Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system", *Journal of the Franklin Institute* 348 1927–1946.
- [10] S. Panda (2011), "Differential evolution algorithm for SSSC-based damping controller design considering time delay", *Journal of the Franklin Institute* 348 (8) 1903–1926.
- [11] Y. Tang, M. Cui, C. Hua, L. Li, Y. Yang (2012), "Optimum design of fraction order PID controller for AVR system using chaotic ant swarm", *Expert Systems with Applications* 39 6887–6896.
- [12] S. Panda, B.K. Sahu, P.K. Mohanty (2012), "Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization" *Journal of the Franklin Institute* 349 (8), 2609–2625.
- [13] Alfi, H. Modares (2011), "System identification and control using adaptive particle swarm optimization" *Applied Mathematical Modelling* 35, 1210–1221.
- [14] J. G. Ziegler, N. B. Nichols (1942), "Optimum settings for automatic controllers", *Transactions on ASME*, 64(8), 759–768.
- [15] G. H.Cohen, G. A. Coon (1953), "Theoretical considerations of retarded control", *Transactions on ASME*, 75, 827–834.
- [16] P. Kundur, *Power System Stability and Control*, TMH, 8th reprint, 2009.
- [17] O.I. Elgerd, *Electric Energy Systems Theory. An Introduction*, Tata McGraw-Hill, New Delhi, 1983.
- [18] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, 1998.