International Journal For Technological Research In Engineering
Volume 10, Issue 4, December-2022 *ISSN (Online): 2347 - 4718*

OPTIMIZATION OF PID CONTROLLER BY PSO TECHNIQUE FOR LOAD FREQUENCY CONTROL

Mr. Lakshman Kumar[1], Prof. Nishi Singh[2]
[1]Research Scholar, M.Tech (Power System), [2]Asst. Prof
Department of Electrical & Electronics Engineering, RNTU Bhopal

Abstract: In this thesis, a classical PID tuning methods and Optimal PID tuning methods based on performance indices are discussed and they are implemented for the selected Power system Model to control the load frequency. The Model is Designed in MATLAB software taken from the Literature survey and the same model is learnt in Under graduate course of Power system Modelling. The model is based on Transfer Function Approach in Simulation form. Their results are observed and discussed. PID controller is compared with Particle Swarm Optimization (PSO) algorithm-based optimization. The results of open loop and close PID tuning methods are observed to note overshoot, rise time, settling time while making choice between two classical categories of PID tuning.

Optimal PID Tuning technique based on PSO algorithm is applied for obtaining optimal values for PID parameters for better system performance. Performance indices provide flexibility to range of rise time, settling time along with 0 overshoot and thus proves better as compared to the conventional PID Controller.
After classical approach and Optimal Tuning method, Using PSO, we can achieve 0 overshoot with good rise time and settling time values.

Key words: PID Controller, Load Frequency Control Model, MATLAB software, PSO Algorithm

1. INTRODUCTION

According to practical point of view, the load frequency control problem of interconnected power system is much more important than the isolated (single area) power systems. Whereas the theory and knowledge of a isolated power system is equally important for understanding the overall view of interconnected power system.
Generally, now days all power systems are tied with their neighboring areas and the Load Frequency Control Problem become a joint undertaking. Some basic operating principle of an interconnected power system is written below:
1. The loads should strive to be carried by their own control areas under normal operating conditions, except the scheduled portion of the loads of other members, as mutually agreed upon.
Each area must have to agree upon adopting, regulating, control strategies and equipment which are beneficial for both normal and abnormal conditions
The two main objective of Load Frequency Control (LFC) are

1. To maintain the real frequency and the desired power output (megawatt) in the interconnected power system.
2. To control the change in tie line power between control areas.

2. MAJOR DRAWBACKS OF CONVENTIONAL INTEGRAL CONTROLLER

The drawbacks can be summarized as

1. They are very slow in operation.
2. There is some inherent nonlinearity of different power system components, which the integral controller does not care. Governor dead band effects, generation rate constraints (GRCs) and the use of reheat type turbines in thermal systems are some of the examples of inherent nonlinearities.
3. While there is continuously load changes occur during daily cycle, this changes the operating point accordingly. It is generally known as the inherent characteristic of power system. For good results the gain of the integrator should has to be changed repeatedly according to the change in operating point. Again, it should also be ensured that, the value of the gain compromises the best between fast transient recovery and low overshoot in case of dynamic response. Practically to achieve this is very difficult. So basically, an integral controller is known as a fixed type of controller. It is optimal in one condition but at another operating point it is unsuitable.

Therefore, the control rule applied should be suitable with the dynamics of power system. So, an advance controller would be suitable for controlling the system.

2.1 Need of Advance Control Technique

Implementation of advanced control technique provides great help in LFC of power systems. Now days there are more complex power systems and required operation in less structured and uncertain environment. Similarly innovative and improved control is required for economic, secure and stable operation. Advance control techniques are having the ability to provide high adaption for changing conditions. They are having the ability for making quick decisions. Optimal control pole placement, Linear Quadratic Regulator, Linear Quadratic Gaussian), Robust Control, sliding mode control, Internal Model Control are some examples of advanced control techniques. LQR, LQG, IMC has been used here for LFC of power system.

2.2 PID

PID is a feedback-based controller which gets the error value and calculates the output based on the characteristics of the error. it is very widely used in plants as it is simple and gives good result.

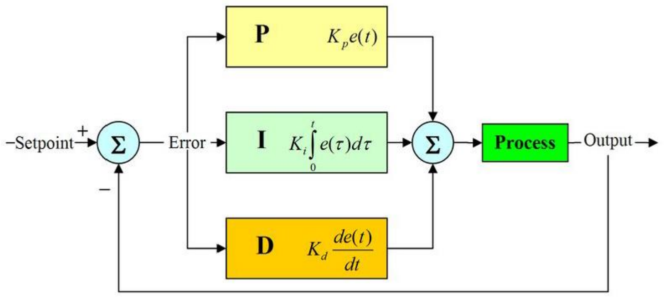


Fig .1 PID Controller

PID is used in a closed loop .it has three elements P, I, D. Every parameter has gain by which we control the contribution.

PID ALGORITHM

Equation
Were
Pout: Proportional term of output
Kp: Proportional gain, a tuning parameter
Ki: Integral gain, a tuning parameter
Kd: Derivative gain, a tuning parameter
e: Error = SP − PV
t: Time or instantaneous time (the present)

Proportional term
The proportional term makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain.
The proportional term is given by:
Equation………
Derivative term
The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, Kd.
The derivative term is given by:
Equation ……….
Integral term
The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.
The integral term is given by:
Equation ………

2.3 PID TUNING

Performance of PID depends on the gain parameters. so, we need to adjust them. Different methods are used
i) open loop method ii) close loop method
Here we apply a step to the process and get the response like as shown in the graph and get the dead time, reaction rate and process gain.

- Put the controller in manual mode
- Wait until the process value (Y) is stable and not changing
- Step the output of the PID controller - The step must be big enough to see a significant change in the process value. A rule of thumb is the signal to noise ratio should be greater than 5.
- Collect data and plot as shown below.
- Repeat making the step in the opposite direction.
- K = the process gain=change in process value /change in manipulated value

2.4 PARTICLE SWARM OPTIMIZATION Algorithm for PSO

-The ith particle in the swarm is represented as
Xi = (xi1 , xi2, xi3,..................xid) in the d-dimensional space.
-The best previous positions of the ith particle is represented as: Pbest = (Pbesti,1 ,Pbesti,2 ,Pbesti,3..........Pbesti,d)
-The index of the best particle among the group is Gbestd.
-Velocity of the ith particle is represented as Vi = (Vi,1 Vi,2 Vi,3.......... Vi,d).
-The updated velocity and the distance from Pbestid to Gbesti,d is given as ;
m = 1,2,3....d.

where,

n:- Number of particles in the group. d:- dimension index.
t:- Pointer of iteration.
C1 , C2 :- Acceleration Constant.
rand() :- Random number between 0 and 1.
Pbest :- Best previous position of the ith particle.
Gbest:- Best particle among all the particle in the swarming population.
2.4.1 Algorithmic Approach for the Specified Design: -
In our case, we cast the PID controller design problem in PSO framework as given. We consider the three-dimensional search space. KP , KI and KD are the three dimensions. We consider the fitness function based on time domain characteristics for adaptation. We set the number of adaptation iterations based on expected parameters and time of computation.
2.4.2 A Small Illustration of Program: -
-Initially we fixed the values of PSO algorithm constants as:
Inertia weight factor W = 0.3
Acceleration constants C1 , C2 = 1.5
-As we have to optimize three parameters, namely KP ,KD ,KI of the controller, we have to search for their optimal value in the three dimensional search space, so we randomly initialized a swarm of population “100” in the three dimensional search

space with [Xi,1 Xi,2 Xi,3] and [Vi1 Vi2 Vi3] as initial position
and velocity.

-Calculated the initial fitness function of each point and the point with minimum fitness function is displayed as gbest (initial value of global best optima) and the optimal fitness function as fbest1(Initial best fitness function).
-Runned the program with the PSO algorithm with thousands (or even more numbers) of iterations and the program returned final optimal value of fitness function as "fbest" and final global optimum point as "Gbest".

3. SIMULATION BLOCK REPRESENTATION MODEL

- The Power system is modelled in MAtlab Software after reviewing the litreature survey and associate work. In the below Power system model there are two parts (i) Generator Model with PID controller Technique (ii) Generator Load frequency Control Model Using PSO technique .
- Responses of both are compared using a single scope in MATLAb by Bus selector Option.In figure below 3 the MATLAB modelling is shown. The same analysis of Power system model is given in Every power system books and in this research the work for load frequency control is implemented in MATLAB Software.
- In figure 4 the blocks are highlighted were PSO is Implemented and PID controller.

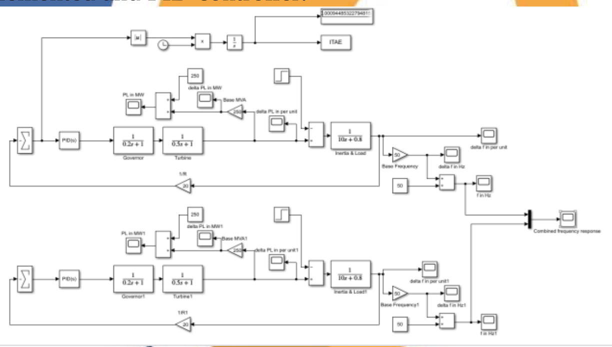


Figure 3 MATLAB Model of LOAD FREQUENCY CONTROL SYSTEM

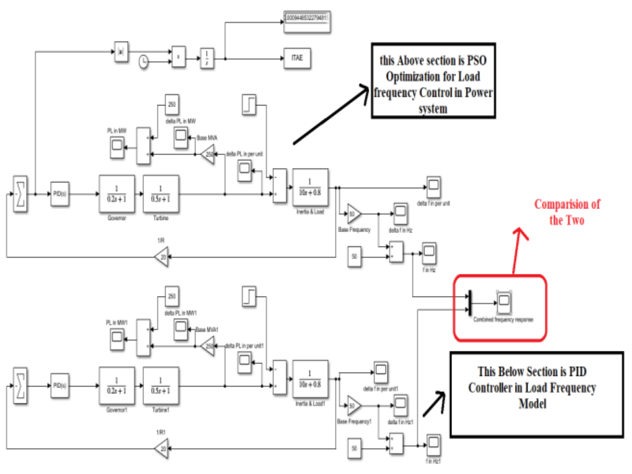

this Above section is PSO Optimization for Load frequency Control in Power system
Comparision of the Two
This Below Section is PID Controller in Load Frequency Model

Figure 4 MATLAB Model of LOAD FREQUENCY CONTROL SYSTEM with highlights of Implementation Explanation

4. Result Discussion

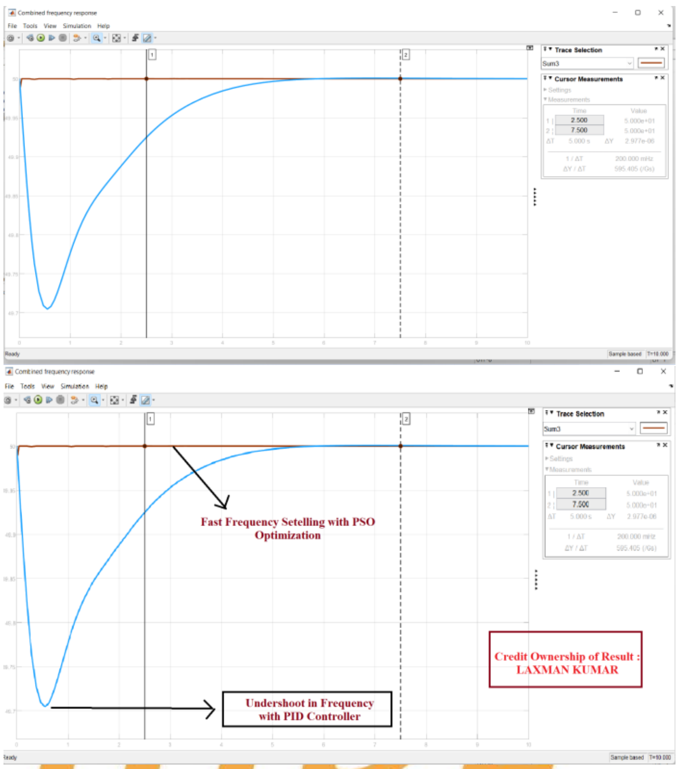

Fast Frequency Setelling with PSO Optimization
Undershoot in Frequency with PID Controller
Credit Ownership of Result : LAXMAN KUMAR

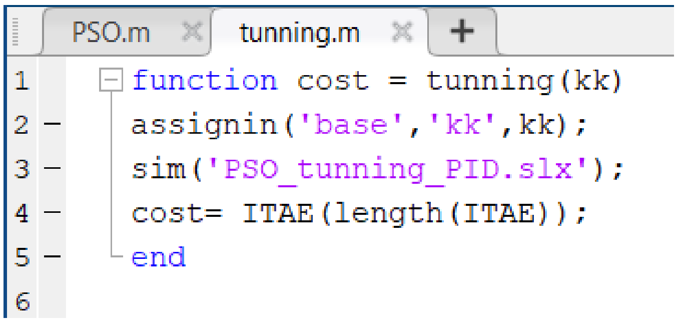
Figure 5. MATLAB Model Results outcome of PSO and PID controller for LOAD FREQUENCY CONTROL SYSTEM [maroon red PSO and blue is Conventional PID Controller

4.1 Result Explanation

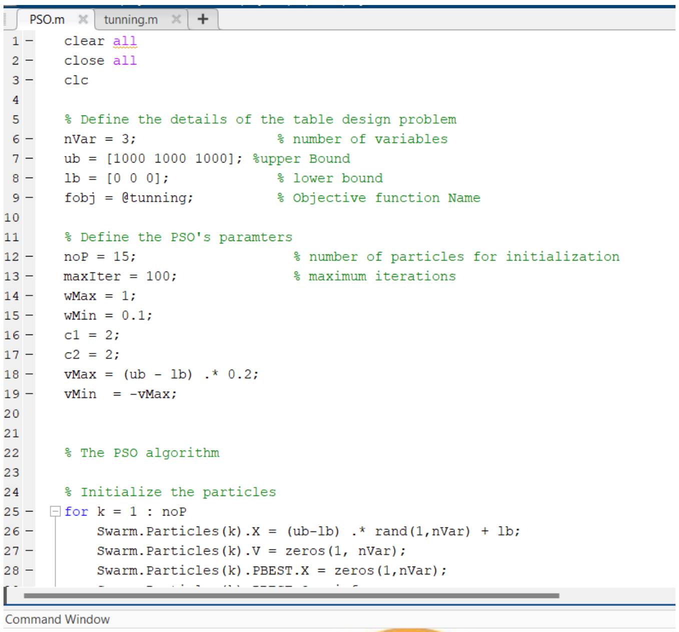
- In control system there are some technical terms which decides the performance of Model like Settling time, Over and Undershoot, Response Time and Steady state error. In figure 5. two graphs are shown in which the maroon color is for PID controller and blue Color is for Optimized PID controller using PSO Technique.
- it can be clearly understood that conventional PID controller takes a undershoot and settles to 50HZ after 5.5 seconds were as in PSO optimization technique the frequency settles immediately with going to undershoots.
- The above discussion proves the stable behavior of PSO instead of PID Controller.
- Below the codes are given for Implementing any Power system Model were PID controller needs an Optimized value depending upon any selected system values irrespective of anything as the values of PID controller constants Kp, Ki and Kd are automatically recovered after running the PSO algorithms
- It should be noted while modelling that the File name should be same of the model which is Implemented in the coding and while tunning the PSO and Model.
- PSO takes 100 Iterations and after that it Provides and Optimized value of Kp ,Ki, and Kd . the iterations are shown in the later section of this chapter where it can be seen that at last all the values merged at same values upto 8th decimal place.

5. PSO ALGORITHM IMPLEMENTED IN MATLAB

```
clear all
close all
clc

% Define the details of the table design problem
nVar = 3;               % number of variables
ub = [1000 1000 1000]; %upper Bound
lb = [0 0 0];           % lower bound
fobj = @tunning;         % Objective function Name

% Define the PSO's paramters
noP = 15;                 % number of particles for initialization
maxIter = 100;            % maximum iterations
wMax = 1;
wMin = 0.1;
c1 = 2;
c2 = 2;
vMax = (ub - lb) .* 0.2;
vMin  = -vMax;


% The PSO algorithm

% Initialize the particles
for k = 1 : noP
   Swarm.Particles(k).X = (ub-lb) .* rand(1,nVar) + lb;
   Swarm.Particles(k).V = zeros(1, nVar);
   Swarm.Particles(k).PBEST.X = zeros(1,nVar);
   Swarm.Particles(k).PBEST.O = inf;

   Swarm.GBEST.X = zeros(1,nVar);
   Swarm.GBEST.O = inf;
end


% Main loop
for t = 1 : maxIter

   % Calcualte the objective value
   for k = 1 : noP
      currentX = Swarm.Particles(k).X;
      Swarm.Particles(k).O = fobj(currentX);

      % Update the PBEST
      if Swarm.Particles(k).O < Swarm.Particles(k).PBEST.O
         Swarm.Particles(k).PBEST.X = currentX;
         Swarm.Particles(k).PBEST.O = Swarm.Particles(k).O;
      end

      % Update the GBEST
      if Swarm.Particles(k).O < Swarm.GBEST.O
         Swarm.GBEST.X = currentX;
         Swarm.GBEST.O = Swarm.Particles(k).O;
      end
   end

   % Update the X and V vectors
   w = wMax - t .* ((wMax - wMin) / maxIter);

   for k = 1 : noP
      Swarm.Particles(k).V = w .* Swarm.Particles(k).V + c1
.* rand(1,nVar) .* (Swarm.Particles(k).PBEST.X -
Swarm.Particles(k).X) ...
                                                  + c2 .*
rand(1,nVar) .* (Swarm.GBEST.X - Swarm.Particles(k).X);


      % Check velocities
      index1 = find(Swarm.Particles(k).V > vMax);
      index2 = find(Swarm.Particles(k).V < vMin);

      Swarm.Particles(k).V(index1) = vMax(index1);
      Swarm.Particles(k).V(index2) = vMin(index2);

      Swarm.Particles(k).X = Swarm.Particles(k).X +
Swarm.Particles(k).V;

      % Check positions
      index1 = find(Swarm.Particles(k).X > ub);
      index2 = find(Swarm.Particles(k).X < lb);

      Swarm.Particles(k).X(index1) = ub(index1);
      Swarm.Particles(k).X(index2) = lb(index2);

   end

   outmsg = ['Iteration# ', num2str(t) , ' Swarm.GBEST.O = ' ,
num2str(Swarm.GBEST.O)];
   disp(outmsg);

   cgCurve(t) = Swarm.GBEST.O;
end

semilogy(cgCurve);
xlabel('Iteration#')
ylabel('Weight')
```



```
function cost = tunning(kk)
assignin('base','kk',kk);
sim('PSO_tunning_PID.slx');
cost= ITAE(length(ITAE));
end
```



```
clear all
close all
clc

% Define the details of the table design problem
nVar = 3;                    % number of variables
ub = [1000 1000 1000]; %upper Bound
lb = [0 0 0];                % lower bound
fobj = @tunning;             % Objective function Name

% Define the PSO's paramters
noP = 15;                       % number of particles for initialization
maxIter = 100;                  % maximum iterations
wMax = 1;
wMin = 0.1;
c1 = 2;
c2 = 2;
vMax = (ub - lb) .* 0.2;
vMin  = -vMax;


% The PSO algorithm

% Initialize the particles
for k = 1 : noP
    Swarm.Particles(k).X = (ub-lb) .* rand(1,nVar) + lb;
    Swarm.Particles(k).V = zeros(1, nVar);
    Swarm.Particles(k).PBEST.X = zeros(1,nVar);
```

Figure .6 Codes of PSO Implemented in MATLAB Model of LOAD FREQUENCY CONTROL SYSTEM

Iterations obtained after Running PSO in MATLAB

```
Iteration# 1 Swarm.GBEST.O = 0.00070903
Iteration# 2 Swarm.GBEST.O = 0.00070903
Iteration# 3 Swarm.GBEST.O = 0.00070903
Iteration# 4 Swarm.GBEST.O = 0.00059707
Iteration# 5 Swarm.GBEST.O = 0.00059707
Iteration# 6 Swarm.GBEST.O = 0.00057946
….
……………
……………
………………
Iteration# 95 Swarm.GBEST.O = 0.00057567
Iteration# 96 Swarm.GBEST.O = 0.00057567
Iteration# 97 Swarm.GBEST.O = 0.00057567
Iteration# 98 Swarm.GBEST.O = 0.00057567
Iteration# 99 Swarm.GBEST.O = 0.00057567
Iteration# 100 Swarm.GBEST.O = 0.00057567
```

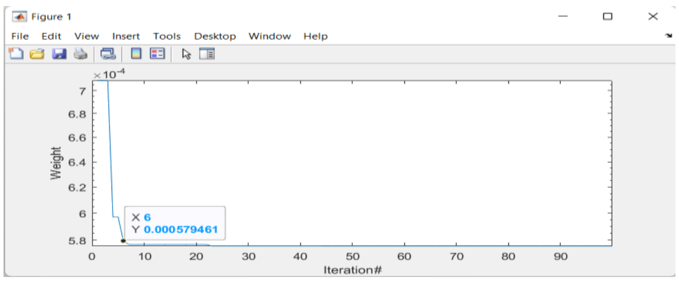


Figure .7 Iterations Submerging Graph

6. CONCLUSION

In this thesis, a classical PID tuning methods and Optimal PID tuning methods based on performance indices are discussed and they are implemented for the selected Power system Model to control the load frequency. The Model is Designed in MATLAB software taken from the Literature survey and the same model is learnt in Under graduate course of Power system Modelling. The model is based on Transfer Function Approach in Simulation form. Their results are observed and discussed. PID controller is compared with Particle Swarm Optimization (PSO) algorithm-based optimization. The results of open loop and close PID tuning methods are observed to note overshoot, rise time, settling time while making choice between two classical categories of PID tuning.

Optimal PID Tuning technique based on PSO algorithm is applied for obtaining optimal values for PID parameters for better system performance. Performance indices provide flexibility to range of rise time, settling time along with 0 overshoot and thus proves better as compared to the conventional PID Controller.
After classical approach and Optimal Tuning method, Using PSO, we can achieve 0 overshoot with good rise time and settling time values.

7. FUTURE WORK

In Future BFO Algorithm can also be used. BFO algorithm calculates optimal PID values much faster than PSO. PSO on the other hand is much simpler in structure as compared to BFO.
PSO and BFO can be further modified for more accurate adaptation of Swarm Intelligence for further studies in this field as well as other fields also. Micro-BFO algorithm, Smart BFO algorithm are some of the modified BFO approaches with fast speed and better performance and can be used for DC motor model speed control. A hybrid approach combining the best qualities of PSO and BFO algorithms can also be applied. Here ideal parallel PID controller form is used. This work can be extended further by using other related PID controller configurations like series PID, ideal PID with first order lag etc.. PSO and BFO are very good performers when time domain analysis is considered, but their performance for frequency domain and robustness analysis is not very good. So,
this can be taken as a research work in future so that PSO and BFO gives optimum performance in all three fields of time domain, frequency domain and robustness analysis.

REFERENCE

[1] A.J. Wood and B.F. Wollenberg, Power Generation, Operation and Control, 2nd edition, Wiley, New York, 1996
[2] A. Bakirtzis, V. Petridis and S. Kazarlis, “Genetic Algorithm solution of economic dispatch problem,” Proc. Inst. Elect. Eng. –Gen. Transm. Dist., vol. 141, no. 4, pp-377-382, July-1994.
[3] F. N. Lee and A. M. Breipohl, “Reserve constrained economic dispatch with prohibited operating zones”, IEEE Trans. Power syst., Vol. 8, no.1, pp. 246-254, 1993
[4] C. T. Su and G. J. Chiou, “A fast-computation Hopfield method to economic dispatch of power system,” IEEE Trans. Power System, vol. 12, pp. 1759-1764, Nov. 1997
[5] T. Yalcinoz and M. J. Short., “Neural networks approach for solving economic dispatch problem with transmission capacity constraints,” IEEE Trans. Power System, vol. 13, pp. 307-313, May. 1998

[6] C. T. Su andC. T. Lin, "New approach with a Hopfield Modelling framework to economic dispatch", IEEE Trans. Power syst., vpl 15, no. 2, p.541, May 2000.
[7] D.C. Walter, G.B. Sheble, "Genetic Algorithm solution of Economic Dispatch with valve point loading", IEEE Trans. Power syst. Vol. 8, No.3, pp 1325-1332,1993
[8] K. P. Wong and Y. W. Wong, "Genetic and genetic/simulated annealing approach to economic dispatch," Proc. Inst. Elect. Eng. Pt. C, vol.141, no. 5, pp. 507-513, Sept. 1994
[9] G. B. Sheble and K. Britting, " Refined genetic algorithm- economic dispatch example," IEEE Trans. Power syst., vpl 10, pp. 117-124, Feb. 1995.
[10] N. Sinha, R. Chakrabarti and P. K. Chattopadhyay, "Evolution Programming techniques for economic load dispatch", IEEE Trans. on Evolutionary Computations, Vol. 7, No.1, pp. 83-94, Feb. 2003.
[11] H.T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary Programming based economic dispatch for units with non-smooth fuel cost function", IEEE Trans on Power system, Vol. 11, no.1, pp 112-118, Feb. 1996.
[12] W.M. Lin, F.S. Cheng and M.T. Tsay, "An improved Tabu search for economic dispatch with multiple minima," IEEE Trans. Power Syst., 17 (February (1)) (2002), pp. 108–112.
[13] P. Attaviriyanupap, H. Kita, E. Tanaka and J. Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function," IEEE Trans. Power Syst., 17 (May (2)) (2002), pp. 411–416.
[14] A. Bhattacharya, P. Chattopadhyay, "Biogeography based optimization for different economic load dispatch problems", IEEE Trans Power Syst., Vol 25, pp no. 1064-1077, 2010.
[15] P.K. Chattopadhyay & A Bhattacharya, "A Modified Particle Swarm Optimization for Solving the Non-Convex Economic Dispatch", IEEE Trans Power system, vol-7,pp 11-15, 2009
[16] V. Hosseinnezhad, M.T. Hagh, E Babaei, "Quantum Particle Swarm Optimization for Economic Dispatch Problem with valve-point effect", Electrical Engineering (ICEE) 2011 19th Iranian Conference, IEEE 2011,pp 1-4
[17] P. Sriyanyong, "Solving Economic Dispatch Using Particle Swarm Optimization Combined with Guassian Mutation". ECTI-CON 2008, 5th International Conference, vol 2, pp 885-888, 2008 (IEEE Publication:2008, pp 885-888, vol 2)
[18] A. I. Selvakumar (IEEE member), K. Thanushkodi, "A New Particle Swarm Optimization Solution to Non-Convex Economic Dispatch Problems", IEEE Trans on Power syst. Vol. 22 No.1 February 2007

Bibliography

Mr. Lakshman Kumar I am a Research Scholar for MTech (Power System), In the Department of Electrical & Electronics Engineering RNTU, Bhopal, my areas of work are related to power quality issues in power system.	
Prof Nishi Singh I am working as Asst. Prof, in Department of Electrical & Electronics Engineering RNTU Bhopal, I have focused my learning in power Quality issues in power system and also in hybrid areas of wind and solar connected system	