MULTIOBJECTIVE OPTIMAL POWER PLANT OPERATION USING PARTICLE SWARM OPTIMIZATION TECHNIQUE

Jin S. Heo*, Kwang Y. Lee and Raul Garduno-Ramirez*****

** Department of Electrical Engineering,*
The Pennsylvania State University, University Park, PA 16802, U.S.A.
(email: juh138@psu.edu)
*** Department of Electrical Engineering,*
The Pennsylvania State University, University Park, PA 16802, U.S.A.
(email: kwanglee@psu.edu)
**** Electric Research Institute,*
Cuernvaca, Morelos 62490, Mexico (email: rgarduno@iie.org.mx).

Abstract: Multiobjective optimal power plant operation requires an optimal mapping between unit load demand and pressure set-point in a Fossil Fuel Power Unit (FFPU). In general, the optimization problem with varying unit load demand cannot be solved using a fixed nonlinear mapping. This paper presents a modern heuristic method, Particle Swarm Optimization (PSO), to realize the optimal mapping by searching the best solution to the multiobjective optimization problem, where the objective functions are given with preferences. This optimization procedure is used to design the reference governor for the control system. The approach provides the means to specify optimal set-points for controllers under a diversity of operating scenarios. Moreover, the PSO makes it possible for the optimization process to be implemented on-line in the operation of the FFPU.

Keywords: Multiobjective optimization, power plant control, set-points, particle swarm optimization.

1. INTRODUCTION

In recent years, reliable supply of electric power has been challenged severely since accidental blackouts and environmental impacts cause many critical problems in the society. Furthermore, stringent requirements on conservation and life extension of major equipment of power plants have to be fulfilled. To solve these problems, various mathematical approaches have been suggested for multiobjective optimization of power plant, such as minimization of load tracking error, minimization of fuel consumption and heat rate, maximization of duty life, minimization of pollutant emissions, etc.

First of all, the Fossil Fuel Power Unit (FFPU) must meet the load demand of electric power at all the time, at constant voltage and at constant frequency (Elgerd, 1971). Although a typical daily cycle exists on the load demand for the FFPU, a control system basically has to provide optimized wide-range cyclic operation, by being able to follow any given unit load demand. In order to realize the wide-range operation, a set-point scheduler is used by mapping demand for power and pressure from the given unit load demand. Both multiobjective optimization and a set-point scheduling are achieved through optimal mapping between the given unit load demand and pressure set-point scheduling. In general, a fixed nonlinear mapping does not allow for process optimization under operating conditions different from the originals. Moreover, the optimization process has to be implemented in the on-line operation of the FFPU.

This paper presents a modern heuristic method, Particle Swarm Optimization (PSO), for the multiobjective optimal power plant operation. Basically, the PSO has been developed for nonlinear continuous optimization problem based on the experience gained from the study of artificial life and psychological researches. Eberhart and Kennedy (1995) developed the PSO based on the analogy of the swarm of bird and the school of fish (Lee and El-Sharkawi, 2002). One of the main researches is to examine how natural creatures behave as a swarm and to reconfigure the swarm model inside a computer. The basic PSO and its variations have been applied to many engineering applications for the optimization (Park, *et al*., 2003; Eberhart and Shi, 2000). In FFPU, the swarm is consisting of agents, which are components of the control system. Each agent searches for the best solution in the solution space with given rules and informs its performance to other agents. The agent is expressed as a vector in the solution space, which is a set of control inputs. Thus, it will be shown that the PSO technique can be successfully applied to the multiobjective power plant optimization problem. Furthermore, it will be shown that on-line implementation of the PSO is also possible.

Following the introduction, the power plant control system is described in Section 2. Section 3 describes the multiobjective optimization technique (PSO). Section 4 shows simulation results to demonstrate the feasibility of the proposed approach. The final section draws some conclusions.

2. POWER PLANT CONTROL SYSTEM

2.1 Control Structure

In order for the control system to have more stable and faster response to load changes, this paper uses the coordinated control scheme (CCS), which requires references (or set-points) for both power demand (E_d) and pressure demand (P_d) (Gery, 1988). The control structure is shown in Fig. 1, where the controller is developed in three main modules: reference governor, feedforward controller, and feedback controller. The multiobjective optimization is performed in the reference governor. The results of the multiobjective optimization are the set points for the power and pressure (E_d and P_d) for the feedforward and feedback controllers. The outputs of the two controllers are added to become input to the FFPU. The output of the FFPU is fed back to the feedback controller, which regulates the output variations due to load disturbances and compensates for the variation in load demand.

2.2 Power Unit Model

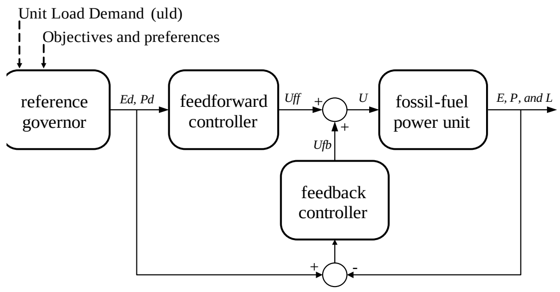

Fig. 1. Coordinated controller structure for FFPU.

The FFPU under study is a 160MW oil-fired drum-type boiler-turbine generator unit. It is represented by a third order Multiple Input-Multiple Output (MIMO) nonlinear model with three inputs and three outputs (Bell and Åström, 1987). The inputs are positions of valve actuators that control the mass flow rates of fuel (u_1 in pu), steam to the turbine (u_2 in pu), and feedwater to the drum (u_3 in pu). The outputs are electric power (E in MW), drum steam pressure (P in kg/cm2), and drum water level deviation (L in m). The state variables are electric power (E), drum steam pressure (P), and fluid (steam-water) density (ρ_f). The state equations are:

$$\frac{dP}{dt} = 0.9u_1 - 0.0018u_2P^{9/8} - 0.15u_3 \qquad (1,a)$$

$$\frac{dE}{dt} = ((0.73u_2 - 0.16)P^{9/8} - E)/10 \qquad (1,b)$$

$$\frac{d\rho_f}{dt} = (141u_3 - (1.1u_2 - 0.19)P)/85 \qquad (1,c)$$

The drum water level output is calculated using the following algebraic equations:

$$q_e = (0.85u_2 - 0.14)P + 45.59u_1 - 2.51u_3 - 2.09 \quad (2.a)$$

$$\alpha_s = (1/\rho_f - 0.0015)/(1/(0.8P - 25.6) - 0.0015) \quad (2.b)$$

$$L = 50(0.13\rho_f + 60\alpha_s + 0.11q_e - 65.5) \qquad (2,c)$$

where α_s is the steam quality, and q_e is the evaporation rate (kg/sec). Positions of valve actuators are constrained to [0,1], and their rates of change (pu/sec) are limited to:

$$-0.007 \le du_1/dt \le 0.007 \qquad (3,a)$$

$$-2.0 \le du_2/dt \le 0.02 \qquad (3,b)$$

$$-0.05 \le du_3/dt \le 0.05 \qquad (3,c)$$

2.3 Operating Windows

In order to get optimal solution, the solution space must be predefined from the given model and constraints, (1)-(3). The solution space is obtained using power-input operating windows which are driven by the inverse steady-state equations:

$$u_1 = (0.0018u_2P^{9/8} + 0.15u_3)/0.9 \qquad (4,a)$$

$$u_2 = (0.16P^{9/8} + E)/0.73P^{9/8} \qquad (4,b)$$

$$u_3 = ((1.1u_2 - 0.19)P)/141 \qquad (4,c)$$

However, the inverse steady-state equations require the relationship between power and pressure. For this

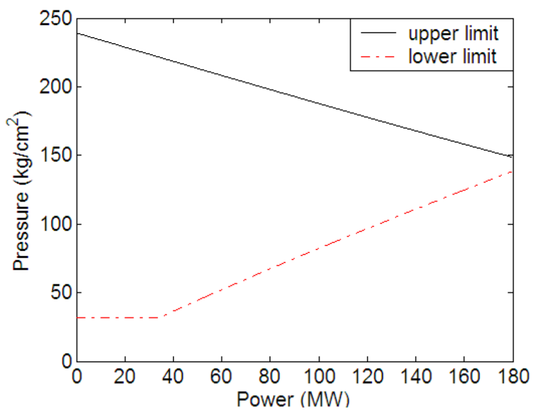

Fig. 2. Power-pressure operating window.

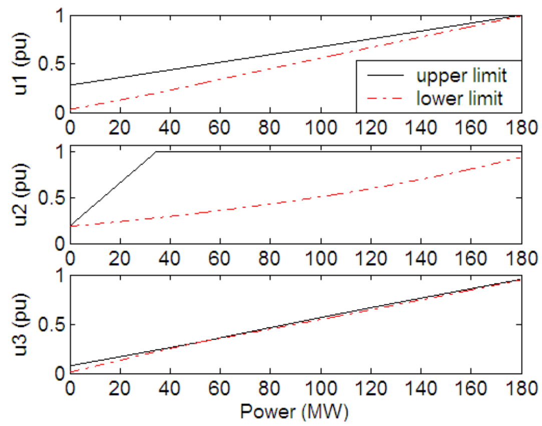

Fig. 3. Power-input operating windows.

relationship, the equilibrium points need to be found from the given model and constraints. First, solve the given model equation (1) by setting the derivatives equal to zero and find all possible and meaningful equilibrium points. The resulting power-pressure operating window is shown in Fig. 2, which is represented by upper and lower limits. Secondly, determine the power-input operating windows by solving the inverse steady-state model (4) for all points in the power-pressure operating window. Fig. 3 shows the power-input operating windows corresponding to various power ranges in the solution space for the optimization.

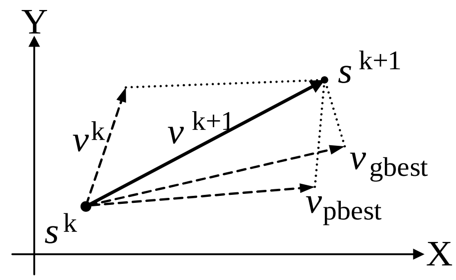

Fig. 4. Concept of modification of a search point by PSO.

3. MULTIOBJECTIVE OPTIMIZATION

3.1 Overview of the Basic PSO

Basically, the PSO is developed through simulation of birds flocking in two-dimensional space (Reynolds, 1987). The position of each bird (called agent) is represented by a point in the X-Y coordinates and also the velocity is similarly defined. Bird flocking is assumed to optimize a certain objective function. Each agent knows its best value so far (*pbest*) and its current position. This information is an analogy of personal experience of an agent. Moreover, each agent knows the best value so far in the group (*gbest*) among *pbests* of all agents. This information is an analogy of an agent knowing how other agents around it have performed. Each agent tries to modify its position using the concept of velocity. The velocity of each agent can be updated by the following equation:

$$v_i^{k+1} = wv_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k) \quad (5)$$

where v_i^k is velocity of agent i at iteration k, w is weighting function, c_1 and c_2 are weighting factors, $rand_1$ and $rand_2$ are random numbers between 0 and 1, s_i^k is current position of agent i at iteration k, $pbest_i$ is the *pbest* of agent i, and *gbest* is the *gbest* of the group. The following weighting function is usually utilized in (5):

$$w = w_{\max} - ((w_{\max} - w_{\min}) / (iter_{\max})) \times iter \quad (6)$$

where $w_{\max}$ is the initial weight, $w_{\min}$ is the final weight, $iter_{\max}$ is the maximum iteration number, and $iter$ is the current iteration number. Using the above equations, a certain velocity, which gradually bring the agents close to *pbest* and *gbest* can be calculated. The current position (search point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (7)$$

The model using (5) is called *Gbest model*. The model using (6) in (5) is called *Inertia Weights Approach* (IWA). Fig. 4 shows the concept of modification of a search point by the PSO.

3.2 Multiobjective Optimization in FFPU

The multiobjective optimization problem of the FFPU is to find an optimal solution in the solution space that minimizes the load tracking error, fuel usage, and throttling losses in the main steam and feedwater control valves (Garduno-Ramirez and Lee 2000). Therefore, the following objective functions can be described for minimization:

$$J_1(u) = |E_{uld} - E_{ss}| \quad (8,a)$$

$$J_2(u) = u_1 \quad (8,b)$$

$$J_3(u) = -u_2 \quad (8,c)$$

$$J_4(u) = -u_3 \quad (8,d)$$

where E_{uld} is the unit load demand (MW), and E_{ss} is the corresponding generation (MW) as provided by the steady-state equation:

$$E_{ss} = ((0.73u_2 - 0.16)/0.0018u_2)(0.9u_1 - 0.15u_3) \quad (9)$$

The objective functions are described as following: $J_1(u)$ accounts for the power generation error, $J_2(u)$ accounts for fuel consumption through the fuel valve position, and $J_3(u)$ accounts for energy loss due to pressure drop across the steam valve. Since the pressure drop increases as the valve closes, it is desired to keep it open as wide as possible, thus it is desired to maximize u_2 , or equivalently minimize $-u_2$. Similarly, $J_4(u)$ accounts for energy loss due to the pressure drop in the feedwater control valve. Thus, the multiobjective optimization is to be performed to minimize all objective functions defined above under a given set of preference.

3.3 PSO for Multiobjective Optimization in the FFPU

Initialization; The first step of the PSO for the FFPU is random generation of the agents in the solution space. The agents represent the search points in the solution space, which are expressed by controls u_1, u_2, and u_3. Moreover, the initial velocities are also generated randomly within the same space. Whenever the unit load demand is changed, the initial agents and velocities are created in the solution space corresponding to the given unit load demand. In order to speed up the search for an optimal solution, c_1 and c_2 are set to 2, $w_{max} = 0.8$, and $w_{min} = 0.3$ in the PSO. These values are obtained from experimental results by testing the convergence rate. The number of agents is 40 and the iteration is 130. The initial *pbests* are equal to the current search points and *gbest* is found by comparing the *pbests* among the agents.

Evaluation; The evaluation of search point for each agent is performed by using the deviation of each objective function from its possible minimum value, which then is weighted with a preference value. In the multiobjective optimization, the objective functions are often in conflict to each other when performing the optimization. Thus, it is proposed to minimize the maximum deviation of the objective functions instead of directly minimizing the multiobjective functions (Garduno-Ramirez and Lee 2001). The maximum deviation of the multiobjective functions is defined as following:

$$\delta_m = \max_{i=1,\dots,k} \delta_{pi} \qquad \delta_{pi} \geq 0 \qquad (10.a)$$

$$\delta_{pi} = \beta_i \left| J_i(u) - J_i(u)^* \right|, \; i = 1,2,\dots,k, \; u \in \Omega \qquad (10.b)$$

$$J_i^* = \min\{J_i(u); u \in \Omega\}, \; i = 1,2,\dots,k \qquad (10.c)$$

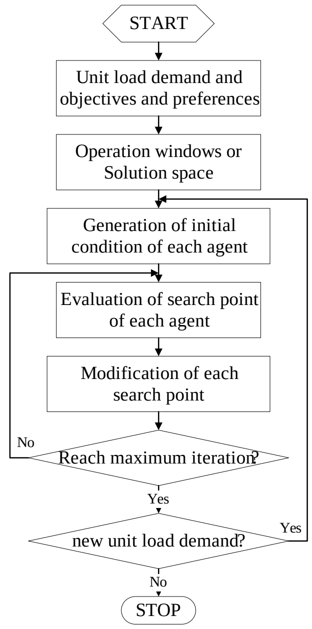

Fig. 5. Total flow chart of the PSO in the FFPU.

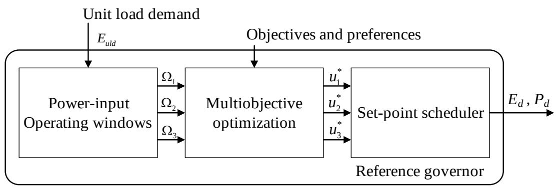

Fig. 6. Configuration of reference governor for FFPU.

where δ_m is the maximum deviation of the multiobjective functions, δ_{pi} is weighed deviation, β_i is the preference value, J_i^* is the minimum possible value of the objective function J_i , and Ω is the solution space. After evaluation, if the new value is better than the current *pbest* of agent, the *pbest* value is replaced by the new value. If the best new value among all *pbests* is better than the current *gbest*, *gbest* is replaced by the best new *pbest* value and the agent number with the best value is stored.

Modification; The modification of current search point is performed by (5), (6), and (7) in every iteration. The first term in the right-hand side of (5) is for diversification in the search procedure, which tries to explore new areas. The second and third terms are for intensification in the search procedure. It helps to converge to their *pbests* and/or *gbest*. The method has a well-balanced mechanism to utilize diversification and intensification efficiently in the search procedure (Shi and Eberhart, 1998). Fig. 5 shows the total flow chart of the PSO in the FFPU.

3.4 Set-point Scheduler

The result of PSO procedure gives a set of optimal solution (u_1^*, u_2^*, and u_3^*) from the solution space (Ω_1, Ω_2 and Ω_3) for the given unit load demand (E_{uld}) as shown in Fig. 6, which is the configuration of reference governor in FFPU. For the controllers, the scheduler maps the optimal solutions into set-points, demand power E_d and pressure P_d, by direct steady-state equations:

$$E_d = ((0.73u_2^* - 0.16)/(0.0018u_2^*))(0.9u_1^* - 0.15u_3^*) \quad (11.a)$$

$$P_d = 141u_3^*/(1.1u_2^* - 0.19) \quad (11.b)$$

4. SIMULATION RESULTS

In the following simulations, the results by the basic PSO technique will be shown. Simulations deal with three different cases:

Case 1; minimize only $J_1(u)$

Case 2; minimize $J_1(u)$ and $J_2(u)$

Case 3; minimize $J_1(u)$, $J_2(u)$, $J_3(u)$ and $J_4(u)$

The objective functions are equation (8) and a vector of preference values is given as β = [1, 0.5, 1, 0]. This means that $\beta_1 = 1$ is for $J_1(u)$, $\beta_2 = 0.5$ for $J_2(u)$, $\beta_3 = 1$ for $J_3(u)$, and $\beta_4 = 0$ for $J_4(u)$. These values imply the priorities of each objective function in the multiobjective optimization problem, where 1 is the highest and 0 is the lowest priorities in the optimization.

4.1 Solution Space

Fig. 7 shows a unit load demand that resembles a typical load cycle. It has different rising and falling slopes and different constant powers. With the given unit load demand and the plant model, the solution space is obtained from the power-input operating windows (Fig. 4). Fig. 8 shows the solution space (Ω_1, Ω_2 and Ω_3) for the given unit load demand. The gaps between upper and lower limits are the solution space for the optimization process.

4.2 Optimal Solution Trajectories

Next step is to perform the PSO for the multiobjective optimization with predefined objective functions and preference values. Figs. 9-11 show the optimal input trajectories that are optimal solutions in the solution space. In Fig. 9, fuel consumption, u_1 is reduced as the number of objectives is increased, which is desirable. Fig. 10 shows that the valve opening u_2 is increased as the number of objectives is increased, which is also desirable. On the other hand, Fig. 11 shows that the results are almost same for all cases. This is because the solution space for the feedwater valve is very small as shown in Fig. 8. All simulation results are improved as the number of objectives is increased. These optimal solutions are the values of *gbest* that are found through the PSO technique.

4.3 Set-Point Trajectories

Finally, the power and pressure set-point trajectories are obtained by set-point scheduler as shown in Fig. 12 and Fig. 13, respectively. The demand power (E_d) is the same as the unit load demand as shown in Fig. 12. The demand pressure set-points, (P_d) mapped for different number of objective functions are shown in Fig. 13. This is because the power-pressure operating window is quite large and the same amount of power can be produced on a wide range of pressure as shown in Fig. 2. As additional objective functions are added in the optimization, the plant is operating more conservatively in lower pressure. Thus, all simulation results show that the PSO technique can be accommodated well in the multiobjective optimization problem and also in the on-line implementation since the pressure set-point need to be updated only when the unit load demand is changed during the load cycle. Moreover, the computing time is reasonably short for on-line implementation in the FFPU.

5. CONCULSION

The Particle Swarm Optimization (PSO) technique is presented as an alternative optimization technique for solving a multiobjective optimization problem. The feasibility of the using the PSO is demonstrated in designing optimal set-points for the multiobjective optimal power plant operation. The optimal mapping between unit load demand and pressure set-point is realized and the mapping can also be realized for time-varying load demand. Therefore, the Particle Swarm Optimization (PSO) is shown to solve the multiobjective optimization effectively.

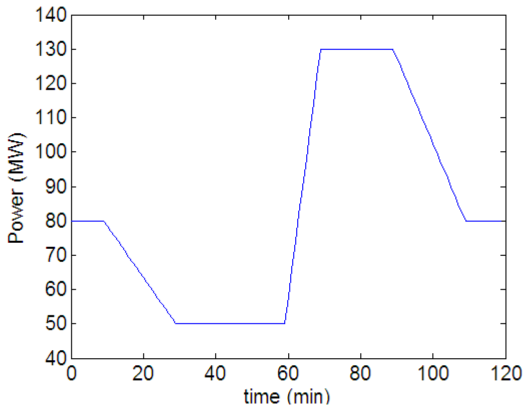

Fig. 7. Unit load demand.

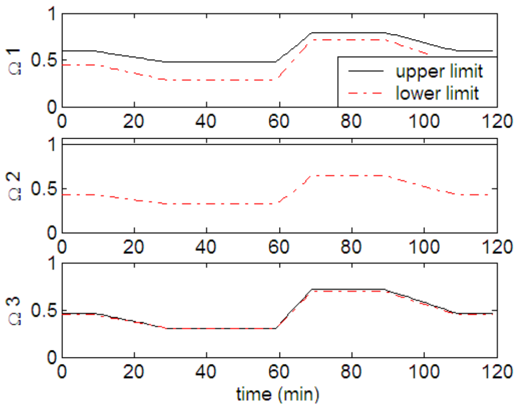

Fig. 8. Solution space by the given unit load demand.

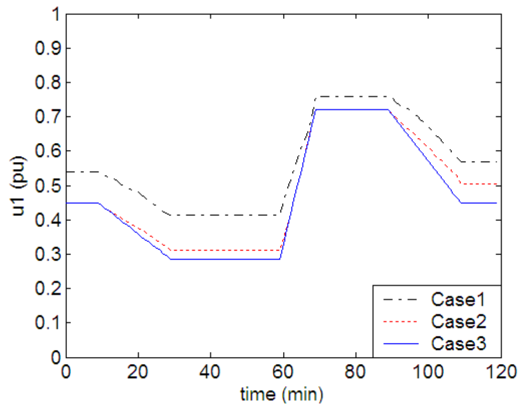

Fig. 9. Optimal input u_1 trajectories.

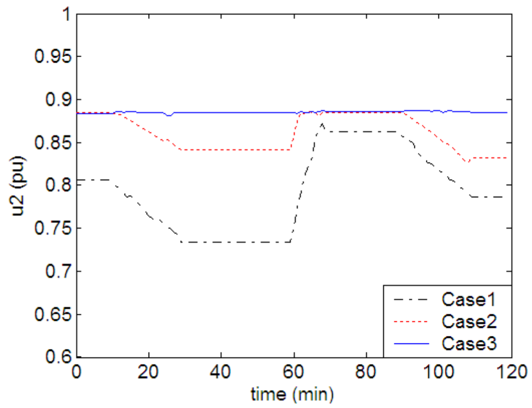

Fig. 10. Optimal input u_2 trajectories.

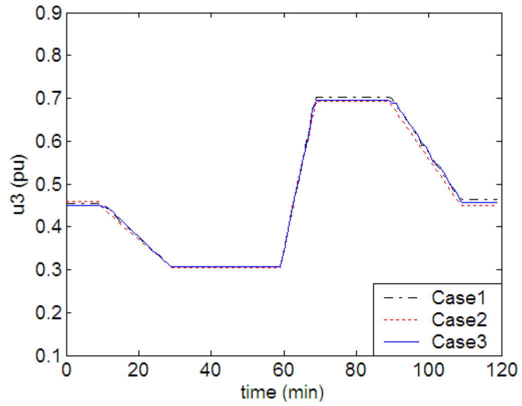

Fig. 11. Optimal input u_3 trajectories.

REFERENCES

Bell, R. D. and Åström, K. J. (1987). *Dynamic models for boiler-turbine-alternator units: Data logs and parameter estimation for 160 MW unit*, Lund Institute of Technology, Sweden TFRT-3192.

Eberhart, R. and Shi, Y. (2000). Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, *Proc. of the Congress on Evolutionary Computation(CEC2000)*, pp.84-88.

Elgerd, O. I. (1971). *Electric Energy Systems Theory: An Introduction*: McGraw–Hill: New York.

Gery, H. C. (1988). The evolution of coordinated control, *Instrument Society of America*, Paper No. 88-0417, pp. 109-112.

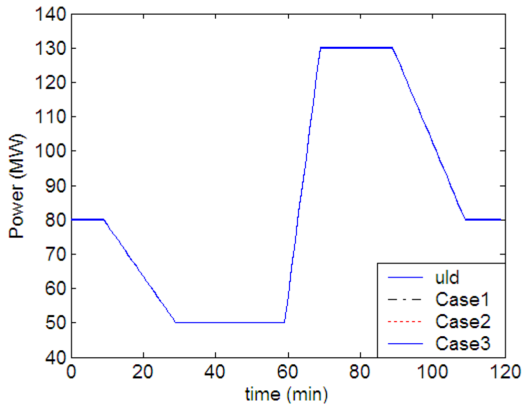

Fig. 12. Demand power set-point trajectories.

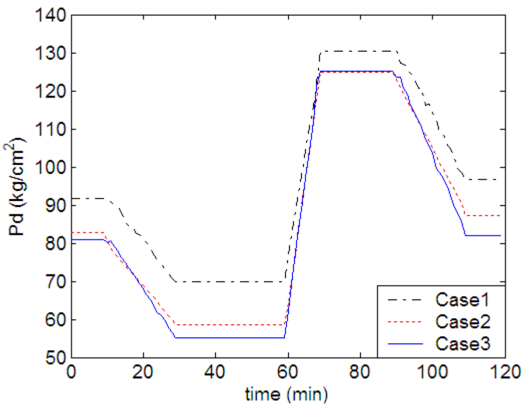

Fig. 13. Demand pressure set-point trajectories.

Garduno-Ramirez, R. and Lee, K. Y. (2000). Wide-range operation of a power unit via feedforward fuzzy control, *IEEE Trans. on Energy Conversion*, 15 (4), pp. 421-426.

Garduno-Ramirez, R. and Lee, K. Y. (2001). Multiobjective optimal power plant operation through coordinate control with pressure set point scheduling, *IEEE Trans. on Energy Conversion*, 16 (2), pp. 115-122.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, *Proc. of IEEE international Conference on Neural Networks (ICNN'95*), Vol. IV, pp. 1942-1948, Perth, Australia.

Lee, K. Y. and El-Sharkawi, M. A. (Editors). (2002). *Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems*, IEEE Power Engineering Society, IEEE Catalog Number 02TP160, Piscataway, NJ.

Park, J.-B., Lee, K.-S., Shin, J.-R. and Lee, K. Y. (2003). Economic Load Dispatch Based on a Hybrid Particle Swarm Optimization, *Proc. International Conference on Intelligent System Application to Power Systems*, CD ISAP03-070.pdf, Lemnos, Greece.

Reynolds, C. (1987). Flocks, herds, and schools: A distributed behavioral model, *Computer Graphics*, 21 (4), pp. 25-34.

Shi, Y. and Eberhart, R. (1998). A Modified particle swarm optimizer, *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'98*), pp. 60-73, Anchorage.