Multiobjective Control of Power Plants Using Particle Swarm Optimization Techniques

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

***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 for 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. This approach provides the means to specify optimal set points for controllers under a diversity of operating scenarios. Variations of the PSO technique, hybrid PSO, evolutionary PSO, and constriction factor approach are applied to the FFPU, and the comparison is made among the PSO techniques and genetic algorithm.**

***Index Terms*—Genetic algorithm (GA), multiobjective optimization, particle swarm optimization (PSO), power plant control, pressure set point scheduling.**

I. Introduction

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

First, the fossil fuel power unit (FFPU) must meet the load demand of electric power at all times, at constant voltage, and at constant frequency [1]. 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. 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 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. PSO has been developed for the nonlinear continuous optimization problem, based on the experience gained from the study of artificial life and psychological research. Eberhart and Kennedy developed PSO, based on the analogy of the swarm of birds and the school of fish [2], [3]. One of the main goals is to examine how natural creatures behave as a swarm and to reconfigure the swarm model computationally. It is well known that the PSO techniques can provide a high-quality solution with simple implementation and fast convergence [2]–[13].

When the unit load demand is received from the control center, the balance of the plant should be maintained by controlling the boiler and the turbine so the generator can generate the power. In generating the power required by the unit load demand, the steam pressure needs to be set to an optimal value, which depends on various operation requirements, such as load following, fuel conservation, life extension of equipment, reducing pollution, and so on. These requirements are often conflicting, and in this paper, the optimal set points are determined by solving the multiobjective optimization problem of these conflicting requirements. The multiobjective optimization problem was tackled earlier with an analytic mathematical programming [17] and genetic algorithm (GA) [20]. However, to implement the optimization in real time, it is necessary to develop a technique that requires less computation time and is easier to implement.

Therefore, this paper tests the performance of the PSO techniques in the multiobjective optimization problems in the dynamic environment of the FFPU. There are several variations of the basic PSO technique that are known to perform better than the basic PSO, but their performance is problem dependent. Therefore, by comparing these techniques, the paper investigates which technique is most appropriate for the multiobjective optimization of the FFPU.

In FFPU, the swarm consists of agents that 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. In this paper, three variations of the PSO technique i.e., the hybrid PSO (HPSO), evolutionary PSO (EPSO), and constriction factor approach (CFA)—are applied to the FFPU, and the comparison is made among the PSO techniques and GA. Thus, the PSO technique that can be successfully applied to the FFPU and that

Manuscript received August 13, 2004; revised February 20, 2005. Paper no. TEC-00242-2004.

J. S. Heo and K. Y. Lee are with the Department of Electrical Engineering, the Pennsylvania State University, University Park, PA 16802 USA (e-mail: juh138@psu.edu; kwanglee@psu.edu).

R. Garduno-Ramirez is with Unit of Control and Instrumentation, Electric Research Institute, Cuernvaca, Morelos 62490, Mexico (e-mail: rgarduno@iie.org.mx).

Digital Object Identifier 10.1109/TEC.2005.858078

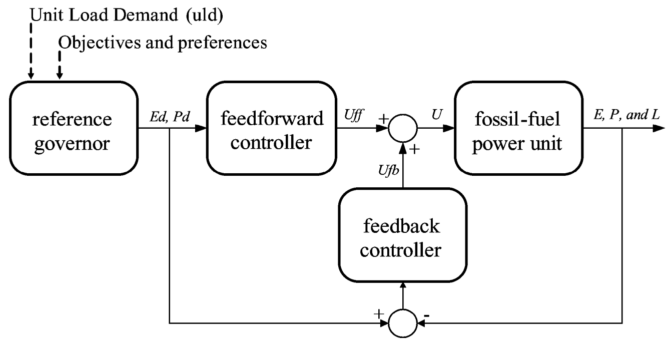

Fig. 1. Coordinated control structure for FFPU.

is most appropriate for the multiobjective optimization problem in the dynamic environment is shown.

Following this section, the power plant control system is described in Section II. Section III describes the PSO techniques (HPSO, EPSO, and CFA). Section IV shows the implementation of the PSO techniques in the FFPU to demonstrate the feasibility of the proposed approach. The final section draws some conclusions.

II. Power Plant Control System

A. Control Structure

There have been several control strategies for the FFPU: boiler following control, turbine following control, and coordinated boiler-turbine control strategies. The boiler following approach has faster but less stable response to load changes. The turbine following approach has more stable but slower response to load changes. The coordinated control strategy is meant to synthesize the advantages of the two approaches while minimizing their disadvantages [14]. For the control system to have more stable and faster response to load changes, this paper uses the coordinated control scheme, which requires references (or set points) for both power demand (E_d) and pressure demand (P_d). The control structure of the coordinated control 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 the load demand.

B. Power Unit Model

The FFPU under study is a 160-MW 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 [15]. 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/cm^2), 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 following equations are a summary of the third-order model by Bell and Åström [15]. The state equations are as follows:

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

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

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

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

$$q_e = (0.85u_2 - 0.14)P + 45.59u_1 - 2.51u_3 - 2.09 \tag{2a}$$

$$\alpha_s = (1/\rho_f - 0.0015)/(1/(0.8P - 25.6) - 0.0015) \tag{2b}$$

$$L = 50(0.13\rho_f + 60\alpha_s + 0.11q_e - 65.5) \tag{2c}$$

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

$$-0.007 \leq du_1/dt \leq 0.007 \tag{3a}$$

$$-2.0 \leq du_2/dt \leq 0.02 \tag{3b}$$

$$-0.05 \leq du_3/dt \leq 0.05. \tag{3c}$$

C. Reference Governor

The multiobjective optimization is performed at the reference governor. The essence of the reference governor in the coordinated control is that of designing the optimal mappings from the unit load demand E_{uld} to the set points E_d and P_d

$$\mathrm{SP}_E: \quad E_{\mathrm{uld}} \rightarrow E_d$$

$$\mathrm{SP}_P: \quad E_{\mathrm{uld}} \rightarrow P_d$$

which will be used to transform any unit load demand pattern (E_{uld}, t) into optimal set point trajectories for the power (E_d, t) and pressure (P_d, t) control loops.

The set point mappings *SP* are basically designed by solving a multiobjective optimization problem that takes into account the specified operation objectives and the steady-state model of the plant. Then, the reference governor performs the design process in three steps (Fig. 2):

- Determination of the feasibility regions for the decision variables;
- Solution of the multiobjective optimization problem to find optimal steady-state control signals; and
- Calculation of the set points through direct evaluation of the steady-state model of the unit.

1) Feasibility Regions of Control Inputs: The feasibility regions $\Omega_i, i = 1, 2, 3$ for the decision variables u_1, u_2 and u_3 are determined using power-input operating windows. In this paper, the nonlinear mathematical model of the FFPU is used to obtain the operating windows of the control valve demands. The operating windows, or operating regions, are the sets of all feasible

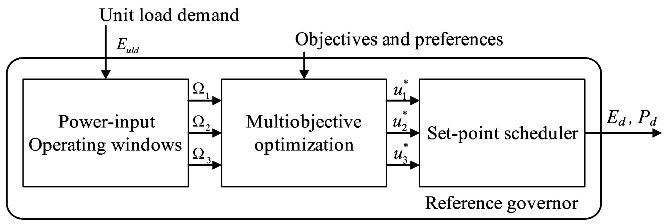

Fig. 2. Configuration of reference governor for the FFPU.

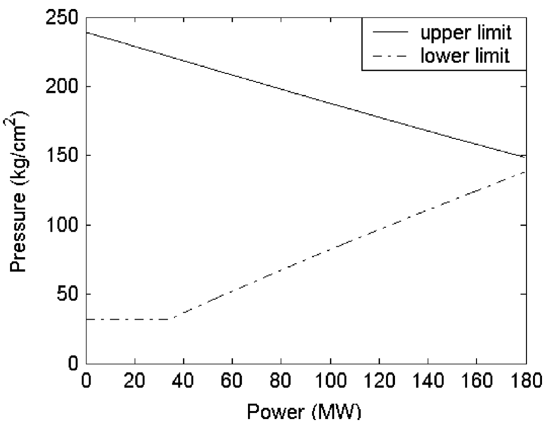

Fig. 3. Power-pressure operating window.

operating points for the FFPU. An operating point is declared feasible when stable steady solution is achievable at that point, whereas all imposed constraints are satisfied. Fig. 3 shows the power-pressure operating window, which is determined through simulations using the FFPU model in steady state, taking into account equipment physical limitations (hard constraints) and operational limitations (soft constraints). From the process optimization perspective, the power-pressure operating window is the most important one. As shown in Fig. 3, it clearly indicates that any required power can be generated at any pressure value between the depicted upper and lower pressure limits. The power-input operating windows can be determined using the inverse static model of the plant

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

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

$$u_3 = ((1.1u_2 - 0.19)P)/141 \tag{4c}$$

which is obtained by solving the FFPU plant model (1) in steady state for the control inputs. The operating windows for the inputs u_1, u_2 and u_3 can be generated by (4) using all possible power E and pressure P in the power-pressure operating window.

The power-input operating windows for the control signals u_1, u_2, and u_3 are shown in Fig. 4. When a unit load demand $E_{\rm uld}$ is given, the operating windows block will generate the feasible input regions Ω_1, Ω_2, and Ω_3 from the power-input operating windows (Fig. 4). Note that at any given power operating point the fuel and steam valve demands may vary substantially, whereas the feed water valve demand shows a relatively small

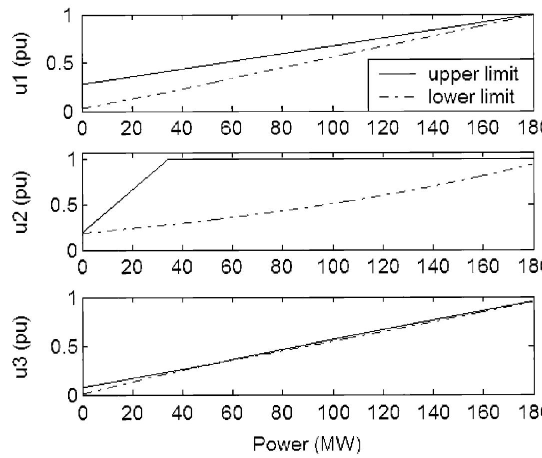

Fig. 4. Power-input operating windows.

variation. Later, these facts will be of great relevance for optimization.

2) Optimal Steady-State Control Inputs: In the second stage, a multiobjective optimization problem is solved for a prescribed value of the unit load demand $E_{\rm uld}$. The purpose is to find, in a multiobjective sense, an optimal vector of units $u = [u_1\ u_2\ u_3]^T$ in the feasible regions $\Omega_i, i = 1, 2, 3$ previously determined, that minimizes the desired objective functions. As is shown later in this paper, the objective functions may account for load tracking error, thermal stress, heat loss rate, pollution, or any other operating objective of interest to be optimized.

At the end of the second stage, the calculated vector of optimal inputs u^* and the corresponding unit load demand value being considered, determine an optimal operating point, in the specified multiobjective sense, for the plant.

3) Calculation of Set Points: In the third stage, the vector of optimal control signals u^* is used to generate a vector of optimal set points through the steady-state model of the power unit

$$[E_d P_d L_d]^T = M_{\rm SS}([u_1^*\ u_2^*\ u_3^*]^T)$$

where $M_{\rm SS}$ is the power unit steady-state model solved with u as input and the controlled variables as outputs. The steady-state model is obtained by setting the dynamic equation (1) to zero

$$E_d = ((0.73u_2^* - 0.16)/(0.0018u_2^*))(0.9u_1^* - 0.15u_3^*) \tag{5a}$$

$$P_d = 141u_3^*/(1.1u_2^* - 0.19) \tag{5b}$$

where the demand on water-level deviation L_d is simply set to zero always.

D. Multiobjective Optimization

The core of the reference governor is the multiobjective optimization in the second stage. 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 [16]. Therefore, the following objective functions can be described for minimization:

$$J_1(u) = |E_{\text{uld}} - E| \tag{6a}$$

$$J_2(u) = u_1 \tag{6b}$$

$$J_3(u) = -u_2 \tag{6c}$$

$$J_4(u) = -u_3 \tag{6d}$$

where E_{uld} is the unit load demand (MW) and E is the corresponding generation (MW), as provided by the steady-state (5a). The objective function $J_1(u)$ accounts for the power generation error, $J_2(u)$ accounts for fuel consumption through the fuel valve position u_1. $J_3(u)$ accounts for energy loss due to the pressure drop across the steam valve. Because the pressure drop, and consequently the energy loss, increases as the valve closes, it is desired to keep it as widely open as possible; thus, maximizing u_2, or equivalently, minimizing $-u_2$ will reduce the loss in the steam valve. Similarly, $J_4(u)$ accounts for energy loss due to the pressure drop in the feedwater valve. Thus, the multiobjective optimization is to minimize all objective functions defined previously under a given set of priorities or preferences β defined as follows.

In the multiobjective optimization, the objective functions are often in conflict with 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 [17]. The maximum deviation of the multiobjective functions is defined as follows:

$$\delta_m = \max_{i=1,\ldots,k} \delta_i, \qquad \delta_i \geq 0 \tag{7a}$$

$$\delta_i = \beta_i |J_i(u) - J_i(u)^*|, \quad i = 1, 2, \ldots, k, \quad u \in \Omega \tag{7b}$$

$$J_i^* = \min\{J_i(u); u \in \Omega\}, \quad i = 1, 2, \ldots, k \tag{7c}$$

where δ_m is the maximum deviation of the multiobjective functions, δ_i is weighed deviation, β_i is the preference value, J_i^* is the minimum possible value of the single objective function J_i, and Ω is the solution space. The preference values give the relative priorities of the objectives in searching for the optimal solution. In this paper, the multiobjective optimization using the PSO techniques is the main issue. With the objective functions (6) and maximum deviation function (7), the PSO techniques are used to find the optimal input u^*.

III. Particle Swarm Optimization

A. Overview of the Basic PSO

Basically, the PSO was developed through simulation of birds flocking in two-dimensional space [4]. The position of each bird (called agent) is represented by a point in the X–Y coordinates, and 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:

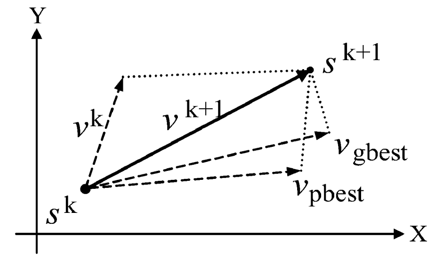

Fig. 5. Concept of modification of a searching point by PSO.

$$v_i^{k+1} = wv_i^k + c_1\,\text{rand}_1 \times \left(\text{pbest}_i - s_i^k\right) + c_2\,\text{rand}_2 \times \left(\text{gbest} - s_i^k\right) \tag{8}$$

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 best value so far in the group among the *pbests* of all agents. The following weighting function is usually used in (8):

$$w = w_{\max} - ((w_{\max} - w_{\min})/(\text{iter}_{\max})) \times \text{iter} \tag{9}$$

where $w_{\max}$ is the initial weight, $w_{\min}$ is the final weight, $\text{iter}_{\max}$ is the maximum iteration number, and $iter$ is the current iteration number. Using the previous equations, a certain velocity, which gradually brings 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:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{10}$$

The model using (8) is called *Gbest model*. The model using (9) in (8) is called inertia weights approach (IWA). Fig. 5 shows the concept of modification of a search point by the PSO.

B. Variations of the Basic PSO

1) Hybrid PSO [5], [6]: The HPSO uses the basic mechanism of the PSO and the natural selection mechanism, which is usually performed by evolutionary computation (EC) methods such as genetic algorithm (GA). Because search procedure by the PSO depends strongly on *pbest* and *gbest*, the search area can be limited by them. However, by introducing a natural selection mechanism, the effect of *pbest* and *gbest* is gradually eliminated and a broader search area can be realized. Agent positions with better performance replace those with poor performance. However, the *pbest* information of each agent is maintained. Therefore, both intensive search in a currently effective area and dependence on the past high-performance position are used at the same time. Fig. 6 shows the concept of search process by the HPSO.

2) Evolutionary Self-Adapting PSO [7], [8]: The main differences between EPSO and the basic PSO are an explicit selection procedure and self-adapting properties for its parameters. Instead of moving the agents to find an optimal solution in solution space, EPSO reproduces the agents with the movement rule of PSO and the mutation rule of evolutionary strategy

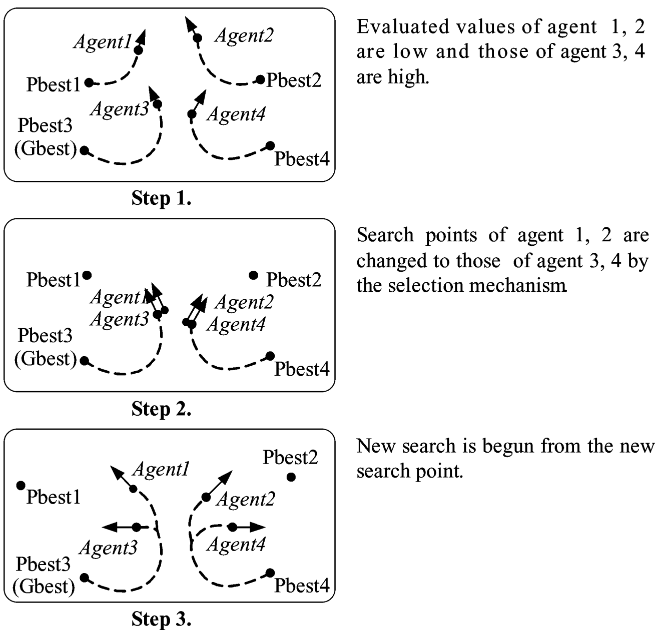

Fig. 6. Concept of search process by HPSO.

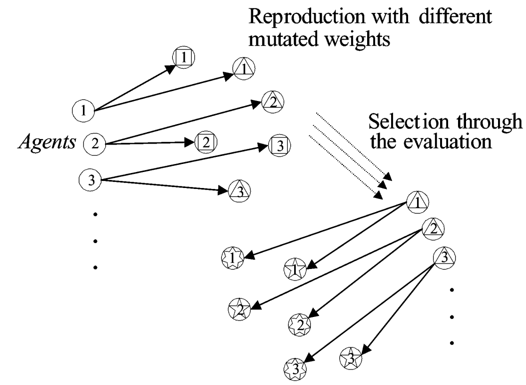

Fig. 7. Concept of reproduction and selection in EPSO.

(ES). Then, the best agents are selected by stochastic tournament through the evaluation. The general scheme of EPSO is as follows.

REPLICATION: Each agent is replicated r times.

MUTATION: Each agent has its weights mutated.

REPRODUCTION: Each mutated agent generates an offspring according to the agent movement rule.

EVALUATION: Each offspring has its fitness evaluated.

SELECTION: By stochastic tournament, the best agents survive to form a new generation.

The movement rule for EPSO is the following:

$$v_i^{\text{new}} = w_{i0}^* v_i + w_{i1}^* (\text{pbest}_i - s_i) + w_{i2}^* (\text{gbest}^* - s_i) \tag{11a}$$

$$w_{ik}^* = w_{ik} + \tau \cdot N(0,1), \text{gbest}^* = \text{gbest} + \tau' \cdot N(0,1) \tag{11b}$$

$$s_i^{\text{new}} = s_i + v_i^{\text{new}} \tag{11c}$$

where w_{ik}^* are the weights that undergo mutation, gbest* is the *gbest* distributed randomly, τ, τ' are learning parameters (either fixed or treated as strategic parameters and therefore subject to mutation), and $N(0,1)$ is a Gaussian random variable with 0 mean and variance 1. Fig. 7 shows the concept of reproduction and selection in EPSO.

This scheme benefits from two "pushes" in the right direction—the selection process of ES and the agent movement rule of PSO; therefore, it is natural to expect that it may display advantageous convergence properties when compared with ES or PSO alone. Furthermore, EPSO can also be classified as a self-adaptive algorithm because it relies on the mutation and selection of strategic parameters [7], [8].

3) Constriction Factor Approach [9], [10]: The basic system equations of the PSO [(8)–(10)] can be considered as dynamic difference equations. Therefore, the system dynamics, namely the search procedure, can be analyzed by the eigenvalue analysis and controlled so the system converges and can search different regions efficiently.

The velocity of the constriction factor approach (simplest constriction) can be expressed as follows, instead of (8) and (9):

$$v_i^{k+1} = K \left[v_i^k + c_1 \times \text{rand}_1 \times \left(\text{pbest}_i - s_i^k\right) + c_2 \times \text{rand}_2 \times \left(\text{gbest} - s_i^k\right)\right] \tag{12a}$$

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad \text{where } \varphi = c_1 + c_2, \varphi > 4. \tag{12b}$$

In the CFA, the φ must be greater than 4 to guarantee stability. However, as φ increases the weight K decreases and diversification is reduced, yielding slower response. Therefore, we choose 4.1 as the smallest φ that guarantees stability but yields the fastest response. It has been observed here, as well as in other paper [6], that $\varphi = 4.1$ leads to good solutions. The CFA results in convergence of the agents over time. Unlike other EC methods, the CFA ensures the convergence of the search procedure based on mathematical theory. Therefore, the CFA can generate higher quality solutions than the basic PSO approach.

However, the constriction factor only considers dynamic behavior of one agent and the effect of the interaction among agents. Namely, the equations were developed with a fixed set of best positions (*pbest* and *gbest*), although *pbest* and *gbest* change during search procedure in the basic PSO equation.

IV. Numerical Simulation

A. Implementation of PSO Techniques in FFPU

The PSO techniques are applied to solve the multiobjective optimization problem in the reference governor. Fig. 8 shows the flowchart of the PSO in designing the reference governor.

1) Initialization: The first step of the PSO for the FFPU is random generation of the agents in the solution space, which is the feasible input regions Ω_1, Ω_2, and Ω_3. 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. To expedite the search for an optimal solution, c_1

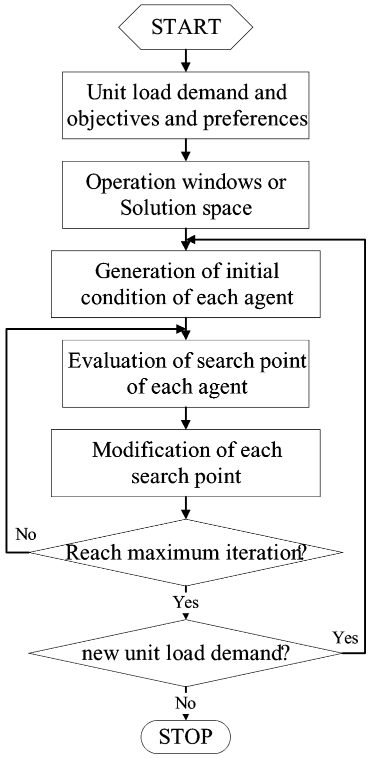

Fig. 8. Total flowchart of PSO in the FFPU.

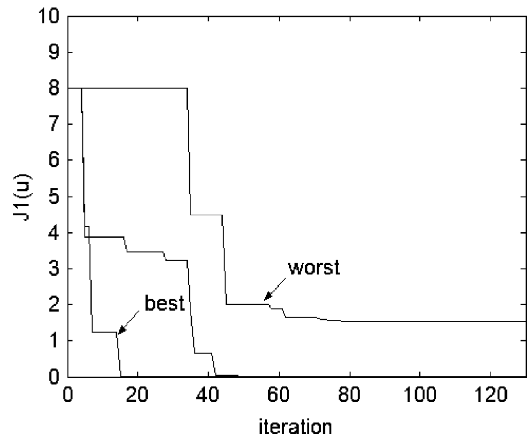

Fig. 9. Evaluation of convergence rate with different values of parameters.

and c_2 are set to 2, $w_{\max} = 0.8$, and $w_{\min} = 0.3$ in the basic PSO and HPSO. However, CFA uses only two parameters, c_1 and c_2. For the best performance, the values of c_1 and c_2 are set to 2.1 and 2.0, respectively. These values are obtained from experimental results (Fig. 9). The number of agents is 40, and the iteration is 130 for the basic PSO and its variations. The initial *pbests* are equal to the current search points, and *gbest* is found by comparing the *pbests* among the agents.

2) Evaluation: The evaluation of search point for each agent is performed using maximum deviation function (7) in all PSO techniques. However, there are some differences between the PSO techniques in the procedures. In the basic PSO and CFA, if the new position of the agent has better performance than the current *pbest*, the *pbest* is replaced by the new position. If the best new position among all *pbests* is better than the current *gbest*, the *gbest* is replaced by the best new *pbest*, and the agent number with the best *pbest* is stored [11]. In the HPSO, the natural selection mechanism is performed during the step "Evaluation of search point of each agent" in Fig. 8. In finding the best solution, the selection is achieved by forming subgroups from the entire group in the solution space. Then, each agent is evaluated, and the best agent is found in each subgroup. Finally, the agents with low performance are replaced by the agent with the best performance in the subgroup. Thus, the new search with the selection mechanism would provide more chance to find an optimal solution. The current points, however, are evaluated with the original *pbests* and *gbest*. Therefore, the HPSO method realizes more intensive search nearby the best agents [12]. In the EPSO, because EPSO applies the movement rule (11), each agent generates an offspring. During the evaluation, the better agent is selected between the two descendants, which have different w and $gbest$. Thus, EPSO adapts to have better weights while searching for an optimal solution. However, it has a drawback of requiring two evaluations per agent per iteration.

3) Modification: In the basic PSO and HPSO, the modification of current search point is performed by (8)–(10) in each iteration. The first term in the right-hand side of (8) is for diversification in the search procedure, which keeps trying to explore new areas. The second and third terms are for intensification in the search procedure. They help in moving toward the *pbests* and/or *gbest* [13]. The method has a well-balanced mechanism to use diversification and intensification efficiently in the search procedure. However, the EPSO and CFA have their own modification rule. The EPSO is performed with (11). First, each agent is replicated twice. Then, w and *gbest* are mutated with fixed learning parameters τ and τ', respectively, as shown in (11), which is called the "movement rule" in the EPSO. Finally, each agent generates an offspring for the modification. The CFA performs with (12) in this step. When the search algorithm looks for an optimal solution in a solution space, it has a velocity multiplied with the factor K of (12b) instead of w in the IWA (9).

B. Simulation Results

In the following simulations, only the results by the basic PSO technique are shown due to space limitations. However, the comparison between the basic PSO and its three variations is shown at the end of the simulation. 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 given in (6), and a vector of preference values is given as $\beta = [1, 0.5, 1, 0]$. This means that $\beta_1 = 1$ is for $J_1(u), \beta_2 = 0.5$ is for $J_2(u), \beta_3 = 1$ is for $J_3(u)$, and $\beta_4 = 0$ is for $J_4(u)$. The 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

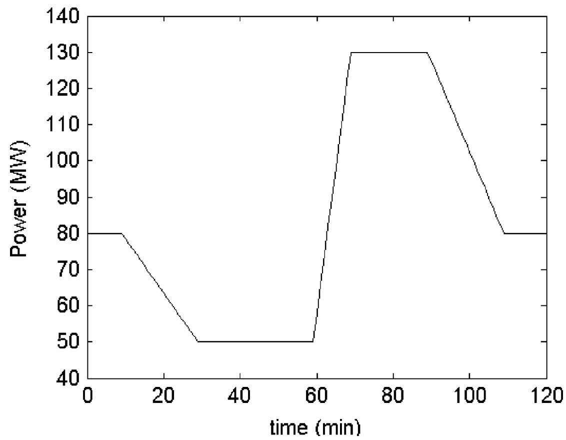

Fig. 10. Unit load demand.

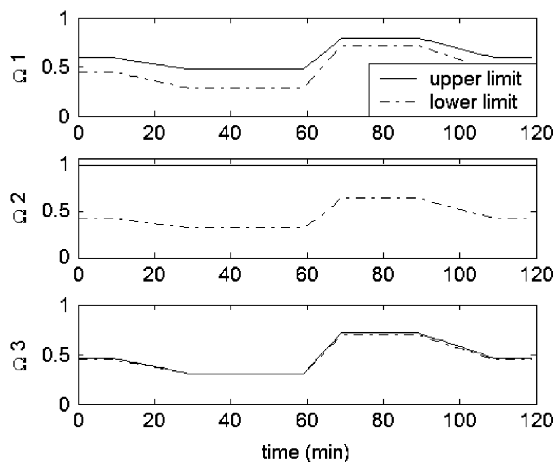

Fig. 11. Solution space by the given unit load demand.

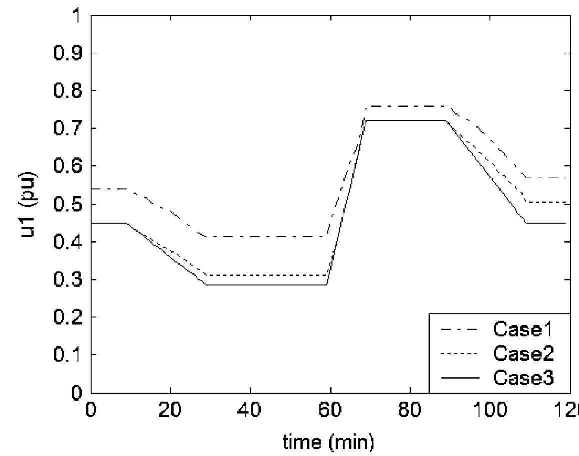

Fig. 12. Optimal input μ_1 trajectories.

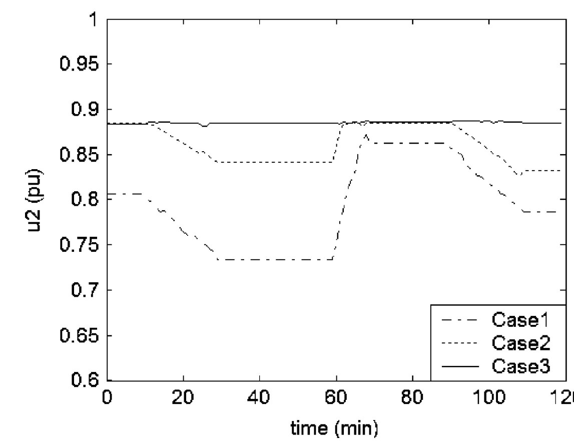

Fig. 13. Optimal input μ_2 trajectories.

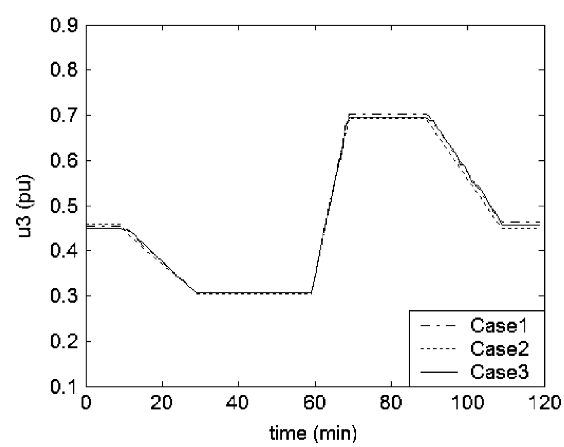

Fig. 14. Optimal input μ_3 trajectories.

the optimization. To select the appropriate values of parameters $c_1, c_2, w_{\max}$, and $w_{\min}$, experiments are performed in trial and error, by testing the convergence rate with many different parameter values. Fig. 9 shows the evaluation of convergence rate with different values of parameters.

1) Solution Space: Fig. 10 shows a unit load demand that resembles a typical load cycle. It has different rising and falling slopes and different level of constant powers. With the given unit load demand and the plant model, the solution space is obtained using the power-input operating windows (Fig. 4). Fig. 11 shows the solution space for the given unit load demand. The gaps between upper and lower limits are the solution space for the optimization process.

2) Optimal Solution Trajectories: The next step is to perform the PSO for the multiobjective optimization with predefined objective functions and preference values. Figs. 12–14 show the optimal input valve trajectories that are optimal solutions found by the PSO in the solution space. The result agrees with the practice that for a long time, electric utilities have used the heat rate method [18] to evaluate the actual performance of power units, and the heat rate can be improved while fuel usage and throttling losses in the main steam and feedwater valves are taken into account. Moreover, Buchwald [19] showed that lowering the pressure drop across the control valves can reduce the energy dissipated in a process, so that the energy required to operate many processes can be significantly reduced.

In Fig. 12, as the number of objectives is increased, fuel consumption is reduced for cases 2 and 3. Fig. 13 shows that

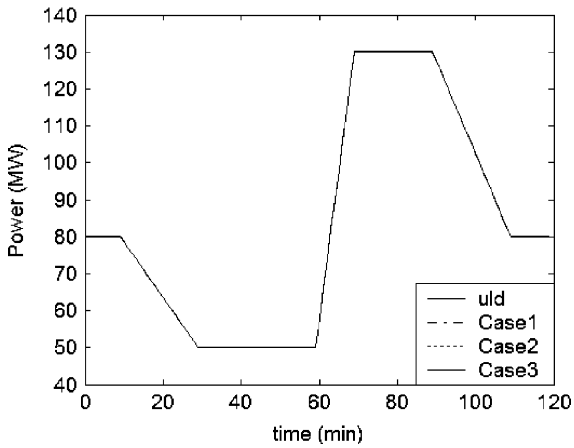

Fig. 15. Demand power set point trajectories.

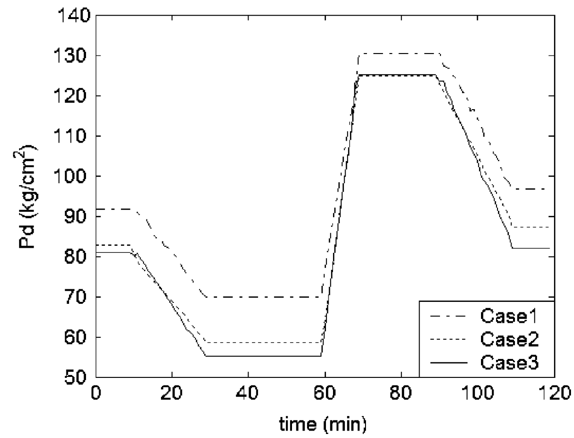

Fig. 16. Demand pressure set point trajectories.

the valve opening u_2 is increased as the number of objectives is increased, which is also desirable because the pressure valve was required to be kept open as wide as possible in case 3. In contrast, Fig. 14 shows that the results on the feedwater valve are about the same for all cases. This is because the solution space for the feedwater valve is very small, as shown in Fig. 11. All simulation results are improved as the number of objectives is increased. These optimal solutions are the values of *gbest* that are found though the PSO technique.

3) Set Point Trajectories: Finally, the power and pressure set points are obtained by set point scheduler (5), as shown in Figs. 15 and 16, respectively. The demand power set point (E_d) is almost the same for all cases as the unit load demand (Fig. 15). The demand pressure set points (P_d) mapped for different number of objective functions are shown in Fig. 16. It is interesting to note that although the demand power set point profile is almost the same for all cases, the demand pressure set point profiles differ significantly among cases. 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. 3. As additional objective functions are added in the optimization, the plant is operating more conservatively in lower pressure.

C. Comparison Among the PSO Techniques and GA

Table I compares the performance of the four different PSO techniques and GA. The results by GA are obtained using the general method of GA. GA used the same preference values and objective functions (6), [20]. The performance is given numerically by integrating the objective functions (6) over one load cycle (120 min) for all cases. For case 1, the first column is highlighted because only the first objective function $J_1(u)$ is included in the optimization. The remaining columns are not optimized. Similarly, in case 2 the first two columns are highlighted because both objective functions $J_1(u)$ and $J_2(u)$ are included in the optimization. However, in case 3 all columns are highlighted because all four objective functions are considered in the optimization.

As we examine the objective function $J_1(u)$ in all three cases, it is clear that HPSO performs the best among all optimization techniques. The objective function $J_2(u)$ shows that HPSO still performs the best for cases 2 and 3. However, as more objective functions are added in case 3, EPSO performs slightly better than HPSO for the objective functions $J_3(u)$ and $J_4(u)$. Table II summarizes the techniques that perform the best in each case. It shows that HPSO performs the best in cases 1 and 2, and as good as EPSO in case 3.

Table I compares the average performance of the techniques, rather than the instantaneous error for $J_1(u)$. Therefore, the values of about 13 in case 3 are obtained by integrating the generation error for 120 min. The average instantaneous error is about 0.108 MW. For example, the error is 0.08% at the load demand of 130 MW. Objective functions $J_3(u)$ and $J_4(u)$ reflect the negative of the openings of steam and water valves, respectively. Because the power-input operating window of the water valve opening is very small (Fig. 4), the objective function $J_4(u)$ is not included in the optimization by assigning a zero preference value. Objective function $J_3(u)$ accounts for energy loss due to the pressure drop across the steam valve. Because the pressure drop, and consequently the energy loss, increases as the valve closes, it is desired to keep it as widely open as possible. Thus, maximizing the steam valve u_2 or minimizing $-u_2$ will reduce loss due to the pressure drop in the steam valve. This makes the steam valve almost constant to the upper limit. This forces the boiler to lower the pressure demand in following the demand more closely (Fig. 16). To keep the pressure lower, fuel needs to be reduced (Fig. 12). However, this is at the expense of generation error; it reduces generation and thus slightly increases the generation error reflected in the objective function $J_1(u)$ (Table I, case 3).

In the basic PSO and HPSO, the weighting function is adjusted by the IWA (9), which gives a fast initial response and then allows the weight to decrease as the convergence is occurring. Moreover, EPSO has an evolutionary strategy for the weighting function (11b) that is adaptively changed to obtain a fast convergence. In contrast, the weight is a fixed constant in the CFA. Although the weight is selected to guarantee stability, it does not give the fast initial response, and before it converges,

TABLE I
COMPARISON OF PERFORMANCE FOR DIFFERENT PSO TECHNIQUES

Case	Case I				Case 2				Case 3			
Objective Optimization techniques	$J_1(u)$	$J_2(u)$	$J_3(u)$	$J_4(u)$	$J_1(u)$	$J_2(u)$	$J_3(u)$	$J_4(u)$	$J_1(u)$	$J_2(u)$	$J_3(u)$	$J_4(u)$
Basic PSO	6.9E-9	67.509	−95.247	−56.797	1.162	58.358	−103.345	−56.131	13.202	55.943	−106.232	−56.647
CFA	0.0019	63.779	−98.304	−56.557	1.485	59.912	−102.072	−56.131	13.215	56.403	−105.904	−56.495
HPSO	6.7E-9	68.195	−94.847	−56.682	0.074	56.110	−105.905	−56.352	11.625	55.940	−106.258	−56.850
EPSO	0.00086	65.906	−96.335	−56.873	0.425	57.063	−104.948	−56.486	13.122	55.942	−106.306	−57.168
GA	0.03000	61.122	−93.314	−56.135	1.538	58.417	−101.202	−55.557	13.476	56.707	−105.562	−55.571

TABLE II
BEST-PERFORMING TECHNIQUES FOR EACH CASE

Objective \ Case	Case 1	Case 2	Case 3
$J_1(u)$	HPSO	HPSO	HPSO
$J_2(u)$	X	HPSO	HPSO
$J_3(u)$	X	X	EPSO
$J_4(u)$	X	X	EPSO

the load demand changes dynamically in the FFPU. In other words, the CFA is designed for a static optimization problem and therefore is less efficient in a dynamic optimization problem. This is the disadvantage of the CFA.

In view of the computational complexity and efficiency, HPSO is preferred over EPSO because EPSO needs two evaluations of the performance per agent in each iteration. Nevertheless, all simulation results show that PSO techniques can be accommodated well in the multiobjective optimization problem. They can also be adopted for online implementation because the pressure set point needs to be updated only when the unit load demand is changed during the load cycle. Due to the fast convergence of PSO techniques, it is possible to search for the optimal solution at every different unit load demand. When the unit load demand is changing continuously, the optimization should be performed quickly if the unit is to be in the load-following mode. Therefore, a fast optimization scheme such as the PSO techniques is preferred. For some units, where the unit load demand is given in advance, the multiobjective optimization can be performed in advance, and the pressure set point can be made available in the form of a lookup table. However, when the unit load demand is given on a very short notice or the operation procedure changes due to changing priorities of different objective functions [given by β in (7)], the pressure set points need to be calculated again. Because the real power plant model is very large and of high order, any savings in computation time will be valuable.

V. CONCLUSION

The PSO technique is presented as an alternative optimization technique for solving a multiobjective optimization problem. The feasibility of using the PSO is demonstrated in the design of optimal set points for the multiobjective optimal power plant operation. The optimal mapping between the unit load demand and pressure set point is realized. Furthermore, the mapping can also be realized for time-varying load demand. This paper shows that improvements can be made to the basic PSO technique to solve the multiobjective optimization problem effectively. It has been demonstrated that the hybrid PSO technique improves the convergence and performs the best compared with other PSO techniques in FFPU. Finally, the feasibility of online implementation is discussed in the event that the unit load demand is given in advance.

Because the PSO techniques are random search with experimental trials, they have a weakness in the theoretical proof of convergence. Therefore, in future work, we plan to study the PSO techniques with stochastic analysis to show how the particles move to the optimal solution.

REFERENCES

[1] O. I. Elgerd, *Electric Energy Systems Theory: An Introduction*. New York: McGraw-Hill, 1971.
[2] IEEE Catalog Number 02TP160, *Tutorial on Modern Heuristic Optimization Techniques With Applications to Power Systems*, K. Y. Lee and M. A. El-Sharkawi, Eds. Piscataway, NJ: IEEE Power Engineering Society, 2002.
[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE Int. Conf. Neural Netw.*, vol. IV, pp. 1942–1948.
[4] C. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Comp. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.
[5] P. Angeline, "Using selection to improve particle swarm optimization," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1998, pp. 84–89.
[6] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "Hybrid particle swarm optimization based distribution state estimation using constriction factor approach," in *Proc. Int. Conf. SCIS & ISIS*, vol. 2, 2002, pp. 1083–1088.
[7] V. Miranda and N. Fonseca, "EPSO—evolutionary particle swarm optimization, a new algorithm with applications in power systems," in *Proc. IEEE Trans. Distribution Conf. Exhibition*, vol. 2, 2002, pp. 6–10.
[8] ——, "EPSO-best-of-two-worlds meta-heuristic applied to power system problems," in *Proc. Congr. Evolutionary Computation*, vol. 2, 2002, pp. 12–17.
[9] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1999, pp. 1951–1957.
[10] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *Proc. Congr. Evolutionary Computation*, pp. 84–88, 2000.
[11] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with non-smooth cost functions," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34–42, Feb. 2005.
[12] ——, "Economic load dispatch based on a hybrid particle swarm optimization," in *Proc. Int. Conf. Intell. Syst. Appl. Power Syst.*, 2003, CD ISAP03-070.pdf.

[13] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1998, pp. 60–73.
[14] H. C. Gery, "The evolution of coordinated control," *Inst. Soc. Amer. Power Symp.*, St. Petersburg, FL, 1988, pp. 109–112, paper no. 88-0417.
[15] R. D. Bell and K. J. Åström, *Dynamic Models for Boiler-Turbine-Alternator Units: Data Logs and Parameter Estimation for 160 MW Unit*. Lund, Sweden: Lund Institute of Technology, 1987, TFRT-3192.
[16] R. Garduno-Ramirez and K. Y. Lee, "Wide-range operation of a power unit via feedforward fuzzy control," *IEEE Trans. Energy Conv.*, vol. 15, no. 4, pp. 421–426, Dec. 2000.
[17] ——, "Multiobjective optimal power plant operation through coordinate control with pressure set point scheduling," *IEEE Trans. Energy Conv.*, vol. 16, no. 2, pp. 115–122, Jun. 2001.
[18] J. K. Salisbury, "Analysis of the steam-turbine reheat cycle," *ASME Trans. J. Eng., Power*, vol. 80, pp. 1629–1642, 1988.
[19] W. G. Buchwald, "Control performance with reduced valve pressure drop for energy conservation," *ISA Trans.*, vol. 14, no. 3, pp. 231–236, 1975.
[20] J. Chang, K. Y. Lee, and R. Garduno-Ramirez, "Optimal task decomposition agents of the multiagent power plant control system," in *Proc. Int. Conf. Intelligent Syst. Appl. Power Syst.*, 2003, CD ISAP03-085.pdf.

Jin S. Heo received the B.S. and M.S. degrees in electronics engineering from Inje University, Korea, in 1999 and 2001, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at the Pennsylvania State University, University Park.

His interests are multiobjective optimization in control system, intelligent distributed control, multiagent system, modeling and control of fuel cell power plants, and real-time embedded system.

Kwang Y. Lee (S'70–M'72–SM'86–F'01) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, the M.S. degree in electrical engineering from North Dakota State University, Fargo, and the Ph.D. degree in system science from Michigan State University, East Lansing, in 1964, 1968, and 1971 respectively.

He is currently Professor of electrical engineering and Director of the Power Systems Control Laboratory with the Pennsylvania State University. He has also been with Michigan State University, Oregon State University, and the University of Houston. His interests include power system control, operation, planning, and intelligent system applications to power systems.

Dr. Lee is an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS and Editor of IEEE TRANSACTIONS ON ENERGY CONVERSION. He is also a Registered Professional Engineer in the state of Texas.

Raul Garduno-Ramirez received the B.S. degree in electrical engineering from the National Polytechnic Institute (IPN), Mexico, the M.S. degree in electrical engineering from the Advanced Research Centre of the IPN, Mexico, and the Ph.D. degree in electrical engineering at the Pennsylvania State University, University Park in 1985, 1987, and 2000, respectively. He was a Fulbright Fellow during his Ph.D. program at Pennsylvania State University.

In 1986, he worked at the National Mechanical Laboratory, Tsukuba, Japan. Since 1987, he has been with the Electric Research Institute, Mexico, where he has been involved in the development of control systems for power plants. His current interests are in control software development and intelligent control.