APPLICATION OF NOVEL REINFORCEMENT LEARNING AUTOMATA APPROACH IN POWER SYSTEM REGULATION

MOHAMMAD KASHKI
Electrical and Computer Engineering Department,
University of Kerman,
Kerman, Iran
kashki@gmail.com

YOUSSEF L. ABDEL-MAGID
Electrical Engineering Program,
The Petroleum Institute,
Abu Dhabi, United Arab Emirate
yabdelmagid@pi.ac.ae

MOHAMMAD A. ABIDO
Department of Electrical Engineering,
King Fahd University of Petroleum & Minerals,
Dhahran, Saudi Arabia
mabido@kfupm.edu.sa

In this paper, a novel efficient optimization method based on reinforcement learning automata (RLA) for optimum parameters setting of conventional proportional-integral-derivative (PID) controller for AVR system of power synchronous generator is proposed. The proposed method is Combinatorial Discrete and Continuous Action Reinforcement Learning Automata (CDCARLA) which is able to explore and learn to improve control performance without the knowledge of the analytical system model. This paper demonstrates the full details of the CDCARLA technique and compares its performance with Particle Swarm Optimization (PSO) as an efficient evolutionary optimization method. The proposed method has been applied to PID controller design. The simulation results show the superior efficiency and robustness of the proposed method.

Keywords: Reinforcement learning automata; synchronous generator; AVR; PID; evolutionary computations.

1. Introduction

The PID controller is the most frequently used control element in the industrial world in comparison to other controllers such as adaptive controllers, artificial neural network based controllers, fuzzy and neuro-fuzzy controllers. It is estimated that, at least, 90% of the controllers employed in the industry are PIDs or its variants. The popularity of the PID controller is attributed to its simple structure, high
reliability, and robust performance in a wide range of operating conditions. Despite all of the PID controller good features, unfortunately, its appropriate gain tuning is still a problem in many practical industrial applications because of high order, time delay and nonlinearity of the plants.\textsuperscript{2} The normal tuning method in many applications is carried out using the classical tuning rules proposed by Ziegler-Nichols,\textsuperscript{3} which in general, does not yield optimal or near-optimal behavior in many industrial plants and just can be counted as a feasible solution.

In recent years, many heuristic methods for the optimum tuning of PID parameters such as genetic algorithms (GA) and simulated annealing (SA) have been proposed with noticeable success in solving complex optimization problems.\textsuperscript{4–9} Recently, a modern heuristic algorithm called Particle Swarm Optimization was proposed by Kennedy and Eberhart which is developed through simulation of a simplified social system and have been found to be robust in solving nonlinear optimization problems.\textsuperscript{10}

Stochastic learning automata can operate in random and unknown environments. They operate by selecting actions via a stochastic process; these actions operate on an environment and are assessed according to a measure of system performance.\textsuperscript{11} Howell \textit{et al.}\textsuperscript{12} first introduced an online optimization approach of this learning as Continuous Action Reinforcement Learning Automata (CARLA) which searches in continuous space for finding optimum solution. The method was successfully applied to several applications such as suspension control,\textsuperscript{12} online tuning of PID controller,\textsuperscript{13} digital filter design,\textsuperscript{14} power system stabilization,\textsuperscript{15,16} and power system regulation.\textsuperscript{17} It is shown that the performance of a system will improve by adaptation of learning automata units. However, when there are numerous actions, i.e., there is a large number of decision variables or the pre-specified variation interval of decision variables is long, the CARLA suffers low speed of convergence. Regarding this, a novel approach based on CARLA was developed which is called Combinatorial Discrete and Continuous Action Reinforcement Learning Automata (CDCARLA) that accelerate the convergence.\textsuperscript{18} CDCARLA comprises two successive steps; in the first step, the total variation interval of decision variables is divided into a finite number of sub-intervals and Discrete Action Reinforcement Learning Automata (DARLA) searches the optimum sub-interval of each variable in discrete space based on a pre-specified cost function which reflects the system performance. In the second step, the CARLA will look for the optimal values in the DARLA-specified sub-intervals.

Recently, CDCARLA was applied successfully in several applications such as optimum design of conventional and fuzzy logic based power system stabilizers,\textsuperscript{19–21} and fuzzy logic based controllers.\textsuperscript{22}

The generator excitation system maintains generator voltage and controls the reactive power flow using an automatic voltage regulator (AVR).\textsuperscript{23} The role of an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. Hence, the stability of the AVR system would seriously affect the security of the power system.
In this paper, application of CDCARLA technique for the optimum tuning of the PID controller of a synchronous generator is fully investigated and its performance is compared with a PSO-based PID controllers. Moreover, the robustness of both approaches due to plant uncertainties is verified.

2. PID Controller

The PID controller is composed of three main components: proportional, integral and derivative. Figure 1 shows this structure and components, where $y_{\text{ref}}(t)$ is the reference output, $e(t)$ is the error signal, $u(t)$ is the control signal, and $y(t)$ is the output. Each of the PID controller components has its own specific effect on the controller performance: the proportional component has the effect of increasing the loop gain to make the system less sensitive to disturbances, the integral component is used principally to eliminate steady-state errors, and the derivative action helps to improve closed loop stability. The gain parameters $k_p$, $k_i$, and $k_d$ are thus chosen to meet prescribed performance criteria, classically specified in terms of rise and settling times, overshoot and steady state error, following a step change in the reference output signal. The transfer function of PID controller can be expressed by Eq. (1).

$$ u(t) = k_p e(t) + k_i \int_0^t e(t) dt + k_d \frac{de(t)}{dt} \quad \text{(1)} $$

Thus, the decision variables involved in the optimization problem are the gain parameters: $k_p$, $k_i$, and $k_d$.

3. Automatic Voltage Regulator

The role of an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. A simple AVR system is composed of four main components, namely the amplifier, the exciter, the generator, and the sensor. In this paper, a linearized model of the AVR is considered which takes into account the major time constant of the AVR components and ignores nonlinearities, as shown in Fig. 2, where, $V_{\text{ref}}$ is the reference voltage and $V_r$ is the excitation voltage.

![Fig. 1. PID structure.](image)
Table 1 summarizes the typical range and the considered value of linearized model parameters.

4. CDCARLA Algorithm

Learning automata (LA) are adaptive decision-making units that can learn to choose the optimal action from a set of actions by interaction with an unknown random environment. At each instant $n$, the LA chooses an action $\alpha_n$ from its action probability distribution and applies it to the random environment providing a stochastic response which is called a reinforcement signal, to the LA. Then the LA uses the reinforcement signal and a learning algorithm to update the action probability distribution.

Generally, in the CARLA approach when there is a large number of actions due to a large number of decision variables or wide variation interval of these variables in complex optimization problem, the number of iterations for obtaining optimum values will increase significantly. In the proposed method, this concept is taken into account, thus, the design procedure is divided into two successive steps to speed up the optimization procedure convergence. In the first step, the total variation interval of each decision variable is divided into sub-intervals of usually equal length, and then the DARLA algorithm determines the optimum sub-interval of each decision variable. In the second step, the CARLA algorithm searches for the optimum value of each decision variable in the predetermined optimal sub-interval. Both algorithms find the optimum values of decision variable using a predefined cost function.
One of the most important features of the CARLA and CDCARLA as a method of control system design is the fact that minimal knowledge of the plant under investigation is required. Since they optimize a performance index based on input/output relationships only, far less information than other design techniques is needed. Further, as the CDCARLA search is directed towards decreasing a specified objective function, the net result is a controller, which ultimately meets the performance criteria. In addition, because CARLA and CDCARLA do not need an explicit mathematical relationship between the performance of the system and the search update, they offer a more general optimization methodology than conventional analytical techniques. Details of the CDCARLA algorithm are as follows.

4.1. DARLA step

In the DARLA optimization algorithm, the total variation interval of each decision variable is divided into numbers of generally equal length sub-intervals. For each decision variable, an individual DARLA is considered which runs in a parallel implementation with other DARLAs. The only interconnection between DARLAs is through the environment and via a shared cost function. The computational flow of DARLA can be described as follows.

Discrete probability distribution function. DARLA considers a discrete probability distribution function (DPDF) for each decision variable which is initially uniform and can be defined as

\[ f_{i}^{(0)}(d_{i}) = \begin{cases} \frac{1}{N_{i}} & d_{i} = 1, 2, \ldots, N_{i}, \\ 0 & \text{other}, \end{cases} \quad i = 1, 2, \ldots, n, \tag{2} \]

where \( f_{i}^{(0)} \) is the initial DPDF for the \( i \)th decision variable; \( N_{i} \) is the number of sub-intervals of the \( i \)th decision variable and \( n \) is the number of decision variables.

Stochastic selection. In each iteration, a cumulative probability distribution function is computed for each decision variable. Then, a discrete action, i.e., the value of decision variables in this application, based on this function is stochastically provided as

\[ \sum_{d_{i}=1}^{N_{i}} f_{i}^{(k)}(d_{i}) = z_{i}^{(k)}, \tag{3} \]

where \( z \) is a random number that varies uniformly in the range \([0, 1]\).

Cost function. The objective of DARLA algorithm is to find the optimum sub-interval for each decision variable that minimizes a predefined cost function, i.e., objective function. In DARLA, the center of selected sub-intervals is used for calculating of cost function value.
Reinforcement signal. The reinforcement signal indicates the relative suitability of selected action. In other words, the lower value of reinforcement signal implies that the selection of sub-interval was poor while a higher value indicates a good selection. One of the common mappings between the cost function and the reinforcement signal can be expressed as

$$\beta^{(k)}(J^{(k)}) = \min \left\{ 1, \max \left\{ 0, \frac{J_{\text{avg}} - J^{(k)}}{J_{\text{avg}} - J_{\text{min}}} \right\} \right\},$$

where $\beta^{(k)}$ and $J^{(k)}$ are the reinforcement signal and cost value in $k$th iteration, respectively; $J_{\text{avg}}$ and $J_{\text{min}}$ are the average and minimum of previous cost values, respectively.

This definition of reinforcement signal performs a reward/inaction rule in DPDFs modification. In other words, if the current selected action is less than the mean value of the previous cost, i.e., $\beta = 0$, then no modification of CPDFs will be performed (inaction) and, if the selected action leads to a cost value less than the minimum of the previous cost, i.e., $\beta = 1$, then maximum reinforcement will be done (reward).

Updating DPDFs. At the end of each iteration, the DARLA algorithm learns about selection actions of that iteration. The learning logic is that, if the selection of a sub-interval leads to good performance, likely neighbor sub-intervals have relative good performance. The updating rule of DPDFs can be described as

$$f_i^{(k+1)}(d_i) = \alpha_i^{(k)} \left[ f_i^{(k)}(d_i) + \beta^{(k)} Q_i(d_i) \right],$$

where $f_i^{(k)}$ is the DPDF of the $i$th variable in the $k$th iteration; $Q_i$ is a Gaussian function centralized in $\tilde{d}_i$, the selected sub-interval of $i$th decision variable, and is defined as

$$Q_i(d) = q 2^{-(d - \tilde{d}_i)^2},$$

where $q$ is a positive constant that determines the speed and resolution of the learning algorithm. $\alpha$ in Eq. (5) is a normalization factor and is defined as

$$\alpha_i^{(k)} = \frac{1}{\sum_{d_i=1}^{N_i} f_i^{(k)} + \beta^{(k)} Q_i(d_i)}. $$

Now, the DARLA algorithm steps will continue with the new DPDFs and after sufficient number of iteration it converges to the optimum sub-interval for each decision variable.

4.2. CARLA step

After obtaining the optimum sub-intervals by the DARLA algorithm, the CARLA searches for optimal values of decision variables in the corresponding optimal sub-intervals. The CARLA algorithm is similar to DARLA except that the DPDFs will be replaced with Continuous Probability Distribution Function (CPDF). In fact, the
only difference between CARLA and DARLA is that CARLA takes the actions in a continuous space instead of a discrete one.

In this algorithm, the probability distribution functions are initially defined uniformly in continuous space over the optimum sub-interval as

\[
 f_i^{(0)}(x_i) = \begin{cases} 
 \frac{1}{x_{i,\text{max}} - x_{i,\text{min}}}; & x_i \in [x_{i,\text{min}}, x_{i,\text{max}}], \\
 0; & \text{other}, 
\end{cases} 
\]

(8)

where \( f_i^{(0)} \) is the initial CPDF for \( i \)th decision variable; \( x_{i,\text{min}} \) and \( x_{i,\text{max}} \) are the lower and upper bounds of the \( i \)th decision variable optimal sub-interval, respectively.

The stochastic action selection, cost function and reinforcement signal calculations are similar to DARLA. Also, the CPDF updating in CARLA has the same philosophy, but differs slightly as:

\[
 f_i^{(k+1)}(x_i) = \alpha_i^{(k)} f_i^{(k)}(x_i) + \beta^{(k)} H_i(x_i),
\]

(9)

where \( H_i \) is a Gaussian function centralized in \( \tilde{x}_i \), the selected value of \( i \)th decision variable, and is defined as

\[
 H_i(x_i) = \frac{gh}{x_{i,\text{max}} - x_{i,\text{min}}} \exp\left(-\frac{(x_i - \tilde{x}_i)^2}{2(gw(x_{i,\text{max}} - x_{i,\text{min}}))^2}\right),
\]

(10)

where \( gh \) and \( gw \) are the height and width of \( H \) function and determine the speed and resolution of the learning algorithm. The normalization factor can be defined as

\[
 \alpha_i^{(k)} = \frac{1}{\int_{x_{i,\text{min}}}^{x_{i,\text{max}}} f_i^{(k)}(x) + \beta^{(k)} H_i(x) \, dx}.
\]

(11)

Similar to DARLA, by repeating the above steps for enough number of iteration the optimum values of decision variables can be determined.

Hence, the DARLA and CARLA algorithms do not require the knowledge of the system dynamics, but the designer should be aware of the system behavior in order to define an appropriate cost function.

Figure 3 shows the flowchart of the DARLA and CARLA algorithms. Hence, the PDF denotes DPDF for DARLA step and CPDF for CARLA.

5. Implementation

In this paper, the optimum PID parameter setting is expressed as the following optimization problem:

Minimize \( J \)

Subject to:

\[
 k_p^{\text{min}} \leq k_p \leq k_p^{\text{max}},
\]

\[
 k_i^{\text{min}} \leq k_i \leq k_i^{\text{max}},
\]

\[
 k_d^{\text{min}} \leq k_d \leq k_d^{\text{max}}.
\]

(12)
In our application, the minimum and maximum values of the PID gains are set at 0.0 and 5; respectively.

The CDCARLA optimization technique is applied for solving the optimization problem defined in Eq. (12). Moreover, for assessment of effectiveness and performance of the proposed CDCARLA–PID controller, a particle swarm optimization has been applied to this optimization problem.
5.1. Optimal CDCARLA–PID controller

In CDCARLA implementation, decision variables are PID parameters and their total variation intervals are divided to 10 equally-length sub-intervals. Both DARLA and CARLA algorithms will terminate if (1) the best solution does not change for more than 50 chains or, (2) the number of iterations reaches 100. The other parameters of CDCARLA are selected as $q = 2$, $g_w = 0.003$ and $g_h = 1$, as a result of previous investigations which show robust DARLA and CARLA performance over a wide range of applications.\textsuperscript{17–22}

In this paper, several cost function have been considered. In general, the PID controller design methods the integrated absolute error ($IAE$), or the integral of squared-error ($ISE$), or the integrated of time-weighted-squared-error ($ITSE$) is often employed in control system design because it can be evaluated analytically in the frequency domain.\textsuperscript{3–6} The three integral performance criteria in the frequency domain have their own advantages and disadvantages. For example, a disadvantage of the $IAE$ and $ISE$ criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the $ISE$ performance criterion weighs all errors equally independent of time. Although the $ITSE$ performance criterion can overcome the disadvantage of the $ISE$ criterion, the derivation processes of the analytical formula are complex and time-consuming.\textsuperscript{6} The $IAE$, $ISE$, and $ITSE$ performance criterion formulas are as follows:

$$IAE = \int_{t=0}^{\infty} |e(t)|dt ,$$ \hspace{1cm} (13)

$$ISE = \int_{t=0}^{\infty} e^2(t)dt ,$$ \hspace{1cm} (14)

$$ITSE = \int_{t=0}^{\infty} te^2(t)dt ,$$ \hspace{1cm} (15)

where, $e$ is the error signal, i.e., $e(t) = r(t) - y(t)$, and $r$ is the reference value. In addition to above performance criteria, another cost function proposed by Gaing\textsuperscript{25} to overcome the above disadvantages is considered and is expressed as

$$J = (1 - e^{-\lambda}) \cdot (M_p + E_{ss}) + e^{-\lambda} \cdot (t_s - t_r) ,$$ \hspace{1cm} (16)

where, $J$ is the cost function, $\lambda$ is the weighting factor and set between 0.8 to 1.5, $M_p$ is the overshoot of the output signal, $E_{ss}$ is the steady error, and $t_s$, $t_r$ are the settling and rising time respectively. The weighting factor can be set to larger than 0.7 to reduce the overshoot and steady-state error. On the other hand, it can be set to be smaller than 0.7 to reduce the rise time and settling time. In this paper, the weighting factor is set to 1.0, i.e., $\lambda = 1.0$. In addition, for calculating cost functions a step change is considered as reference terminal voltage.

The CDCARLA algorithm is applied for each of cost functions defined in Eqs. (13)–(16). Typically, the variation of proposed cost function values due to iterations in Eq. (16) is shown in Fig. 4.
Also, Fig. 5 typically depicts variation trends of DPDFs in DARLA algorithm for proposed cost function in Eq. (16). For the above, DARLA took 40 iterations and 356 seconds, and CARLA took 98 iterations and 873 seconds.

As discussed in Sec. 4.1, at the end of DARLA step, the sub-interval with the highest probability value will be selected for CARLA step. For example, according to the above figures, optimal sub-intervals of PID proportional, integral and derivative gain parameters are in the 7th, 1st, and 2nd sub-interval, respectively.

Also, the variation trends of CPDFs in the CARLA algorithm for proposed cost function in Eq. (16) are shown in Fig. 6.

Similarly, the gain value with the highest probability value is the optimal value for that gain parameter. For example, according to the above figures, optimal values of the PID proportional, integral and derivative gain parameters are 3.4444, 0.3737, and 0.7676, respectively.

Table 2 summarizes the optimum sub-interval and optimum value of PID gain parameters for different cost functions.

### 5.2. Optimal PSO–PID controller

For comparison purposes, Particle-Swarm Optimization (PSO)\textsuperscript{26} is implemented in this work. This new approach that is based on “constructive cooperation” features many advantages; it is simple, fast and can be coded in few lines, and its storage requirement is minimal.

PSO starts with a population of random solutions “particles” in a D-dimension space. The \(i\)th particle is represented by \(X_i = (x_{i1}, x_{i2}, \ldots, x_{iD}); \ i = 1, 2, \ldots, n_p\). Each particle keeps track of its coordinates in hyperspace, which are associated with the fittest solution it has achieved so far. The value of the fitness for particle \(i\) (pbest) is also stored as \(P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})\). The global version of the PSO keeps track of the overall best value (gbest), and its location, obtained thus far by any particle in the population.\textsuperscript{26,27}
PSO consists of, at each step, changing the velocity of each particle toward its pbest and gbest according to following.

\[ v_{id} = w \times v_{id} + c_1 \times \text{rand}() \times (p_{bd} - x_{id}) + c_2 \times \text{rand}() \times (p_{gd} - x_{id}), \]
\[ i = 1, 2, \ldots, n_p; \quad d = 1, 2, \ldots, D, \]

(17)

where \( w \) is inertia weight which leads to reduction in the number of iterations, the velocity of particle \( i \) is represented as \( V_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \), \( p_{bd} \) = pbest and
Fig. 6. CPDFs variations of PID gain parameters in CARLA algorithm. (a) proportional gain, (b) derivative gain, (c) integral gain.

Table 2. Optimum sub-interval and optimum value of CDCARLA PID gain parameters due to different cost functions.

<table>
<thead>
<tr>
<th>Gain</th>
<th>Interval</th>
<th>Value</th>
<th>IAE</th>
<th>Value</th>
<th>ISE</th>
<th>Value</th>
<th>ITSE</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_i$</td>
<td>[0.0 0.5]</td>
<td>0.3737</td>
<td>[0.5 1.0]</td>
<td>0.5230</td>
<td>[0.5 1.0]</td>
<td>0.7082</td>
<td>[0.5 1.0]</td>
<td>0.5776</td>
</tr>
<tr>
<td>$k_d$</td>
<td>[0.5 1.0]</td>
<td>0.7576</td>
<td>[0.5 1.0]</td>
<td>0.9064</td>
<td>[0.5 1.0]</td>
<td>0.8514</td>
<td>[0.5 1.0]</td>
<td>0.8669</td>
</tr>
</tbody>
</table>

$p_{gd} = g_{best}$, $c_1$ and $c_2$ are acceleration constants. Acceleration is weighted by a random term, with separated random numbers being generated for acceleration toward pbest and gbest. The position of the $i$th particle is then updated according to following.

$$x_{id} = x_{id} + v_{id}.$$  \hspace{1cm} (18)

Similarly, PSO is implemented in conjunction with the optimization algorithm defined in Eq. (12) for the different cost functions in Eqs. (13)–(16). The parameters
Table 3. Optimum PSO–PID gain parameters for different cost functions.

<table>
<thead>
<tr>
<th>Gain</th>
<th>Proposed Eq. (16)</th>
<th>IAE</th>
<th>ISE</th>
<th>ITSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_p$</td>
<td>4.1945</td>
<td>6.3973</td>
<td>11.9197</td>
<td>8.4960</td>
</tr>
<tr>
<td>$k_i$</td>
<td>0.5892</td>
<td>0.5231</td>
<td>2.0123</td>
<td>0.6649</td>
</tr>
<tr>
<td>$k_d$</td>
<td>0.9350</td>
<td>1.1271</td>
<td>1.9850</td>
<td>1.4822</td>
</tr>
</tbody>
</table>

of the PSO algorithm are as follows:

- members of particles are $x_1 = k_p$, $x_2 = k_i$, and $x_3 = k_d$;
- number of population: $n_p = 30$;
- acceleration constants: $c_1 = c_2 = 2$;
- inertia weight, $w$, is linearly decreasing from 0.9 to 0.4;
- maximum number of iterations is 200.

Table 3 summarizes the optimum value of PSO–PID gain parameters for different cost functions.

6. Results and Discussion

6.1. Performance evaluation

To assess the effectiveness of optimal CDCARLA–PID and PSO–PID controllers, they are implemented on the AVR system shown in Fig. 2 and their performances in the tracking of a reference terminal voltage are simulated in MATLAB. In this simulation, the reference terminal voltage is 1.1 pu for 2 s and then 0.9 pu thereafter.

In addition, the performance of the optimal PID controllers are compared with a PID controller designed with the classic Ziegler–Nichols method, i.e., $k_p = 1.0228$, $k_i = 1.8423$, and $k_d = 0.1357$.

The simulation results of the above PID controllers for different cost functions are given in Fig. 7.

It can be seen that for all of the cost function schemes, the optimal CDCARLA–PID and PSO–PID controllers outperform that of the Ziegler–Nichols PID controller. Moreover, the CDCARLA–PID controller has better performance than that of the PSO–PID controller in terms of overshoot, steady-state error, settling time, and number of oscillations. For quantitative comparison of optimal PID controllers, some performance indexes are considered including overshoot, steady-state error, rise time, and settling time. Table 4 summarizes these performance indexes for optimal controllers. For each index, the first value is with the CDCARLA–PID and the second one is with the PSO–PID.

6.2. Robustness assessment

For robustness verification of the optimal PID controllers, some uncertainties for the components of the system under study are considered. These uncertainties include
10% increase in the amplifier gain, $K_A$; 10% decrease in the amplifier time constant, $\tau_A$; 5% increase in exciter gain, $K_E$; 10% decrease in generator gain, $K_G$; 20% increase in sensor time constant, $\tau_S$.

The reference terminal voltage is a 0.8 pu step change for 2 s and 1.0 pu thereafter. Figure 8 shows the performance of PID controllers for this uncertain system.
It can be seen that the CDCARLA–PID are more robust than that of PSO–PID controller.

7. Conclusion
In this paper, a two-step novel approach based on reinforcement learning automata called Combinatorial Discrete and Continuous Action Reinforcement Learning Automata (CDCARLA) was implemented to search for the optimal PID controller parameters for AVR system of power synchronous generator. Here are some facts about this novel method:

(1) In Ref. 17 the optimum PID controller for this system had been designed using a one-step CARLA design approach. The results showed it took around 100 iterations for convergence. But as shown in this paper results, the CDCARLA...
design method just takes around 40 iterations to be converged. This is because
the design process is divided into two successive steps as described earlier.

(2) For comparison of performance and robustness the CDCARLA have been com-
pared with PSO optimization method that have had good sounds in solving
optimization problems. The simulation results show that the CDCARLA have
better performance and robustness than that of PSO for different cost functions.

(3) It is worth mentioning that CDCARLA is not a perfect optimization method.
Decision on number of sub-intervals, its learning parameters and cost function
definition is still an important problem which improper selection would lead to
fail to find convenient solution.

(4) We recommend the application of CDCARLA for other optimization problems
that satisfactory results are anticipating.

Acknowledgments

M. Kashki acknowledges the support of Key Sun Pars Consultants Engineering
Co., which is a famous and good sound engineering company in the fields of oil,
gas and petrochemical. Dr. Y. Abdel-Magid and Dr. M. A. Abido acknowledge the
support of the Petroleum Institute, Abu Dhabi, UAE, and King Fahd University
of Petroleum & Minerals, Saudi Arabia, respectively.

References

1. M. Santos, J. M. de la Cruz, S. Dormido and A. P. de Madrid, Between fuzzy-PID and
2. A. Visioli, Tuning of PID controllers with fuzzy logic, IEE Proc. Control Theory and
3. J. Ziegler, G. Nichols and N. Y. Rochester, Optimum setting for automatic controllers,
Trans. ASME (1942), pp. 759–768.
4. T. L. Seng, M. B. Khalid and R. Yusof, Tuning of a neuro-fuzzy controller by genetic
5. T. Kawabe and T. Tagami, A real coded genetic algorithm for matrix inequality
design approach of robust PID controller with two degrees of freedom, Proc. IEEE
6. R. A. Krohling, H. Jaschek and J. P. Rey, Designing PI/PID controller for a motion
control system based on genetic algorithm, Proc. 12th IEEE Int. Symp. Intelligent
7. D. P. Kwok and F. Sheng, Genetic algorithm and simulated annealing for optimal
pp. 707–713.
Instrument Electrical Engineering Conference on Genetic Algorithm and Engineering