A Lyapunov-based Extension to PSO dynamics for Continuous Function Optimization

Sayantani Bhattacharya
Artificial Intelligence Lab
ETCE Dept
Jadavpur University
Kolkata 700032, India
bhattacharya.sayantani@gmail.com

Amit Konar
Artificial Intelligence Lab
ETCE Dept
Jadavpur University
Kolkata 700032, India
konaramit@yahoo.co.in

Atulya Nagar
Intelligence and Distributed Systems Lab
Deanery of Business & Computer Science
Liverpool Hope University
Liverpool L169JD
nagara@hope.ac.uk

Abstract

The paper proposes three alternative extensions to the classical global-best particle swarm optimization dynamics, and compares their relative performance with the classical particle swarm optimization algorithm. The first extension, which readily follows from the well-known Lyapunov's stability theorem, provides a mathematical basis of the particle dynamics with a guaranteed convergence at an optimum. The inclusion of local and global attractors to this dynamics leads to faster convergence speed and better accuracy than the classical one. The second extension augments the velocity adaptation equation by a negative randomly weighted positional term of individual particle, while the third extension considers the negative positional term in place of the inertial term. Computer simulations further reveal that the last two extensions outperform both the classical and the first extension in convergence speed and accuracy.

1. Introduction

The concept of particle swarm, although initially introduced for simulating human social behavior, has become very popular these days as an efficient means for intelligent search and optimization. The Particle Swarm Optimization (PSO) [1], [2], as it is called now, emulates the swarming behavior of insects, animals herding, birds flocking and fish schooling, where these swarms forage for food in a collaborative manner.

Since its inception, the research on PSO is centered on the improvement of the particle dynamics and the algorithm. There exists an extensive literature on improving the performance of the PSO algorithm [1]-[16]. In this paper, we improve the swarm behavior in terms of accuracy and convergence time by determining a suitable dynamics, and then attempted to empirically determine the optimal parameter settings.

The formal basis of our study originates from the well-known Lyapunov's stability theorem of classical control theory which provides the necessary conditions for stability of a dynamical system. In this paper, we indirectly used Lyapunov's stability theorem to determine a dynamics that necessarily converges to an optima of the Lyapunov-like search landscape. The principles of guiding particle dynamics towards the global and local optima, here too, is realized by adding local and global attractor terms in the modified PSO dynamics. The rationale of selecting a dynamics that converges at one of the optima on a multimodal surface, and the principle of forcing the dynamics to move towards local and global optima together makes it attractive for use in continuous nonlinear optimization.

The particle dynamics derived from the Lyapunov-like benchmark functions usually includes a factor involving a negative position term. Augmentation of the velocity adaptation rule by this positional term both in presence and absence of the inertial term is studied. Computer simulations undertaken on a set of five benchmark functions reveals that the above extensions of the PSO dynamics results in a significant improvement in the PSO algorithm with respect to both convergence speed and accuracy.

The rest of the paper is organized as follows. Section 2 provides the typical PSO dynamics. In section 3, we propose the extensions of the PSO dynamics. Experimental results are given in section 4. Finally, conclusions are listed in section 5.

2. Particle Swarm Optimization Dynamics

In this section, we briefly outline one typical PSO dynamics, and the PSO algorithm. In PSO, the

978-0-7695-3325-4/08 $25.00 © 2008 IEEE
DOI 10.1109/EMS.2008.62

particles are flying randomly in the D-dimensional search space. Each particle's position is a solution to a problem in the search space. The particles maintain a memory of their previous best position and the global best position found so far by the entire swarm. These are used to compute a new velocity and position for the particles until the termination criterion is not reached. The global-best (g-best) PSO dynamics for the j[th] particle in the i[th] dimension is given in equation 1 and equation 2.

$$v_{j,i}(t+1) = \omega v_{j,i}(t) + \alpha^{l}(t)(p^{l}_{j,i}(t) - x_{j,i}(t)) + \alpha^{g}(t)(p^{g}_{i}(t) - x_{j,i}(t)) \tag{1}$$

$$x_{j,i}(t+1) = x_{j,i}(t) + v_{j,i}(t+1) \tag{2}$$

where, $v_{j,i}(t)$ is the i^{th} component of the velocity vector of particle j at t^{th} iteration, $x_{j,i}(t)$ is the i^{th} component of the position vector of particle j at t^{th} iteration, $p^{l}_{j,i}(t)$ is the i^{th} component of the personal (local) best position of particle j, so far achieved until iteration t, $p^{g}_{i}(t)$ is the i^{th} component of the global best position found so far by the entire swarm at iteration t, ω is the inertia factor, $\alpha^{l}(t)$ denotes local acceleration coefficient at time t, $\alpha^{g}(t)$ denotes global acceleration coefficient at time t. Empirically, ω is a random no. in [0, 1], $\alpha^{l}(t)$ and $\alpha^{g}(t)$ are random coefficients in [0, 2] and [0, 4] respectively. Inertia factor ω is selected randomly only once in the PSO algorithm, whereas $\alpha^{l}(t)$ and $\alpha^{g}(t)$ are selected randomly in each iteration of the PSO algorithm.

3. Proposed Extensions of the Classical PSO Dynamics

3.1. Identifying a Stable Dynamics for a Lyapunov-like Surface

In this section, we would look for a dynamics that has a tendency to move towards optima, which, however, need not essentially be the global optima. This can be attained by identifying a suitable dynamics that ensures asymptotic stability in the vicinity of an optimum over the search landscape. This needs an additional restriction on the surface to satisfy the necessary conditions to be Lyapunov-like [20]. If a suitable dynamics ensuring the convergence to an optimum is identified, we can control the motion of the particles towards the global/local optima by adding global and local attractors in the dynamics as used in the PSO dynamics.

According to Lyapunov's Theorem [17]-[19], the asymptotic stability of an equilibrium state guided by the dynamics $\frac{dx_i}{dt}$ is ascertained if

$$\frac{df}{dt} = \sum_{i=1}^{D} \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} < 0 \tag{3}$$

where, $f(x)$ is the objective function. The inequality (3) essentially holds when $\frac{dx_i}{dt} = -\frac{\partial f}{\partial x_i}$. (4)

It is indeed important to note that the condition (4) holds for the i[th] dimension of a particle roaming over the Lyapunov-like surface for $1 \le i \le D$.

We now define Lyapunov-based PSO dynamics (LyPSO) by adding the local and global attractor terms of classical PSO to the derived expression for asymptotically stable Lyapunov dynamics, given in (5)-(6).

$$v_{j,i}(t+1) = -\omega \frac{\partial f}{\partial x_i} + \alpha^{l}(t)(p^{l}_{j,i}(t) - x_{j,i}(t)) + \alpha^{g}(t)(p^{g}_{j}(t) - x_{j,i}(t)) \tag{5}$$

$$x_{j,i}(t+1) = x_{j,i}(t) + v_{j,i}(t+1) \tag{6}$$

The first term in the right hand side of equation (5) ensures motion of the particle towards minima, while the second and third term controls the motion towards local and global optima respectively. For example, we have considered here five well-known benchmark functions as mentioned in Table 1. Table 2 provides the derived dynamics for these benchmarks.

Table 1. Benchmark functions used in the experiment

Function name	Function Expression
Sphere function	$f1(\mathbf{X}) = \sum_{i=1}^{D} x_i^2$
Rosenbrock function	$f2(\mathbf{X}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigin's function	$f3(\mathbf{X}) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$
Ackley function	$f4(\mathbf{X}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i) + 20 + e$
Griewank function	$f5(\mathbf{X}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

It is apparent from Table 2 that $\frac{dx_i}{dt}$ obtained for different Lyapunov-like surfaces include a factor of

($-x_i$) with or without some conditions. For example, we can extract $-x_i$ from the expressions of *f3*, *f4* and *f5* functions when x_i is very small, for *f2*, the condition is $x_i x_{i+1} >> -\frac{1}{200}$ and there is no condition in case of *f1*. Consequently, instead of computing $\frac{dx_i}{dt}$ by the approach stated earlier, we can simply add a term $-\omega x_i$ to the ith component of the updated velocity in the classical PSO. The resulting dynamics then looks like (7) – (8).

$$v_{j,i}(t+1) = -\omega x_{j,i}(t) + \omega v_{j,i}(t) + \alpha^l(t)(p^l_{j,i}(t) - x_{j,i}(t)) + \alpha^g(t)(p^g_i(t) - x_{j,i}(t)) \quad (7)$$

$$x_{j,i}(t+1) = x_{j,i}(t) + v_{j,i}(t+1) \quad (8)$$

The dynamics given by (7) - (8) is referred to as Position-based PSO (PPSO).

Table 2. The derived dynamics for the selected benchmark functions

Function	$\frac{dx_i}{dt}$
*f*1	$-2x_i$
*f*2	$-(400x_i^2 + 2)x_i + 2(1 + 200x_i x_{i+1})$
*f*3	$-2x_i - 20\pi\sin(2\pi x_i)$
*f*4	$-\frac{4}{D}\left(\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)\right)\frac{x_i}{\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}$ $-\frac{2\pi}{D}\sin 2\pi x_i\left[\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)\right]$
*f*5	$-\frac{x_i}{2000} - \frac{1}{\sqrt{i}}\sin\left(\frac{x_i}{\sqrt{i}}\right)\left[\prod_{j=1, j\neq i}^{D}\cos\left(\frac{x_j}{\sqrt{j}}\right)\right]$

We consider a third category of the dynamics, where the inertial term is dropped from the PSO dynamics, indicated in (9) - (10). The modified dynamics, called Steepest-PSO (SPSO) is

$$v_{j,i}(t+1) = -\omega x_{j,i}(t) + \alpha^l(t)(p^l_{j,i}(t) - x_{j,i}(t)) + \alpha^g(t)(p^g_i(t) - x_{j,i}(t)) \quad (9)$$

$$x_{j,i}(t+1) = x_{j,i}(t) + v_{j,i}(t+1) \quad (10)$$

In the next section, we would justify the reason for accuracy and speed-up of PPSO and SPSO over classical PSO.

3.2. The Rationale of Speed-up of the PPSO and SPSO Dynamics over Classical PSO

To compare the relative performance in speed-up and convergence of the proposed algorithms, we study the stability behavior of the proposed PPSO and SPSO dynamics, in absence of the local and the global attractors. Theorems 1 to 2 provide interesting results, indicating asymptotic stability of the SPSO and PPSO dynamics to the origin irrespective of the search landscape, whereas Theorem 3 indicates asymptotic stability of the classical PSO to a stable point, which need not essentially be the origin. The rate at which the particle position approaches the origin further indicates that the speed of convergence of the SPSO algorithm far exceeds that of PPSO, while the speed of PPSO algorithms beats classical PSO.

Theorem 1: *The dynamics of the jth particle in the ith dimension given by*

$$v_{j,i}(t+1) = -\omega x_{j,i}(t) \quad (11)$$

has a stable point at the origin, when $\omega \le 1$.

Proof:

$$v_{j,i}(t+1) = \frac{x_{j,i}(t+1) - x_{j,i}(t)}{(t+1) - (t)} = x_{j,i}(t+1) - x_{j,i}(t) = (E-1)x_{j,i}(t) \quad (11a)$$

where, $x_{j,i}(t+1) = x_{j,i}(t) + \Delta x_{j,i}(t) = (1+\Delta)x_{j,i}(t) \equiv Ex_{j,i}(t)$ and E is called the extended difference operator. Combining (11) and (11a) we have,

$$(E - (1-\omega))x_{j,i}(t) = 0,$$

the solution of which is

$$x_{j,i}(t) = A(1-\omega)^t, \quad (12)$$

where A is a constant. The expression (12) indicates that for $\omega < 1$, $x_{j,i}(t) \to 0$ when $t \to \infty$. Therefore, the dynamics is asymptotically stable at the origin for $\omega < 1$. When $\omega = 1$, $x_{j,i}(t) = 0$ at all time t. Hence, the theorem follows.

Theorem 2: *The dynamics of the jth particle in the ith dimension given by*

$$v_{j,i}(t+1) = \omega v_{j,i}(t) - \omega x_{j,i}(t) \quad (13)$$

is asymptotically stable with a stable point at the origin, when $\omega < 1$.

Proof: Since $x_{j,i}(t+n) \equiv E^n x_{j,i}(t)$ for positive/negative integer n, (13) can be reduced to

$$(E^2 - 1 + \omega)x_{j,i}(t) = 0 \quad (13a)$$

The condition for asymptotic stability for $\omega < 1$ follows readily from the solution of (13a).

Theorem 3: *The dynamics of j^{th} particle in the i^{th} dimension given by*

$$v_{j,i}(t+1) = \omega v_{j,i}(t) \quad (14)$$

is asymptotically stable and it converges to a stable point, which need not essentially be zero.

Proof: Similar to the proof of Theorem 1 and 2.

Table 3 provides the results of computation of $\frac{dx_i}{dt}$ for SPSO, PPSO and classical PSO from equations (11), (13), and (14) respectively. Figure 1 shows the variation of $\frac{dx_i}{dt}$ with respect to time for $\omega = 0.6$. It is apparent from the graph that in the absence of local and global attractors, the dynamics of SPSO converges faster than that of PPSO, which further converges faster than the classical PSO.

Table 3. $\frac{dx_i}{dt}$ for SPSO, PPSO and classical PSO

Dynamics	$\frac{dx_i}{dt}$
SPSO	$A(1-\omega)^t \log_e(1-\omega)$
PPSO	$\frac{(A-B)}{2}(1-\omega)^{t/2} \log_e\left(\sqrt{1-\omega}\right)$
Classical PSO	$B\omega^t \log_e \omega$

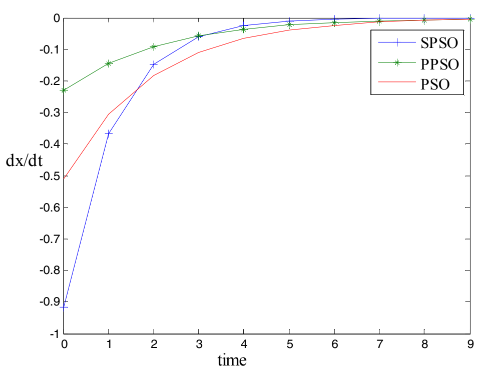

Figure 1. Variation of $\frac{dx_i}{dt}$ with time for $\omega = 0.6$.

4. Experimental Results

4.1. Benchmarks

In order to study the performance of the proposed three alternative PSO dynamics, we have considered five benchmark functions listed in Table 1. The performance of the three proposed dynamics for these five functions is compared with that of classical PSO.

4.2. Parametric Range and Error Criterion

Early methods of performance evaluations for evolutionary algorithms were restricted to symmetric initializations. In recent time, researchers prefer asymmetric initialization. We here used the asymmetric initialization method to evaluate the performance of proposed three dynamics along with the classical PSO. In Table 4, we provide the initialization range of the objective function variables, the position of theoretical optima and the error criterion used to terminate the algorithm.

Table 4. Parametric range of benchmark functions

Function	Dimension	Initialization Range	Theoretical Optima	Error Criterion
*f*1	30	[50, 100]	[0,0……,0]	0.01
*f*2	30	[15, 30]	[1,1……,1]	0.001
*f*3	30	[2.56, 5.12]	[0,0……,0]	0.1
*f*4	30	[15, 32]	[0,0……,0]	0.01
*f*5	30	[300, 600]	[0,0……,0]	0.001

4.3. Simulation Strategies

Parameter selection of the PSO dynamics also is a crucial issue for speed-up and accuracy of the PSO algorithm. For a given benchmark function, we initially took wider range of the PSO dynamics parameters: $\alpha^g(t)$ in [0, 4], $\alpha^l(t)$ in [0, 2] and ω <1. Several hundred runs of the PSO programs with random parameter settings in the above ranges confirm that for a specific function, the best choice of parameters are

Table 5. Range of optimal values of $\alpha^g(t)$, $\alpha^l(t)$ and ω for LyPSO, PPSO and SPSO

Function	Parameters					
	$\alpha^g(t)$, $\alpha^l(t)$ and ω of LyPSO			$\alpha^g(t)$, $\alpha^l(t)$ and ω of PPSO/SPSO		
	$\alpha^g(t)$	$\alpha^l(t)$	ω	$\alpha^g(t)$	$\alpha^l(t)$	ω
*f*1	0.999 -1.999	0.0001 - 0.001	0.5-0.7	.199 -1.9	0.0001 -0.01	0.4 - 0.7
*f*2	0.199-0.399	0.001 - 0.003	10^{-12} -10^{-11}	.001 - .999	0.001 - 0.009	0.3-0.6
*f*3	0.299-0.599	0.0001 - 0.0005	0.0001 - 0.0009	.299 - .599	0.0001 - 0.0005	0.3-0.6
*f*4	0.1-0.2	0.7-0.8	0.001	Random	Random	0.3-0.7
*f*5	Random	Random	56-60	Random	Random	0.3-0.7

restrictive as indicated in Table 5. Moreover, the size of the population is taken as 40 and a maximum of 5000 iterations were taken for 30-dimensional particles.

4.4. Experimental Results from the Simulations

The relative comparison of the convergence time of the three algorithms with respect to classical PSO in case of Rastrigin function and Griewank function are given in Figure 2 and Figure 3. Rest of the figures cannot be included because of space limitations.

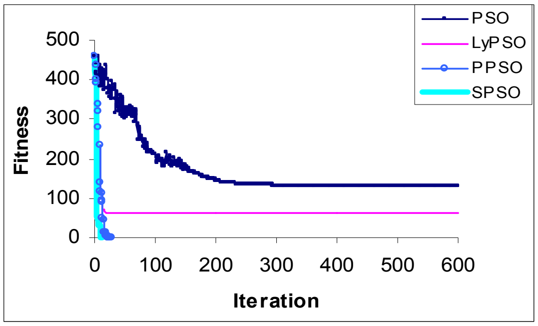

Figure 2. Progress towards optima: Rastrigin function

Table 6 provides the mean error and standard deviation for the globally best particle obtained by execution of the PPSO, SPSO, LyPSO and classical PSO over five benchmark functions. The error was obtained by taking the Euclidean distance between the theoretical optima and the position of the best-fit for a given program run.

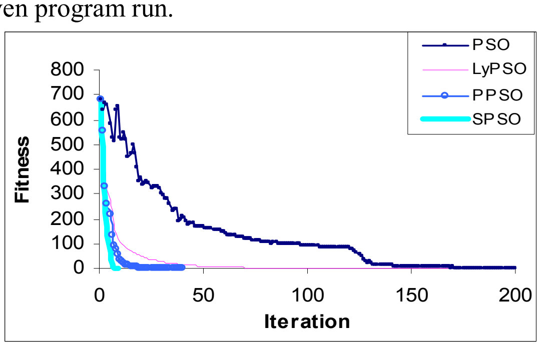

Figure 3. Progress towards optima: Griewank function

The mean error designates the average of errors over 50 independent runs. In order to make the comparison fair enough, runs of all the algorithms were let start from the same initial population. It is clear from Table 6 that for mean error for the SPSO algorithm is comparable but less than that obtained by PPSO algorithm, and the mean error obtained by the PPSO algorithm is insignificantly less than that of LyPSO algorithm. Further, the mean error obtained by the LyPSO algorithm is less in comparison to that of the classical PSO algorithm. This confirms that the SPSO algorithm outperforms the PPSO and LyPSO and definitely the classical PSO algorithm from the point of view of accuracy in solution.

Table 6. Mean Error and Standard Deviation over the Benchmarks

Function	Mean Error ±Standard Deviation			
	Classical PSO	LyPSO	PPSO	SPSO
*f*1	2.04e+00 ± 1.08e+00	4.3e−03 ± 7.94e−04	1.32e−02 ± 4.00e−03	4.3e−03 ± 7.94e−04
*f*2	7.77e+00 ± 0.77e+00	1.58e+00 ± 2.07e−01	9.99e−01 ± 9.4e−04	9.94e−01 ± 2.7e−03
*f*3	2.97e+00 ± 1.9e-01	1.48e+00 ± 8.06e−02	2.70e−03 ± 8.27e−04	2.79e−04 ± 6.82e−05
*f*4	7.03e+00 ± 6.22e+00	2.76e+00 ± 3.85e−01	1.70e−03 ± 6.20e−04	1.74e−04 ± 4.95e−05
*f*5	1.73e+00 ± 1.13e+00	9.93e−01 ± 3.00e−03	5.17e−02 ± 1.81e−02	1.58e−02 ± 3.50e−03

For each algorithm the average of the convergence time for 50 independent runs to meet the error limit for five benchmark functions is recorded in Table 7. It is clear from this Table that the mean convergence time of SPSO is less than that of PPSO. The mean convergence time of PPSO is less than that of LyPSO, and the mean convergence time of the latter is less than the mean convergence time of classical PSO. The above phenomena is true for all benchmark functions except the sphere, where the LyPSO and SPSO gives identical results because of same functional form in the SPSO and LyPSO dynamics.

Table 7: Mean Convergence Time of the Benchmarks over 30-dimensions

Function	Convergence Time (In seconds)			
	Classical PSO	LyPSO	PPSO	SPSO
*f*1	24.0	4.8	6.0	4.8
*f*2	21.1	11.5	7.2	5.9
*f*3	563.5	23.0	26.0	9.0
*f*4	148.9	67.2	23.2	10.9
*f*5	172.1	60.5	11.5	8.0

5. Conclusion

The paper proposed three alternative approaches to improve the speed of convergence of the PSO dynamics. The first approach attempted to replace the inertial term in the dynamics by a factor that ensures asymptotic stability of the PSO dynamics. The second alternative was to add the weighted negative position of the particle to the velocity update formula. The third approach was to replace the inertial term by the negative position of the particle itself. A random factor is attached to this term to maintain explorative power of the PSO dynamics to avoid its premature convergence. Computer simulations undertaken ensure that the third attempt results in significant improvement in convergence time and accuracy compared to the results obtained by the first and second attempt. We have also done unpaired *t*-tests between the best and second best algorithms in each case (standard error of difference of the two means, 95% confidence interval of this difference, the *t* value, and the two-tailed *P* value), which cannot be provided here because of space limitations. The t-tests also show that one or more of the proposed PSO methods can always beat the classical PSO in a statistically significant way.

6. References

[1] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, vol. 1, Mar. 1995, pp. 39-43.

[2] F. van den Bargh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evo. Comp.*, vol. 8, no. 3, June 2004.

[3] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evolutionary Computation (CEC' 98)*, Anchorage, AK, May 1998.

[4] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. Evol. Computation,* vol.6, no.2, pp. 58-73, Feb. 2002.

[5] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congress on Evolutionary Computing (CEC 2000)*, pp. 84-89, 2000.

[6] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proc. IEEE Congress on Evolutionary Computing*, Anchorage, pp. 84-89, May 1998.

[7] P. N. Suganathan, "Particle swarm optimizer with neighborhood operator," in *Proc. IEEE Congress on Evolutionary Computing*, Piscataway, NJ, pp. 1958-1962, 1999.

[8] R. C. Eberhart and X. Hu, "Adaptive particle swarm optimization: detection and response to dynamic systems," in *Proc. IEEE Congress on Evolutionary Computing*, pp. 1666-1670, May 2002.

[9] K. Parsopoulos and M. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 211-224, Jun. 2004.

[10] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. on Evo. Compu.*, vol. 8, no. 3, pp. 240-255, Jun. 2004.

[11]M. LØvbberg, T. Rasmussen and T. Krink, "Hybrid Particle Swarm Optimizer with breeding and subpopulations", *Proceedings of the third Genetic and Evolutionary Computation Conference (GECCO)*, vol. 1, pp 469-476, 2001.

[12] V. Miranda and N. Fonseca, "New evolutionary particle swarm algorithm (epso) applied to voltage/VAR control", *The 14th Power Systems Computation Conference (PSCC'02)*, Seville, Spain, June, 2002.

[13] M. LØvbberg and T. Krink, "Extending particle swarms with self-organized criticality", *Proceedings of the Fourth Congress on Evolutionary Computation (CEC-2002)*.

[14] T. Blackwell, and P.J. Bentley, "Don't push me! Collision-avoiding swarms", *IEEE Congress on Evolutionary Computation*, Honolulu, Hawaii USA, 2002.

[15] T. Krink, J. S. VesterstrØm and J. Riget, "Particle swarm optimization with spatial particle explosion", *Proceedings of the Fourth Congress on Evolutionary Computation (CEC-2002).*

[16] X. Xie, W. Zhang, and Z. Yang, "A dissipative particle swarm optimization", *IEEE Congress on Evolutionary Computation, Honolulu, Hawaii,* USA, 2002.

[17] B. C. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[18] R.E. Kalman, and J.E. Bertram, "*Control Systems Analysis and Design Via the Second Method of Liapunov: II, Discrete Time Systems,*" Trans. ASME, J. Basic Eng., Series D, No. 3, June 1960,pp. 371-400.

[19] W. Hahn, *Theory and Application of Liapunov's Direct Method*, Prentice-Hall, Englewood Cliffs, N.J, 1963.

[20] K. Ogata, *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1990.