[S1]International Conference On DESIGN AND MANUFACTURING, IConDM 2013

Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis

Nizar Rokbani*, Adel.M Alimi

REGIM-lab, REsearch Group on Intelligent Machines (LR11ES48), National Engineering School of Sfax, University of Sfax , Tunisia.

Abstract

This paper investigates the relative performances of PSO variants when used to solve inverse kinematics. Inverse kinematics is a key issue in robotics; for problems such as path planning, motion generation or trajectories optimization, they are classically involved. In the specific case of articulated robotics, inverse kinematics is needed to generate the joint motions, correspondent to a known target position. Articulated systems are very important in humanoid robotics, since arms and legs belong typically to this kind of mechanisms. In this paper the IK-PSO, Inverse Kinematics PSO, is applied to a double link articulated system. A statistical analysis is conducted to survey the convergence and relative performances of the main PSO variants when applied to solve IK; the PO variants tested are: Inertia weight PSO, Constriction factor PSO, linear decreasing weight and two simplified PSO variants.

1. Introduction

In robotics, modeling consists in writing the forward kinematics, the inverse kinematics and the dynamic model. Forward kinematics gives the system position according to a reference frame as a result of the joints motions. The inverse kinematics gives the joints motions that are needed to achieve a specific position in the reference frame;

---

* Corresponding author. Tel.: +0-216-50-660-056; fax: +0-216-73-382-658.
   E-mail address: nizar.rokbani@ieee.org

this can be assimilated to a geometry problem and solved using geometry and trigonometry paradigms [1]. Matrix based formulation of the kinematics model offer also a possible solution to inverse kinematics, when the model is invertible, and if not, some analytical solutions are used, such as the pseudo inverse, which gives an approximate solution of the problem [2-3].

Intelligent techniques such as Evolutionary algorithms, neural networks, were investigated to solve the inverse kinematics in robotics [4-5]. To use techniques such as PSO, ANT or Bee colony optimization algorithms, the inverse kinematics problem is transformed into and optimization problem using a forward kinematics and a fitness function, the fitness function has to measure the quality of the rendered solutions [4-5-6]. In the case of humanoid robotics, these techniques are called intelligent gait generation methods, and could lead to machine prototyping and control [7].

This paper is organized as follows; the next paragraph gives a brief overview on particle swarm optimization with a focus on the variants that will be used for the comparative tests. Paragraph 3, reviews the IK-PSO algorithm. Paragraph 4, reports the experimental test bench and the obtained results. The paper is ended by paragraph 5, which is dedicated to the conclusion.

Nomenclature

PSO	Particle Swarm Optimization
x_i	Position of the particle at iteration (i)
v_i	Velocity of the particle
w	Inertia weight factor
c_1	Cognitive parameter
c_2	Social parameter
p_{gbest}	Best global position
p_{lbest}	Best local position
X_t	Target point position
$Q(), q()$	Angular position vector
Θ_1	Angular position of link 1 in radian
Θ_2	Angular position of link 2 in radian
l_1	Length of link (1) expressed in meters
l_2	Length of link (2) expressed in meters
f_i	Fitness function at iteration (i)

---

2. Particle Swarm Optimization

2.1. The PSO Heuristic

Bird flocks or fish schools could be considered as Animals society's or Animals social groups, with evidence of intelligent group behaviors', which are based on a limited set of individual skills and interactions. The group has mechanisms enabling it to optimize its global behavior without any central managing or centralized decision. This underline the capacities of a self-organized intelligence based on simple rules and interactions.

Observing bird flock flights, a basic formulation describing the complex comportment of the group could be simplified as follows: "Every individual must update his position, taken into consideration his neighbors' and the overall dynamic of the group ». To use that as meta-heuristic, the groups of animals are replaced by vectors; the environment is assimilated to the search space and the objective and it effectiveness is evaluated by a function, called fitness function. The particle swarm optimization, PSO, is an effective technique for nonlinear systems with continuous variables or mixed. The first mathematical formulation, and the most simple was proposed in Kennedy et al, in [8], since that several variants have emerged [9]. In its basic version, PSO algorithm begins by defining a number of group individuals, called particles, the search space and an optimality criterion. The particles are

randomly distributed over the search space, their fitness functions are initialized, given this situation, the best particle of the swarm and the best local particle, are defined according to a neighborhood policy. Each particle is then moved toward a new position taken into consideration its current position and the bests, local and global, positions. The amount of the displacement is called velocity, see equations (1) and (2).

$$v_i = v_i + c_1.rand().(p_{lbest} - x_i) + c_2.rand().(p_{gbest} - x_i)$$

(1)

$$x_i = x_i + v_i$$

(2)

The particles are displaced in the search space as the "goal" is not satisfied, or the maximum number of iterations is not reached. The stop condition ensures the convergence of the algorithm but does not ensure the optimality of the solution. In the next paragraphs, some variants of PSO are introduced, the variants listed here are limited to those used in this paper.

2.2. PSO Variants

The development of PSO, leads to the raising of several variants [9-10-11]. Inertia weight PSO, is a modification introduced by Shi and Eberhart [9], it is a variant in witch a moderation weight was added to control the velocity, this lead to new equation of the velocity while the position equation remains unchanged, as in equations (3) and (4). Initially, the proposal was about a constant factor, about (0.7), applied to all iterations and optimization process. A large inertia weight is used to allow fast and global optimum search strategies while a reduced one is used to focus on local optimums.

$$v_i = w.v_i + c_1.rand()*(p_{lbest} - x_i) + c_2.rand().(p_{glbest} - x_i)$$

(3)

$$x_i = x_i + v_i$$

(4)

The inertia weight is then used to moderate and adapt the velocity while the PSO is running; it allowed working with different velocities ranges between the early search and final steps. By the end of processing, the velocity is reduced allowing a more precise displacement of the particles, (w) ranges from 0.9 to 0.2, note that negative values are also used.

Several inertia weight were proposed including random inertia weight, in which (w) is composed by a fixed amount, 0.5, and a random generated number with an equivalent moderation, see equation (5).

$$w = 0.5 + \frac{rand()}{2}$$

(5)

The linear decreasing inertia weight reduces the amount of the inertia iteratively as the PSO is close to its maximum of iteration number. The PSO start processing with a maximum inertia value *(Wmax)* that is decreased linearly according to equation (6).

$$w_k = w_{max} - \frac{w_{max} - w_{min}}{N} k$$

(6)

Where *(N)*, is the maximum iteration number of the PSO, maximum and minimum inertia are fixed, the respective values of 0.9 and 0.4 are commonly used, and *(k)* is the current iteration counter [13].

The constriction-factor variant is also a variant close to the inertia weight formulation, since the whole equation of velocity is subject to a factorization. The new factor, (K), stands for the constriction factor and is defined as in (7). Clerk's parameters were K= 0.729, and the convergence is insured if $\varphi > 4$ [10-11].

$$v_i = K * \left[\ v_i + c_1 * rand() * (p_{lbest} - x_i) + c_2 * rand() * (p_{gbest} - x_i) \ \right]$$

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}$$

$$\varphi = c_1 + c_2 \tag{7}$$

A constriction factor variant is assumed to be a specific case of inertia weight PSO [10].

2.3. Simplified PSO Variants

For Simplified PSO variants a focus is made on the socially only PSO, this simplified PSO could reduce computational time and convergence for problem with a single global optimum [12]. PSO-VG, is a variant in which the particles velocity depends on the best global solution only, this means that cognitive contribution is completely ignored and the velocity equation is transformed into equation (8).

$$v_i = w * v_i + c_2 * rand() * (p_{gbest} - x_i) \tag{8}$$

In equation (8), the cognitive aspect of the displacement is completely ignored, only the global best is used to moderate the individual comportment, this is a typical case of PSO interpretation [13-14].

A very close proposal to PSO-VG, is the PSO-G, where particles positions are updated directly using the global solution only, see equation (9). Here the velocity, is simply directly plugged into the position vector, once more the local best, representing the cognitive comportment is reduce, particles update their position according to the best solution.

$$x_{i+1} = x_i + c * rand()(p_{gbest} - x_i) \tag{9}$$

Such a variants could viewed as a PSO with a cognitive parameter ($c1 = 0$), these variants are faster than the classical PSO since there is reduction in the algorithm, the processing needed to evaluate the best local is simply omitted.

3. Solving inverse Kinematics Using PSO

3.1. IK-PSO algorithm

Assuming an articulated robotic system, and supposing that we need to move it from its initial position *Xo* to a new target position *Xt;* we need a set of joints rotations allowing the system to achieve the target, this is a typical inverse kinematics problem. IK-PSO, Inverse Kinematics PSO, returns a possible solution of a legged robot inverse kinematics using a PSO and a Forward kinematics model [5].

The PSO particle is coded as: *Qi= (θ1, θ2,..., θn)*;   the IK-PSO algorithm iterates in order to generate a potential solution. For any particle, and using the forward kinematic model, the robot position in a reference frame is obtained as: *Xi= (X1, X2,..., Xn)i.* This position is compared with the target position, *Xt*, if it satisfies the fitness function and the constraints, *Qi* is returned as a solution. A typical fitness function is given in equation (10), it is

homogenous with an Euclidian distance of the target position to an end-link position $X(j)$ evaluated by iteration (i), in our case (j=2).

$$f_i = \left\| Xt - X(j)_i \right\| \tag{10}$$

The fitness function is satisfied if it is less than a fixed amount (ε); if not the PSO, returns the best global position by the end of its processing. IK-PSO, procedure is resumed in figure 1; its stop criterion depends on the fitness function or the achievement of the maximum number of iterations. In fig 1, the stop criteria is noted (C), IK-PSO stops if the maximum number of iteration is attended or if the error toward to target position is less than a fixed amount.
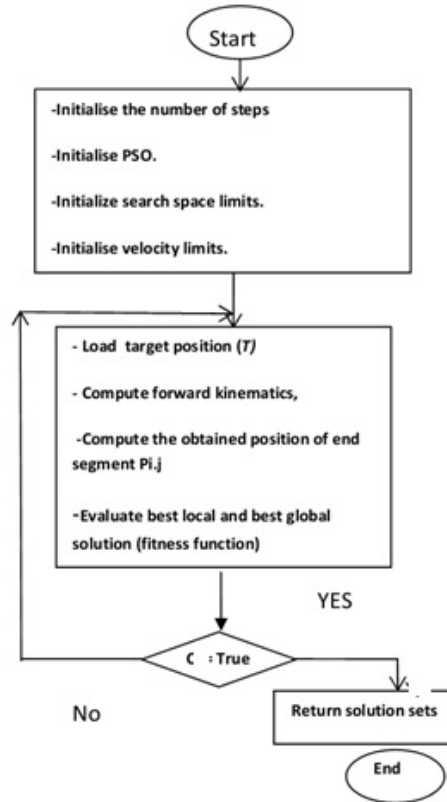

Fig1. IK-PSO, The Pseudo code of the PSO Inverse Kinematics Solver, the fitness function is a distance toward the target position, the stop condition (C) is fitness less than (ε) or maximum iteration reached.

For an articulated system, where joints are only subject to rotations, the PSO, position and velocity formulation could be re-arranged to fit better the particle representation, which is here a join rotation, see equation (11) and (12).

$$\vec{V}_{i+1} = w * \vec{V}_i + \vec{\varphi}_1 * (\vec{q}_{lbest(i)} - \vec{q}_i) + \vec{\varphi}_2 * (\vec{q}_{best} - \vec{q}_i)$$

(11)

$$\vec{q}_{i+1} = \vec{q}_i + \vec{V}_{i+1}$$

(12)

Equations (11) and (12) are similar to inertia weight PSO, developed previously, while here, the particle performs a rotation instead of a classical 2D displacement. To smooth the obtained gaits, we can add, a set of joint-rotations constraints, geometric constraints, human-biomechanical inspired as well as pure robotic constraints coming from the mechanical design or the working space configuration [5].

In this paper, equations (11) and (12) were replaced by the PSO variants listed in the previous paragraph. The obtained results were compared in term of precision and time needed to converge, time is evaluated based on the average of the convergence iteration; Precision will be appreciated based the error distance between the target point and the obtained solution.

3.2. Problem formulation for 2 links articulated system.

Assume an articulated system composed by two links (1 and 2) and two revolute joints, as in figure2. The first revolute joint is placed at the first extremity of link 1and the reference frame; the second revolute joint is placed between the links; the extremity of link 2 is free, it is supposed to be the active end-link that should attend the target position, see figure 2. The first joint of the system performs a rotation toward the *(X)* axe of the frame of *(Θ1)*. The joint between link1 and link 2, performs also a rotation *(Θ2)*, allowing to this link to have a displacement relative to the first one. Such a system could typically represent an arm or a leg; it is widely used in robotics.

The inverse kinematics problem formulation could be said as follows: Given a know position, *Xt,* that the extremity of link2 has to achieve, find the needed angular position vector *Q(),* corresponding to it with respect to a set of joint motions constraints. For the system that appears in figure 2, the problem consists in finding a couple of rotations $Q = (\theta_1, \theta_2)$ that leads the system to the target position $X_t = (y_t, z_t)$. In this case $X_1 = (y_1, z_1)$ and $X_2 = (y_2, z_2)$, represent respectively the positions of the extremities of links 1 and 2 as in figure 2.

Known the forward kinematics model of the system, the inverse kinematics problem can be expressed as follows:

find a set of *Q()* ,

$$Q = [(\theta_1, \theta_2)]$$
satisfying :
$$X_2 = (y_2, z_2) = X_t$$
under
$$\theta_1 \in [\theta_{1min}, \theta_{1max}]$$
$$\theta_2 \in [\theta_{2min}, \theta_{2max}]$$

(13)

The forward kinematics model of the system, as it appears in figure 2, could be expressed by a set of equations as in (14).

$$\begin{cases} y_1 = l_1 \cos(\theta_1) \\ z_1 = l_2 \sin(\theta_1) \\ y_2 = z_1 + l_2 \cos(\theta_1 + \theta_2) \\ z_2 = y_1 + l_2 \sin(\theta_1 + \theta_2) \end{cases}$$

$$(14)$$

Where *(l1)* is the length of link1 and *(l2)* is the length of link2. These equations represent the forward kinematics model of the system, note that the rotation of the link2 is expressed using link1 as reference, see fig2.
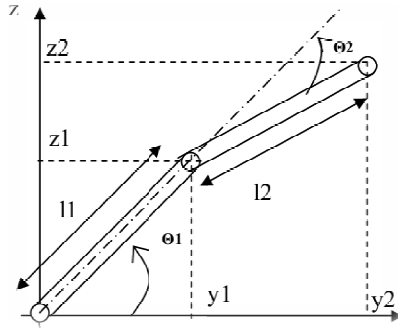

Fig.2. A double link articulated system composed of links with respective lengths l1 and l2

4. Experimental Results

4.1. Experimental Protocol

Experimentations are conducted using the PSO variants, detailed in paragraph 2, and a normalized articulated system. Normalized system is obtained by expressing the segments position using their normalized length according to the total length of the articulated limb (a leg or an arm). This means that the equations (14) are modified as follows:

$$\begin{cases} y_1 = \frac{l1}{l_1 + l_2} * \cos(\theta_1) \\ z_1 = \frac{l1}{l_1 + l_2} * \sin(\theta_1) \\ y_2 = z_1 + \frac{l2}{l_1 + l_2} \cos(\theta_1 + \theta_2) \\ z_2 = y_1 + \frac{l2}{l_1 + l_2} \sin(\theta_1 + \theta_2) \end{cases}$$

$$(15)$$

Figure 3, shows a typical the IK-PSO solution search process for a given target position *Xt (0.4, -0.8)*, The best solution find by the algorithm is tagged in red, in this case the screenshots concern the results ongoing by iteration 100, the best solution find is tagged in red. In figure 3, we can see the articulated system positions correspondent to several particles attempts, normally only the best position is returned by the IK-PSO algorithm, here the processing

is showed as well as the best obtained position obtained by iteration 100. The IK-PSO is intentionally stopped at iteration 100 for all variants to produce the figure 3.
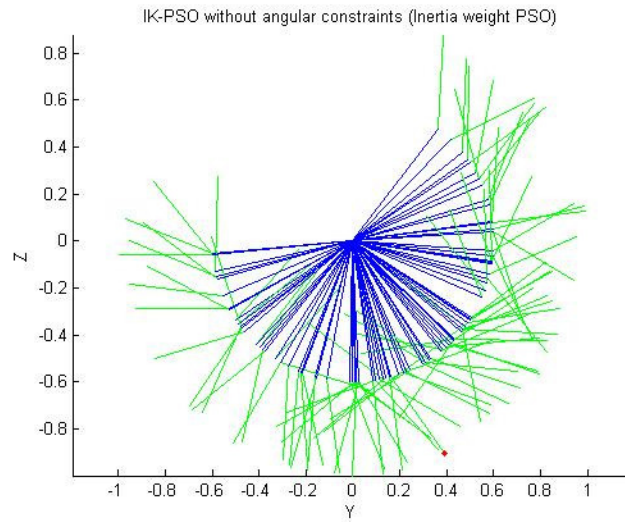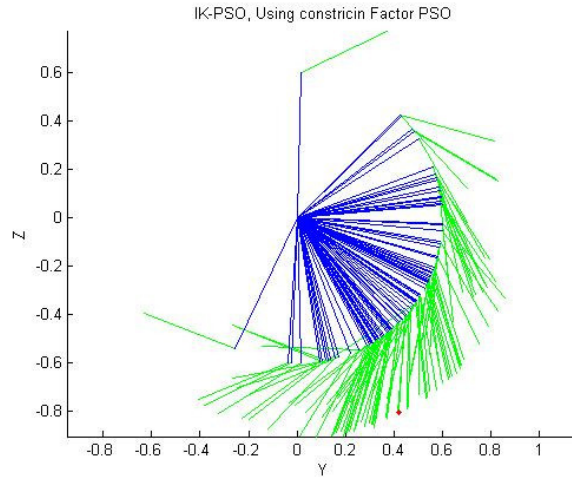

(a)   Inertia weight PSO, solution search process and best solution at iteration 100, the best solution is tagged in red.
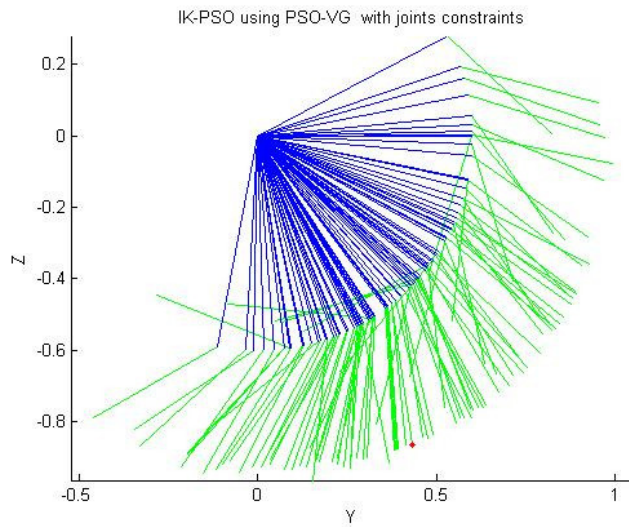
(b)   PSO-VG, solution search process and best solution at iteration 100, the best solution is tagged in red.

(c) Constriction factor PSO, solution search process and best solution at iteration 100, the best solution is tagged in red.

Fig.3. IK-PSO, Typical application of IK PSO on a double link articulated system; here the best solution toward the target point appears in red, processing is stopped at iteration 100 for all variants.

4.2. Results

For the comparative test, a target point is fixed at the position : $Xt = (0.4, -0.8)$, the segments lengths are respectively (0.6 m for link 1 and 0.4 m for link 2), the maximum number of iteration are fixed to 5000 and the fitness function is supposed satisfied if it is less than (ε=0.0001).

For each PSO variant the test is repeated 100 times and the average error is computed as well as the average convergence iteration, note that the number of particles is (15).

Table.1. PSO fast variants for inverse using IK-PSO,

PSO variants	Parameters	Average fitness	Average stop iteration
PSO, Eq (1)	(c1 = 1.4047, c2 =1.494)	4.119189e-003	4887 (4886.997)
IWPSO, Eq (3)	(w= 0.729, c1 =c2 = 1.494)	2.119189e-005	3380 (3380,789)
K-PSO, Eq (7)	K=0.729	4.119189e-002	3705 (3705,562)
IW-PSO, Eq (5)	(w =x*, c1 = 2, c2 =2)	6.654755e-005	4024 (4024.346)
IW-PSO, Eq (6)	(w =x *, c1 = 2, c2 =2)	4.199458e-002	5000
PSO-VG, Eq (8)	(w =0.8, c2 = 2)	4.186197e-005	740 (740.422)
PSO-VG, Eq (8)	(w =0.3600, c2 = 3.800)	3.145837e-005	743 (742.850)
PSO-G, Eq (9)	(c = 2)	7.746535e-005	1445 (1445.621)
PSO-G, Eq (9)	(c= 1.494)	5.556231e-005	2101 (210.931)

5. Conclusion

What comes first, from this limited test bench, is that The IK-PSO algorithm showed evidence at overcoming the inverse kinematics problem with no need to compute the inverse model. The inverse model is obtained using

matrix computing, algebraic, iterative or geometric resolutions, assumed to be complex. All tested variants showed evidence of convergence, using their respective classical parameters. Some variants showed high speed convergence.

The fast variant PSO-VG showed evidence of fast convergence since the average of convergence iteration for a set of 100 runs is 740. PSO-G average convergence iteration is about 1445 for 100 tests with the same configuration of PSO-VG. From our experimental results PSO-VG can be assumed to be faster than classical inertia weight PSO. PSO-G, convergence is observed around an iteration average about (1445), wile the average convergence iteration of inertia weight PSO (w= 0.729, c1 =, c2 = 1.494), is around (3380), see table 1 for more details.

The current statistical analysis is limited to one hundred try for each variant of PSO, the results are promising while larger statistical analyzes are needed to confirm these results. Only 7 set of parameters covering the main PSO variants were tested, the comparative results are achieved on the basis of 2 Degree of freedom articulated system. The impact of the complexity of the kinematic structure, as well as the PSO variants and parameters are under investigation.

Acknowledgment

References

[1] Lee, CS George, and Ziegler, M., 1984, "Geometric approach in solving inverse kinematics of PUMA robots.", IEEE Transactions on Aerospace and Electronic Systems, Vol 6, 1984, pp. 695-706. IEEE.

[2] Gaurav, T., and Schaal, S., 2000, "Inverse kinematics for humanoid robots.", Robotics and Automation, 2000. Proceedings of ICRA'00. IEEE International Conference on Robotics and Automation. Vol. 1. IEEE.

[3] Lee, K-M., and Dharman K. Shah., 1988, "Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator.", IEEE Journal of Robotics and Automation, vol 4.3, 1998, pp.354-360, IEEE.

[4] Wang, X., Ming-Lin, H., and Yu-Hu, C., 2008, "On the use of differential evolution for forward kinematics of parallel manipulators.", Applied mathematics and computation 205.2, pp. 760-769.

[5] Rokbani, N. and Alimi, Adel M., 2012, "IK-PSO, PSO Inverse Kinematics Solver with Application to Biped Gait Generation.", International Journal of Computer Applications, vol 58(22), pp.33-39, November 2012, Published by Foundation of Computer Science, New York, USA.

[6] Rokbani, N., E. Benbousaada, B. Ammar, and Alimi, Adel M., 2010, "Biped robot control using particle swarm optimization.", Proceedings of the IEEE Conference on, Systems Man and Cybernetics (SMC), 2010, pp.506-512. IEEE.

[7] Rokbani, N., Zaidi, A. and Alimi, Adel M., 2012, "Prototyping a Biped Robot Using an Educational Robotics Kit.", Proceedings of the International Conference on Education and E-Learning Innovations, IEEE.

[8] Eberhart, Russell, and James Kennedy., 1995, "A new optimizer using particle swarm theory.", Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE.

[9] Shi, Yuhui, and Russell C. Eberhart., 1999, "Empirical study of particle swarm optimization.", Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99.. Vol. 3. IEEE.

[10] Eberhart, R. C., & Shi, Y., 2000, "Comparing inertia weights and constriction factors in particle swarm optimization.", Proceedings of the 2000 Congress on In Evolutionary Computation, Vol. 1, pp. 84-88, 2000.IEEE

[11] Shi, Y., 2001, "Particle swarm optimization: developments, applications and resources.", Proceedings of the 2001 Congress on Evolutionary Computation, Vol. 1. IEEE.

[12] Pedersen, M.E.H., and Chipperfield, A.J.,2010, "Simplifying particle swarm optimization. In Applied Soft Computing", volume 10, Issue 2, March 2010, pp. 618-628, Elsevier.

[13] Vazquez, J. C., Valdez, F., and Melin, P., 2013, "Comparative Study of Particle Swarm Optimization Variants in Complex Mathematics Functions.", Recent Advances on Hybrid Intelligent Systems (pp. 223-235). Springer Berlin Heidelberg.

[14] Banks, A., Vincent, J., and Anyakoha, C., 2007, " A review of particle swarm optimization. Part I: background and development.", Natural Computing,Vol 6(4), pp. 467-484, Springuer.