

A Method Football Team Model Optimization and Application of the Optimization Control

Nguyen Hoang Mai
Electrical Engineering
Danang University of the Science and Technology
Danang, Vietnam
nhmai@dut.udn.vn

Tran Van Dung
Electrical Engineering
Daklak Technical College
Daklak, Vietnam
tranvandungcdn@gmail.com

Abstract—The development of the AI, IoT, and Big Data have to become strongly apply to discrete event strings systems. That are modern developments of the world. Therefore, we have to have an advanced method to develop adaptive applications, especially with MIMO discrete event systems. There is a limit while using a continuous calculation to control systems because the big calculation is an obstacle. So we have to find an optimization method to reduce the number of parameters in the calculation at any time. We could do it by choice the main parameters and except auxiliary parameters. In this paper, we introduce a Football Team Optimization (FTO) method, which is a new method to do optimization problem while control with many parameters system. The application and analysis to compare any method as PSO, traditional PID, which takes out the difference of this algorithm.

Keywords—Football team model, traditional PID, discrete event system, robot team, self-organize.

I. INTRODUCTION

We have to meet many applications in the practice to control with a selection of the parameters for one target. In the MIMO systems, as swarm optimization of robot team, choice sensors system in the machines, control planes go up and go down to runway and etc... we have one target for control variables. We call that is a Football Team Model, which is advanced control by self-organized. We could describe as figure 1.



Figure 1. The football team model on the stadium.

While the team is in the stadium, members of the team shall run to the ball and touch it into the goal of the

opposition.

So only member contact to the ball at a time (we call this is the main member) and other members find good positions to wait for opportunities from the main member or guide leader. This gives the benefit of loss energy, motion time, save a total of the action time of the system.



Figure 2. Model of robotics team of military

In the technical military at figure 2, soldiers or robots will have to meet cases, which choice some objects to supervise with mapping point to point. That is time subject have to select objects for them. They need use optimization algorithm to take out final.

In the sensor systems, which use the sensor to measure process or parameter. We could use priority to main sensors, which change role by time. So we need to use optimal selection to the choice main sensor. Figure 3.

In the control systems with many parameters change by time, for example of adaptive control [1], [3],[4], we change the value of parameters to do adaptation into the process. We have priority to select forward and feedback signals of the MIMO systems [1],[2],[3],[6].

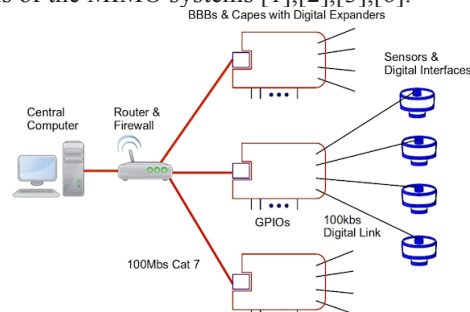


Figure 3. Model of sensors system.

Print ISSN 1691-5402

Online ISSN 2256-070X

<http://dx.doi.org/10.17770/etr2019vol2.4195>

© 2019 Nguyen Hoang Mai, Tran Van Dung.

Published by Rezekne Academy of Technologies.

This is an open access article under the Creative Commons Attribution 4.0 International License.

Many controllers always calibrate parameters, for example, hybrid PID controller in figure 4, the factors K_p , K_i , K_d change value to adopt state of the system. So if you have a matrix of parameters, you will have the difficult choice main factor to calibrate for accuracy is best. And also many other applications in the practice.

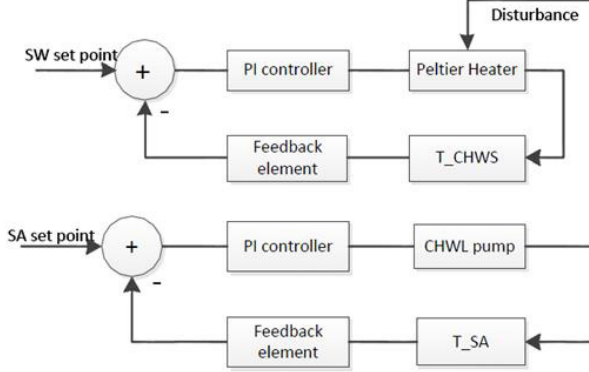


Figure 4. PID system

The main problem of FTO is a choice correction of the parameter. If you wrongly choice, you will spend much time to calculate values you need and the system can't respond well, it is very important for nonlinear systems. The parameters of nonlinear systems change chaos and maybe not a converging, so we can estimate value to decide choice any best parameter by gradient and acceleration of signal.

II. PROBLEMS

There is a system MIMO as figure 5. In that, $\mathbf{x} = [x_1, x_2, \dots, x_n]'$ is a state vector with assume continuous signal and exists high order of derivations, the \mathbf{x} includes input signals \mathbf{y}_0 , which is a subset of \mathbf{x} , $\mathbf{y} = [y_1, y_2, \dots, y_m]'$ is an output vector, $\mathbf{d} = [d_1, d_2, \dots, d_r]'$ is a disturbance vector. We call target function $\mathbf{f}(\mathbf{e})$, with is an error of the output signals y_i , $i=1..m$.

$$e_i = y_{i0} - y_i; i=1..m \quad (1)$$

With y_{i0} is a set point of output y_i .

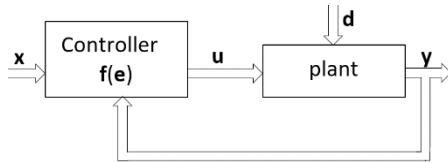


Figure 5. Model of the system.

The $\mathbf{f}(\mathbf{e})$ does the problem to reduce error to minimum values.

From that, we have:

$$\begin{aligned} \mathbf{x} &= [x_1, x_2, \dots, x_n]' \\ \mathbf{x}^{(1)} &= [\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n] \\ &\dots \\ \mathbf{x}^{(l)} &= [x_1^{(l)}, x_2^{(l)}, \dots, x_n^{(l)}] \end{aligned} \quad (2)$$

We define "distance of control" is a state variable x . That is a difference of set parameter value and output value. Because output signal depends on some input signals (MISO), so we can see that's crossing of the input signals

with output signals as figure 6 to calibrate parameters of controller [8],[9].

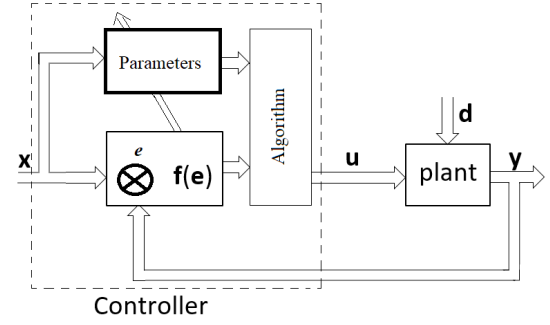


Figure 6. Model of the system with calibrates parameters.

In figure 6, the block $\mathbf{f}(\mathbf{e})$ decides vector and choice parameter area for calibration. After reduces area of searching, $\mathbf{f}(\mathbf{e})$ does any parameter to directly calibrate.

III. CONTROL ALGORITHM

We go back to our football team as figure 1. A system controls one target at a time with the choice main parameter. Start at $\mathbf{x} \in \mathbb{R}^{n \times 1}$, we choose the main parameter as follow with absolute values:

$$\tilde{x}_k = \text{Min}(x_1, x_2, \dots, x_n) \quad (3)$$

$$\dot{\tilde{x}}_i = \text{Max}(\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n) \quad (4)$$

$$\ddot{\tilde{x}}_j = \text{Max}(\ddot{x}_1, \ddot{x}_2, \dots, \ddot{x}_n) \quad (5)$$

...

We could choice higher order so it isn't well because have to difficultly calculate. After defining the main parameter, we have tree variables with sequence k, i, j . In general, $k \neq i \neq j$, so we continue to filter for one time to decide tree couples of variables

$(\dot{x}_k, \ddot{x}_k), (x_i, \dot{x}_i), (\dot{x}_j, x_j)$. If we choose a higher

order then we will have set as follow:

$$\begin{cases} \tilde{x}_k = \text{Min}(\mathbf{x}) \text{ with } k \in 1..n \\ \dot{\tilde{x}}_i = \text{Max}(\dot{\mathbf{x}}) \text{ with } i \in 1..n \\ \ddot{\tilde{x}}_j = \text{Max}(\ddot{\mathbf{x}}) \text{ with } j \in 1..n \\ \dots \\ \tilde{x}_q^{(p)} = \text{Max}(\mathbf{x}^{(p)}) \text{ with } q \in 1..n; \end{cases} \quad (6)$$

And set filter:

$$\begin{cases} (\dot{x}_k, \ddot{x}_k, \dots, x_k^{(p)}) \\ (x_i, \dot{x}_i, \dots, x_i^{(p)}) \\ (x_j, \dot{x}_j, \dots, x_j^{(p)}) \\ \dots \\ (x_q, \dot{x}_q, \dots, x_q^{(p-1)}) \end{cases} \quad (7)$$

We define a set of time variable:

$$t_k = \frac{x_k}{\dot{x}_k}, t_k^{(1)} = \frac{\dot{x}_k}{\ddot{x}_k}, \dots, t_k^{(p)} = \frac{x_k^{(p-1)}}{x_k^{(p)}} \quad (8)$$

$$t_i = \frac{x_i}{\dot{x}_i}, t_i^{(1)} = \frac{\dot{x}_i}{\ddot{x}_i}, \dots, t_i^{(p)} = \frac{x_i^{(p-1)}}{x_i^{(p)}} \quad (9)$$

$$t_j = \frac{x_j}{\dot{x}_j}, t_j^{(1)} = \frac{\dot{x}_j}{\ddot{x}_j}, \dots, t_j^{(p)} = \frac{x_j^{(p-1)}}{x_j^{(p)}} \quad (10)$$

$$\dots$$

$$t_q = \frac{x_q}{\dot{x}_q}, t_q^{(1)} = \frac{\dot{x}_q}{\ddot{x}_q}, \dots, t_q^{(p)} = \frac{x_q^{(p-1)}}{x_q^{(p)}} \quad (11)$$

After we have a set of times, we choice main variable with minimum time in the absolute domain:

$$t_l = \text{Min}(t_k, t_i, t_j, \dots, t_q) \quad (12)$$

In that, we have the main parameter is an x_r . If we have any time variables as same at the time then we continue choice:

$$t_l^{(1)} = \text{Min}(t_k^{(1)}, t_i^{(1)}, t_j^{(1)}, \dots, t_q^{(1)}) \quad (13)$$

We choice x_l with $t_l^{(1)}$. If we have any time variables as same then we continue select to the end of the set variable. After we calculate:

$$t_l = \text{Min} \left[\left(\prod_{d=0}^{p-1} t_k^{(d)} \right), \left(\prod_{d=0}^{p-1} t_i^{(d)} \right), \dots, \left(\prod_{d=0}^{p-1} t_q^{(d)} \right) \right] \quad (14)$$

And the main parameter is an x_r . So we have to choose the correct parameter to do priority control.

From that, we have a theory:

Theory: If there is a MIMO as (2), we can choice control variable with the law from (6) to (14) to ensure the control time is minimum.

Proof: we use loss function as follow:

$$f_k(e) = x_k - e_k = x_k - \dot{x}_k t_k = x_k - \ddot{x}_k t_k^{(1)} t_k =$$

$$x_k - x_k^{(3)} t_k^{(2)} t_k^{(1)} t_k = \dots = x_k - x_k^{(p-1)} \prod_{d=0}^{p-1} t_k^{(d)}$$

From there, we see that if $e_k \rightarrow \min$ then $\dot{x}_k t_k \rightarrow \min$. So \dot{x}_k is a speed to goes to zero position, we want speed bigger than more. Therefore, the time t_k will be minimum.

If we have many \dot{x}_k as same then we choose $\ddot{x}_k \rightarrow \max$, and etc. Finally, we choice time as (14) then insure error is minimum.

The calculation process uses a Min-Max function to compare and refine the main parameter with minimum time. The parameter x_l shall be optimal parameter because it has minimum time to go to the control variable.

Some system could use this method to do priority control as ANN, Gen Network, AI, PID-MIMO... It has a good benefit for multi-direction system control.

IV. APPLICATION

We apply about result into some examples to illustrate application.

The first, we use tree order SISO with transfer function:

$$G(s) = \frac{1}{s^3 + s^2 + s + 1} \quad (15)$$

Use PID controller, we find stabilization domain of K_p , K_i , and K_d and simulation result as figure 7.

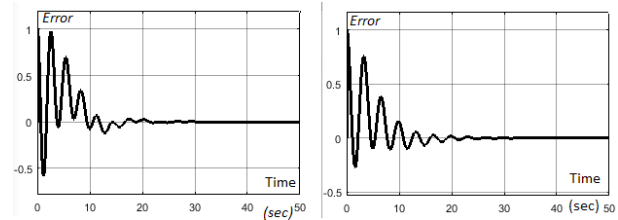


Figure 7. Simulation of the system (15)

From that, we can see the response of the system has stabilization and could control. However, the response is not “nice” for inter oscillation and we have to improve the controller.

Start $K_p = 1.5$; $K_i = 2.6$; $K_d = 0.5$, decided by Nichols – Ziegler method. Apply (6) and (12), we have to see K_p is an important factor at the start point. So we will measure error and calibrate K_p to reduce error. Use testing calculation, we choice K_p from zero time to established time. Next, we calibrate K_i for static error. We have resulted in figure 8.

By figure 8, we can see that the change of K_p has changed the response of the system. The transient time reduces to 15 seconds, so the number of oscillation cycle increase. Therefore, we could calibrate K_p to improve that.

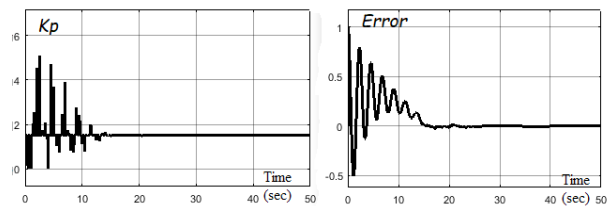


Figure 8. After calibrating K_p .

We see an example more for the problem, that is an application of FTO into MIMO system with robot 2DoF, which uses DC motor as figure 9.

For this model as ideal links, we calculate inertial torque and Coriolis force, we have a dynamic equation as follow:

$$T_1 = \left(\frac{1}{3} m_1 l_1^2 + m_2 l_2^2 + \frac{1}{3} m_2 l_2^2 + m_2 l_1 l_2 \cos \theta_2 \right) \ddot{\theta}_1 + \left(\frac{1}{3} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos \theta_2 \right) \ddot{\theta}_2 - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 - m_2 l_1 l_2 \dot{\theta}_2^2 \sin \theta_2 + \left(\frac{1}{2} m_1 + m_2 \right) g l_1 \cos \theta_1 + \frac{1}{2} m_2 g l_2 \cos(\theta_1 + \theta_2) \quad (16)$$

$$T_2 = \left(\frac{1}{3} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos \theta_2 \right) \ddot{\theta}_1 + \frac{1}{3} m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1^2 \sin \theta_2 + \frac{1}{2} m_2 g l_2 \cos(\theta_1 + \theta_2) \quad (17)$$

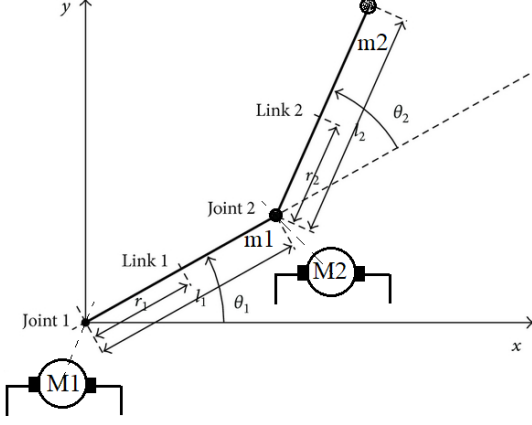


Figure 9. Model of robot 2DoF.

We use the PID controller with K_p , T_p , T_D change value by nonlinear properties of the system. The equations (15), (16) could write as a matrix:

$$\mathbf{T} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} g_1(\theta) \\ g_2(\theta) \end{bmatrix} = \mathbf{M}(\theta) \ddot{\theta} + \mathbf{N}(\theta, \dot{\theta}) \dot{\theta} + \mathbf{G}(\theta) \quad (18)$$

With:

$$m_{11} = \frac{1}{3} m_1 l_1^2 + m_2 l_2^2 + \frac{1}{3} m_2 l_2^2 + m_2 l_1 l_2 \cos \theta_2$$

$$m_{12} = \frac{1}{3} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos \theta_2$$

$$m_{21} = \frac{1}{3} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos \theta_2$$

$$m_{22} = \frac{1}{3} m_2 l_2^2 ; n_{11} = m_2 l_1 l_2 \dot{\theta}_2 \sin \theta_2$$

$$n_{12} = -m_2 l_1 l_2 \dot{\theta}_2 \sin \theta_2 ; n_{21} = m_2 l_1 l_2 \dot{\theta}_1 \sin \theta_2$$

$$n_{22} = 0 ;$$

$$g_1(\theta) = \left(\frac{1}{2} m_1 + m_2 \right) g l_1 \cos \theta_1 +$$

$$\frac{1}{2} m_2 g l_2 \cos(\theta_1 + \theta_2)$$

$$g_2(\theta) = \frac{1}{2} m_2 g l_2 \cos(\theta_1 + \theta_2)$$

Define the error of joint variables as follow:

$$\begin{cases} \theta_{e1} = \theta_{1sv} - \theta_1 \\ \theta_{e2} = \theta_{2sv} - \theta_2 \end{cases} \Rightarrow \boldsymbol{\theta}_e = \begin{bmatrix} \theta_{e1} \\ \theta_{e2} \end{bmatrix} \quad (19)$$

We will measure derivative of error to decide the main variable as (6) to (14). After the found error, we do a calibration of joint's parameters, and the process does continue...

We define "parameter's distance" is a difference of parameter in the online process. It depends on the measurement unit of the variable. From (17) we have:

$$\ddot{\boldsymbol{\theta}} = \mathbf{M}^{-1}(\theta) [\mathbf{T} - \mathbf{N}(\theta, \dot{\theta}) \dot{\theta} - \mathbf{G}(\theta)] = \ddot{\boldsymbol{\theta}}_{sv} - \ddot{\boldsymbol{\theta}}_e \quad (20)$$

V. SIMULATION RESULTS

We build a simulation model as figure 10. Use an observer of controller parameters from estimation values at start time, we continue measure parameters value and their derivative values. Use from (6) to (14) to calculate and choice priority parameter at the time.

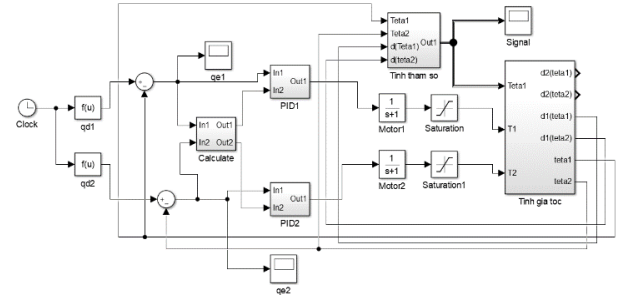


Figure 10. Model of robot 2 DoF to simulate.

We start action with estimation of PID parameters [10],[11]. Although robot manipulator is a nonlinear system, so we could use PID controller to control with strain domain. After design controller and do it, we receive an error of the trajectories of joint 1st is 15% and joint 2nd is 10% (average value). This is a big error in practice if use big trajectory more as figure 11.

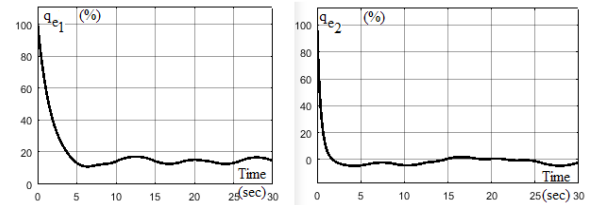


Figure 11. Simulation results with fix PID

Now we use advance algorithm above, from (6) to (14). We choice K_p is a starting strategy. We measure the error of trajectory and calculate time by derivation of the error. The first result is minimum time with K_p and next K_i . So we only use K_p to explain. The simulation results with new algorithm give better error than the traditional method of PID. We can see the error of the 1st joint is 2% and 2nd is 1% (average value) as figure 12. That is a big weight of the advanced algorithm in this paper.

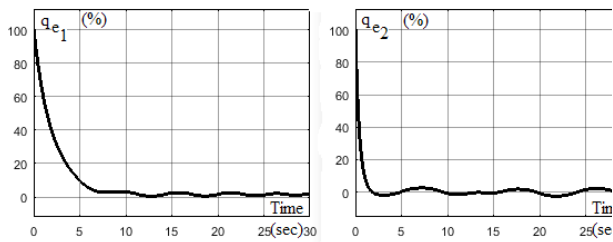


Figure 12. Simulation results with FTO

We want to do note an important issue to systems with self – calibration (adaptive regulators) controller. If choice true parameter then could do optimization time to find a balance point. In this case, we choice K_p factor by experiment because this is important of linear systems. The choice parameters can be affected to other parameters, so we need measure and change calibration between parameters.

We could take out many examples use FTO in practice as a team robot, team machine, distribution systems, etc...

VI. DISCUSSION

The theory and simulation results give a new acknowledge of control problems in the nonlinear MIMO system. There is the difference to PSO method [1],[2],[3],[4],[5] because there is a collection in PSO, so only there is an election in FTO.

The FTO method is best used in the distribution discrete system with discrete event string. In the some of analog event string, we also use FTO, so there are difficulty calculations will be apparent. Especially with un-focus parameters (chaos), we can't decide a time to finish process or accept the result with the big error.

The FTO will become a good application if we combine to AI or ANN to process discrete events in factories, medicine, military, game, transport control, and so on...

VII. CONCLUSION

In this paper, we have introduced a new method (FTO) to elect the main parameter to control in the nonlinear MIMO system. This method helps to reduce the calculation steps to find optimal elements of the process, especially in the self-organized system. When systems go outside your eye, then self-organize is an important very much, as robot team in space, distribution system with share optimization of parameters...

REFERENCE

- [1] W. Deng, H. M. Zhao, J.J. Liu, X.L. Yan, Y.Y. Li, L.F. Yin, C.H. Ding, "An improved CACO algorithm based on the adaptive method and multi-variant strategies", *Soft Computing*, vol. 19 no. 3, (2015), pp. 701- 713.
- [2] D. N. Wilke. *Analysis of the particle swarm optimization algorithm*, Master's Dissertation, University of Pretoria, 2005.
- [3] Natika W. Newton, *Understanding and Self-Organization*, *Front Syst Neurosci.* 2017, Mar 2. doi: 10.3389/ fnsys. 2017.00008
- [4] D. N. Wilke, S. Kok, and A. A. Groenwold, *Comparison of linear and classical velocity update rules in particle swarm optimization: notes on diversity*, *International Journal for Numerical Methods in Engineering*, Vol. 70, No. 8, pp. 962–984, 2007.
- [5] P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley, 2005.
- [6] Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of IEEE congress evolutionary computation*, San Diego, CA, pp 84–88 10.
- [7] Kennedy J (2006) *Swarm intelligence*. In: *Handbook of nature-inspired and innovative computing*. Springer, New York, pp 187–219 13.
- [8] Talbi EG (2009) *Metaheuristics: from design to implementation*. Wiley, UK 14. Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization. In: *The proceedings of evolutionary programming VII (EP98)*, pp 591–600 15.
- [9] W. Deng, R. Chen, B. He, Y.Q. Liu, L.F. Yin, J.H. Guo, "A novel two-stage hybrid swarm intelligence optimization algorithm and application", *Soft Computing*, vol. 16, no. 10, (2012), pp. 1707-1722
- [10] Pedersen MEH (2010) Good parameters for particle swarm optimization. Hvass Laboratories Technical Report HL1001
- [11] Bansal JC, Singh PK, Saraswat M, Verma A, Jadon SS, Abraham A (2011) Inertia weight strategies in particle swarm optimization. In: *Third world congress on nature and biologically inspired computing (NaBIC 2011)*, IEEE, Salamanca, Spain, pp 640–647
- [12] Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. *Acta Mech* 213(3–4):267–289 42 2 Particle Swarm Optimization
- [13] Qian X, Cao M, Su Z, Chen J (2012) A hybrid particle swarm optimization (PSO)-simplex algorithm for damage identification of delaminated beams. *Math Probl Eng*, Article ID 607418, p 11
- [14] Chen WN, Zhang J, Chung HSH, Zhong WL, Wu WG, Shi Y (2010) A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Trans Evol Comput* 14 (2):278–300
- [15] Gomes MH (2011) Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Syst Appl* 38:957–968
- [16] Kaveh A, Zolghadr A (2011) Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm. *Asian J Civil Eng* 12:487–509