HPSO BASED LQR APPLIED TO SERVO CONTROL OF SIMO SYSTEM

Karthick SHANMUGAM **Kanthalakshmi SRINIVASAN**
PSG College of Technology, Peelamedu, Coimbatore, 9514264651, skpattitude@gmail.com, skl@ice.psgtech.ac.in

Abstract: *Optimal control is mainly concerned in operating the system with minimum cost. The most promising optimal control strategy available in literature is linear quadratic regulator (LQR). In LQR, it is important to select the state (Q) and control (R) weighting matrices to get optimal results. With no standard guideline for selection of these weighting matrices, the generally adopted trial and error method makes the job of a control engineer more tedious and tiresome. To address this issue, a hybrid particle swarm optimization algorithm (HPSO) to obtain optimal weighting matrices is proposed in this paper. Moreover, the premature convergence of the particles leading to suboptimal results is eliminated by introducing a local convergence monitor, which transforms the entire population at the occurrence of local convergence to a new search space. The proposed HPSO tuned LQR control strategy is applied to cart position tracking and pendulum angle regulatory control of a single inverted pendulum, which is a highly nonlinear unstable system. Experimental results reveal that compared to PSO tuned LQR, HPSO tuned LQR has improved tracking response with smooth error convergence.*

Key words: *HPSO, LQR, Inverted pendulum, Servo control.*

1. Introduction.

The theory of optimal control focuses on operating the system with minimal cost without compromising the quality. One such optimal control algorithm is LQR. The challenges in LQR design lie in the proper selection of Q and R weighting matrices, which determines the performance of the controller. As an usual practice the weights are selected either using the past experience or by trial and error method. As a measure to optimally select the weight matrices, metaheuristic algorithms are used. Particle swarm optimization is the latest member of this kind.

PSO based optimal weighting matrices selection of LQR is carried out for a sine wave three phase four wire voltage source inverter and it is reported that the effort in tuning the weighting matrices is very much reduced using PSO compared to conventional guess and check method [1]. An adaptive PSO is proposed in [2] to better the search efficacy and to improve the convergence speed. Multiobjective design optimization simulation studies using PSO for switched reluctance motor (SRM) is carried out and it is reported that the designed SRM developed better torque compared to the normal design [3]. Adaptive PSO tuned LQR for the attitude tracking of a 2 DoF helicopter and servo control of an inverted pendulum was proposed in [4] and [5] respectively. It is reported that adaptive PSO tuned LQR outperforms the conventional PSO tuned LQR. PSO algorithm is effectively used in load frequency control of power systems [6]. PSO together with dynamic objective constraint handling is used to find the state feedback controller gains for stabilizing controller in a linear inverted pendulum [7]. Trajectory optimization for manipulator motion planning using PSO is investigated and experimented successfully [8]. A state estimation method using HPSO was developed and applied to practical power distribution system. It is claimed that the proposed method has better convergence characteristics compared to the standard PSO [9]. In [10] PSO is used to find the direct and quadrature axis stator inductances and resistances of permanent-magnet synchronous machines with the aid of experimental measurements. It is reported that the developed PSO has fast and stable convergence characteristics, and it is relatively easy to implement. A new mechanism is added along with tuning of few parameters in PSO to improve its robustness in finding the global solution [11].This stochastic-based search algorithm had been widely used in recent years to find optimum solutions in both academic and practical problems. On the other end, PSO has a demerit that particles converge in local optima resulting in suboptimal results. A perturbed particle updating strategy is employed to deal with the problem of premature convergence [12]. To address the premature convergence, hybrid PSO (HPSO) combining the advantages of space transformation search (STS) and modified velocity model for standard mathematical functions is theoretically proven in [13]. In this work, the HPSO based LQR approach is extended to a single input multi output (SIMO) system, performance assessment of the proposed approach is verified over a single inverted pendulum, a benchmark SIMO system and its effectiveness is demonstrated in comparison with standard PSO results.

2. Problem Formulation

Consider a linear time invariant (LTI) system whose state and output dynamics can be written as follows

$$\overset{\circ}{X}(t) = AX(t) + Bu(t) \quad (1)$$

$$Y(t) = CX(t) + Du(t) \quad (2)$$

where A, B, C and D are the system, input, output and direct transition matrices respectively. The purpose of LQR design is to minimize the following cost function

$$J(u^*) = \frac{1}{2}\int_0^\alpha [X^T(t)QX(t) + u^T(t)Ru(t)]dt \quad (3)$$

where Q and R are the positive semi definite and positive definite matrices respectively, popularly called as the state and control weighting matrices. The state feedback gain K can be calculated by solving

$$K = R^{-1}B^T P \quad (4)$$

where P is the solution of following algebraic Riccati equation (ARE)

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (5)$$

The Q and R matrices play a vital role in determining the performance of the controller. The Q and R matrices are selected based on the

(i) Trial and error approach. This method is time consuming and it does not result in optimal response.

(ii) Particle swarm intelligence approach, which may lead to suboptimal results due to premature convergence of the particles.

Hence, to address these problems in weighting matrices selection of LQR, hybrid particle swarm intelligence, a combination of state transformation search and modified velocity model is proposed.

3. HPSO Algorithm.

3.1. Space Transformation Search (STS)

Evolutionary algorithm starts with some arbitrary solution and move towards the optimal solution. The iteration or process usually terminates either with predefined iteration number or with the satisfaction of predefined conditions. In PSO, particles migrate through the search space using the following position (x) and velocity (v) update equations.

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (6)$$

$$v_i^d(t+1) = w \times v_i^d(t) + c_1 r_1 (p_{bestp_i}^d - x_i^d(t)) + c_2 r_2 (p_{gbest}^d - x_i^d(t)) \quad (7)$$

where $p_{bestp_i}^d$ and p_{gbest}^d are the particles local best and global best positions, r_1 and r_2 are the random numbers, c_1 and c_2 are the cognitive coefficients , w is the inertia weight, i is the particle index and d is the dimension of the decision variables. STS mechanism introduces a watchdog to monitor the occurrence of premature convergence. If the current search space hardly contains any global solution, STS mechanism transforms current search space to a new search space called the transformed space. The new transformed solution $x*$ can be calculated as follows:

$$x^* = k(a+b) - x \quad (8)$$

x € R within an interval of [a, b] and k can be set as a random number within [0, 1], where a, b are the particles minimum and maximum values. To be more specific, for an optimization problem of d decision variables, according to the definition of the STS [13], the new dynamic STS model is defined by

$$x_i^d = k[a_i^d(t) + b_i^d(t)] - x_i^d \quad (9)$$

$$a_i^d(t) = \min(x_i^d(t)), \quad b_i^d(t) = \max(x_i^d(t)) \quad (10)$$

The sum of the particles maximum and minimum positions are multiplied by a random value k and it is subtracted from the actual particle positions to transform the search space. The simultaneous evaluation of solutions in the current search space and transformed space is done and the search space giving the minimum cost is finalized as the current search space. Moreover, the interval boundaries $[a_i^d(t), b_i^d(t)]$ are dynamically updated according to the size of current search space.

3.2. Modified Velocity Model

In PSO, particles are attracted to their corresponding previous personal best (p_{bestp_i}) and global best (p_{gbest}) positions. As iteration progresses, particles move very close to p_{bestp_i} and p_{gbest} respectively. Due to this the difference between p_{bestp_i} and the current particle position x_i becomes very small, and this will be same for the global best particles. Moreover, according to the velocity update equation the velocity becomes very small. Once p_{bestp_i} or p_{gbest} falls into local minima, all particles in the swarm will quickly converge into local minima leading to premature convergence. All the particles will be stagnant and the chance to escape from local minima becomes very less. As a measure to overcome local minima trapping, this paper proposes a convergence monitor to watchdog each p_{bestp_i} and p_{gbest} positions in the search space. If the value of the convergence monitor reaches the threshold limit, a new modified velocity model is introduced to disturb the position of the particles by providing a disturbance factor in the cognitive and social part of the velocity update equation.

$$v_i^d(t+1) = wv_i^d(t) + c_1r_1(p_i^d(t) - d_1x_i^d(t)) + c_2r_2(p_i^d(t) - d_2x_i^d(t)) \tag{11}$$

Where d_1 and d_2 are the disturbance factors with a random value within [0,1]. The pseudo code of hybrid particle swarm optimization algorithm is shown in Table 1.

Table 1
Pseudo code: HPSO

```
Initialize the particles in swarm arbitrarily
for i ≤ 100
    set convergence monitor (S) = 0
    Evaluate the cost function $f = ISE = \int e^2(t)dt$
    for i = 1 to 30
        if $f < f_{\text{pbesti}}$
            $f_{\text{pbesti}} \leftarrow f$
            $x_{\text{pbesti}} \leftarrow x_{\text{i}}$
        end if
        if $f < f_{\text{gbesti}}$
            $f_{\text{gbesti}} \leftarrow f$
            $x_{\text{gbesti}} \leftarrow x_{\text{i}}$
        else if
            S = S+1
        end if
        if S > Sthreshold
            for d = 1 to dimensions
                update the particles position and velocities using equations 6 and 11
            end for
        else if
            for d = 1 to dimensions
                update the particles position and velocities using equations 6 and 7
            end for
        end if
    end for
end for
```

4. Single Inverted Pendulum.

The effectiveness of HPSO tuned LQR framework is demonstrated using single inverted pendulum, a typical single input multiple output (SIMO) benchmark system. This system consists of two encoders, one to measure the pendulum angle and the other to measure the position of the cart. Fig. 1 shows the schematic diagram of a single inverted pendulum.

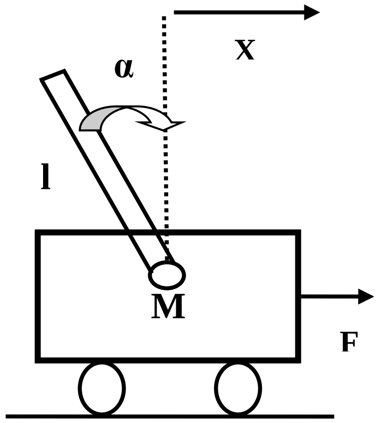

Fig. 1. Schematic diagram of Single Inverted Pendulum.

Stabilization control is the control scheme used to meet the control objective of maintaining the pendulum angle at zero degree, while the cart tracks the reference trajectory. Due to the practical limitation on control input (motor voltage) given to the cart system, stabilization control is implemented using LQR. Based on Euler-Lagrangian energy approach the nonlinear equation of motion of pendulum can be written as

$$(M_c + M_p)\ddot{x}_c(t) + B_{eq}\dot{x}_c(t) - (M_pl_p\cos(\alpha(t)))\ddot{\alpha}(t) + M_pl_p\sin(\alpha(t))\dot{\alpha}^2(t) = F_c(t)$$

and

$$-M_pl_p\cos((\alpha(t))\ddot{x}_c(t) + (I_p + M_pl_p{}^2)\ddot{\alpha}(t) + B_p\dot{\alpha}(t) - M_pgl_p\sin(\alpha(t)) = 0$$

Four variables namely, cart position, cart velocity, pendulum angle, and pendulum velocity are taken as state variables and the state space model is obtained. By linearizing the model around the equilibrium point $\sin(\alpha) \cong \alpha, \cos(\alpha) \cong 1$, the linearized model of the inverted pendulum can be written as

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{(M_pl)^2}{q} & \dfrac{-B_{eq}(M_pl^2+I)}{q} & \dfrac{M_plB_p}{q} \\ 0 & \dfrac{(M+M_p)M_pgl}{q} & \dfrac{M_plB_{eq}}{q} & \dfrac{(M+M_p)B_p}{q} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \dfrac{M_pl^2+I}{q} \\ \dfrac{M_pl}{q} \end{bmatrix}$$

For the controller design, the system parameters are borrowed from [5], and by substituting system parameters the following state model is arrived.

$$\begin{bmatrix}\dot{x}_c\\ \dot{\alpha}\\ \ddot{x}_c\\ \ddot{\alpha}\end{bmatrix}=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&2.2643&-15.8866&-0.0073\\0&27.8203&-36.6044&-0.0896\end{bmatrix}\begin{bmatrix}x_c\\ \alpha\\ \dot{x}_c\\ \dot{\alpha}\end{bmatrix}+\begin{bmatrix}0\\0\\2.2772\\5.2470\end{bmatrix}u$$

$$y=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x_c\\ \alpha\\ \dot{x}_c\\ \dot{\alpha}\end{bmatrix}$$

5. Results and Discussion.

HPSO tuned LQR framework is implemented for servo control problem of an inverted pendulum and the dynamic performance of conventional PSO tuned LQR framework is also compared in this work. HPSO based LQR servo control algorithm is implemented in MATLAB. The parameters used for HPSO and PSO algorithms are shown in Table 2.

Table 2
Parameters of HPSO and PSO algorithms

Parameters	HPSO	PSO
No. of Population (N)	30	30
No. of Iterations (i)	100	100
Dimensions (d)	3	3
c_1	0.9	0.9
c_2	1.2	1.2
Inertia weight (w)	0.9	0.9
Convergence Monitor	Yes	-
d_1 and d_2	Random Values	-

Parameters for both the algorithms remain the same except for the presence of convergence monitor and the disturbance factors d_1 and d_2. In HPSO technique according to the cost or fitness function ISE, the optimization algorithms are executed for the specified number of iterations and with the help of convergence monitor and the disturbance factor in the velocity update, the global best of the particles, so called the weights of LQR, are obtained. Table 3 gives the corresponding Q and R matrices and controller gain of LQR obtained using the PSO and HPSO algorithms.

Table 3
Parameters of PSO and HPSO algorithms

Optimization algorithm	Weighting matrices	Controller gain
PSO	$Q=\begin{bmatrix}31.88&0&0&0\\0&8.97&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$ $R=[0.22]$	$K=\begin{bmatrix}-82.61\\145.47\\-53.16\\18.85\end{bmatrix}^T$
HPSO	$Q=\begin{bmatrix}630.51&0&0&0\\0&26.017&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$ $R=[0.0008]$	$K=\begin{bmatrix}-280.73\\350.60\\-144.76\\43.86\end{bmatrix}^T$

The particles best positions of the HPSO and PSO algorithms are illustrated in Fig. 2. where the X-axis represents the number of decision variables, Y-axis represents the number of iterations and Z-axis represents the matrix dimensions.

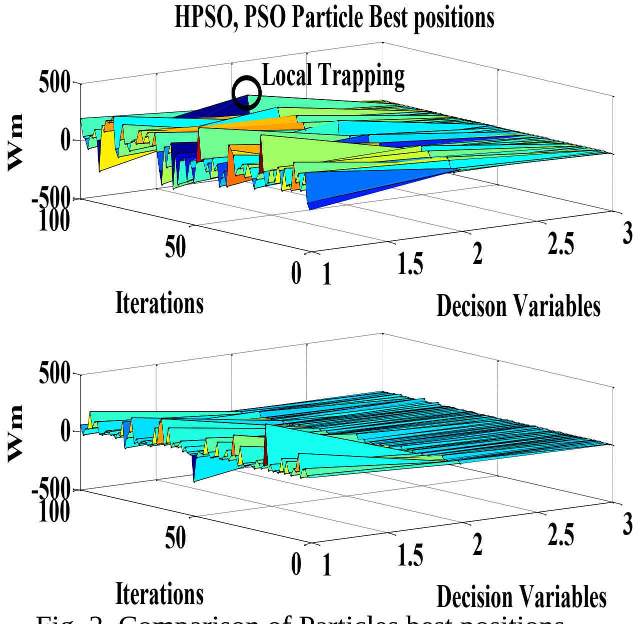

Fig. 2. Comparison of Particles best positions.

From the Z-axis dimensions it is evident that an intensive search occurs in HPSO. All the three decision variables are scattered in the search space to find the global best solution, whereas in PSO the first two decision variables are more scattered than the third decision variable. It is worth to note that in the iteration number 75 of HPSO, whole population transformation occurs due to local trapping. Integral square error is taken as the performance index and the fitness function convergence is illustrated in Fig.3.

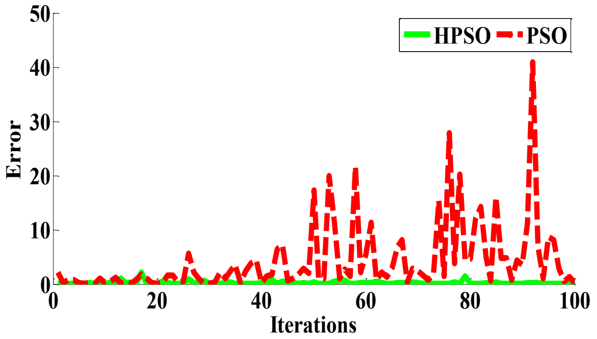

Fig. 3. Fitness function convergence of PSO and HPSO.

From the illustration it is evident that smooth convergence occurs in HPSO compared to PSO tuned LQR framework. On the successful completion of the specified number of iterations, global best of the particles are obtained.

4.1. Trajectory Tracking Response

A square trajectory having amplitude of 20 cm (peak to peak) with a frequency of 0.05 Hz is given as input to the system. The output responses of HPSO and PSO tuned LQR is illustrated in Fig. 4 and the zoomed view of the same is given in Fig. 5.

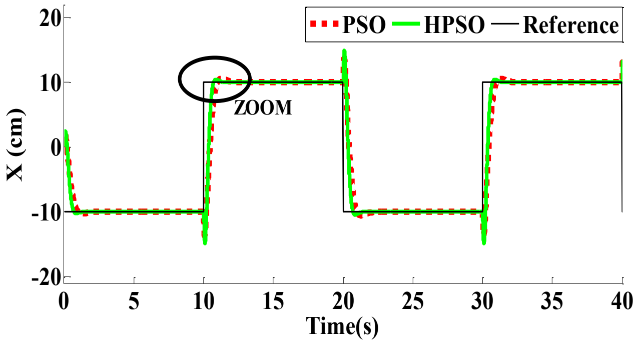

Fig. 4. Cart position for square trajectory.

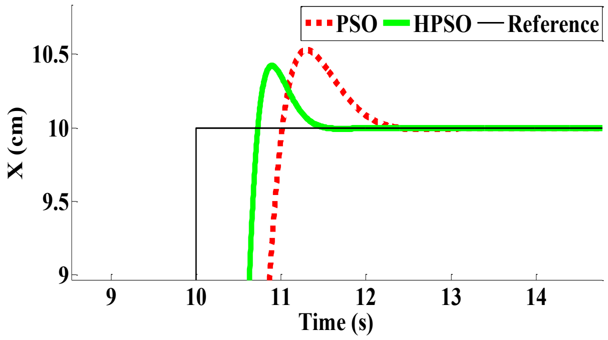

Fig. 5. Zoomed view of Cart position.

It is evident that the response of HPSO tuned LQR framework is appealing compared to PSO tuned framework in terms of maximum peak overshoot, rise time and settling time. Pendulum angular response for the test signals are shown in Fig. 6, 7 and 8.

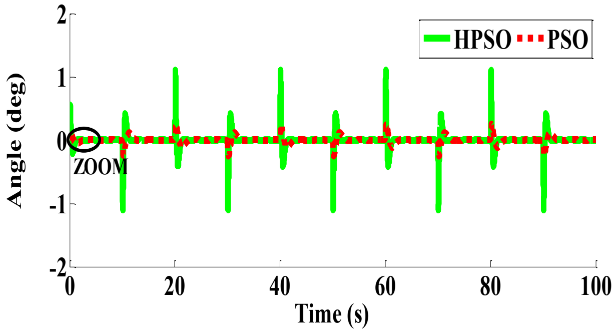

Fig. 6. Pendulum angular response for square trajectory.

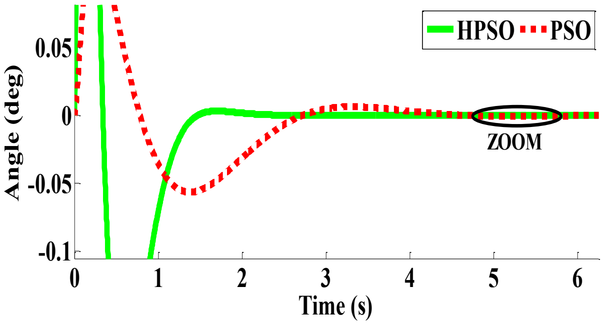

Fig. 7. Zoomed view of pendulum angular response for square trajectory.

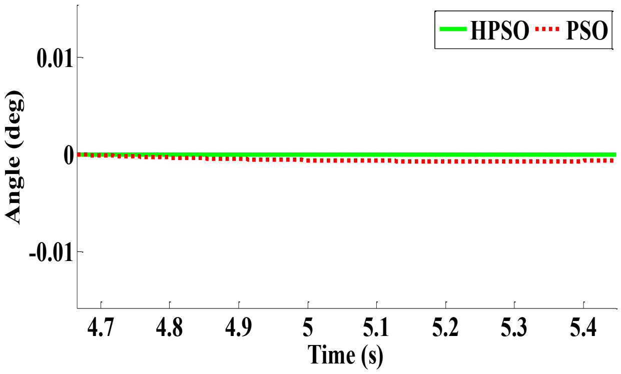

Fig. 8. Zoomed view of pendulum angular response for square trajectory in steady state.

Table 4
Comparison of Cart position response

Optimization method	Time domain parameters		
	t_d	t_s	%Mp
PSO	0.45	1.01	5.3
HPSO	0.37	0.72	4.2

Moreover, from table 4 it can be inferred that maximum peak overshoot is reduced by 20.7 %, settling time is reduced by 28 % and the delay time is reduced by 17.7 % in HPSO algorithm compared to

PSO algorithm. Table 5 gives the deviation and convergence time of pendulum angular response.

Table 5
Pendulum angle response

Optimization algorithm	Convergence time (s)	$e_{ss}(\theta)$
PSO	4.8	0.001
HPSO	2.5	0

The convergence time of HPSO tuned LQR framework is appealing than the PSO tuned LQR. From table 5 it can be inferred that in comparison of HPSO with PSO the convergence time is reduced by 47.9 % and the steady state error is zero in HPSO algorithm. It is evident from the analysis that the HPSO tuned LQR controller performance is dynamic in servo control applications.

6. Conclusions

In this paper, the premature convergence problem of PSO tuned LQR has been solved using HPSO and the efficacy of the controller has been tested on a single inverted pendulum. Trapping up of the particles in local optima is identified by the convergence monitor and, the convergence in sub optimal solutions due to premature convergence is avoided by introducing a disturbance factor in the velocity update along with the transformation in search space. The trajectory tracking response of inverted pendulum shows that compared to PSO tuned LQR, the HPSO tuned LQR can result in not only improved tracking response but also reduced tracking error.

References

1. Bartlomiej Ufnalski, Arkadiusz Kaszewski, Lech Grzesiak M.: *Particle Swarm Optimization of the Multioscillatory LQR for a Three-Phase Four-Wire Voltage-Source Inverter With an LC Output Filter*, In: IEEE Transactions On Industrial Electronics, (2015), Vol. 62, No.1, p. 484-493.
2. Zhi-Hui Zhan, Jun Zhang, Yun Li, Henry Shu-Hung Chung: *Adaptive Particle Swarm Optimization,* In: IEEE Transactions on Systems, Man, And Cybernetics: Systems, (2009), Vol. 39, No. 6 pp. 1362-1381.
3. Cong Ma, Liyan Qu.: *Multiobjective Optimization of* Switched *Reluctance Motors Based on Design of Experiments and Particle Swarm Optimization*, In: IEEE Transactions on Energy Conversion, (2015), Vol. 30, No. 3, p. 1144–1153.
4. Elumalai Vinodh Kumar, Ganapathy Subramaniam Raaja, Jovitha Jerome: *Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter*, In: Applied Soft Computing, (2016), Vol. 41, p. 77–90.
5. Karthick S., Jovitha Jerome, Vinodh Kumar E., Raaja G.: *APSO Based Weighting Matrices Selection of LQR Applied to Tracking Control of SIMO System*, In: Proceedings of 3rd International Conference on Advanced Computing Networking and Informatics, Smart Innovation, Systems and Technologies, (2016), No.43, Vol. 1, no. 2, p.11–20
6. Rania Helmy Mansour Hennas Magdy A.S., Aboelela: *Development of distributed LFC controllers using adaptive weighted PSO algorithm*, In: Journal of Electrical Engineering, (2016), Vol. 16, p. 55-60.
7. Mahmud Iwan Solihin, Rini Akmeliawati: *Particle Swarm Optimization for Stabilizing Controller for a Self-Erecting Linear Inverted Pendulum*, In: International Journal of Electrical and Electronic Systems Research, (2010), Vol. 3, p. 13-23.
8. Jeong-Jung Kim, Ju-Jang Lee: *Trajectory Optimization with Particle Swarm Optimization for Manipulator Motion Planning*, In: IEEE Transactions on Industrial Informatics, (2015), Vol. 11, no.3, p. 620-631.
9. Shigenori Naka, Takamu Genji, Toshiki Yura, Yoshikazu Fukuyama: *A Hybrid Particle Swarm Optimization for Distribution State Estimation*, In: IEEE Transactions on Power Systems, (2003), Vol. 18, No.1, p. 60-68.
10. Omar Sandre-Hernandez, Roberto Morales-Caporal, Jose Rangel-Magdaleno, Hayde Peregrina-Barreto, Julio Noel Hernandez-Perez: In: *Parameter Identification of PMSM's Using Experimental Measurements and a PSO Algorithm*, In: IEEE Transactions on Instrumentation and Measurement, (2015), Vol. 64, No.8, p. 2146-2154.
11. Ting-Yu Chen, Tzu-Ming Chi: *On the improvements of the optimization algorithm,* In: Advances in Engineering Software (2010), Vol. 41, No.2, p. 229-239.
12. Zhao Xinchao: *A perturbed particle swarm algorithm for numerical optimization*, In: Applied Soft Computing, (2010), Vol. 10, No. 1, pp. 119-124.
13. Song Yu, Zhijian Wu, Hui Wang, Zhangxin Chen, and He Zhong.: *A Hybrid Particle Swarm Optimization Algorithm Based on Space Transformation Search and a Modified Velocity Model*, In: International Journal of Numerical Analysis and Modeling, (2012), Vol. 9, No. 2, p.371–377.