International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 03, 2021

Real Time Hpso Tuned Lqr For Servo Control Of A Simo System

Karthick S[1,] Kanthalakshmi S[2]

[1]Assistant Professor(Sr.Gr), Dept of I&CE, PSG College of Technology, Coimbatore.
[2]Professor, Dept of EEE, PSG College of Technology, Coimbatore.

Abstract: This work deals with servo control problem of a single inverted pendulum using Hybrid Particle Swarm Optimization (HPSO) tuned Linear Quadratic Regulator (LQR). It is important to select the state (Q) and control (R) weighting matrices of LQR in an optimal manner to get optimal results. As a general practice these weighting matrices are selected either through trial and error approach or through experience. This practice in particular makes the job of a control person more tedious and tiresome. To address this issue, a hybrid particle swarm optimization algorithm is proposed to obtain optimal weighting matrices. Moreover, the premature convergence of the particles leading to suboptimal results is accounted by introducing a local convergence monitor, which not only transforms the entire population at the occurrence of local convergence to a new search space but also introduces a disturbance factor in the velocity update equation. 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 open loop unstable system. Experimental results reveal that compared to PSO tuned LQR, HPSO tuned LQR has improved tracking response with smooth error convergence.

Keywords— LQR; PSO; HPSO; Single Inverted Pendulum.

1. INTRODUCTION

The theory on optimal control focuses in operating the system with minimal cost without compromising the quality. One such well established optimal control algorithm, which made the life of a control engineer more sophisticated, is linear quadratic regulator (LQR). However, the real challenge in LQR design lies in the proper selection of Q and R weighting matrices, which limits the usage of LQR. As a measure to overcome these issues, recently, metaheuristic algorithms are used to select the weights in an optimal manner. In this context, evolutionary algorithms having the ability in converging to global optima have been widely used. One such computationally inexpensive algorithm chosen to solve this problem is particle swarm optimization (PSO), available in the literature for more than a decade, introduced by J. Kennedy and R. Eberhart [1], Multi-objective binary probability optimization algorithm (MBPOA) is introduced to search for optimal weighting matrices [2]. In an effort to yield an optimal response, LQR plays a vital role in minimizing the quadratic cost function even at small perturbations. This leads to the usage of LQR in many complex systems such as aircraft [3], vibration control [4], and fuel cell systems [5]. The performance evaluation in terms of computational time, computational effort and convergence rate of PSO are compared with GA based feedback controller design in [6], and it is reported that the performance of PSO is healthier than GA. In Fighter tracking problems [7], PSO based LQR is proved superior to LMI based methods. PSO algorithm is effectively used in load frequency control of power systems [8-9], and in shunt active power filter design [10]. Co-evolutionary PSO (CPSO) for constrained optimization problems was proposed in [11]. Simulation studies are carried out and it is claimed that CPSO obtain some solutions better than those which are available in the literature. In reference [12], a hybrid PSO with a feasibility based rule for constrained optimization, in a motive to eliminate the drawback of determining suitable priority factors was proposed. Digital FIR filter with cuckoo search is presented in [13]. Reference [14] presented on the application of PSO combined with computational intelligence for fault detection in machines and it is claimed that PSO gives a success rate of 98.6 to 100%. Even though PSO has all these merits, it has two undesirable characteristics that degrade its exploration abilities. One is premature convergence, that result in diversity loss of the particles and the second is the inability to balance between local search exploitation and global exploration. Too much search exploitation leads to premature convergence of swarm and overemphasize of the global exploration prevents the convergence speed of swarm. All these limitations impose constraint on wider applications of PSO in real world problems [15]. Hence to address this issue in LQR design, hybrid particle swarm optimization (HPSO), a combination of space transformation search and modified velocity model is engaged. The efficiency of the HPSO algorithm is tested on a single inverted pendulum, which is a typical single input multi output (SIMO) open loop unstable system, where the input is the motor voltage and, the cart position and pendulum angle are the outputs.

Problem formulation

Consider a linear time invariant (LTI) system whose state and output equations are written as follows

$$\overset{\circ}{X}(t) = Ax(t) + Bu(t); t \geq 0 \qquad (1)$$

$$Y(t) = Cx(t) + Du(t); t \geq 0 \qquad (2)$$

where *A*, *B*, *C* and *D* are the system, input, output and direct transition matrices respectively. The purpose of LQR design is to compute the optimal weighting matrices that minimizes the following cost function.

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

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

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

Where *P* is the solution of the following algebraic Riccati equation.

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

The *Q* and *R* matrices play an essential role in determining the performance of the controller. If this *Q* and *R* matrices are selected based on the

- Trial and error approach, it does not result in optimal response.
- Particle swarm intelligence, it may lead to suboptimal results due to premature convergence of the particles.

Hence, to address this problem in LQR, hybrid particle swarm intelligence is proposed.

HPSO

The proposed HPSO to tune the LQR weights is a combination of space transformation search and modified velocity model.

1.1. Space Transformation Search (STS)

Most of the evolutionary algorithms starts with some arbitrary solution and make an effort to improve towards the optimal solutions. The iteration or process terminates either with predefined iteration number or with the satisfaction of predefined conditions. In PSO, particles fly through the search space using the following position and velocity update equations.

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

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

where $p_{bestp_i}^d$ and p_{gbesti}^d are the particles 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. In a few cases the search ends with local optima leading to sub-optimal solutions. This is one of the major demerits of PSO and this problem is addressed by space transformation search (STS) algorithm. STS algorithm introduces a mechanism that will act as a watchdog to monitor the occurrence of premature convergence. Under these situations the current search space hardly contains the global solution [16]. Now, STS algorithm transforms current search space to a new search space called the transformed space. The new transformed solution x^* in the transformed space S can be calculated as follows:

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

x € R within an interval of [a, b] and k can be set as a random number within [0, 1]. Where a and 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, the new dynamic STS model is defined by

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

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

The sum of the particles maximum and minimum positions are multiplied by a random number k and it is subtracted from the actual particle positions to get the transformed 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.

1.2. Modified velocity model

In PSO particles are attracted to their corresponding previous personal best (P_{besti}) and global best (g_{besti}) positions. As iteration progresses, particles move very close to P_{besti} and g_{besti} respectively. Due to this the difference between P_{besti} 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_{besti} or g_{besti} 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 this drawback this paper proposes a convergence monitor to watchdog each P_{besti} and g_{besti} 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

$$v_i^d(t+1) = wv_i^d(t) + c_1 r_1 (p_i^d(t) - d_1 x_i^d(t)) + c_2 r_2 (p_i^d(t) - d_2 x_i^d(t)) \qquad (10)$$

where d_1 and d_2 are the disturbance factors with a random value within [0,1]. The pseudocode of HPSO is shown in Table 1.

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. Problem formulation starts with linear time invariant (LTI) system and here nonlinearity is duly appreciated.

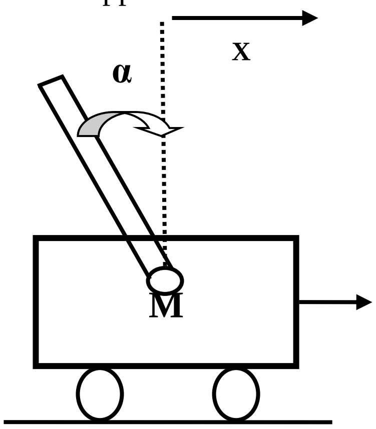

Fig. 1. Schematic diagram of Single Inverted Pendulum.

Fig. 2.

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. Stabilization control is the control scheme used to meet the control objectives of maintaining the pendulum angle at zero degree with simultaneous tracking of cart 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

$$(M_c + M_p)\ddot{x}_c(t) + B_{eq}\dot{x}_c(t) - (M_p l_p \cos(\alpha(t)))\ddot{\alpha}(t) + M_p l_p \sin(\alpha(t))\dot{\alpha}^2(t) = F_c(t) \quad (11)$$

and

$$-M_p l_p \cos((\alpha(t))\ddot{x}_c(t) + (I_p + M_p l_p^{\,2})\ddot{\alpha}(t) + B_p\dot{\alpha}(t) - M_p g l_p \sin(\alpha(t)) = 0 \quad (12)$$

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 $\sin(\alpha) \cong \alpha, \cos(\alpha) \cong 1$. Therefore the linearized model of the inverted pendulum can be written as

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{(M_p l)^2}{q} & \frac{-B_{eq}(M_p l^2 + I)}{q} & \frac{M_p l B_p}{q} \\ 0 & \frac{(M + M_p) M_p g l}{q} & \frac{M_p l B_{eq}}{q} & \frac{(M + M_p) B_p}{q} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{M_p l^2 + I}{q} \\ \frac{M_p l}{q} \end{bmatrix} \quad (13)$$

For the controller design the single inverted pendulum system parameters shown in Table 2 are taken from quanser IP01 and IP02 user manual [17]. By substituting those parameters in the A and B matrices the following state representation is arrived.

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

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

Table 1
Pseudocode of HPSO

```
Initialize the particles in  swarm arbitrarily
for  i ≤ N; Fixed number of iterations
                              set convergence monitor (S) = 0
Evaluate the cost function f = ISE = ∫e²(t)dt
for  i = 1 to n; To check for local convergence.
if f < fpbesti
fpbesti  ←  f
xpbesti ←  xi
end if
if f < fgbesti
fgbesti  ←  f
xgbesti  ←  xi
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
end if
end for
end for
```

Table 2
List of parameters

Symbol	Description	Value/Unit
R	Motor armature resistance	2.6Ω
L	Motor armature inductance	0.18 mH

K_t	Motor torque constant	0.00767 Nm/A
K_m	Motor EMF constant	0.00767 Ns/rad
J_m	Rotor moment of inertia	3.9*10–7 kgm2
K_g	Gearbox ratio	3.71
r_{mp}	Motor pinion radius	6.35*10–3 m
r_p	Position pinion radius	1.48*10–2 m
B_{eq}	Equivalent viscous damping coefficient at motor	5.4 Nms/rad
B_p	Viscous damping coefficient at pendulum pivot	5.4 Nms/rad
l_p	Pendulum length from pivot to centre of mass	0.3302 m
I	Pendulum moment of inertia	7.88*10–3 kgm2
M_p	Pendulum mass	0.23 kg
M	Cart mass	0.94 kg
K_t	Motor torque constant	

2. EXPERIMENTAL RESULTS AND DISCUSSION

HPSO tuned LQR framework is implemented in servo control problem of an inverted pendulum and the dynamic performance over conventional PSO tuned LQR framework is also compared in this work. HPSO based LQR servo control algorithm is implemented in MATLAB 2011a. Hardware setup is shown in Fig.2. The number of decision variables to be optimized for the servo control of the single inverted pendulum is chosen to be three (q11, q22 and R) and the parameters used for HPSO and PSO algorithms are shown in Table 3. From Table 3 it can be inferred that the parameters for both algorithms remain the same except the presence of convergence monitor and the disturbance factors d1 and d2. According to the cost or fitness function ISE, the optimization algorithms are executed for specified number of iterations and with the help of convergence monitor and disturbance factor in the velocity update, the global best of the particles, so called the weights of LQR, are obtained. Table 4 gives the corresponding Q and R matrices and controller gain K of LQR obtained using the PSO and HPSO algorithms.

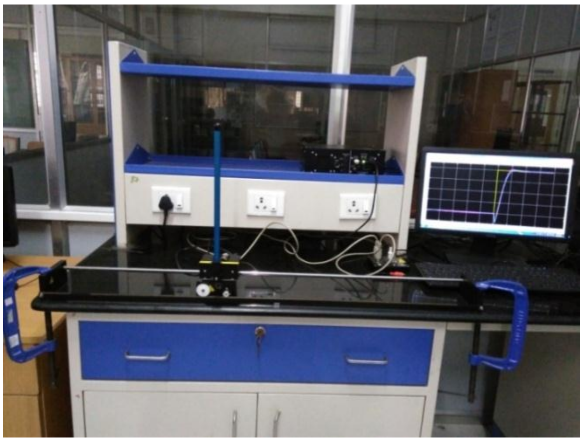

Fig. 3. Single Inverted Pendulum hardware set-up

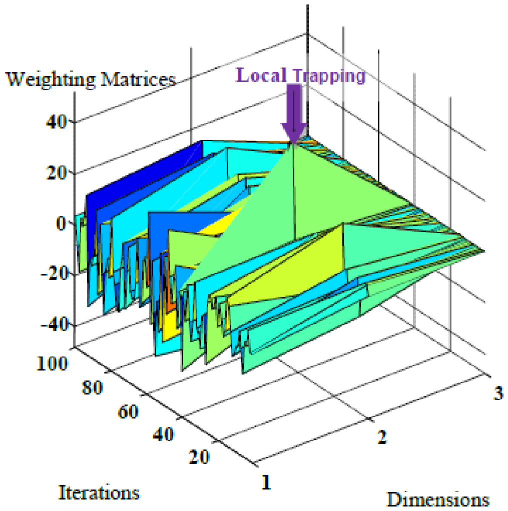

Fig. 4. HPSO Particle Best Positions.

The particles best positions of the HPSO and PSO algorithms are illustrated in Fig. 3 and Fig. 4, where the X-axis represents the number of decision variables, Y-axis represents the number of iterations and Z-axis represents the weighting matrix dimensions. It is worth to note in Fig. 3 that in the iteration number 70, whole population transformation occurs due to local trapping, and it is also clear that the particle is transformed to new search space in the consecutive iterations. Thus trapping up of particles in local minima is identified by the convergence monitor and the transformation of particles to new search space is taken care by the STS algorithm and, the speed of convergence is taken care by modified velocity model.

International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 02, 2021

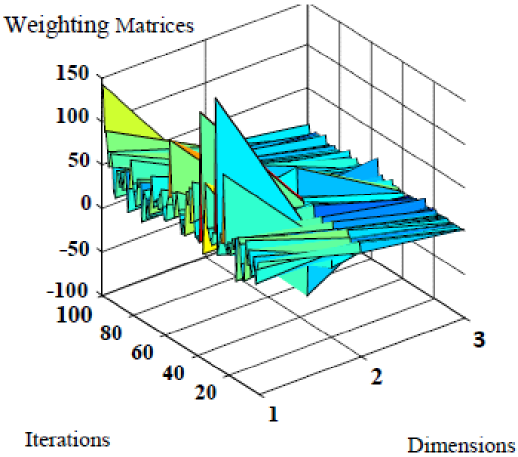

Fig. 5. PSO particle best positions

Comparing the Z-axis dimension of Fig. 3 and Fig. 4 it is evident that, smooth convergence occurs in HPSO compared to PSO tuned LQR framework.

Table 3
Parameters of HPSO and PSO algorithms

2.1. Trajectory tracking response

A magnitude of 200 mm (peak to peak), frequency of 0.05 Hz square trajectory is given as input to the system to test the transient and steady state behavior of the system for the proposed HPSO tuned weighting matrices of LQR. The corresponding output responses of PSO and HPSO tuned LQR are illustrated in Fig. 5.

Table 4
Controller Parameters of PSO and HPSO Algorithm

Parameters	HPSO	PSO
No of Population (N)	30	30
No of Iterations (i)	100	100
Dimensions (d)	3	3
C1	0.9	0.9
C2	1.2	1.2
Inertia weight (w)	0.9	0.9
Convergence Monitor	yes	-
d1 and d2	Random values	-

Optimization algorithm	Weighting matrices	Controller gain
PSO	$Q=\begin{bmatrix}31.88 & 0 & 0 & 0\\ 0 & 8.97 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$ $R=[0.0006]$	$K=\begin{bmatrix}-164.8039\\ 269.1288\\ -98.5455\\ 32.0641\end{bmatrix}^T$
HPSO	$Q=\begin{bmatrix}63.0519 & 0 & 0 & 0\\ 0 & 26.0176 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$ $R=[0.0008]$	$K=\begin{bmatrix}-280.7399\\ 350.6088\\ -144.7649\\ 43.8623\end{bmatrix}^T$

Table 5
Time Response Analysis of Cart Position

Optimization method	**Time domain parameters**	
	$t_d(s)$	$t_s(s)$
PSO	0.522	3.25
HPSO	0.426	1.273

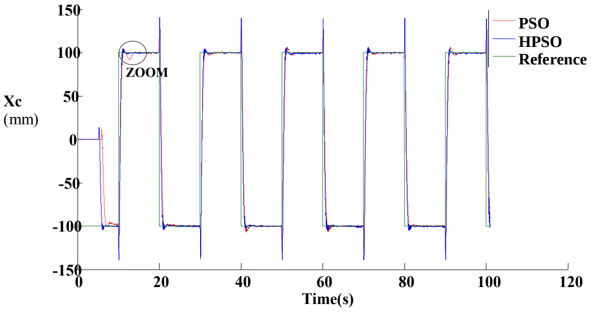

Fig. 6. Cart position for square trajectory

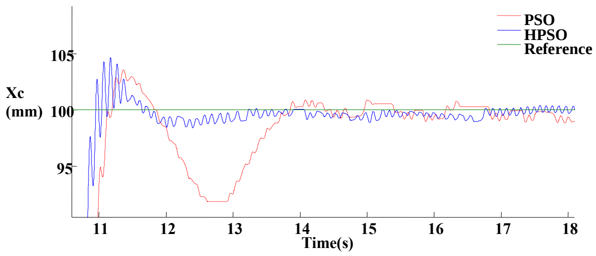

Fig. 7. Zoomed view of cart position for square trajectory

Fig. 8.

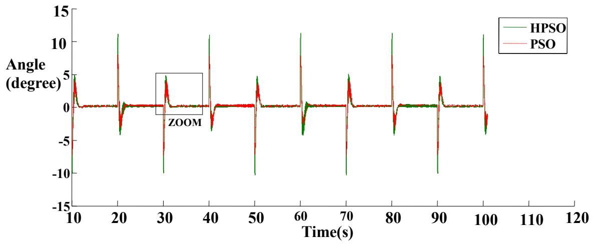

Fig. 9. Pendulum angle for square trajectory

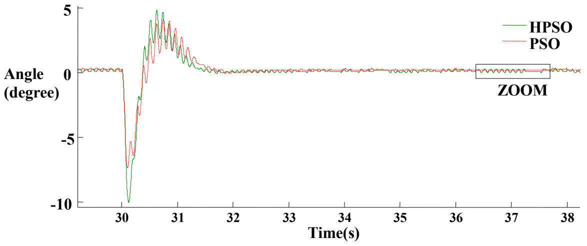

Fig. 10. Zoomed view of pendulum angle showing the transient response

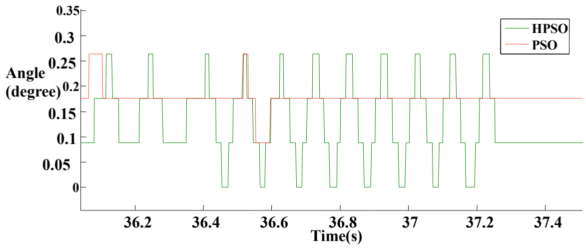

Fig. 11. Zoomed view of pendulum angle showing the steady state response

Table 6
Statistics of Pendulum Angle Response

Optimization algorithm	**Median**	**Mode**
HPSO	0.1758	0.0878
PSO	0.1758	0.1758

From table 5 it can be inferred that settling time is reduced by 60.6 % and the delay time is reduced by 18.3 % in HPSO algorithm compared to PSO algorithm. It is evident from the illustrations that the response of HPSO tuned LQR framework is appealing compared to PSO tuned framework in terms of delay time and settling time. Pendulum angular response for the test signal is shown in Fig. 7, the zoomed view shown in Fig. 8 is to depict the transient behavior, and the zoomed view in Fig.9 is to suppose the steady state behavior.
The steady state statistical analysis of the pendulum angular response is depicted in Table 6. It is worth to note that the mode of the pendulum angle for HPSO tuned LQR is 0.0878 and that of PSO tuned LQR is 0.1758. The mode in Table 6 represents the value that appears most often in a set of data. It is evident from the mode that HPSO outperforms PSO.

Table 7
Pendulum Angle Response

Optimization algorithm	**Convergence time (s)**
PSO	1.83
HPSO	1.57

From Table 7 it can be inferred that the convergence time is reduced by 14.2 % in HPSO algorithm compared to PSO algorithm. It is evident from the analysis that, the HPSO tuned LQR controller performance is dynamic in servo control applications.

3. 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 quanser 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 prevented 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, HPSO tuned LQR can result in not only improved tracking response but also reduced tracking error.

4. REFERENCES

[1] Kennedy, J and Eberhart, R.C., “Particle swarm optimization,” In: Proceedings of the 1995 IEEE International Conference on Neural Networks, IEEE Service Center, Piscataway, NJ, pp. 1942–1948, 1995.

[2] Wang L., Ni H., Zhou W., Pardalos P.M., Fang J and Fei M.,. "MBPOA-based LQR controller and its application to the double-parallel inverted pendulum system," Engineering Applications of Artificial Intelligence 36, pp. 262-268, 2014.
[3] Usta M.A., Akyazi O and Akpinar A.S., "Aircraft roll control system using LQR and fuzzy logic controller," In: IEEE conference on Innovations in Intelligent Systems and Applications (INISTA), Istanbul, pp. 223–227, 2011.
[4] Ang K.K., Wang S.Y and Quek S.T., "Weighted energy linear quadratic regulator vibration control of piezoelectric composite plates," Journal of Smart Materials and Structures vol. 1, pp. 98–106. 2002.
[5] Niknezhadi A., Miguel A.F., Kunusch C and Carlos O.M., "Design and implementation of LQR/LQG strategies for oxygen stoichiometry control in PEM fuel cells based systems," Journal of Power Sources vol. 9, pp. 4277–4282, 2011.
[6] Karthick S., Jovitha Jerome, Vinodh Kumar E and 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 , vol. 1, pp.11–20, 2016.
[7] Panda S and Padhy N.P., "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design," Applied Soft Computing, vol. 8, No.4, pp. 1418-1427, 2008.
[8] Rania Helmy Mansour Hennas Magdy A.S and Aboelela., "Development of distributed LFC controllers using adaptive weighted PSO algorithm," Journal of Electrical Engineering, vol. 16, pp. 55-60, 2016.
[9] Ngoc-Khoat Nguyen Qi Huang and Thi-Mai-Phuong Dao., "Applying particle swarm optimization to design adaptive fuzzy logic load-frequency controllers for a large-scale power system," Journal of Electrical Engineering, vol. 15, pp. 155-162, 2015.
[10] Hussien Ibrahim Abdul-Ghaffar Essamudin Ali Ebrahim Abou-Hashima and El-Sayed., "Particle swarm optimization-based control of shunt active- power filter with both static and dynamic loads, vol. 16, pp. 296-307, 2016.
[11] Qie He and Ling Wang., "An effective co-evolutionary particle swarm optimization for constrained engineering design problems." Engineering applications of artificial intelligence, vol. 20, pp. 89-99, 2006.
[12] Qie He and Ling Wang., "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," Applied mathematics and computation, pp. 1407-1422, 2007.
[13] I. Sharma , B. Kuldeep, A. Kumar and V.K. Singh., "Performance of swarm based optimization techniques for designing digital FIR filter: A comparative study," Engineering Science and Technology, an International Journal, vol. 19, pp. 1564–1572, 2016.
[14] B. Samanta and C.Nataraj., "Use of particle swarm optimization for machinery fault detection," Engineering Applications of Artificial Intelligence, vol. 22, pp. 308–316, 2009.
[15] Lim W.H and Isa N.A.M., "Teaching and peer-learning particle swarm optimization," Applied Soft Computing, vol. 18, pp. 39-58, 2014.
[16] 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," International Journal of Numerical Analysis and Modeling, vol. 9, No. 2, pp.371–377, 2012.
[17] IP01 and IP02 – Single inverted pendulum (SIP): Quanser User Manual