# Performance of Preview Control based on Evolutionary Algorithms

Nidhika Birla and Akhilesh Swarup

*Department of Electrical Engineering, National Institute of Technology, Kurukshetra (INDIA)*
*nidhikabirla@gmail.com, akhilesh.swarup@gmail.com*

***Abstract***

*Preview Control is a field well suited for application to systems that have reference signals known a priori. The use of advance knowledge of reference signal can improve the tracking quality of the concerned control system. The classical solution to the Preview Control problem is obtained using the Algebraic Riccati Equation. The solution obtained is good but it is not optimal and has a scope of improvement, as the Preview Control problem has many parameters to be defined and optimized. The Evolutionary Algorithms, inspired by real-time natural systems, are a solution to this multi-dimensional problem.*

*This paper studies the performance of Preview Control optimized by three Evolutionary Algorithms (Genetic Algorithm, Particle Swarm Optimization and Marriage in Honey Bees Optimization) for two bench-mark control systems: Industrial Servo and Inverted Pendulum System in terms of processing time, convergence characteristics and the quality of solution obtained.*

*The results of this paper illustrate the benefits and weaknesses of the Evolutionary Algorithms for solving the Preview Control problem. The PSO algorithm proves to be the best for Preview Control based problems. It performs well in both small and large search spaces. The study reveals that the MBO algorithm requires more computation time while the performance of GA degrades with the increase in number of parameters to be tuned.*

***Keywords:*** *Preview Control, Genetic Algorithm, Particle Swarm Optimization, Marriage in Honey Bees Algorithm*

## 1. Introduction

The field of Preview Control has attracted many researchers as its applications include guidance of autonomous vehicles, robotics and process control. The use of advanced knowledge of reference can improve the tracking quality and performance of transient response of the control system. The classical solution of Preview Control problem is given using $H^\infty$ control and state augmentation, solved using Algebraic Riccati Equation. The mathematical formulation and solution of the $H^\infty$ Preview Control problem is given by A. Kojima, et al. and G. Tadmor, et al., for preview compensation, output feedback setting and fixed lag smoothing [1-4]. The discrete version of the preview control problem and its various issues are studied with numerical examples by Polyakov, et al. [5]. Y. Kuroiwa, et al. have analysed the $H^\infty$ Preview Control problem for the systems with delay [6]. M. M. Negm, et al. has synthesized Optimal Preview Control for three-phase induction motor [7]. Analysis and Design of $H^\infty$ Preview Tracking Control Systems and its various variations using state augmentation have also been studied [8-10]. The solutions to all the problems are given using Algebraic Riccati Equation for the continuous and Discrete Algebraic Riccati Equation for the discrete – time systems.

Despite the mature theory available for Preview control design, there is a scope of improvement in the results obtained using DARE. The solution of Discrete Algebraic Ricccati Equation depends very strongly on the critical parameters chosen, namely, Q (error weighting matrix), R (control weighting matrix) and $\gamma$ (constraint on $H^{\infty}$ norm). The classical method for choice of these parameters is cumbersome and requires complicated calculations. Also, the solution obtained from the classical method has a scope of improvement. The choice of these parameters is more critical in case of $H^{\infty}$ Preview Control problem and the problem complicates even more with the increase in the length of preview. An automated solution to both such problems is the use of Evolutionary Algorithms (EAs).

Evolutionary Algorithms are a class of stochastic search and optimization techniques that work on a principle inspired by nature. These algorithms are inspired by models of adaptation in natural systems that combine the evolutionary adaptation of a population with individual learning within the lifetime of its members. Many of these algorithms are inspired from molecular evolution, population genetics, immunology, etc. The need of Evolutionary Algorithms has developed as the traditional computational algorithms are good at accurate and exact computation but are not designed for processing inaccurate, noisy and complex data. Evolutionary Algorithms, derived from the efficient real-time natural systems, are robust and efficient for solving complex real-world problems. These heuristic algorithms can search the global optimum with a higher probability than deterministic ones.

The paper aims is to find the superlative optimization procedure for automating the solution of Preview Control Problem. The evolutionary techniques evaluated for this purpose are genetic algorithm, particle swarm optimization and marriage in honey bees algorithm, and their performance is assessed in terms of processing time, convergence speed and quality of results.

The paper is organized into six sections. The first section introduces the literature and the objective of the paper. The preview control problem is presented and introduction to the algorithms under study is presented in the second section. The third section deals with the use of EAs to synthesize the Preview Controller. The system models are described in the fourth section. The results and discussions are summed up in the fifth section and the conclusion is drawn in the last section.

## 2. Preliminaries

This section briefly recalls the basics behind the problem. For more details about the basics the references mentioned can be studied.

### 2.1 Preview Control Problem

The term Preview Control is usually associated with a particular class of anticipative control problems with a preview horizon that extends for a fixed time into the future. The field of Preview Control is concerned with using the advance knowledge of disturbances or references in order to improve the tracking quality or disturbance rejection. If the future information of references or disturbances is available, then the performance of transient responses can be improved remarkably. This type of control problem that utilizes the future information on the reference signal and/or disturbances is called Preview Control Problem.

The feedback control system, shown in figure 1, has the control signal dependent on the present error between reference and system output. The Preview Control System has its control signal dependent on the present error between reference signal and system output and the future information available for the reference signal or the disturbance.

The classical solution of Preview Control problem is given using H∞ control and state augmentation, solved using Algebraic Riccati Equation [10].

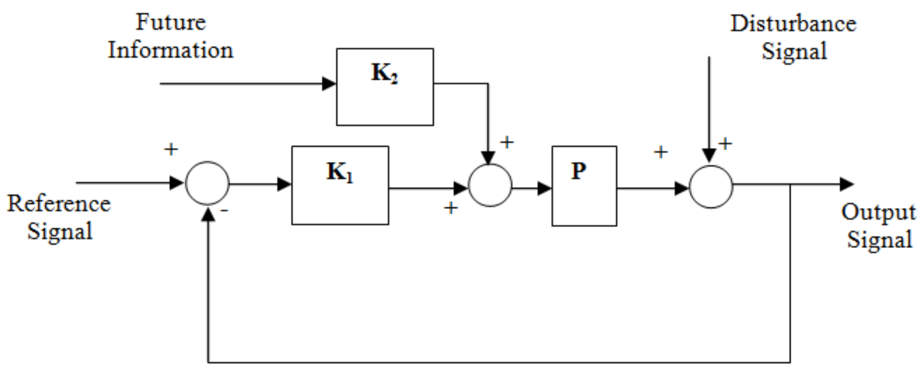


**Figure 1. Preview Control System**

The general discrete-time system is described by

$$x(t+1) = Ax(t) + Bu(t) + Ed(t)$$

$$z(t) = Cx(t) + Du(t) \tag{1}$$

where $x(t) \in R^n$ and $u(t) \in R^m$ are the state vector and control input, respectively. The signal $z(t) \in R^p$ denotes the controlled output or the tracking error. Moreover, $d(t) \in R^l$ denotes the exogenous signal which can be considered as the reference signal or the disturbance.

The purpose of preview control is to design a static controller in the form of

$$u(t) = K_1 x(t) + \sum_{i=0}^{h} K_{2i}\, d(t+i) \tag{2}$$

so that the following quadratic performance index is made satisfactorily small even in the presence of the exogenous input d.

$$J = \|z\|_2^2 = \sum_{i=0}^{\infty} \|z(t)\|^2 \tag{3}$$

The first and second terms on the right-hand-side of the controller equation (2) represent the static state feedback and preview compensation, respectively.

The solution of the Preview Control Problem can be obtained by solving the Discrete Algebraic Ricatti Equation for the augmented system, as explained by K. Takaba in [10].

The solution of Discrete Algebraic Ricccati Equation depends very strongly on the critical parameters chosen, namely, Q (error weighting matrix), R (control weighting matrix) and γ (constraint on H∞ norm). The classical method for choice of these parameters is cumbersome and requires complicated calculations. The improvement in performance can also be obtained by solving the DARE and then modifying the preview gains. The choice of these parameters is critical in case of H∞ Preview Control problem and the problem complicates even more with the increase in the length of preview. An automated solution to both such problems can be the use of evolutionary techniques. The basic methodology of three evolutionary algorithms (Genetic Algorithm, Particle Swarm Optimization and Marriage in Honey Bees Algorithm) is explained in the next section.

### 2.2 Evolutionary Algorithms

EAs share a common approach for their application to a given problem. The problem first requires some representation to suit each method. Then, the evolutionary search algorithm is applied iteratively to arrive at a near-optimum solution.

The first evolutionary-based technique introduced in the literature was the genetic algorithms (GAs) [11]. GAs are procedures for solving problems by using principles inspired by natural population genetics, based on Darwin's Theory of "survival of the fittest". Starting with the random population of solutions, GAs converge to quality solution but do not guarantee convergence to the single best solution to the problem. Based on the demonstrated abilities, GAs have found applications in many disciplines of science and engineering [12-13]. GAs may require long processing time for a near-optimum solution to evolve. Also, solution to all the problems is not possible with GA.

As an attempt to reduce processing time and improve the quality of solutions, particularly to avoid being trapped in local optima, other EAs have been introduced. A few of the developed techniques are particle swarm optimization (PSO) and marriage in honey bees (MBO) algorithm.

The PSO is a high performance optimizer that possesses several highly desirable attributes. It is a population based stochastic optimization technique developed by Eberhart and Kennedy [14-15]. It is inspired by social behaviour of bird flocking. PSO shares many similarities with evolutionary computation techniques such as genetic algorithm. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike genetic algorithms, particle swarm optimization has no evolution operators such as crossover and mutation.

The MBO [17] is a swarm based algorithm that is inspired by the phylogenetic of sociality in Hymenoptera, such as bees, ants and wasps, and the mating process in honey bees. This algorithm is a result of the behaviour of honey-bees which is further a product of their genetic potentiality, ecological and psychological environments, the social conditions of the colony, and various prior and ongoing interactions among these three. Each bee performs sequences of action which unfold according to genetic, environmental and social regulation. The outcome of each action itself becomes a portion of the environment and greatly influences the subsequent actions of both a single bee and her hive mates.

## 3. Synthesis of Preview Controller Using Evolutionary Algorithms

This section studies the use of EAs for the critical parameters tuning and modifying the preview gains obtained from the classical solution using DARE.

### 3.1 Tuning of Critical Parameters

The critical parameters Q, R and $\gamma$ affect the solution obtained from classical methodology (using DARE) very strongly. These are the parameters to be specified by the designer based on the system's requirements but the requirements always specify the range amongst which the values for this triplet can be selected. The selection of the optimal value of this triplet from the 3D space is a cumbersome process if done manually. The automation of this process is proposed using EAs as below:

**Step 1:** Initialize the population of random solutions (Q, R and $\gamma$ triplet) in the 3D space.

**Step 2:** For each solution triplet, solve the DARE then formed and find the preview controller gains. Then, calculate the objective function value i.e. Integral of Absolute Error.

**Step 3:** Based on the objective function value, the solution population is updated using specific modification equations of the EAs and the range of space specified for the new solution triplets.

**Step 4:** The step 2 and step 3 are repeated until the stopping criterion is met, i.e. a fixed number of iterations or a minimum value of objective function is reached.

### 3.2 Tuning of Preview Gains

The requirement of EAs for tuning the gains of Preview Controllers is supported by the fact that there is a scope of improvement in the solution obtained from classical methodology. The classical methods cannot find the best solution as the problem dimension increase with the length of preview. The solution of this multi-dimensional optimization problem using EAs is proposed as below:

**Step 1:** Initialize the population of random solutions i.e. the preview controller gains in the multidimensional space, depending on the length of preview.

**Step 2:** For each solution set, calculate the objective function value i.e. Integral of Absolute Error.

**Step 3:** Based on the objective function value, the solution population is updated using specific modification equations of the EAs and the range of space specified for the new solution sets.

**Step 4:** The step 2 and step 3 are repeated until the stopping criterion is met, i.e. a fixed number of iterations or a minimum value of objective function is reached.

## 4. System Description

The problem of Preview Control System is implemented for two systems Industrial Servo System and Inverted Pendulum System. The description of the system models is given as below:

### 4.1 Industrial Servo System

The state-variable model of the system [10] in discrete domain is given as:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y &= Cx(t) \end{aligned} \tag{7}$$

with

$$A = \begin{bmatrix} 0.9752 & 0.0248 & 0.1983 & 0.0017 \\ 0.0248 & 0.9752 & 0.0017 & 0.1983 \\ -0.2459 & 0.2459 & 0.9752 & 0.0248 \\ 0.2459 & -0.2459 & 0.0248 & 0.9752 \end{bmatrix}; B = \begin{bmatrix} -0.0199 \\ -0.0001 \\ -0.1983 \\ -0.0017 \end{bmatrix}; C = [0 \quad 1 \quad 0 \quad 0]$$

The system model is of fourth order and is discretized with a sampling interval of ts=0.1.

### 4.2 Inverted Pendulum System

The inverted pendulum system [18] is depicted in figure 3 and consists of an inverted pendulum mounted on a motor-driven cart. The inverted pendulum is unstable in that it may fall over any time in any direction unless a suitable control force is applied. Assumption is made that the pendulum mass is concentrated at the end of the rod (the rod is mass less) and the control force is applied to the cart. The system has one input i.e. control force u and two outputs, namely, pendulum angle θ and cart position x.
The goal for the system is to keep the pendulum upright, as much as possible and yet control the position of the cart.

With reference to the figure 2, applying Newton's second law to the system motion results in the following dynamics of the system:

$M\ddot{x} + mg\theta = u$

$$Ml\theta - (M + \ddot{m})g\theta = -u \qquad (8)$$

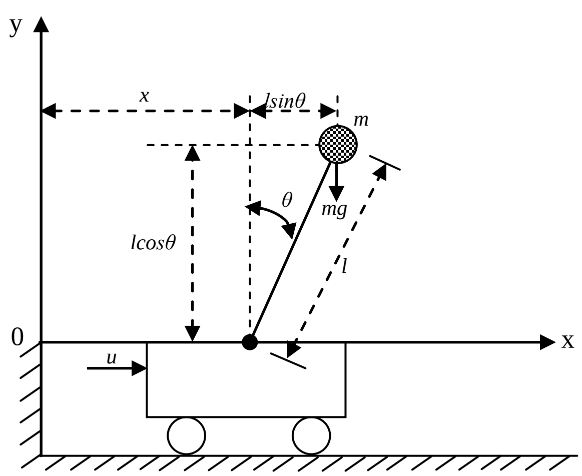


**Figure 2. Inverted Pendulum Model**

The state variables of the system are defined by: $x_1 = \theta, x_2 = \dot{\theta}, x_3 = x, x_4 = \dot{x}$, where angle θ indicates the rotation of pendulum about point P and x is the location of the cart. From these definitions of the variables and above equations the system model is defined as below:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ \dfrac{M+m}{Ml}g & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ -\dfrac{m}{M}g & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0\\ -\dfrac{1}{Ml}\\ 0\\ \dfrac{1}{M}\end{bmatrix}u$$

$$y = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} \qquad (9)$$

The discrete state space model of the system described by above equation is obtained with a sampling interval of $t_s = 0.1$.

The state-variable model of the system in discrete domain can be modelled with $m = 0.1\ kg, M = 2\ kg, l = 0.5\ m\ and\ g = 9.81\ m/s^2$, the system matrices are obtained as below:

$$A = \begin{bmatrix} 1.1048 & 0.1035 & 0 & 0 \\ 2.1316 & 1.1048 & 0 & 0 \\ -0.0025 & -0.0001 & 1 & 0.1 \\ -0.0508 & -0.0025 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} -0.0051 \\ -0.1035 \\ 0.0025 \\ 0.0501 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix}$$

## 5. Results and Discussion

The performances of the intelligent algorithm can be described typically with the following aspects. The first is the convergence to the optimum point, which is a basic performance of the intelligent algorithm. The second is the accuracy of the optimum value. The third is the number of objective calculations. The last one is the computation time to find the optimum value, through which the simplicity of the algorithm can be inferred.

Each algorithm has its own parameters that affect its performance in terms of solution quality and processing time. To obtain the most suitable parameter values that suit the test problems, a large number of experiments were conducted. For each algorithm, an initial setting of the parameters was established using values previously reported in the literature. Then, the parameter values were changed one by one and the results were monitored in terms of the solution quality and speed. The final parameter values adopted for each of the three EAs are listed in Table 1.

**Table 1: Parameters Adopted for Evolutionary Algorithms**

| **Genetic Algorithm (GA)** | |
|---|---|
| Population Size | 40 |
| Crossover Probability | 0.8 |
| Mutation Probability | 0.08 |
| Iterations | 100 |
| **Particle Swarm Optimization (PSO)** | |
| Population Size | 40 |
| Initial Inertia Weight | 0.9 |
| Final Inertia Weight | 0.4 |
| Iterations | 100 |
| **Marriage in Honey-Bees (MBO) Algorithm** | |
| Number of Queens | 10 |
| Number of Drones | 20 |
| Number of Broods | 20 |
| Spermetheca Size | 20 |
| Number of Mating Flights | 10 |

The analysis is done for the use of Evolutionary Algorithms for tuning critical parameter values (Q, R and γ) and Preview Gains. The results are summed up as below:

### 5.1 Tuning of Q, R and γ Values

For the Industrial Servo System, the weighting matrices $Q$ and $R$ were chosen as $Q = 1$ and $R = 2$. The value of $\gamma$ was chosen to be 22.

The transient response of the system for classically determined parameters and parameters tuned using EAs is shown in figure 3; while the convergence characteristics of the EAs studied with best objective function values is shown in figures 4.

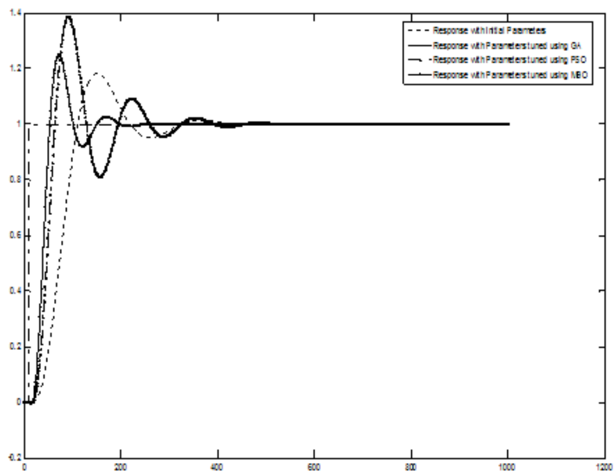


**Figure 3. Transient Response of Industrial Servo System for [Q R γ] Tuning**

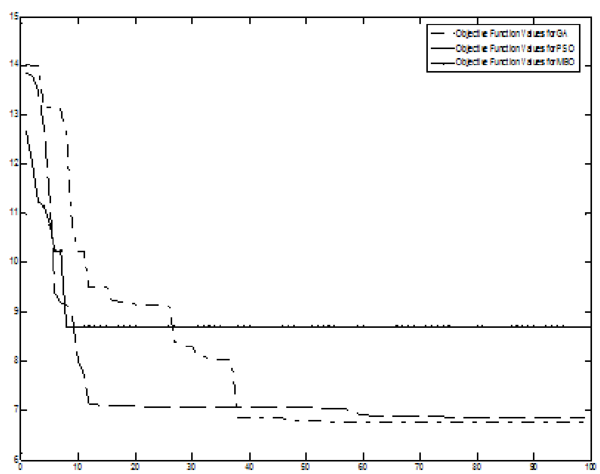


**Figure 4. Convergence Characteristics for Industrial Servo System for [Q R γ] Tuning**

For Inverted Pendulum System, the weighting matrices Q and R and parameter γ were chosen as:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}, R = 0.1, \gamma = 100$$

The transient response of the Inverted Pendulum System for classically obtained parameters and parameters tuned using EAs is shown in figure 5; while the convergence characteristics of the EAs studied with best objective function values is shown in figures 6.

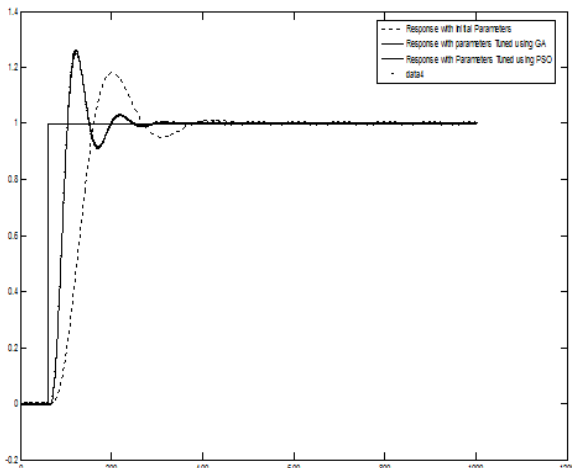


**Figure 5. Transient Response of Inverted Pendulum System for [Q R γ] Tuning**

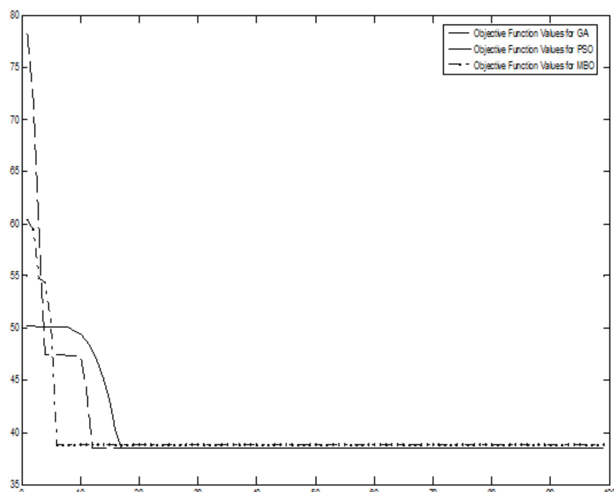


**Figure 6. Convergence Characteristics for Inverted Pendulum System for [Q R γ] Tuning**

A comparison of the processing time variation and transient characteristics of the two systems tuned using EAs is summed up in table II and III.

### 5.2 Tuning of Preview Gains

For the Industrial Servo System and Inverted Pendulum System, defined in equation (7) and (9), analysis is made to study the variation of characteristics of EAs with the Preview Length. The variation of the processing time of the EAs with respect to preview length are shown in figure 7 and 8; figure 9 and 10 show the variation of objective function values with preview length. A comparative evaluation of the EAs with preview length is presented in table 2 and 3.

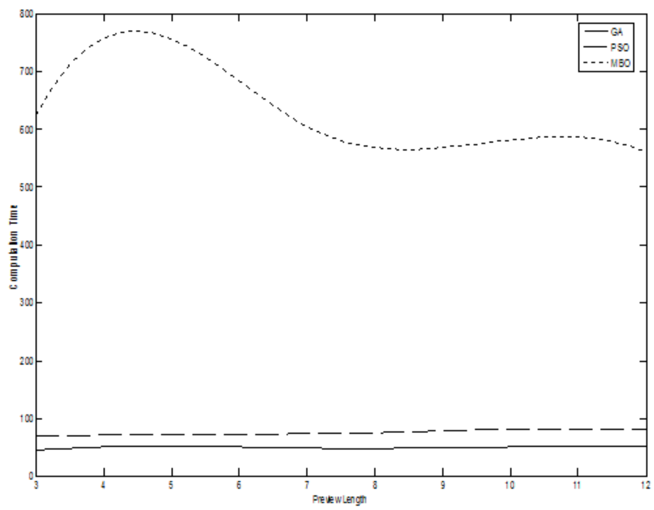


**Figure 7. Variation of Processing Time with Preview Length for Industrial Servo System**

**Table 2. Processing Time of Evolutionary Algorithms**

| System | Preview Length | GA | PSO | MBO |
|---|---|---|---|---|
| Industrial Servo System | h = 0 ([Q R γ] tuning) | 50.7810 | 71.767 | 774.165 |
| | h =3 | 45.3110 | 68.766 | 624.292 |
| | h =5 | 51.844 | 72.2 | 756.039 |
| | h =7 | 48.992 | 73.065 | 604.173 |
| | h =10 | 50.535 | 81.079 | 581.217 |
| | h =12 | 50.426 | 80.417 | 560.374 |

| | | | | |
|---|---|---|---|---|
| Inverted Pendulum System | h = 0 ([Q R γ] tuning) | 50.7810 | 71.767 | 774.165 |
| | h =6 | 65.027 | 97.221 | 521.694 |
| | h =7 | 65.795 | 99.251 | 512.721 |
| | h =8 | 67.109 | 98.414 | 507.683 |
| | h =9 | 66.408 | 99.812 | 507.932 |
| | h =10 | 66.64 | 106.637 | 507.024 |
| | h =11 | 66.324 | 99.051 | 507.716 |

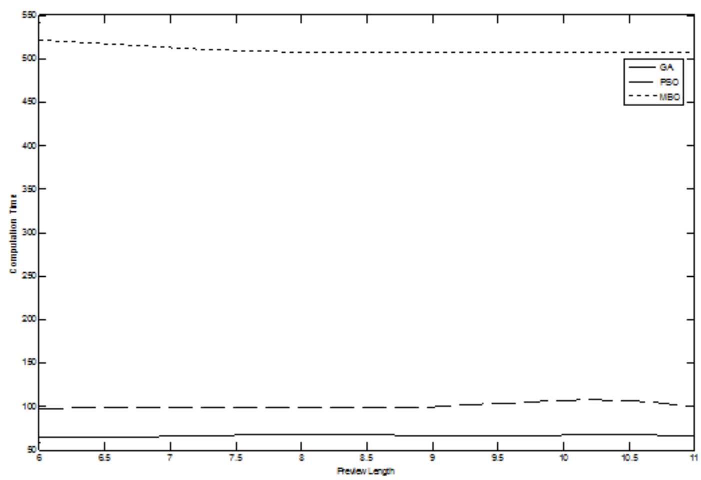


**Figure 8. Variation of Processing Time with Preview Length for Inverted Pendulum System**

**Table 3. Objective Function (IAE) Values of Evolutionary Algorithms**

| System | Preview Length | Classical Solution | GA | PSO | MBO |
|---|---|---|---|---|---|
| Industrial Servo System | h = 0 ([Q R γ] tuning) | 14.6019 | 6.7674 | 6.8573 | 8.698 |
| | h =3 | 4.3859 | 3.9903 | 3.9903 | 4.003 |
| | h =5 | 3.0728 | 2.5609 | 2.404 | 2.8692 |
| | h =7 | 2.4314 | 1.7113 | 1.637 | 2.3969 |
| | h =10 | 2.4025 | 1.7093 | 1.5669 | 2.4025 |
| | h =12 | 2.3040 | 1.7702 | 1.56 | 2.304 |

| | | | | | |
|---|---|---|---|---|---|
| Inverted Pendulum System | h = 0 ([Q R γ] tuning) | 78.2496 | 38.4752 | 38.4775 | 38.4752 |
| | h =6 | 9.3725 | 2.8211 | 2.9462 | 12.717 |
| | h =7 | 3.8341 | 1.6983 | 1.8421 | 6.745 |
| | h =8 | 2.8256 | 1.4668 | 1.3839 | 13.5366 |
| | h =9 | 2.2904 | 1.464 | 1.3947 | 12.2494 |
| | h =10 | 2.0379 | 1.1627 | 1.3306 | 24.9142 |
| | h =11 | 1.9507 | 1.4474 | 1.2568 | 21.4296 |

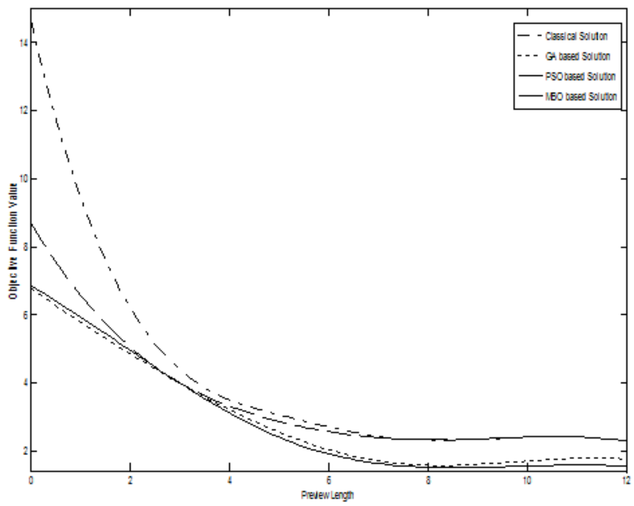


**Figure 9. Variation of Objective Function Value with Preview Length for Industrial Servo System**

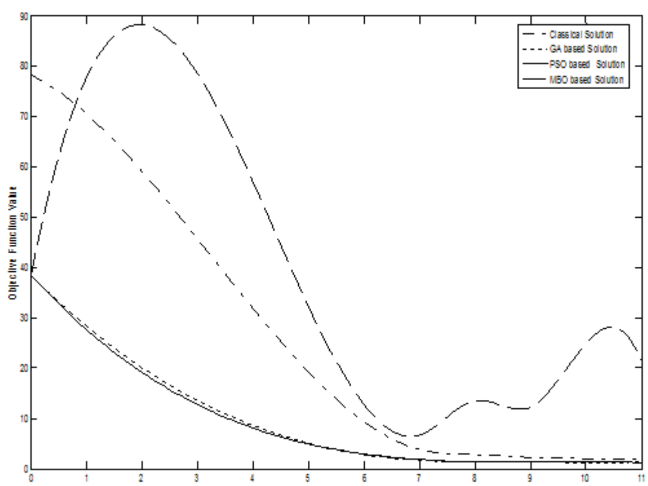


**Figure 10. Variation of Objective Function Value with Preview Length for Inverted Pendulum System**

The results show that for small search spaces, all the three algorithms produce optimized and comparable results, with the computation time being maximum for MBO algorithm.

With the increase in number of parameters to be tuned, PSO algorithm performs the best. The GA obtains comparable results to PSO algorithm but the performance degrades with the increase in number of parameters to be tuned. The processing time and number of iterations required to converge to optimized solution increases steeply with the increase in number of parameters to be tuned for MBO algorithm.

The processing time of the algorithm increases with the complexity of the system and the number of parameters to be tuned. The MBO algorithm takes the maximum time to complete a fixed number of iterations. A comparison of the result for variation with Preview Length shows that the processing time of GA and PSO algorithms is almost same and has a very less variation with the number of parameters to be tuned. The processing time for MBO algorithm is more than GA and PSO algorithm, has large initial value but stabilizes for increased number of parameters.

The variation of Objective Function Values with Preview Length reveals that all the three EAs produce similar results for smaller number of parameters to be tuned for a linear system but as the number of parameters to be tuned increase or the non-linear system, the GA and PSO algorithm take the lead due to their fast computation and produce better results.

## 6. Conclusion

This paper presents an evaluation of automated solution to Preview Control problem. The automated solution is based on evolutionary algorithms: Genetic Algorithm, Particle Swarm Optimization and Marriage in Honey Bee Algorithm that are tested for the bench-mark control problems: Industrial Servo and Inverted Pendulum System. The performance of the procedures is assessed in terms of processing time, convergence speed and quality of solution.

The results show that for small search spaces the optimized parameters obtain comparable values from all the three algorithms. For small search space, the MBO algorithm takes the maximum time to converge to the optimized results. The performance of GA degrades with the increase in the number of parameters to be tuned and PSO algorithm outperforms the other two algorithms in the optimization for large search spaces. The MBO algorithm needs more number of iterations to converge to optimal solution in large search space, hence, increasing its processing time steeply.

Thus, the PSO algorithm produces the best results for both small and large search spaces of Preview Control based problems. The PSO algorithm has least number of initial parameters to be defined that adds to its advantage. Also, it achieves faster convergence in multi-dimensional spaces due to simple calculations for next iteration.

## References

[1] Kojima and S. Ishijima, “H∞ Control with Preview Compensation”, Proceedings of American Control Conference, New Mexico, pp. 1692-1697, **(June 1997)**

[2] Kojima and S. Ishijima, “H∞ Preview Tracking in Output Feedback Setting”, Proceedings of Conference on Decision and Control, Arizona, USA, pp. 3162-3164, **(December 1999)**

[3] G. Tadmor and L. Mirkinn, “H∞ Preview Control and Fixed- Lag Smoothing I: Matrix ARE Solutions in Continuous-Time Systems”, Proceedings of IEEE Conference on Decision and Control, Hawaii, USA, pp. 6515-6520, **(December 2003)**

[4] G. Tadmor and L. Mirkinn, “H∞ Preview Control and Fixed- Lag Smoothing II: Matrix ARE Solutions in Discrete-Time Systems”, Proceedings of IEEE Conference on Decision and Control, Hawaii, USA, pp. 6521-6526, **(December 2003)**

[5] K. Y. Polyakov, E. N. Rosenwasser and B. P. Lampe, "Optimal Open-Loop Tracking using Sampled Data System with Preview", IEEE.

[6] Y. Kuroiwa and H. Kimura, "H∞ Control with Preview and Delay", Proceedings of American Control Conference, Boston, pp. 4794-4799, **(June 2004)**

[7] M. M. Negm, J. M. Bakhashwain and M. H. Shwehdi, "Speed Control of Three Phase Induction Motor based on Robust Optimal Preview Control Theory", IEEE Transactions on Energy Conversion, Vol. 21, No. 1, pp. 77-84, **(March 2006)**

[8] Moran, Y. Mikami and M. Hayase, "Analysis and Design of H∞ Preview Tracking Control Systems", AMC'96- MIE, pp. 482-487, **(1996)**

[9] Farooq and D. J. N. Limebeer, "Path Following of Optimal Trajectories using Preview Control", Proceedings of IEEE Conference on Decision and Control and the European Control Conference, Spain, pp. 2787-2792, **(December 2005)**

[10] K. Takaba, "A Tutorial on Preview Control Systems", SICE Annual Conference, Fukui, **(August 2003)**, pp. 1388-1393.

[11] Holland, J., "Adaptation in Natural and Artificial Systems", Ann Arbor, MI: University of Michigan Press, **(1975)**

[12] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Pearson Education, Delhi, Seven Indian Reprint, **(2004)**

[13] M. Mitchell, "An Introduction to Genetic Algorithms", P.H.I., New Delhi, **(2004)**

[14] R. Eberhart and J. Kennedy, "A New Optimizer using Particle Swarm Theory", Proc. 6th Int. Symp. Micro Machine and Human Science, **(1995)**, pp. 39-43.

[15] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proc. IEEE Int. Conf. Neural Network, vol. IV, Perth, Australia, pp. 1942-1948, **(1995)**

[16] Y. Shi and R. C. Eberhart, "Empirical Study of Particle Swarm Optimization", IEEE, pp. 1945-1950, **(1999)**

[17] H.A. Abbass, "MBO: Marriage in Honey Bees Optimization A Haplometrosis Polygynous Swarming Approach", IEEE, **(2001)**, pp. 207-214.

[18] M. Gopal, "Digital Control and State Variable Methods", TMH New Delhi, Second Reprint, **(1997)**

[19] G.A. Medrano-Cedra, "Robust Computer Control of an Inverted Pendulum", IEEE Control Systems Magazine, pp. 58-67, **(1999)**

[20] Al-Jarrah and A. Fahim, "New Robust Controller Design for Multivariable Control Systems", IEEE, **(2008)**

## Appendix A. Pseudocode for a GA Procedure

```
Begin;
Generate random population of P solutions (chromosomes);
For each individual i∈P: calculate fitness (i);
For i=1 to number of generations;
Randomly select an operation (crossover or
mutation);
If crossover;
Select two parents at random ia and ib;
Generate an offspring ic=crossover (ia and ib);
Else If mutation;
Select one chromosome i at random;
Generate an offspring ic=mutate (i);
End if;
Calculate the fitness of the offspring ic;
If ic is better than the worst chromosome then
replace the worst chromosome by ic;
Next i;
Check if termination=true;
End;
```

## Appendix B. Pseudocode for a PSO Procedure

Begin;
Generate random population of N solutions (particles);
For each individual $i \in N$: calculate fitness (i);
Initialize the value of the weight factor, w;
For each particle;
Set pBest as the best position of particle i;
If fitness (i) is better than pBest;
pBest(i)=fitness (i);
End;
Set gBest as the best fitness of all particles;
For each particle;
Calculate particle velocity according to Eq. (1-a);
Update particle position according to Eq. (1-b);
End;
Update the value of the weight factor, w;
Check if termination=true;
End;

## Appendix C. Pseudocode for a MBO Procedure

Begin;
Initialize Workers;
Generate random population of Q solutions (queens);
For a pre-defined maximum number of mating flights;
For each individual $i \in Q$
Initialize energy, speed and position;
Generate random population of D drones;
Probabilistically choose a drone;
If a drone is selected;
Add its sperm to queen's spermatheca;
End;
Update the queen's energy and speed;
End;
Generate broods by crossover and mutation;
Use workers to improve the broods;
Update workers' fitness.
While the best brood is better than the worst queen;
Replace the least-fittest queen with the best brood;
Remove the best brood from the brood list;
End;
End;
End;

## Authors

**Dr. Akhilesh Swarup** received his Ph.D. in 1993 from the Indian Institute of Technology (IIT), New Delhi, India. He is currently working as Professor in the Department of Electrical Engineering at the National Institute of Technology (N.I.T), Kurukshetra, India. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). His research interests include robotics and artificial intelligence, system identification, computer networking and control systems.

**Nidhika Birla** had received the degree Masters in Technology in the field of Electrical Engineering with specialization "Control Systems" from National Institute of Technology, Kurukshetra (India) in 2006 and now, she is pursuing research in the field of Intelligent Control from National Institute of Technology, Kurukshetra (India). Her fields of interest lie in Soft Computing, Intelligent Control Techniques, Fuzzy Logic and Self-Organizing Systems.