NOVEL APPROACH ON ADAPTIVE CONTROL OF NONLINEAR SYSTEMS USING PSO & FFA

DISSERTATION REPORT

Submitted in Partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

IN

CONTROL AND INSTRUMENTATION

Submitted by:

SARATH S. PILLAI
(2K11/C&I/11)

Under the esteemed guidance of

DR. BHARAT BHUSHAN
Associate Professor
Dept. of Electrical Engineering

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
DELHI-110042
2011-2013

DELHI TECHNOLOGICAL UNIVERSITY

Department of Electrical Engineering

CERTIFICATE

This is to certify that the dissertation titled "**Novel Approach on Adaptive Control of Nonlinear Systems using PSO & FFA**" submitted in partial fulfillment of the requirement for the award of the degree of Master of Technology in Control and Instrumentation (C&I) by **Sarath S. Pillai (Roll No: 2K11/C&I/11)** is a bonafide record of the candidate's own work carried out by him under my supervision and guidance.

This work has not been submitted earlier in any university or institute for the award of any degree to the best of my knowledge.

Dr. BHARAT BHUSHAN
Associate Professor
Department of Electrical Engineering
Delhi Technological University

PROF. MADHUSUDAN SINGH
HOD
Department of Electrical Engineering
Delhi Technological University

ACKNOWLEDGEMENT

The elation of this dissertation would be incomplete without mentioning the people who helped me make it possible, whose encouragement, support and inspiration is much valuable.

I sincerely acknowledge the earnestness and the patronage of my guide **Dr. Bharat Bhushan**, Associate Professor, Department of Electrical Engineering, Delhi Technological University, New Delhi, for his valuable guidance, support and motivation throughout this project work. The valuable hours of discussion and suggestion that I had with him have undoubtedly helped me in supplementing my thoughts in the right direction for attaining the desired objective.

I wish to express my gratefulness to **Prof. Madhusudan Singh**, HOD, Department of Electrical Engineering, Delhi Technological University, New Delhi, for providing the necessary lab facilities. And I wish to thank all faculty members whoever helped to finish my thesis in all aspects.

I am indebted to my dear **parents** for their continuous love and support that helped me achieve my target. I am also grateful to all those who are directly or indirectly a part of this work.

The dissertation is built upon the help of many people and this successful completion of the work is largely due to the suggestions, comments and guidance of my friends **Sangeeta Devra**, **Pramila** and **Rohit Goyal**.

Above all, thanks to the **Almighty** for blessing and guiding me throughout my life.

Sarath S. Pillai

2K11/C&I/11

ABSTRACT

With the help of evolutionary concepts and behaviour of biotic components of nature, many optimization algorithms were developed. Optimization techniques like Particle Swarm Optimization and Firefly Algorithms are among the latest research topics. Several advancements have also been made in these algorithms.

A large amount of research has been done to solve large scale nonlinear optimization problems. An effective solution in this regard could be the use of Particle Swarm Optimization (PSO) and Firefly Algorithm (FFA). This dissertation presents application of PSO and FFA on indirect adaptive control of nonlinear systems for liquid level control in surge tank system and inverted pendulum system. The model of the systems has been derived and indirect adaptive control technique has also been explained in detail with a tilt towards its implementation using PSO and FFA techniques. A comparative analysis has been made on both the systems separately using both PSO and FFA adaptive control and the simulation results have been discussed extensively.

TABLE OF CONTENTS

Certificate	***i***
Acknowledgement	***ii***
Abstract	***iii***
Table of Contents	***iv***
List of Figures	***vi***
List of Tables	***vii***
Chapter 1. Introduction	**1**
Chapter 2. Review of PSO and FFA Algorithms	**5**
2.1 Particle Swarm Optimization (PSO)	5
2.1.1 PSO Operation	5
2.1.2 Pseudocode for PSO	7
2.2 Firefly Algorithm (FFA)	9
2.2.1 FFA Operation	9
2.2.2 Pseudocode for FFA	10
Chapter 3. Adaptive Control Implementation using PSO and FFA	**12**
3.1 Direct Adaptive Control	13
3.2 Indirect Adaptive Control	14
3.3 Implementation of PSO and FFA as controllers	16

3.3.1 Estimating an Unknown Ideal Controller 17

3.3.2 Certainty Equivalence Controller 17

3.3.3 Error Equation Representation 18

3.3.4 Adaptation Method for Cost Function 18

Chapter 4. Dynamics of Nonlinear Systems 24

4.1 Level Control in a Nonlinear Surge Tank System 24

4.2 Pole Angle Control in an Inverted Pendulum System 27

Chapter 5. Simulation Results and Discussions 34

5.1 Indirect Adaptive Control of Liquid Level in a Surge Tank 34

5.2 Indirect Adaptive Control of Pole Angle in an Inverted Pendulum 39

Chapter 6. Conclusions and Further Scope of Work 48

References 51

LIST OF FIGURES

FIGURES	***PAGE NO.***
2.1 Position (x) and Velocity (y) update of i^{th} particle in swarm	6
2.2 Flowchart for Particle Swarm Optimization	8
2.3 Flowchart for Firefly Algorithm	11
3.1 Direct Adaptive Control	13
3.2 Indirect Adaptive Control	15
3.3 Adaptive Control Using PSO or FFA	20
4.1 Surge Tank System	24
4.2 Inverted Pendulum on Cart	27
4.3 Mathematical Model of Inverted Pendulum on Cart	28
5.1 Reference Input for the surge tank	35
5.2 Liquid level ‘h’ and the reference input for PSO and FFA	35
5.3 Cost function graph for both PSO and FFA for surge tank	36
5.4 Estimates of the liquid level ‘h’ for both PSO and FFA	37
5.5 Error between the liquid level and the reference trajectory for PSO and FFA	38
5.6 Angle of the pole w.r.t. vertical axis for both PSO and FFA	40
5.7 Angular velocity of the pole for both PSO and FFA	41
5.8 Position of the cart for both PSO and FFA	42
5.9 Velocity of the cart for both PSO and FFA	43
5.10 Error between the pole angle obtained and the reference for both PSO and FFA	44
5.11 Error between the cart position obtained and the reference for both PSO and FFA	45
5.12 Cost function graph for both PSO and FFA for inverted pendulum	46

LIST OF TABLES

TABLES ***PAGE NO.***

5.1 Elapsed time and Minimum Cost for the Surge tank level system 38

5.2 Elapsed time and Minimum Cost for the inverted pendulum system 47

CHAPTER 1

INTRODUCTION

Learning from naturally abundant biological systems and structures to design and develop a number of different kinds of optimization algorithms have been widely used in both theoretical study and practical applications. The evolution is used to tune the algorithm parameters [3] while the study of bionics inspires and improves our modern technologies with the principles found in nature, biological structures and functions. Bionics study inspires us not only with its physical yields, but also with various computation methods that can be applied in different areas. In engineering, this ‘‘bioinspired’’ design approach has been used to exploit the evolved ‘‘tricks’’ of nature to design robust high performance technological solutions. Amongst them, one of the most popular bioinspired design approaches is what is called ‘‘Swarm Intelligence’’ [4]. Biologically inspired optimization algorithms can efficiently deal with non-linear optimization problems.

These techniques which are inspired by the collective behavior of animal societies as well as other social insect colonies that are naturally able to solve large-scale distributed problems are grouped in the Swarm Intelligence [5]. The swarm intelligence is defined as an algorithm or distributed problem-solving devices designed inspired by the collective behavior of many animal societies, particularly social insect colonies [6]. In current peer-to-peer systems swarm availability is proven to be a serious issue [7]. It has become a research interest among scientists of related fields in recent years [8].

Particle Swarm Optimization (PSO) technique was developed by Eberhart and Kennedy [9] and is an approach that mimics the behavior of social organisms where the behavior of different types of social interactions (e.g., flock of birds) is mimicked in order to create an optimization method that is able to solve continuous optimization problems [4]. PSO is a computational intelligence-based technique that is not largely affected by the size and nonlinearity of the problem thus converges to the optimal solution in many problems where most analytical methods fail to converge. Therefore, it can be effectively applied to different optimization problems in power systems [10] and [11]. Literature tells that it has been successfully applied to various problems viz. Economic Dispatch [12], Generation Expansion Problem (GEP), State estimation etc. It has been again improved [13] upon to get, Modified PSO [15], Adaptive Particle Swarm Optimization (APSO) [16], Multi-dimensional Particle Swarm Optimization for Dynamic Environments [17] and [18], Self-Organizing Swarm (SOSwarm) for Financial Credit-Risk Assessment [19] and has been applied for PID control [20], Neural Network [21] and to train the PROAFTN [22], Multi-Criteria Decision Aid (MCDA) method etc. A valuable point of comparison can be made which can be used throughout the research areas to better test new advances while having a strictly-defined, well-known, standard algorithm [23].

Mechanisms of firefly communication have been implemented effectively in various areas of wireless networks design [24], dynamic market pricing [25] and mobile robotics [26] and [27] via luminescent flashes and their synchronization [28]. Many bio-inspired algorithm exist that use local information and simple rules to create a global agreement of a single parameter, such as clock time in the flashing firefly algorithm, or location in the slime algorithm. One of the most important techniques for performing image segmentation is Thresholding [29]. It is generally simple and computationally efficient. The segmentation results of the algorithm are promising which inspires and encourages further researches to apply this algorithm to real-time and complex image analysis problems [10] such as target recognition, complex document analysis and as biomedical image application. Other application includes synchronization of wireless network [31]. An application in pulse-coupled oscillator's model addresses the problem of synchronization of the oscillators with different frequencies. Firefly algorithm has been applied in real time systems [32]-[33] and heartbeat synchronization techniques [34].

The research here aims at adaptive control of nonlinear systems using Particle Swarm Optimization and Firefly Algorithm. Adaptive Control is used by a controller which must adapt to a controlled system with parameters which are initially uncertain or vary. A rocket could be an apt example because when it flies its mass will slowly decrease as a result of fuel consumption, thus a control law is required that adapts itself to such changing conditions.

Adaptive control differs from robust control in that it does not need a priori information about the bounds on these time-varying or uncertain parameters. While adaptive control is concerned with control law changing themselves, robust control guarantees that if the changes are within given bounds the control law need not be changed.

The foundation of adaptive control is parameter estimation [1]. Common methods of estimation include gradient descent and recursive least squares. Both of these methods provide update laws which are used to modify estimates in real time (i.e., as the system operates).

Here, the controller will succeed in controlling the plant if the controller designer can specify a controller for each set of plant parameter estimates. In adjusting the nonlinear mapping of the controller to match the unknown nonlinear mapping of the plant an online function approximation problem is solved.

For such online model tuning generally gradient methods are used but in this analysis use of Particle Swarm Optimization Algorithm and Firefly Algorithm is done which is by far much more efficient than the gradient methods in optimization.

Here implementation of indirect adaptive control using PSO and FFA to control two nonlinear systems is carried out:

- Level Control in a Nonlinear Surge Tank system
- Nonlinear Inverted Pendulum system

An introduction to the systems along with the objectives to be controlled is included in later chapters.

This dissertation comprises of six chapters. While the present ongoing introduction and literature review is included in chapter 1, a review of both the PSO and FFA algorithm is given in the second chapter which will entail the preliminary basics of each algorithm alongwith their working principles.

Chapter 3 gives a basic insight to the adaptive control techniques. Emphasis on indirect adaptive control technique is given in the chapter and a description on its implementation with the use of Particle Swarm Optimization and Firefly Algorithm as estimating and control algorithms is dealt with.

Chapter 4 describes the nonlinear systems used to analyze the behavior of indirect adaptive control utilizing PSO and FFA techniques. One of the nonlinear system is a surge tank where the level of the liquid inside the tank is to be controlled while the other one is an inverted pendulum system where the target is to set the pendulum rod to an upright position.

Chapter 5 shows the simulated results and discussions on them pertaining to each nonlinear system on implementing indirect adaptive control using PSO and FFA separately. Results have been analyzed on a comparative basis for both PSO and FFA implemented on each system.

Chapter 6 concludes the dissertation with discussions on the performance of both the algorithms individually and comparatively.

CHAPTER 2

REVIEW OF PSO AND FFA ALGORITHMS

2.1 Particle Swarm Optimization (PSO)

PSO initiates a swarm of particles to move in a search space of possible solutions for a problem. Each particle has a position vector representing a candidate solution to the problem and a velocity vector and also contains a small memory that stores its own best position seen so far and a global best position obtained by communicating with the particles at the neighborhood. The advancement towards best location (xi*) and global best (g*) by the particle swarm optimizer is ideologically similar to the crossover operation utilized by genetic algorithms [2]. Similar to all evolutionary computation paradigms, it uses the concept of fitness. The swarm works through the interactions of members of the population, even though the exact methods for moving the particles are quite flexible [35].

By assigning shortest path and by studying trajectories of particles [34] the optimization behavior of standard PSO can be made invariant to the rotations of the optimization function [7]. There are two major components in the movement of a swarming particle: a deterministic component and a stochastic component.

2.1.1 PSO Operation

Modification in the initial simulations was done to incorporate acceleration by distance and multidimensional search, nearest-neighbor velocity matching and eliminate ancillary variables [9]. Through a series of trial and error, a number of parameters extrinsic to optimization were eliminated from the algorithm which resulted in the very simple original implementation.

Consider flying through the parameter space swarm of particles searching for optimum. Each particle is attributed with position vector $x_i(t)$ and velocity vector $y_i(t)$.

The velocity vector is calculated by the following:

$$V_i^{t+1} = V_i^t + C_1 e_1 \Theta [g^* - x_i^t] + C_2 e_2 \Theta [x_i^* - x_i^t] \quad (2.1)$$

where e_1 and e_2 are two random vectors taking the values between 0 and 1. C_1 and C_2 are the learning parameters or acceleration constants reflecting the weighting of stochastic acceleration terms, which can typically be taken as $C_1 \approx C_2 \approx 2$ and g* is the fitness value.

The initial velocity of a particle can be taken as zero, that is, $V_i^{t=0} = 0$. The new position can then be calculated by:

$$x_i(t+1) = x_i(t) + V_i^{t+1} \quad (2.2)$$

Although V_i could be any value, usually it is bounded in some range [0, V_{max}].

The Hadamard product of two matrices u Θ v is defined as the entry wise product, that is $[u \Theta v]_{ij} = u_{ij}v_{ij}$. Figure 2.1 gives the update of a particle in swarm at the next time instant where g(t) is the best value of the particle fitness.

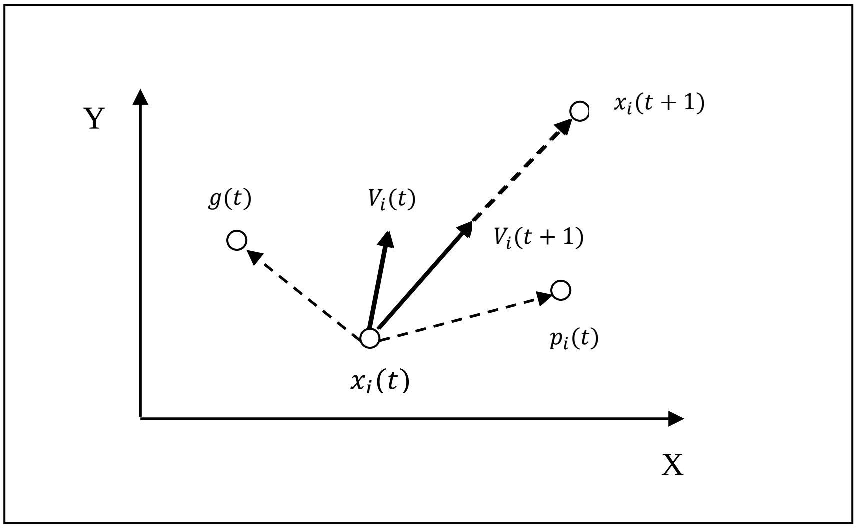

Fig.2.1: Position (x) and Velocity (y) update of i[th] particle in swarm

The velocity update equation in (2.1) has three major components:

1) The first component is referred to as momentum, inertia or habit.

2) The second component is a linear attractiveness towards the best position during travel ever found by the given particle scaled by a random weight, referred to as self-knowledge, remembrance, memory or nostalgia.

3) The third component of the equation is a linear attraction towards the best position found by any particle scaled by another random weight, referred to as group knowledge, cooperation, shared information, social knowledge.

2.1.2 Pseudocode for PSO

For implementation of PSO algorithm, following procedure can be used

1) Initialize the particles by assigning a random position to each particle in the problem hyperspace.

2) Evaluate the fitness function for every particle.

3) Compare the particle's fitness value with its x_i* for every particle. If the value at present is better than the x_i* value, then set this value as the x_i* and the current particle's position x_i.

4) The particle that has the best fitness value is identified. The value of its fitness function is identified as g* and its position as g.

5) Calculate the velocities and positions of all the particles using (1) and (2).

6) Repeat steps 2–5 till a stopping criterion is reached (e.g. a sufficiently good fitness value or maximum number of iterations).

PSO is attractive for the reason that there are few parameters to adjust to get a proper response and thus it can be used for specific applications focused on specific requirement or it can also be used for approaches for wide range of applications.

Figure 2.2 below describes the flowchart for the PSO algorithm for ease of understanding.

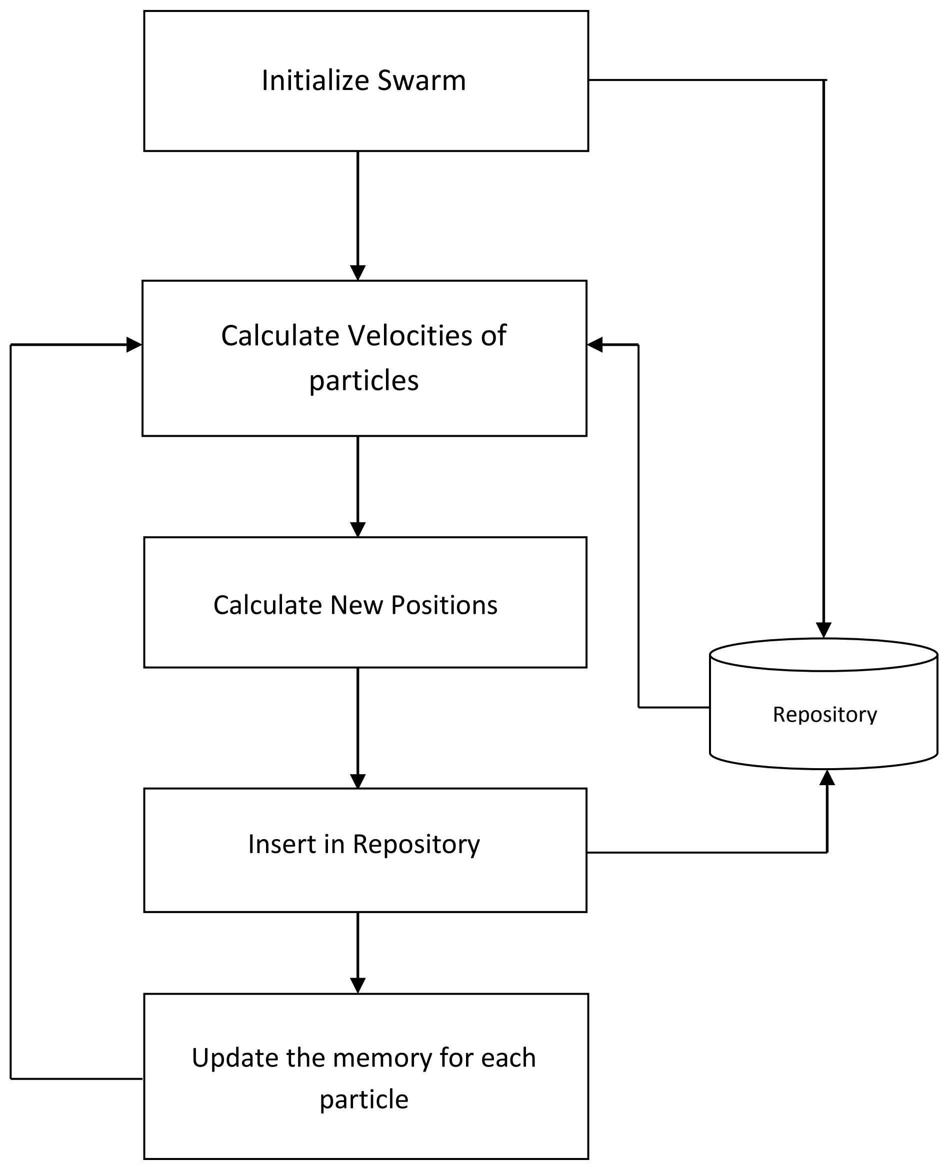

Fig. 2.2 Flowchart for Particle Swarm Optimization

2.2 Firefly Algorithm (FFA)

Fireflies produce luminescent flashes as a signal system to communicate with other fireflies, especially to prey attractions. FFA is inspired by the firefly's biochemical and social aspects. The flashing light is produced by a process called bioluminescence.

2.2.1 FFA Operation

The rhythmic flash, the rate of flashing and the amount of time constitutes the part of the signal system that brings both sexes together. Light intensity obeys inverse square law at a particular distance r from the light source.

$$\mathrm{I}o \propto {}^{1}/_{r^2} \tag{2.3}$$

Thus, the light intensity I_o decreases with the distance r, which makes most fireflies visual to a limited distance.

Following are the assumptions made in the firefly algorithm:

- All fireflies will be attracted to every other firefly regardless of their sex.
- The attractiveness and brightness decrease as the distance increase and are also proportional to each other. The less bright will be moving towards the brighter one. It will move randomly if there is no brighter one.
- The brightness of a firefly is determined or affected by the shape of the objective function.

In the Firefly Algorithm, there are two important issues: formulation of the attractiveness and the variation of light intensity. It is assumed that the attractiveness of a firefly is determined by its brightness which is associated with the encoded objective function.

However, the attractiveness β is relative, it should be judged by the other fireflies or seen in the eyes of the beholder. Thus, intensity will vary with the distance r_{ij} between firefly i and firefly j. Adding to that, light intensity decreases with the distance from its source, and it is also absorbed in the media, so the attractiveness is allowed to vary with the degree of absorption.

In the simplest case particular location x can be $I(x) \propto f(x)$ however attractiveness β is relative. The light intensity I varies according to the inverse square law given by

$$I \propto \frac{I_s}{r^2}$$

where *Is* is the intensity at the source. The light intensity *I* varies with the distance r for a given medium with a fixed light absorption coefficient γ as given by

$$I = I_o e^{-\gamma r}$$

where I_o is the original light intensity. To avoid the singularity at r=0 in the expression, combined effect of both the laws can be approximated as:

$$I(r) = \mathrm{I_0 e^{-\gamma r^2}} \tag{2.4}$$

Attraction between them is proportional to the light intensity seen by the adjacent fireflies. The attractiveness β of the firefly can be defined as

$$\beta = \beta_0 e^{-\gamma r^2} \tag{2.5}$$

where β_0 is attractiveness at r=0. And attraction of firefly *i* to another brighter firefly *j* is given by

$$x_i = x_i + \beta_o e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + \alpha \epsilon_i \tag{2.6}$$

α is a randomization parameter and ϵ_i is a vector of random numbers.

2.2.2 Pseudocode for FFA

1) Define objective function. $F(x), X = (x_1, \dots, x_d)^T$.
2) Generate initial population x_i (i = 1, 2, … , n)
3) Determine light intensity I_i at x_i is determined by $f(x_i)$.
4) Define light absorption coefficient γ.
5) Compare each firefly with all others and move to the firefly having maximum intensity.

   Variation of attraction with distance r via $\exp(-\gamma r)$ and evaluate new values and update light intensity.Repeat step 5 until maximum generation is reached

6) Rank the fireflies, find the current global best g*
7) Go to step 4 till satisfying condition is reached.

Fig. 2.3 shows flowchart for Firefly Algorithm

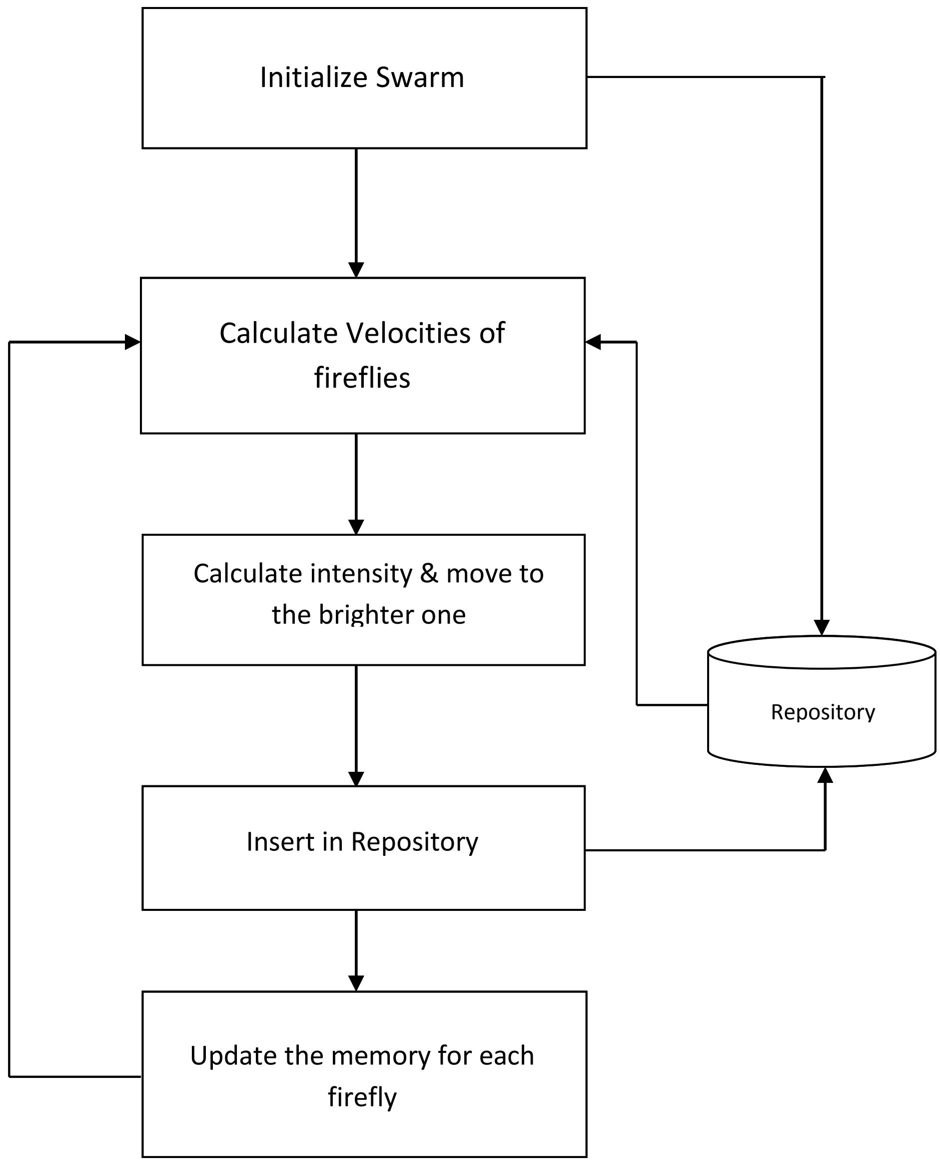

Fig. 2.3 Flowchart for Firefly Algorithm

CHAPTER 3

ADAPTIVE CONTROL IMPLEMENTATION USING PSO AND FFA

Adaptive control has been extensively developed and investigated in both application and theory during the past few decades and it still exists as a very active research field. In order to achieve or to maintain a desired level of control system performance when the parameters of the plant dynamic model are unknown and/or change in time, adaptive Control covers a set of techniques which provide a systematic approach for automatic adjustment of controllers in real time. If a situation arises where the parameters of the dynamic model of the plant to be controlled are constant but are unknown in a certain region of operation. Although the model of the control or the controller structure will not depend upon the particular values of the parameters of the plant model, without knowledge of their values correct tuning of the controller parameters cannot be done.

Here, an automatic tuning procedure in closed loop for the controller parameters can be provided by the adaptive control techniques. In these situations, effect of the adaptation vanishes as time increases. If there are changes in the operation conditions, a restart of the adaptation procedure may be required.

Now, if we consider a situation when the parameters of the dynamic model of the plant change in an unpredictable manner with time. These cases occur either because a simplified linear model for nonlinear systems have been considered as a change in operation condition will lead to a different linearized model or because the environmental conditions change as is the case when the models vary with the load types.

The various adaptation techniques are characterized by the way in which information is processed in real time to tune the controller for achieving the desired performances.

Since the parameters of the controller will depend upon measurements of system variables through the adaptation loop, the adaptive control system behaves nonlinearly. The Adaptive Control can be classified broadly as Direct and Indirect adaptive control.

3.1 Direct Adaptive Control

There at least two general approaches adaptive control and in the first approach, the adaptation mechanism adapts the parameters of the controller by observing the signal from the control system to maintain performance even if there are changes in the plant. In either the direct or indirect adaptive controllers, sometimes, the desired performance is characterized with a reference model and the controller then seeks to make the closed-loop system, even if the plant changes, behave as the reference model would. This is called "**Model Reference Adaptive Control (MRAC)**".Here, in direct adaptive control, the controller could be PSO or FFA or any other optimization method based on biomimicry of foraging.

The block diagram for a direct adaptive control is shown below:

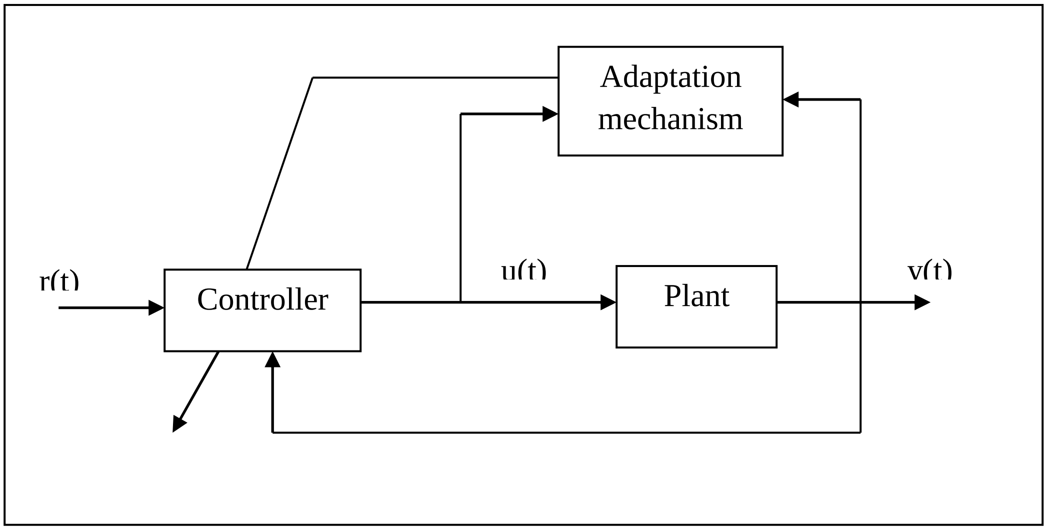

Fig. 3.1: Direct Adaptive Control

It can be seen from the block diagram that the plant which is to be controlled is analyzed by an adaptation mechanism. The adaptation mechanism takes the input and output obtained from the plant for a particular instant and generates a signal which controls the controller. The controller in accordance with the signal generated by the adaptation mechanism gives a control signal to control the plant. This adaptation mechanism is where a biologically inspired algorithm is added to coordinate and control overall plant.

3.2 Indirect Adaptive Control

Here online identification method is used to estimate the plant input-output mapping and a "controller designer" module to subsequently specify the parameters of the controller. Generally indirect adaptive controller can be taken as automating the mode-building and control design process that is used for fixed controller.

The identifier will provide estimates of changes that occur during plant input-output mapping and the controller will be subsequently tuned by the controller designer. It is assumed that we are certain that the estimated plant mapping is equivalent to the actual one at all instants which is called the Certainty equivalence principle. Since we tune the controller indirectly by first estimating the plant parameters, the overall approach is called Indirect Adaptive Control. The identifier model we use here is implemented by using Particle Swarm Optimization or Firefly Algorithm with tunable parameters that enter in a nonlinear fashion.

As a part of this work Indirect Adaptive Control strategy is used for non-linear systems and it can be represented by a block diagram as shown in fig 3.2. It can be seen from the block diagram that the plant to be controlled is analyzed by a system identification block. This identification block is utilizes plant input and output to generate a signal varying with the plant conditions at a particular instant.

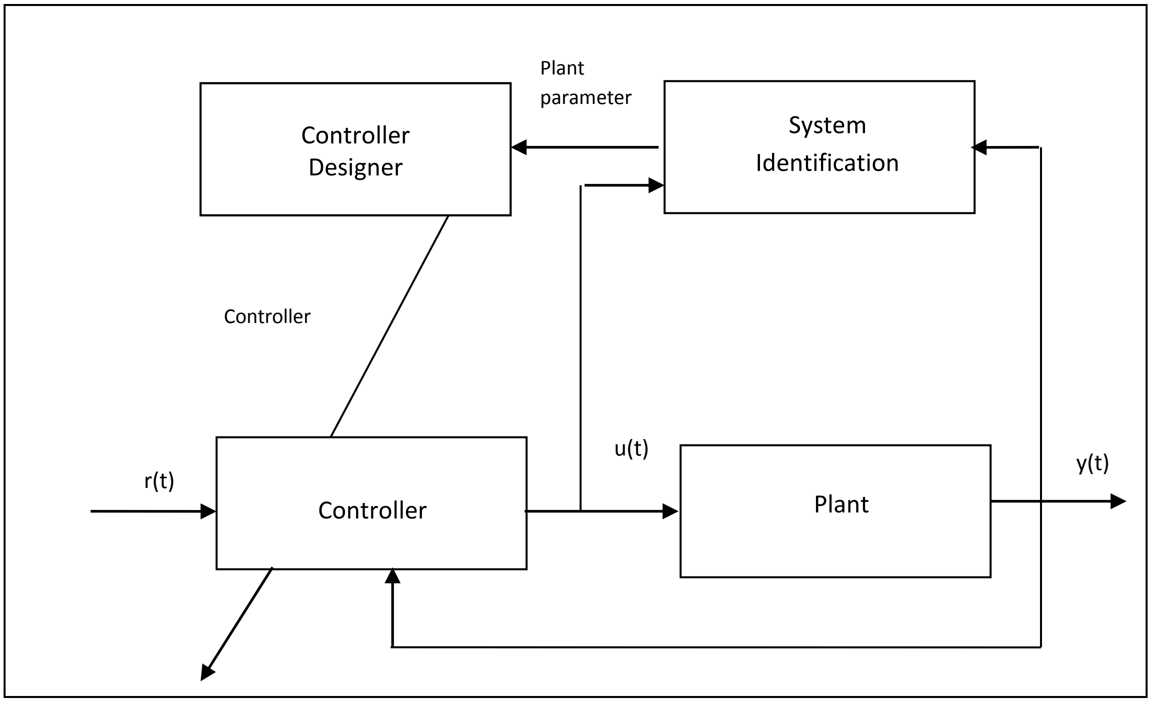

Fig. 3.2: Indirect Adaptive Control

The information accessed by the system identification block is then used to generate a signal which characterizes the plant parameters. This signal is then given to the controller designer block. The controller designer block generates a signal in response to the signal received by it pertaining to the plant parameters by the system identification block. Controller derives an input which is nothing but the output of the plant to be controlled. Apart from that, it also receives the signal from the controller designer block. These signals alongwith the reference signal makes the controller to generate a control signal efficient enough to control the plant and thus, by each passing instant it moves the output of the plant close to the desired reference trajectory.

The system identification block plays a major role here in controlling the plant as it estimates the plant parameters and accordingly adjusts the controller output indirectly while the direct adaptive control just gives an adaptation depending on only the input and output of the plant.

The model that is being tuned is a nonlinear function. The nonlinear mapping it implements is unknown since the plant is assumed to be unknown. In order to adjust the nonlinear mapping implemented by the PSO or FFA to match the unknown nonlinear mapping of the plant, an online function approximation is carried out.

3.3 Implementation of PSO and FFA as controllers

Consider a system which is represented as

$$y(k+d) = f\big(x(k), u(k)\big) \tag{3.1}$$

where $f(x(k), u(k))$ is a smooth (but unknown) function of its arguments, $u(k)$ is the measurable scalar input, $y(k)$ is the measurable (scalar) output, d>=1 is the measurable delay between the input and output, and $x(k)$ is the state vector.

In the indirect adaptive control case, a special subclass of plants is considered which can be represented by

$$y(k+d) = \alpha\big(x(k)\big) + \beta\big(x(k)\big)u(k) \tag{3.2}$$

$$\text{or} \quad y(k+d) = \alpha_u\big(x(k)\big) + \alpha_k(k) + \Big(\beta_u\big(x(k)\big) + \beta_k(k)\Big)u(k) \tag{3.3}$$

where $\alpha_u\big(x(k)\big)$ and $\beta_u\big(x(k)\big)$ are unknown smooth functions of the state $x(k)$ and represent the unknown nonlinear dynamics of the plant. These functions are to be estimated so that we can specify a controller. $\alpha_k(k)$ and $\beta_k(k)$ are defined to be known parts of the plant dynamics where we know portions of the nonlinear dynamics. We assume $\beta\big(x(k)\big)$ to satisfy

$$0 < \beta_o \leq \beta\big(x(k)\big)$$

for some known $\beta_o > 0$ for all $x(k)$. This places a restriction on the class of the plants that we can consider. Intuitively, we require that the gain on $u(k)$ be bounded from below due to how an estimate of β will be used to specify the control. It is also possible to develop a scheme where $\beta\big(x(k)\big)$is known to be negative and bounded from above by a constant that is less than zero.

3.3.1 Estimating an Unknown Ideal Controller

Let $r(k+d)$ be the reference, which is reasonable for many applications as it is specified by the user, we know that there exists an ideal controller

$$u^*(k) = \frac{-\alpha\big(x(k)\big) + r(k+d)}{\beta\big(x(k)\big)} \tag{3.4}$$

that linearizes the dynamics of equation (3.2) such that $y(k) \rightarrow r(k)$.

To verify this, substitute $u(k) = u^*(k)$ in equation (3.2) we get

$$y(k+d) = r(k+d) \tag{3.5}$$

so that tracking of the reference is achieved within d steps. Since we do not know $\alpha\big(x(k)\big)$ and $\beta\big(x(k)\big)$. Thus an estimator for these plant nonlinearities is developed and is used to form an approximation to $u^*(k)$.

3.3.2 Certainty Equivalence Controller

The certainty equivalence approach entails specifying a control input using an estimate of the plant model that would cancel appropriate plant dynamics and achieve good tracking if the estimate was accurate.

The control input using a certainty equivalence approach is defined as

$$u(k) = \frac{-\hat{\alpha}\big(x(k)\big) + r(k+d)}{\hat{\beta}\big(x(k)\big)} \tag{3.6}$$

where $\hat{\alpha}\big(x(k)\big)$ and $\hat{\beta}\big(x(k)\big)$ are estimates of $\alpha\big(x(k)\big)$ and $\beta\big(x(k)\big)$, respectively. They are defined as

$$\hat{\alpha}\big(x(k)\big) = \theta_\alpha^T(k)\emptyset_\alpha\big(x(k)\big) + \alpha_k(k) \tag{3.7}$$

$$\hat{\beta}\big(x(k)\big) = \theta_\beta^T(k)\emptyset_\beta\big(x(k)\big) + \beta_k(k) \tag{3.8}$$

In order to pick $\theta_\alpha(k)$ and $\theta_\beta(k)$ Particle Swarm Optimization or Firefly Algorithm is used to try to minimize the approximation error.

Projection algorithms can be used to ensure that $\theta_\alpha(k)$ and $\theta_\beta(k)$ lie in a valid region for all k. They can also be used to ensure that $\hat{\beta}\big(x(k)\big) \geq \beta_o$ so that the control signal is well defined. If it is known that each element of the $\emptyset_\beta$vector is always positive, then to ensure that $\hat{\beta}\big(x(k)\big) \geq \beta_o$ a projection method could be used to keep each component of $\theta_\beta(k)$ greater than or equal toβ_o.

3.3.3 Error Equation Representation

The tracking error that results from the above definitions is denoted as

$$e(k) = r(k) - y(k) \tag{3.9}$$

$$e(k) = r(k) - \alpha\big(x(k)\big) - \beta\big(x(k)\big)u(k)$$

from equation (3.6) we get the value of r(k) (delay step is removed to be specific).Thus, equation (3.9) could be rewritten as

$$e(k) = \Big(\hat{\alpha}\big(x(k)\big) - \alpha\big(x(k)\big)\Big) + \Big(\hat{\beta}\big(x(k)\big) - \beta\big(x(k)\big)\Big)u(k) \tag{3.10}$$

If $\hat{y}(k) = \hat{\alpha}\big(x(k)\big) + \hat{\beta}\big(x(k)\big)u(k)$ then with certainty equivalence control law, the output tracking error could be viewed as the identification error, i.e. a measure of the quality of the model that we are tuning to represent the plant, could be represented as

$$e(k) = \hat{y}(k) - y(k) \tag{3.11}$$

3.3.4 Adaptation Method for Cost Function

Steepest descent approach is used to define a cost function represented as

$$J\big(\theta^i\big) = e^{i^2}(k) \tag{3.12}$$

or

$$J\big(\theta^i\big) = \big(\hat{y}^i(k) - y(k)\big)^2 \tag{3.13}$$

where i=1,2,….S and S is the maximum number of Particles or Fireflies as the case may be. As a swarm perspective θ^i is viewed as the location of the i^{th} particle or firefly in the search space. A set of approximations for $\hat{\alpha}\big(x(k)\big)$ and $\hat{\beta}\big(x(k)\big)$ for i^{th} particle or firefly are denoted by $F_\alpha\big(x(k),\theta_\alpha^i(k)\big)$ and $F_\beta\big(x(k),\theta_\beta^i(k)\big)$.

Let the i^{th} estimate of the output and identification error be given as

$$\hat{y}^i(k) = F_\alpha\big(x(k),\theta_\alpha^i(k-1)\big) + F_\beta\left(x(k),\theta_\beta^i(k-1)\right)u(k-1) \qquad (3.14)$$

Now, the cost function for i^{th} particle or firefly could be given as

$$J\big(\theta^i(k)\big) = \left(F_\alpha\big(x(k),\theta_\alpha^i(k-1)\big) + F_\beta\left(x(k),\theta_\beta^i(k-1)\right)u(k-1) - y(k)\right)^2 \qquad (3.15)$$

It is required to minimize this cost function which actually measures the size of the estimation error and this is done by using the position of the particle or firefly in the search spaceθ^i. The particle's or firefly's position in one dimension is given by θ_α and in the other dimension by θ_β so that the position$\theta^i = \left[\theta_\alpha^i, \theta_\beta^i\right]^T$, i=1,2,…S.

A block diagram depicting indirect adaptive control using PSO or FFA is shown in figure 3.3. Here the system identification part is shown to be containing either the Particle Swarm Optimization Algorithm or the Firefly Algorithm which analyzes the plant to be controlled and estimates the parameters of the plant.

The output from the system identification is given to a certainty equivalence control law which is governed by the required trajectory response and is also responsible to generate signal determining the control parameters which in turn will assist the controller in successfully and efficiently control the plant.

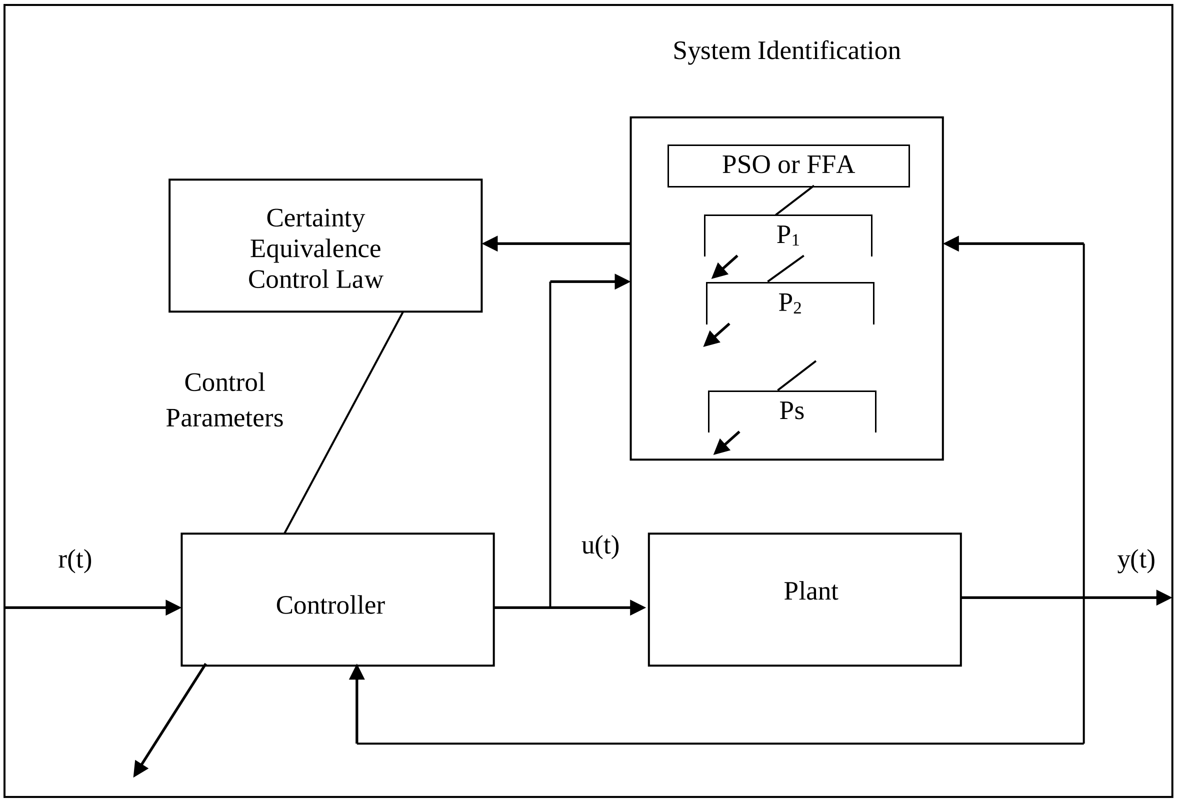

Figure 3.3: Adaptive Control using PSO or FFA

Adaptation using PSO

Each particle has a randomized position $\theta = [\theta_\alpha^i, \theta_\beta^i]^T$ in the search space. Each particle calculates the estimates $\hat{\alpha}(x(k))$ and$\hat{\beta}(x(k))$. Using these estimates, the output estimate $\hat{y}^i(k)$ is calculated using (3.14) alongwith the tracking error $e^i(k)$ and the cost function$J(\theta^i(k))$. After getting a set of values for$J(\theta(k))$ having size of the total population of the particles, it is searched for the minimum value amongst the values calculated by each particle. The particle which gives the minimum value of cost function is selected as the best particle and considered as having the best position θ_α^* and θ_β^*. Each particle is then moved towards the direction of the best particle using the update laws as described in (2.1) & (2.2) earlier. The estimates $\hat{\alpha}(x(k))$ and $\hat{\beta}(x(k))$ are then calculated using the best position θ_α^* and θ_β^*. These estimates are then used to calculate the control output $u(k)$ using certainty equivalence approach as described by (3.6). The system $y(k)$ is updated with this value of controller output using (3.2) which moves it close to the reference trajectory. The process is repeated for a finite number of iterations which ensures that the system tracks the reference trajectory with minimum tracking error.

Adaptation using FFA

Each firefly has a randomized position $\theta = \left[\theta_\alpha^i, \theta_\beta^i\right]^T$ and intensity of light I_i in the search space. Each firefly calculates the estimates $\hat{\alpha}\big(x(k)\big)$ and $\hat{\beta}\big(x(k)\big)$. Using these estimates, the output estimate $\hat{y}^i(k)$ is calculated using (3.14) alongwith the tracking error $e^i(k)$ and the cost function $J\big(\theta^i(k)\big)$. After getting a set of values for$J(\theta(k))$ having size of the total population of the fireflies, it is searched for the minimum value amongst the values calculated by each firefly. The firefly which gives the minimum value of cost function is selected as the best firefly and considered as having the best position θ_α^* and θ_β^* and having the maximum intensity of light I_i^*. Each less bright firefly is then moved towards the direction of the brightest firefly using the update laws as described in (2.4) & (2.6) earlier. The estimates $\hat{\alpha}\big(x(k)\big)$ and$\hat{\beta}\big(x(k)\big)$ are then calculated using the best position θ_α^* and θ_β^*. These estimates are then used to calculate the control output $u(k)$ using certainty equivalence approach as described by (3.6). The system $y(k)$ is updated with this value of controller output using (3.2) which moves it close to the reference trajectory. The process is repeated for a finite number of iterations which ensures that the system tracks the reference trajectory with minimum tracking error.

Adaptive Control and Conventional Feedback Control

The unmeasurable and unknown variations of the process parameters lead to a degrade in the performances of the control systems. In a similar manner to the disturbances acting upon the controlled variables, it can be considered that the variations of the process parameters are caused by disturbances acting upon the parameters which are called parameter disturbances.

These parameter disturbances will affect the performance of the control systems. Therefore the disturbances acting upon a control system can be classified as follows:

(a) by the disturbances acting upon the controlled variables

(b) by the parameter disturbances acting upon the performance of the control system.

Feedback is basically used in conventional control systems in order to reject the effect of disturbances which are incident on the controlled variables. It brings them back to their desired values according to a certain performance index. The controlled variables are first measured in order to achieve this and then the measurements are compared with the desired values and the difference is fed into the controller which will generate the appropriate control.

For the problem of achieving and maintaining the desired performance of a control system in the presence of parameter disturbances, a similar conceptual approach can be considered . A performance index (IP) is first defined for the control system which is a measure of the performance of the system itself. As an example the damping factor for a closed-loop system characterized by a second-order transfer function is an IP which allows to quantify a desired performance expressed in terms of damping. Then this IP is measured. The measured IP will be compared to the desired IP and their difference will be fed into an adaptation mechanism. The output of the adaptation mechanism will act upon the control signal and/or the parameters of the controller in order to modify the system performance accordingly.

An adaptive control system measures a certain performance index (IP) of the control system using the outputs, the inputs and the known disturbances alongwith the states. From a set of given performance indices, a comparison of the measured one is carried out and the adaptation mechanism modifies the parameters of the adjustable controller and/or generates an auxiliary control in order to maintain the performance index of the control system close to the set of given ones. A conventional feedback control system will monitor the controlled variables under the effect of disturbances acting on them, but its performance will vary, and not actually monitor, under the effect of parameter disturbances as the design is done assuming constant and known process parameters.

An adaptive control system monitors the performance of the system in the presence of parameter disturbances since it contains a supplementary loop acting upon the adjustable parameters of the controller in addition to a feedback control with adjustable parameters.

A conventional feedback control system design is oriented toward the elimination of the effect of disturbances upon the controlled variables while the design of adaptive control systems is oriented toward the elimination of the effect of parameter disturbances upon the performance of the control system. When the controlled variable is considered as a performance index (PI), an adaptive control system can be interpreted as a feedback system.

CHAPTER 4

DYNAMICS OF NONLINEAR SYSTEMS

The Indirect Adaptive Control utilizing PSO and FFA technique is implemented to control two nonlinear systems whose dynamics are described here.

4.1 Level Control in a Nonlinear Surge Tank system

Design Problem: In this problem we will study the development of indirect adaptive control for the liquid level process control problem. θ^i is viewed as the location of the i^{th} particle in the search space. Consider the "surge tank" shown below with input u(t), the height of the liquid h(t) and an outlet below the tank characterized by $\tilde{d}$ which is related to the outlet pipe diameter.

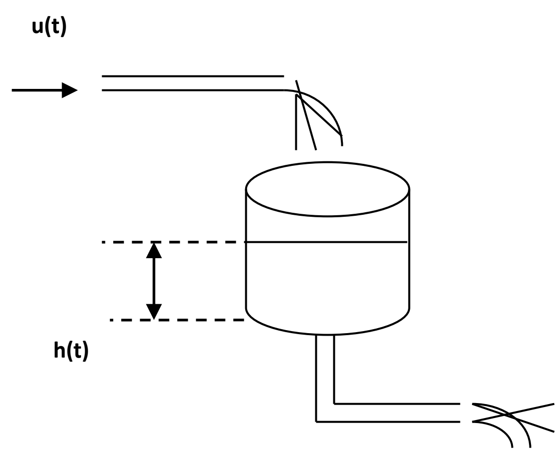

Figure 4.1: Surge Tank System

The input changes dynamically and the objective is to maintain the level of liquid in the surge tank at 'h' while u(t) is the varying input to the system. The mathematical model of the system can be given as

$$h(k+1) = h(k) + T\left(\frac{-\tilde{d}\sqrt{19.6h(k)}}{\left|\tilde{a}h(k)+\tilde{b}\right|} + \frac{\tilde{c}}{\left|\tilde{a}h(k)+\tilde{b}\right|}u(k)\right) \quad (4.1)$$

The end-use of the model defines the model requirements. The above model is derived by relating the inputs to measured outputs that needs to be regulated. The volume of liquid in the vessel varies as a function of the inlet and outlet flow rates.

The density is assumed to be constant. Balanced equations based on an instantaneous rate of change can be given as

$$\begin{bmatrix}\text{Rate of change of}\\ \text{total mass of fluid}\\ \text{inside the vessel}\end{bmatrix} = \begin{bmatrix}\text{mass flow rate}\\ \text{of fluid}\\ \text{into the vessel}\end{bmatrix} - \begin{bmatrix}\text{mass flow rate}\\ \text{of fluid}\\ \text{out of the vessel}\end{bmatrix}$$

Following notations are used in the modeling equation:

u =inlet volumetric flow rate (volume/time);

o =outlet volumetric flow rate (volume/time);

V =volume of liquid in vessel;

h =height of liquid in vessel;

ρ =liquid density (mass/volume);

A =cross-sectional area of vessel = $|\tilde{a}h(k) + \tilde{b}|$ where $\tilde{a} > 0$ and $\tilde{b} > 0$.

$\tilde{c}$ = clogging factor ϵ (0,1). $\tilde{c} = 1$shows that the filter in the actuator

$\tilde{d}$ =area of liquid discharge related to the outlet pipe diameter.

where the total mass of fluid inside the vessel is denoted by$V\rho$, the rate of change is$\frac{dV\rho}{dt}$, and the density of the outlet stream is equal to the density of the vessel contents.

$$\frac{dV\rho}{dt} = \tilde{c}u\rho - o\rho \tag{4.2}$$

The volumetric flow rate Q can be written as $Q = Av$ where A= area and v = velocity. Thus, the outlet volumetric flow rate can be expressed with $\tilde{d}$ and outlet velocity v as

$$o = \tilde{d}\, v \tag{4.3}$$

If we equate kinetic and potential energies of liquid, we get

$$mgh = \frac{1}{2}mv^2$$

g =acceleration due to gravity and m = mass of liquid. 'm' gets cancelled on both sides of the equation and hence we get,

$$v = \sqrt{2gh} \tag{4.4}$$

using (4.3) & (4.4) a modified (4.2), after cancelling out ρ on both sides of the equation

$$\frac{dV}{dt} = \tilde{c}u - \tilde{d}\sqrt{2gh} \tag{4.5}$$

Volume V of the tank can be expressed as $V = Ah$. Thus, the equation can be modified as,

$$\frac{dh}{dt} = \frac{\tilde{c}u - \tilde{d}\sqrt{2gh}}{A}$$

this can be modified further as

$$\frac{dh}{dt} = -\frac{\tilde{d}\sqrt{2gh}}{|\tilde{a}h(k) + \tilde{b}|} + \frac{\tilde{c}}{|\tilde{a}h(k) + \tilde{b}|}u \tag{4.6}$$

Putting the value of g=9.8 and converting into discrete form

$$h(k+1) = h(k) + T\left(\frac{-\tilde{d}\sqrt{19.6h(k)}}{|\tilde{a}h(k) + \tilde{b}|} + \frac{\tilde{c}}{|\tilde{a}h(k) + \tilde{b}|}u(k)\right) \tag{4.7}$$

In the simulation model we use $\tilde{a} = 0.01, \tilde{b} = 0.2, \tilde{c} = 1, \tilde{d} = 1$ and $T = 0.1$.We assume that the plant input saturates at ± 50 so that if the controller generates an input $u(k)$, then

$$u(k) = \begin{Bmatrix} 50 & if\ u(k) > 50 \\ u(k) & if -50 < u(k) < 50 \\ -50 & if\ u(k) < -50 \end{Bmatrix}$$

Since the liquid level cannot go negative, the system model is modified as

$$h(k+1) = max\left\{0.0001, h(k) + T\left(\frac{-\tilde{d}\sqrt{19.6h(k)}}{|\tilde{a}h(k) + \tilde{b}|} + \frac{\tilde{c}}{|\tilde{a}h(k) + \tilde{b}|}u(k)\right)\right\} \tag{4.8}$$

The above model described by (4.8) is used for the adaptive control of the liquid level using both Particle Swarm Optimization and Firefly Algorithm for a comparative analysis.

4.2 Pole Angle Control in an Inverted Pendulum System

In recent Years, projects on the themes of robotics and mechatronics are the most attractive. Many interesting robotic benchmark systems exist in the literature in this framework. The inverted pendulum system is always considered as the most fundamental benchmark among others. This system exists in many different versions offering a variety of interesting control challenges.

Design Problem: In this problem we will see the development of indirect adaptive control for the control of angle of pole in an Inverted Pendulum System. Figure 4.2 below shows an inverted pendulum on a cart system.

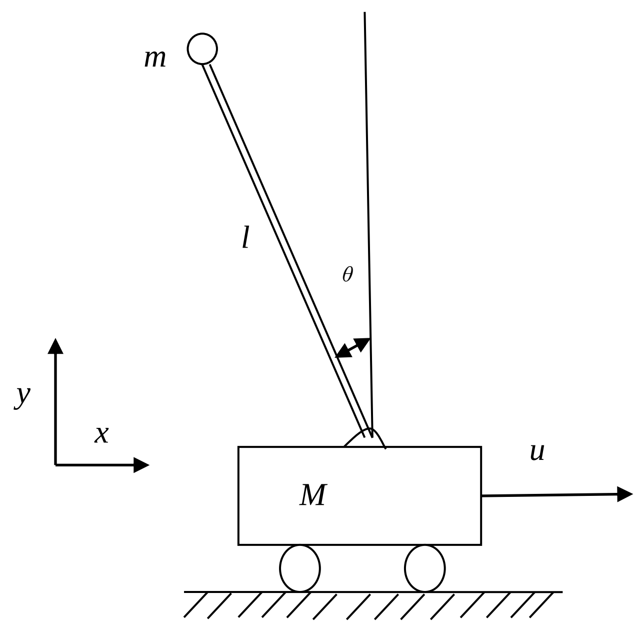

Figure 4.2: Inverted Pendulum on Cart

The system consists of an inverted pole hinged on a cart. When the cart is given a push or force in the x axis direction, it moves along the direction of force. This results in a change in the angle of the pole with the vertical. If an initial deviation in the angle is considered, the objective of the control would be to keep the angle of the pole with respect to the vertical axis zero, i.e. in the upright position. Thus, the system could be modeled by taking u as the control input variable and θ as the controlled variable. While controlling, certain constraints should be taken care of, i.e. the cart should not go beyond a specified finite value, i.e. the cart should be moving through a certain specified range. The angle of pole must lie within the range of $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Model of Inverted Pendulum

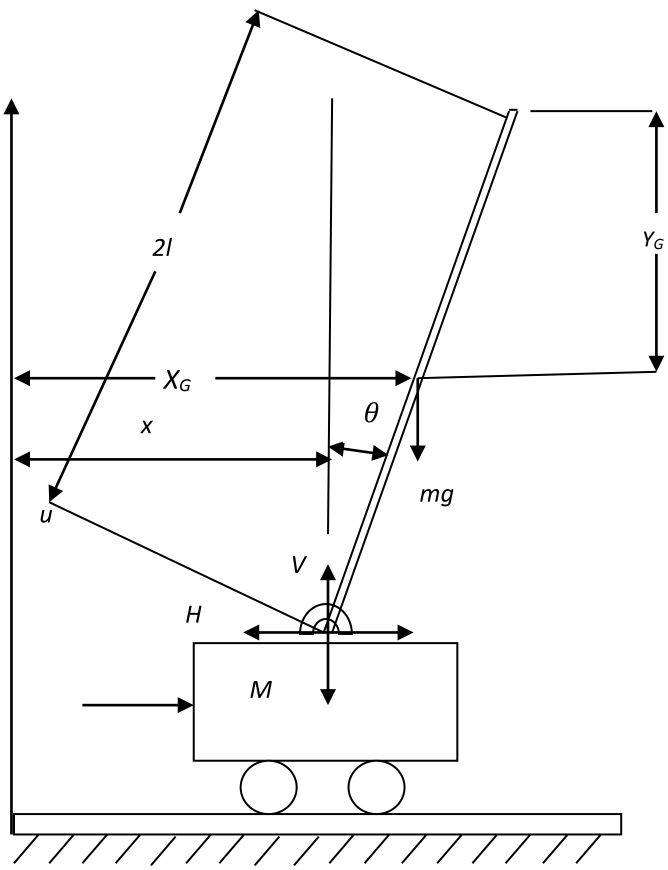

Figure 4.3: Mathematical model of Inverted pendulum on cart

M (Mass of Cart) = 3Kg

m (Mass of pendulum) = 0.2Kg

*l (*Length to center of mass of pendulum) = 0.31m

u- Force applied to the cart

x- Cart position coordinate

θ-Pendulum angle from vertical

b (Frictional force of cart) = 0.1 N/ms^{-1}

The dynamical equation of the system can be derived as follows:

Force analysis and system equation

The center of gravity of the rod in pendulum with respect to the origin on the Cartesian axis of reference can be written in the form of equation as

$$X_G = x + l\sin\theta$$

$$Y_G = l\cos\theta$$

The balance of forces for the rod with respect to its center of gravity in the vertical direction is given by

$$m\frac{d^2}{dt^2}l\cos\theta = V - mg$$

$$m\left[\frac{d}{dt}\left(l\sin\theta\,\dot{\theta}\right)\right] = V - mg$$

$$m\left[-l\cos\theta\dot{\theta}^2 - l\sin\theta\ddot{\theta}\right] = V - mg$$

On completing the differentiation above we get

$$mg - ml\ddot{\theta}\sin\theta - ml\cos\theta\,\dot{\theta}^2 = V \tag{4.9}$$

The balance of forces of the rod with respect to its center of gravity in the horizontal direction is given by

$$m\frac{d^2}{dt^2}(x + l\sin\theta) = H$$

$$m\left[\ddot{x} + \frac{d}{dt}\left(l\dot{\theta}\cos\theta\right)\right] = H$$

$$m\left[\ddot{x} + l\cos\theta\,\ddot{\theta} - l\sin\theta\,\dot{\theta}^2\right] = H$$

$$m\ddot{x} + ml\cos\theta\,\ddot{\theta} - ml\sin\theta\,\dot{\theta}^2 = H \tag{4.10}$$

Balancing the forces of the cart in the x direction is described by

$$M\frac{d^2x}{dt^2} + b\frac{dx}{dt} = u - H \tag{4.11}$$

Balancing the rotational motion of the pendulum rod around its center of gravity is given by

$$I\ddot{\theta} = Vl\sin\theta - Hl\cos\theta \tag{4.12}$$

Put (4.10) in (4.11)

$$M\ddot{x} = u - m\ddot{x} - ml\cos\theta\,\ddot{\theta} + ml\sin\theta\,\dot{\theta}^2$$

After solving this equation, we get

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = u \tag{4.13}$$

Put (4.9) & (4.10) in (4.13)

$$I\ddot{\theta} = \left(mg - ml\ddot{\theta}\sin\theta - ml\cos\theta\,\dot{\theta}^2\right)l\sin\theta - \left(m\ddot{x} + ml\cos\theta\,\ddot{\theta} - ml\sin\theta\,\dot{\theta}^2\right)l\cos\theta$$

After solving it, we get

$$(ml^2 + I)\ddot{\theta} = mgl\sin\theta - m\ddot{x}l\cos\theta \tag{4.14}$$

From (4.14)

$$\ddot{\theta} = \frac{mgl\sin\theta - m\ddot{x}l\cos\theta}{ml^2 + I}$$

Put the above expression in (4.13)

$$(M+m)\ddot{x} + ml\left(\frac{mgl\sin\theta - m\ddot{x}l\cos\theta}{ml^2 + I}\right)\cos\theta + b\dot{x} - ml\dot{\theta}^2\sin\theta = u$$

$$(M+m)\ddot{x} + \frac{m^2l^2g\sin\theta\cos\theta}{ml^2 + I} - \frac{m^2l^2\ddot{x}cos^2\theta}{ml^2 + I} + b\dot{x} + ml\dot{\theta}^2\sin\theta = u$$

To approximate the equation, we use

$$\sin\theta = \theta$$

$$\dot{\theta}^2 = 0$$

$$\cos\theta = 1$$

$$sin^2\theta = 0$$

$$cos^2\theta = 1$$

Then we get,

$$\ddot{x} = -\frac{m^2l^2g}{(ml^2+I)M+mI}\theta - \frac{b(ml^2+I)}{(ml^2+I)M+mI}\dot{x} + \frac{(ml^2+I)}{(ml^2+I)M+mI}u \qquad (4.15)$$

From (4.14)

$$\ddot{x} = \frac{mgl\sin\theta - (ml^2+I)\ddot{\theta}}{ml\cos\theta}$$

Put the above expression in (4.13)

$$(M+m)\left[\frac{mgl\sin\theta - (ml^2+I)\ddot{\theta}}{ml\cos\theta}\right] + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = u$$

$$\frac{(M+m)g\sin\theta}{\cos\theta} - \frac{(M+m)(ml^2+I)}{ml\cos\theta} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = u$$

By approximating the equation, we get

$$(M+m)g\theta - \frac{(M+m)(ml^2+I)}{ml}\ddot{\theta} + b\dot{x} + ml\ddot{\theta} = u$$

Solving the above equation, we get

$$\ddot{\theta} = \frac{(M+m)mgl}{M(ml^2+I)+mI}\theta + \frac{bml}{M(ml^2+I)+mI}\dot{x} - \frac{ml}{M(ml^2+I)+mI}u \qquad (4.16)$$

Designing a state space model requires further transformations to be applied to these equations. Introducing the substitution variables for (4.15) and (4.16)

$$A = \frac{(M+m)mgl}{M(ml^2+I)+mI}$$

$$B = \frac{bml}{M(ml^2+I)+mI}$$

$$C = -\frac{m^2l^2g}{(ml^2+I)M+mI}$$

$$D = -\frac{b(m^2l+I)}{(ml^2+I)M+mI}$$

With the data, the actual values of the constants are calculated and given as:

$$A = 33.75$$

$$B = 0.11$$

$$C = -0.65$$

$$D = -0.03$$

Therefore, the linear state space model is:

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ x\\ \dot{x}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ A & 0 & 0 & B\\ 0 & 0 & 0 & 1\\ C & 0 & 0 & D\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ x\\ \dot{x}\end{bmatrix} + \begin{bmatrix}0\\ -ml/[(ml^2+I)M+mI]\\ 0\\ (m^2l+I)/[(ml^2+I)M+mI]\end{bmatrix}u$$

After inserting constant data, the state space model is:

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ x\\ \dot{x}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 33.75 & 0 & 0 & 0.11\\ 0 & 0 & 0 & 1\\ -0.65 & 0 & 0 & -0.03\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ x\\ \dot{x}\end{bmatrix} + \begin{bmatrix}0\\ -1.08\\ 0\\ 0.33\end{bmatrix}u \qquad (4.17)$$

The state space model obtained in (4.17) is converted into discrete form and we get the discrete state space representation of the above system as

$$\begin{bmatrix} \Delta\theta_{k+1} \\ \Delta\dot{\theta}_{k+1} \\ \Delta x_{k+1} \\ \Delta\dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.0013 & 0 & 0 \\ 0.0430 & 1 & 0 & 0.0001 \\ 0 & 0 & 1 & 0.0012 \\ -0.0008 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\theta_{k} \\ \Delta\dot{\theta}_{k} \\ \Delta x_{k} \\ \Delta\dot{x}_{k} \end{bmatrix} + \begin{bmatrix} 0 \\ -1.35 \times 10^{-3} \\ 0 \\ 0.4125 \times 10^{-3} \end{bmatrix} \Delta u_k \quad (4.18)$$

The state space model obtained in (4.18) is used for the adaptive control of pole angle using both Particle Swarm Optimization and Firefly Algorithm for a comparative analysis.

CHAPTER 5

SIMULATION RESULTS AND DISCUSSIONS

The indirect adaptive control of both the level control in a surge tank and the pole angle control in an inverted pendulum system using both PSO and FFA is described in the following sections.

5.1 Indirect Adaptive Control of Liquid Level in a Surge Tank System

The model of the system was derived in section 4.2 given by (4.8) which is again repeated below for convenience.

$$h(k+1) = max\left\{0.0001, h(k) + T\left(\frac{-\tilde{d}\sqrt{19.6h(k)}}{\left|\tilde{a}h(k)+\tilde{b}\right|} + \frac{\tilde{c}}{\left|\tilde{a}h(k)+\tilde{b}\right|}u(k)\right)\right\}$$

The indirect adaptive control technique estimates the height of the liquid by representing the estimate in the form given by (3.2) as

$$\hat{h}(k) = \alpha + \beta u(k) \tag{5.1}$$

Here, α and β are calculated by the location of the particles or the fireflies as the case may be. The fitness function is described by (3.13) i.e.

$$J(\theta^i) = \left(\hat{h}^i(k) - h(k)\right)^2$$

This equation is calculated by using the location of the particles/fireflies and the best particle/firefly is calculated such that the value of the fitness function for that particle/firefly is the least. Accordingly the estimate is calculated for the next iteration and the process is repeated for a certain number of iterations.

Simulations have been performed in MATLAB 7.7 and scratches have been developed using Intel(R) Core(TM) 2 Duo2.80GHz, 4GB of RAM. Both PSO and FFA indirect adaptive control have been conducted for same number of population which is 100 and the algorithm performs 1000 iterations with a sampling period of 0.1 secs.

For the surge tank system, the reference signal given is a square wave shown below in fig 5.1. The height of the liquid is not allowed to go negative as is evident from the graph. A time scale of 100 seconds is used

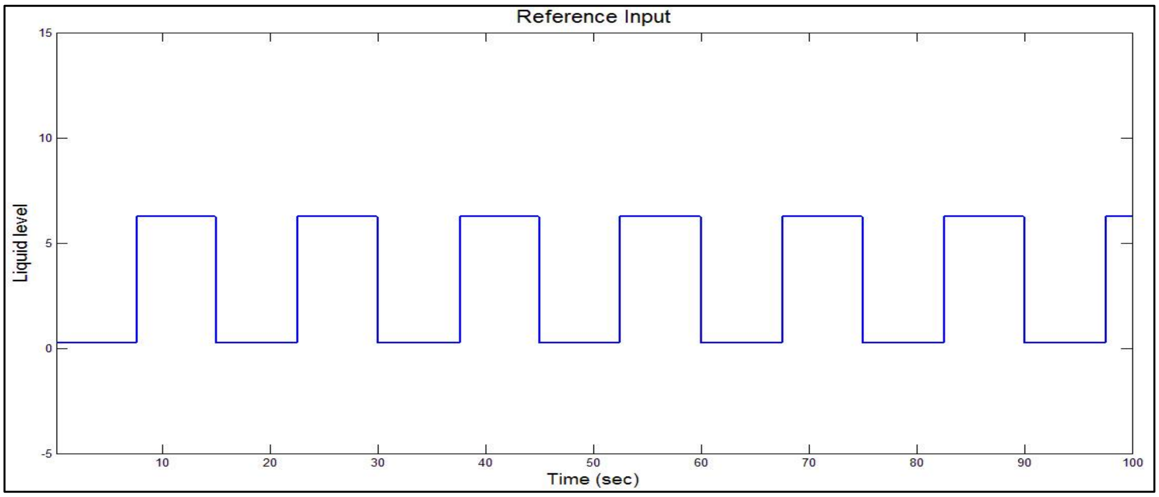

Figure 5.1: Reference input for the surge tank

The liquid level output for both PSO and FFA is shown in fig 5.2 below. The first graph is plotted for PSO and the second for FFA.

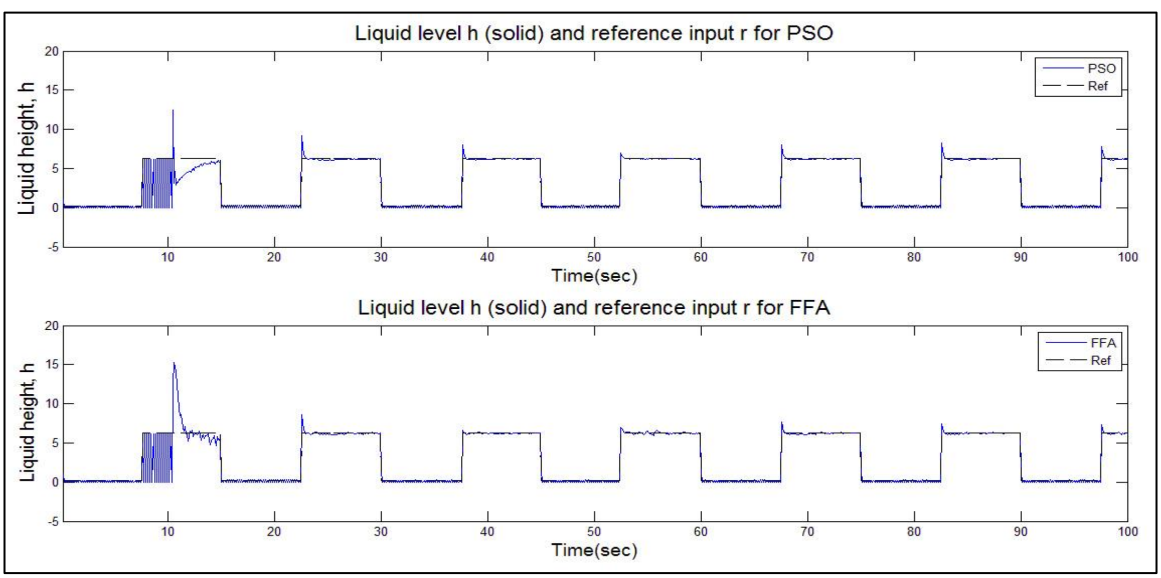

Figure 5.2: Liquid level ‘h’ and the reference input for PSO and FFA.

A comparative analysis on both the graphs show that peak overshoot for the first pulse is less for PSO when compared to the peak overshoot of FFA. Although the peak for both PSO and FFA significantly goes on decreasing after successive iterations, more number of oscillations can be observed in the case of fireflies.

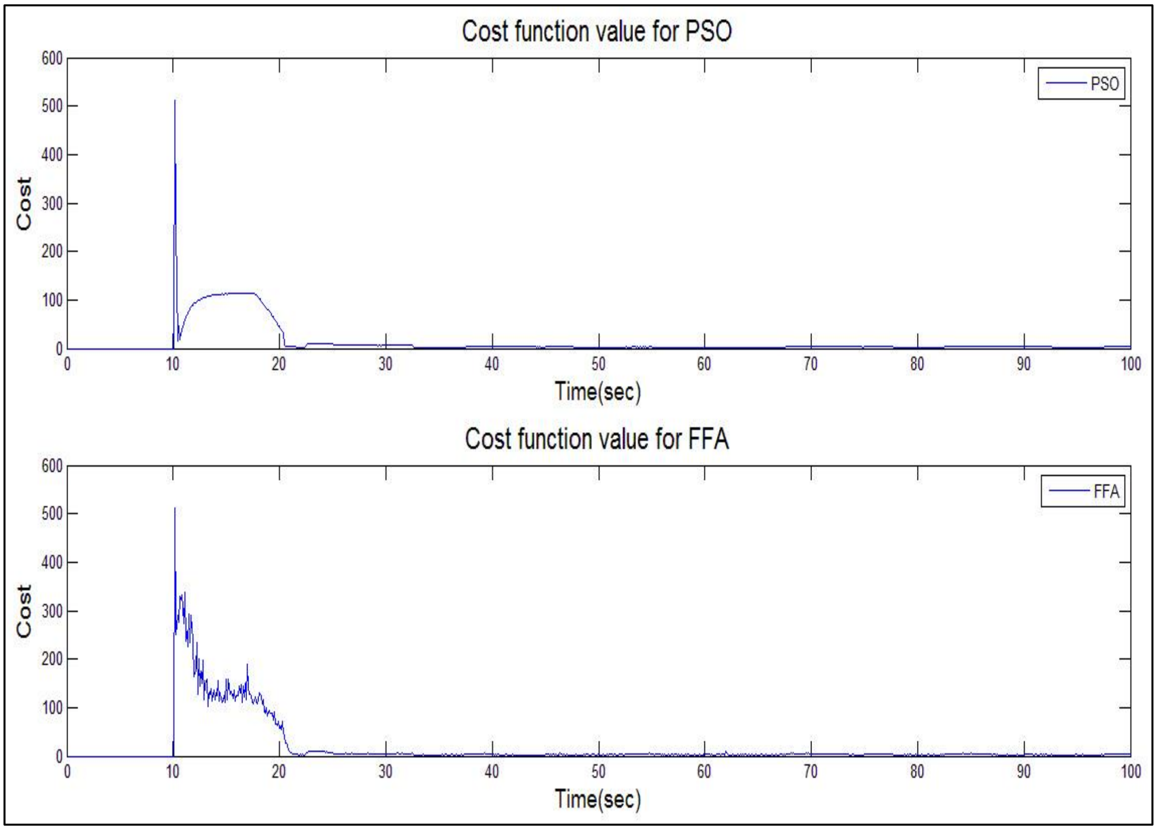

Figure 5.3: Cost function graph for both PSO and FFA for surge tank

Figure 5.3 shows the cost function for the particles in PSO and the fireflies in FFA. In both the cases, the value of the cost function decreases to zero. The cost value trend in FFA contains lots of variations while approaching the minimum value while in PSO shows a smooth variation. This shows that although the cost value for both the algorithm techniques reach a minimum value, the particle in PSO converge more smoothly towards the best particle at a particular iteration than the fireflies in the FFA.

A comparative analysis of the estimates of the liquid level ‘h’ can be performed with the help of the graphs shown in fig 5.4.

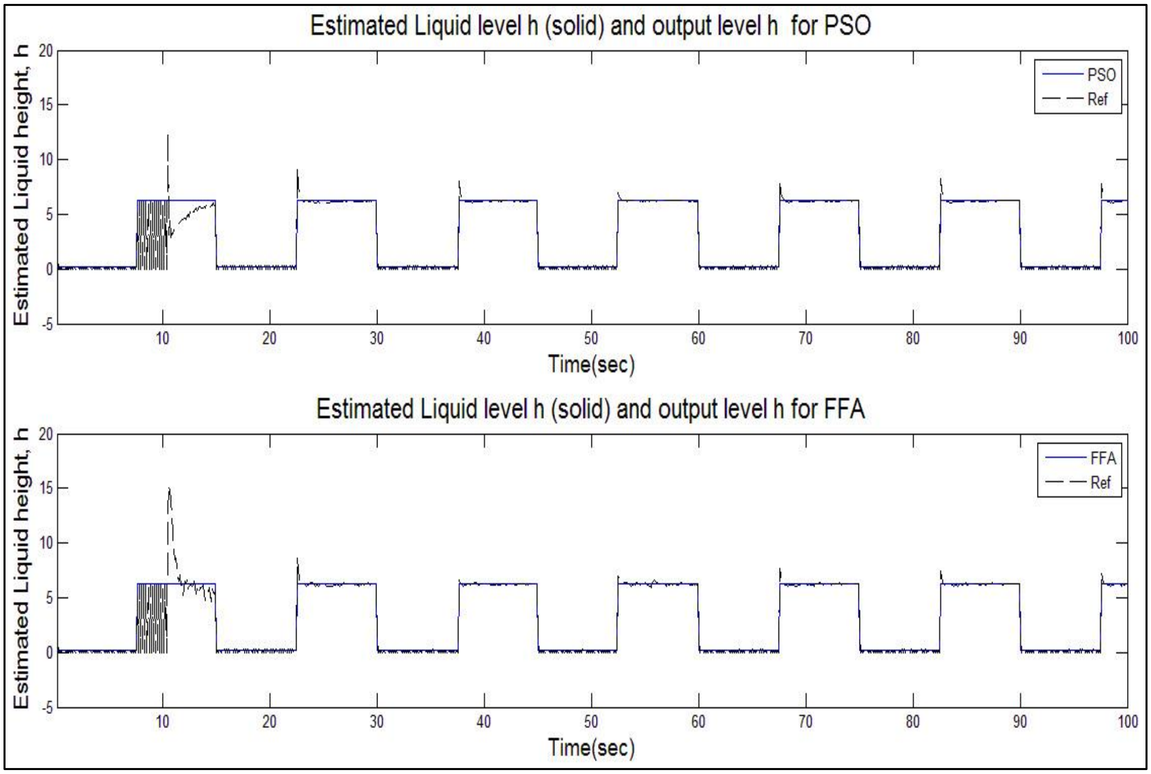

Figure 5.4: Estimates of the liquid level ‘h’ for both PSO and FFA.

The adaptive control of the tank system generates estimated liquid level in both the cases. It is evident from the graph that the estimated level is smooth with respect to the reference in the case of PSO when compared with FFA which has a large number of oscillations.

Eventually, both the algorithms reach the reference trajectory with minimum deviation from the desired response. In accordance with these estimates, the error between the estimated liquid level and the output liquid level ‘h’ but PSO estimate merges with the reference much faster than the FFA.

The graphs shown in fig. 5.5 depict error between the liquid level 'h' and the reference trajectory 'r' for both PSO and FFA.

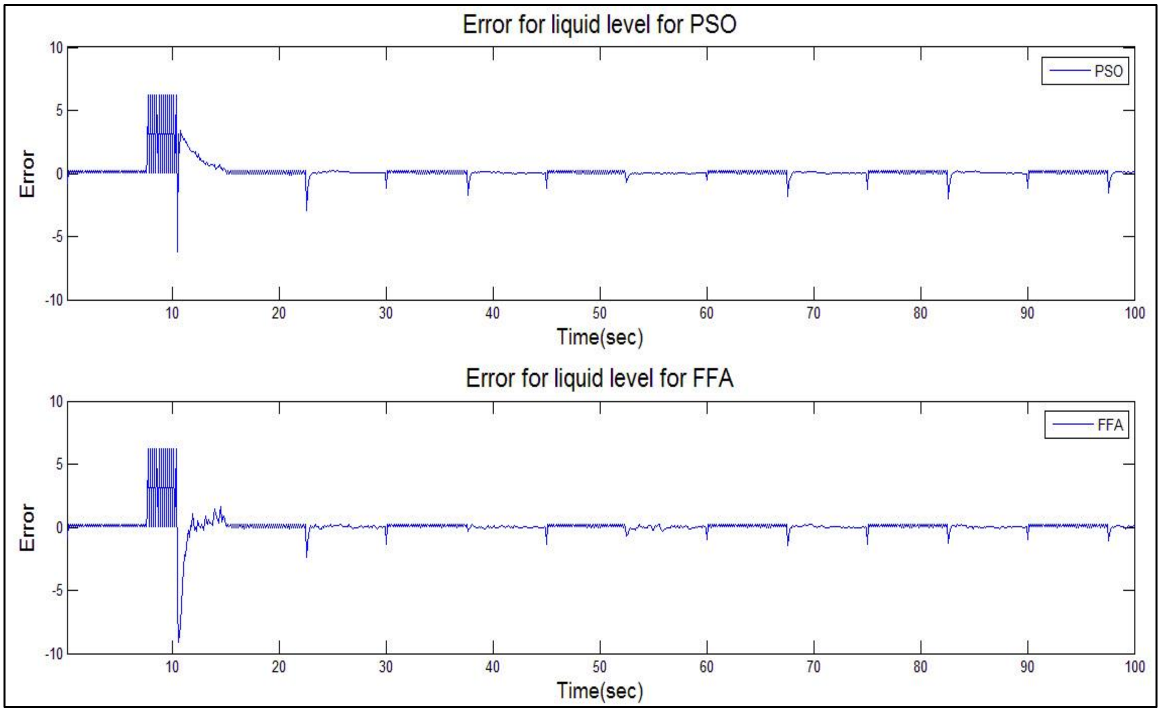

Figure 5.5: Error between the liquid level and the reference trajectory for PSO and FFA

In both the cases, it can be seen that the error reduces to zero after just 15 secs, but it can be seen that error trend in FFA has number of oscillations but the trend in PSO has a smooth response. The table 5.1 below shows a comparative result on the performance of PSO and FFA on the liquid level control in surge tank system.

Table 5.1: Elapsed time and Minimum Cost for the Surge tank level system

Algorithm	Elapsed Time	Minimum Cost
PSO	3.8429 secs	0.9779
FFA	93.1496 secs	1.0973

From table 5.1, it can be inferred from the data that elapsed time in PSO is much smaller than that in the case of FFA and also the minimum cost for PSO is also smaller than that of FFA.

5.2 Indirect Adaptive Control of Pole Angle in an Inverted Pendulum System

The model of the system was derived in section 4.1 given by (4.18) which is again repeated below for convenience.

$$\begin{bmatrix} \Delta\theta_{k+1} \\ \Delta\dot{\theta}_{k+1} \\ \Delta x_{k+1} \\ \Delta\dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.0013 & 0 & 0 \\ 0.0430 & 1 & 0 & 0.0001 \\ 0 & 0 & 1 & 0.0012 \\ -0.0008 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\theta_{k} \\ \Delta\dot{\theta}_{k} \\ \Delta x_{k} \\ \Delta\dot{x}_{k} \end{bmatrix} + \begin{bmatrix} 0 \\ -1.35 \times 10^{-3} \\ 0 \\ 0.4125 \times 10^{-3} \end{bmatrix} \Delta u_k$$

The states can be defined as

$x_1 = \theta$, Pole Angle

$x_2 = \dot{\theta}$, Angular velocity

$x_3 = x$, Position of the cart

$x_4 = \dot{x}$, Velocity of the cart

The indirect adaptive control technique estimates the angle of pole by representing the estimate in the form given by (3.2) as

$$\hat{x}_1(k) = \alpha + \beta u(k) \tag{5.2}$$

Here, α and β are calculated by the location of the particles or the fireflies as the case may be. The fitness function is described by (3.13) i.e.

$$J(\theta^i) = \left(\hat{x}_1{}^i(k) - x_1(k)\right)^2$$

This equation is calculated by using the location of the particles/fireflies and the best particle/firefly is calculated such that the value of the fitness function for that particle/firefly is the least. Accordingly the estimate is calculated for the next iteration and the process is repeated for a certain number of iterations and at the same time all the interdependent states are updated accordingly.

Simulations have been performed in MATLAB 7.7 and scratches have been developed using Intel(R) Core(TM) 2 Duo2.80GHz, 4GB of RAM. Both PSO and FFA indirect adaptive control have been conducted for same number of population which is 100 and the algorithm performs 3000 iterations with a sampling period of 0.1 secs.

The reference signal given to the system is zero, i.e. each state variable has to reach the zero state from their initial disturbance which gets initiated by the initial pole angle deviation with respect to the vertical. The initial pole angle deviation given to the system was 18^o.

Figure 5.6 shown below describes the result obtained regarding the pole angle deviation from the vertical by simulating the system with both PSO and FFA.

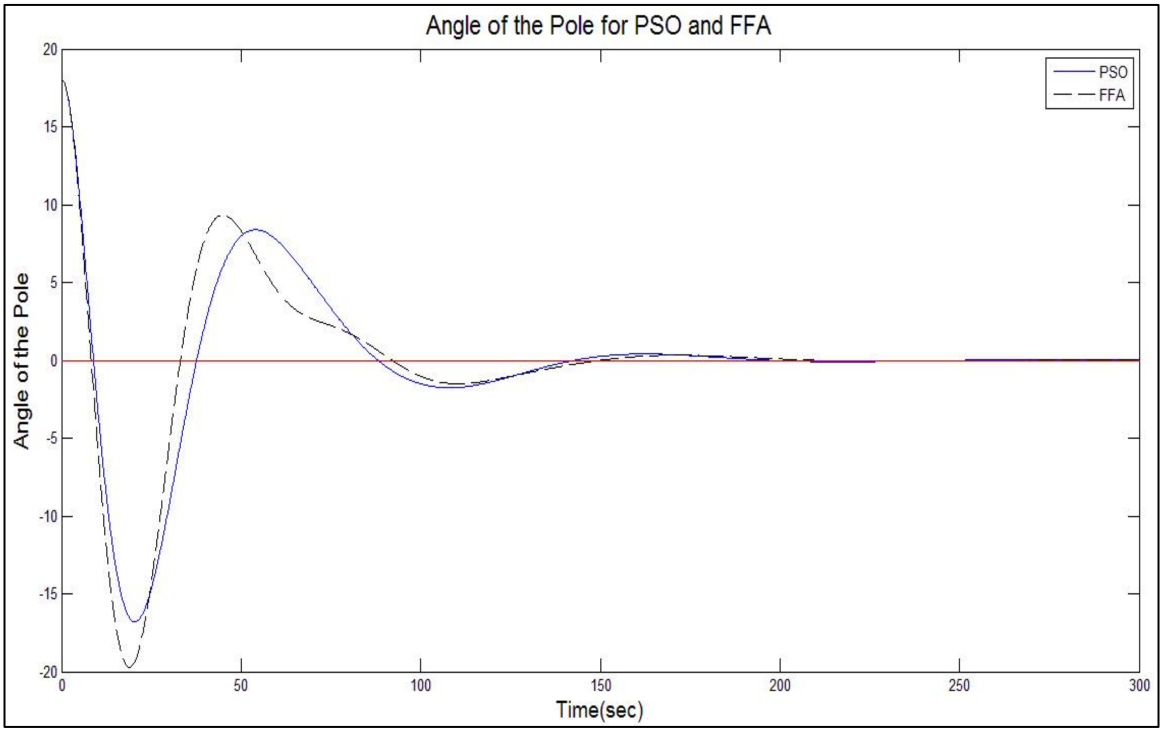

Figure 5.6: Angle of the pole with respect to vertical axis for both PSO and FFA

It can be clearly seen that pole angle for PSO has a less peak overshoot near 17^o while the FFA has a higher overshoot comparatively (20^o). Apart from that it can be seen that PSO settles faster than FFA as well. Particles are seen to be converging more smoothly than the fireflies. Both PSO and FFA converge to the reference trajectory successfully.

The variations in the angular velocity of the pole in the inverted pendulum system can be seen in figure 5.7 below for both PSO and FFA on a comparative basis.

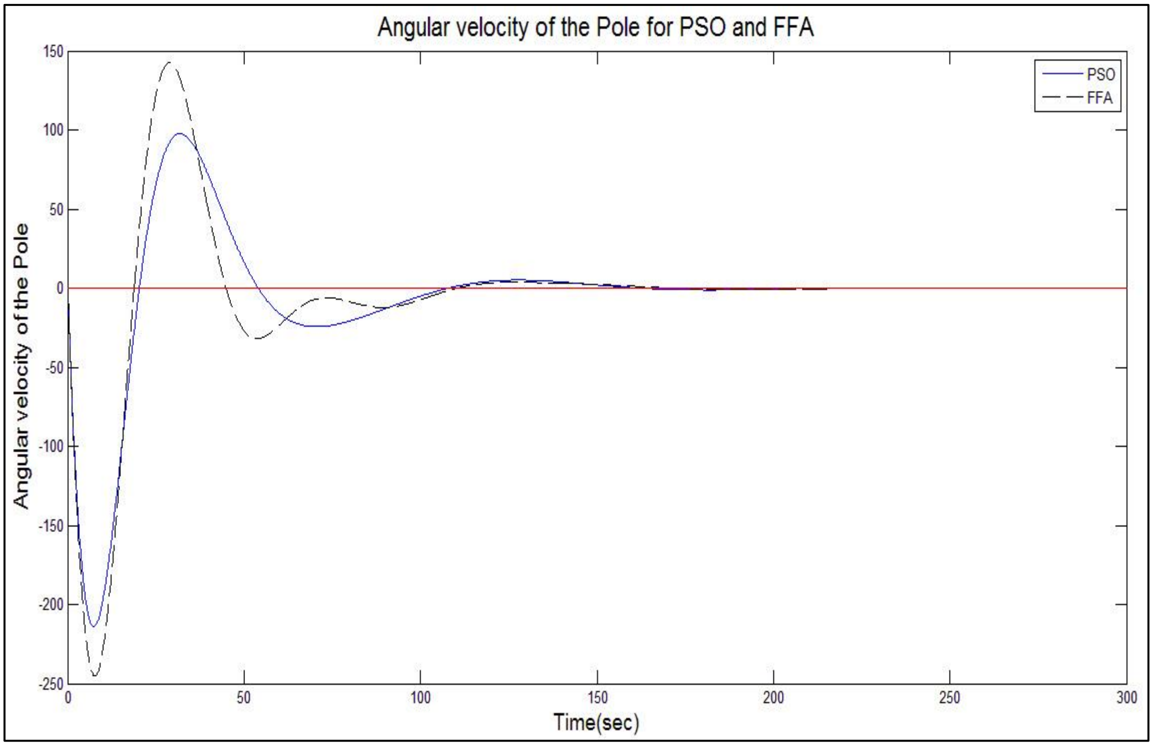

Figure 5.7: Angular velocity of the pole for both PSO and FFA

Since the state variable x_2 i.e. angular velocity is dependent on the pole angle deviation also, the variations in angular velocity are updated pertaining to each iteration and calculation of the pole angle. Thus, to achieve the goal of pole angle control, one should also make the angular velocity variation reach the reference trajectory i.e. the zero line.

From the above figure, it can be seen that oscillations in FFA are slightly more when compared to PSO which are also smoother than the former. Again, the peak overshoot in the case of PSO is seen to be less than that of FFA and the particles are seen to be settling faster than the fireflies. Subsequently, both PSO and FFA are successfully bringing the angular velocity to the desired reference trajectory.

The position of the cart for the inverted pendulum system is shown in fig 5.8 below for both PSO and FFA on a comparative basis.

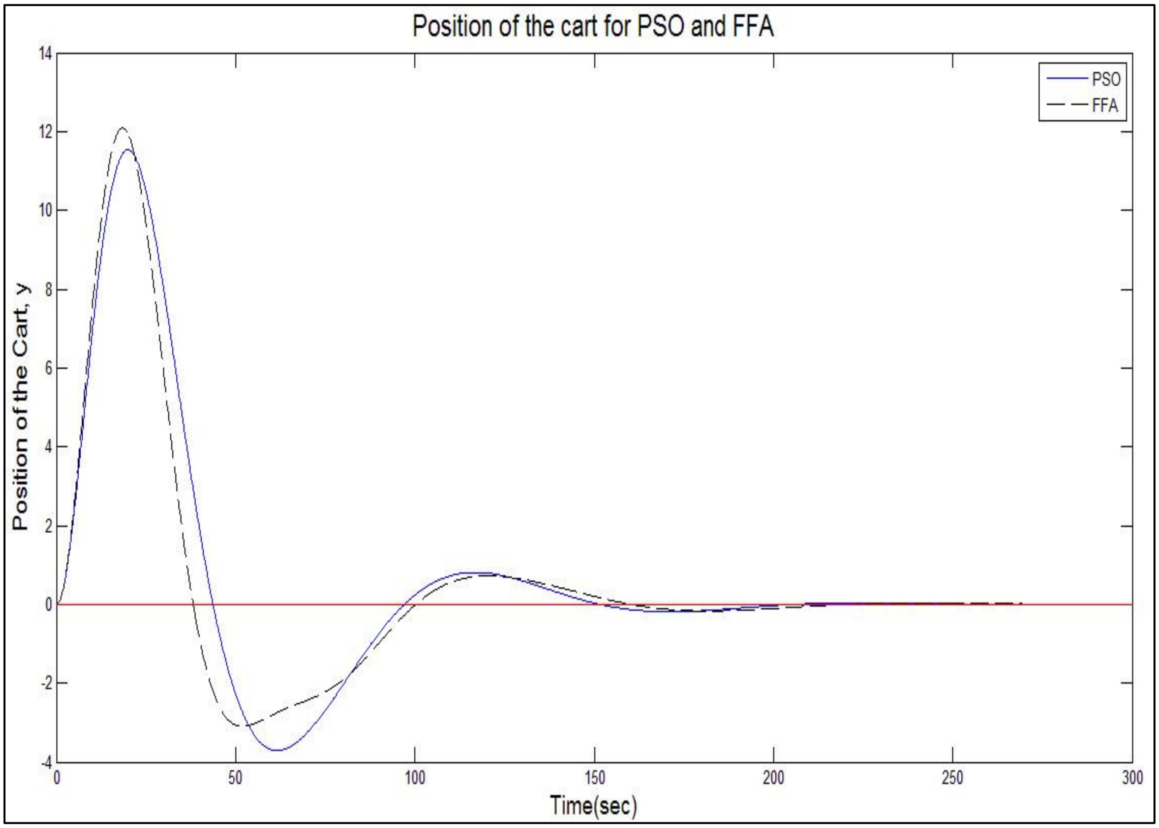

Figure 5.8: Position of the cart for both PSO and FFA

Since the state variable x_3i.e. cart position is dependent on the pole angle deviation also, the variations in position of cart are updated pertaining to each iteration and calculation of the pole angle. Thus, to achieve the goal of pole angle control, one should also make the variation in position of the cart reach the reference trajectory i.e. the zero line. As a matter of fact, the variation in the position of the cart has to be finite, therefore, it is considered that it moves in a track length of 30 centimeters. The midpoint of the track is considered to be the origin and the right hand side from the origin is taken to be positive which restricts the movement of cart within $[-15, +15]$.

From fig 5.8, it can be seen that again the PSO has a lesser peak overshoot than the FFA. Both the methods settle the cart to the desired reference, but particles outperform fireflies with a quicker settlement and a smoother response as well.

The variation in the velocity of the cart for both PSO and FFA is depicted in the folowing graph shown in fig 5.9.

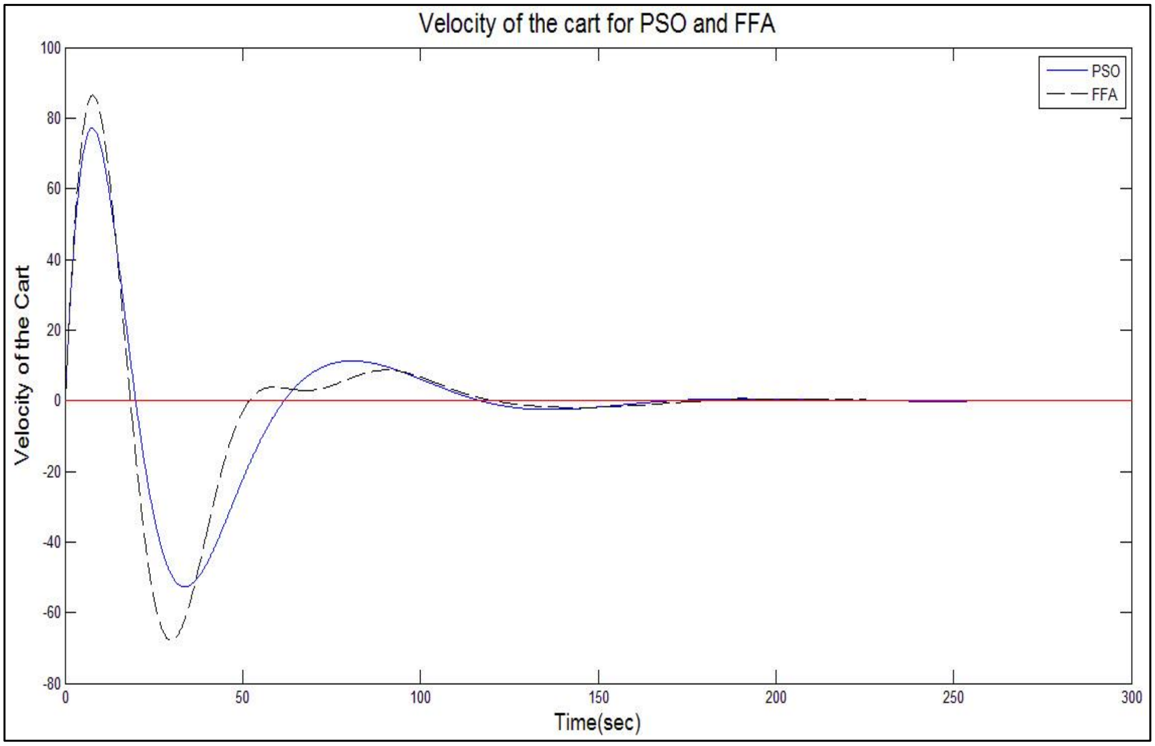

Figure 5.9: Velocity of the cart for both PSO and FFA.

Since the state variable x_4i.e. cart velocity is dependent on the cart position and pole angle deviation, the variations in velocity of the cart are updated pertaining to each iteration and calculation of the pole angle and cart position. Thus, to achieve the goal of pole angle control, one should also make the variation in velocity of the cart reach the reference trajectory i.e. the zero line.

From the above figure, it can be seen that both PSO and FFA successfully achieve the desired response, but PSO has a slightly smoother variation than the FFA as is evident between the time 50 to 100 secs in the graph. Also, the peak overshoot in case of FFA is again seen higher when compared with the case of PSO.

The error between the obtained response and the desired trajectory can be analysed with the help of fig 5.10 shown below.

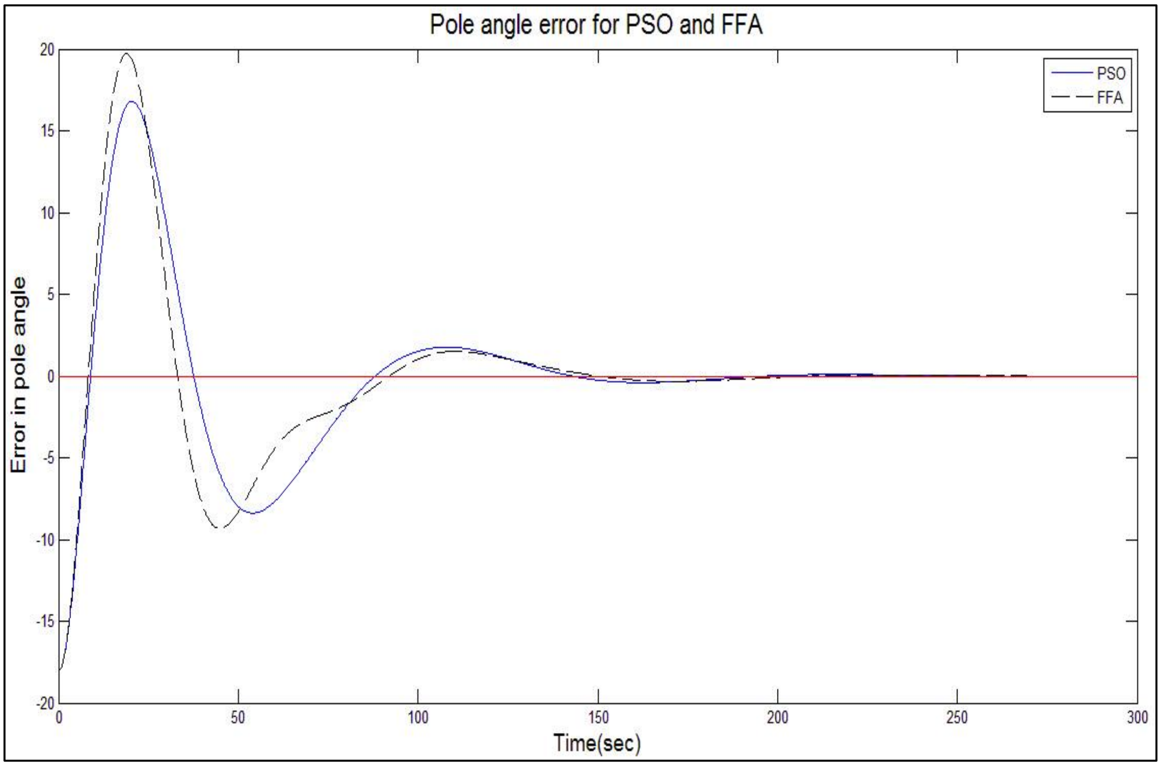

Figure 5.10: Error between the pole angle obtained and the reference for both PSO and FFA.

The figure above shows that the error between the reference and the pole angle deviation is approaching zero for both PSO and FFA. But, maximum error for the case of FFA (about 20) is seen to be more than the maximum error in case of PSO (about 17). If the time interval between 50 to 100 secs is considered, it can be seen that there is an abrupt increase and then decrease in the error value for FFA, while the error dynamics in the case of PSO is seen to be smooth enough.

Thus, it can be commented that PSO tracks the desired reference trajectory more closely than its counterpart i.e. the FFA.

Since the stabilization of the position of the cart is also an important factor for the control of the system, the error dynamics for the position of the cart with respect to the reference trajectory for both PSO and FFA is shown in fig 5.11.

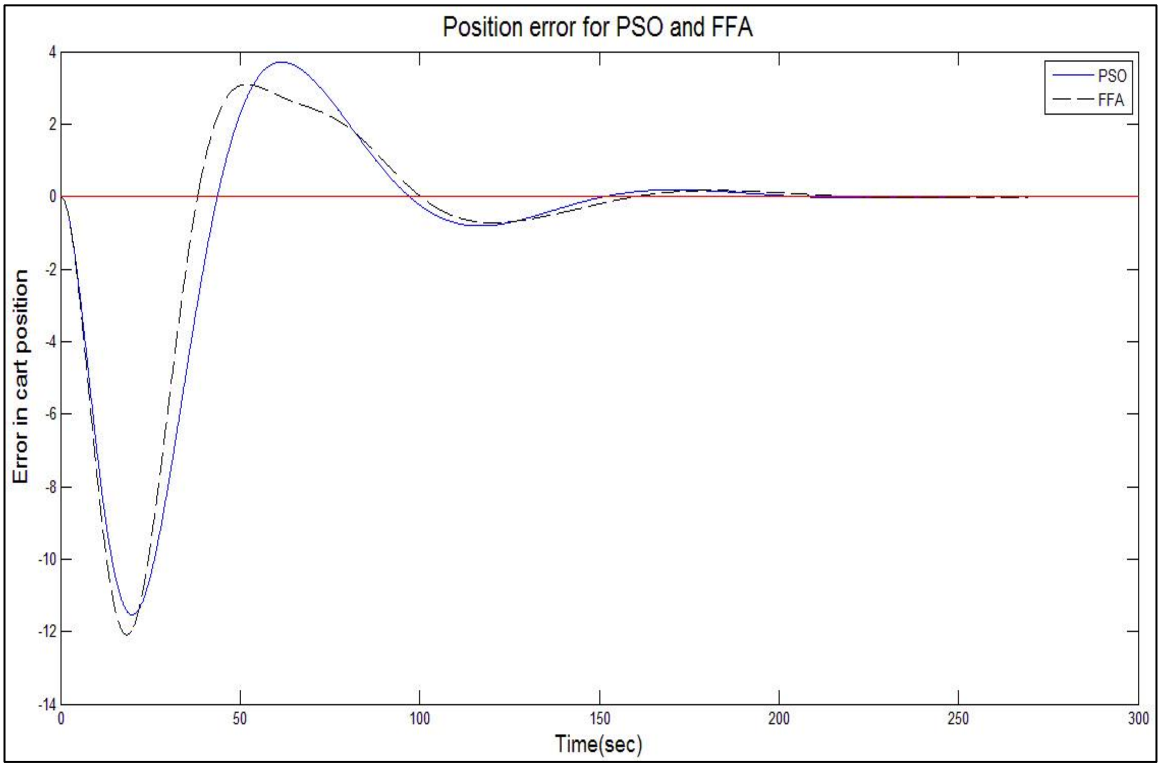

Figure 5.11: Error between the cart position obtained and the reference for both PSO and FFA.

It can be seen from the above figure that both PSO and FFA make the error between desired response and the obtained response for position of the cart equal to zero. Comparatively, the peak overshoot for PSO is seen to be less than that of the FFA which means that the error difference is less for PSO when compared with FFA. An abrupt variation in the error dynamics is seen between the time period 50 to 100 secs in the case of FFA while the dynamics is smooth for PSO.

Thus, the graph indicates that PSO tracks the desired reference trajectory more closely than FFA with minimum error overshoots.

The estimated value of the pole angle for both PSO and FFA method in order to implement indirect adaptive control is shown in fig 5.12 below on a comparative basis.

The initial estimates for both PSO and FFA were oscillatory in nature with extreme values, but gradually they were in tune with the desired response with smooth dynamics.

It is to be noticed that the estimate for the pole angle by the particle's best position in the case of PSO reaches the desired response faster than the estimates calculated by the best position of the fireflies in the case of FFA. This quick responsiveness in the estimation of the pole angle is a decisive factor in effectively controlling the pole angle in the pendulum system. It ultimately decides the speed of the controller as well which is also an essential factor for a control system.

For both the methods i.e. PSO and FFA, cost function minimization is the basis for selection of the best particles or the best fireflies respectively. The cost function variation for both the methods is depicted in fig 5.13.

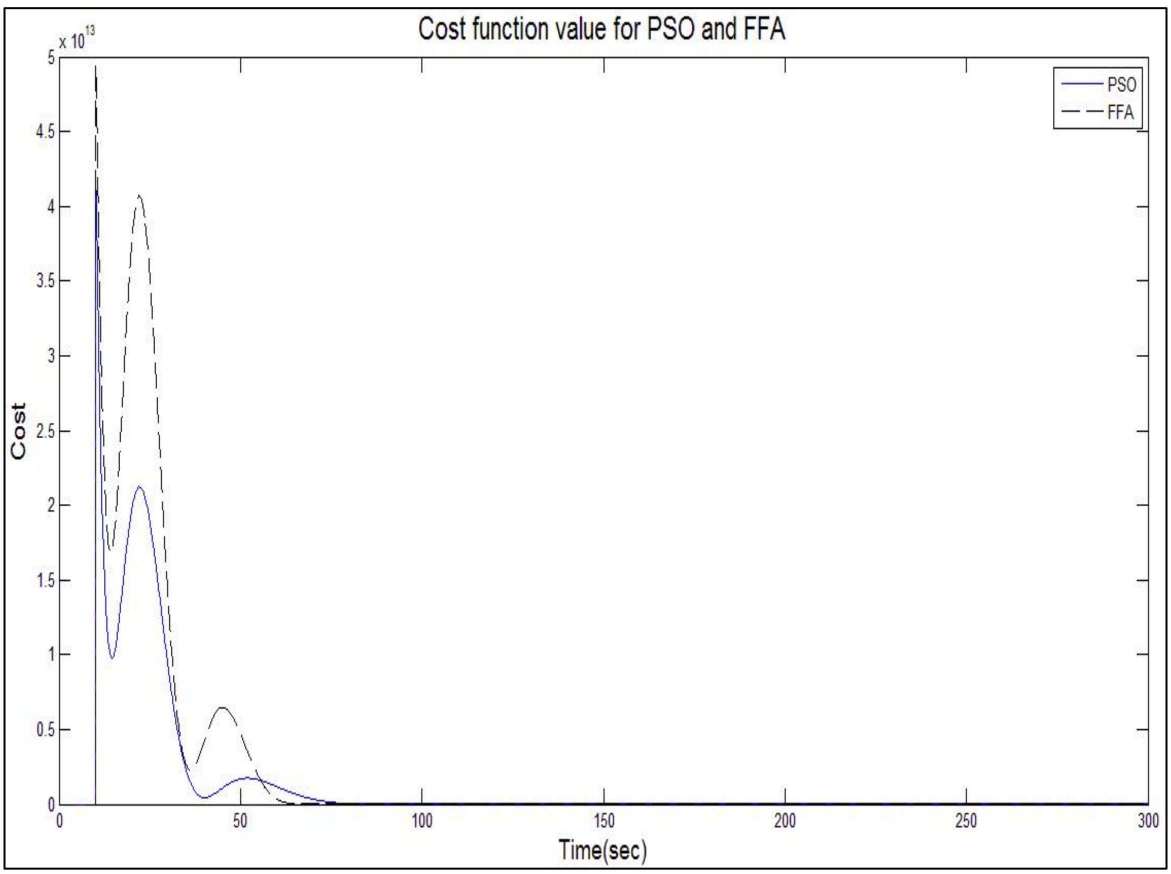

Figure 5.12: Cost function graph for both PSO and FFA for inverted pendulum

Fig 5.12 shows the cost function for the particles in PSO and the fireflies in FFA. In both the cases, the value of the cost function decreases to zero. The cost value trend in FFA contains lots of variations with large overshoot while approaching the minimum value but in PSO, although there are minor variations, the overshoot is much less when compared with FFA.

This shows that although the cost value for both the algorithm techniques reach a minimum value, the particle in PSO converge more effectively towards the best particle at a particular iteration than the fireflies in the FFA.

The table 5.2 below shows a comparative result on the performance of PSO and FFA on the liquid level control in surge tank system.

Table 5.2: Elapsed time and Minimum Cost for the inverted pendulum system

Algorithm	**Elapsed Time**	**Minimum Cost**
PSO	17.6038secs	0.001
FFA	105.7061secs	0.002

From table 5.2, it can be inferred from the data that elapsed time in PSO is much smaller than that in the case of FFA and also the minimum cost for PSO is also slightly smaller than that of FFA.

CHAPTER 6

CONCLUSIONS AND FURTHER SCOPE OF WORK

Indirect adaptive control is implemented on two nonlinear systems using two methods i.e. Particle Swarm Optimization (PSO) and Firefly Algorithm (FFA). The two nonlinear systems which were used for analysis were liquid level control in a surge tank system and pole angle control in an inverted pendulum system.

The results obtained with the liquid level control in a surge tank system show that both PSO and FFA were successful in controlling the liquid level in accordance with the desired trajectory. If analyzed on a comparative basis, it can be seen that in almost all the responses PSO gave less peak overshoots and comparatively less oscillations than FFA. The settling time for PSO is found to be lesser than the settling time for FFA. If the cost function is considered, it is found that particles in PSO gave a more minimum value than fireflies in FFA. It is also evident from the graphs that particles in PSO converge more smoothly towards their global best than the fireflies in FFA and thus the responses for PSO were much smoother than its counterpart. In all cases of iteration for the estimation of liquid height in the surge tank, it is found that PSO method for indirect adaptive control is much faster with an elapsed time of just 3.8429 secs when compared with FFA method for indirect adaptive control which has an elapsed time of 93.1496 secs. These observations converge to the fact that PSO method of indirect adaptive control performs better than the FFA method in the case of controlling liquid level in a surge tank.

The results obtained with the second system i.e. pole angle control in an inverted pendulum system show that again both the methods were successful in controlling the pole angle between $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and keeping it in upright position.If the analysis is done on a comparative basis, it can be seen that in the dynamics of all the four states i.e. pole angle, angular velocity, position of the cart and the velocity of the cart, PSO gave less overshoots and comparatively less oscillations than FFA. FFA takes more time to settle to the desired response than the PSO method in all the cases. It is also noticed that particles in PSO converge more quickly towards their global best when compared with the fireflies in FFA and thus the responses for PSO method were much smoother and quicker than its counterpart.

In all cases of iteration for the estimation of pole angle control in the inverted pendulum system, it is found that PSO method for indirect adaptive control is much faster with an elapsed time of just 17.6038 secs when compared with FFA method for indirect adaptive control which has an elapsed time of 105.7061 secs. The cost function value obtained with PSO is also slightly less compared with that obtained by using FFA. These observations converge to the fact that PSO method of indirect adaptive control performs better than the FFA method in the case of controlling pole angle to an upright position in an inverted pendulum system.

The error response for both the systems are seen to be converging to the zero line thereby indicating close tracking with the desired response by either method i.e. PSO or FFA. Thus, it can be concluded that PSO performs better than FFA for both the systems in terms of elapsed time, maximum peak overshoot and settling time. Although there is a very marginal difference in cost function value and smoothness of the response for both the methods, it can be clearly seen that both of them are equally successful in implementing indirect adaptive control for both the nonlinear systems.

Further Scope:

A brief research in the implementation of PSO and FFA in this area of indirect adaptive control will pave way for easy and effective control of more complex nonlinear systems where estimation of parameters is the most challenging task. This dissertation would prove useful for implementation of such indirect adaptive control techniques employing more different bioinspired algorithms for an effective and improved control of complex MIMO nonlinear systems.

Although basic firefly algorithm is very efficient but it can be seen that the solutions are still changing as the optima are approaching. Solution quality can possibly be improved by reducing the randomness.

Convergence of the algorithm can further be improved by varying the randomization parameter α so that it decreases gradually with the approaching optima. Recent studies show that the efficiency may be significantly improved if an extra term $\lambda\epsilon_i(g^* - x_i)$ is added to the position update formula for each firefly i given by the update equation (2.6). Here, λ is a parameter which is similar to α and β while ϵ_i is a set of random numbers.

PSO algorithms can outperform various other conventional algorithms for solving many optimization problems as can be evidently seen in various studies, which is partially due to that fact that the broadcasting ability of the current best estimates gives a quicker and better convergence towards the optimality. PSO algorithms are almost memoryless since they do not record the movement paths of each particle. Thus, it is expected that it can be further improved using short-term memory.

These variants of FFA and PSO can be tested upon various other nonlinear complex control problems which can be further analyzed to develop more new methods for efficient control and thus these could form important topics for further research.

REFERENCES

[1] Xin-She Yang, “Particle Swarm Optimization”, in Engineering Optimization, 1st ed., USA, J. Wiley and Sons, 2010, pp. 203-211.

[2] S.N. Sivanandam and S.N Deepa, “Introduction to Particle Swarm Optimization and Ant Colony Optimization”, Introduction to Genetic Algorithms, 1st ed., USA, Springer, 2008, pp. 403-424.

[3] H.Zang, S.Zhang, K.Hapeshi, "A Review of Nature-Inspired Algorithms", Journal of Bionic Engineering 7 Suppl.,2010, pp. 232–237.

[4] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory.” Proc.of 6th Int. Symp. Micro Machine and Human Science (MHS ’95), 1995, pp. 39–43.

[5] G.Theraulaz and J.L.Deneubourg, "On Formal Constraints in Swarm Dynamics", Proceedings of the 1992 IEEE International Symposium on Intelligent Control,1992, pp. 225-233.

[6] E. Bonabeau, M. Dorigo, G. Theraulaz, “Swarm Intelligence: From Natural to Artificial Intelligence”, NY: Oxford University Press, NewYork, 1999.

[7] Y. Zhao, Z. Zhang, Li Guo and B. Fang, "Revisiting the Swarm Evolution: A Long Term Perspective", IEEE Symposium on Computers and Communications (ISCC), 2011, pp. 1062 - 1067.

[8] B. Yang, Y. Chen and Z. Zhao, “A Hybrid Evolutionary Algorithm by Combination of PSO and GA for Unconstrained and Constrained Optimization Problems”, Proc of IEEE Int. Conf. on Control and Automation, 2007, pp. 166-170.

[9] J. Kennedy and R. Eberhart, "PSO", Proc.of IEEE Int. Conf. Neural Networks, vol. 4, 1995, pp. 1942–1948.

[10] B. Y. Y. Chen and Z. Zhao, "Survey on Applications of Particle Swarm Optimization in Electric Power Systems", IEEE International Conference on Control and Automation, 2007, pp. 481-486.

[11] Yamille del Valle, G.K. Venayagamoorthy et.al, "PSO: Basic Concepts, Variants and Applications in Power Systems", IEEE Transactions On Evolutionary Computation, VOL. 12, NO. 2, april 2008, pp. 171-195.

[12] D.N.Jeyakumar, T.Jayabarathi and T.Raghunathan, "Particle swarm optimization for various types of economic dispatch problems", Electrical Power and Energy Systems 28, Elsevier, 2006, pp. 36–42.

[13] T.Y.Chen and T.M.Chi, "On the improvements of the particle swarm optimization algorithm", Advances in Engineering Software 41, 2010, pp. 229–239.

[14] K.M. Passino,"Adaptive Control",in Biomimicry for Optimization, Control, and Automation, 1st ed.,london, Springer-verlag,2005, pp. 547-600.

[15] S.F. Zheng et.al, "A Modified PSO Algorithm and Application", Proc. of the Sixth Intnl. Conf. on Machine Learning and Cybernetics, IEEE, 2007, pp. 945-951.

[16] Z.Zhan, J.Zhang and H.S.Chung, "Adaptive Particle Swarm Optimization", IEEE Transactions on Systems, Man, and Cybernetics—Part B, December 2009, Vol. 39, pp. 1362-1381.

[17] B. Wayne Bequette, "Introduction", in Process Control: Modeling, Design, and Simulation,Prentice Hall Professional,NJ, USA, 2003, pp. 1-30.

[18] Li Zhi-jie, Liu Xiang-dong, Duan Xiao-dong and Wang Cun-rui, "An Improved Particle Swarm Algorithm for Search Optimization", Global Congress on Intelligent Systems, IEEE, 2009, pp. 154-158.

[19] M. O’Neill and A. Brabazon, "Self-Organizing Swarm (SOSwarm) for Financial Credit-Risk Assessment", IEEE Congress on Evolutionary Comp., 2008, pp. 3087-3093.

[20] Rong-Jong Wai, Jeng-Dao Lee and Kun-Lun Chuang, "Real-Time PID Control Strategy for Maglev Transportation System via PSO", IEEE Transactions On Industrial Electronics, Vol. 58, No. 2, FEBRUARY 2011, pp. 629-646.

[21] A.K.Yadav, P. Gaur, A.P.Mittal, M.Anzar, "Comparative analysis of various control techniques for inverted pendulum", India International Conference on Power Electronics (IICPE-2010) ,2011, pp. 1-6.

[22] Feras Al-Obeidat, Nabil Belacel, Juan A. Carretero and Prabhat Mahanti, "An evolutionary framework using PSO for classification method PROAFTN", Applied Soft Computing 11, Elsevier, 2011, pp. 4971–4980.

[23] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization", Proc. of the IEEE Swarm Intelligence Symposium (SIS), 2007, pp. 120 - 127.

[24] L.Cui and H.Wang, "Reach back Firefly synchronicity with Late Sensitivity Window in Wireless Sensor Networks," Proc. of IEEE Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 451-456.

[25] J. Jumadinova and P. Dasgupta, "Firefly-inspired Synchronization for Improved Dynamic Pricing in Online Markets," Proc of 2nd IEEE Intnl. Conf. on Self-Adaptive and Self-Organizing Systems, 2008, pp. 403-412.

[26] X.S. Yang, "Firefly algorithms for multimodal optimization, Stochastic Algorithms Foundation and Applications," SAGA 2009, Lecture Notes in Computer Sciences, 5792, 2009, pp.169-178.

[27] A. L.Christensen, R. O'Grady and M. Dorigo, "From Fireflies to Fault-Tolerant Swarms of Robots," Trans. of IEEE on Evolutionary Computation, vol. 13, no. 4, 2009, pp. 754-766.

[28] L. Coelho, D.Bernert and V.Mariani, "A Chaotic Firefly Algorithm Applied to Reliability-Redundancy Optimization", IEEE Congress on Evolutionary Computation (CEC), 2011, pp. 517 – 521.

[29] M. H. Horng and T. W. Jiang, "Multilevel Image Thresholding Selection based on the Firefly Algorithm." Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2010, pp. 58-63.

[30] S. Janson and M. Middendorf, "On Trajectories of Particles in PSO", Proc. of the IEEE Swarm Intelligence Symposium (SIS), 2007, pp:150-155.

[31] S. Nakamura, "Numerical Analysis and Graphic Visualization with MATLAB," Prentice-Hall, Englewood Cliffs, NJ, 1996.

[32] H. Someya, "Cautious particle swarm", IEEE Swarm Intelligence Symposium (SIS), 2008, pp. 1-6.

[33] A. W. Mohemmed, N. C. Sahoo and T. K. Geok, "Solving shortest path problem using PSO", Applied Soft Computing 8, Elsevier, 2008, pp. 1643–1653.

[34] O. Babaoglu, T. Binci and M. Jelasity, "Firefly-inspired Heartbeat Synchronization in Overlay Networks", First International Conference on SASOs , 2007, pp.77 - 86.

[35] James Kennedy, "Some Issues and Practices for Particle Swarms", Proc. of the IEEE Swarm Intelligence Symposium (SIS), 2007, pp. 162-169.

[36] Stellet, Jan. "Control of an Inverted Pendulum." chicago, 2011, pp. 1-14.

[37] El-Gammal, A.A.A.; El-Samahy, A.A., “A modified design of PID controller for DC motor drives using Particle Swarm Optimization PSO”, International Conference on Power Engineering, Energy and Electrical Drives, 2009 (POWERENG '09), 2009 , pp. 419-424.

Publications

[1] Bharat Bhushan and Sarath S Pillai, “Particle Swarm Optimization and Firefly Algorithm: Performance Analysis”, 3rd International Advanced Computing Conference (IACC-2013), published in IEEE Xplore, pp: 746-751.

[2] Bharat Bhushan and Sarath S Pillai, “Particle Swarm Optimization and Firefly Algorithm: Comparative Analyses”, National Electrical Engineering Conference (NEEC) conducted jointly by DTU and IEEE Delhi Chapter.