DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

CANDIDATE's DECLARATION

I Swati Sharma Roll No 2k16/C&I/20 student of M.Tech in Control and Instrumenation , hereby declare that the project Dissertation titled "**Adaptive control using Jaya Algorithm** " which is submitted by me to the Department of Electrical Engineering , Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: New Delhi **Swati Sharma**

Date:

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Adaptive control using Jaya Algorithm**" which is submitted by Swati Sharma, Roll No. 2K16/C&I/20 of Electrical Engineering Department , Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology in Control and Instrumentation, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: New Delhi **Prof. Bharat Bhushan**

Date: **SUPERVISOR**

ACKNOWLEDGEMENT

At the very outset, I would like to record my heartfelt gratitude to my respected supervisor Prof. Bharat Bhushan for his excellent guidance and motivation throughout the elaboration of this work. His practical insight and valuable suggestions have been a constant source of inspiration to me. This thesis is a result of his continuous advice and impetus during every stage of my research .

I shall be failing in my duty if I do not express thanks to faculty members of department of Electrical Engineering , Delhi Technological University , Delhi for their feedback and valuable comments during our regular presentations which helped me a lot in my master of technology degree.

I find myself spellbound to acknowledge thanks to my family for their silent support , patience , encouragement and affection without which this work would never have been possible.

I would also like to thank my friends at DTU,Delhi for their necessary and unconditional help during the project.

Finally , I wish to thank my almighty god , the constant source of mercy and compassion whose blessing has guided to make this study and completion of this work successfully.

ABSTRACT

There are very simple and powerful optimization algorithm which determine various constrained and unconstrained optimization problems. As optimization means many different attributes. There are various optimization techniques which are implemented for various non-linear models. Genetic algorithms and Jaya algorithms are widely used for model building , validation and policy designs for non-linear models. The main model on which Jaya Algorithm has been implemented is of Surge Tank. The improvement in these algorithms brings necessity in evaluation of the efficacy of the surge tank. This is the main purpose of this thesis.

This Thesis carries the work on implementing the Jaya Algorithm and Genetic Algorithm with Adaptive control on Surge tank. The different Algorithms are also compared with Jaya Algorithm for different benchmark functions. The common control parameters are required by Jaya Algorithm but not the algorithm specific control parameters.

The Genetic Algorithm and the Jaya Algorithm are also implemented on Surge tank with adaptive control to obtain the system dynamics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

SYMBOL	DESCRIPTION
h(t)	Height of the Surge Tank
u(t)	Input of the Tank
t	Sampling Time
α,β	Non-linearities

LIST OF ABBREVIATIONS

PSO	Particle Swarm Optimization
GA	Genetic Algorithm
HS	Harmony-Search Optimization
TLBO	Teaching and learning based Optimization
ABC	Artificial-Bee Colony Optimization
BFO	Bacterial Fore-ageing Optimization

CHAPTER 1
INTRODUCTION

1.1 OPTIMIZATION TECHNIQUES

Optimization Techniques are used everywhere, from engineering designs to financial markets and from our daily activity to planning our holidays and computer sciences to industrial applications.

In every aspect we always try to maximize or minimize elements for better results, profits and performances. Just to find the solution of optimization problem, whether intentionally or subconsciously determines as old as human history [34]. So many physical phenomena are used in governing of action principle or its variants.

Optimization include the wide range of problems with the aim of searching for certain optimality. The optimization techniques varies from system to system or we can say problem to problem and the complexity of the optimization problems depends on function to function which are formed from its objective functions and constraints [31].

This thesis not only deals with Genetic Algorithm but also based on Jaya Algorithm by comparing the system dynamics with other algorithms such as Harmony Search, Teaching learning based optimization algorithm , Particle swarm Optimization and Genetic algorithm by comparing them with four Benchmark functions[3 , 10].

Thus, the Genetic Algorithm and Jaya Algorithm are also implemented for Non-Linear Liquid Level control for Surge Tank just to determine the system dynamics , its performance and the Compared Results[5 ,15].

The Evolutionary and Swarm Intelligence based Algorithms are Probabilistic algorithms and require common controlling parameters such as population_size , Number of generations and Elite size and many more.Besides all these evolutionary based methods , some Algorithms requires their own algorithm-specific control parameters[17].

Different usage criteria for different Algorithms are as follows:

GA uses selection operator , crossover probability and mutation probability[5,15]. PSO uses inertia weight , social and cognitive parameters[3,10]. ABC uses number of onlooker bees , employed bees, scout bees and limit[22,29]. HS uses harmony memory considerations rate, pitch adjusting rate and the number of improvisions[38]. Jaya Algorithm uses the best and the worst value of the updated population[1,8]. TLBO uses Teacher phase and learner phase[2,24].

Optimization techniques consists of different type of algorithm such as Particle-Swarm Opimization , Genetic Algorithm , Harmony search Optimization , Jaya Algorithm, Teaching and learning based algorithm , Artificial Bee-colony and Bacterial Fore-ageing and many more[1,8,19,20,22,29].

All these Algorithms are widely used for different types of non-linear model and are widely improvised with different performances and characteristics. There are different ways in classification of optimization problems. The optimization is classified as follows:

(i) Objective : (i) Single Objective. (ii) Multi Objective.

(ii) Constraint : (i) Unconstrained (ii) Constrained

(iii) Landscape : (i) Unimodal(convex) (ii) Multimodal

(iv) Function Form : (i) Linear (ii) Non-Linear

(v) Variables/Responses : (i) Discrete (ii) Continuous (iii) Mixed

(vi) Determinancy : (i) Deterministic (ii) Stochastic

The different types of functions used for these algorithms are Easom function ,schwefel function, Himmelblau function, Booth function, Rastringin function Sphere Function [40].

Thus, all the evolutionary and swarm intelligence based algorithm require proper tuning of algorithm-specific parameters in addition tuning of common controlling parameters[17].

Jaya Algorithm when compared with other TLBO , GA , PSO AND HS [1,8,19,20]and proved the best algorithm among all. Jaya Algorithm even used to solve for constrained benchmark function[38]. Thus, the process parameters has to be set to obtain better performance results through Jaya Algorithm.

1.2 PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization was developed by Kennedy and Eberhart in 1995 which is based on swarm behaviour such as fish and bird schooling in nature. There are so many algorithms which uses the behaviour of swarm intelligence such as Ant Colony Algorithm and Ant Algorithms[3]. It is a very simple technique as it mostly similar to Genetic Algorithm but it never undergoes for mutation/Crossover operators or pheromone. As , it uses the randomness of real numbers and the global communication among the swarm particles.

It is easier because it does not implement encoding or decoding of the parameters into binary strings as those in genetic algorithms.

The particles are defined in PSO is that the space of an objective function by adjusting the trajectories of individual agents. These trajectories form the piece wise paths in a quasi-stochastic manner[10].

Two major components of PSO are:

(i) Stochastic component

(ii) Deterministic component

We have different constraints for implementing the unconstrained problems. For constrained optimization,[28] there are many ways in obtaining the constraint inequalities and equalities as there are two approaches for determining the constraint optimization i.e.

(i) Direct Implementation.

It is basically used to determine the system new particle location just to see whether they can satisfy the constraints or not.

(ii) Transformation.

In this constraints are replaced by new generated locations such that they can obtain the better results for different benchmark functions for unconstraint optimization problems.

1.3 GENETIC ALGORITHM

Genetic algorithm is a metaheuristic algorithm which was developed by John Holland in 1960s and 1970s. It is a model or abstraction of biological evolution based on Charles Darwin Theory of natural selection and the Holland used the recombination and crossover , mutation and selection in the study of adaptive and artificial systems. These genetic operators formed the essential part of GA as problem-solving strategy.

The optimal solution of the problem does not guarantee by Genetic Algorithm however all its empirical evidences are between acceptable levels, in a competitive time with the rest of all other optimization techniques such as simulated annealing , sequential search methods , hyper climbing , Jaya Algorithm , TLBO, Harmony-search etc. Genetic Algorithm does not make any kind of assumptions for any other optimization techniques[5,15,16,18]. Genetic Algorithm are applied to solve the wide range of engineering and scientific optimization discussions. Just to understand the basic functionality of the genetic algorithm we have to understand the system and modelling of the Genetic algorithm and vary all its optimization problem variables and then have to recombine to obtain better results and better characteristics of the system[30]. The main objective of the system using Genetic algorithm can find new set of parameters that are produced by optimum values in the variables to the optimization problem. If a variable encoding is decided , the first main step will be to encode the initial population of the GA with random encoded parameter values just to form n number of chromosomes[32].

The second step is to evaluate the order of the measurable value that indicates all the set of parameters to obtain the solution of the system.

The selection operators of GA gave more systematic system accuracy to the chromosomes for the optimization problem. Selection and crossover operators always try the combination of parameters just to obtain better results for non-linear models for better system characteristics.

A new set of parameters must be chosen to improve the current value of criteria function in all optimization algorithm[6] . Genes were mixed with the crossover operators

The thirty three ne off springs were generated by mixing it with the strings of parameters which represents the chromosomes for the two parents.

There are three different crossover techniques such as:

(i) One-point crossover technique.

(ii) Two-point crossover technique.

(iii) Uniform crossover technique.

1.4 HARMONY-SEARCH OPTIMIZATION

This optimization algorithm was first developed Z.W.Geem et.al. in 2001 as it is relatively a new heuristic optimization such that it helps in determining the solution for different optimization problems[38]. This algorithm can easily combine with other algorithm to obtain better optimization results. Harmony-search can be explained by improvisation process by musician and four possible choices can be determined i.e.

(i) A series of pitches in harmony through memory.

(ii) Adjusting the pitch..

1.5 TLBO

Teaching-Learning Based Optimization (TLBO) is a new optimization method which is proposed for the mechanical and electrical design problems such that this method works on the basis of learners through teachers[2]. Various other algorithms are naturally inspired , Teaching-Learning based optimization which is based on population and even the population obtain a solution globally. The population is considered The group of learners are considered as population based classes of learners[24]. The TLBO is divided as follows :

(i) The first phenomena is 'Teacher Phase'.

(ii) The second phenomena is 'Learner phase'.

'Teacher Phase' is known for learning through teachers and 'Learner Phase' is known by learning through learners. TLBO is explained as follows:

(i) TLBO method get tested on four different benchmark function just to determine the effectiveness for its different characteristics.

(ii) Average solution , convergence rate , best solution and computational effort are the main characteristics to determine the effectiveness of system.

(iii) It is better optimization technique as compared to other optimization problems for better and efficient results.

(iv) This method is easily used in designing of mechanical and electrical problems for better characteristics and better optimization results.

(v) TLBO method not only designed for better characteristics but it determines two phases in a single algorithm which make it more efficient and better from other algorithms.

1.5.1 TEACHER PHASE

In this phase the teacher teaches the first phenomena of the algorithm to all learners. During this phase the teacher teaches the learners the mean results about the subject on the basis of his capability[25]. At any iteration 'i ' the main assumption leads to 'm' total subjects i.e. all the variables such that 'n' determines the total learners(i.e. size of the population , k=1,2,…..,n) and Mj,i be the mean result of the subject learned by the learners 'j'(j=1,2,…..,m) . the best overall result can be calculated as follows:

Xtotal-Kbest , i (1.1)

When all the subject are considered together just to obtain the total population of learners such that the best learner result can be determine by kbest. The highly learned person are the teachers who teaches the learners and help them to obtain the better results such that they can have better learning which is important for the consideration for the TLBO algorithm as a teacher[26].

In this phase the mean result of every subject is directly directed to the corresponding result shown by the teacher in every subject which is given by:

$$Diff\_Mean_{j,k,I} = r_i (X_{j,kbest,I} - T_F M_{j,i}) \quad (1.2)$$

$X_{j,kbest,I}$ is used to determine the best result in subject j. T_F is the teaching factor which decides the changed mean value for obtaining the better results whereas r_i is known as the random number for a specific range of [0 ,1]. All the values of T_F cn be decided for 1 or 2.

$$T_F=round[1+rand(0,1)\{2-1\}] \quad (1.3)$$

T_F in TLBO algorithm is parameter-less such that it even not act as the algorithm input and random values so decided in this algorithm to obtain better results[27] . However, the performance of TLBO found better value of T_F which may be 1 or 2 thus, this algorithm simplifies the system teaching factor so suggested just by taking 1 or 2 as it all depends not only rounding-up criteria but also $Diff\_Mean_{j,k,I}$. As all the desired solution is updated for the teacher phase .

$$X'_{j,k,i} = X_{j,k,i} + Diff\_Mean_{j,k,i} \quad (1.4)$$

Where, $X'_{j,k,I}$ is determined as the updated value for the $X_{j,k,i}$- $X'_{j,k,I}$ which is accepted as the function value and all the values will be accepted in the end of the phase by the teachers . All the values can be maintained and became the part of the input in the learner phase. The teacher phase is the main process on which learner phase is dependent[2,26].

1.5.2 LEARNER PHASE

This phase increases the learners knowledge through interaction with themselves as it is the second part of the algorithm such that the learner interaction with other learners just by improving the knowledge. The learner whose knowledge is better than other learners in terms of new things than just by considering size of population ‘ n ‘ , the learning phenomena of phase can be determined by the knowledge of learner to learner[27,24].

Just by considering the learners Pand Q randomly the :

$X'_{total-P,i} \neq X'_{total-Q,i}$ (where, $X'_{total-P,i}$ and $X'_{total-Q,i}$ these are the updated function values of $X_{total-P,i}$ and $X_{total-Q,i}$ of P and Q respectively at the end of teacher phase).

$$X''_{j,P,i} = X'_{j,P,i} + r_i\,(X'_{j,P,i} - X'_{j,Q,i}),\ \text{If } X'_{total-P,i} < X'_{total-Q,i} \quad (1.5)$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i\,(X'_{j,Q,i} - X'_{j,P,i}),\ \text{If } X'_{total-Q,I} < X'_{total-P,i} \quad (1.6)$$

$X''_{j,P,I}$ is accepted if it gives a better function value.

The Eqs. (3.7) and (3.8) are for minimization problems. In the case of maximization problems, the Eqs. (3.9) and (3.10) are used.

$$X''_{j,P,i} = X'_{j,P,i} + r_i\,(X'_{j,P,i} - X'_{j,Q,i}),\ \text{If } X'_{total-Q,i} < X'_{total-P,i} \quad (1.7)$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i\,(X'_{j,Q,i} - X'_{j,P,i}),\ \text{If } X'_{total-P,i} < X'_{total-Q,i} \quad (1.8)$$

1.6 LITERATURE REVIEW

Jaya Algorithm is a simple and more powerful optimization technique for solving the constrained and unconstrained optimization problems. This algorithm has a concept of rejecting the worst solution and obtain the best solution for the given problem. Jaya algorithm provides the effective results and better performance as compared to other optimization technique. Jaya algorithm is implemented on different controllers and devices .

In [43], the author applied the Jaya algorithm on automatic generation control using fuzzy-PIDF controller such that this work is proposed for the AGC of a two area power system . The five gain constants and the filter derivative constant is implemented on Jaya algorithm. The controller in this technique minimizes the frequency deviations and the tie line power deviations within a short span of time and with small values of peak overshoot and peak undershoot. The comparison is obtained between fuzzy-PID controller and the fuzzy-PIDF controller by tuning with Jaya algorithm. The comparison determine the fuzzy-PIDF controller is better and more efficient with Jaya algorithm.

In [44], the author applied Jaya algorithm for Tea category identification using a novel fractional fourier entropy such that this technique introduces this novel feature of fractional fourier entropy (FRFE) and its effectiveness is determined for tea images and even a novel classifier is determined as Jaya-FNN which is combined with feedforward neural network to determine better features and better results. This method is also applied to X-ray images , magnetic resonance imags , microglia images and the Alzheimer's disease images.

In [45], the author applied the Jaya algorithm in urban traffic signal control such that this algorithm determines the different types of lights used in traffic signal by automatic updating it with the Jaya Algorithm such that the blinking of lights will take place within a short span of time and obtain the better results as compared to other optimization algorithm used in implementing the urban traffic signal. The performance characteristics so obtained for the system is better and efficient than other optimization algorithms.

In [24] , the author describes Jaya algorithm with different benchmark functions for constrained and unconstrained optimization problems such that the best and the worst values are obtained .

In [46] , the author determines the system consistency with Jaya algorithm such that variable power can be calculated and the other electric power sources will determine the reliable power supply . Using Jaya algorithm the wind power proportion is calculated just to obtain the power . The best and worst power value is calculated using Jaya algorithm to upgrade the grid and lowering the ability to supplement the conventional production of variable power. Using Jaya algorithm the power management is done in wind power plants , in addition it determines the weather forecasting and permits the electric power network to produce more and more variations in production of variable power.

In [47], the author determines the facial recognition based on emotion recognition expressions proves to be the important field for effective computing. Using Jaya algorithm the problem is resolved using a novel intelligent emotion recognition system. The stationary wavelets are used to extract a single hidden layer using feed forward neural network such that the Jaya algorithm helps in preventing the training of the classifiers which will not fall into the local optimum points.

In [48], the author determines the photovoltaic models parameters based on the voltage characteristic curves which is significant for the simulation , control of the system and the evaluation. Of the system. The Jaya algorithm is proposed in the PV models such that to adjust the tendency for approaching the best solution and avoiding the worst solution at different stages and improves the exploration ability of the system. The Jaya algorithm used in PV models solves the parameter identification parameters such as PV module comprehensive experiment, single diodes and the multiple diodes. It determines the system accuracy and reliability of the system using Jaya algorithm.

In [22], the author determines the G-Jaya which is proposed to estimate the graphics processing unit (GPU) of the Li-ion battery model which helps in tuning the algorithm specific parameters using Jaya and the procedures so used in this system are the solution updates , the fitness value and the best/ worst values used in determining the system GPU.

All the experimental results determines the estimated battery model parameters and reduces the execution time using entry level and professional level of GPU's.

This dissertation mainly comprises of five chapters which basically determines the adaptive control on Surge Tank system using Jaya algorithm and GA algorithm to determine the system non-linearities and the liquid level'h'. The main objectives of dissertation is to :

- To stabilize the liquid level surge tank by imposing physical constraints.
- To develop controllers which also impart robustness to the non-linear system , by learning different algorithm such as the latest Jaya Algorithm which helps in improving the system tendency.

- To obtain the parameters of surge tank using Jaya Algorithm such as h (height) , α and β (non-linearities) of the system such that the worst value and the best value of the system will be determined.

In the above desired system a careful and critical analysis of all the papers the following research gap findings are obtained such as:

- Jaya is faster than GA and has been employed for all other optimization technique but degradation in its efficiency is a major concern and can be overcome by using Jaya Algorithm[12,13].
- GA needs to converge data optimally by using variance of data[32].
- Jaya algorithm helps in solving many problems which are not solved by BFA and GA , e.g, by calculating the Best Value and the Worst value and the Standard deviation of the system[20] .
- Although Jaya algorithm is promising and has successful applications , number of challenges still remain like handling the learning rate where learning of the agent happens directly from the experiences in operational environments.

The suggested approach for adaptive control using Jaya algorithm on Surge tank obtain most suitable results using Jaya Algorithm . Our motivation for this novel is based on the approach that Jaya Algorithm has very few requirements:

(i) An appropriate mapping between the best value and the worst value using nonlinearities such as alpha and beta.

(ii) An appropriate fitness function.

(iii) Jaya Algorithm has been used with adaptive control on surge tank such that it has descent convergence speed and high accuracy and better convergence as compared to

Genetic Algorithm . As it provides better accuracy and better performance characteristics results as compared to genetic algorithm and fuzzy –logic system .

CHAPTER 2

JAYA ALGORITHM

2.1 INTRODUCTION

Jaya Algorithm which is a newly evolved optimization technique , this algorithm is introduced by R.Venkat Rao in year 2016[1]. Jaya algorithm is a single phase technique and simpler as compared to other algorithms such that this algorithm quite similar to that of Teaching-Learning Based algorithm such that it has better performance characteristics and provide better results as compared to other optimization techniques.

Let us take f(x) as an objective function which has to be minimized or maximized such that taking an iteration' i' ,' m' as number of variables and' n 'as candidates solutions by taking size of population as k = 1,2,…..,n.

The basic term best will be used to obtain the best value of the population and the term worst determines the worst value of the population[8].

$$\mathbf{X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\,(X_{j,best,i} - \left|X_{j,k,i}\right|) - r_{2,j,i}\,(X_{j,worst,i} - \left|X_{j,k,i}\right|)} \qquad (2.1)$$

Such that,

$X_{j,best,j}$ is the value of variable.

'j' act best among all the candidates .

$X_{j,worst,j}$ is the variable values . In this term 'j' acts as worst among all the candidates.

$\mathbf{X'_{j,k,i}}$ is the updated value for $\mathbf{X_{j,k,i}}$.

When all the iterations are accepted than these functional value will be used as input to the next iteration. This theorem strives the best result and hence termed as Jaya such that the Sanskrit meaning of Jaya means Victory or triumph[19].

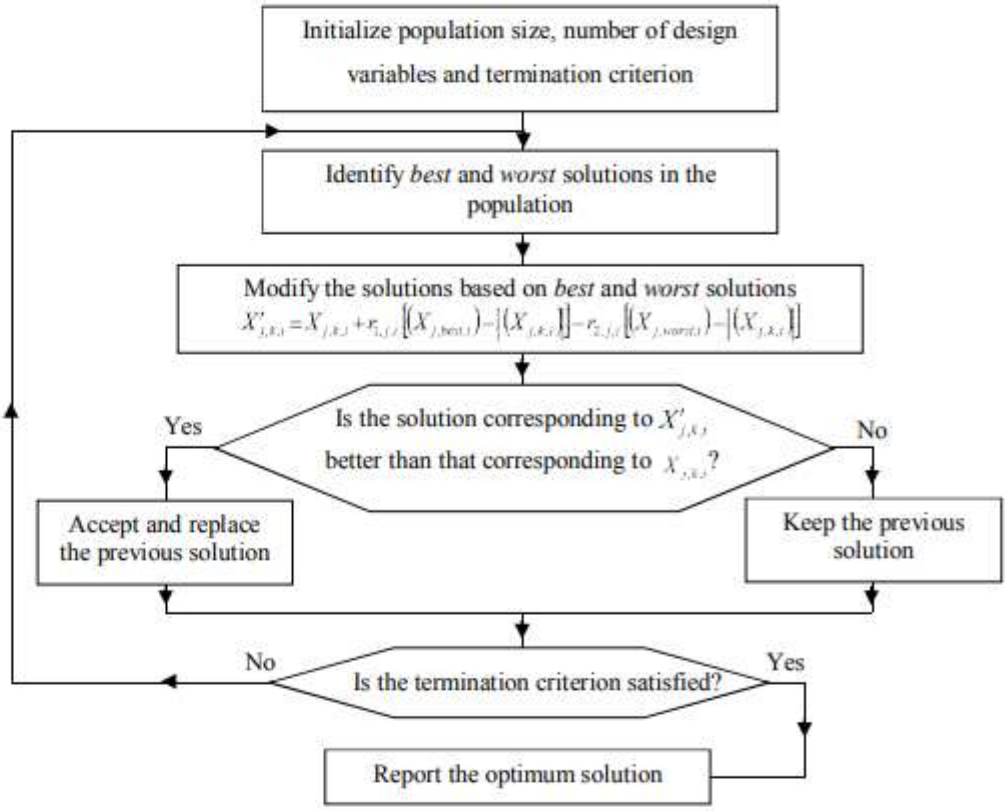

Fig 2.1 Flow chart describing Jaya Algorithm

The above flow chart determines how to obtain the solution by initializing the population size , number of design variables and termination criterion so that the system will provide the best value and the worst value as shown in Fig 2.1.

PSEUDO CODE OF JAYA ALGORITHM

- ***The count of updated population (Xj) is an estimate of the solutions either worst value or the best value of a particular solution k. For a particular front F, let j =|N| then for each member in F, X desired value is calculated as follows.***
- ***Initialize X_k= 0***
- ***Sorting all solutions in N for estimating the worst order of objective function value f_m.***
- ***In the given list of m^{th} objective assigned for infinite distance to solutions at the extremes of the list (i.e. $X_1 = X_z = \infty$), for k = 2 to (j – 1), calculate Xj as follows:***

$$X_k = X'_k \frac{f_m^{j+1} - f_m^{j-1}}{f_m^{max} - f_m^{min}} \tag{2.2}$$

2.2 DEMONSTRATION OF THE WORKING OF JAYA ALGORITHM FOR A SPHERE FUNCTION

The Jaya Algorithm demonstration on a sphere function which is an unconstrained benchmark function such that the objective function determines the values to maximize or to minimize such as :

$$\min f(x_i) = \sum_{i=1}^{n} x_i^2 \tag{2.3}$$

Subject to -100≤x_i ≤100

Now , let us assume the size of the population to be 10 (i.e.taking two varaiables x_1 and x_2 and two iterations by using tables we can determine the initial population to determine the objective function[20] . through this we can calculate the best and the worst solution such that we can obtain the mean population.

Table 2.1 Initial Population determined for sphere function

Candidate	X_1	X_2	f(x)	Status
1	-5	18	349	
2	14	63	4165	
3	70	-6	4936	Worst
4	-8	7	113	Best
5	-12	-18	468	

From above Table 2.1 the best solution is obtained by fourth candidate and similarly the worst solution is calculated by the third can be determined. Thus r_1=0.58 and r_2=0.81 for x_1 and for x_2 the value of r_1=0.92 and r_2=0.49 . By initializing the value of the population size , the design variables and the termination criterion . Through this we can determine the best and worst solutions and the mean of the system and the standard deviation of the system.

$$X'1,1,1 = X1,1,1 + r1,1,1\,(X1,4,1 - |X1,1,1|) - r2,1,1\,(X1,3,1 - |X1,1,1|) \quad (2.4)$$

$$= -5 + 0.58\,(-8 - |-5|) - 0.81\,(70 - |-5|) = -65.19 \quad (2.2.1)$$

$$X'2,1,1 = X2,1,1 + r1,2,1\,(X2,4,1 - |X2,1,1|) - r2,2,1\,(X2,3,1 - |X2,1,1|) \quad (2.5)$$

$$= 18 + 0.92\,(7 - |18|) - 0.49\,(-6 - |18|) = 19.64 \quad (2.5.1)$$

Thus the table 2.2 determines the new values and the objective function of x_1 and x_2 and thus first iteration for the function has been determined. During first iteration , the new values of the variables and the objective function are as follows.

Table 2.2 The new values with the objective function (sphere)

Candidate	X_1	X_2	f(x)
1	-65.19	19.64	4644.466
2	-44.12	45.29	3997.76
3	24.76	0.8	613.697
4	67.6	13.37	4735
5	70.58	-16.36	5249.186

Thus, table 2.3 determines the system best and the worst value for the function such as x_1 and x_2. At the end of first iteration , for the updated values of the variables and the objective function based on fitness comparison[1,8]

Table 2.3 The system best and the worst value

Candidate	X_1	X_2	f(x)	Status
1	-5	18	349	
2	-44.12	45.29	3997.76	Worst
3	24.76	0.8	613.697	
4	-8	7	113	Best
5	-12	-18	468	

For 1st iteration :

Best Value :113

Worst value : 4936

For 2nd iteration

Best Value = 113

Worst Value =3997.76

2.3 CONCLUSION

The newly proposed Jaya Algorithm has strong potential to solve all types of constrained and unconstrained optimization problems such that this algorithm is simple to apply as compared to other algorithm , it has no specific algorithmic parameters and it provides results with optimum values.

CHAPTER 3

THE BENCHMARKS CONSIDERED

3.1 INTRODUCTION

The hybrid control techniques and the algorithms which are used in previous chapters which are intend to devise the process control problem using indirect adaptive control on Liquid Level control Surge Tank so as to make system more stable and robust and to determine system parameters such as height , alpha and beta which are the non-linearities of the system[17].

In order to elucidate the supremacy of the system we are determining two algorithms for the liquid level control surge tank problem :

- Jaya Algorithm
- Genetic Algorithm

Basically , the main difference lies in both the system is to elucidate the values of Height of liquid level surge tank , and the non-linearities in case of Genetic algorithm .

For Jaya algorithm to determine the values of Height of the liquid level control , best value , worst value and the mean value of the system for the non-linearities such as alpha and beta[11,35].

3.2 ADAPTIVE CONTROL

An adaptive control strategy which is widely applicable for Adaptive Control such that it combines with the plant models in closed loop by redesigning the controllers as shown in fig. There are various techniques which are used in adaptive control so that the system become stable and robust. All the applications used in indirect adaptive control are flexible and highly appropriate. These controller are used for different non-linear systems[35].

The non-linear system on which this indirect adaptive control technique is implemented is liquid level control surge tank using Jaya algorithm and [39] Genetic Algorithm as shown in Fig 3.1.

Let us discuss about the nominal motion of indirect adaptive control which is as follows:

Let x*(t) is the main solution for the nominal trajectory such that the initial condition which determine the system are: $X^*(0)=x_0$ (3.1)

Such that $\dot{x} = f(x)$ and the initial condition are $x(0)=x_0+\delta x_0$

The error dynamics can be calculated as follows:

$e(t)=x(t)-x^*(t)$ (3.2)

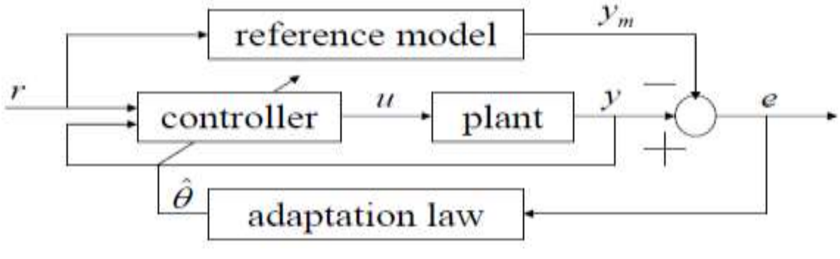

Fig 3.1 Block diagram of Adaptive Control

Taking a reference input r(t) taking laplace transform R(s) and $Y_m(s)$ as the laplace transform transform $y_m(t)$,

$$\frac{Y_m(s)}{R(s)} = \frac{q(s)}{p(s)} = \frac{q_0}{s^d + p_{d-1}s^{d-1} + \cdots \ldots + p_0} \qquad (3.3)$$

is a reference model where p(s) is pole of the polynomial such that $q_0 \rightarrow$ constant.

$$y_m(t) = \dot{y_m}(t) = \cdots \ldots \ldots = y_m^{(d)}(t) = 0 \qquad (3.4)$$

in this the output goes to zero such that its dynamics varies from r(t)=0,t≥0, so R(s)=0 and

$$p(s)Y_m(s) = 0$$

$$(s^d + p_{d-1}s^{d-1} + \cdots \ldots + p_0)Y_m(s) = 0 \qquad (3.5)$$

While determining the online approximation for plant non-linearities we have functions such as α(x) and β(x)

$$\theta_\alpha^T \emptyset_\propto(x) \qquad (3.6)$$

And $$\theta_\beta^T \phi_\beta(x) \qquad (3.7)$$

$$\alpha(x) = \theta_\alpha^* \emptyset_\propto(x) + w_\alpha(x) \qquad (3.8)$$

$$\beta(x) = \theta_\beta^* \phi_\beta(x) + w_\beta(x) \qquad (3.9)$$

where θ_α *and* θ_β are the parameter vectors

$$w_\alpha(x) \geq |w_\alpha(x)| \qquad (3.10)$$

$$w_\beta(x) \geq |w_\beta(x)| \qquad (3.11)$$

And to determine the certainty equivalence control term

$$e(t) = y_m(t) - y(t) \qquad (3.12)$$

3.3 GENETIC ADAPTIVE CONTROL ON SURGE TANK

In this controller we tune the system model to specify the system parameters. The Genetic Algorithm helps in tuning with the non-gradient methods so as to tune it with the set of models (approximators)[9].

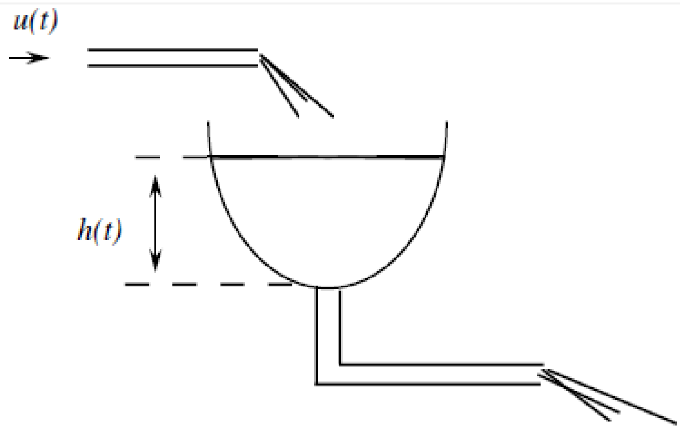

Fig 3.2 Diagram of Surge Tank

The population of estimators used in this method with a restriction such as β(x(k))≥β_0 for some known $\beta_0 > 0$,by assuming thatd=1and $\alpha_k = \beta_k = 0$ if we have d >1 for all set of populations[7].

Suppose all the set of approximators as α and β where the i^{th} can be defined as follows:

$$F_\alpha(x, \theta_\alpha^i) \tag{3.13}$$

and

$$F_\beta(x, \theta_\beta^i) \tag{3.14}$$

For i=1,2,3,4,…..,S. let i^{th} estimate the output and identification error which is shown as:

$$\widehat{y^i}(k+1) = F_\alpha(\text{x(k)}, \theta_\alpha^i(k)) + F_\beta(x(k), \theta_\beta^i(k))u(k) \tag{3.15}$$

e^{i}(k)= $\widehat{y^{\iota}}$(k)-y(k) (3.16)

$\theta^{i}(k) = [\theta_{\alpha}^{iT}(k), \theta_{\beta}^{iT}(k))^{T}$ (3.17)

For i= 1,2,3,….,S where θ^{i} acts as the chromosome of the i[th] individual of the population.

Now, we are searching for estimator parameters:

We consider

$J(\theta^{i}(k-1) = (e^{i}(k))^{2}$ (3.18)

= $(\widehat{y^{\iota}}$(k)-y(k)$)^{2}$ (3.18.1)

=$(F_{\alpha}(x(k-1), \theta_{\alpha}^{i}(k-1) + F_{\beta}\left(x(k-1), \theta_{\beta}^{i}(k-1)\right)u(k-1) - y(k))^{2}$

(3.18.2)

It measures the estimation error size such that all terms in the above equation are computable and minimize→ $J(\theta^{i}(k-1)$.

$\bar{J}(\theta^{i}(K-1)) = \frac{1}{\gamma + J(\theta^{i}(k-1)}$ (3.19)

Where;

γ>0 , is a design variable such that by using genetic operation we determine the population of next generation where , we determine crossover , mutation probability, selection and elitism. By comparing all the parameters we determine the system generations . The parameters can be adjust with the help of θ(k), where all the parameters are estimated at time k, which is used in finding the equivalence control law :

θ(k) = argmin$\{J\left(\theta^{i}(k-1)\right) : i = 1,2,3, \dots, S\}$ (3.20)

which can be seen as the parameters for obtaining the best model for population such that elitism is used in determining the fitness function of the system[12,13].

In determining the cost function that it quantifies the estimated error i. e.

$J_s\big(\theta^i(k-1), N\big) = \sum_{j=k}^{k-N} e_s^i\,(j))^2$ (3.21)

$= \sum_{j=K}^{k-N} (\widehat{y^\iota}(\mathrm{j}) - \mathrm{y(j)})^2$ (3.21.1)

Where ;

$\widehat{y^\iota}(\mathrm{j}) = F_\alpha\left(x(j-1), \theta_\alpha^i(k-1), \theta_\beta^i(k-1)\right) + F_\beta(x(j-1)\theta_\beta^i(k-1))$ u(j-1)

(3.22)

$J_s\big(\theta^i(k-1), N\big)$ →determines the approximation error over the last N steps.

$\theta^i(k-1)$ →determines the parameters for the estimator, by computing all these terms we can determine the system accuracy and performance characteristics much better and the estimated data can be obtained.

For computing the surge tank problem:

We have n=m=0 so x(k)=h(k). By picking the structured approximator we designed the GA and its adapted mechanism .

The approximator choice for the plant non-linearities can be determined as follows:

$F_\alpha\big(h(k), \theta_\alpha(k)\big) = \theta_\alpha(k)h(k)$ (3.23)

$F_\beta\big(h(k)\big), \theta_\beta(k)) = \theta_\beta(k)$ (3.24)

Where ;

$\theta_\alpha(k)$ and $\theta_\beta(k)$ are the scalars for determining the approximator w.r.t the tunable parameters, by using the non-gradient based methods.

The boundary conditions for the system are:

$-2 \leq \theta_\alpha(k) \leq 6 \; and \; 0.25 \leq \theta_\beta(k) \leq 0.5$

Liquid level control of surge tank is a non-linear model where h(t) determines the value of liquid level(saturated) , u(t) as the liquid level control , c and d are the constants and A(h(t))=|a(h(t)+b| is the known cross-sectional area in which a and b are also constants .

$$\frac{dh(t)}{dt} = \frac{-d\sqrt{2gh(t)}}{Ah(t)} + \frac{c}{A(h(t))}u(t) \tag{3.25}$$

When we determine the input of the actuator and the fact that the level of the liquid never go negative in any case than the dynamics can be viewed as :

$$h(k+1) = \left(h(k) + T\frac{-d\sqrt{19.6h(k)}}{|Ah(k)+b|} \right) + \left(\frac{cT}{|Ah(k)+b|} \right) \tag{3.26}$$

$$\alpha(h(k)) = \left(h(k) + T\frac{-d\sqrt{19.6h(k)}}{|Ah(k)+b|}\right) \tag{3.27}$$

and

$$\beta(h(k)) = \left(\frac{cT}{|Ah(k)+b|} \right) \tag{3.28}$$

Thus we have

$$h(k+1) = \alpha(h(k)) + \beta(h(k))\, u(k) \tag{3.29}$$

3.4 JAYA ADAPTIVE CONTROL ON SURGE TANK

A global search-based population which is proposed by Ravipudi Venkat Rao in 2016 was Jaya Algorithm. The concept on which Jaya Algorithm determine the best solution just to avoid failure condition . It is easier in implementation and required common controlling parameters such as number of generations and population size[1,8].

The liquid level control Surge tank we have n = m = 0 , such that , x(k)=h(k) ,by picking the approximator for designing the Jaya algorithm for the surge tank such that the choice of the approximator can be determined as follows :

$$F_\alpha\big(h(k), \theta_\alpha(k)\big) = \theta_\alpha(k)h(k) \tag{3.30}$$

$$F_\beta\big(h(k)\big), \theta_\beta(k)) = \theta_\beta(k) \quad (3.31)$$

Where ;

$\theta_\alpha(k)$ and $\theta_\beta(k)$ are the scalars for determining the approximator with respect to the tunable parameters, so clearly we can use the non-gradient based methods[4].

The boundary conditions for the system are:

-2$\leq\theta_\alpha(k) \leq 6$ and $0.25 \leq \theta_\beta(k) \leq 0.5$

Such that for i[th] population we can determine the system modelling in terms of θ^i;

$$\theta^i = (\theta_\alpha^i, \theta_\beta^i)^T \quad (3.32)$$

The online approximators used for plant non-linearities can be approximated by α(x) and β(x) :

$$\theta_\alpha^T \emptyset_\propto(x) \quad (3.33)$$

And

$$\theta_\beta^T \phi_\beta(x) \quad (3.34)$$

The parameter vectors θ_α and θ_β as the state trajectory that can be used for closed loop control are :

$$\alpha(x) = \theta_\alpha^* \emptyset_\propto(x) + w_\alpha(x) \quad (3.35)$$

$$\beta(x) = \theta_\beta^* \phi_\beta(x) + w_\beta(x) \quad (3.36)$$

similarly, by determining the compact parameters such as Ω_α and Ω_β :

$$\theta_\alpha^* = \arg\min \textit{for all } \theta_\alpha \epsilon\ \boldsymbol{\Omega_\alpha}(\sup |\ \theta_\alpha^T \emptyset_\propto(x) \text{-} \alpha(x)|) \quad (3.37)$$

$$\theta_\beta^* = \arg\min \textit{for all } \theta_\beta \epsilon \Omega_\beta (\sup |\theta_\beta^T \phi_\beta(x) - \beta(x)|) \quad (3.38)$$

Where , $w_\alpha(x)$ and $w_\beta(x)$ are the approximation errors for which α(x) and $\beta(x)$ are represented approximators finite size .

$$W_\alpha(x) \geq |w_\alpha(x)| \quad (3.39)$$

$$W_\beta(x) \geq |w_\beta(x)| \tag{3.40}$$

Where these are the known as state bounds depends on the error in the system such that they represent the actual system with approximators .

$$\widetilde{\alpha(x)} = \theta_\alpha^T(t)\phi_\alpha(x) \tag{3.41}$$

$$\widetilde{\beta(x)} = \theta_\beta^T(t)\emptyset_\beta(x) \tag{3.42}$$

For determining the parameter errors that can be calculated as follows:

$$\widetilde{\theta_\alpha}(t) = \theta_\alpha(t) - \theta_\alpha^* \tag{3.43}$$

$$\widetilde{\theta_\beta}(t) = \theta_\beta(t) - \theta_\beta^* \tag{3.44}$$

Now we can consider the indirect adaptive control law :

$$u = u_{ce} + u_{si} \tag{3.45}$$

The error can be determined as follows :

$$e(t) = y_m(t) - y(t) \tag{3.46}$$

First phase of Jaya Algorithm :

Now determining the dynamics of Jaya Algorithm , in which each individual comprises of n design variables , $x_j= (x_1,x_2,x_3,\ldots\ldots.,x_n)$

$$x_{j,i}= x'_{j,i}+rand[0,1] * (x^u_{j,i}-x'_{j,i}) \tag{3.47}$$

where $x'_{j,i}$ and $x^u_{j,I}$ are the lower and the upper bounds such that the rand[0,1] is the uniformly distributed between 0 and 1 so that population size can be determined.

$x'_{j,k,i}=x_{j,k,i}+r_{1,j,G}*(x_{j,best,G}-|x_{j,i,G}|)-r_{2,j,G}*(x_{j,worst.G}-|x_{j,i,G}|)$ (3.48)

where , $x_{j,best,G}$ → determines the values of the best candidate .

$x_{j,worst.G}$ → determines the values of the worst candidate.

Range to be between [0,1].

$r_{1,j,G}*(x_{j,best,G}-|x_{j,i,G}|)-r_{2,j,G}$ → it points the tendency of the best solution.

$(x_{j,worst.G}-|x_{j,i,G}|)$ → it points out the tendency of the system to be worst .

$r_{1,j,G}*(x_{j,best,G}-|x_{j,i,G}|)$ → it points out the tendency of the system by making system better and obtain the best solution and avoids the worst solution of the system[1,8].

The exploration ability of the system determines by $|x_{j,i,G}|$.

Second phase of the Jaya Algorithm can be determined by $x'_{j,i,G}$ such that we can determine the values of lower and upper bounds as follows :

$$x'_{j,I,G}=\begin{cases} 2x'_j - x'_{j,i,G} & if\, x'_{j,i,G} < x'_j \\ 2x^u_j - x'_{j,i,G} & if\, x^u_{j,i,G} > x^u_j \\ x'_{j,i,G} & otherwise \end{cases} \quad (3.49)$$

Finally all the values of the objective function where the vector $x'_{i,G}$ is compared to its next generation for obtaining the best and the worst value results in the population.

$$x_{i,G+1}=\begin{cases} x'_{i,G} & if\ f(x'_{i,G}) \le f(x_{i,G}) \\ x_{i,G} & otherwise \end{cases} \quad (3.50)$$

where f is the cost function which has to be minimized.

The fitness function of the system can be determined as :

$$J'_{G-1}(x_{i,(G-1),N} = \sum_{i=k}^{k=N} [x_{i,G-1,N}] \qquad (3.51)$$

$$=(F_i(x(G-1), x_i'(G-1) + F_G\big(x(G-1), F_G'(G-1)\big)u(G-1) - y(G))^2 \qquad (3.50.1)$$

Liquid level control of surge tank is a non-linear model where h(t) determines the value of liquid level(saturated) , u(t) as the liquid level control , c and d are the constants and A(h(t))=|a(h(t)+b| is the known as total area in which a and b are also constants [19,20].

$$\frac{dh(t)}{dt} = \frac{-d\sqrt{2gh(t)}}{Ah(t)} + \frac{c}{A(h(t))}u(t) \qquad (3.52)$$

When we determine the actuator input and the fact that the liquid level never goes negative in any case than the dynamics can be viewed as :

$$h(G+1)= (h(G)+T \frac{-d\sqrt{19.6h(k)}}{|Ah(G)+b|})+ (\frac{cT}{|Ah(G)+b|}) \qquad (3.53)$$

$$\alpha(h(G)) = (h(G) +T \frac{-d\sqrt{19.6h(k)}}{|Ah(G)+b|}) \qquad (3.54)$$

$$\beta(h(G))= (\frac{cT}{|Ah(G)+b|}) \qquad (3.55)$$

Thus we have

$$h(G+1)=\alpha(h(G))+\beta(h(G))\ u(G) \qquad (3.56)$$

Such that we can determine the system dynamics through the modelling of Jaya such that it can determine the best value and the worst value of the candidate in the population by determining the height of the liquid level surge tank by Eq 3.52 [1,8] and the Non-Linearities α and β as shown above by the Eq 3.54 and Eq. 3.55.

3.5 CONCLUSION

In this chapter we have described two algorithms Jaya Algorithm and Genetic Algorithm which is used to determine the liquid level surge tank parameters such that we can determine the closed loop response of the system , non-linearity estimation of the system and the fitness function value of the system.

CHAPTER 4

SIMULATION RESULTS AND DISCUSSIONS

To illustrate the supremacy of Indirect Jaya Adaptive control on Liquid level control Surge Tank , performance of the designed algorithm for the Liquid Level Control Surge tank has been depicted graphically by means of MATLAB Simulation and the comparison of Algorithm with their respective functions is also depicted in this chapter.

4.1 COMPARISON OF BENCHMARK FUNCTIONS ON ALGORITHMS

In this section we will study about the comparison of different functions such that the values so obtained with help of their MATLAB code for different function such as Rosenbrock , Himmelblau , Rastringin and Easom Function as shown in Table 4.1. The Results obtained by implementing Jaya adaptive control and Genetic adaptive control Using Surge tank.

In first section a comparison of Rosenbrock function and Himmelblau function using GA and Jaya is shown and the table determining the values obtained for other algorithm such as PSO, HS TLBO with Rastringin and Easom function .

In second section the comparison of Jaya adaptive control with Genetic adaptive control is shown in which tank parameters values varaiations and determining the height of the liquid level and the non-linearities alpha and beta is determined with the help of programming . The Jaya adaptive programming determines the values of different function and similarly for the genetic Adaptive control as shown in next pages of this dissertation.

Table 4.1 Comparison of different benchmark functions on different algorithms

S.No	FUNCTIONS	ALGORITHM	BEST VALUE	MEAN VALUE
1.	**Rosenbrock Function** **Rastringin Function** **Himmelblau function** **Easom Function**	Jaya Algorithm	16.91*10^5 0.000000 6356810294.677 2936802884.14	16.20*10^3 174482914927.71402 2934542884.141142 22265928725.917694
2.	**Rosenbrock Function** **Rastringin Function** **Himmelblau function** **Easom Function**	Genetic Algorithm	0.664192 0.398124 65226.5 56.450	4.61776e+06 7.2526e-12 71699.5 35.2234
3.	**Rosenbrock Function** **Rastringin Function** **Himmelblau function**	HS	62.2429 1.0000 61.5452	23.5455 0.6000 26.9051
4.	**Easom Function**	PSO	8.054481	5.5454
5.	**Himmelblau function** **Easom Function**	TLBO	26.9051 19.8008	20.3453 14.45645

The Graphs showing the best and mean value of GA , JAYA using Himmelblau function in Fig 4.1 , 4.2 .

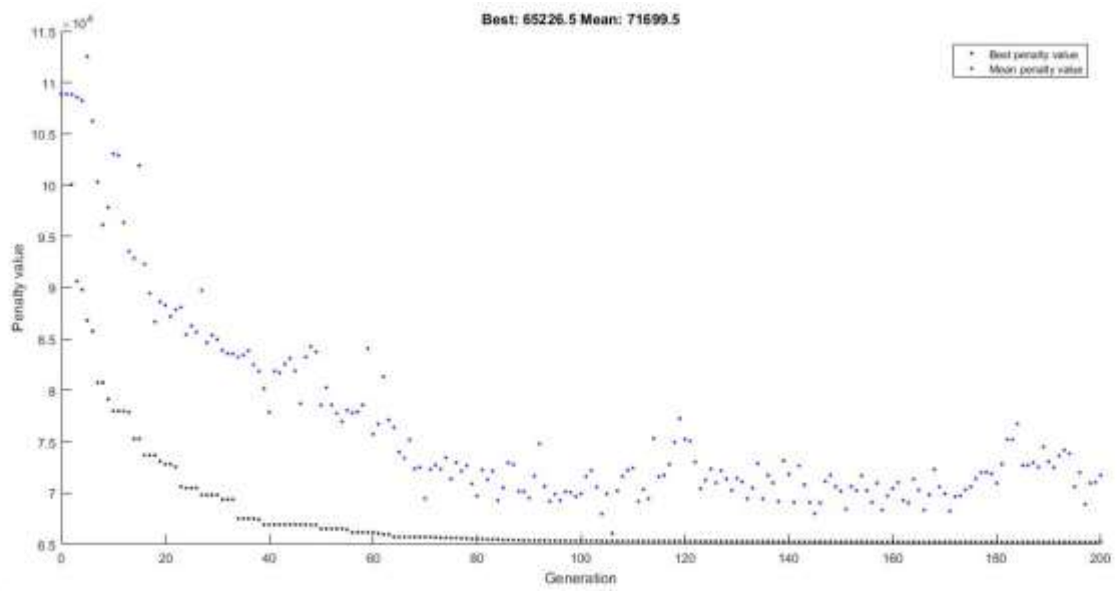

Fig 4.1 Himmelblau function using GA

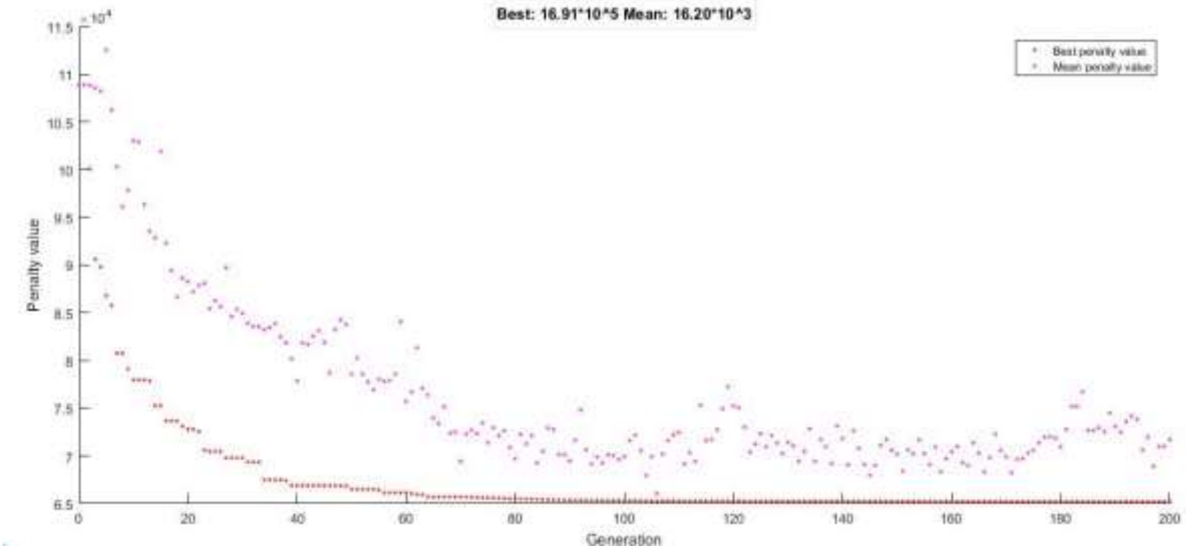

Fig 4.2 Himmelblau function using Jaya algorithm

The fig 4.3 ,4.4 represents the rosenbrock function using GA , Jaya by determining the fitness value and the maximum generation of the function .

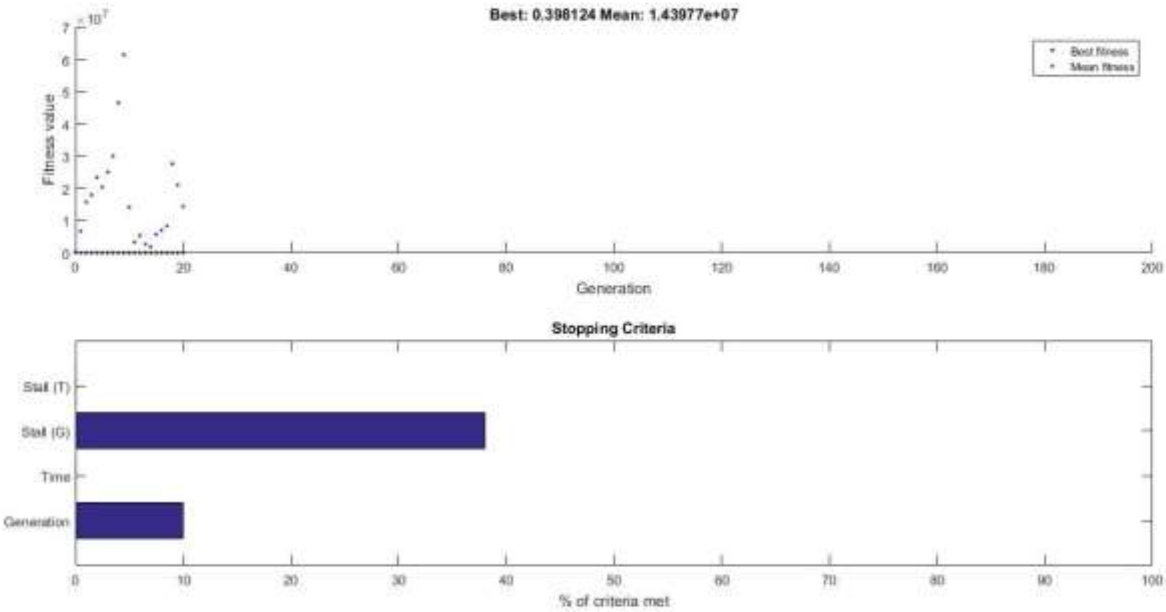

Fig 4.3 Rosenbrock function using GA

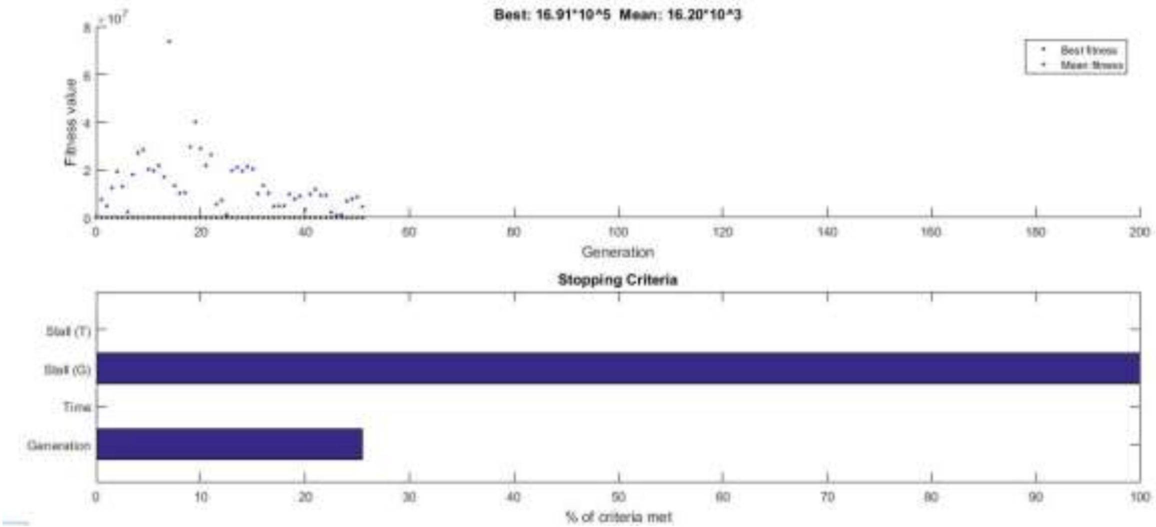

Fig 4.4 Rosenbrock function using Jaya algorithm

4.2 COMPARISON OF JAYA ADAPTIVE CONTROL FROM GENETIC ADAPTIVE CONTROL FOR SURGE TANK PROBLEM

The Fig 4.5 and Fig 4.6 determines the liquid level ‘h’ and reference input ‘r’ and tank input ‘u’ such that closed loop system performance is shown and in this we can see that after an initial transient period such that these results is due to poor initialization of estimators and we get good tracking of the reference input.

Tank parameters taken for Indirect Jaya Adaptive control on liquid level control Surge Tank and Indirect Genetic Adaptive Control on liquid level control surge tank at value of Jayanc :1000 and Nnc :1000 for the case of Genetic Algorithm.

abar :0.01 ; bbar :0.2; cbar:1 and dbar:1 as these parameters determine the clogging factor and parameters for lower bound and upper bound .

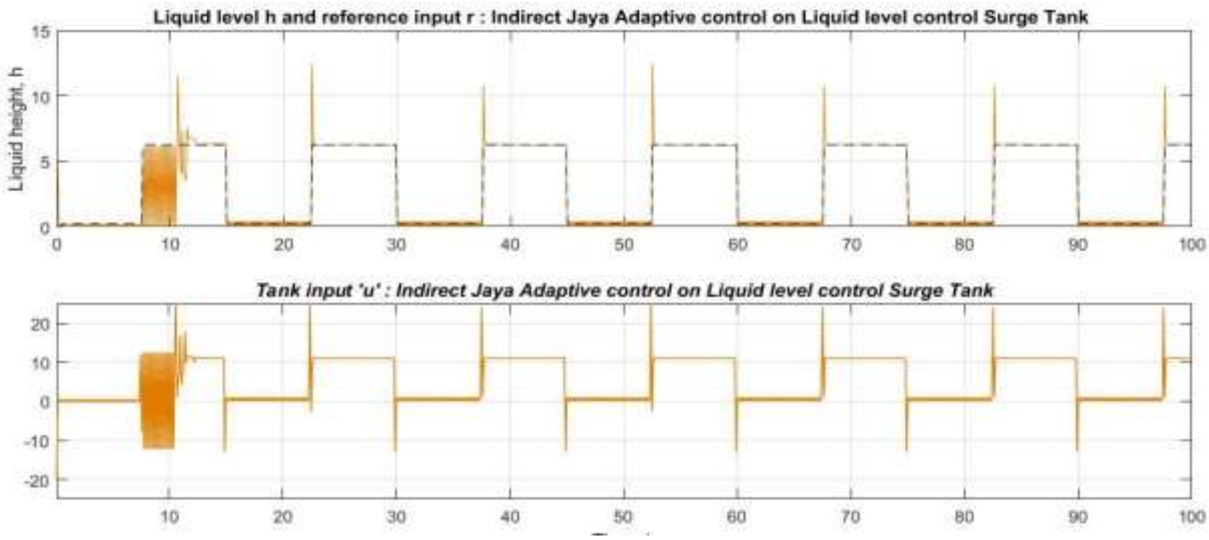

Fig 4.5 Jaya Adaptive Control Closed loop response

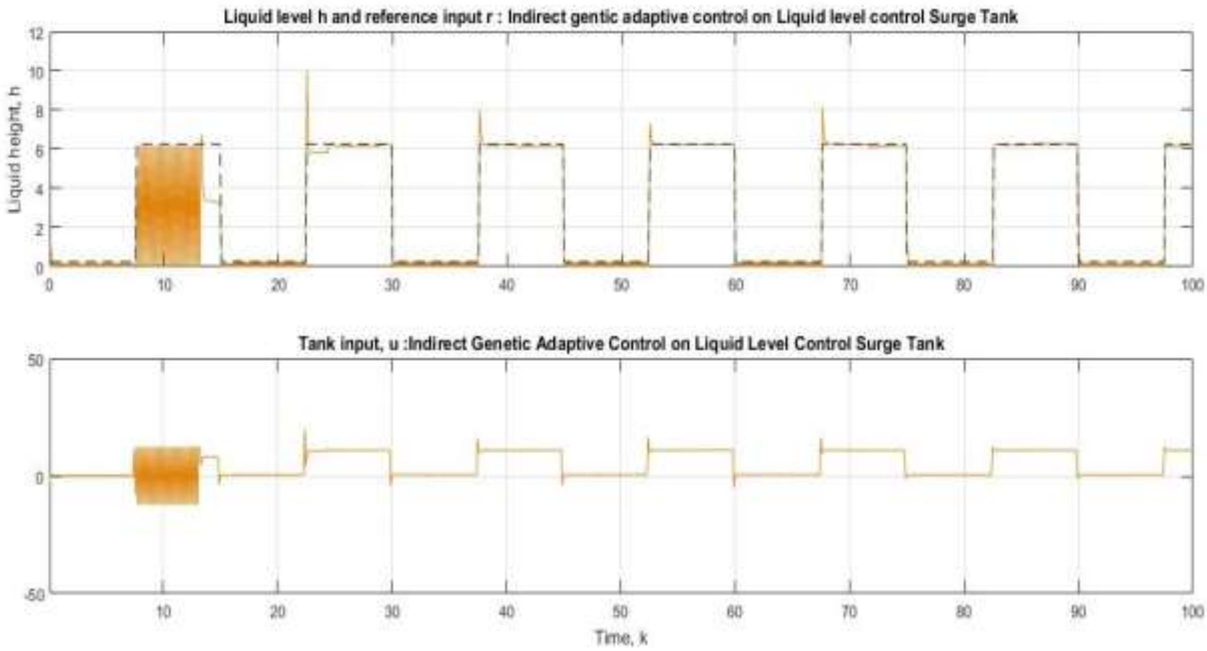

Fig 4.6 Genetic adaptive control closed loop response

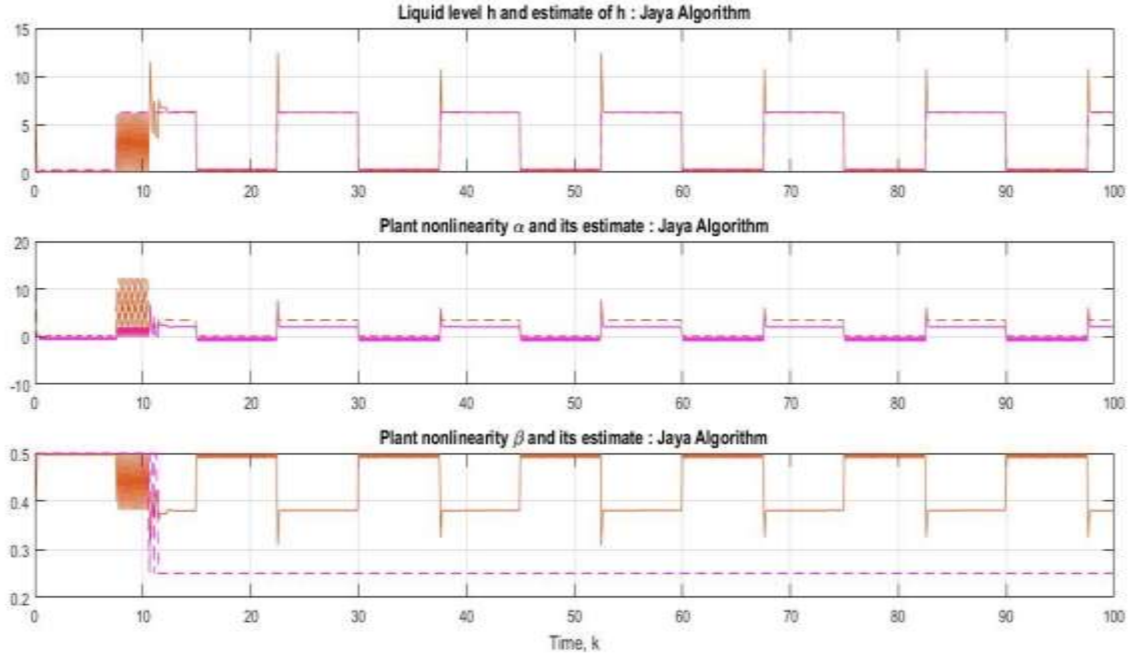

Fig 4.7 Jaya Adaptive control for estimation of the liquid level height and non-linearities

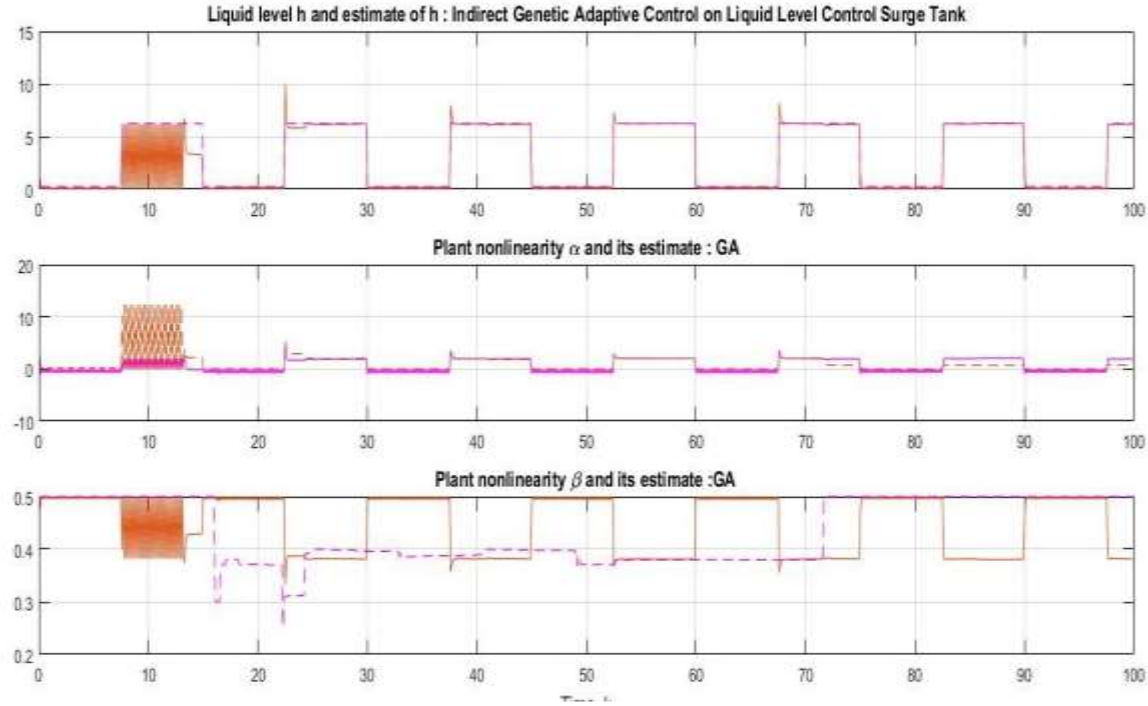

Fig 4.8 Genetic Adaptive control that estimates the liquid level and non-linearities

The Fig 4.7 and Fig 4.8 determine the estimate of liquid level control which is quite good for both the Algorithms : Jaya Algorithms and Genetic Algorithm ; even though the individual estimates of the non-linearities are not.

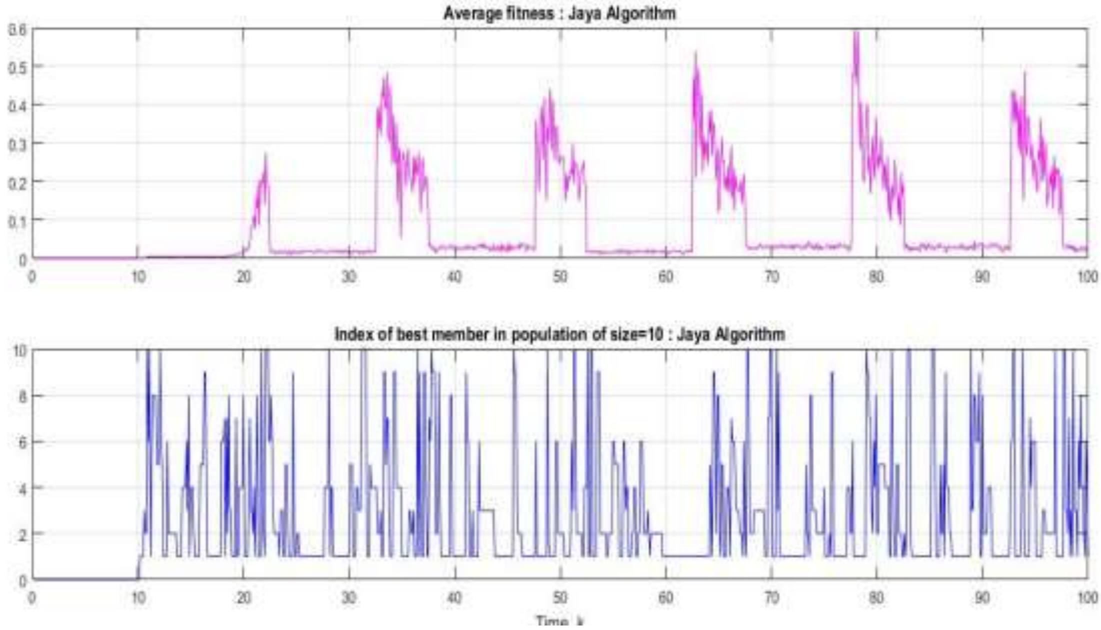

Fig 4.9 Jaya Adaptive Control average Fitness value and Index of best member in Population

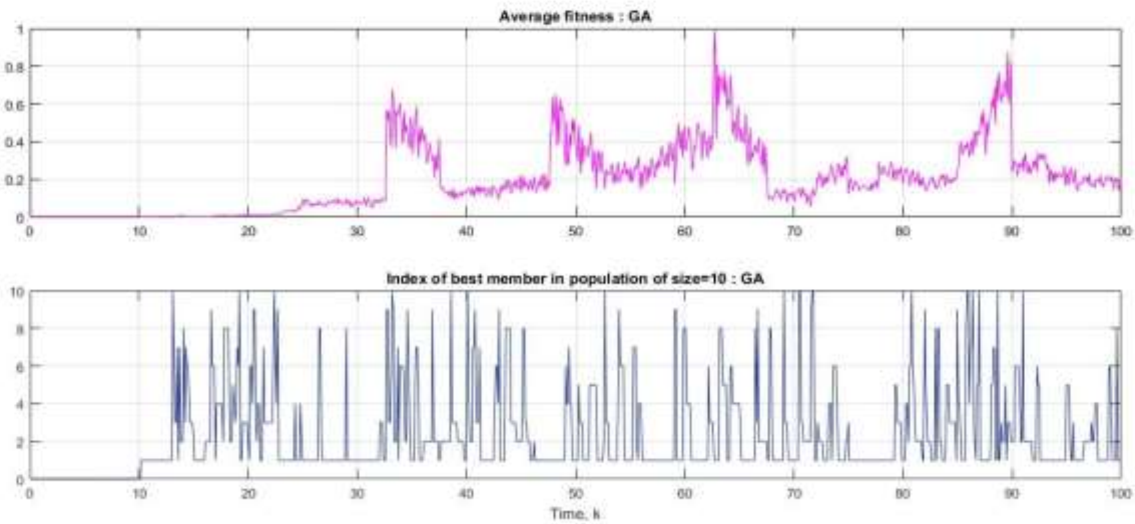

Fig 4.10 Genetic Adaptive Control , average Fitness value and Index of best member in Population

The Fig 4.9 and Fig 4.10 determine the average fitness of the population and the index 'i'of the best individual for every time step. The average fitness is very low for both the cases because of the poor choice for the initial population .In both the population we get the "premature convergence "to improve the large mutation rate.

Tank parameters taken for Indirect Jaya Adaptive control on liquid level control Surge Tank and Indirect Genetic Adaptive Control on liquid level control surge tank at value of Jayanc :1000 and Nnc :1000 for the case of Genetic Algorithm.

abar :0.02 ; bbar :0.3 ; cbar:0.9 and dbar:0.9 as these parameters determine the clogging factor and parameters for lower bound and upper bound and Gravity is to be taken 10 as shown in Fig 4.11 and 4.12 .

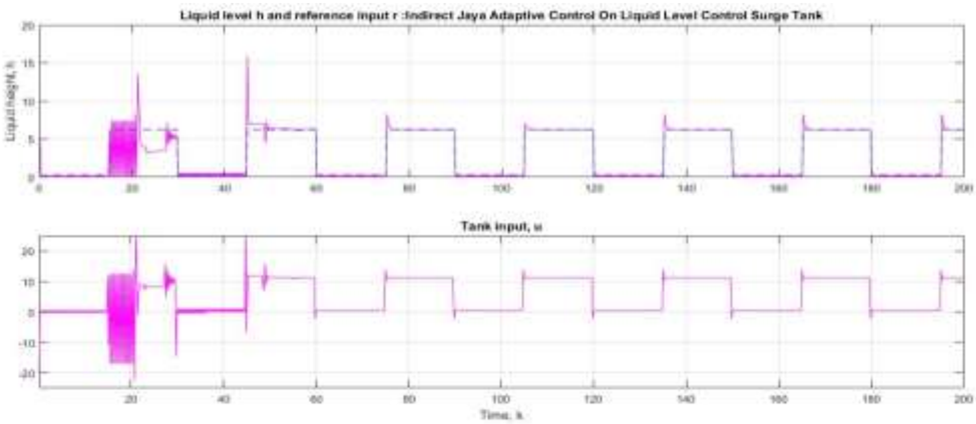

Fig 4.11 Jaya Adaptive control closed loop response when abar →0.02

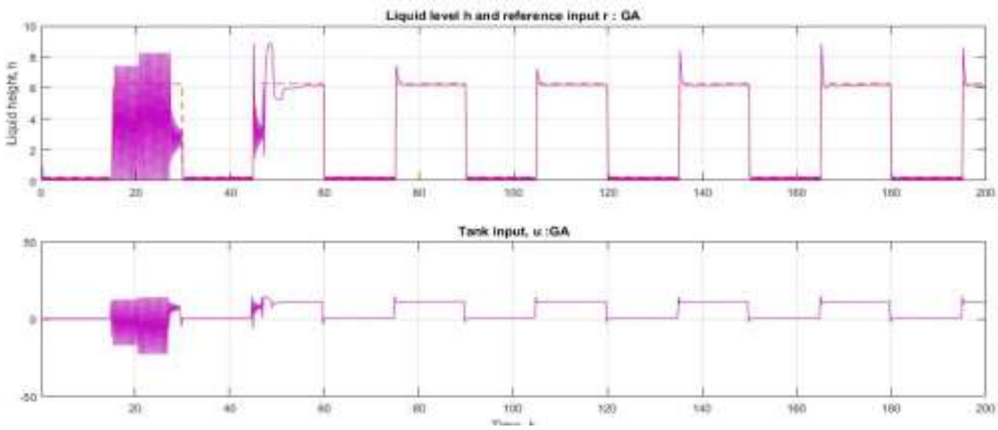

Fig 4.12 Genetic Adaptive control closed loop response when abar →0.02

The Fig 4.13 and 4.14 determine the estimate of liquid level control which is quite good for both the Algorithms : Jaya Algorithms and Genetic Algorithm ; even though the individual estimates of the non-linearities are not.

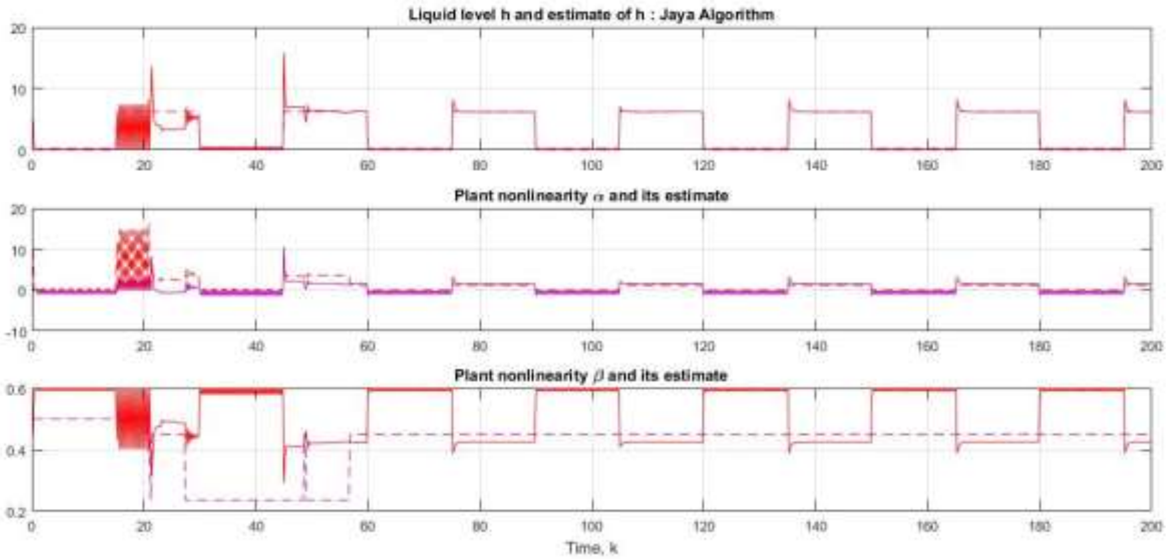

Fig 4.13 Jaya Adaptive control estimates of liquid level and non-linearities when abar →0.02

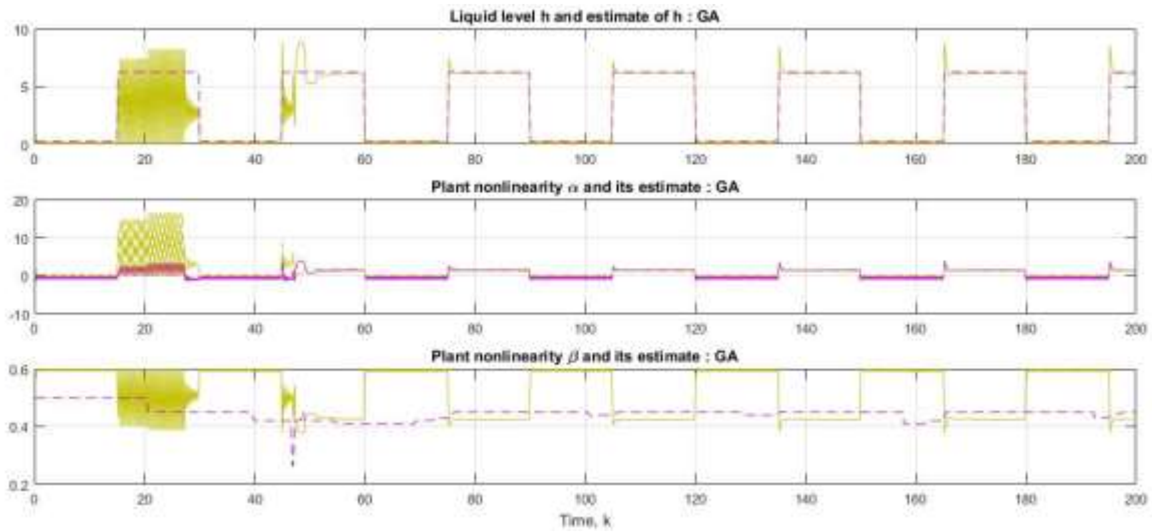

Fig 4.14 Genetic Adaptive control estimates of liquid level and non-linearities when abar →0.02

The Fig 4.15 and Fig 4.16 determine the average fitness of the population and the index ‘i’of the best individual for every time step. The average fitness is very low for both the cases because of the poor choice for the initial population .In both the population we get the “ premature convergence “to improve the large mutation rate.

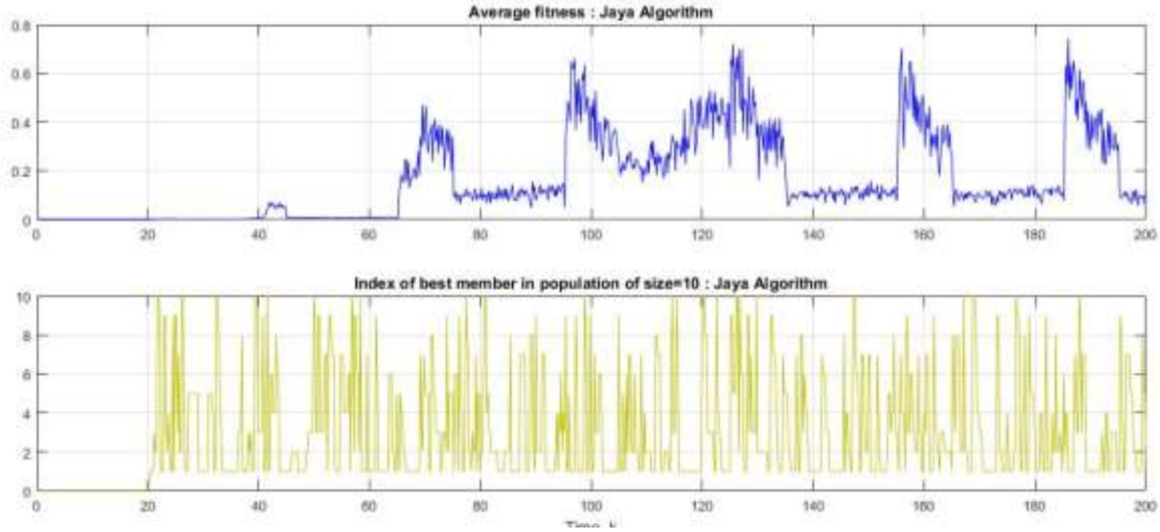

Fig 4.15 Jaya Adaptive control fitness value when abar →0.02

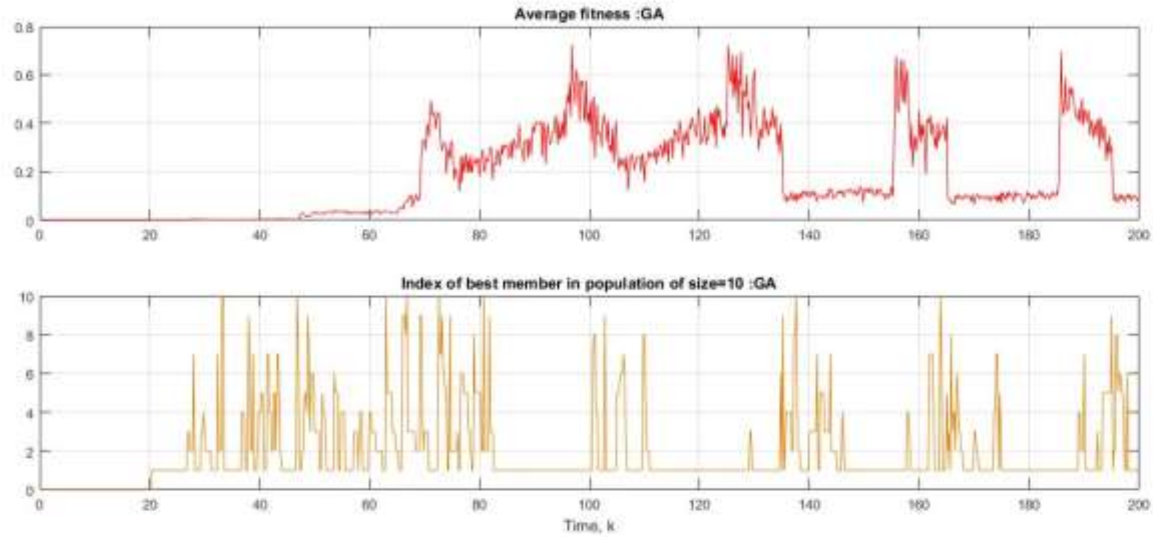

Fig 4.16 Genetic Adaptive control fitness value when abar →0.02

The Fig 4.17 and 4.18 given below determines the liquid level ‘h’ and reference input ‘r’ and tank input ‘u’ such that the performance of the closed loop system is illustrated and in this we can see that after an initial transient period such that these results is due to poor initialization of estimators and we get reasonably good tracking of the reference input. Tank parameters taken for Indirect Jaya Adaptive control on liquid level control Surge Tank and Indirect Genetic Adaptive Control on liquid level control surge tank at value of Jayanc :2000 and Nnc :2000 for the case of Genetic Algorithm. abar :0.02 ; bbar :0.3; cbar:0.9 and dbar:0.9 as these parameters determine the clogging factor and parameters for lower bound and upper bound and gravity to be taken 10.

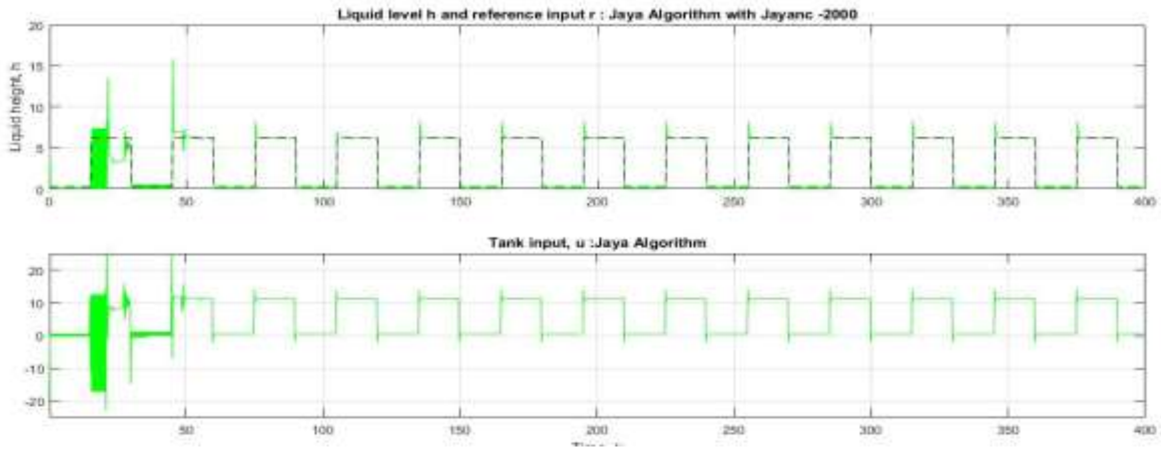

Fig 4.17 Jaya Adaptive control, closed loop response at Jayanc-2000

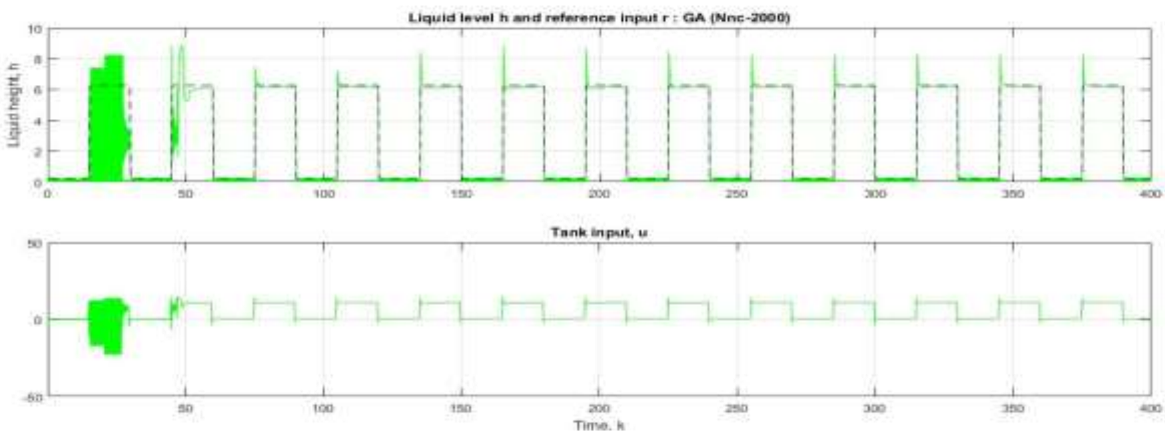

Fig 4.18 Genetic Adaptive control, closed loop response at Nnc-2000

The Fig 4.19 and Fig 4.20 , determine the estimate of liquid level control which is quite good for both the Algorithms.

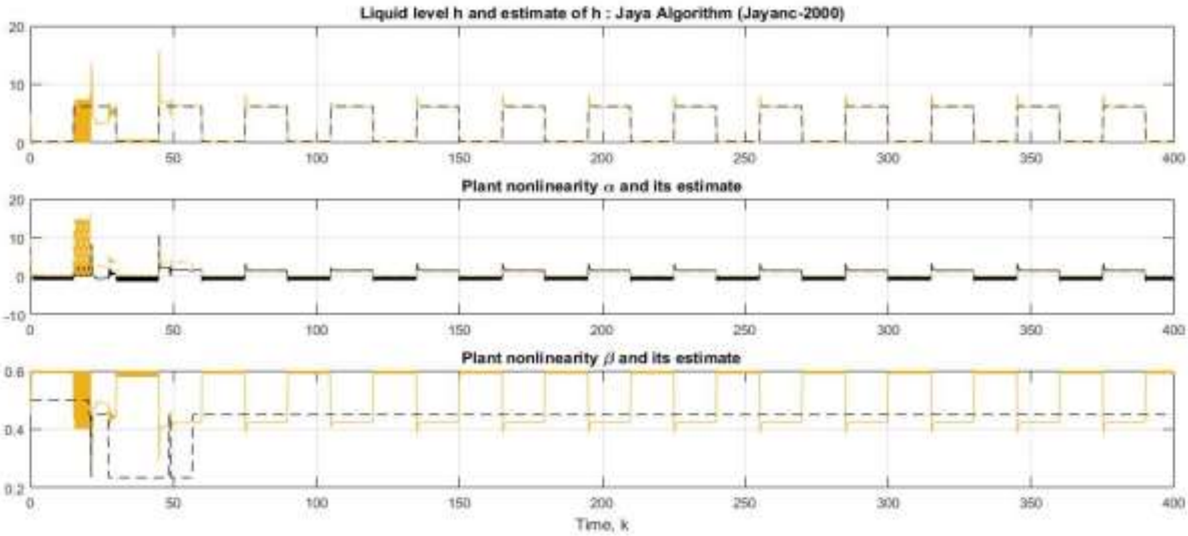

Fig 4.19 Jaya Adaptive control, estimates liquid level and non-linearities at Jayanc-2000

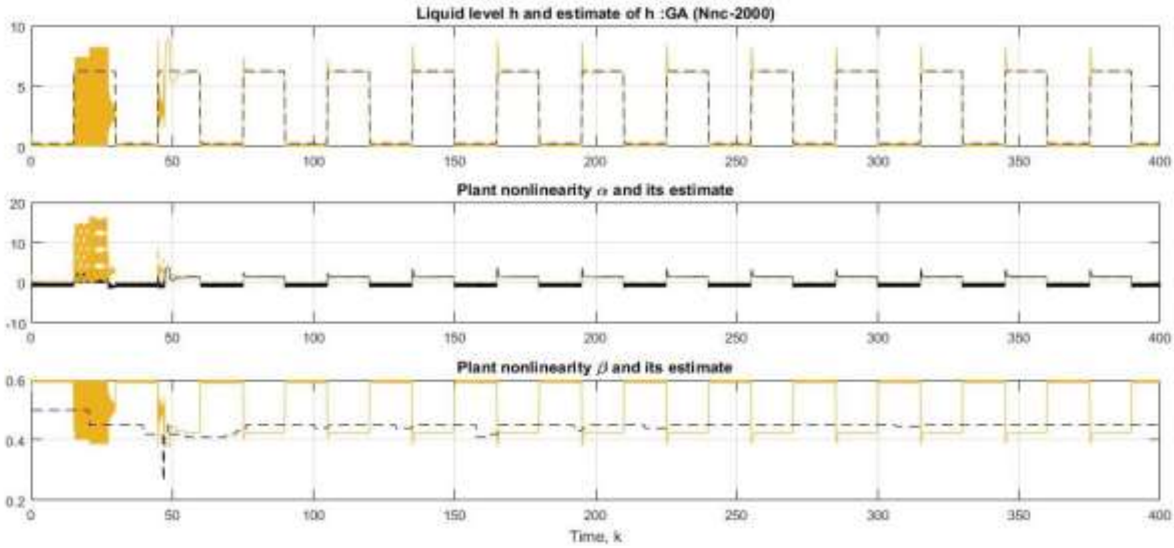

Fig 4.20 Genetic Adaptive control, estimates liquid level and non-linearities at Nnc-2000

The average fitness is very low for both the cases because of the poor choice for the initial population .In both the population we get the " premature convergence "to improve the large mutation rate as shown in Fig 4.21 and Fig 4.22.

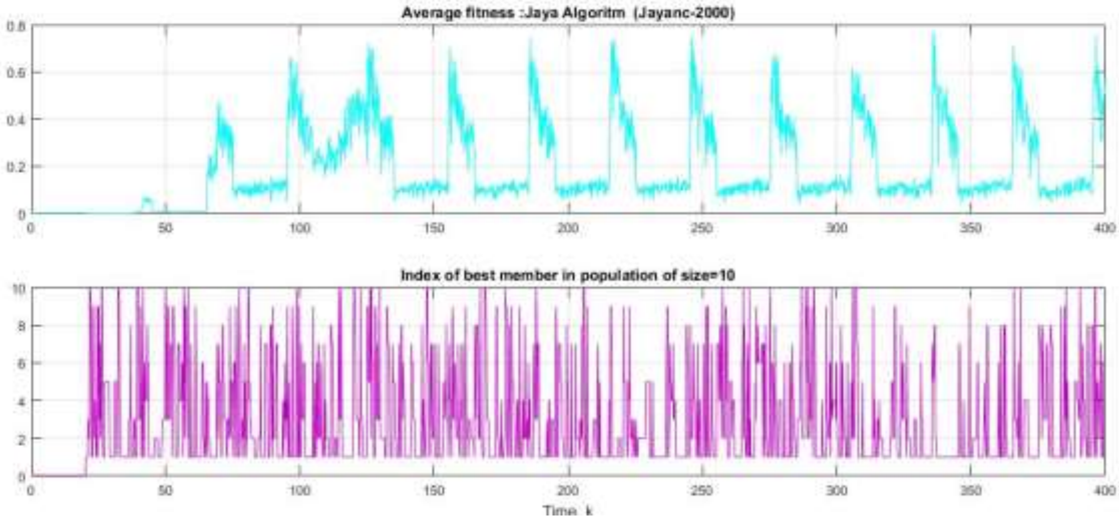

Fig 4.21 Jaya Adaptive control, Fitness function at Jayanc-2000

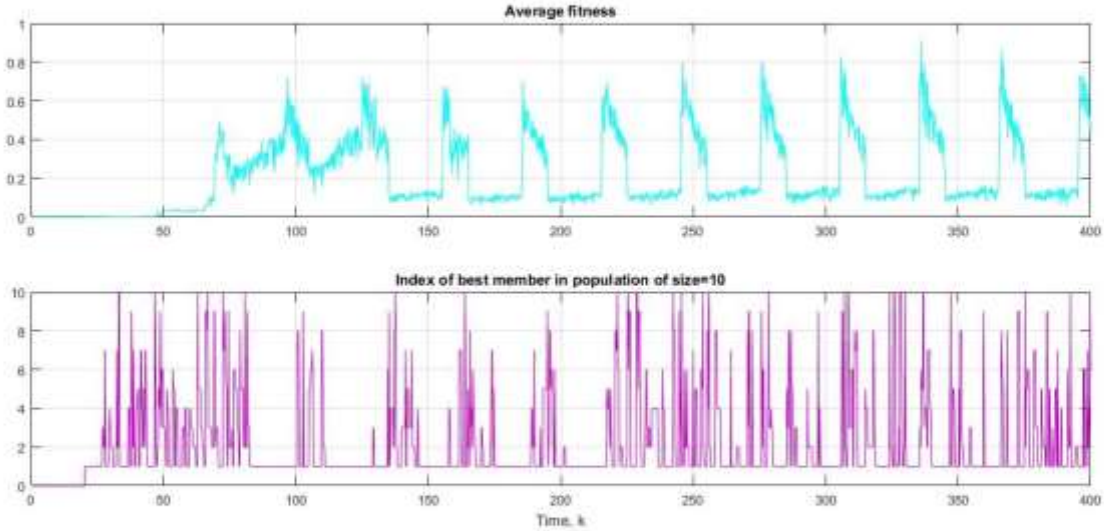

Fig 4.22 Genetic Adaptive control, Fitness function at Nnc-2000

The Table 4.2 determines the values of all the parameters of Jaya Algorithm such that it shows the best result and the worst result of the system and make the system more stable as compared to Genetic Algorithm as time taken by Jaya Algorithm is less than that of Genetic Algorithm.

Table 4.2 Parameters and the values of Jaya algorithm on Surge Tank

S.No	Jaya Algorithm parameters	Values obtained	Best Result	Worst Result
1.	**Jayanc**	1000	1000	2000
2.	**maxGen**	1 1	1	-
3.	**Bbest**	8 19 30	-	-
4.	**Mbest**	2	2	-
5.	**Jaya_surge**	0.00	-	-
6.	**Updatepopulation**	0.4116	0.4116	-
7.	**Maxi**	0.9000	-	-
8.	**Mini**	0.0500	-	-
9.	**STD_BEST**	22	-	-
10.	**Bestvalue**	12 0.50	12	0.50
11.	**Worstvalue**	−8.00 0.25	0.25	-8.00
12.	**Height of liquid level surge tank 'h'**	11.0939 11.0437 11.0861	-	-
13.	**col0 'act as non-linearity α '**	0.2500	-	-
14.	**col1'act as non-linearity β '**	0.5000	-	-
15.	**Sampling time**	Self-time : 1.440s Total time : 3.485 s	1.440	6.773

The Table 4.3 determine all the parameters of the Genetic algorithm and the best result obtain is in the range of 0-100 when Nnc is taken as 1000 , where the time taken by the system is more than that of indirect Jaya Adaptive control.

Table 4.3 Parameters and the values of Genetic algorithm on Surge Tank

S.No	GA Parameters	Values obtained	Best result	Worst result
1.	Nnc	1000	1000	2000
2.	alpha(1)	-1.1082	-1.1082	--
3.	beta(1)	0.4762	0.4762	--
4.	thetaalpha(1)	2	2	--
5.	Thetabeta(1)	0.50	0.50	--
6.	Alphahat(1)	2	2	--
7.	Betahat(1)	0.5	0.5	--
8.	Gamma 'γ'	1.000e-03	--	--
9.	Hhat	6.2500 6.2500 6.2500 6.2500	6.2500	0.00
10.	CHROM_LENGTH	12	12	0.00
11.	Sumfitness	2.1717	2.17	--
12.	Bestmember	8	8	1
13.	Thetaalpha(k)	0.1202	--	--
14.	Thetabeta(k)	0.5000	0.5	00
15.	Time Taken	Self time :3.746 Total time :8.664	2.072	8.664

4.3 CONCLUSION

In this chapter results are obtained for JAYA and GA algorithm using adaptive control on Liquid level control surge tank and the comparison of different functions with Jaya Algorithm and the benchmark functions.

CHAPTER 5
CONCLUSIONS AND FUTURE SCOPE

The work presented in this thesis describes the dynamics of Surge Tank using Jaya Algorithm and Genetic algorithm with adaptive control. The performance of the system with Jaya algorithm shows better results as compared to GA. The change in tank parameters shows that the Jaya adaptive control for closed loop response shows better performance characteristics but in case of GA the performance characteristics get more exaggerated. The performance of Jaya algorithm is better than that of the Genetic algorithm such that the system dynamics can be easily calculated .Jaya Algorithm is able to provide the reconnaissance mission properly for the Surge Tank. The self-time taken by Jaya adaptive control is 1.440 sec and the total time take is 3.485 sec whereas in case of Genetic adaptive control the self-time taken by the system is 2.072 sec and total time taken by the system is 8.664 sec. The Best practical value and the worst practical values obtained for the surge tank problem by Jaya adaptive control are 12 and – 8 respectively. In case of Genetic adaptive control the value of alpha, beta and the height of the system so obtained are -1.082, 0.4762 and 6.25 respectively.

Discussing about the future scope of Jaya Algorithm:

- It determines the economic dispatch and operational cycles for probabilistic unit commitment with demand response and high wind power.
- It would be used as pathological brain detection system for determining the stress level and depression level of a person without using needle and using different types of sensors.
- Using Jaya algorithm with other latest algorithms such as Memtic algorithms and SOMA algorithms for determining the heat and mass transfer, by synchronising these

algorithms with Jaya algorithm provide appreciable results and appreciable performance characteristics.

There are few algorithm which after using with Jaya algorithm in collaboration would provide better results, these algorithms are Rain Drop optimization , Simulated annealing , biogeography based optimization , chemical reaction optimization , group search optimizer , Bayesian optimization technique , population based incremental learning , continuous scatter search, for which adaptive control on surge tank can be implemented.

BIBLIOGRAPHY

[1] Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. 2016-17, pp. 19–34.

[2] Ghasemi, M.; Ghavidel, S.; Gitizadeh, M.; Akbari, E. An improved teaching–learning-based optimization algorithm using Lévy mutation strategy for non-smooth optimal power flow. Int. J. Electr. Power Energy Syst. 2015-16, pp. 375–384.

[3] Abido, M.A. Optimal power flow using particle swarm optimization. Int. J. Electr. Power Energy Syst. 2002, pp. 563–571.

[4] K. M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," IEEE Control System Magazine, 2002, pp. 52-67.

[5] D.E.Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Publishing Corporation, Inc., 1989, pp. 48-56.

[6] M.Gopal, "Digital Control and State Variable Methods," Tata McGraw-Hill Publishing Company Limited New Delhi, 2003.

[7] Bharat Bhushan,Madusudan Singh ,"Adaptive control of Non-Linear systems using Bacterial Foreaging Algorithm", International journal of computer and electrical engineering ,Vol3,No.3,June 2011, pp. 335-342.

[8] Du Dinh-Cong, Vinh Ho-Huu,, Trung Vo-Duy, Ngo-Thi Hong Quyen," Efficiency of Jaya algorithm for solving the optimization-based structural damage identification problem based on a hybrid objective function", Article in Engineering Optimization · September 2017 ,pp. 3451-3460.

[9] Bharat Bhushan, Madhusudan Singh," Adaptive control of DC motor using bacterial foraging algorithm" ,17 June 2011, pp. 4913-4920.

[10] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[11] K.M. Passino, S. Yurkovich, Fuzzy Control, Addison Wesley, 1998, pp. 35-43.

[12] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. (2002), pp. 52–67.

[13] W.M. Korani, H.T. Dorrah, H.M. Emara, Bacterial foraging oriented by particle swarm optimization strategy for PID tuning, in: Proceedings of the 10th Annual conference on Genetic and Evolutionary Computation, USA, July, 2008, pp. 1823–1826.

[14] Y. Chu, H. Mi, H. Liao, Z. Ji, Q.H. Wu, A fast bacterial swarming algorithm for high dimensional function optimization, in: Proceedings of IEEE World Congress on Computational Intelligence, 2008, pp. 3135–3140.

[15] N. Osmic, J. Velagic, S. Konjicija, A. Galijasevic, Genetic algorithm based identification of a nonlinear 2 DOF helicopter model, in: Proceedings of 18th Mediterraneam Conference on Control and Automation, 2010, pp. 333–338.

[16] H.M. Pandey, A. Chaudhary, and D. Mehrotra. "A comparative review of approaches to prevent premature convergence in GA." Applied Soft Computing 24 (2014), pp. 1047-1077.

[17] Repinšek, Matej, Shih-Hsi Liu, and Marjan Mernik. "Exploration and exploitation in evolutionary algorithms: a survey." ACM Computing Surveys (CSUR) 45.3 (2013), pp. 331-337.

[18] H.M. Pandey, A. Chaudhary, and D. Mehrotra. "Grammar induction using bit masking oriented genetic algorithm and comparative analysis." Applied Soft Computing 38 (2016), pp. 453-468.

[19] Venkata Rao Jaya Algorithm. Rao, R. "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems." International Journal of Industrial Engineering Computations 7.1 (2016), pp. 19-34.

[20] Hari Mohan Pandey ," Jaya a Novel Optimization Algorithm: What, How and Why? ",June 2017, pp. 1111-1120.

[21] Joaquin, D., Salvador, G., Daniel, M., & Francisco, H. (2011). A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1(1), pp. 3–18.

[22] Karaboga, D., &Akay, B. (2011). A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. Applied Soft Computing, 11, pp. 3021-3031.

[23] Patel, V.K., &Savsani, V.J. (2015). Heat transfer search (HTS): a novel optimization algorithm. Information Sciences, 324, pp. 217-246.

[24] Rao, R.V. (2015). Teaching Learning Based Optimization And Its Engineering Applications. Springer Verlag, London, pp. 946-955.

[25] Rao, R.V., & Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. International Journal of Industrial Engineering Computations, 3(4), pp. 535-560.

[26] Rao, R.V., & Patel, V. (2013). Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems. International Journal of Industrial Engineering Computations, 4(1), pp. 29-50.

[27] Rao, R.V., Savsani, V.J. & Vakharia, D.P. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), pp. 303-315.

[28] Ahrari, A. & Atai A. A. (2010). Grenade explosion method - A novel tool for optimization of multimodal functions. Applied Soft Computing, 10, pp. 1132-1140.

[29] Basturk, B & Karaboga, D. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization, in IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, pp. 116-122.

[30] Deb, K. (2000). An efficient constraint handling method for genetic algorithm. Computer Methods in Applied Mechanics and Engineering, 186, pp. 311–338.

[31] Fogel, L. J, Owens, A. J. & Walsh, M.J. (1966). Artificial intelligence through simulated evolution. John Wiley, New York, pp. 1240-1244.

[32] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Computer Engineering Department. Erciyes University, Turkey, pp. 2222-2232.

[33] Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, IEEE Press, Piscataway, pp. 1942-1948.

[34] Rao, R.V. & Savsani, V.J. (2012). Mechanical design optimization using advanced optimization techniques. Springer-Verlag London, pp. 1312-1330.

[35] R. Venkata Rao, "Decision making in the manufacturing environment using Graph theory and Fuzzy multiple attribute decision making Methods", 2016, pp. 224-229.

[36] Kirk, D.E.: Optimal Control Theory: An Introduction. Prentice-Hall, Englewood Cliffs (1970).

[37] Kumar, S.; Chaturvedi, D.K. Optimal power flow solution using fuzzy evolutionary and swarm optimization. Int. J. Electr. Power Energy Syst. 2013, 47, pp. 416–423.

[38] Sivasubramani, S.; Swarup, K.S. Multi-objective harmony search algorithm for optimal power flow problem. Int. J. Electr. Power Energy Syst. 2011, 33, pp. 745–752.

[39] K.M. Passino and S. Yurkovich. Fuzzy Control. Addison Wesley Longman, Menlo Park, CA, 1998, pp. 54-58.

[40] K.M. Passino. Intelligent control for autonomous systems. IEEE Spectrum, 32(6), pp. 55–62.

[41] D. Joaquin, G. Salvador, M. Daniel, H. Francisco, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2016), pp. 3–18.

[42] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (2015), pp. 43–60.

[43] Chan, Wah-Chun, and Yuan-Yih Hsu., Application of Fuzzy-PIDF Controller For Automatic Generation Control Using JAYA Algorithm, IEEE Proceedings C (Generation, Transmission and Distribution). Vol. 128. No. 5. IET Digital Library, 1981, pp. 468-472.

[44] Yudong Zhang , Xiaojun Yang , Carlo Cattani , Ravipudi Venkata Rao , Shuihua Wang and Preetha Phillips [7]Tea Category Identification Using a Novel Fractional Fourier Entropy and Jaya Algorithm, Inf. Sci. 291 (2015), pp. 43–60.

[45] "Jaya algorithm for solving urban traffic signal control problem"J. Miller , Proc. 2ndInt. Symp. Traffic Theory, pp. 200-220.

[46] "Parameters identification of photovoltaic models using an improved Jaya optimization algorithm" Energy Conversion and Management , Volume 150, 15 October 2017, pp. 742-753.

[47] "Intelligent facial emotion recognition based on stationary wavelet entropy and Jaya algorithm" Neurocomputing ,Volume 272, 10 January 2018, pp. 314-319.

[48] "A novel Elite Opposition-based Jaya algorithm for parameter estimation of photovoltaic cell models", Optik Volume 155, February 2018, pp. 351-356.