Efficient Channel Estimation of OFDM Using PSO and CS

[1]S. Nithya, [2]R. Surna Keerthi
[1]UG Students, [2]Assiatant professor
Department of ECE,
University college of Engineering, Tamil Nadu, India.

ABSTRACT:

In industrial environments arrangement of embedded systems required pre configuration for operation, reconfiguration capabilities are also desirable. Therefore it is useful to define a mechanism for embedded devices that will operate in sensor to be remotely (re)configured. A node component should be arranged in any embedded device and implements application programming interface (API), configuration, processing, and communication. Maintenance is very easy and basic instructions are used in this method. On-line process is use to control the embedded devices which is used in the industries. Cloud is use to collect the data"s and store in a server and share the information. Server monitoring the status and control the sensors in anywhere in the world.

I.INTRODUCTION

In many industries have hundreds or thousands of sensors which automatically monitoring and control functionalities. In past the industries are using wired sensors which have heavy costly but present the industries using wireless sensor which are easy to arrangement and less costly. The inclusion of embedded devices in industrial control and monitor application provides compactability and less cost when compared with wired sensors.

In industries arrangements are wired sensors, wireless sensors (WSNs), and wired backbone nodes form a heterogeneous network system.

In the user point of view a network control system which has the wireless sensor network is offers configuration capability. In engineer point of view developed the heterogeneous system that include WSN node and detail of processing and communication programming by hand, the arrangement of middleware, standard components anyone can deploy the node using basic instructions. In this architecture the programming is required to develop the node component for a specific platform by "driver suppliers". But this is applicable for a each new operating system. An important research issue in this context is how to provide interoperability between different nodes and provide a single configuration and data processing model in that kind of systems that handles different realizations. Our work aims at designing the architecture using sensors, servers and embedded devices in industries which sensor has any fault is (re)configure using online process. Also the data store process is by SQL language and the cloud computing is used for data collection and data sharing process.

The remainder of this paper is organized as follows. Section II discusses related work concerning strategies to configure and program sensor network devices and to connect those to internet. In Section III requirements needed for WSN industrial applications are specified. Section IV describes the proposed architecture. It explains how the architecture achieves remote easy (re)configuration and processing flexibility of any embedded systems. Section V shows an example of practical implementation and results obtained from an oil refinery, where data collection, alarms, and actions must be defined over the devices to prevent accidents. Finally, Section VI concludes the paper.

II. RELATED WORK

The approach proposed in this paper provides configuration/reconfiguration capabilities for embedded devices using online process. With it, distributed sensor can be managed without any manual handling, using only simple online automatic commands. Related work includes strategies to configure and program sensor network devices and to connect those to internet. In the literature, there exist several works that address reconfiguration. normally, reconfigurable devices are based on complex programmable logic (CPLDs) or field-programmable gate arrays (FPGAs) that have capabilities to change hardware. These are aimed at only hardware changes and do not offer the application configuration flexibility that our approach does. But in the industrial environment the necessary flexibility to adapt to industrial process changes.

Dynamic software updating over WSNs was surveyed in [2]. Such mechanisms allow reconfiguring the nodes without manually removing them from the architecture, programming them, and putting them back into the site. We next review some recent works on the subject, and then we reach conclusions on the comparison with our approach. In this first paper [1] describe the architecture that provides the remote (re)configuration and processing capabilities to any embedded devices and to distributed system comprising embedded devices and other computing devices. The proposed architecture of [1] builds an intermediate computing layer which will serve as an abstraction hiding the different hardware implementations from embedded devices network applications. It assumes a network distributed monitoring work, including sensor nodes, such as embedded devices, and more powerful nodes, such as PC‟s or servers. In this model PCs are not just data sinks but may fully participate in the distributed computation environment. In this first paper is described about MidSN. The architecture works on top of a network communication infrastructure that is used to exchange data message between nodes, sends configuration commands to nodes and send acknowledgements from them.

The node component (MidSN–NC) provides uniform configuration and processing capabilities over heterogeneous distributed sensor and actuator networks. It must be developed only once for each hardware and/or operating system and offers functionality to allow any node to be configured remotely for the same operations. MidSN can be developed for hardware with at least a certain number of features. The minimum requirements to run it are: a programmable micro-controller; a communication stack for data exchange; enough ROM memory to hold the MidSN-NC software component. The amount of RAM needed is configurable, since it depends on the number of operation related structures that are allowed. Sensing nodes must also have ADCs to connect to sensors and actuators need DACs to interface with external hardware. The operation of MidSN-NC based on timer and network events. There are three main types in our prototype of the architecture.

- Acquisition: triggers acquisition of sensor signals.
- Computation: computes from data that is in memory. To compute average or maximum values, to filter data, to rise alarms, to merge data from streams.
- Send: send data to external applications.

Network events are messages that arrive through the network interface. A parser identifies the type of message, which may be either a configuration command or data message.MidSN-NC includes a Data Collector module. It is implemented by SQL (Structured Query Language). It was designed by platform and communication protocol.

In this paper [2] the communication and processing methods is then modified to eliminate the redundancies. Interval cover graph to minimize communication redundancies and a joint flow graph optimization to remove computational redundancies. Both methods operate online and allow application requests to be dynamically added or removed.

Many sensors may be wireless or battery operated. Some sensors may be connected to the internet using low bandwidth or remote links. They may be running or low end processors. These constraints make it crucial to minimize the communication resource usage. Computation is likely to be performed at central servers that maintain index of available sensors. Large number of sensors and applications served imply that the server resource usage per sensor and per task for performing online data processing on sensor streams cannot be very high.

As a result both communication and computation redundancy elimination methods must adapt dynamically to changing tasks being concurrently served. Also the cost of detecting the redundancies should not be greater than the savings achieved by exploiting the redundancies. To communication redundancy minimize receives streaming requirements for all tasks that use this sensor. It determines the minimum common set of time instances at which sample must be collected and uploaded to satisfy all tasks, and updates this set as tasks are added or removed. This module is implemented at the individual sensors as it does not require information from other sensors. The computational redundancy minimize receives all tasks submitted by applications and jointly optimizes the computation across these tasks. This module must operate at a central point through which all tasks pass.

An application that needs to sense the environment submits a sensing task specifying the spatial region to be covered. The task is received by the central server, and the relevant sensors are determined based on the spatial coverage and sensor types desired, using an index of all sensors stored at the server. The sensor is thus only responsible for collecting the required data and the communication redundancy minimize at the sensor uploads sufficient data to satisfy all tasks. The computation redundancy minimize at the central server will compute the metrics requested by applications in an optimized using the sensor data streams.

3. PROPOSED CHANNEL ESTIMATION METHOD:

In our proposed method, Particle Swarm Optimization is applied in LS and MMSE to calculate the best channel and fitness value. Combining LS and MMSE method using Cuckoo Search algorithm programming can greatly reduce the error in the area of wireless communication. The detailed information will be given in following sections.

3.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based, meta- heuristic, iterative optimization algorithm, inspired by the coordinated movements of birds flocking introduced by Kennedy and Eberhart in 1995.The system is initialized with a population of random solutions and searches for optima by updating generations.

In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. PSO is the search based method used to improve the speed of convergence and identify the global optimum value of fitness function. PSO initiates with a population of random solutions „„„particles"" in a 2D-dimension space. PSO is applied to LS and MMSE channel estimation in the area of wireless communication.

To compare these LS and MMSE channel estimation, to detect where the mean square error is minimum. The block diagram of the OFDM with Particle Swarm Optimization is shown in Figure 3.1.

In Figure 3.1, PSO is applied with LS and MMSE channel estimation between the Fast Fourier Transform (FFT) and demodulation block. After appling PSO, to select the best channel in LS and MMSE channel estimation, Cuckoo Search algorithm is employed.

3.1.1 ALGORITHM FOR PSO

The PSO algorithm works by simultaneously maintaining several candidate solutions in the search space. During each iteration of the algorithm, each candidate solution is evaluated by the objective function being optimized, determining the fitness of that solution. Each candidate solution can be thought of as a particle "flying" through the fitness landscape finding the maximum or minimum of the objective function. In this PSO algorithm, it has six steps,

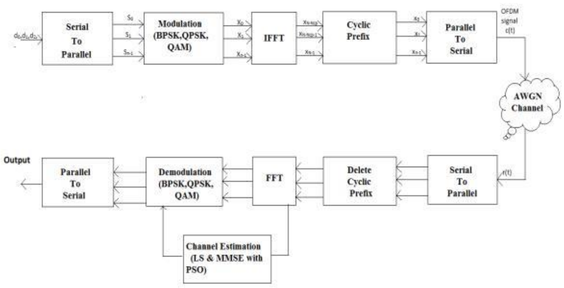

Figure 3.1 Block Diagram of OFDM with PSO

- Step 1: Create a „population" of agents (particles) uniformly distributed over X.
- Step 2: Evaluate each particle"s position according to the objective function.
- Step 3: If a particle"s current position is better than its previous best position, update it.
- Step 4: Determine the best particle (according to the particle"s previous best positions).
- Step 5: Update particles velocities.
- Step 6: Giving best optimal solution.

Initially, the PSO algorithm chooses candidate solutions randomly within the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particle.

The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values.

The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions with following equation(a)and(b).

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] – present[]) (a)

present[]=present[]+v[](b)

where v[] is the particle velocity, present[] is the current particle (solution). pbest[] and gbest[] aredefined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

Each particle maintains its position, composed of the candidate solution and its evaluated fitness, and its velocity. Additionally, it remembers the best fitness value it has achieved thus far during the operation of the algorithm, referred to as the individual best fitness, and the candidate solution that achieved this fitness, referred to as the individual best position or individual best candidate solution.

The pseudo code of the procedure is as follows:

For each particle
Initialize particle
END
Do for each particle
Calculate fitness value
if the fitness value is better than the best fitness value (pBest)
set current value as the new
pBest End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
Calculate particle velocity according equation (a)
Update particle position according equation (b) End

While maximum iterations or minimum error criteria is not attained.

Particles‟ velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

3.1.2 Comparisons between Genetic Algorithm and PSO

Most of evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop. Otherwise go back to 2.

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques.

However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm.

Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area.

In PSO, only gBest (or lBest) gives out the information to others. It is a one -way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases.

3.1.3 FLOWCHART FOR PSO

This flowchart is based on the above algorithm. These six steps are based on the above steps of the algorithm. Initially, initialize the number of particles with its random positions and velocity vectors. Evaluate the each particles positions and then to calculate its fitness. After calculating the fitness value set the best value as gbest and pbest.

Compare each particles current position with its previous position and update each particle‟s current position. According to particles previous positions, to determine its best positions. Update each particle‟s velocity and its position. In this flowchart, finally we update the current best optimal solution.

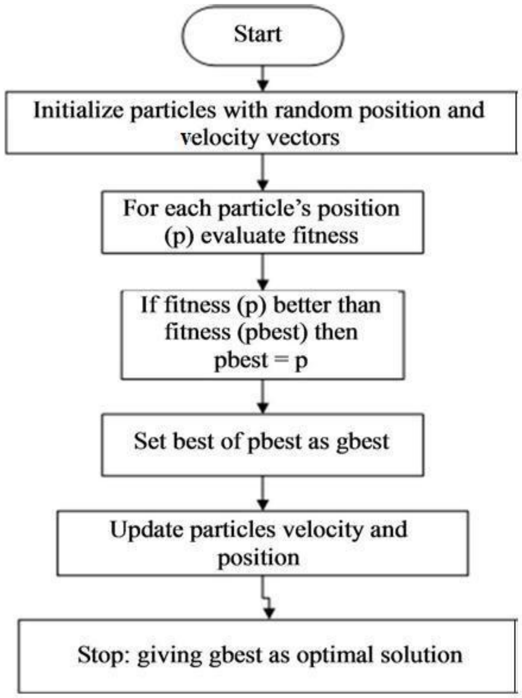

Figure 3.2 Flowchart for PSO

PSO algorithm has many parameters which are used in the coding. The table contains parameters and its rate. Therefore, the following parameters are mentioned in the below table, such as problem dimension, particles, iteration, mutation probability, inertia weight factor, r_1,r_2, C_1, C_2, C_1.

Table 3.1 Parameters and its rates for PSO

Parameter	Rate
Problem dimension	11
Number of particles	100
Number of iteration	100
Mutation probability	0.1
Inertia weight factor	$_{min}$ =0.4, $_{max}$=0.9
r_1, r_2	Selected randomly in (0.1)
C_1	1
C_2	1.5
C	0.9

3.1.4 Advantages of PSO

The advantages of PSO are given below,

- It is very simple to implement.
- Easily parallelized for concurrent processing
- Very few algorithm parameters are used.
- It is insensitive to scaling of designing variables.

3.1.5 Applications of PSO

The applications of PSO are given below,

- Function optimization.
- Image recognition.
- Optimization of power distribution networks.
- To remove unwanted random noise in signal processing.

3.2 Cuckoo Search

Cuckoo Search (CS) is one the of the latest nature-inspired meta-heuristic algorithms, developed in 2009 by Xin-She Yang of Cambridge University and Suash Deb of C.V. Raman College of Engineering. CS is based on the brood parasitism of some cuckoo species as well as the Levy Flights of some birds and fruit flies which follow the random walk.

Cuckoo Search include population based, iterative based, stochastic, deterministic and other approaches. CS is compared with other already existing popular optimization algorithms in connection with numerous optimization problems. By using cuckoo search algorithm, we calculate the best fitness value when compared with LS and MMSE channel estimation. The block diagram of the OFDM with Cuckoo Search is shown in Figure 3.2.

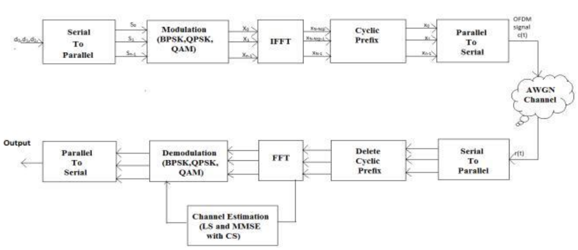

Figure 3.3 Block Diagram of OFDM with CS

In this block diagram, CS is applied with LS and MMSE channel estimation between the FFT and demodulation block. After appling CS, to select the best channel and to calculate the fitness value while combining LS and MMSE channel estimation can greatly reduce the error.

3.2.1 Algorithm for CS

Cuckoo search algorithm is used to select the best nest and to calculate the fitness value. Combining LS and MMSE channel estimation using cuckoo search can greatly reduce the error. In this algorithm, it has six steps.

Initially, CS algorithm choose number of nests and random initial solutions. Evaluate the each nest and random initial solutions to get the current best nest. After evaluating each nest, get the cuckoo value by randomly walk and it is not replace by Levy‟s flights. Evaluating the quality fitness, randomly choosing the best nest. By ranking each solution and nest, to choose the best nest. Then pass to next generation, otherwise go to step 2.

- **Step 1: Initialization**

  Initialization of nests and random initial solution.

- **Step 2: Evaluation**

  Get the current best nest.

- **Step 3: Loop construction**

  While (fmin > Max generation)

  Get the cuckoo value by random walk, if not replace it by Levy‟s flights.

- **Step 4: Evaluation**

  Evaluate the quality fitness .

  Randomly choose nest among n, say j.

- **Step 5: Condition**

  If (fni>fnj)

  Replace j value by new solution.

  End

- **Step 6: Solution construction**

  Retain the best solution and nests.

  Rank the solution and nests to choose the best.

  Pass to next generation.

  End while, else go to step 2.

3.2.2 FLOWCHART FOR CS

Cuckoo search flowchart is based on the above algorithm. Initialization of the nests and the random initial solution is performed. The current best nest is chosen by the random walk and then the evaluation of the quality is fitness is performed. Else apply levy‟s flight for the evaluation. Execution is carried out till all the solution is constructed.

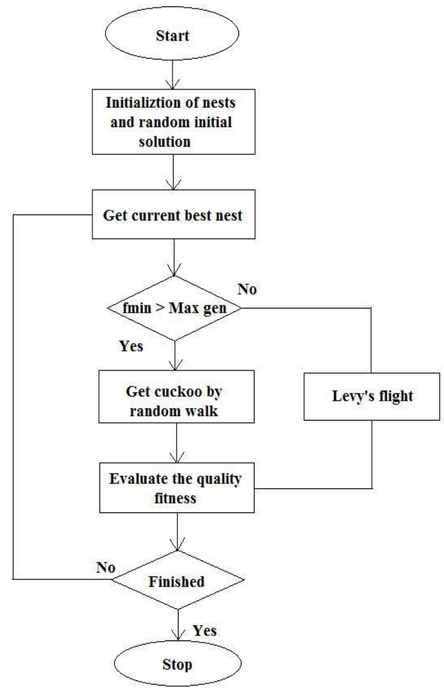

Figure 3.4 Flowchart for CS

3.2.3 Advantages of CS

The advantages of CS are given below,

- Less complex and more efficient.
- It controls elitism and balance of randomization.
- More generic and robust for many optimization problems.
- More significant in multi objective optimization problems.
- It can be used in scientific computing, and high power computing.

IV. IMPLEMENTATION PROCEDURE:

In the proposed method LS and MMSE methods are combined using Cuckoo Search algorithm. In OFDM channel model, initially the best channel is estimated by means of LS and MMSE independently using PSO and then, LS and MMSE methods are combined using CS algorithm to estimate best channel with reduced error. Finally, Obtaining the best channel with minimized error and fitness value by using Particle Swarm Optimization and Cuckoo Search algorithm. In this proposed work, it has three stages PSO is used in two stages and the CS is performed in one stage.

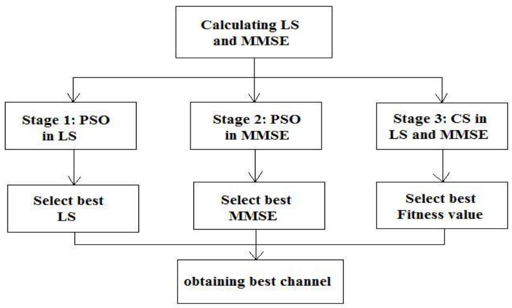

Figure 4.5 Block Diagram for Proposed Method

Stage 1: Implement the LS method using PSO and find the best fitness value to minimize error.

Stage 2: Implement the MMSE method using PSO and find the best fitness value to minimize error.

Stage 3: Combine the LS and MMSE methods using cuckoo search algorithm and find the best fitness value to minimize error.

Stage 4: Compare the performance.

Stage 1:

In this stage, initialize the number of particles, evaluate each particles position, and to determine the each particles previous position and its current position then update its current position. LS estimator becomes,

$$H_{LS}=F.Q_{LS}.F^{H}.X^{H}.y \quad (4.1)$$

$$\text{Where,}Q_{LS}=(F^{H}.X^{H}.X.F)^{-1} \quad (4.2)$$

Fitness can be calculated by using the evaluation function such as

$$\text{Evaluationfunction}=(H-H_{LS}/H)^{2} \quad (4.3)$$

Finally, the Least Square method using PSO algorithm to calculate the best channel in LS method with minimized error.

Stage 2:

In this stage, initialize the number of particles and evaluate each particles position. MMSE estimator becomes

$$H_{MMSE}=R_{gy}R^{-1}_{yy}y \quad (4.4)$$

$$H_{MMSE}=F.Q_{MMSE} \quad (4.5)$$

Where, $Q_{MMSE}=R_{gg}[(F^{H}.\ X^{H}.\ X.\ F)^{-1}\ \sigma_n^{2}\ +\ R_{gg}]^{-1}.(F^{H}.\ X^{H}.\ X.\ F)^{-1}$ (3.6)

In MMSE, Fitness can be calculated by using the evaluation function such as

$$\text{Evaluationfunction}=(H-H_{MMSE}/H)^{2} \quad (4.6)$$

Finally, Minimum Mean Square Error method using PSO algorithm to calculate the best channel in MMSE method with minimized error.

Stage 3:

Here, the best LS channel is identified using with CS. Initially, the LS channels are computed based on LS Channel estimation Model and MMSE channels are computed based on MMSE channel estimation Model. Using Cuckoo Search algorithm to calculate the best fitness value. LS and MMSE methods are combined using Cuckoo Search algorithm to estimate the best fitness value with minimized error.

Stage 4:

For obtaining the best channel estimation, the channel estimation obtained from stage 1, 2 & 3 are compared and among that channel estimation, the best channel is selected based on the minimum error value. The optimization rate is changed and generated more number of new channel estimation. The minimum error value channel estimation is selected as the best channel estimation. Then, the error values obtained from all the channel estimation are compared and finally, the channel estimation with minimum error is chosen as the best channel estimation.

V.IMPLEMENTATION

The different methods of channel estimation were implemented in this chapter. The different methods of channel estimation such as LS and MMSE with PSO and CS were simulated using MATLAB 2008 program.

5.1 Implementation parameters

The MIMO-OFDM, PSO and CS system was simulated using MATLAB program so that pilots are inserted among data for initial MMSE channel estimation. The OFDM block size is chosen as N=256 and a cyclic prefix(CP) with length of 8 is inserted at the head of each OFDM block. The four sub-channels are assumed to be independent to each other and to have maximum length L=5 individually.

The Table consists of following parameters such as MIMO Scheme, Modulation schemes which consists of BPSK,QPSK,16QAM and 64 QAM, Block size, Channel length, Number of iteration, PSO and CS length, Population size, and Probability(p_a)

Table 4.1 Parameters for Implementation

IMPLEMENTATION PARAMETERS	PARAMETER VALUES
MIMO Scheme	X×X
Modulation Scheme	QAM Modulation
Block Size(N)	256
Beta	17//9
Cyclic Prefix	8
Channel Length(L)	5
Number of Iteration	1000
PSO & CS length(M)	64
Population size(n)	25
Probability(p_a)	0.25

5.2 Sub-functions Used

In this project, the sub-functions such as mse, Hls, Hmmse, best nest and fmin which are used in channel estimation of OFDM with PSO and CS.

5.3 Results and Discuss

Here the plot for SNR Vs MSE. SNR is defined as the ratio of signal to noise power. Symbol rate is the number of symbol changes made to the transmission medium per second using a digitally modulated signal. MSE measures the average of the square of the error. The error is the amount by which the estimator differs from the quantity to be estimated. for LS and MMSE with Different Modulation Techniques and Channel estimation and Channel estimation with PSO-CS is shown in figure 4.1 and figure 4.2.

The matlab simulation were done on the basis of two parameter analysis

- Signal to noise ratio
- Mean square error

SNR means Signal to Noise Ratio and it is defined as signal power to noise power and MSE means Mean Square Error and it measures the average of the square of the error.

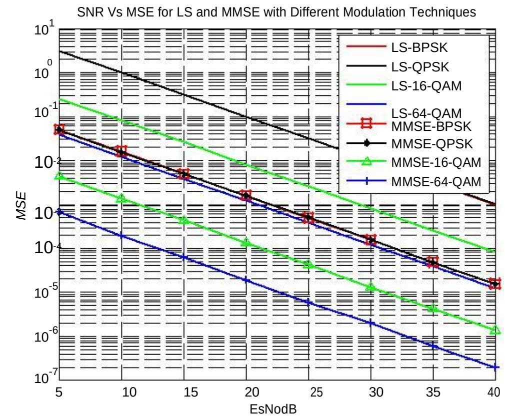

Figure 4.1 SNR Vs MSE for LS and MMSE with different modulation schemes

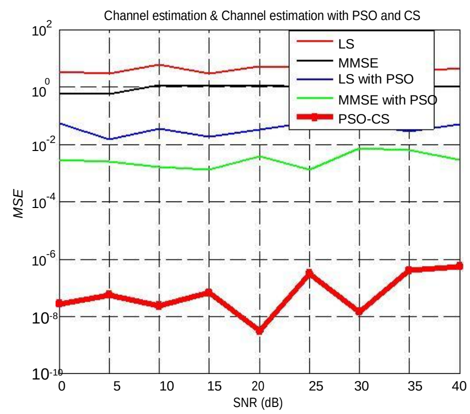

Figure 4.2 SNR Vs MSE for Channel estimation & Channel estimation with PSO and CS

These are the Simulation results of SNR Vs MSE plot for different modulation schemes and channel estimation and channel estimation with PSO and CS.

4.2.2 Scatter Plot For Estimated Channel with PSO-CS

SCATTERPLOT(X) generates a scatter plot of X. X can be a real or complex vector, or a two-column matrix with real signal in the first column and imaginary signal in the second column.

By using PSO-CS algorithm, the estimated channel for MIMO-OFDM system having negative correlation between In-phase and Quadrature components. So that it can relates the various variables in the Channel Estimation using PSO and CS algorithm. The Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7, Figure 4.8, Figure 4.9, Figure 4,10, Figure 4.11, Figure 4.12 shows Channel Estimation with PSO and CS for H, HhatLS, HhatLMMSE, HLS, Hls, mseLS1, HMMSE, Hmmse, mseMMSE, Fmin.

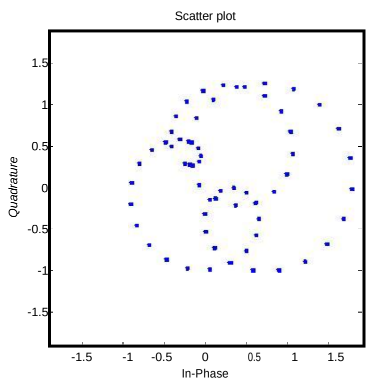

Figure 4.3 Estimated Channel with PSO-CS for H

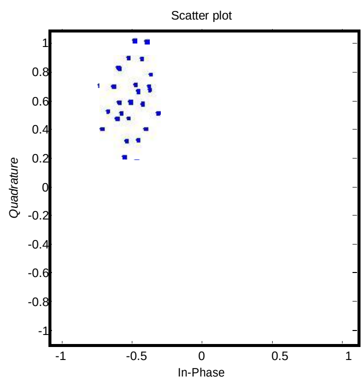

Figure 4.4 Estimated Channel with PSO-CS for HhatLS

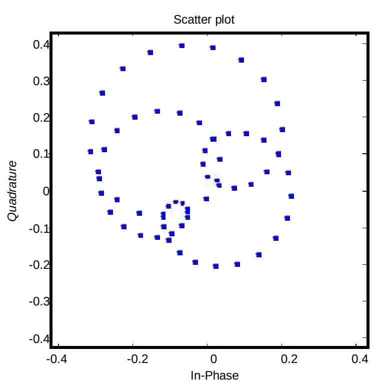

Figure 4.5 Estimated Channel with PSO-CS for HhatLMMSE

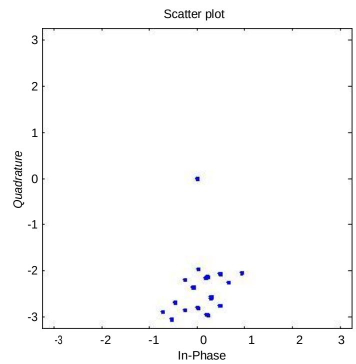

Figure 4.6 Estimated Channel with PSO-CS for HLS

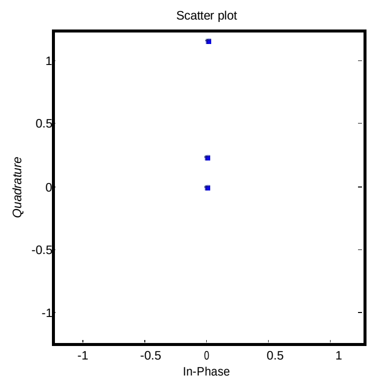

Figure 4.7 Estimated Channel With PSO-CS for Hls

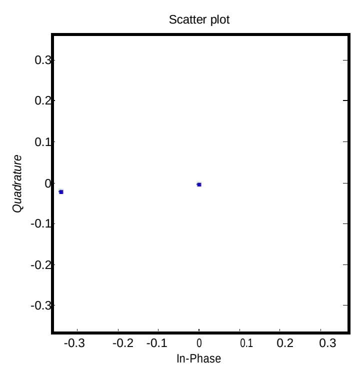

Figure 4.8 Estimated Channel with PSO-CS for mseLS1

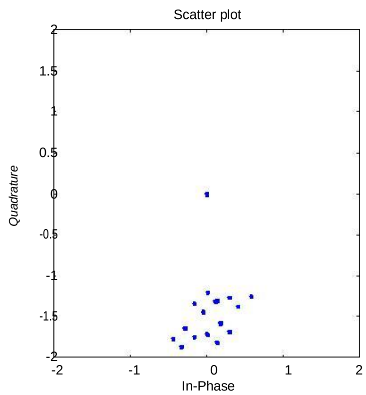

Figure 4.9 Estimated Channel with PSO-CS for HMMSE

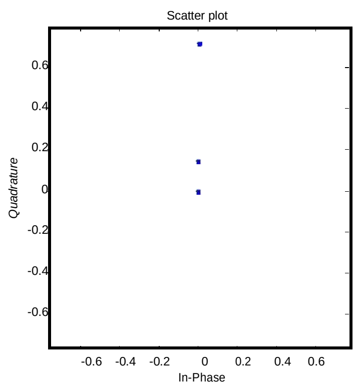

Figure 4.10 Estimated Channel with PSO-CS for Hmmse

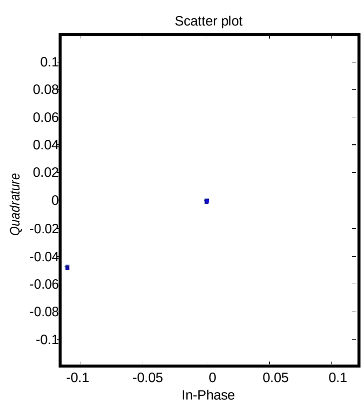

Figure 4.11 Estimated Channel with PSO-CS for mseMMSE1

These are the Scatterplot for Estimated Channel with PSO and CS for H, HhatLS, HhatLMMSE, HLS, Hls, mseLS1, HMMSE, Hmmse, mseMMSE, Fmin.

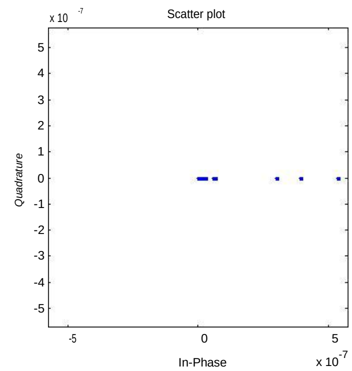

Figure 4.12 Estimated Channel with PSO-CS for Fmin

VI.CONCLUSION

An efficient channel estimation technique using PSO and CS algorithm for MIMO-OFDM systems is impleted using MATLAB coding. In channel estimation technique, the MMSE with 64-QAM is better than any other modulation schemes of LS and MMSE methods. The Channel Estimation of MMSE with 64-QAM is based on the PSO and CS algorithm and done directly in frequency domain. Combinig LS and MMSE using PSO and CS algorithm provides low MSE at high SNR. The implementation results prove that MMSE with PSO and CS algorithm in MIMO-OFDM system gives low Mean Square Error with high efficiency. In future it can be implemented in DSP processer kit.

VII.REFERENCES

1. Ehsan Valian, Shahram Mohanna and Saeed Tavakoli “Improved Cuckoo Search Algorithm for Global Optimization”, Journal of Communication and Information Technology, December 2011. Vol. 1 No.1.
2. Lenin.S and Malarkkan.S. “An Extensive Review of Significant Researches on Channel Estimation in MIMO-OFDM” Journal of Theoretical and Applied Information Technology, 20th June 2014. Vol. 64 No.2.
3. Sajjad Ahmed Ghauri and Sheraz Alam “Implementation of OFDM and Channel Estimation using LS and MMSE Estimators” Journal of Computer and Electronics Research, February 2013. Vol. 2 No.1.
4. Vidhya.K and Shankar Kumar.K.R, “Channel Estimation of MIMO-OFDM using Cuckoo Search Algorithm”, Journal of Theoretical and Applied Information Technology, 20th September 2013. Vol. 55 No.2 , PP. 260-272.