8th Biennial International Conference
& Exposition on Petroleum Geophysics

"HYDERABAD 2010"

P-302

Inversion of seismic data using a hybrid approach between Particle Swarm Optimization and VFSA

T.Vikranth Babu and Mrinal K. Sen***

Summary

*Optimization problems attempt to find out the minimum or maximum of a certain function (usually referred to as the cost function). The cost function can either be continuous or discrete. Discrete optimization problems arise, when the variables occurring in the optimization function can take only a finite number of discrete values and also subject to constraint conditions. In continuous optimization problems, variables can take a continuous set of real values. In recent years it has become clear that different application domains lend themselves to different solution techniques. Many heuristics have been developed which aim at finding a good or "**closely optimal**" solutions iteratively.*

*Fractals are one of the most beautiful mathematical figures. Many natural phenomena are modeled using fractals. Seismic data do not contain some low and high frequency information because of the band limited nature of the source wavelet. A deterministic inversion of such band limited seismic data produces smooth models which are devoid of high frequency variations. A stochastic inversion algorithm based on fractal models using VFSA algorithm has been implemented in **"R P Srivatsava and M K Sen"**. Here the same process has been implemented using a hybrid model of Very Fast Simulated Annealing (VFSA) and Particle Swarm Optimization (PSO).*

1. Introduction

Seismic data do not contain some low and high frequency information because of the band limited nature of the source wavelet. A deterministic inversion of such band limited seismic data produces smooth models which are devoid of high frequency variations.

Inversion of seismic data plays a vital role in reservoir characterization. High resolution inversion methods add significant value to the inversion results and increase the confidence level in interpretation of seismic data. Compared to raw seismic traces, an inversion provides more high resolution results which facilitate better interpretation.

Fractal based stochastic inversion of poststack seismic data using very fast simulated annealing **(VFSA)** has been implemented in **"R P Srivatsava and M K Sen"**. I have tried to implement the same using particle swarm optimization **(PSO)**, and a combination of **VFSA and PSO.**

Basically, poststack inversion is a process to analyze stacked seismic traces and estimate acoustic impedance structure of the earth. This estimation process relies on the fundamental relation between earth‟s reflectivity and observed seismic data given by 1-d convolution model.

$$s(t) = r(t) * w(t) + n(t)$$

Where s (t)is the observed seismic trace, w(t) is the source wavelet r(t) is the earth reflectivity and n (t) id noise.

Further reflectivity **(rk)** is related to acoustic impedance by:

$$r_k = \frac{I_{k+1} - I_k}{I_{k+1} + I_k},$$

where $Ik= \rho k\ vk\ and\ \rho k$ is density, vk is velocity $of\ the$ kth layer

Thus post stack inversion can be thought of estimation of reflectivity or acoustic impedance. From the above

**Indian Institute of Technology Madras, Chennai*
***Institute of Geophysics, University of Texas at Austin, Texas*

tvikrantbabu@gmail.com

equation, estimation of reflectivity is a linear inverse problem whereas estimation of acoustic impedance is a non linear inverse problem. To solve the non-linear problem of acoustic impedance VFSA has been used in "R.P.Srivatsava and M.K.Sen". I have modified this part of the code and applied PSO (particle swarm optimization). In the paper, fractal theory has been applied to generate prior models as an initial guesses to the VFSA optimization module.

So in simple terms, the objective is to determine ***r(t)*** from ***s(t)*** using ***w(t)*** (we ignore noise). Note that all the data is in the form of discrete samples. The equations specified in this section give the relation between these quantities. Using fractal theory, an initial random guess (s theoretical) is drawn. The objective function to be minimized the error function which can be:

$$\boldsymbol{F} = (\boldsymbol{s}_{\boldsymbol{observed}} - \boldsymbol{s}_{\boldsymbol{theoretical}})^2$$

Then the data is interpolated and inverted to estimate the acoustic impedance.

A Matlab code implementing this using VFSA has been done by R.P.Srivatsava. I have modified just the file VFSA implementation and applied particle swarm optimization and a combination of both VFSA and PSO.

More details of the implementation can be found in **"*R.P.Srivatsava and M.K.Sen*"**

2. Simulated annealing

Simulated annealing or SA is one of the many heuristic approaches used widely. The method is especially helpful for optimizing stochastic processes. It is often used when the search space is discrete. As the name implies, this method exploits an analogy of the physical annealing process i.e. the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and search for a minimum in a more general system.

2.1 A brief note on the annealing process

In the annealing process, the metal is first heated to a very high temperature. In this state's' all the particles are assumed to have a high internal energy E(s). After reaching this state of high energy the metal is cooled very slowly to a very low temperature. Slow cooling ensures thermal equilibrium at all times, then the atoms place themselves in a pattern that corresponds to the minimum global energy of a perfect crystal. A point to be noted is that particles in equilibrium at temperature T have the probability ***p*** to occupy energy E, i.e. observes the Boltzmann‟s condition given by,

$$p \propto \exp(-\frac{E}{kT})$$
$$where\ k = Boltzmann's\ constant$$

i.e. systems in equilibrium have a probability p of occupying the energy E. The internal energy E(s) is seen to be analogous to the cost function of the optimization problem. Simulated annealing is an iteratively improving algorithm. At each step, the SA heuristic considers some neighbor e' of the current state e, and probabilistically decides between moving the system to state e' or staying in state e. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically this step is repeated until the system reaches a state that is good enough for the application. The method is also an interpretation of the Markov chain, i.e. the next state is dependent on the current state only and is independent of all the previous states occupied. There are many variations of the SA algorithm like VFSA (very fast simulated annealing), the heat bath algorithm, etc.

2.2 Perturbation

In every iteration of the SA heuristic, the state has to be perturbed (slightly disturbed) to generate new solution. The perturbation can be implemented in different ways. We can select a random new position uniformly or in accordance with a density function, depending upon the prior knowledge of the solution. Typically we generate the new position based on a uniformly distributed density function.

2.3 Acceptance probability

If **P (e, e' , T)** is the probability that the new position e‟ is accepted over the current position e, where t is the current temperature, then one essential requirement of the probability function P is that it must be non zero when

e' >e. This means that the system may move to the new state even when it is worse (has a higher energy) than the current one. It is this feature that prevents the method from becoming stuck at a local minimum. On the other hand as T tends to 0 the probability **P (e', e', T)** must tend to 0. This ensures that for sufficiently small values of T the position does not change. One way to implement this is by

$$P(e^{'}, e, T) = \exp\left(-\frac{\Delta e}{T}\right), where\ \Delta e = e^{'} - e$$

2.4 The cooling schedule

The process must begin at a high temperature and must proceed by slowly decreasing the temperature up to a very low temperature (**T=0**) where all (majority of) the particles occupy a state of very low energy, which corresponds to the global minimum position. I have applied this method to few applications and the method proves to be a good method which gives reasonable solution.

2.5 VFSA (Very fast simulated annealing)

In typical SA the next position is chosen randomly over an interval. This random number is supposedly uniformly random over the interval, i.e. random but still uniform. We mean if we generate a large number of random numbers then all the numbers will be uniformly distributed over the interval. A modification of this is to generate the random number over some probability distribution. This is implemented in the VFSA, where the random number is generated over a **Gaussian distribution**. Upon doing so, the code seems faster and accurate.

3.0 Particle Swarm Optimization:

Particles swarm optimization **(PSO)** is a population based stochastic optimization technique developed by Dr. Elerhart and Dr. Kennedy in 1995, inspired by social behavior and of bird flocking or fish schooling. Swarm intelligence belongs to a class of direct search methods used to find an optimal (near optimal) solution to an objective function. The PSO is simple, intuitive and very basic. It imitates the social behavior, collective behavior of a population of animals and is used to optimize stochastic processes. The method simulates social behavior, and in every iteration tries to optimize the fitness function. The application is very simple, and the computer code is simple and fast to execute.

To understand the analogy, suppose a region of land has many hills and valleys and the objective is to find the location of the deepest valley (the optimum or the minimum). We employ say **'n'** workers for the purpose. Suppose every worker is provided with a device to communicate (say a cell phone) and a device which tells his position (latitude and longitude) and his height above sea level '**h'** (the value of the **fitness function**). Now one way is, every worker will wander randomly and individually and collect data (position and height above sea level) at discrete points, and in the end we combine all the data from all the workmen and find the position of the minimum. A better way would be to divide the region into **'n'** parts and assign one part to every person and ask him to collect data pertaining to his region and do the same operation. Finding the optimum in this manner is a '**direct search method**' and is typically very formal. Note that this method requires large amount of data storage. If every workman collects data at **50** positions we have **50n** values of data that needs to be stored. Also the number of iterations is huge.

The PSO solves this problem in a better manner. Initially the workmen are positioned randomly. Now all the workmen communicate and determine the best guess. Every workman now sets out in a direction that is pointing towards the best current guess. The best guess may change for every iteration. The velocity of each man and his path is slightly different from being a direct straight path. In PSO we implement this in a special manner. Now if any person comes across a better solution, all the other workmen now start converging to this position and also searching for better positions in the process. Observe that, at every iteration only **'n'** number of values have to be stored. Also in this method the accuracy of the ultimate solution is better because of the directional property. In this method the result obtained is much closer to the actual optimum. Note that the utility, the level of accuracy of the PSO is decided by the number of workmen employed i.e. the value of **'n'.**

The iteration is started with „**n**" particles located randomly and the position of each particle ***ith particle*** is given by $\vec{X}_i$ Let $\overrightarrow{XCAP}_i$ be the best position (i.e. the position with least value of the fitness function) attained by each particle. Let $\overrightarrow{Vi}$ be the velocity vector of each particle. Let $\vec{g}$ be the global best, i.e. best position of $\overrightarrow{XCAP}$ In every iteration the velocity vector of each particle is changed as

$$\vec{V} = w\vec{V} + c_1.r_1.(\overrightarrow{XCAP} - \vec{X}) + c_2.r_2.(\vec{g} - \vec{X})$$

where r_1 and r_2 are random numbers in $(0,1)$ and w, c_1, c_2 are PSO variables.

and hence the position vector of each particle is updated as

$$\vec{X} = \vec{X} + \vec{V}$$

The PSO update conditions are the key for the method. The particles tend towards the global best in every iteration. This feature explores a large number of points in the neighborhood of the actual minimum, which gives more accurate results. However the speed and accuracy of the program depends on the number of particles '**n**'and the PSO variables **w**, **c1, c2**. It has been found that **w= 0.5, c= 2** gives the best results. The loop can be terminated when all the particles are sufficiently close to each other.

4. The method

Our main objective is to minimize the error function given by

$$F = (s_{observed} - s_{theoretical})^2$$

where the observed data is obtained based on a fractal based initial guess and progressively the guess is improved by minimizing the error function using some optimization algorithm.

In the VFSA implementation, the initial guess comes from fractal inversion of seismic data and the next guess is randomly selected based on a Gaussian distribution. The error function is progressively minimized in the process.

In the PSO implementation, we start off with "n" guesses from the fractal inversion and in every step a new value is generated by the PSO algorithm.

In the hybrid approach, the new guess is generated using PSO algorithm, but here the random variables used (r1, r2) is generated from a Gaussian distribution as in the VFSA algorithm. This is the innovative part of the paper.

5. Results

On application the following results were obtained.

1) **VFSA: Figure 1** shows a plot of error function as a function of the iteration number, and also and also a comparison between the observed and inverted data, obtained using VFSA

   Number of iterations = 2000;
   Error = 0.255442;
   Program time = 106.2835 seconds.

2) **PSO: Figure 2** shows the plots of the error function as a function of the iteration number, and also a comparison between the observed and inverted data obtained using PSO.

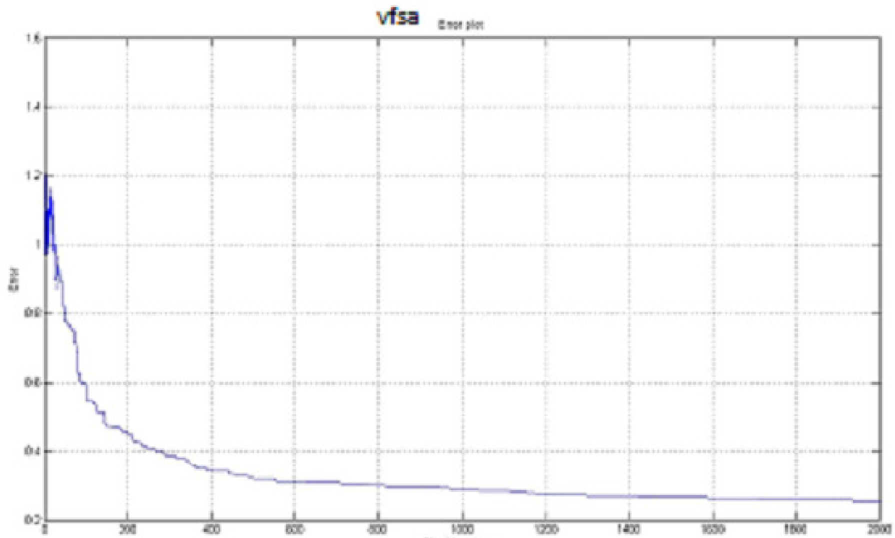
   Number of swarm particles employed = 10 Number of iterations = 1000
   Error obtained = 0.230287
   Program time = 81.8693 seconds

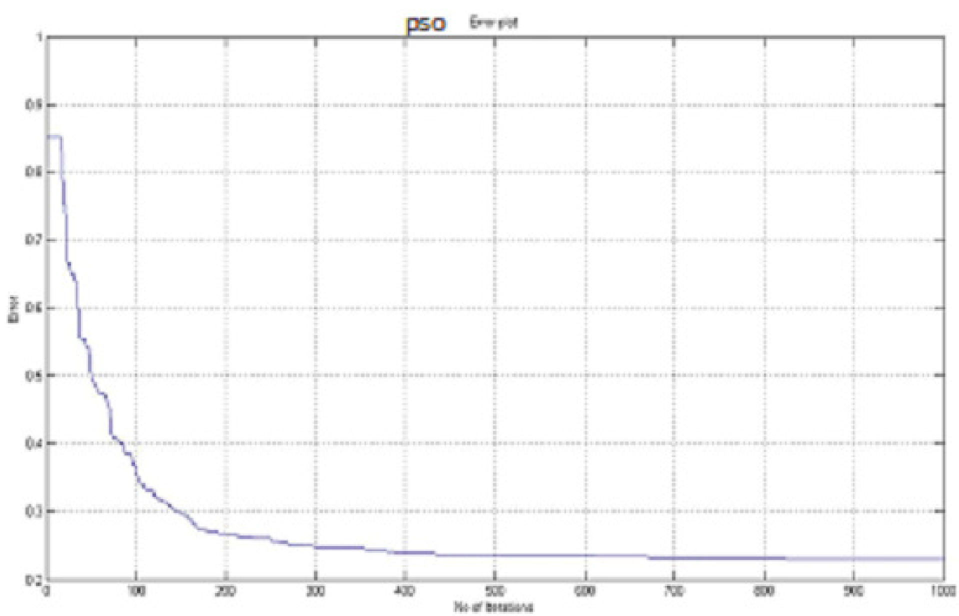
3) **Hybrid model of VFSA and PSO: Figure 3** shows the plots of error function as a function of the iteration number and a comparison between observed and inverted data, obtained using a hybrid model of VFSA and PSO.

   Number of swarm particles = 10
   Number of iterations = 1000
   Error obtained = 0.2446
   Program time = 81.1205 seconds

6. Conclusions

A stochastic inversion of seismic data using a fractal based prior model facilitates to achieve models with realistic frequency. A fractal based model is used to get an intelligent initial guess, and the VFSA code for seismic data inversion applied by R.P.Srivatsava and M.K.Sen has been replaced by a hybrid approach between PSO and VFSA. The method shown in this paper improves the initial guess using a code which is in between a typical PSO and VFSA code.


vfsa


pso

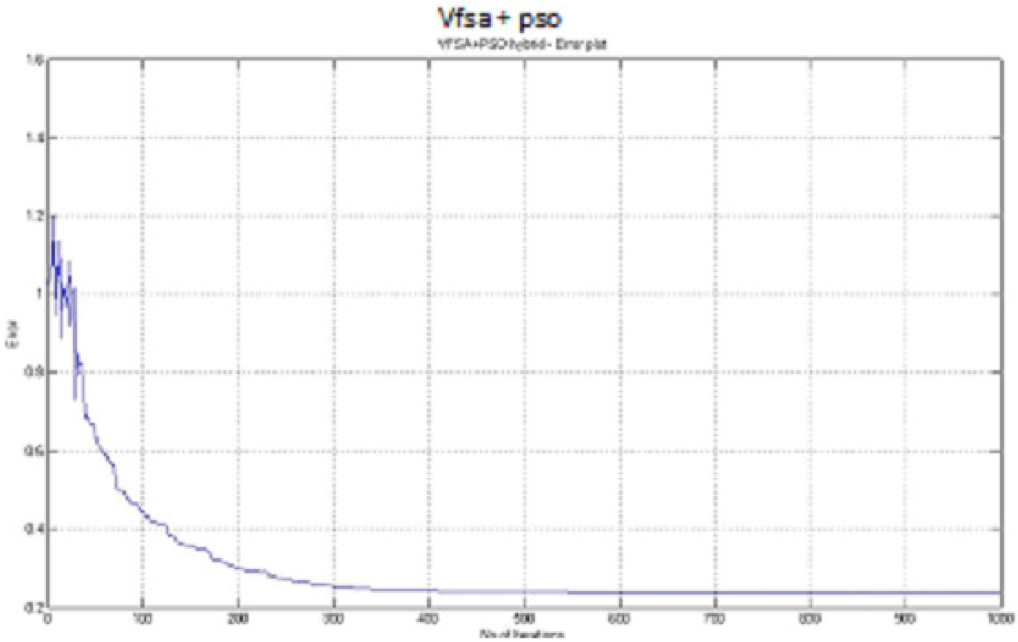
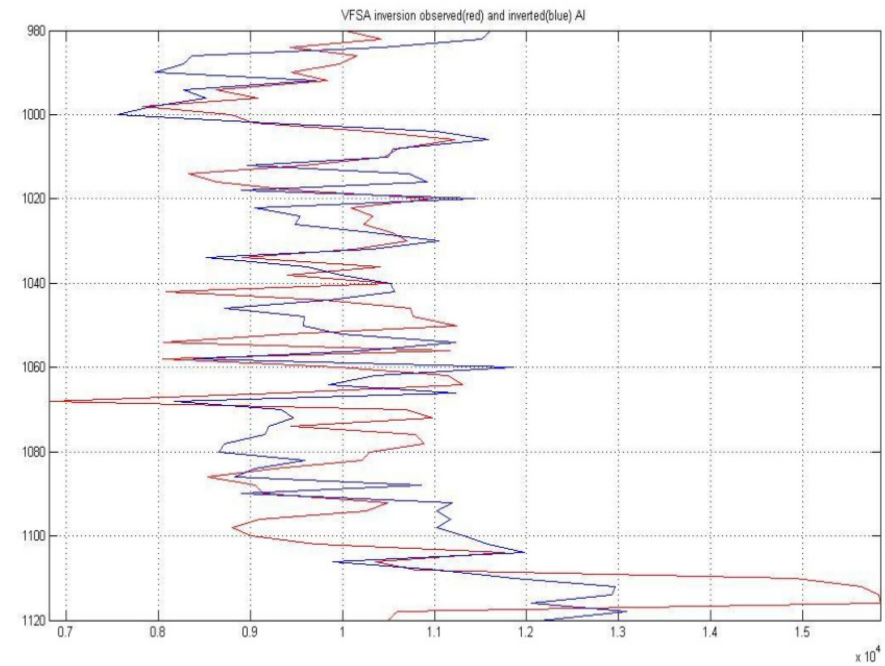


Figure showing the error plots using the three methods


VFSA inversion observed(red) and inverted(blue) AI
980
1000
1020
1040
1060
1080
1100
1120
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
x 10^4

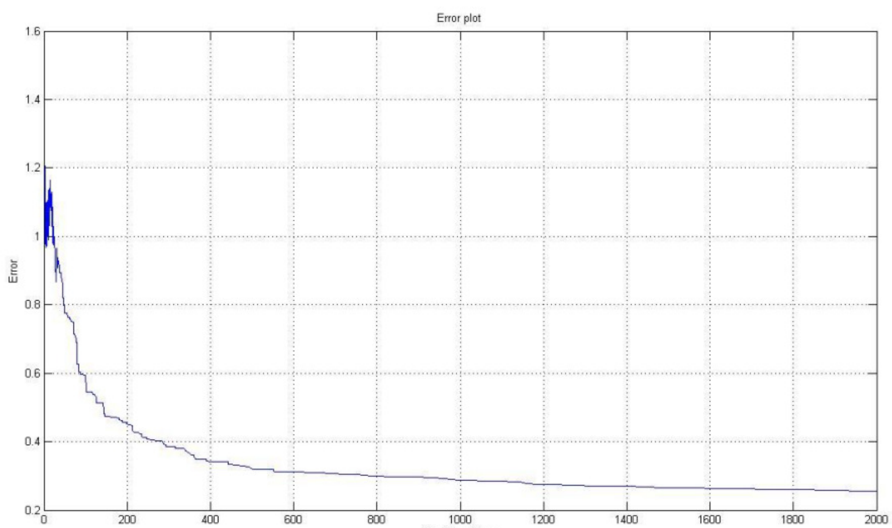

Error plot
Error
No of Iterations
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
200
400
600
800
1000
1200
1400
1600
1800
2000

Figure 1

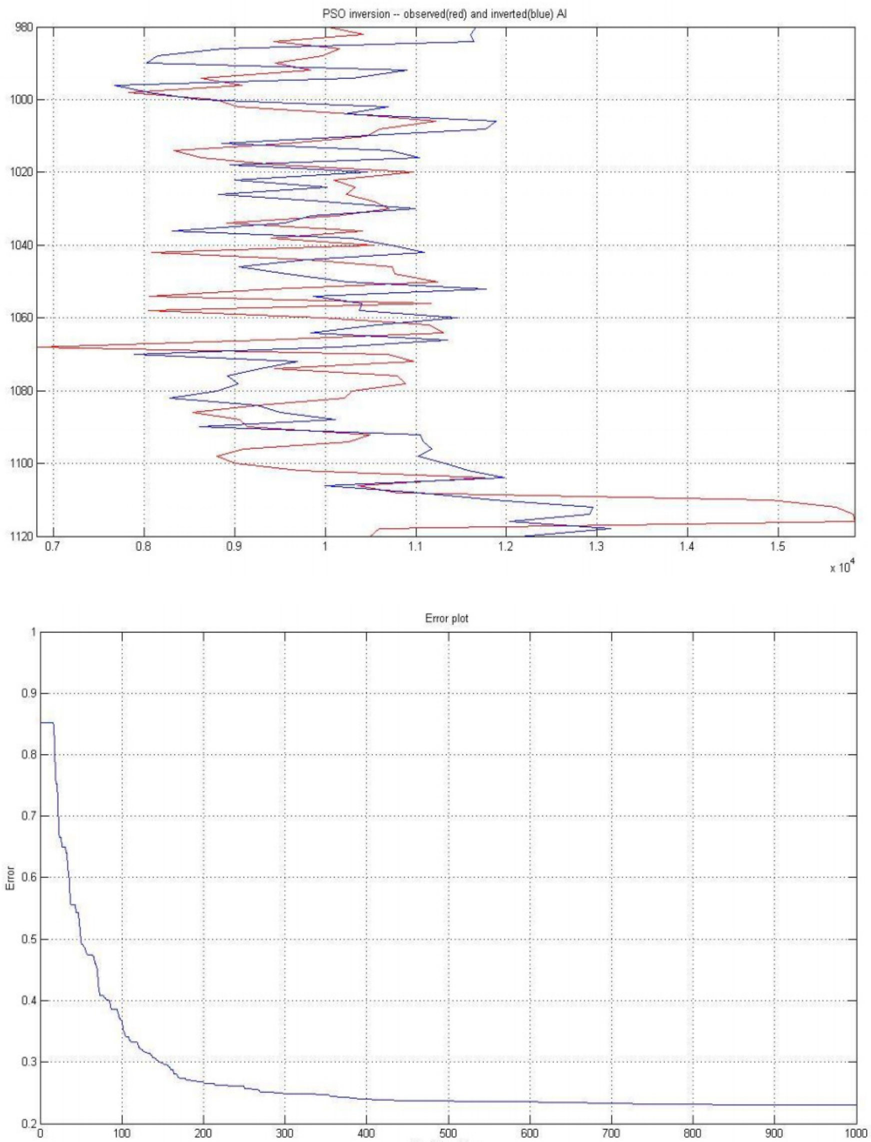
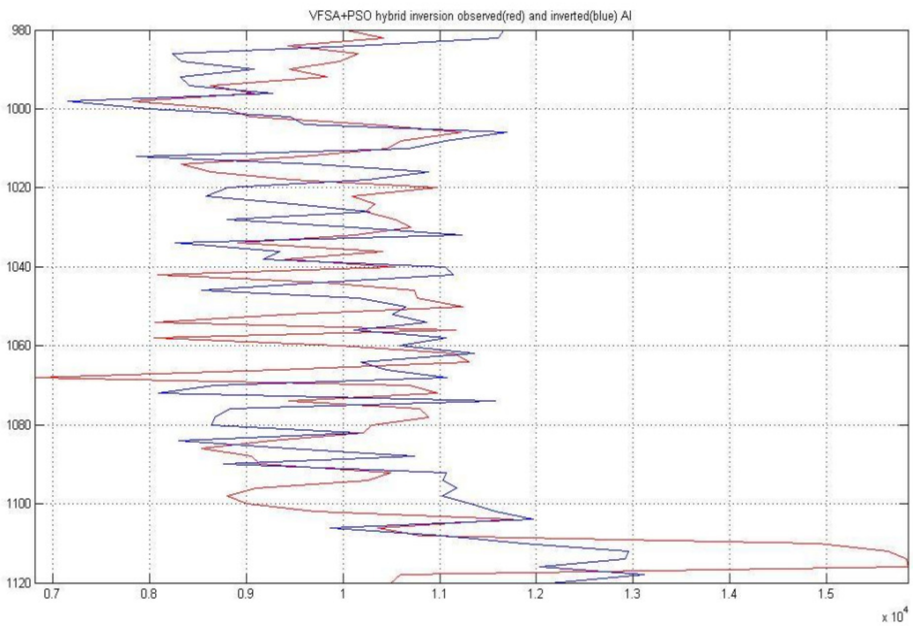

PSO inversion -- observed(red) and inverted(blue) AI
980
1000
1020
1040
1060
1080
1100
1120
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
x 10^4
Error plot
Error
No of Iterations
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800
900
1000

Figure 2


VFSA+PSO hybrid inversion observed(red) and inverted(blue) AI
980
1000
1020
1040
1060
1080
1100
1120
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
x 10^4

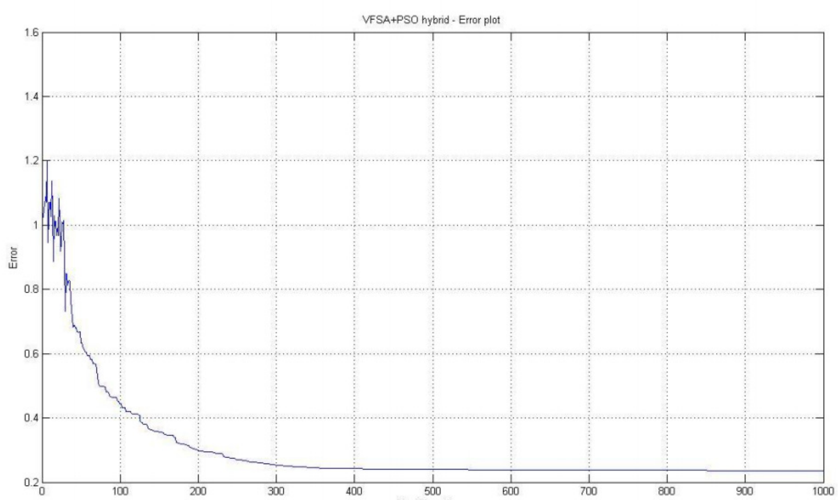

VFSA+PSO hybrid - Error plot
Error
No of Iterations
0
100
200
300
400
500
600
700
800
900
1000
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

Figure 3