Interface Identification using a GPR Signal: A Monte Carlo Markov Chain Approach.

Arnaud Coatanhay, Jean-Jacques Szkolnik,

Laboratoire en Extraction et Exploitation d'Infomation en Environement Incertain (E^3I^2), ENSIETA, 29806 Brest cedex 9, France. E-mail: coatanar@ensieta.fr .

Abstract— **This paper presents a new signal processing method to improve the identification of interface between different layered media, using a Ground Penetrating Radar (GPR) recording. Our methological approach is based on Monte Carlo Markov Chain (MCMC) model. The deconvolution of the GPR signal is obtained in considering a stochastic estimation related to a maximum a posteriori criterion. The only known elements are the signal recorded from the GPR backscattering (one dimension approximation), and the order of the ARMA signal model for the emitted pulse.**

Keywords— **GPR, Markov Chain, Layered media.**

I. Introduction

In this paper, the backscattering of a normal incidence radar wave by the ground is appoximated to a one dimension propagation problem. So, the characteristics of the ground is supposed to be unvarying (or slowly changeable) with regard to a lateral shift.

In fact, the ground is supposed to be an area made of horizontally stratified media, and the purpose of our study is to determine the interface positions between the homogeneous layers.

The electromagnetic wave emitted by the Ground Penetrating Radar (GPR), that must be considered as an approximately known signal, is reflected by each interface. In any cases, the emitted burst is not a perfect dirac impulse, and the identification of each interface reflection raises a great ambiguity problem for close layers.

For a linear approximation, the observed backscattered signal can be modelled as the result of a convolution product:

$$z\left(t\right) = h\left(t\right) * r\left(t\right) + w\left(t\right) \ , \tag{1}$$

where $*$ is the canonical convolution product, $z\left(t\right)$ is the signal coming from the subsurface, $h\left(t\right)$ is the emitted wave, $r\left(t\right)$ is the reflectivity sequence and $w\left(t\right)$ is a gaussian noise signal ($\mathcal{N}\left(o, \sigma_w^2\right)$).

The interface detection problem boils down to the determination of the reflectivity sequence $r\left(t\right)$ from the recorded signal $z\left(t\right)$. Theoretically, the $z\left(t\right)$ signal is a sequence of dirac impulses, situated at the exact interface positions. Mathematically speaking, the determination of the reflectivity sequence constitutes a linear inverse problem.

The standard or blind deconvolution constitues a very common problem in many application fields (signal transmission, seismic propagation, etc...), and different algorithms have been already developed, see for example [1] and [2].

But, for electromagnetic propagation, due to the conductivity and the dispersivity of the ground, the characteristics of emitted wave deeply change in function of the depth, when the wave propagates in the ground. And most of time, the deconvolution algorithms are not robust enough to efficiently process GPR signals. So, we must investigate a new approach that does not require a too restritive definition of the emitted pulse.

The discretization of the emitted wave yields a Z-transform expression of $h\left(t\right)$ (ARMA model):

$$h\left(z\right) = \frac{b\left(z\right)}{a\left(z\right)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + \cdots + a_p z^{-p}} \ . \tag{2}$$

Then, the discretized state-space model related to the convolution product (1) is in the form:

$$\begin{aligned} \mathbf{x}(\mathrm{k}) &= \mathrm{A}\cdot\mathbf{x}(\mathrm{k}-1) + \mathrm{B}\cdot \mathrm{r}\left(\mathrm{k}\right) \ , \\ z\left(k\right) &= \mathrm{C}\cdot\mathbf{x}(\mathrm{k}) + \mathrm{w}\left(\mathrm{k}\right) \ , \end{aligned} \tag{3}$$

where $\mathbf{x}= \left(\mathrm{k}\right) = \left(\mathrm{x}\left(\mathrm{k}\right), \cdots, \mathrm{x}\left(\mathrm{k}-\mathrm{p}\right)\right)^{\mathrm{T}}$ is the state variable vector, $\mathrm{B} = \left(\begin{array}{cc} 1 & 0_{p\times 1} \end{array}\right)^{\mathrm{T}}$ and z is the recorded signal.

Letting $\mathbf{a}= \left[a_1, a_2, \cdots, a_p\right]^T$ and $\mathbf{b}= \left[b_1, b_2, \cdots, b_q\right]^T$, we have:

$$\begin{aligned} \mathrm{A} &= \left(\begin{array}{cc} -\mathbf{a}^{\mathbf{T}} & 0 \\ I_{p\times p} & 0_{p\times 1} \end{array}\right) , \\ \mathrm{C} &= \left(\begin{array}{ccc} 1 & \mathbf{b}^{\mathbf{T}} & 0_{1\times(p-q)} \end{array}\right) . \end{aligned} \tag{4}$$

The $\mathbf{a}$ and $\mathbf{b}$ vectors are unknown parameters to be estimated. The only given parameters are p and q values (order of the ARMA model).

For estimation of the interface positions, the dimension of the complete reflectivity sequence represents too many unknown parameters for efficient direct identification. In consequence, more prior information must be given to find the solution of this inverse problem.

In reference to other field studies (seismic and nuclear sciences, see [3]), the reflectivity sequence is supposed

0-7803-7357-X/02/$17.00 ©2002 IEEE

to be the result of a stochastic Bernoulli-Gaussian process. This Bernoulli-Gaussian process is characterized by (q_s, r_s) values. The q_s term is a stochastic value associated with two states: $q_s = 1$ if there is a reflector (interface) or $q_s = 0$ for not a reflector.

Obviously, letting $P(q_s = 1)$ (resp. $P(q_s = 0)$) the probability that $q_s = 1$ (resp. $q_s = 0$), we can write $P(q_s = 1) = 1 - P(q_s = 0) = \lambda$. The amplitude of the reflectivity signal is represented by r_s that is a two state gaussian stochastic value:

$$\begin{array}{l} r_s \sim \mathcal{N}(0, \sigma_1^2) \quad si\ q_s = 1\ , \\ r_s \sim \mathcal{N}(0, \sigma_0^2) \quad si\ q_s = 0\ . \end{array} \tag{5}$$

The vectors of every q_s and r_s values are denoted $\mathbf{q_{1\rightarrow N}} = (\mathbf{q_s})$ and $\mathbf{r_{1\rightarrow N}} = (\mathbf{r_s})$.

Finally, λ, σ_w, σ_0 and σ_1 are undetermined parameters in the state-space model. And the blind deconvolution process must try to estimate the reflectivity sequence $\mathbf{r_{1\rightarrow N}}$. More, in practice, due to physical coupling phenomenon and electronic incertainty, the incident wave h can not be considered as a determinist one. So, the model of the incident wave and ground interaction involves stochastic approach.

II. Monte Carlo Markov Chain approach

In fact, to solve this deconvolution problem, the statistical methology, called Monte Carlo Markov Chain (MCMC), seems to be one of the most appropriate tool. In few words, the MCMC methods consist in evaluating the unknown parameters using random sampling. More acurately, this stochastic process corresponds to a Markov Chain, obtained by optimization of the maximum a posteriori criterion.

A. Bayesian criterion

For the Bernoulli-Gaussian process $(\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})$, the a posteriori criterion related to the observed sequence $\mathbf{z_{1\rightarrow N}} = (\mathbf{z}(\mathbf{k}))$ is the probability density: $p((\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})|\mathbf{z_{1\rightarrow N}})$. According to the Bayes relation, we can write:

$$p((\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})|\mathbf{z_{1\rightarrow N}}) = \frac{p(\mathbf{z_{1\rightarrow N}}|(\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})) \cdot p(\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})}{p(\mathbf{z_{1\rightarrow N}})}\ . \tag{6}$$

In fact, the probability $P(\mathbf{z_{1\rightarrow N}})$ have no influence on the optimization criterion. And, the bayesian criterion is in the form:

$$\begin{array}{l} p((\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})|\mathbf{z_{1\rightarrow N}}) \propto \\ \qquad p(\mathbf{z_{1\rightarrow N}}|(\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})) \\ \qquad \cdot p(\mathbf{r_{1\rightarrow N}}|\mathbf{q_{1\rightarrow N}}) \cdot P(\mathbf{q_{1\rightarrow N}})\ . \end{array} \tag{7}$$

As $(\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}})$ is a Bernoulli-Gaussian process, we can easily show that:

$$\begin{array}{l} P(\mathbf{q_{1\rightarrow N}}) = \prod_{k=1}^{N} P(q(k))\ , \\ P(q(k)) = \lambda^{q(k)}(1-\lambda)^{1-q(k)}\ , \end{array} \tag{8}$$

and:

$$p(\mathbf{r_{1\rightarrow N}}|\mathbf{q_{1\rightarrow N}}) = \prod_{k=1}^{N} p(r(k)|q(k))\ ,$$

$$\begin{array}{l} p(r(k)|q(k)) = \\ \left[\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-r^2(k)/\sigma_1^2}\right]^{q(k)} \left[\frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-r^2(k)/\sigma_0^2}\right]^{1-q(k)}\ . \end{array} \tag{9}$$

The remaining problem is that the probability $p(\mathbf{z_{1\rightarrow N}}|(\mathbf{q_{1\rightarrow N}}, \mathbf{r_{1\rightarrow N}}))$ can not be expressed as such a simple product form. To estimate this probability, we use a random sampling process (Monte Carlo) simulation.

B. A priori information

To generate random drawings, Monte Carlo simulation requires a priori information about the statistical characteristics of the random variables. In a very common way for bayesian estimation, we suppose the a priori distributions of $\lambda, \sigma_0, \sigma_w, a, b$ values are in the form:

$$\lambda \sim \mathcal{B}(\zeta, \tau)\ , \tag{10}$$

$$\sigma_0^2 \sim \mathcal{IG}\left(\frac{v_0}{2}, \frac{\gamma_0}{2}\right)\ , \tag{11}$$

$$\sigma_w^2 \sim \mathcal{IG}\left(\frac{v_w}{2}, \frac{\gamma_w}{2}\right)\ , \tag{12}$$

and

$$\left(a^T, b^T\right)^T \sim \mathcal{N}\left(0_{(p+q)\times 1}, \delta^2 I_{p+q}\right)\ . \tag{13}$$

Moreover, we suppose $\sigma_1^2 = \alpha\sigma_0^2 \sim \alpha\mathcal{IG}\left(\frac{v_0}{2}, \frac{\gamma_0}{2}\right)$ avec $(\alpha \ll 1)$.

The $(\zeta, \tau, v_0, \gamma_0, v_w, \gamma_w, \delta, \alpha)$ parameters are given coefficients that must be arbitrarily determined. In the following, we let $\nu = \left(\lambda, \sigma_0^2\right)$.

C. The random sampler

The random sampler is a very general algorithm to randomly simulate unknown variables from conditionnal probability. In our case the random sampler proceeds as follows:

- Initialization:

Specify arbitrary starting values for the vector $\left(\mathbf{a}^{(0)}, \mathbf{b}^{(0)}, \mathbf{x}_{1\rightarrow N}^{(0)}, \mathbf{q}_{1\rightarrow N}^{(0)}, \sigma_{\mathrm{w}}^{2(0)}, \nu^{(0)}\right)$.

- Iteration i, $i \geq 1$:

1. Simulate $\left(\sigma_w^{2(i)}, \nu^{(i)}\right)$ from
$p\left(\left(\sigma_w^2, \nu\right) | \left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \mathbf{x}_{1\rightarrow \mathrm{N}}^{(\mathbf{i}-\mathbf{1})}, \mathbf{q}_{1\rightarrow \mathrm{N}}^{(\mathrm{i}-1)}, \mathbf{z}_{1\rightarrow \mathrm{N}}\right)\right)$.

2. For $k = 1, \cdots, N$, simulate $q^{(i)}(k)$ using a Metropolis-Hastings algorithm (see [4], [5]) from
$P\left(q(k) | \left(\mathbf{q}_{-\mathrm{k}}^{(\mathrm{i})}, \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}, \mathbf{z}_{1\rightarrow \mathrm{N}}\right)\right)$.

3. Simulate $\left(\mathbf{a}^{(\mathrm{i})}, \mathbf{b}^{(\mathrm{i})}\right)$ using a Metropolis-Hastings algorithm from
$p\left(\left(\mathbf{a}^{(\mathrm{i})}, \mathbf{b}^{(\mathrm{i})}\right) | \left(\mathbf{q}_{1\rightarrow \mathrm{N}}^{(\mathrm{i})}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}, \mathbf{z}_{1\rightarrow \mathrm{N}}\right)\right)$.

4. Simulate $\mathbf{x}^{(\mathrm{i})}_{1\to\mathrm{N}}$ from
$p\left(\mathbf{x}_{1\to\mathrm{N}}|\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \mathbf{q}^{(\mathrm{i})}_{1\to\mathrm{N}}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu_0^{(\mathrm{i})}, \mathbf{z}_{1\to\mathrm{N}}\right)\right)$.

- $\cdots$

where:

$$\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}} =$$

$$\left(\mathrm{q}^{(\mathrm{i})}(1), \cdots, \mathrm{q}^{(\mathrm{i})}(\mathrm{k}-1), \mathrm{q}^{(\mathrm{i}-1)}(\mathrm{k}+1), \cdots, \mathrm{q}^{(\mathrm{i}-1)}(\mathrm{N})\right)$$

The simulation of every probability density is obtained using specific algorithms, described in the following part.

D. The algorithm implement

In this iterative process, the first step (simulation of $\left(\sigma_w^{2(i)}, \nu^{(i)}\right)$) is directly obtained and does not raise specific problems.

On the opposite, the simulation of the $q^{(i)}(k)$ missing value indicator from
$P\left(q(k) \mid \left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}, \mathbf{z}_{1\to\mathrm{N}}\right)\right)$,
requires a more sophisticated process. First, it is noteworthy that:

$$P\left(q(k) \mid \left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}, \mathbf{z}_{1\to\mathrm{N}}\right)\right) =$$

$$\frac{p\left(\mathbf{z}_{1\to\mathrm{N}}|\left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathbf{q}(\mathbf{k})\, \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}\right)\right)}{p\left(\mathbf{z}_{1\to\mathrm{N}}|\left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}\right)\right)}$$

$$\times P\left(q(k)\, | \nu^{(i)}\right) \qquad (14)$$

where, for given $q(k)$, the probability density $p\left(\mathbf{z}_{1\to\mathrm{N}}|\left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathbf{q}(\mathrm{k})\, \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}\right)\right)$, that is the likelihood of the linear gaussian dynamic model, can be estimated with a Kalman filter. Then, the simulation is obtained using a Metropolis-Hasting algorithm with the a priori distribution $P\left(q(k)\, |\nu^{(i)}\right)$ as proposal distribution. The acceptance probability is in the form (see [6]):

$$\alpha\left(q^{(i)}(k), q^*(k)\right) = min$$

$$\left\{1, \frac{p\left(\mathbf{z}_{1\to\mathrm{N}}|\left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathrm{q}^*(\mathrm{k}), \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}\right)\right)}{p\left(\mathbf{z}_{1\to\mathrm{N}}|\left(\mathbf{q}^{(\mathrm{i})}_{-\mathrm{k}}, \mathrm{q}^{(\mathrm{i})}(\mathrm{k}), \mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}, \sigma_{\mathrm{w}}^{(\mathrm{i})}, \nu^{(\mathrm{i})}\right)\right)}\right\}$$

Generally speaking, the Metropolis-Hastings algorithm sample from a joint distribution for $X = (X_1, \cdots, X_n)$, by repeatedly considering randomly generated changes to the components of X, accepting or rejecting these changes based on how they affect the probability of the state. This process can be seen as the operation of a Markov chain built from a set of base transition probabilities, B_k, for $k = 1, \cdots, n$. The way transition B_k operates to generate a new state, x', from the current state x, can be described as follows:

1. Select a candidate state, x^*, in wich all components other than the kth are the same as in x, while x_k^* is picked at random from a proposal distribution, wich may depend on x, given by the probabilities $S_k\left(x, x_k^*\right)$.

2. Accept this candidate state with probability $A_k\left(x, x^*\right)$; otherwise, reject it, and retain the current state. In detail, this can be done by generating a random number, u, from the uniform distribution on $[0,1]$, and then setting the next state as follows:

$$x' = \begin{cases} x^* & if\ u < A_k\left(x, x^*\right) \\ x & otherwise \end{cases}$$

To simulate the coefficients of the ARMA model $\left(\mathbf{a}^{(\mathrm{i})}, \mathbf{b}^{(\mathrm{i})}\right)$, we used a Metropolis-Hastings algorithm too, with a gaussian proposal distribution:

1. We simulate a candidate $\left(\mathbf{a}^{(\mathrm{i})}, \mathbf{b}^{(\mathrm{i})}\right) \sim \mathcal{N}(\mathrm{m}, \Sigma)$, Where:

$$m = \begin{pmatrix} \mathbf{a}^{(\mathrm{i}-1)} \\ \mathbf{b}^{(\mathrm{i}-1)} \end{pmatrix} + \mu\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right)$$

with:

$$\mu\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right) = \frac{\Delta_{ln}}{2\ max\left(\delta_\mu, ||\Delta_{ln}||\right)}$$

and:

$$\Delta_{ln} =$$

$$\delta_\mu \nabla_{\mathbf{a},\mathbf{b}} \ln p\left((\mathbf{a}, \mathbf{b}) \mid \left(\mathbf{q}^{(\mathrm{i})}_{1\to\mathrm{N}}, \sigma_{\mathbf{w}}^{(\mathbf{i})}, \nu^{(\mathbf{i})}, \mathbf{z}_{1\to\mathrm{N}}\right)\right)_{|\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}}$$

2. We compute the acceptance probability:

$$\alpha\left(\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right), (\mathbf{a}', \mathbf{b}')\right) =$$

$$min\left\{\frac{p\left((\mathbf{a}', \mathbf{b}') \mid \left(\mathbf{q}^{(\mathrm{i})}_{1\to\mathrm{N}}, \sigma_{\mathbf{w}}^{(\mathbf{i})}, \nu^{(\mathbf{i})}, \mathbf{z}_{1\to\mathrm{N}}\right)\right)}{p\left(\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right) \mid \left(\mathbf{q}^{(\mathrm{i})}_{1\to\mathrm{N}}, \sigma_{\mathbf{w}}^{(\mathbf{i})}, \nu^{(\mathbf{i})}, \mathbf{z}_{1\to\mathrm{N}}\right)\right)}\right.$$

$$\times e^{\left[-\frac{1}{2\delta_\Sigma^2}\left(\mu\left(\mathbf{a}', \mathbf{b}'\right)+\mu\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right)\right)^T\left(2\begin{pmatrix} \mathbf{a}' - \mathbf{a}^{(\mathrm{i}-1)} \\ \mathbf{b}' - \mathbf{b}^{(\mathrm{i}-1)} \end{pmatrix}\right.\right.}$$

$$\left.\left.\left.+\left(\mu\left(\mathbf{a}', \mathbf{b}'\right) - \mu\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right)\right)\right)\right]\right\}$$

3. We simulate $u \sim \mathcal{U}_{[0,1]}$.
If $u < \alpha\left(\left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right), (\mathbf{a}', \mathbf{b}')\right)$ then $\left(\mathbf{a}^{(\mathrm{i})}, \mathbf{b}^{(\mathrm{i})}\right) = (\mathbf{a}', \mathbf{b}')$, else $\left(\mathbf{a}^{(\mathrm{i})}, \mathbf{b}^{(\mathrm{i})}\right) = \left(\mathbf{a}^{(\mathrm{i}-1)}, \mathbf{b}^{(\mathrm{i}-1)}\right)$

$\delta_\Sigma^2 > 0$ and δ_{mu} are constant values.

The proposal distribution results from the temporal discretization of the scattering Langevin process.

Finally, to simulate the hidden state process $\mathbf{x}_{1\to N}$, we have to use a standard Kalman smoother, see [7].

The random sampler algorithm enable a stochastic simulation of the global Monte Carlo Markov Chain. Nevertheless, the main purpose of our study is to obtain the reflectivity sequence $\mathbf{r}_{1\to N}$. With regard to the impulse deconvolution problem, the reflector position estimation remains a great problem. To estimate the reflector positions, the problem yields to a nonblind deconvolution using previously computed parameters of the MCMC model.

In concrete terms, we use the previously described MCMC algorithm to generate stochastic sample, and we optimize the maximum a posteriori criterion. Then, the parameter of the MCMC can be consider as known values (estimation). To determine the $\mathbf{q}_{1\to N}$ sequence, we use again the random sampler algorithm with this known parameters, and we implement a Kalman smoother (see [7]) for the dynamic noise estimation. This operation involves the determination of the reflectivity sequence $\mathbf{r}_{1\to N}$.

III. Numerical results

In theory, the previously described MCMC methodology is dedicated to the estimation of a great number of unknown parameters. In practice, the numerical convergence is impossible if the dimension of the state-space model is not low. In the present study, the incident wave $h(k)$ was a wavelet that looks like a ricker (see figure 1), and we admitted that this signal could be approximated with an ARMA(2,1) model. Even for this low dimensional case, our simulation show that a consistent deconvolution is obtained for more than 25000 iterations in the random sampler process. In fact, the approximation by an nonminimum phase ARMA(2,1) model provides good numerical results, but this model involves a constant phase shift between the real reflectivity sequence and the estimated one.

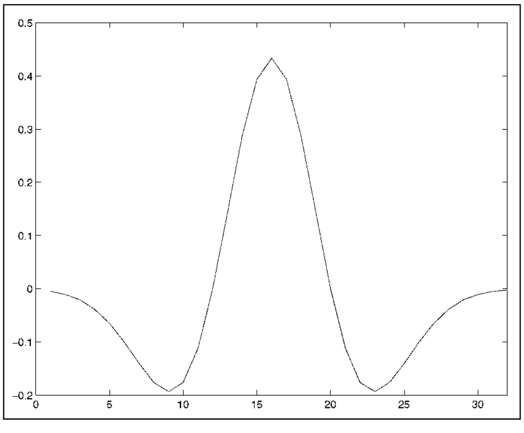

Fig. 1. Approximative shape of the emitted pulse

For example, the figure (2) shows a synthetic reflectivity sequence corresponding to a subsurface space with five homogeneous layered media. The observed signal $z(k)$ is the convolution product of this reflectivity sequence by the emitted ricker plus white gaussian noise ($10dB$ signal to noise ratio). And then, the estimated reflectivity sequence is given at figure (3). The reflector detection is quite correct if the constant phase shift is taken into account.

Unlike many other methods, the MCMC algorithm is a deconvolution methology that does not suppose a perfectly known emitted wave. In consequence, this approach constitutes a very robust deconvolution process if the homogeneous media are dispersive and conductive. The distortion have less influence on the deconvolution process if the algorithm identifies the GPR wave with an approximated and stochastic model. In this conditions, the deconvolution process using a Monte Carlo Markov Chain approach give quite good results for detection of deep interfaces.

In this example, the deconvolution estimation was done considering the one dimensional problem. In experimental conditions, the data usually correspond with a two dimension recording: radargram image. So, the same deconvolution methodology can be used for each position signal, and, if no abrupt variation is admitted for the interface position, a smoother forward-backward filter enables to emphasize the interface, when the signal to noise ratio is low.

IV. Conclusions

Although the MCMC methods are very time consuming algorithms, this deconvolution methods is a very powerful tool to analyse the GPR image of complex subsurface space: different layered media, dispersivity, low signal to noise ratio.

The MCMC approach constitutes a robust approach that is not very influenced with the emitted pulse distortion. However, the MCMC model does not take into account an explicit evolution of the emitted pulse during the propagation. So other stochastic approaches can be investigated (particle filter theory for example, see [6]) to solve the distortion problem.

References

[1] M. D. Sacchi and D. R. Velis and A. H. Cominguez, *Minimum entropy deconvolution with frequency-domain constraints*, Geophysics, Vol. 59, No. 6, 1994, pp. 938-945.

[2] A. Doucet and P. Duvaut, *Bayesian estimation of state-space models applied to deconvolution of Bernoulli-Gaussian processes*, Signal Processing, Vol. 57, 1997, pp. 147-161.

[3] M. Lavielle, *Bayesian Deconvolution of Bernouilli-Gaussian Processes*, Signal Processing, Vol. 33, 1993, pp. 67-79.

[4] W.K. Hastings, *Monte-Carlo sampling methods using Markov chains and their applications*, Biometrika, Vol. 57, 1970, pp. 97-109.

[5] R. M. Neal, *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, Technical report CRG-TR-93-1, Departement of Computer Science University of Toronto.

[6] A. Doucet, N. J. Gordon and V. Krishnamurthy, *Particle Filters for State Estimation of Jump Markov Linear Systems*, IEEE Trans. on signal. processing, Vol. 49, No. 3, 2001, pp. 613-623.

[7] P. Dejong and N. Shepard, *The simulation smoother for time series models*, Biometrika, Vol. 82, No. 2, 1995, pp. 339-350.

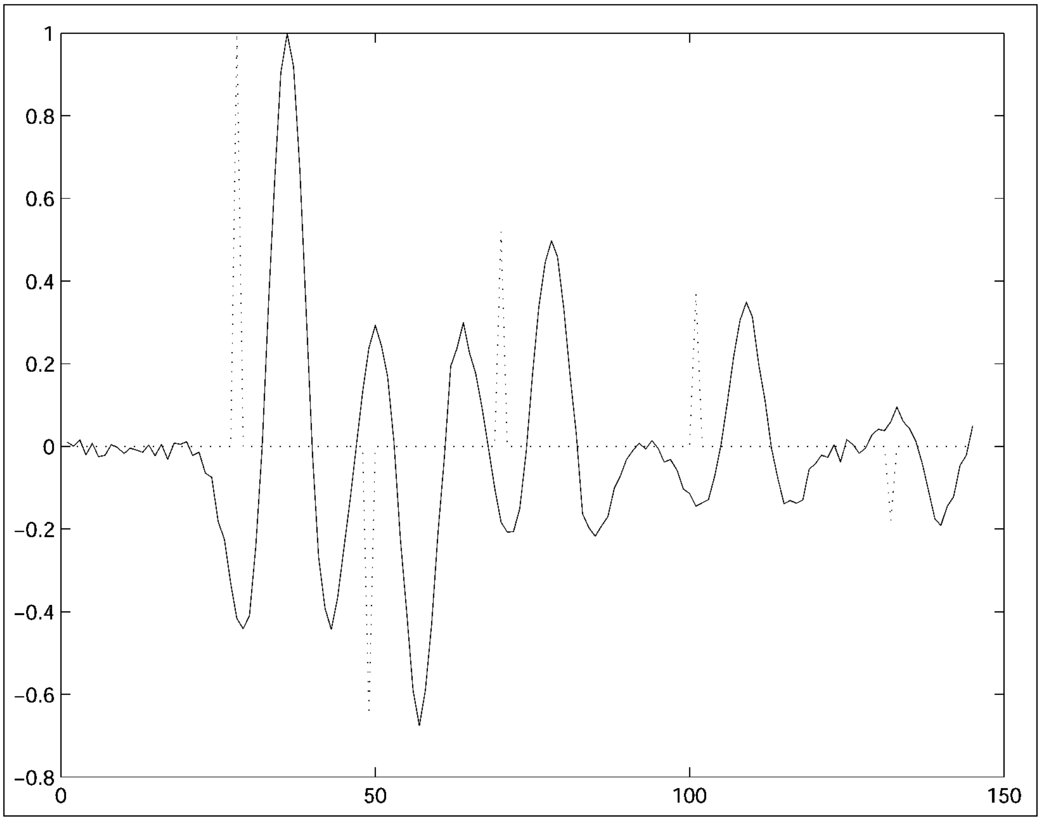

Fig. 2. The reflectivity sequence (dotted line) and the observed signal (plain line)

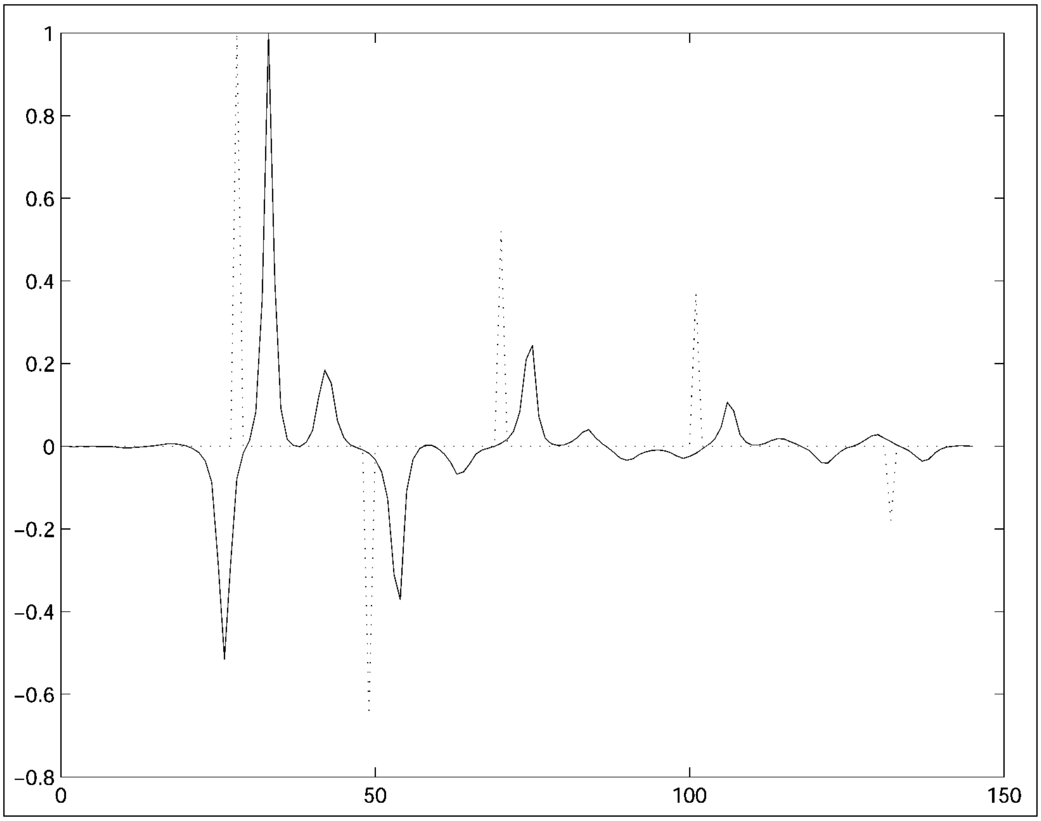

Fig. 3. The reflectivity sequence (dotted line) and the estimated reflectivity sequence (plain line)