Tomography MCMC Method Exploration Software Package

B. Matusiak, K. Grudzień, A. Romanowski, L. Jackowska-Strumiłło

Computer Engineering Department, Technical University of Łódź (bmatus@kis.p.lodz.pl)

Summary: Statistical methods such as Monte Carlo Markov Chain (MCMC) coupled with Bayesian approach proved its usefulness to the Electrical Capacitance Tomography (ECT) inverse problem solution. However, this methodology is very sensitive to the values of method parameters. The dedicated tool for investigation of this statistical method and for algorithm adjusting, testing, verification and optimization is proposed in the paper.

Keywords: ECT, MCMC, Mac OS X tomography software

1. Introduction

In Electrical Capacitance Tomography (ECT) the solution of inverse problem is usually aiming at obtaining the tomographic image in the process of image reconstruction (Lionheart, 2004; Yang et al., 1999). Speed and accuracy of the inverse problem solution in ECT is still a challenge. Statistical methods such as Monte Carlo Markov Chain (MCMC) coupled with Bayesian approach proved its usefulness to image reconstruction (West et al., 2004; Kaipio et al., 2000; Romanowski et al., 2004). Moreover, these techniques revealed capabilities in improving some properties of image reconstruction, as well (Romanowski and Grudzień, 2005). However, the practical use of that type of methodology often proves difficult because of strong computing demands. Particular tomographic methods application requires usually a ready-to-use solution, while in case of MCMC methods it is sometimes better to first suit the algorithm to the given case each time. Such an algorithm optimization may be complicated to implement at once, because of needed resource consumption.

Therefore, the authors propose the dedicated tool for MCMC imaging algorithm adjustment for ECT. The proposed software, named iMCMC, includes application of the method for solving the ECT inverse problem based on statistical methods including Monte Carlo Markov Chain and Bayes' theorem (Winkler, 2003; Liu, 2001). The presented tool can be very useful for algorithm adjusting, testing, verification and optimization because the implemented methodology is very sensitive to the proper adjustment of method parameters values. Each single small change of the individual parameter can cause substantial changes to the output results. The use of the software package for MCMC individual parameter testing can be very helpful in optimal parameters setting for different industrial applications imaging.

2. Statistical Modelling

Consider a complete dataset of N capacity measurements collected in data vector $\mathbf{C}=\{C_1, C_2, ... C_N\}$. Suppose that the area of tomograph sensor is discretized into M triangular pixels using prepared mesh. The permittivity within each pixel is constant and complete set of permittivity values is denoted by $\boldsymbol{\varepsilon}=\{\varepsilon_1, \varepsilon_2, ... \varepsilon_M\}$. The unknown is the distribution of the permittivity values given measurement records, boundary potential values and assuming some *a priori* knowledge, characteristic for the given, examined phenomenon.

2.1. Bayes' theorem

The inverse problem solution can be described as a process of recover the information of the state of a given phenomenon on the basis of the outside observations. The observations (corresponding to ECT measurement records) and the system characteristic parameters (dielectric permittivity distribution inside an ECT measurement space) can be treated as random variables. These variables can be described by a conditional probability relationship according to Bayes' theorem. Moreover, applying this theorem also allows to link the observation results with the *a priori* knowledge:

$$P(a\ posteriori) \propto P(likelihood) * P(a\ priori) \qquad (1)$$

As a result, the inverse problem solution parameters can be estimated as a measure of probability of the *a posterior* density function. Sought measures can be found as expected values computed by integrating the *a posterior* function probability distribution. Monte Carlo Markov Chain method can be used for integrating.

2.2. Monte Carlo Markov Chain

Metropolis–Hastings algorithm is used for Monte Carlo Markov Chain generation. It produces approximate samples from the posterior distribution by simulating Markov Chain with the posterior distribution as its equilibrium distribution.

Integration with the use of MCMC method computes the measure of the function f of random variables $\mathbf{X}$ vector by sampling the *posterior* distribution $\{X_t, t=1,2,...w\}$ and approximating according to formula (2):

$$\overline{f} \approx \frac{1}{w}\sum_{t=1}^{w} f(X_t) \qquad (2)$$

where:

f – function of random variables vector
X – vector of random variables
t – sample index
w – number of samples

For the purpose of integration iterations are used in the most implementations of MCMC method. The iteration process starts with some initial value, which can affect further samples. To make chain forget this value and be more convergent to stationary distribution with the growth of iteration index t, regularization criteria are applied. That is why the samples from the beginning period, called *burn-in* period, should be

discarded. Assuming that *burn-in* period lasts for *wb* iterations, formula (2) can be converted into:

$$\overline{f} \approx \frac{1}{w - wb} \sum_{t=wb+1}^{w} f(X_t) \qquad (3)$$

where:

f – function of random variables vector
X – vector of random variables
t – sample index
w – number of samples
wb – number of samples to discard

The proper choice of the starting point can increase the speed of convergence by decreasing the amount of needed iterations and shorten the length of *burn-in* period.

In practice, the length of *burn-in* period can be determined on the basis of the analysis of graphical interpretation of MCMC method output.

3. Theoretical basis of MCMC application to ECT inverse problem

3.1. Discretizing area

iMCMC application has the build-in algorithm for generating mesh for the purpose of area discretizing. The user can decide about the number of electrodes, which imposes the number of measurements, nodes between electrodes (inter-electrode space), number of rings and the number of triangles to be placed in the first the ring (elements of the outer ring). An example of generated mesh is presented in figure 1.
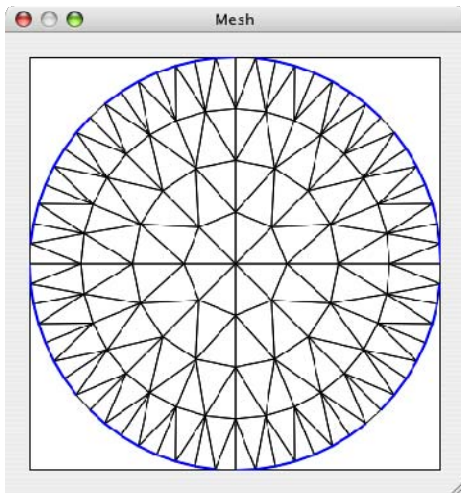

Fig. 1. An example of generated mesh.

3.2. Starting point

The starting permittivity distribution for MCMC method can be either zeros, the result of previous MCMC solution or the solution obtained by implemented Linear Back Projection (LBP) method. For this reason, iMCMC application also includes method for Jacobian computing based on mathematical analysis of electrical field inside the tomographic sensor according to the formula (4), presented in (Wajman at el., 2004) and in (Liu et al., 2001). Finite Element Method (FEM) is used for forward problem solving.

$$S_{sr}(x,y) = \oint_{\Omega(x,y)} \frac{E_s(x,y)}{\varphi_s} \frac{E_r(x,y)}{\varphi_r} dxdy \qquad (4)$$

where:

x, y – pixel indexes
s – source electrode index
r – receive electrode index
$\Omega(x, y)$ – pixel area
$E_s(x, y)$ – electrical field intensity in (x, y) pixel when potential φ_s is set on *s* electrode
$E_r(x, y)$ – electrical field intensity in (x, y) pixel when potential φ_r is set on *r* electrode

3.3. MCMC method

A standard Metropolis-Hastings algorithm is used to produce approximate samples from the posterior distribution by simulating a Markov chain which has the required distribution as its limiting distribution. Inverse problem solving with use of MCMC method is based on iteration process, in which permittivity distributions are proposed in a random way. The proposed permittivity distribution in each iteration is based on the accepted distribution in previously iteration with the random change. The change range level is defined by a variance *τ*. Parameter *τ* indicates, how much random values can affect permittivity distribution.

Metropolis-Hastings algorithm is implemented as the iterations loop. Each iteration another loop - called *sampler*, executes a number of sub-iterations. The number of sampler loop iterations depends on the user decision and in practice is determined experimentally. Whether the proposed permittivity distribution is accepted, depends on *ratio*. *Ratio* is the parameter computed as an exponent of other parameters combination. These parameters are *energy N*, *contrast S* and *difference D*. The way of computing *ratio* is presented by formula (5). Parameters with index *f* correspond to the formerly saved distribution, whereas parameters with index *p* correspond to the proposed distribution.

$$ratio = \exp\{(N_f - N_p) + (S_f - S_p) + D\} \qquad (5)$$

where:

N_f – *energy* of former distribution
N_p – *energy* of proposed distribution
S_f – *contrast* of former distribution
S_p – *contrast* of proposed distribution
D – *difference* between distributions

Computed *ratio* value is compared to the random value from range <0,1> and is accepted only, if it is greater then the proposed permittivity distribution. After last iteration of *sampler* loop, the recently accepted distribution is stored. In the end, the average of all saved permittivity distributions is calculated, which is the solution of inverse problem. Permittivity distribution values of *burn-in* period are not included.

Energy factor (*N*) characterizes how the proposed distribution corresponds to the real distribution. It is computed by the difference between measured capacitance records and calculated capacitance values

coming as the solution of forward problem of the proposed distribution. Forward problem is solved with FEM method. *Energy* is the most important parameter because it determines the convergence of the sequence.

$$N = \beta_N \frac{\left\| C_m - C_p \right\|^2}{n_s}$$
(6)

where:

N – *energy* of distribution
β_N – parameter for *energy*
C_m – measured capacities
C_p – proposed capacities
n_s – noise parameter simulating real measurements

Contrast (*S*) parameter is the difference between the value of permittivity of a pixel and the average value of permittivity of all contiguous neighbouring pixels. It corresponds with the prior knowledge about what the contrast of the real distribution should look like.

$$S = \beta_S \left\| \varepsilon^{px} - \bar{\varepsilon}^n \right\|^2$$
(7)

where:

S – *contrast* of distribution
β_s – parameter for *contrast*
ε^{px} – vector containing values of permittivity of pixels
ε^n – vector containing average values of permittivity of all contiguous neighbouring pixels

Difference (*D*) parameter matches the difference between the proposed distribution and last accepted distribution.

$$D = \beta_D \left\| \varepsilon_f - \varepsilon_p \right\|^2$$
(8)

where:

D – *difference* between distributions
β_D – parameter for *difference*
ε_f – former distribution
ε_p – propose distribution

In order to impose an influence on *ratio* value and, as a result, on acceptance rate of proposed distributions, a group of parameters is applied. These are, described above, *energy*, *contrast* and *difference*. They enable to have the control over adjustment of the impact level on *ratio*. In order to define the individual influence of each of these parameters on *ratio*, constant values should be preserved during the whole calculation loop. There is another, fourth parameter, as well, namely: *noise* n_s. This parameter is added to simulate real measurement uncertainty influenced by interference, noise and real experiment conditions. The *noise* parameter is applied only for simulated data. For real measurement data, no additional noise is added to measurement records.

Acceptance indicator is defined for the rate of proposed permittivity distributions monitoring. It is computed as a proportion between accepted and all drawn permittivity distributions:

$$Acceptance = \frac{n_{accepted}}{n_{all}}$$
(9)

where:

$n_{accepted}$ – number of accepted samples
n_{all} – number of all samples

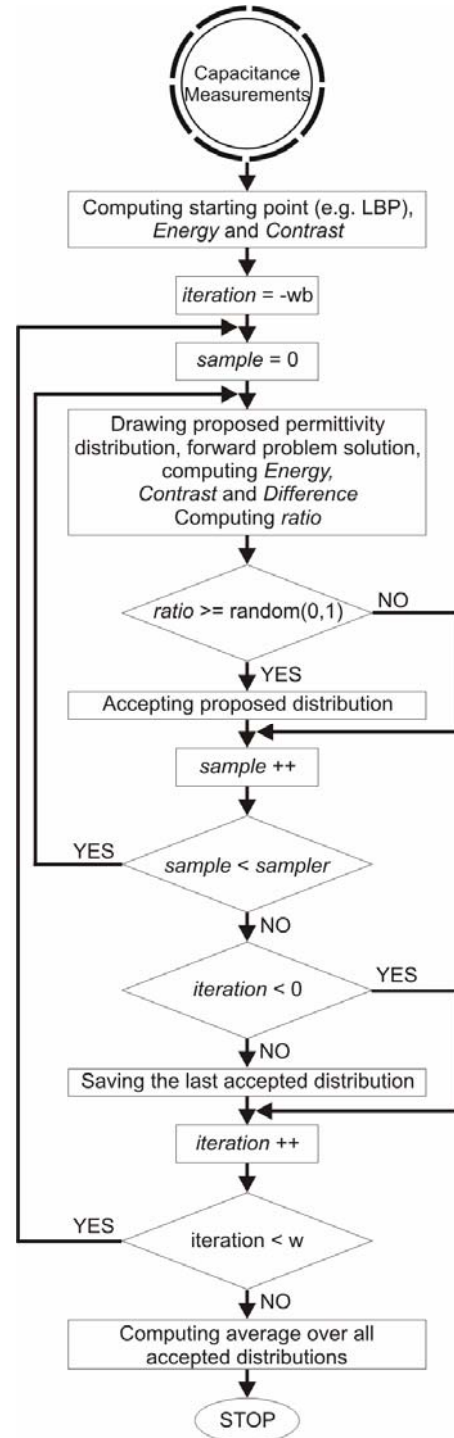The algorithm implemented in the application is presented in figure 2.

Fig. 2. Implemented algorithm of MCMC method.

4. iMCMC Application Implementation

4.1. Platform & Development

For the purpose of design and tests Matlab R13 environment was used. Next the Mac architecture was

chosen for iMCMC application development. The choice was made because the available, at this platform, powerful, user-friendly and easy-programmable graphical user interface, simple memory management and efficient vector operations processing.

Implementation was completed on PowerBook G4 station equipped with 867 MHz PowerPC G4 processor and 768 MB of RAM. However the iMCMC application can be run on Intel-based Apple computers, as well. Mac OS X 10.3.9 operating system was installed and XCode 1.5 was used as a software for Mac OS application developing, This is the latest version of XCode that can be run on PowerPC G4 family processors. The choice of the particular system version ensures backward compatibility. Therefore, also older hardware and software of educational sites, such as academia labs can be utilized for program exploration.

The implementation consists of ten classes written in ObjectiveC language and using the Cocoa application environment. Application was designed according to Model-View-Controller (MVC) architectural pattern but it also uses other mechanisms, such as *delegation*. The huge advantage of MVC pattern is ease of further development, scalability.

4.2. Functionality

The important advantage, in contrast to many powerful research helpful programs, of the presented software package is the full graphical user interface. GUI implementation ensures that the tool is easy to use and the investigation of MCMC method becomes more user-friendly.

- The main function of iMCMC application is solving the inverse problem using the implemented MCMC method. However, in order to complete the task, a number of sub-modules were realized.

Typical use of the iMCMC application starts with mesh parameters input and mesh generation. Having the calculated mesh, it is then possible to calculate Jacobian matrix. Next step is the input of the data; user can upload measured capacitance values, from files or by manual input of data. Measurements in text files should be saved in float numerical format, separated by spaces or new line characters. After that, it is possible to calculate the solution using Linear Back Projection method. The obtained LBP solution is the starting point for MCMC method. User can define each algorithm parameter before the start of iteration loop. When the program is running, it is not allowed to change the parameters until the end of current iteration. During the entire chain run process, user can observe changes of parameters and accepted distributions at the dedicated monitoring window. When the last iteration ends the final solution is computed and presented. It can be then used as a new starting point or user can reset starting point to the solution gained with use of LBP method.

4.3. User interface

User interface of the iMCMC application was developed with Aqua GUI for Mac OS X. Aqua framework development is restricted with certain rules

and contains guidelines for designing and creating graphical user interface which programmer should obey. The whole interface was prepared in English language to make the iMCMC application more accessible.

Interface consists of number of windows. Distinct windows are not modal and each of them can be placed separately anywhere in view space. Such approach allows to display each window of the iMCMC application at the same time, including solutions windows, which gives the user ability for instant comparison.

The most important part of the user interface is the main window of application, which is presented in figure 3. The main window is divided into three parts:

- **Mesh** – associated with mesh generation, display and modification, as well, as Jacobian calculation. The mesh is generated according to the user-defined parameters.
- **Data** – This part concerns the measurement data. User, by means of *Edit measurements* button, gains access to data input window, where either manual or file upload measurement data input is possible. User is responsible for defining the minimum and maximum permittivity values, as well, as for corresponding potential distribution calculation (these are necessary for LBP method employment).
- **Solution** – corresponding to inverse problem solution. *LBP solution* button forces the iMCMC application to compute the inverse problem solution using LBP method. Solution can be displayed with *Show LBP* button. *Show MCMC* opens the window presenting MCMC solution, while *MCMC Solution* button opens the MCMC Solution window covering the control options for MCMC algorithm.

Closing the main window terminates the iMCMC application.

Fig. 3. The iMCMC application main window.

The MCMC Solution window is presented in figure 4. The upper part of this window contains:

- *START/STOP*, *PAUSE* buttons,
- status box informing about current state, e.g. STOPPED, WORKING;
- *Update view* checkbox and boxes informing about process parameters values such as iteration of *sampler*;

- number of *burn-in* period iteration;
- number of iteration after *burn-in* period;
- *ratio*, *energy*, *contrast*, *difference*, and rate of *acceptance* parameters.

In order to observe progress of MCMC procedure, MCMC solution window can be invoked (*Show MCMC*). However drawing *on-line* all the computing stages can slow down the calculations, especially on slower computers. It is possible to prevent this drawback, disabling *Update view* checkbox on the MCMC Solution window.

The lower part of the MCMC Solution window contains three tabs. Firs tab, *parameters* allows to change values of algorithm parameters. Second tab allows to control iterations count, and the last tab gives control over automated τ value selection during *burn-in* period.
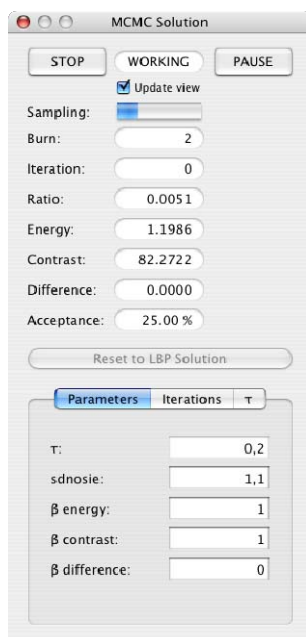
Fig. 4. The MCMC Solution window.

4.4. Numerical experiments

This section presents examples of iMCMC performance. In order to illustrate application capabilities, well-chosen simulated permittivity distributions, the so-called phantoms, were examined.

The example procedure of application performance is the following. Any calculations begin from computing a forward problem solution for a defined phantom. For that purpose, dedicated Matlab m-files are used. Simulations are performed for eight-electrode tomography sensor. Obtained solution serves as simulated measured capacitance values and can be inserted as an iMCMC application input. Then, inverse problem solution for different parameters values and number of iterations can be found. The next subsections present results for two types of phantom distributions (half-, and centric- occupation of measurement space). Results are presented using coarse mesh in order to attain better representation for the results, especially in printed version of the paper.

4.5. Half filled sensor

First simulated experiment assumed the tomography sensor was filled in a half with the material of higher permittivity value than the other, empty part. The phantom shape is presented in figure 5.
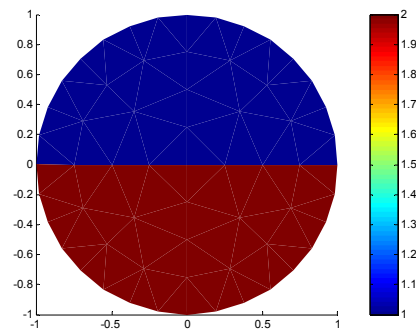

Fig. 5. Half filled sensor.

First attempt of the inverse problem solving using MCMC method is presented in figure 6. Parameter values are the following: *burn-in* period 5 iterations with *sampler* of 10 samples each iteration; τ value of 0.02; *noise* 1.1, *energy* factor 1; while *contrast* and *difference* factors exactly zeros. It can be concluded as the solution (100 iter.) looks similar to the phantom.
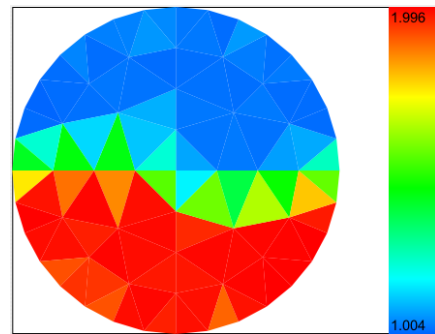

Fig. 6. MCMC method solution for half filled sensor

Another solution example presented in figure 7 was obtained for a different set of parameter value. *Sampler* was changed to 20 samples, *burn-in* to 10 iterations and value of τ was 0.4. Conclusion is the solution, is far from the original distribution, which confirms high sensitivity of MCMC algorithms with respect to parameter values variations.
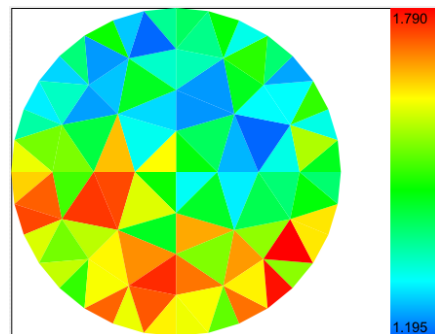

Fig. 7. MCMC method solution for half filled sensor

4.6. Sensor filled in centre

Another simulated experiment was conducted for a centric-located phantom of higher (in contrast to the rest of empty space) permittivity shown in figure 8.
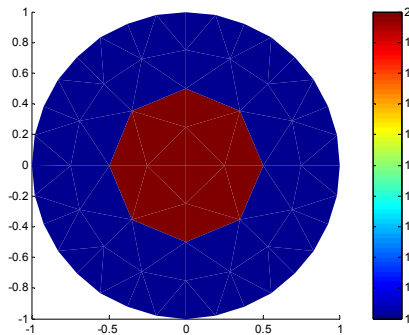

Fig. 8. Sensor filled in centre.

Calculations were performed for *sampler* 5, *burn-in* 5 iter., *sdnoise* 1.1, τ 0.01, β_{energy} 100, $\beta_{contrast}$ 0.0001 and $\beta_{difference}$ 0.

Solutions for 400 and 1000 iterations are presented in figures 9 and 10 respectively. Experiments show that besides appropriate adjustment of method parameters, number of iterations is crucial to the quality of solution. Reconstructed image quality increases with raising number of iterations.
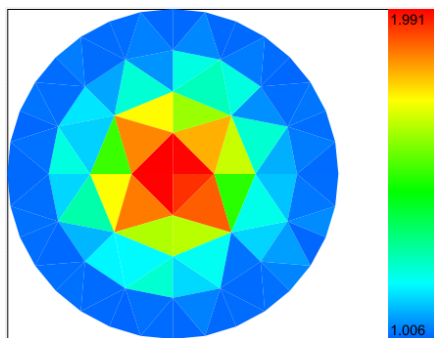

Fig. 9. MCMC method solution for the sensor filled in the centre after 400 iterations.
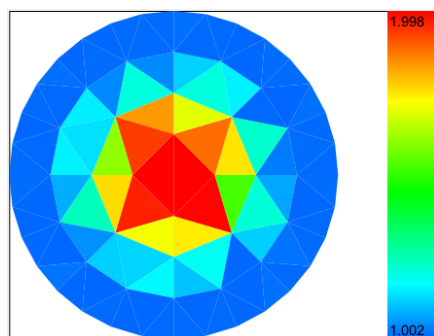

Fig. 10. MCMC method solution for the sensor filled in the centre after 1000 iterations.

5. Conclusions

Described software package presents an interesting tool for both, classical tomography imaging exploration and MCMC methods adjustment. The main advantage of this program is its applicability to educational purposes and tomographic data processing methods examination.

The dedicated tool proposed for investigating Monte Carlo Markov Chain imaging algorithm adjustment for ECT proves its usefulness for the appropriate method parameters values determination. The main advantage of the iMCMC application is giving the user an ability to change each individual parameter during calculations. This feature is very helpful in optimal parameters setting for different industrial applications. The application is under ongoing development in order to enhance its functions and interface.

Presented numerical experiments carried out with use of the iMCMC application showed sensitivity of the implemented methodology to the algorithm parameters. As it can be expected, parameters values fixed as appropriate for a chosen process do not have to automatically fit another application. Another confirmed anticipation is the quality of image reconstruction is highly dependent on the number of iterations. However, without adequate parameters adjustment, the solution may be unsatisfactory even after numerous iterations.

REFERENCE

KAIPIO J.P., KOLEHMAINEN V, SOMERSALO E, VAUHKONEN M., (2000), Statistical Inversion and Monte Carlo Sampling Methods in Electrical Impedance Tomography, *Inverse Problems*, Vol. 16, pp. 1487-1522.

LIONHEART W.R.B.., (2004), Review: Developments in EIT reconstruction algorithms: pitfalls, challenges and recent developments, *Physiological Measurements*, Vol. 25, pp. 125-142.

LIU JS., (2001), *Monte Carlo Strategies in Scientific Computing*. Berlin, Heidelberg: Springer-Verlag, p.364.

LIU S., YANG W.Q., WANG H., JIANG F., SU Y., (2004), Investigation of square fluidized beds using capacitance tomography: preliminary results, *Measurement Science and Technology*, Vol. 12, pp. 1120-1125.

ROMANOWSKI A., GRUDZIEŃ K., (2005), The juxtaposition of image reconstruction methods and 2D spatial modelling for process tomography, (in polish), *Automatyka/ AGH* Cracov, Vol. 9 number 3, pp. 621-630.

ROMANOWSKI A., GRUDZIEN K., AYKROYD R., WILLIAMS R., WEST R., MENG S., (2004), An Introduction to the Bayesian/MCMC Approach as a Recently Developed Methodology for Tomographic Data Analysis, 3rd International Tomography Symposium in Poland, pp. 134-138.

WAJMAN R., MAZURKIEWICZ Ł., SANKOWSKI D., (2004), The sensitivity map in the image reconstruction process for electrical capacitance tomography (in polish), *Automatyka / AGH* Cracov, Vol. 8, nr 3, pp. 415-420.

WEST R.M., AYKROYD R.G., MENG S., WILLIAMS R.A., (2004), Markov Chain Monte Carlo Techniques and Spatial-Temporal Modelling for Medical EIT, *Physiological Measurements*, Vol. 25, pp. 181-194.

WINKLER G., (2003), *Image Analysis, Random Fields and Markov Chain Monte Carlo: A Mathematical Introduction* (2nd Ed.). Berlin, Heidelberg: Springer-Verlag, p. 412.

YANG W.Q., SPINK D.M., YORK T.A., MCCANN H., (1999), An image-reconstruction algorithm based on Landweber's iteration method for electrical-capacitance tomography, *Measurement Science and Technology*, Vol. 10, pp. 1065–1069.