



- (51) International Patent Classification:  
*G06F 17/50* (2006.01)
- (21) International Application Number:  
PCT/US2013/045386
- (22) International Filing Date:  
12 June 2013 (12.06.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
61/660,012 15 June 2012 (15.06.2012) US  
13/843,096 15 March 2013 (15.03.2013) US
- (71) Applicant: CHEVRON U.S.A. INC. [US/US]; 6001 Bollinger Canyon Road, San Ramon, California 94583 (US).
- (72) Inventors: SARMA, Pallav; 6001 Bollinger Canyon Road, San Ramon, California 94583 (US). CHEN, Wen Hsiung; 6001 Bollinger Canyon Road, San Ramon, California 94583 (US).
- (74) Agents: VANDER PLOEG, Craig R. et al.; Chevron Corporation, Law Department, Post Office Box 6006, San Ramon, California 94583-0806 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

**Published:**

- without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: SYSTEM AND METHOD FOR USE IN SIMULATING A SUBTERRANEAN RESERVOIR

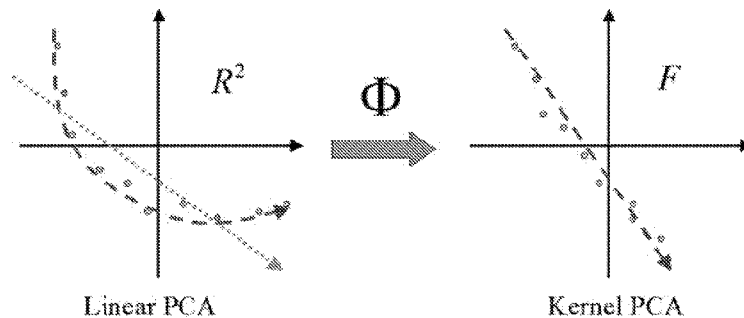
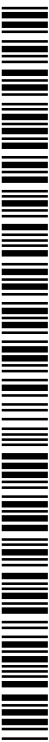


FIG. 1

(57) Abstract: A computer-implemented method, system, and computer program product are disclosed for updating simulation models of a subterranean reservoir. An ensemble of reservoir models representing a subterranean reservoir having non-Gaussian characteristics is provided and the ensemble of reservoir models is updated using a subspace ensemble Kalman filter. Kernel principle component analysis parameterization or K-L expansion parameterization can be used to update the ensemble of reservoir models.



## SYSTEM AND METHOD FOR USE IN SIMULATING A SUBTERRANEAN RESERVOIR

### TECHNICAL FIELD

[0001] This invention relates to simulation of a subterranean reservoir, and more particularly, a system, method, and computer program product for updating simulation models of subterranean reservoir such as by utilizing subspace ensemble Kalman filters.

### BACKGROUND

[0002] The ensemble Kalman filter (EnKF) has recently received significant attention as a robust and efficient tool for data assimilation. Although the EnKF has many advantages such as ease of implementation and efficient uncertainty quantification, it suffers from a few key issues that limit the application of the EnKF to large-scale simulation models of real fields. Among the various issues, one of the important ones is the well known problem of ensemble collapse, which is particularly evident for small ensembles. The problem arises because all the ensemble members are driven in more or less the same direction through each ensemble update. This implies that as the members are updated again and again, they will ultimately converge toward the same ensemble member. This results in an artificial reduction of variability across the ensemble, resulting in a poor quantification of uncertainty. The second, more important problem is that the EnKF is theoretically appropriate only if all ensemble members belong to the same multi-Gaussian random field (geological/geostatistical model). This is because the updated ensemble members are a linear combination of the forecasted ensemble members. If each ensemble member has different geostatistical properties, the updated ensemble members will not be able to honor these geostatistical properties. Ultimately, all the updated ensemble members will have similar geostatistical properties. This is an important problem because there is more than one geological scenario for most real fields, and it is desired to obtain one or more history-matched models for each geological scenario. One way to resolve this problem is to apply a different EnKF for each geological scenario. This however, is not very efficient as this increases the computational burden  $N$  times over the standard EnKF for  $N$  geological scenarios. For example, if each EnKF has 100 ensemble members (a typical number), and there are 10 geological scenarios, this approach would require 1000 simulations, which may not be practical for large-scale models.

### SUMMARY

[0003] In embodiments, a computer-implemented method, system, and computer program product are disclosed for updating simulation models of a subterranean reservoir. An ensemble of reservoir models representing a subterranean reservoir having non-Gaussian characteristics is provided and the ensemble of reservoir models is updated using a subspace ensemble Kalman filter.

[0004] In embodiments, kernel principle component analysis parameterization or K-L expansion parameterization can be used to update the ensemble of reservoir models.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Figures 1A and 1B are schematics comparing linear PCA with kernel PCA.

[0006] Figure 2 shows a synthetic 2D 2-phase model shown with true log permeability field, 4 producers (circles) and 1 injector (diamond).

[0007] Figures 3A-3J show 10 initial ensemble members (log permeability) of the standard EnKF, each having a different correlation length and different anisotropy direction.

[0008] Figure 4 shows variance of log permeability of an initial ensemble.

[0009] Figures 5A-5D show oil and water production rates of an initial ensemble for 4 producers, and the observed rates (circles for oil and X's for water).

[0010] Figures 6A-6J show 10 ensemble members of the final ensemble of the standard EnKF after 10 updates.

[0011] Figure 7 shows variance of log permeability of the final ensemble of the standard EnKF.

[0012] Figures 8A-8D show oil and water production rates of a final ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the standard EnKF.

[0013] Figures 9A-9J show 10 ensemble members of the final ensemble of the subspace EnKF after 10 updates.

[0014] Figure 10 shows variance of log permeability of the final ensemble of the subspace EnKF.

[0015] Figures 11A-11D show oil and water production rates of the final ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the subspace EnKF.

[0016] Figure 12 shows the percentage of time the true production rates are within the predicted rates for 20 different random well placements.

[0017] Figure 13A-13D show oil and water production rates predicted for the 4 producers for a random well placement, and the observed rates (circles for oil and X's for water) for the standard EnKF.

[0018] Figure 14A-14D show oil and water production rates predicted for the 4 producers for a random well placement, and the observed rates (circles for oil and X's for water) for the subspace EnKF.

[0019] Figure 15 shows a synthetic 2D 2-phase model shown with true permeability field (channels), 4 producers (black circles) and 1 injector (black diamond)

[0020] Figure 16 shows a channel training image used to create the original realizations.

[0021] Figure 17 shows initial ensemble members (permeability) for the iterative EnS with PCA parameterization.

[0022] Figure 18 shows variance of permeability of the initial ensemble of the iterative EnS with PCA parameterization.

[0023] Figure 19 shows oil and water production rates of the initial ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the iterative EnS with PCA parameterization.

[0024] Figure 20 shows final ensemble members (permeability) for the iterative EnS with PCA parameterization.

[0025] Figure 21 shows variance of permeability of the final ensemble of the iterative EnS with PCA parameterization.

[0026] Figure 22 shows oil and water production rates of the final ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the iterative EnS with PCA parameterization.

[0027] Figure 23 shows initial ensemble members (permeability) for the iterative EnS with KPCA (order 5) parameterization.

[0028] Figure 24 shows variance of permeability of the initial ensemble of the iterative EnS with KPCA (order 5) parameterization.

[0029] Figure 25 shows oil and water production rates of the initial ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the iterative EnS with KPCA (order 5) parameterization.

[0030] Figure 26 shows final ensemble members (permeability) for the iterative EnS with KPCA (order 5) parameterization.

[0031] Figure 27 shows variance of permeability of the final ensemble of the iterative EnS with KPCA (order 5) parameterization.

[0032] Figure 28 shows oil and water production rates of the final ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the iterative EnS with KPCA (order 5) parameterization.

[0033] Figure 29 shows initial ensemble members (permeability) for the subspace iterative EnS with KPCA (order 5) parameterization.

[0034] Figure 30 shows variance of permeability of the initial ensemble of the subspace iterative EnS with KPCA (order 5) parameterization.

[0035] Figure 31 shows oil and water production rates of the initial ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the subspace iterative EnS with KPCA (order 5) parameterization.

[0036] Figure 32 shows final ensemble members (permeability) for the subspace iterative EnS with KPCA (order 5) parameterization.

[0037] Figure 33 shows variance of permeability of the final ensemble of the subspace iterative EnS with KPCA (order 5) parameterization.

[0038] Figure 34 shows oil and water production rates of the final ensemble for the 4 producers, and the observed rates (circles for oil and X's for water) for the subspace iterative EnS with KPCA (order 5) parameterization.

[0039] Figure 35 shows a system for updating simulation models of a subterranean reservoir using subspace ensemble Kalman filters.

#### DETAILED DESCRIPTION

[0040] Embodiments of the present invention relate to a variant of the Ensemble Kalman Filter (EnKF) method, which is referred to herein as the subspace EnKF. The subspace EnKF constrains each ensemble member to a small subspace of the full space to which the ensemble members belong. This is done by applying a different parameterization for each ensemble member, such that the parameterization ensures that each ensemble member remains in the subspace to which it is constrained through any number of updates. If the subspaces are disjoint, this ensures that the ensemble does not collapse. Further, if each parameterization is chosen to honor different geostatistical properties, then these statistics will also be honored throughout the updates. Accordingly, the geostatistical properties of each ensemble member are retained through each update. However, this approach implies that each ensemble member has a different forward model, which violates an important assumption of the EnKF, whereby each ensemble member should have the same forward model. Through proper modification of the EnKF equations using the gradients of the different parameterizations, this inconsistency can be mitigated. As will be

described, the subspace EnKF prevents ensemble collapse and honors geological constraints much better compared to the standard EnKF.

### The Ensemble Kalman Filter

[0041] The EnKF is a temporally sequential data assimilation method. At each assimilation time, the following steps are performed: a forecast step (evaluation of the dynamic system), followed by a data assimilation step (calculation of Kalman gain), and then by an update of the state variables of the EnKF (Kalman update).

[0042] The state variables of the EnKF typically consist of the following types of variables: static variables (such as permeability, porosity), dynamic variables (such as pressure, saturation), and production data (such as bottom hole pressures, production and injection rates). Thus, the state vector of the EnKF at a given assimilation time can be written as:

$$\mathbf{y}_j = \begin{bmatrix} \mathbf{m} \\ \mathbf{x} \\ \mathbf{d} \end{bmatrix}_j \quad \forall j = 1, \dots, M \quad (1)$$

[0043] Here,  $\mathbf{y}_j \in R^{N_s}$  is the  $j$ th ensemble member,  $\mathbf{m}$  are the static variables,  $\mathbf{x}$  are the dynamic variables, and  $\mathbf{d} \in R^{N_d}$  are the production data to be assimilated. The number of ensemble members is  $M$ . Before the EnKF can be applied, an initial ensemble of the static and dynamic variables is created. Usually, geostatistical techniques are used to create the ensemble of static variables (e.g., permeability field) corresponding to prior knowledge of the geology and hard data. However, the dynamic variables are usually considered known without uncertainty, primarily because the uncertainty of the dynamic variables such as pressure and saturation at the initial time is smaller compared to that of the static variables, as the reservoir is generally in dynamic equilibrium before start of production. In any case, if the initial dynamic variables are considered uncertain, it can be easily reflected through the initial ensemble. There is usually no production data available initially.

[0044] Once the initial ensemble is created, the forecast step can be performed, wherein the dynamic system (i.e., the reservoir simulation model) is evaluated to the next assimilation time

using the current estimates of the static and dynamic variables. The simulator has to be run once for each ensemble member ( $M$  times). This provides the forecasted dynamic variables and production data at the next assimilation step, and the step can be written compactly as:

$$\mathbf{y}_{j,k}^f = f(\mathbf{y}_{j,k-1}^u) \quad \forall j = 1, \dots, M \quad (2)$$

[0045] Here  $\mathbf{y}^f$  is the forecasted state vector obtained after the forecast step,  $\mathbf{y}^u$  is the updated state vector obtained after the Kalman update, and subscript  $k$  stands for the assimilation time.  $f$  depicts the reservoir simulation equations. In most cases, the assimilation time subscript  $k$  is removed from discussions below for brevity.

[0046] After the forecasted state vector is obtained, the data assimilation step is performed, wherein, the Kalman gain is calculated using the static and dynamic variables and production data obtained from the forecast step, and is given as:

$$\mathbf{K}_g = \mathbf{C}_y^f \mathbf{H}^T (\mathbf{H} \mathbf{C}_y^f \mathbf{H}^T + \mathbf{C}_e)^{-1} \quad (3)$$

[0047] Here,  $\mathbf{K}_g$  is known as the Kalman gain matrix, and is of size  $N_s \times N_d$ , and  $\mathbf{C}_y^f$  is the covariance matrix of  $\mathbf{y}^f$ . Matrix  $\mathbf{H} = [\mathbf{0} | \mathbf{I}]$ , where  $\mathbf{0}$  is a  $N_d \times (N_s - N_d)$  matrix of zeros, and  $\mathbf{I}$  is a  $N_d \times N_d$  identity matrix.  $\mathbf{C}_e$  is the error covariance matrix of the production data  $\mathbf{d}$ , which is usually assumed to be diagonal. Accordingly, data measurement errors are independent of each other. From the above, it is clear that  $\mathbf{C}_y^f \mathbf{H}^T$  is the covariance matrix of the full forecasted state vector  $\mathbf{y}^f$  with the forecasted production data  $\mathbf{d}$  and  $\mathbf{H} \mathbf{C}_y^f \mathbf{H}^T$  is the covariance matrix of forecasted production data  $\mathbf{d}$  with itself. Defining the centered vectors,  $\hat{\mathbf{y}}^f = \mathbf{y}^f - \bar{\mathbf{y}}^f$  and  $\hat{\mathbf{d}} = \mathbf{d} - \bar{\mathbf{d}}$ , the following is obtained:

$$\mathbf{C}_y^f \mathbf{H}^T = \frac{1}{M} \sum_{j=1}^M \hat{\mathbf{y}}_j^f \hat{\mathbf{d}}_j^T \quad (4)$$

$$\mathbf{H} \mathbf{C}_y^f \mathbf{H}^T = \frac{1}{M} \sum_{j=1}^M \hat{\mathbf{d}}_j \hat{\mathbf{d}}_j^T \quad (5)$$

[0048] The final step of the EnKF is to update the state variables using the actual observations, which is given as:

$$\mathbf{y}_j^u = \mathbf{y}_j^f + \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \quad (6)$$

[0049] Here,  $\mathbf{d}_o$  are the observed production data, and random perturbations corresponding to  $\mathbf{C}_e$  are added to  $\mathbf{d}_o$  to create the ensemble of observed data,  $\mathbf{d}_{o,j}$ . The updated state vector  $\mathbf{y}^u$  thus calculated is then used to evaluate the next forecast step, and the process repeated for the next assimilation time. The EnKF methodology can be summarized as follows:

$$\mathbf{m}_j^f \xrightarrow{f(\mathbf{m}_j)} \begin{bmatrix} \mathbf{x}_j \\ \mathbf{d}_j \end{bmatrix}^f \rightarrow \mathbf{K}_g \rightarrow \mathbf{y}_j^u = \mathbf{y}_j^f + \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \rightarrow \mathbf{m}_j^u \quad \forall j = 1, \dots, M \quad (7)$$

**EnKF with Parameterization**

[0050] Parameterization of the static and dynamic variables can be used with the EnKF, for example, to increase in efficiency and robustness, handle non-Gaussian random fields, and mitigate issues with non-physical updates of dynamic and static variables. Parameterization implies the creation of a functional relationship between the concerned variables (random field) and a smaller set of preferably independent random variables. In some embodiments, a differentiable mathematical parameterization of the form  $\mathbf{y} = g(\boldsymbol{\xi})$  is obtained. Here, vector  $\mathbf{y}$  is a realization of a discrete random field,  $\mathbf{y} \in R^{N_s}$  ( $N_s$  = EnKF state vector dimension), and vector  $\boldsymbol{\xi}$  (input parameters of dimension much smaller than  $N_s$ ) is a set of independent random variables with a specified distribution (e.g., standard Gaussian). Extending the EnKF to handle parameterized static and/or dynamic variables is well known by one skilled in the art. Equation (8) shows the extension of the EnKF with parameterization only of the static variables.

$$\boldsymbol{\xi}_j^f \xrightarrow{g(\boldsymbol{\xi}_j)} \mathbf{m}_j^f \xrightarrow{f(\mathbf{m}_j)} \begin{bmatrix} \mathbf{x}_j \\ \mathbf{d}_j \end{bmatrix}^f \rightarrow \mathbf{K}_g \rightarrow \mathbf{y}_j^u = \mathbf{y}_j^f + \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \rightarrow \boldsymbol{\xi}_j^u \quad \forall j = 1, \dots, M \quad (8)$$

[0051] Here, the state vector of the parameterized EnKF is  $[\xi, \mathbf{x}, \mathbf{d}]^T$ . Instead of directly updating the static variables  $\mathbf{m}$  with the EnKF, the variables  $\xi$  are updated and the updated variables  $\xi$  are used to obtain the updated  $\mathbf{m}$ . The conventional EnKF steps can then be followed. In the case when such a parameterization is used for honoring non-Gaussian random fields, the parameterization ensures that the updated static variables (e.g., permeability field) honor the parameterization and therefore, the statistical properties of the random field used to create the parameterization.

[0052] Additional details and examples of EnKF and EnKF derivatives (e.g., KEnKF), can be found in U.S. Patent No. 7,584,081, U.S. Patent Application Nos. 12/657,929 and 13/029534, *Reservoir Monitoring and Continuous Model Updating Using Ensemble Kalman Filter* (Naevdal, G., Johnsen, L.M., Aanonsen, S.I., Vefring, E.H., SPE paper 84372 presented at the SPE Annual Technical Conference and Exhibition, Denver, CO, 2003), and *Real-time Reservoir Model Updating Using Ensemble Kalman Filter* (Wen, X.-H., Chen, W.H., SPE 92991 presented at the SPE Reservoir Simulation Symposium, Houston, TX, 2005).

### Subspace EnKF Formulation

[0053] The subspace EnKF generalizes the parameterized EnKF by allowing the parameterization of each ensemble member to be different from each other. That is, for each ensemble member:

$$\mathbf{m}_j = g^j(\xi_j) \quad \forall j = 1, \dots, M \quad (9)$$

[0054] Thus, each ensemble member can belong to different random fields, or can be constrained to different subspaces of the same random field depending on the nature of the parameterization. The different parameterizations ensure that each updated ensemble member honors the statistics of the random field from which it was generated, or constrains the updated variables to lie within the specific subspaces.

$$\xi_j^f \xrightarrow{g^j(\xi_j)} \mathbf{m}_j^f \xrightarrow{f(\mathbf{m}_j)} \begin{bmatrix} \mathbf{x}_j \\ \mathbf{d}_j \end{bmatrix}^f \rightarrow \mathbf{K}_g \rightarrow \mathbf{y}_j^u = \mathbf{y}_j^f + \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \rightarrow \xi_j^u \quad \forall j = 1, \dots, M \quad (10)$$

[0055] The EnKF equations cannot be applied as is if the parameterization is different for different ensemble members. This is because the EnKF equations are only valid if the forward model relating the static variables ( $\xi$ ) to the dynamic ( $x, d$ ) for each member is the same. In order to mitigate this problem, it is realized that the Kalman update equation is similar to the steepest descent equation with the step length set to 1. Thus, as is well known, the Kalman update is an approximation to the gradient of the least-square error with respect to the state variables of the EnKF:

$$\frac{dE}{dy_j} \approx \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \quad \text{or} \quad \frac{dE}{d\mathbf{m}_j} \approx \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \Big|_{\mathbb{R}^{N_m}} \quad (11)$$

[0056] Once  $\mathbf{m}$  is obtained from  $g(\xi)$ , the exact parameterization is irrelevant for the calculation of  $dE/d\mathbf{m}$ , as the calculation of  $dE/d\mathbf{m}$  does not involve  $\xi$  or  $g(\xi)$ . Therefore,  $dE/d\mathbf{m}$  can be evaluated correctly even if different parameterizations are used for the different ensemble members; however, the same cannot be said for  $dE/d\xi$ . Thus, in order to obtain,  $dE/d\xi$ , the ubiquitous chain rule can be used.

$$\frac{dE}{d\xi_j} = \frac{d\mathbf{m}_j}{d\xi_j} \frac{dE}{d\mathbf{m}_j} = \frac{dg^j(\xi)}{d\xi_j} \frac{dE}{d\mathbf{m}_j} = \frac{dg^j(\xi)}{d\xi_j} \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \Big|_{\mathbb{R}^{N_m}} \quad (12)$$

[0057] Equation (12) can be used to obtain the updated  $\xi$ . Notice that although  $dE/d\mathbf{m}$  lies in the same subspace for each ensemble member (i.e., subspace spanned by the columns of  $\mathbf{K}_g$ )  $dE/d\xi$  can be quite different and belong to different subspaces due to the different parameterizations.

$$\xi_j^u = \xi_j^f + \frac{dg^j(\xi)}{d\xi_j^f} \mathbf{K}_g (\mathbf{d}_{o,j} - \mathbf{d}_j) \Big|_{\mathbb{R}^{N_m}} \quad \forall j = 1, \dots, M \quad (13)$$

[0058] Once  $\xi_j^u$  is obtained,  $\mathbf{m}_j^u$  can be obtained using equation (9). Therefore the final Kalman update equation for  $\mathbf{m}$  can be written as follows:

$$\mathbf{m}_j^u = g^j \left( \xi_j^f + \frac{dg^j(\xi)}{d\xi_j^f} \mathbf{K}_g(\mathbf{d}_{o,j} - \mathbf{d}_j) \Big|_{1:N_m} \right) \quad \forall j = 1, \dots, M \quad (14)$$

[0059] The Kalman update equation for the dynamic variables remains the same as the standard EnKF if no parameterization is used for them.

### Subspace EnKF with K-L Expansion

[0060] The subspace EnKF for the well known Karhunen-Loeve expansion can be derived as the parameterization function  $g(\xi)$ . The Karhunen-Loeve expansion is a powerful tool for representing stationary and non-stationary random fields (or processes) with explicitly known covariance functions. It allows representing any random field or process as a series expansion involving a complete set of deterministic functions with corresponding random coefficients, which can be used as a differentiable parameterization of the random field. This method provides a second-moment characterization (2-point statistics) in terms of random variables and deterministic functions. The use of the K-L expansion with orthogonal deterministic basis functions and uncorrelated random coefficients has generated interest because of its bi-orthogonality property. Accordingly, both the deterministic basis functions (eigenfunctions) and the corresponding random coefficients are orthogonal. This allows for an optimal encapsulation of the information contained in a multi-Gaussian random process into a set of independent random variables. The discrete K-L expansion for a given covariance matrix  $\mathbf{C}$  is given as:

$$\mathbf{y} = \mathbf{E}\mathbf{\Lambda}^{1/2}\xi = \Phi\xi \quad \equiv \quad \mathbf{y} = g(\xi) \quad (15)$$

[0061] Here,  $\mathbf{E}$  is the matrix of eigenvectors of the covariance matrix  $\mathbf{C}$ ,  $\mathbf{\Lambda}$  is a diagonal matrix of the eigenvalues of  $\mathbf{C}$ , and  $\xi$  (vector) is a set of uncorrelated random variables. These random variables are also independent if the random field is multi-Gaussian. Therefore, if the elements of  $\xi$  are drawn from the standard Gaussian distribution, the  $\mathbf{y}$  obtained using Equation (15) will be multi-Gaussian correlated with covariance  $\mathbf{C}$ . Thus, the K-L expansion can be used for simulation of multi-Gaussian random fields. On the other hand, if the covariance matrix  $\mathbf{C}$  is obtained from a set of realizations having higher order statistics, Equation (15) can still be used

to obtain realizations  $\mathbf{y}$ , but those realizations will only have the same covariance  $\mathbf{C}$  as the original realizations used to calculate  $\mathbf{C}$ . The higher order moments in the original realizations will in general not be reproduced.

[0062] The K-L expansion is thus a parameterization of the form  $\mathbf{y} = g(\boldsymbol{\xi})$  where the functional relationship is linear. Here, matrix  $\mathbf{C}$  is of size  $N_s \times N_s$ . The maximum size of the matrices  $\mathbf{E}$  and  $\boldsymbol{\Lambda}$  is  $N_s \times N_s$ , and that of vector  $\boldsymbol{\xi}$  is  $N_s \times 1$ . Note that the word maximum is used because we may choose to retain only the largest  $N_K$  of the total  $N_s$  eigenvalues, in which case  $\mathbf{E}$  is of size  $N_s \times N_K$ ,  $\boldsymbol{\Lambda}$  is of size  $N_K \times N_K$ , and  $\boldsymbol{\xi}$  is of size  $N_K \times 1$ . Thus, the random field  $\mathbf{y}$  is parameterized in terms of  $N_K$  independent random variables  $\boldsymbol{\xi}$ .

[0063] If the K-L expansion of equation (15) is used as the parameterization function, then equation (14) is reduced to the following:

$$\mathbf{m}_j^u = \mathbf{m}_j^f + \Phi_j \Phi_j^T \mathbf{K}_g(\mathbf{d}_{o,j} - \mathbf{d}_j) \Big|_{1:N_m} \quad \forall j = 1, \dots, M \quad (16)$$

[0064] If  $\Phi$  is full rank, then  $\Phi\Phi^T$  is the identity matrix, which reverts back to the standard Kalman update equation. However,  $\Phi$  is usually obtained from covariance matrices obtained from a small number of realizations of the random field, and thus has a much smaller rank. Furthermore, if each ensemble member has a different low rank  $\Phi$ , then each updated ensemble member is constrained to a small subspace of the full space of  $\mathbf{m}$ .

### Subspace EnKF with Kernel PCA Parameterization

[0065] As described above, standard principal component analysis (PCA) or the K-L expansion is appropriate for multi-Gaussian random fields. However, most real geological systems (e.g., channels, deltaic fans) are usually non-Gaussian, limiting the applicability of the K-L expansion. To mitigate this problem, kernel methods can be used to create nonlinear, non-Gaussian generalizations of PCA, which can then be used to parameterize non-Gaussian random fields. In particular, polynomial kernels of higher orders can be applied to perform the K-L expansion not in the original space of the realizations  $R^{N_c}$ , but in a high order space called feature space  $F$ . The introduction of the feature space acts to preserve higher order moments (i.e., multipoint

statistics) rather than just the second moment (i.e., two-point statistics). Since the feature space  $F$  is nonlinearly related to the original space  $R^{N_c}$ , the overall approach is essentially a form of nonlinear PCA.

**[0066]** Figures 1A and 1B demonstrate the basic idea behind nonlinear kernel PCA. Consider an arbitrary random field in  $R^2$  ( $N_c = 2$ ). In other words, each realization is a vector of dimension two,  $\mathbf{y} = (y_1, y_2)^T$ . Each realization, which are nonlinearly related to each other, can thus be plotted as a point on a 2-D graph, as shown in Figure 1A. However, if linear PCA or the standard Karhunen-Loeve expansion are used, the major principal component obtained is in the direction drawn as an arrow in the graph. Thus, linear PCA is clearly unable to capture the nonlinear relationship among the realizations. In other words, the Karhunen-Loeve expansion is only able to represent linear relationships among the realizations in the space where it is performed, in this case  $R^2$ . Now, consider a nonlinear mapping  $\Phi$  that relates the input space  $R^{N_c}$  to feature space  $F$ . This can be written as:

$$\Phi: R^{N_c} \rightarrow F; \quad \mathbf{Y} = \Phi(\mathbf{y}); \quad \mathbf{y} \in R^{N_c}, \mathbf{Y} \in F \quad (17)$$

**[0067]** Here,  $F$  can have an arbitrarily large dimensionality. As seen in Figure 1B, after this  $\Phi$  transform, the realizations that were nonlinearly related in  $R^2$  become linearly related in  $F$ . Thus, standard linear PCA (i.e., K-L expansion) can now be performed in  $F$  in order to determine the eigenvectors in this space.

**[0068]** Realizing that nonlinear kernel PCA is essentially linear PCA in a high dimensional feature space (as opposed to the input space  $R^{N_c}$ ), an eigenvalue problem similar to standard PCA can be formulated in the feature space  $F$ . Additionally, because the covariance matrix in feature space can be very large and intractable, instead an equivalent eigenvalue problem of a much smaller kernel matrix  $\mathbf{K}$  can be performed. Here, kernel matrix  $\mathbf{K}$  is defined as:

$$\mathbf{K}: \quad K_{ij} = \left( \Phi(\mathbf{y}_i) \cdot \Phi(\mathbf{y}_j) \right) \quad (18)$$

[0069] Each element of the kernel matrix is a dot product of vectors in  $F$ , and not the dot product of vectors in the input space of the realizations,  $R^{N_c}$ . The dimension of the kernel matrix is  $N_R \times N_R$  ( $N_R$ =number of realizations), and therefore, its eigen-decomposition can still be performed very efficiently. Because only the dot products in the space  $F$  are used for application of kernel PCA, and not  $\Phi(\mathbf{y})$  itself, the components of  $\mathbf{K}$  can be calculated directly (and very efficiently) using what is known as a kernel function,  $k(\mathbf{y}_1, \mathbf{y}_2)$ :

$$(\Phi(\mathbf{y}_1) \cdot \Phi(\mathbf{y}_2)) = k(\mathbf{y}_1, \mathbf{y}_2) \tag{19}$$

[0070] The following polynomial kernel can be used to preserve high order statistics:

$$(\Phi(\mathbf{y}_1) \cdot \Phi(\mathbf{y}_2)) = k(\mathbf{y}_1, \mathbf{y}_2) = \sum_{i=1}^d (\mathbf{y}_1 \cdot \mathbf{y}_2)^i \tag{20}$$

[0071] Here  $d$  is a positive integer. The case of  $d = 1$  is equivalent to linear PCA. Accordingly, the feature space is the same as  $R^{N_c}$  and two-point statistics will be preserved. However, for larger values of  $d$ , all moments up to the  $2d^{th}$  order moment ( $2d$  order statistics) are preserved.

[0072] Because in kernel PCA, the K-L expansion is performed in the high order feature space, the results of the K-L expansion (realizations) thus lie in the feature space (i.e., a simulated realization  $\mathbf{Y} \in F$ ). However, since the goal is to obtain a parameterization of this input random field, there is interest in obtaining realizations in the original space of the input random field  $R^{N_c}$ . This can be obtained by solving what is known as a pre-image problem. The final parameterization obtained by solving the pre-image problem for the kernel of Equation **Error! Reference source not found.** is given as:

$$g(\mathbf{y}, \xi) = \sum_{k=1}^d k(\mathbf{y} \cdot \mathbf{y})^{k-1} \mathbf{y} - \sum_{i=1}^{N_R} \beta_i \sum_{k=1}^d k(\mathbf{y}_i \cdot \mathbf{y})^{k-1} \mathbf{y}_i = 0 \tag{21}$$

[0073] This representation is very useful as it is a continuously differentiable parameterization relating a realization  $\mathbf{y}$  in the input space  $R^{N_c}$  to a set of independent random variables  $\xi$ .

Here,  $\beta_i$  are functions of  $\xi$ ). Unlike the standard K-L expansion, however, this is an implicit nonlinear parameterization, and the gradient of  $\mathbf{y}$  with respect to  $\xi$  required for the subspace formulation can be obtained by setting  $dg = 0$ .

$$\frac{d\mathbf{y}}{d\xi} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \xi} \quad (22)$$

[0074] Note that the subspace formulation with an implicit parameterization of the form  $g(\mathbf{y}, \xi) = 0$  is given by:

$$\xi_j^u = \xi_j^f + \left(\frac{\partial g^j}{\partial \mathbf{y}_j^f}\right)^{-1} \left(\frac{\partial g^j}{\partial \xi_j^f}\right) \mathbf{K}_g(\mathbf{d}_{o,j} - \mathbf{d}_j) \Big|_{1:N_m} \quad \forall j = 1, \dots, M \quad (23)$$

[0075] After the updated  $\xi_j^u$  is obtained, the updated  $\mathbf{y}_j^u$  is obtained by solving  $g(\mathbf{y}^u, \xi^u) = 0$ .

This is done using fixed point iteration with relaxation:

$$\mathbf{y}_j^{u,n+1} = \frac{\theta}{1+\theta} \mathbf{y}_j^{u,n} + \frac{1}{1+\theta} \frac{\sum_{i=1}^{N_R} \beta_{ij} \sum_{k=1}^d (\mathbf{y}_i^f \cdot \mathbf{y}_j^{u,n})^{k-1} \mathbf{y}_i^f}{\sum_{k=1}^d k (\mathbf{y}_j^{u,n} \cdot \mathbf{y}_j^{u,n})^{k-1}} \quad \forall j = 1, \dots, M \quad (24)$$

### Example Application of the Subspace EnKF

[0076] A 2D 2-phase synthetic model is chosen for validation of the subspace EnKF. The first example demonstrates application of the subspace EnKF with the standard PCA parameterization, and the second example demonstrates application of the subspace formulation with an iterative EnS and kernel PCA parameterization.

[0077] In the first example, the model consists of 100X100 cells and 4 producers and one water injector are present in one corner of the reservoir as shown in Figure 2. For the purpose of data assimilation, all properties of the reservoir are assumed to be known except the permeability

field, and the true permeability field is also shown in Figure 2. This permeability field is multi-Gaussian and has a correlation length of 15 grids in one direction and 50 in the other and the anisotropy angle is 25 degrees. The observed data used for assimilation is the oil and water production rates and the injection BHPs. Standard normal noise with a standard deviation of 10% is added to the observed data to mimic reality.

**[0078]** Ten (10) members of the initial 35 member ensemble for the standard EnKF is shown in Figure 3. No information about the true permeability field except that it is multi-Gaussian is assumed to be known a priori. Therefore, the 35 members of the initial ensemble are generated using correlation lengths varying from 10 to 90 grids and anisotropy angles varying from 0 to 90 degrees. Figure 4 shows the variance of log permeability across the initial ensemble. Figure 5 shows the oil and water production rates for the initial ensemble. Clearly, it has a large spread and encompasses the observed data well. Figure 6 shows 10 ensemble members of the final ensemble of the standard EnKF. All the different anisotropy angles and correlation lengths of the initial ensemble are no longer visible in the final ensemble, and the ensemble has collapsed with all ensemble members looking essentially the same. Ensemble collapse can also be verified from Figure 7, which shows that the variance across the final ensemble members is close to zero, even far away from the wells. This clearly demonstrates that such an ensemble, although providing a good match to the data (Figure 8), would do poorly in predictions far away from the wells.

**[0079]** Figure 9 shows 10 members of the final ensemble obtained with the subspace EnKF. It is clear that these ensemble members retain the anisotropy angle and correlation lengths of the initial ensemble, and the variance across the ensemble, although small near the wells as expected, is larger away from the wells (Figure 10), and thus should provide better predictions away from the wells. Figure 11 shows the oil and water production rates for the final ensemble from the subspace EnKF. Although the data match is not as good as the standard EnKF, it is acceptable and is much better than the initial ensemble. The reason why the match is not as good as the standard EnKF is that, unlike the standard EnKF, where all ensemble members converge toward the best matching ensemble member, in the subspace EnKF, because each member is constrained to a subspace, they cannot converge to a best matching ensemble member, but can only converge toward solutions that best match the data within their own subspaces. This also

implies that some members may not provide an acceptable data match if no such solutions exist within their subspace. This indicates that the geological model that the subspace represents is not acceptable and thus, provides a way to discard prior geological models that are not probable.

**[0080]** As mentioned earlier, the purpose of history-matching is not only matching the observed data, but more importantly to provide good predictions of future production. To test the predictive capacity of the standard EnKF vs. the subspace EnKF, the 5 wells are now removed from their original locations and randomly placed throughout the reservoir. Their production rates from the final ensembles for both approaches are then compared to the productions rates from the true permeability field for the same well locations and schedules. The metric chosen to compare the predictive capacity is the percentage of time the true production rates lie within the range predicted by the ensemble. This test is done 20 times for different random well locations, and the results are shown in Figure 12. It is clear that the subspace EnKF does better in 75% of the cases, which is significantly better than the standard EnKF. Overall, the true production rates lie within the predicted production rates 71% of the time on an average for the subspace EnKF, and only 58% of the time for the standard EnKF. Figures 13 and 14 show the predicted production rates and the true production rates for a typical case for the standard and subspace EnKF.

**[0081]** The second example is similar to the first except that the permeability field is a non-Gaussian channel sand as seen in Figure 15, with the sand permeability being about 10 D and the background permeability about 500 mD. The contrast in permeability between the high permeability sand and the background reservoir is about a factor of 20. The reservoir is modeled by a  $45 \times 45 \times 1$  horizontal 2D grid. It is essentially an incompressible two-phase unit mobility oil-water system, with zero connate water saturation and zero residual oil saturation. The model is run for 900 days with 1 injector well under rate control and 4 producer wells under bottom hole pressure (BHP) control with predefined rates and BHPs. The objective, like the first example, is to estimate the unknown permeability field using observed data that consists of the injector BHPs and oil and water production rates. Standard normal noise with a standard deviation of 10% is added to the observed data to mimic reality. Although the permeability field is assumed unknown for the purpose of this study, prior geological knowledge indicates that this is a fluvial

channelized reservoir, as shown in the training image in Figure 16. Geostatistical software *snesim* is used to create 2400 realizations of the permeability field using this training image. Because the KPCA parameterization can be very nonlinear, to obtain a better match to the historical data, an iterative ensemble smoother (EnS) is used instead of the EnKF. The iterative EnS is essentially application of the EnS on the same data multiple times with different initial ensembles, which is very similar to a steepest descent algorithm with step length equal to 1 and an average gradient estimated from the ensemble. Note that  $\alpha$  is set to 1 for all iterations as varying  $\alpha$  or having  $\alpha$  different than 1 did not improve results. The ensemble size is set to 40 in all of the experiments below and the number of EnS iterations is set to 5.

**[0082]** Figures 17 – 22 demonstrate the application of the iterative EnS with PCA parameterization. 1000 realizations from the 2400 created earlier were used to create the PCA parameterization with 60 parameters. Figure 17 shows 10 members of the initial ensemble obtained with this standard PCA parameterization. Although longer correlations are obtained in the horizontal direction corresponding to the channels, the realizations are multi-Gaussian as expected and are not able to reproduce the channel structure of the training image. The variance of permeability across the initial ensemble is reasonable as seen in Figure 18, which is also observed through the large spread of the production data in Figure 19. Figure 20 shows 10 realizations of the final ensemble after 5 iterations of the EnS. The ensemble has clearly collapsed as confirmed by Figure 21, and additionally, the channel structure is completely absent from these realizations. However, the data match is almost perfect, as seen in Figure 22, implying that the production data is not enough to uniquely identify the permeability field.

**[0083]** Figures 23 – 28 demonstrate the application of the iterative EnS with KPCA parameterization with a 5<sup>th</sup> order polynomial kernel. As above, 1000 realizations from the 2400 created earlier were used to create the KPCA parameterization with 60 parameters. Compared to the standard PCA parameterization above, while the final ensemble is clearly able to reproduce channel looking structures as seen in Figure 26, the ensemble has again collapsed as confirmed by Figure 27. Furthermore, the final converged realization(s) is clearly quite different from the true realization. Comparing the true realization to the converged realization, while the upper channel's location is correctly determined due to its proximity to the wells, the lower channel is completely missing. Note that the data match is not as of high quality as standard PCA, but it is

quite good. The main reason for the lower quality data match is the higher nonlinearity of the KPCA parameterization.

**[0084]** Figures 29 – 34 demonstrate the application of the subspace iterative EnS with KPCA parameterization with a 5<sup>th</sup> order polynomial kernel. For each of the 40 ensemble members, a different set of 60 realizations from the 2400 available realizations are used to create 40 KPCA parameterizations, each with 60 parameters as above. Figure 29 shows 10 members of the initial ensemble and Figure 30 shows the variance across the ensemble. Again the variance is reasonable as also confirmed by the spread of the production data in Figure 31. Figure 32 shows 10 members of the final ensemble. While the channel structures are as well defined as the example above without the subspace formulation, with the subspace formulation, the ensemble has clearly not collapsed. Again, while the upper channel exists in all the final realizations due to its proximity to the wells, the lower channel is now present in some realizations and not in others. In other words, while uncertainty in the location of the upper channel is reduced by assimilating production data, because the lower channel is far away from the wells, production data from these wells do not contain enough information about the lower channel to reduce its uncertainty. The variance across the final realizations as seen in Figure 33 also confirms this. Although the data match is not as good at the iterative EnS without the subspace formulation, it is acceptable and is much better than the initial ensemble.

**[0085]** To summarize, the basic idea of the subspace EnKF/EnS is to constrain the different ensemble members to different subspaces of the same or different random fields. This is accomplished by parameterizing the random fields and modifying the EnKF/EnS formulation with the gradients of the parameterizations. The subspace EnKF/EnS prevents ensemble collapse, providing a better quantification of uncertainty, and more importantly, retains key geological characteristics of the initial ensemble, even when each ensemble member belongs to a different geological model. The subspace EnKF/EnS can further be improved to honor multi-point geostatistics by extending the subspace formulation with kernel PCA. Formulating the subspace EnKF/EnS with kernel PCA parameterization alleviates both the key problems of the traditional EnKF/EnS, namely ensemble collapse and inability to honor multi-point geostatistics.

**[0086]** The above described methods can be implemented in the general context of instructions executed by a computer. Such computer-executable instructions may include programs, routines, objects, components, data structures, and computer software technologies that can be used to perform particular tasks and process abstract data types. Software implementations of the above described methods may be coded in different languages for application in a variety of computing platforms and environments. It will be appreciated that the scope and underlying principles of the above described methods are not limited to any particular computer software technology.

**[0087]** Moreover, those skilled in the art will appreciate that the above described methods may be practiced using any one or a combination of computer processing system configurations, including, but not limited to, single and multi-processor systems, hand-held devices, programmable consumer electronics, mini-computers, or mainframe computers. The above described methods may also be practiced in distributed computing environments where tasks are performed by servers or other processing devices that are linked through a one or more data communications networks. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

**[0088]** Also, an article of manufacture for use with a computer processor, such as a CD, pre-recorded disk or other equivalent devices, could include a computer program storage medium and program means recorded thereon for directing the computer processor to facilitate the implementation and practice of the above described methods. Such devices and articles of manufacture also fall within the spirit and scope of the present invention.

**[0089]** As will be described, the invention can be implemented in numerous ways, including for example as a method (including a computer-implemented method), a system (including a computer processing system), an apparatus, a computer readable medium, a computer program product, a graphical user interface, a web portal, or a data structure tangibly fixed in a computer readable memory. Several embodiments of the present invention are discussed below. The appended drawings illustrate only typical embodiments of the present invention and therefore are not to be considered limiting of its scope and breadth.

[0090] Figure 1 illustrates a system 200 for updating simulation models of a subterranean reservoir, such as by using computer-implemented method. System 200 includes user interface 210, such that an operator can actively input information and review operations of system 200. User interface 210 can be any means in which a person is capable of interacting with system 200 such as a keyboard, mouse, or touch-screen display. Operator-entered data input into system 200 through user interface 210, can be stored in database 230. Reservoir field data, including for example, well pressure, oil production, water production, gas production, or seismic data, which is received by one or more reservoir sensors 220, can also be input into system 200 for storage in database 230. Additionally, any information generated by system 200 can be stored in database 230.

[0091] Database 230 can store user-defined variables, equations and parameters such as equality constraints and inequality constraints, as well as, reservoir production data, and system generated computed solutions. For example, reservoir simulation models 231 and reservoir data 233, such as static variables (permeability fields, porosity fields), dynamic variables (pressure, saturation), and production data (production and injection rates, bottom-hole pressures), are all examples of information that can be stored in database 230. In some embodiments, reservoir simulation models 231 and reservoir data 233 are stored in database 230 as a series of state vectors or an ensemble of realizations for a given assimilation time. An initial ensemble of models can be provided that includes initial reservoir characteristics, at a time  $t_0$ , derived by operators and/or computer modeling based on initial observations, knowledge or data gathered about the reservoir. The initial ensemble of models may take into account initial estimates of permeability, porosity and other characteristics used to create simulation models of subterranean reservoirs.

[0092] System 200 includes a computer program product or software 240 that is stored on a non-transitory processor readable medium. Current examples of a processor readable medium include, but are not limited to, an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable programmable ROM (EPROM), a floppy diskette, a compact disk (CD-ROM), an optical disk, a hard disk, and a fiber optic medium. As will be described more fully herein, software 240 can include a plurality of modules for performing system tasks such as performing the methods previously described herein. Processor 250 interprets instructions to execute software 240, as well as, generates automatic instructions to execute software for

system 200 responsive to predetermined conditions. Instructions from both user interface 210 and software 240 are processed by processor 250 for operation of system 200. In some embodiments, a plurality of processors can be utilized such that system operations can be executed more rapidly.

**[0093]** In certain embodiments, system 200 can include reporting unit 260 to provide information to the operator or to other systems (not shown). For example, reporting unit 260 can be a printer, display screen, or a data storage device. However, it should be understood that system 200 need not include reporting unit 260, and alternatively user interface 210 can be utilized for reporting information of system 200 to the operator.

**[0094]** Communication between any components of system 200, such as user interface 210, reservoir sensors 220, database 230, software 240, processor 250 and reporting unit 260, can be transferred over a communications network 270. Communications network 270 can be any means that allows for information transfer. Examples of communications network 270 presently include, but are not limited to, a switch within a computer, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), and a global area network (GAN). Communications network 270 can also include any hardware technology used to connect the individual devices in the network, such as an optical cable or wireless radio frequency.

**[0095]** In operation, an operator initiates software 240, through user interface 210, to perform the methods described herein. Outputs from software 240 can be stored in database 230. For example, software 240 can update an ensemble of reservoir models. A visual display can be produced, such as through reporting unit 260 or user interface 210, using the updated ensemble of reservoir models. For example, the updated ensemble of reservoir models can be transformed into image data representations of the subterranean reservoir for display to a user or operator. The displayed information can be utilized to forecast or optimize the production performance of the subterranean reservoir, which can then be used for reservoir management decisions.

**[0096]** As used in this specification and the following claims, the terms “comprise” (as well as forms, derivatives, or variations thereof, such as “comprising” and “comprises”) and “include” (as well as forms, derivatives, or variations thereof, such as “including” and “includes”) are

inclusive (i.e., open-ended) and do not exclude additional elements or steps. Accordingly, these terms are intended to not only cover the recited element(s) or step(s), but may also include other elements or steps not expressly recited. Furthermore, as used herein, the use of the terms “a” or “an” when used in conjunction with an element may mean “one,” but it is also consistent with the meaning of “one or more,” “at least one,” and “one or more than one.” Therefore, an element preceded by “a” or “an” does not, without more constraints, preclude the existence of additional identical elements.

[0097] The use of the term “about” applies to all numeric values, whether or not explicitly indicated. This term generally refers to a range of numbers that one of ordinary skill in the art would consider as a reasonable amount of deviation to the recited numeric values (i.e., having the equivalent function or result). For example, this term can be construed as including a deviation of  $\pm 10$  percent of the given numeric value provided such a deviation does not alter the end function or result of the value. Therefore, a value of about 1% can be construed to be a range from 0.9% to 1.1%.

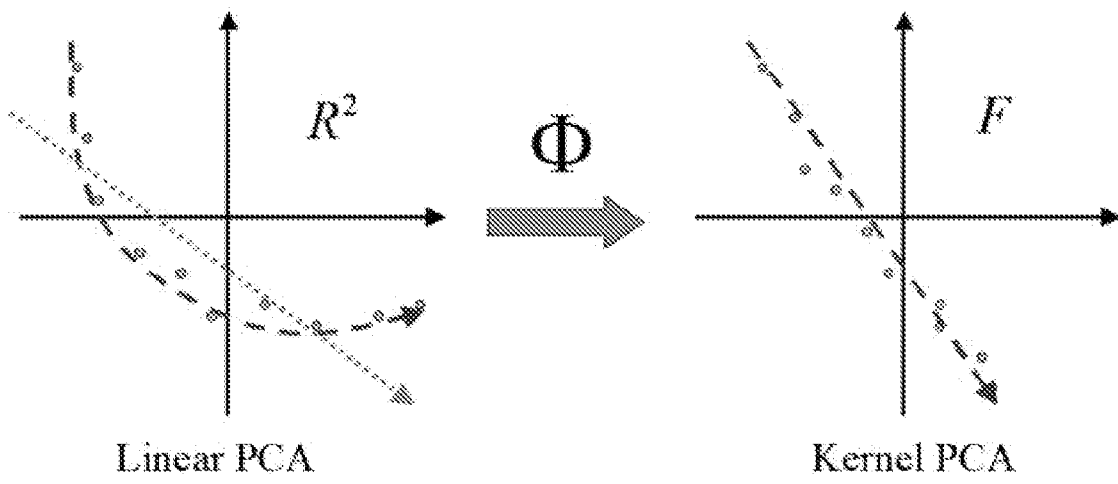
[0098] While in the foregoing specification this invention has been described in relation to certain preferred embodiments thereof, and many details have been set forth for the purpose of illustration, it will be apparent to those skilled in the art that the invention is susceptible to alteration and that certain other details described herein can vary considerably without departing from the basic principles of the invention. For example, while embodiments of the present disclosure are described with reference to block diagrams and/or operational illustrations of methods, systems, and computer program products, the functions/acts described in the figures may occur out of the order (i.e., two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order).

## WHAT IS CLAIMED IS:

1. A computer-implemented method for updating simulation models of a subterranean reservoir, the method comprising:
  - (a) providing an ensemble of reservoir models representing a subterranean reservoir having non-Gaussian characteristics; and
  - (b) updating the ensemble of reservoir models using a subspace ensemble Kalman filter.
2. The computer-implemented method of claim 1, wherein the ensemble Kalman filter constrains each ensemble member of the ensemble of reservoir models to a different subspace.
3. The computer-implemented method of claim 1, wherein updating the ensemble of reservoir models using a subspace ensemble Kalman filter in step (b) further comprises applying a different parameterization to each ensemble member of the ensemble of reservoir models.
4. The computer-implemented method of claim 3, wherein each parameterization ensures that ensemble members remain in a subspace to which it is constrained.
5. The computer-implemented method of claim 3, wherein the parameterization comprises kernel principle component analysis.
6. The computer-implemented method of claim 1, wherein updating the ensemble of reservoir models using a subspace ensemble Kalman filter in step (b) further comprises applying kernel principle component analysis parameterization.
7. The computer-implemented method of claim 1, wherein updating the ensemble of reservoir models using a subspace ensemble Kalman filter in step (b) further comprises applying K-L expansion parameterization.

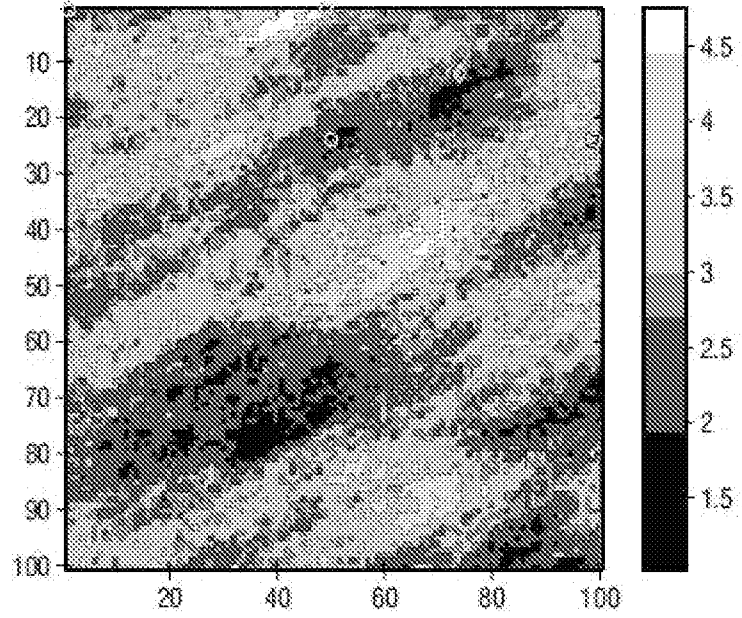
8. A system for updating simulation models of a subterranean reservoir, the system comprising:
  - a database configured to store data comprising reservoir data from a subterranean reservoir having non-Gaussian characteristics and an ensemble of reservoir models representing the subterranean reservoir;
  - a computer processor configured to receive the stored data from the database, and to execute software responsive to the stored data; and
  - a software program executable on the computer processor, the software program containing computer readable software instructions comprising a Subspace Ensemble Kalman Filter module that updates the ensemble of reservoir models using a subspace ensemble Kalman filter.
9. A non-transitory processor readable medium containing computer readable software instructions used for updating simulation models of a subterranean reservoir, the software instructions comprising a Subspace Ensemble Kalman Filter module configured to update an ensemble of reservoir models using a subspace ensemble Kalman filter.
10. Any one of claims 8-9, wherein the Subspace Ensemble Kalman Filter module further constrains each ensemble member of the ensemble of reservoir models to a different subspace.
11. Any one of claims 8-9, wherein the Subspace Ensemble Kalman Filter module further applies a different parameterization to each ensemble member of the ensemble of reservoir models.
12. Claim 11 wherein each parameterization ensures that ensemble members remain in a subspace to which it is constrained.
13. Claim 11 wherein the parameterization comprises kernel principle component analysis.
14. Any one of claims 8-9, wherein the Subspace Ensemble Kalman Filter module applies kernel principle component analysis parameterization to update the ensemble of reservoir models.

15. Any one of claims 8-9, wherein the Subspace Ensemble Kalman Filter module applies K-L expansion parameterization to update the ensemble of reservoir models.

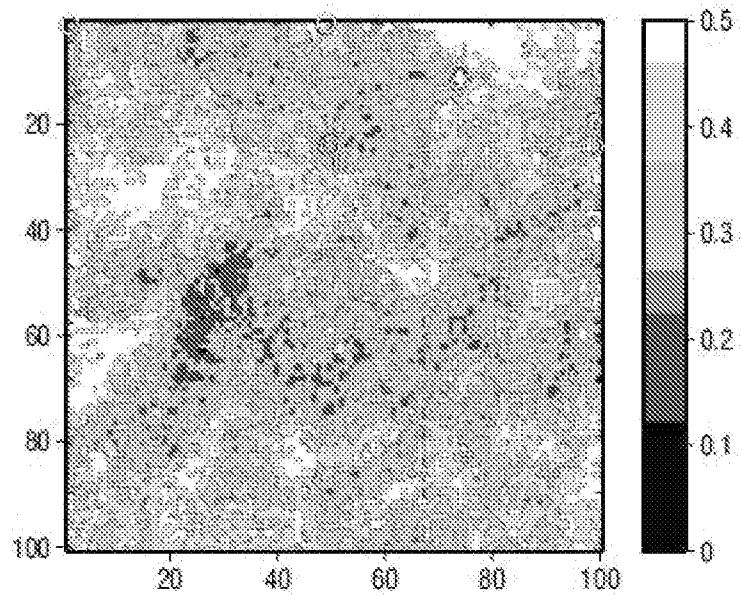


*FIG. 1*

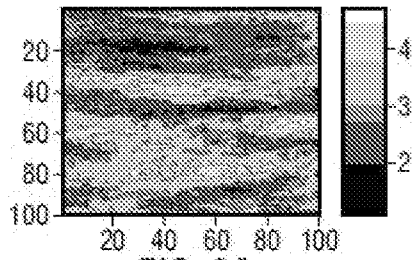
2/35



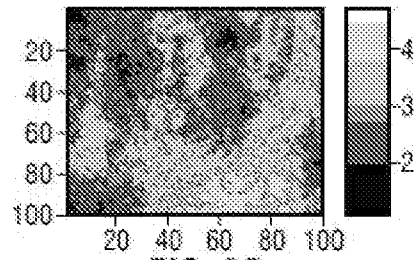
**FIG. 2**



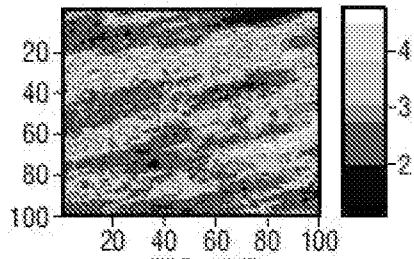
**FIG. 4**



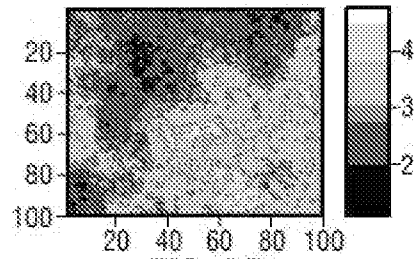
**FIG. 3A**



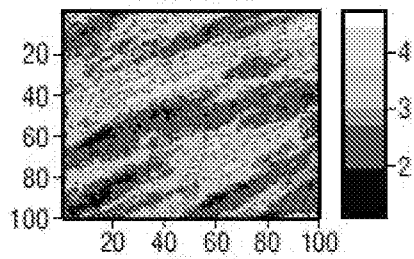
**FIG. 3B**



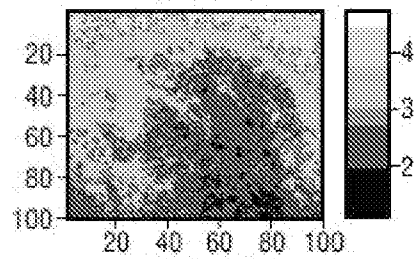
**FIG. 3C**



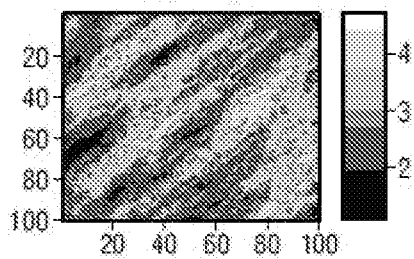
**FIG. 3D**



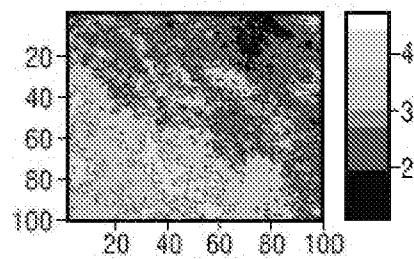
**FIG. 3E**



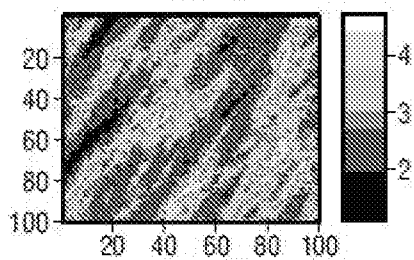
**FIG. 3F**



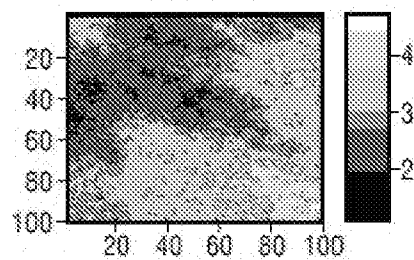
**FIG. 3G**



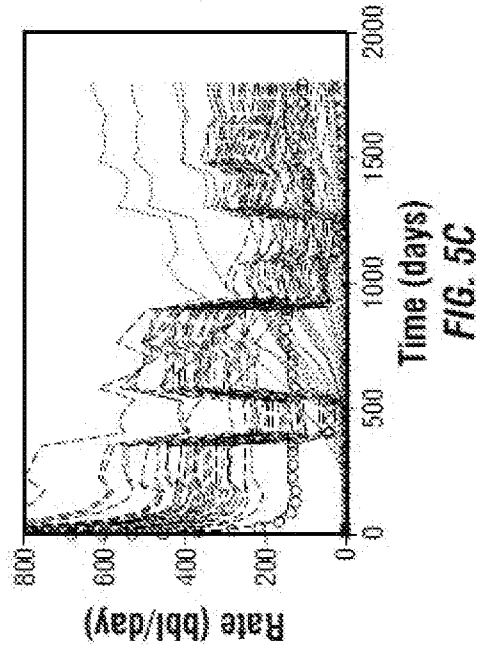
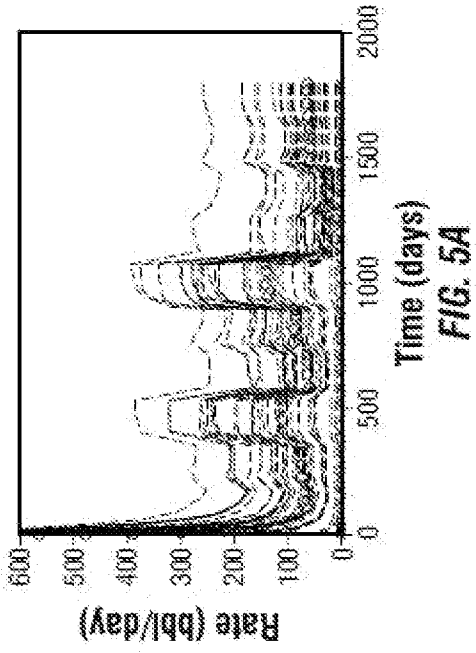
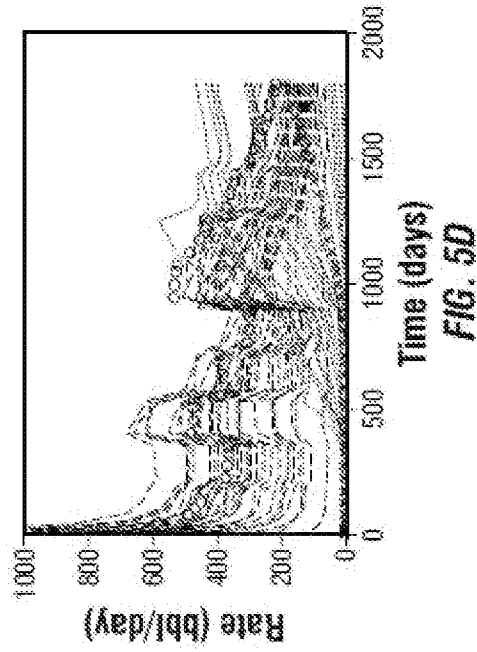
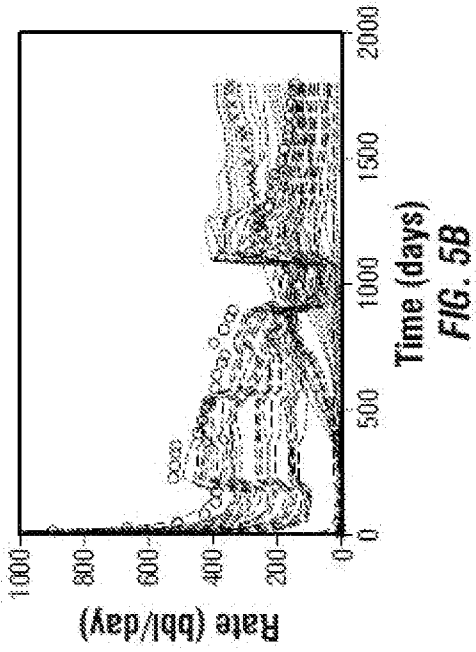
**FIG. 3H**

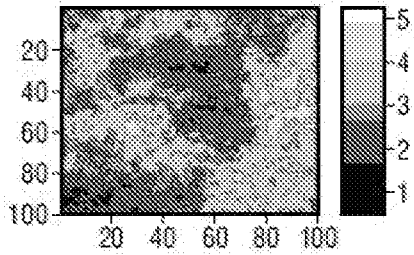


**FIG. 3I**

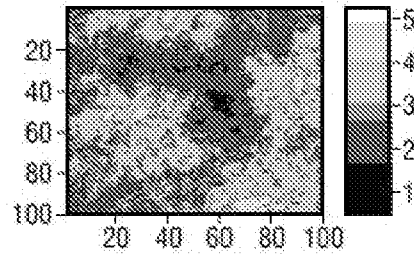


**FIG. 3J**

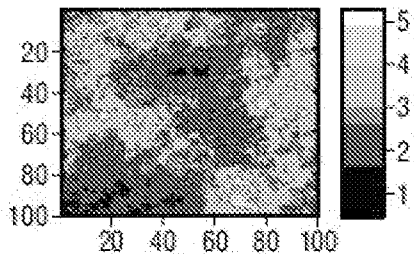




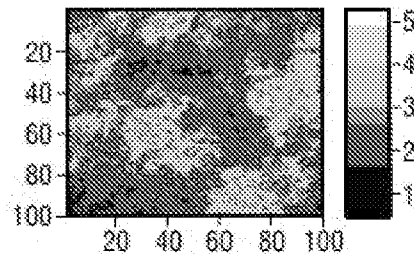
**FIG. 6A**



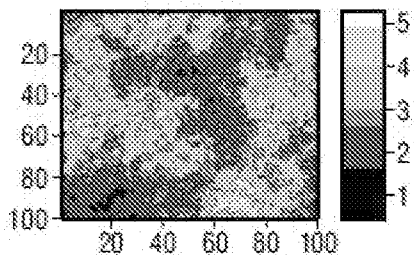
**FIG. 6B**



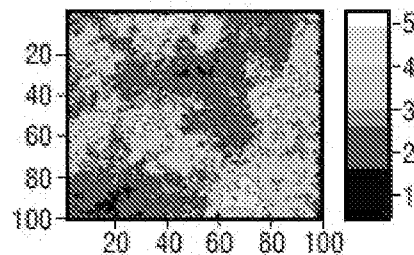
**FIG. 6C**



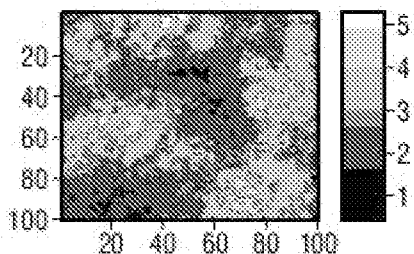
**FIG. 6D**



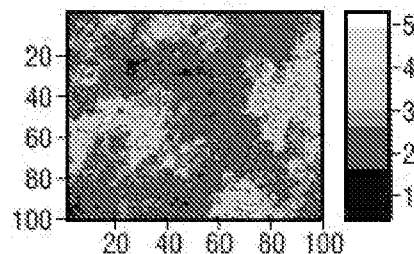
**FIG. 6E**



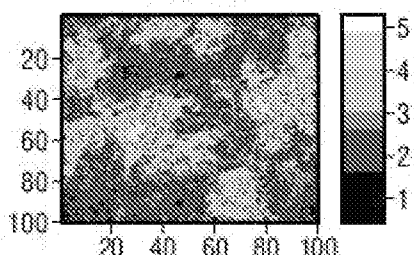
**FIG. 6F**



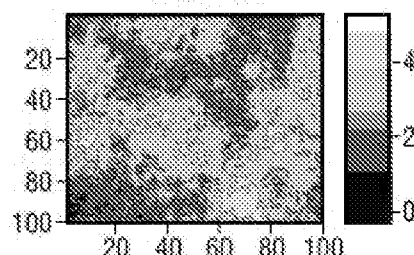
**FIG. 6G**



**FIG. 6H**

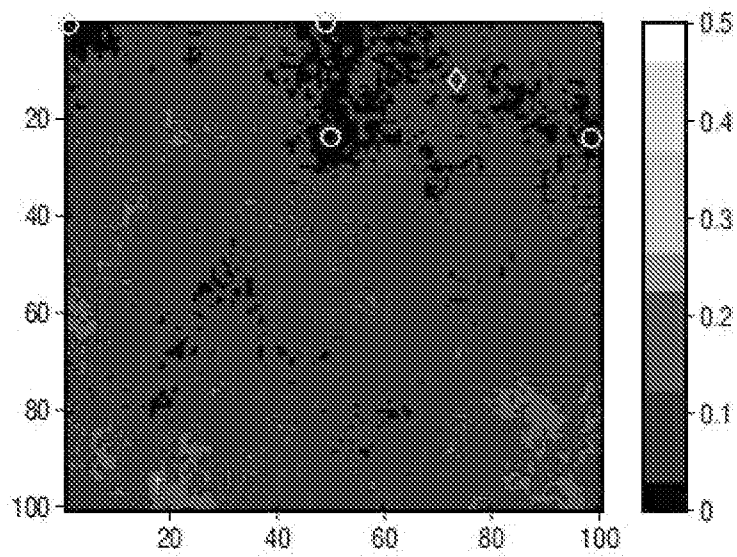


**FIG. 6I**

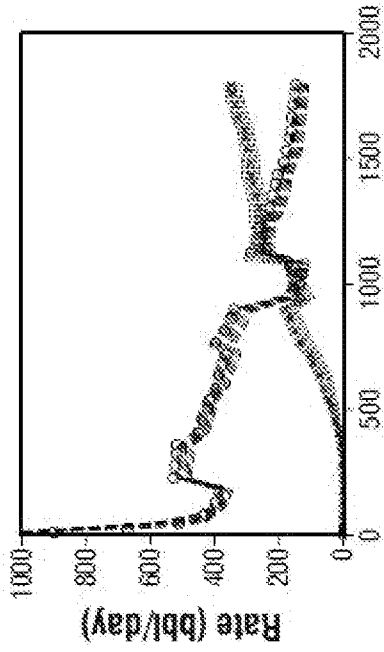


**FIG. 6J**

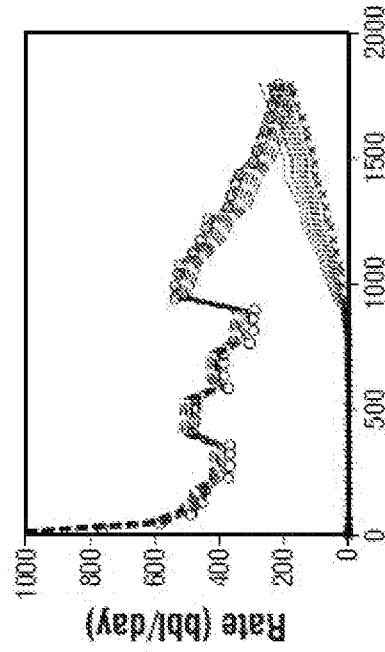
6/35



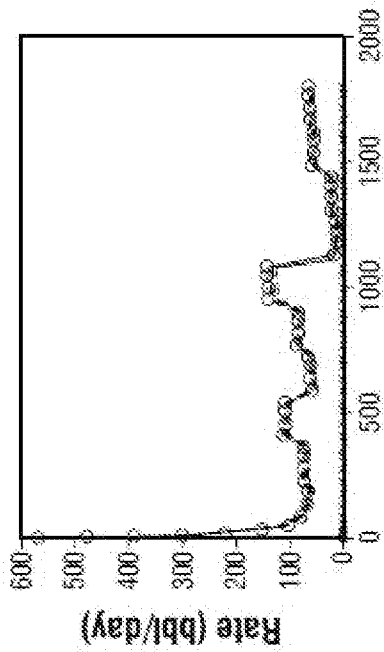
**FIG. 7**



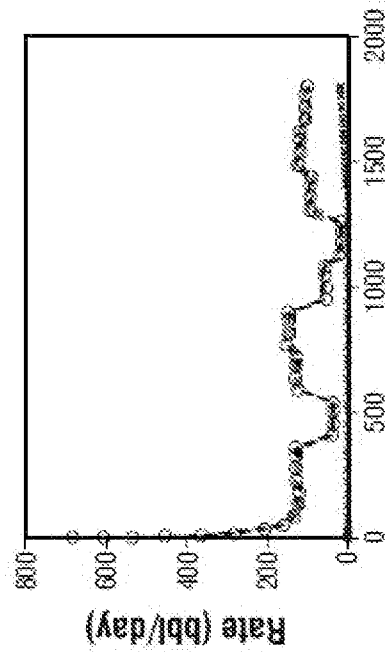
**FIG. 8B**



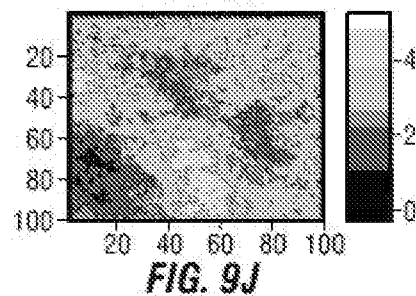
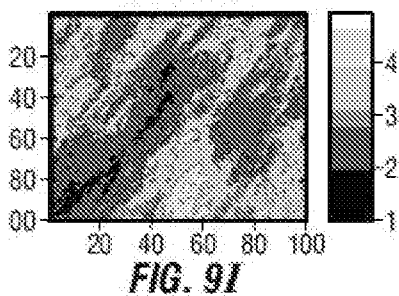
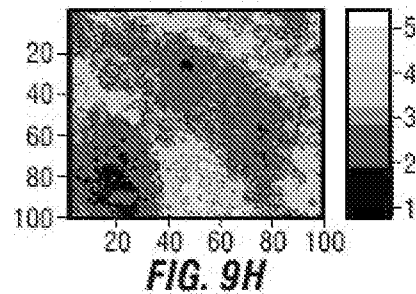
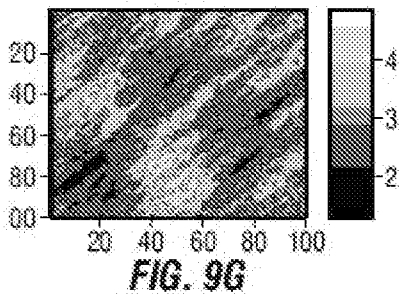
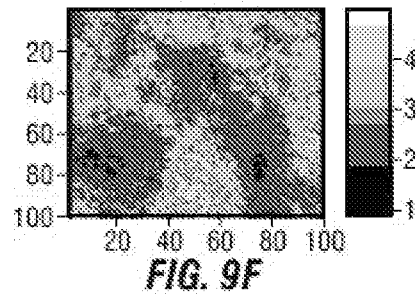
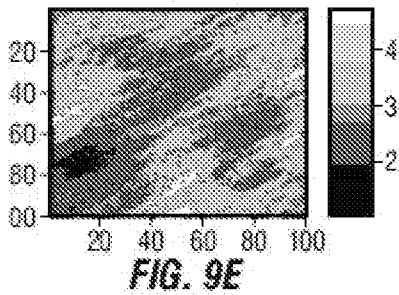
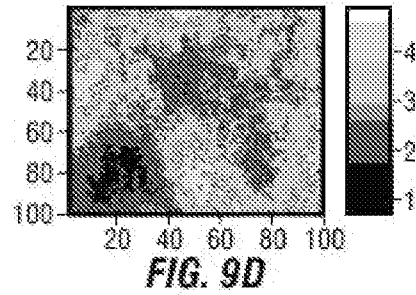
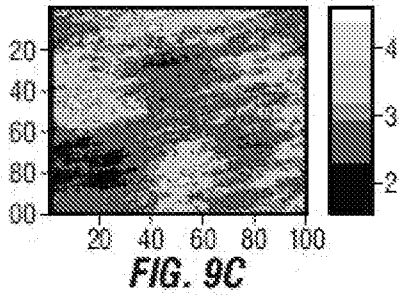
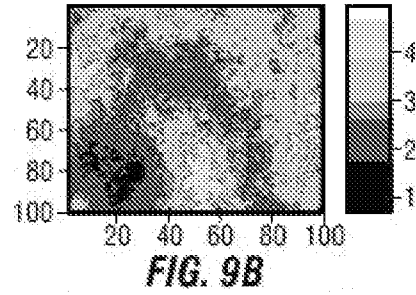
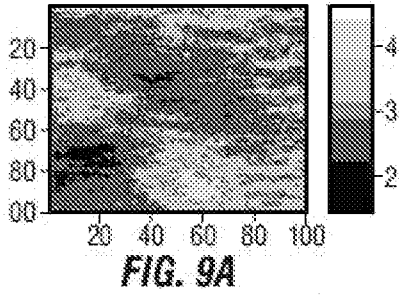
**FIG. 8D**

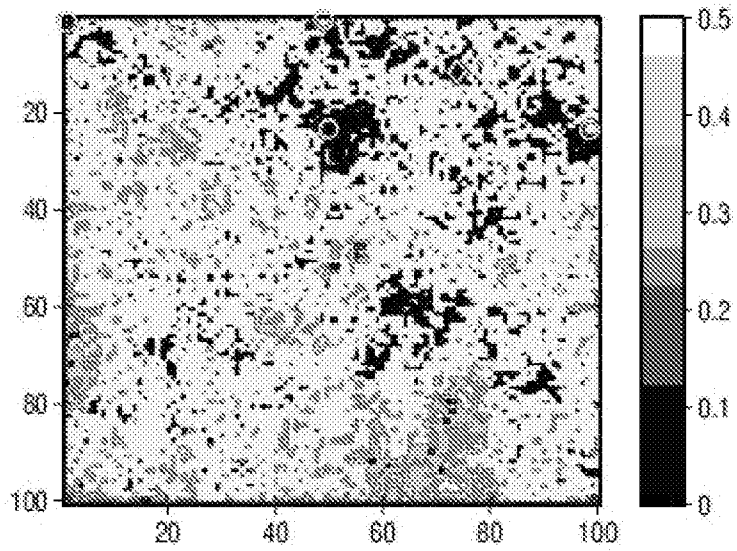


**FIG. 8A**

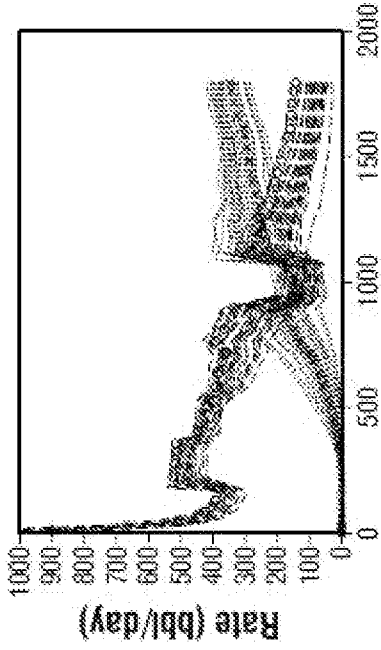


**FIG. 8C**

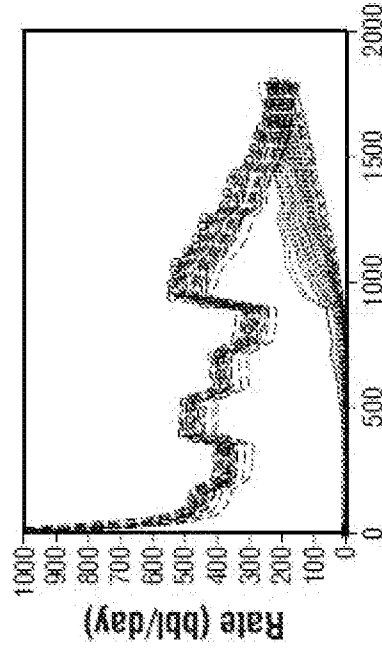




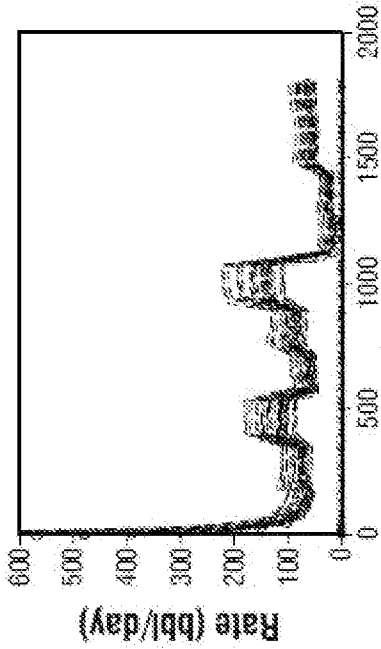
**FIG. 10**



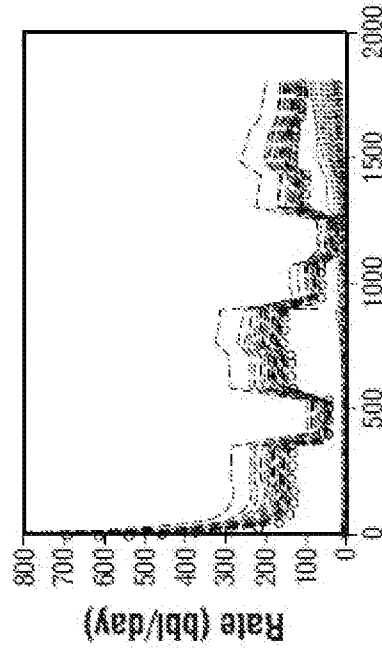
Time (days)  
**FIG. 11B**



Time (days)  
**FIG. 11D**



Time (days)  
**FIG. 11A**



Time (days)  
**FIG. 11C**

11/35

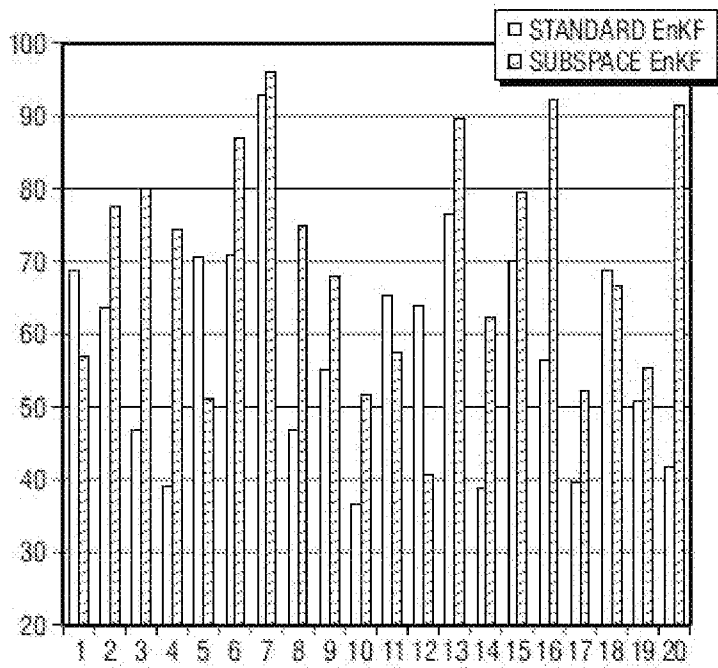


FIG. 12

12/35

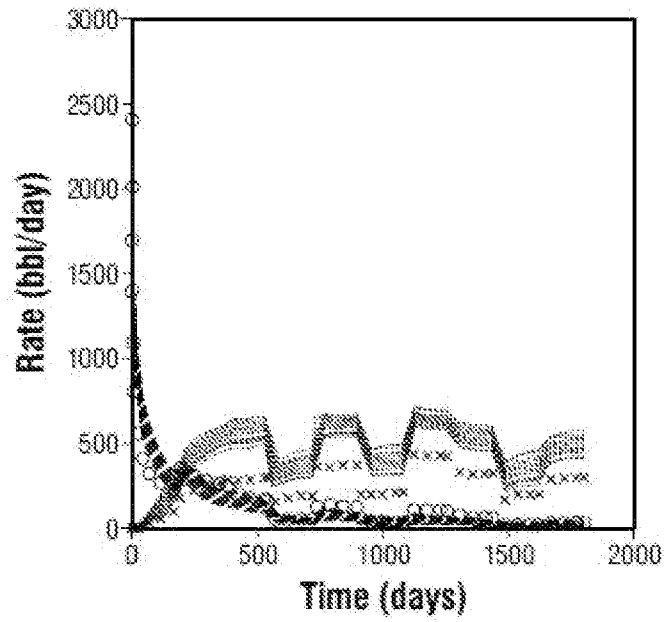


FIG. 13A

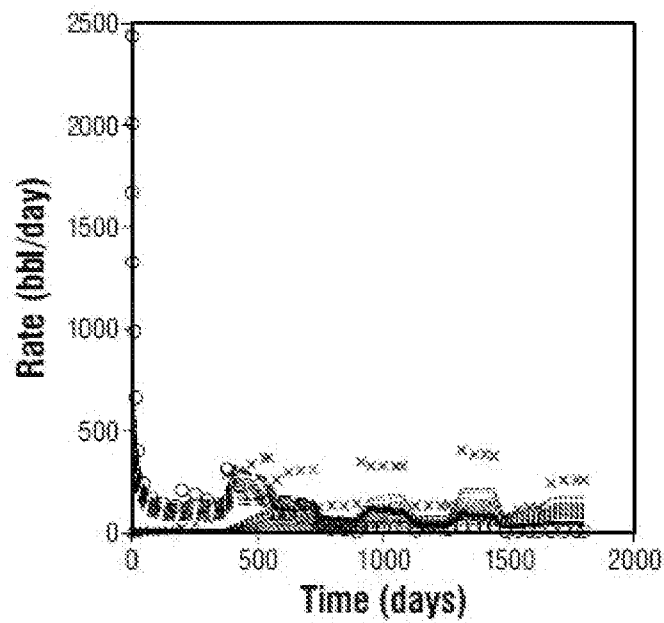


FIG. 13B

13/35

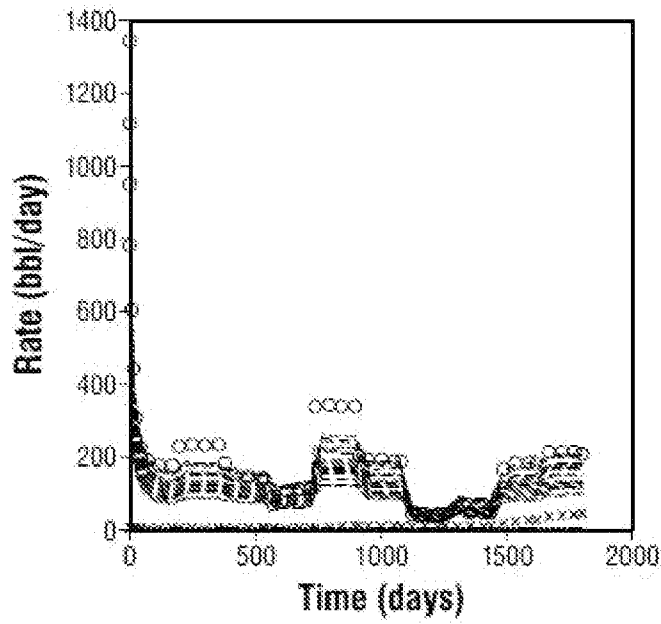


FIG. 13C

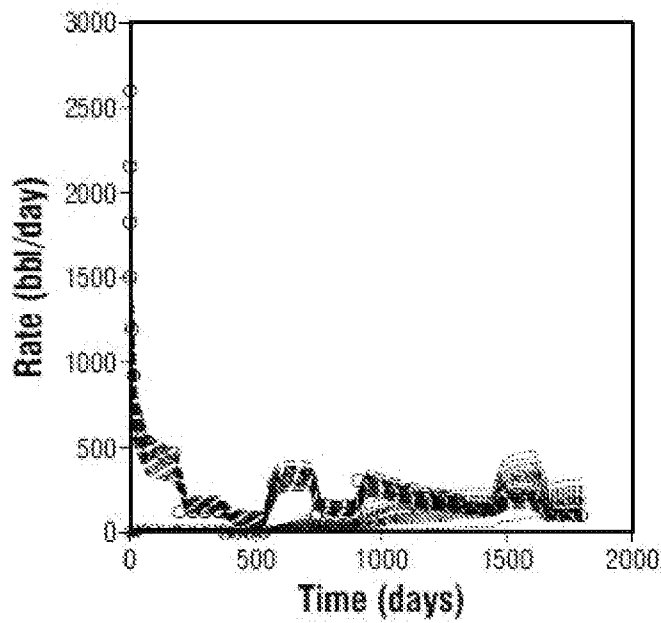
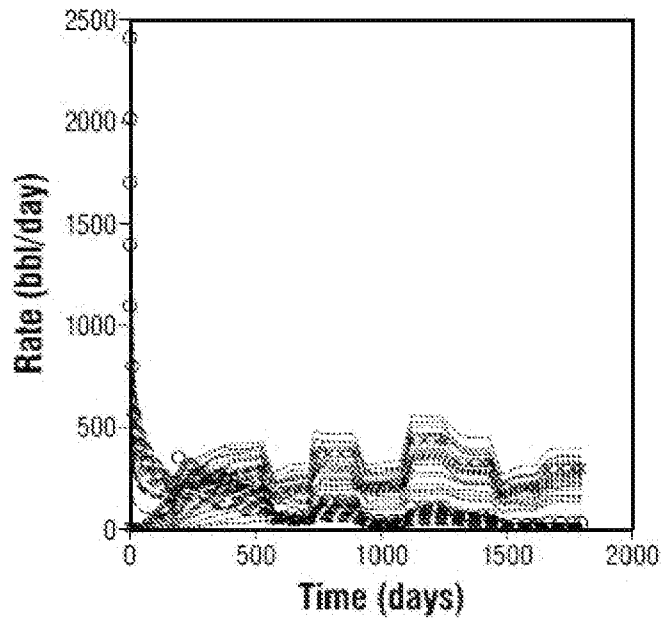
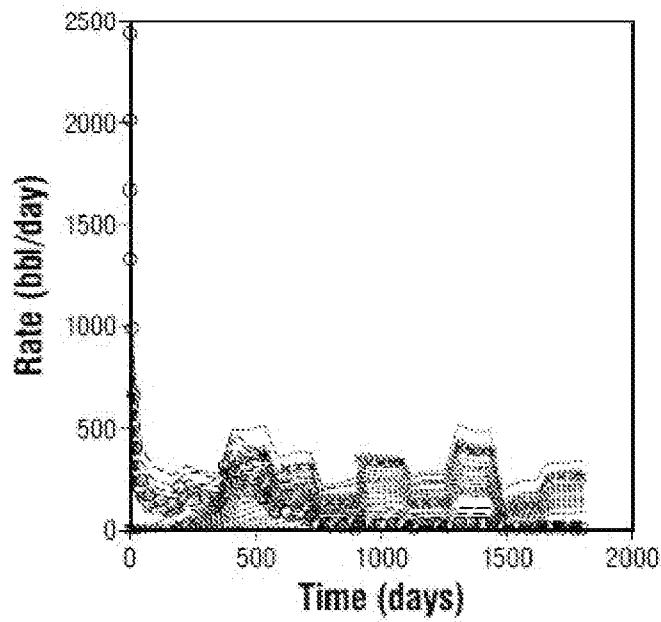


FIG. 13D

14/35

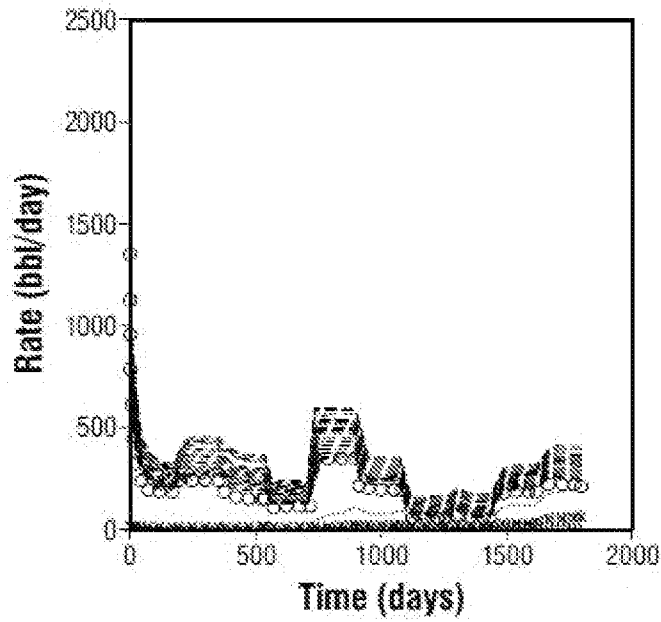


**FIG. 14A**

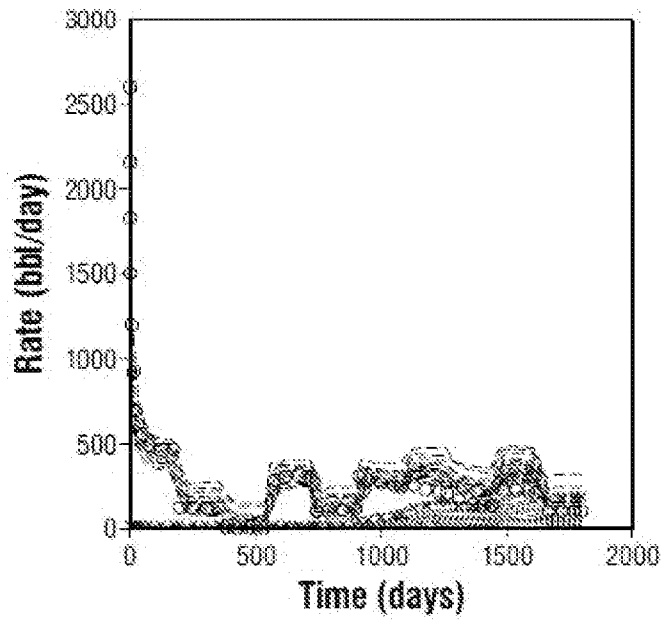


**FIG. 14B**

15/35

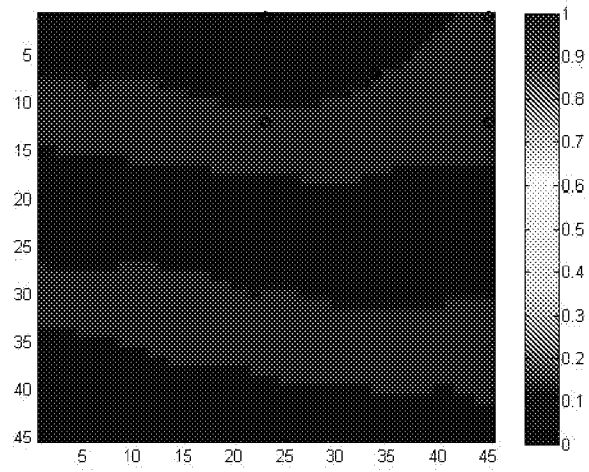


**FIG. 14C**

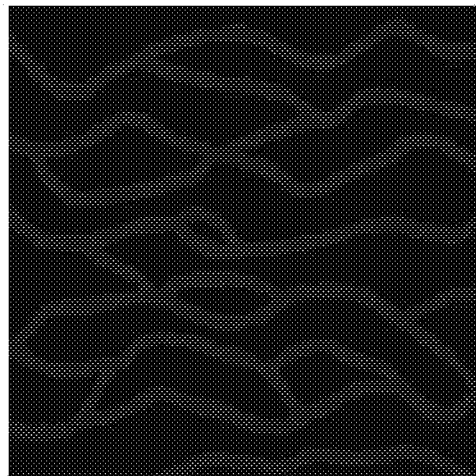


**FIG. 14D**

16/35



*FIG. 15*



*FIG. 16*

17/35

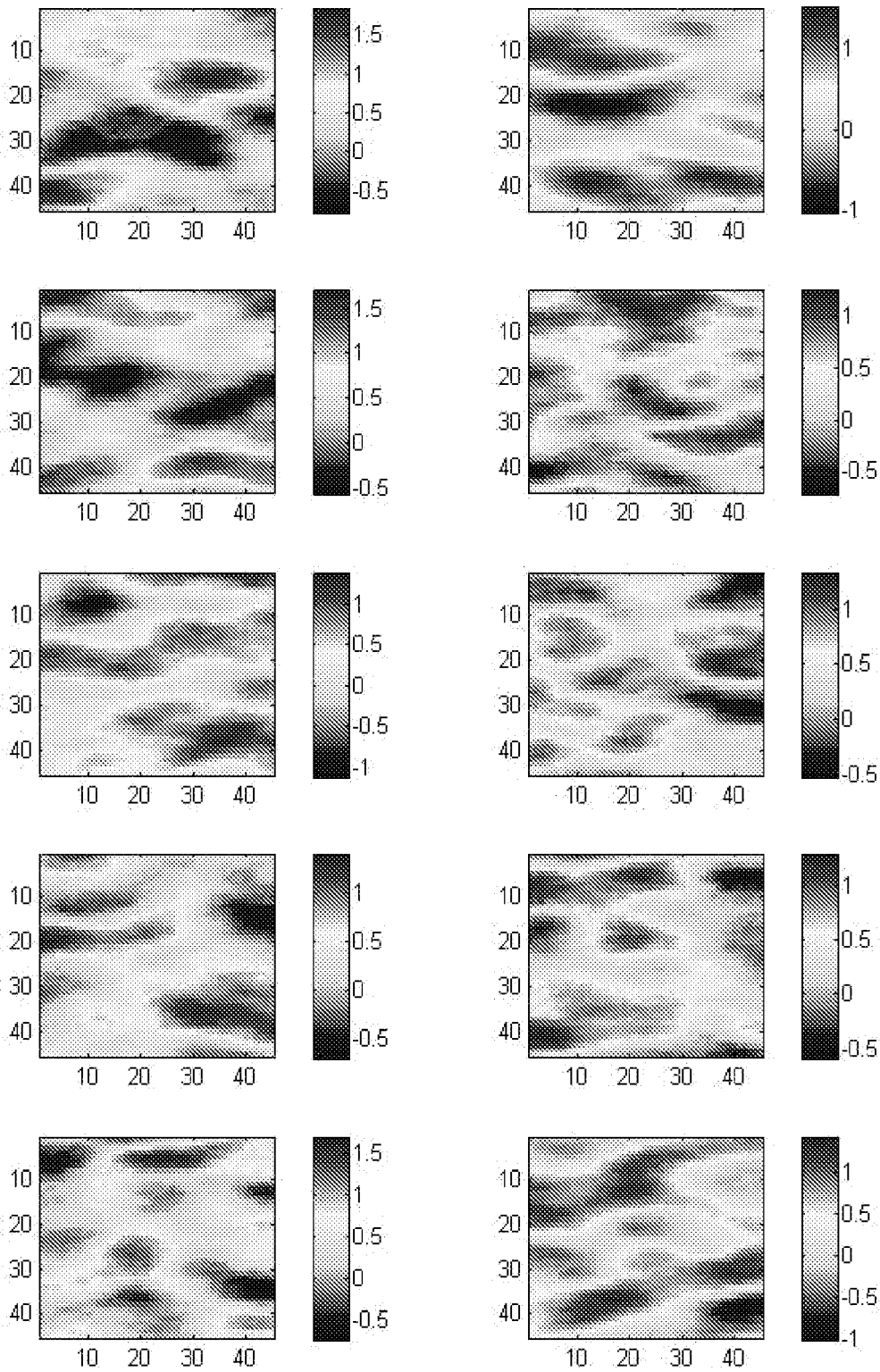
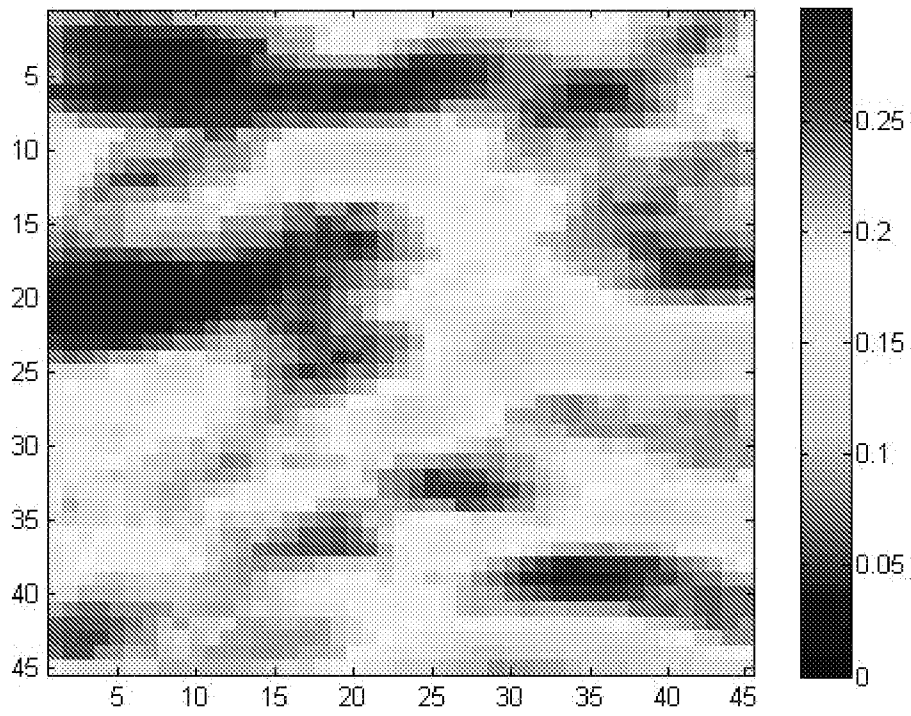
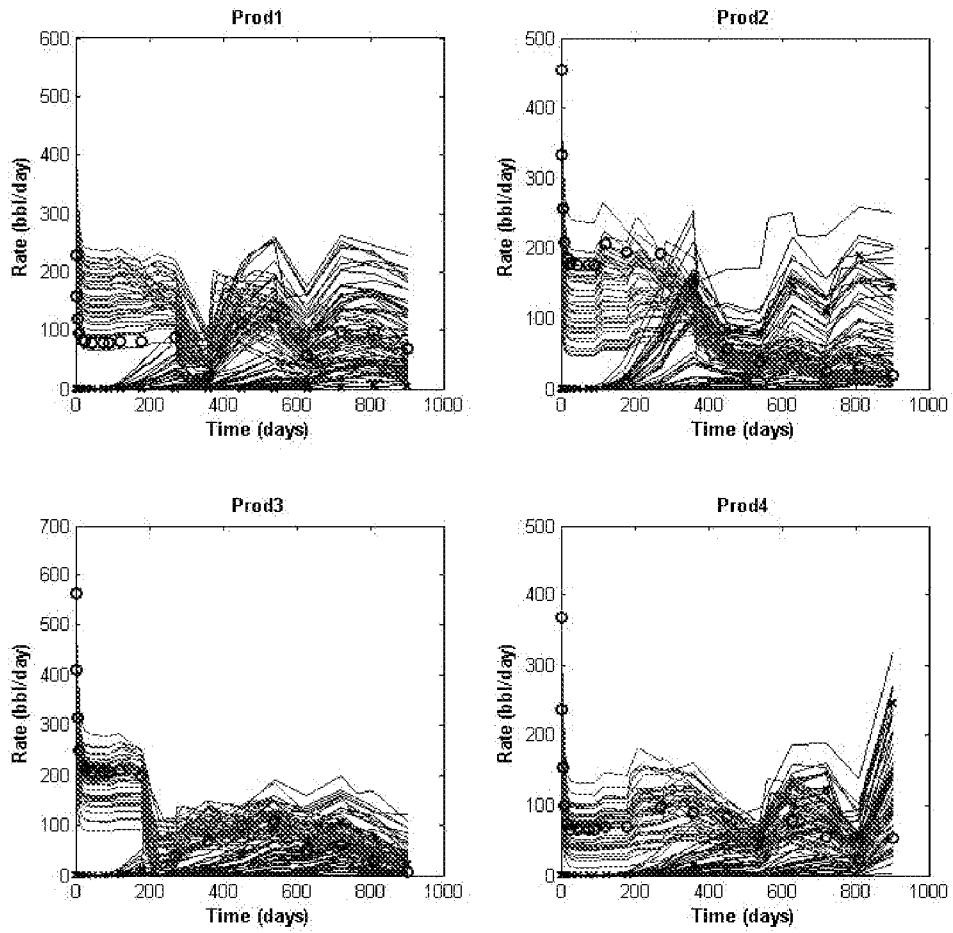


FIG. 17

18/35



*FIG. 18*



*FIG. 19*

20/35

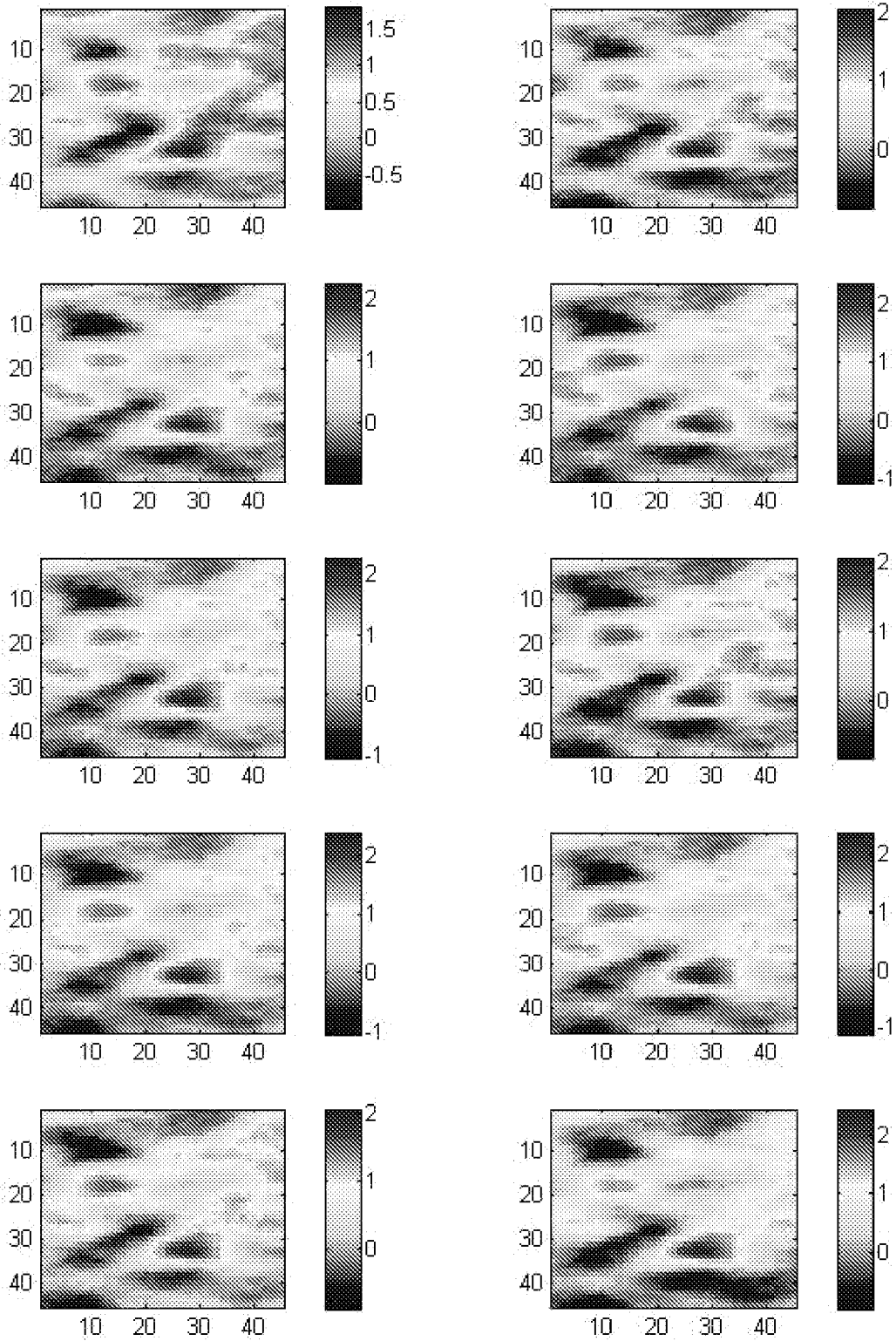
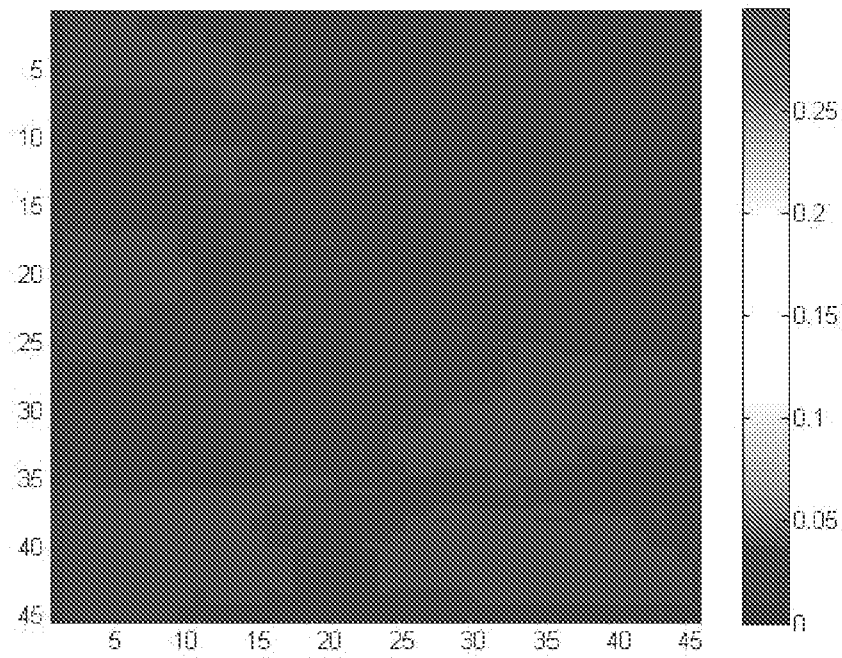


FIG. 20

21/35



*FIG. 21*

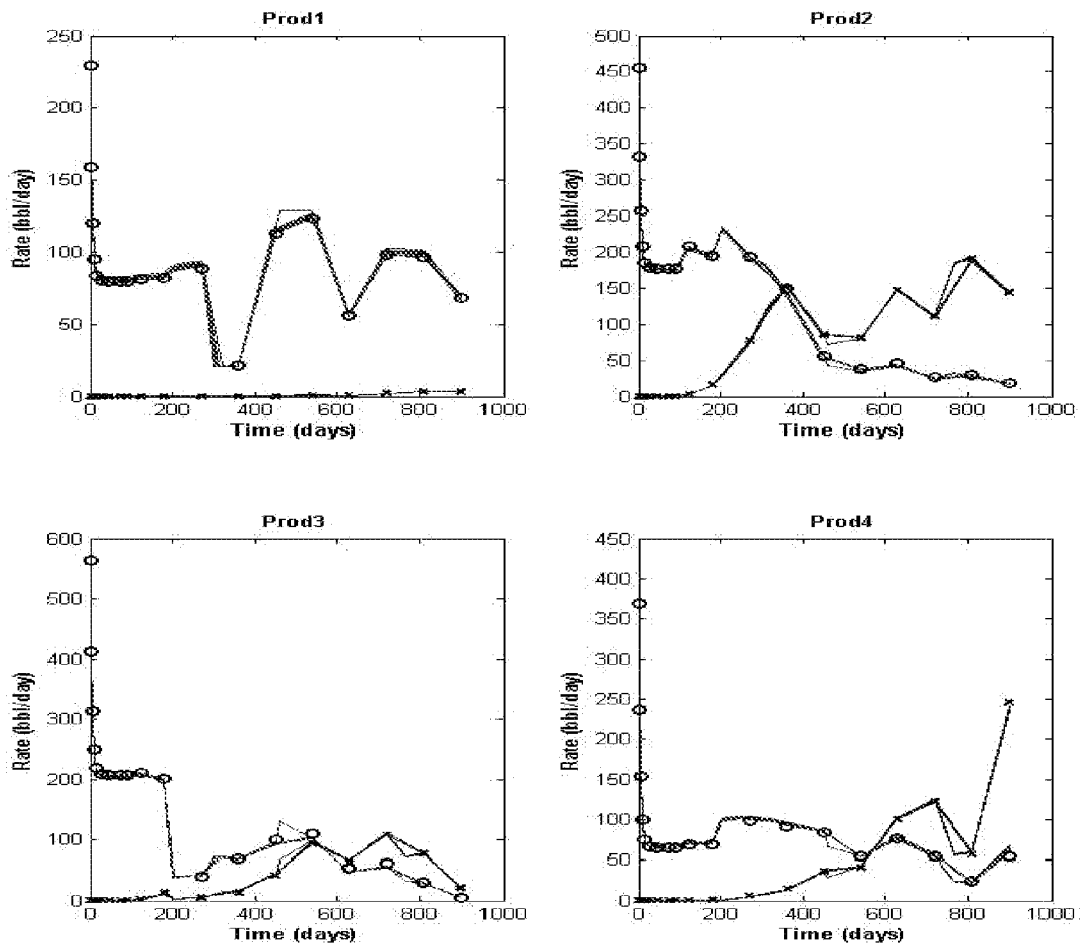
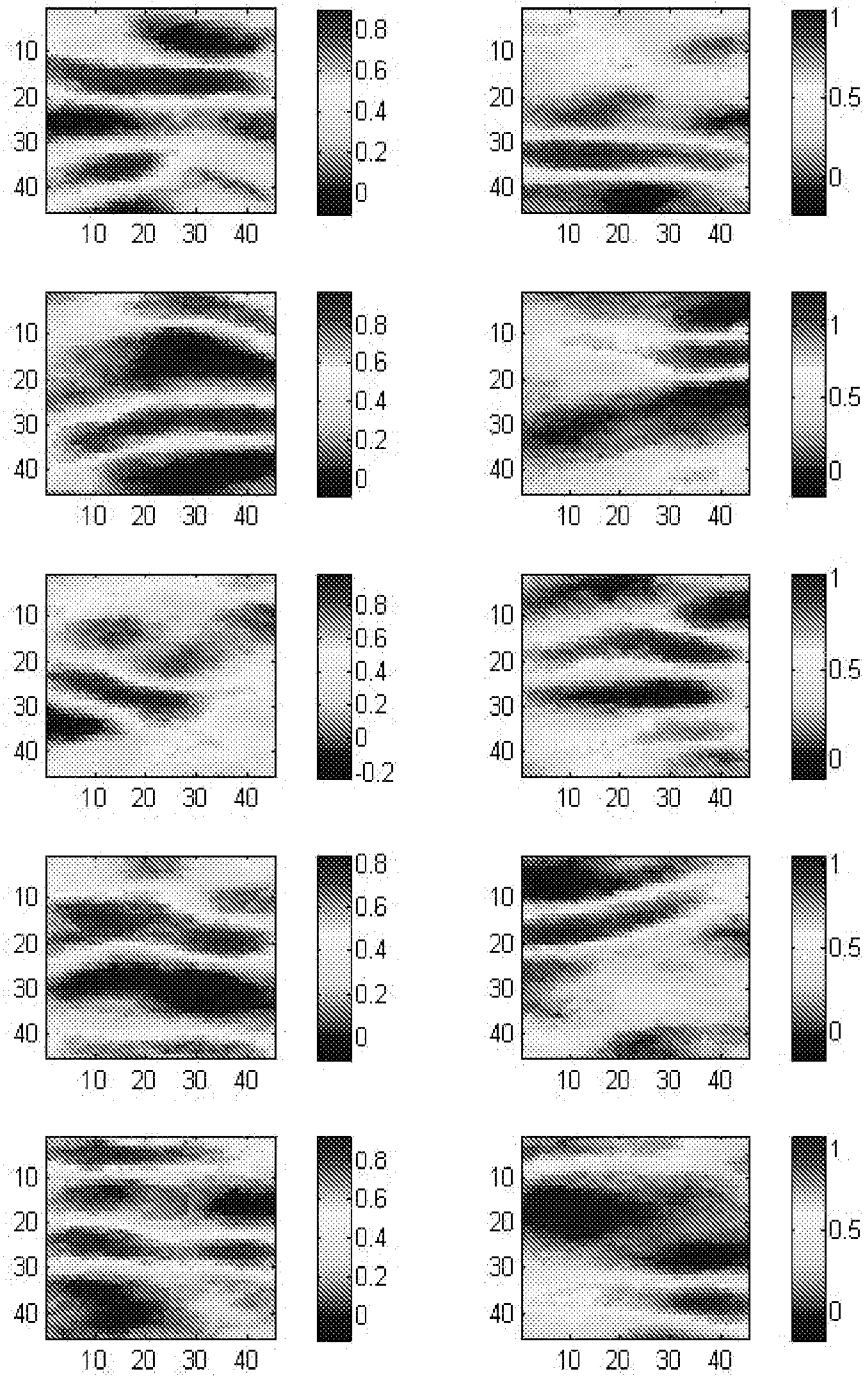
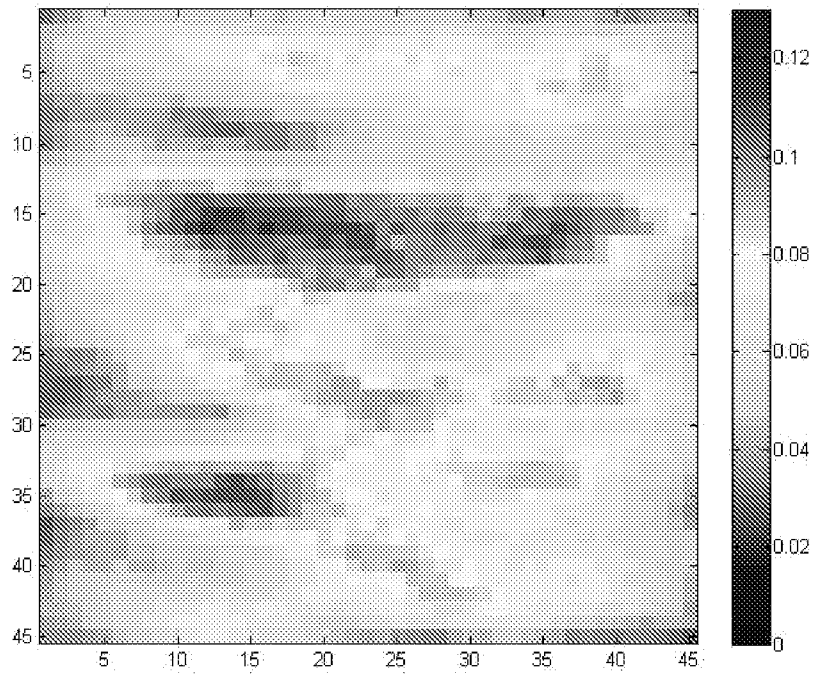


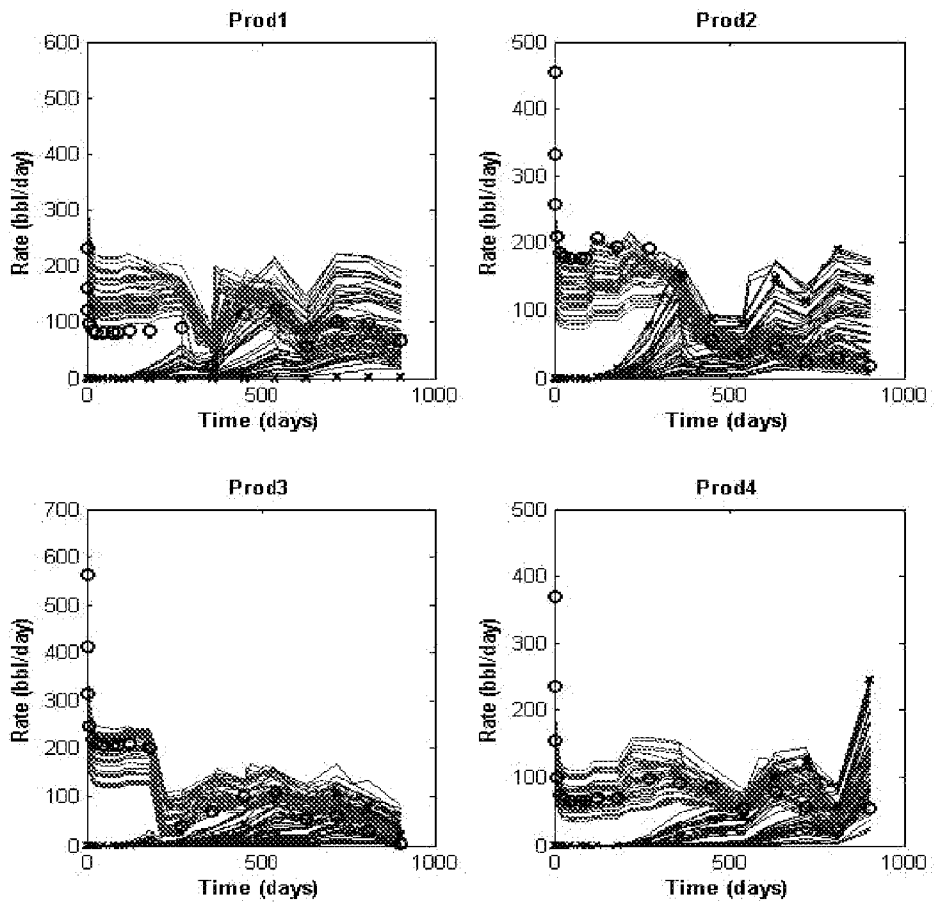
FIG. 22



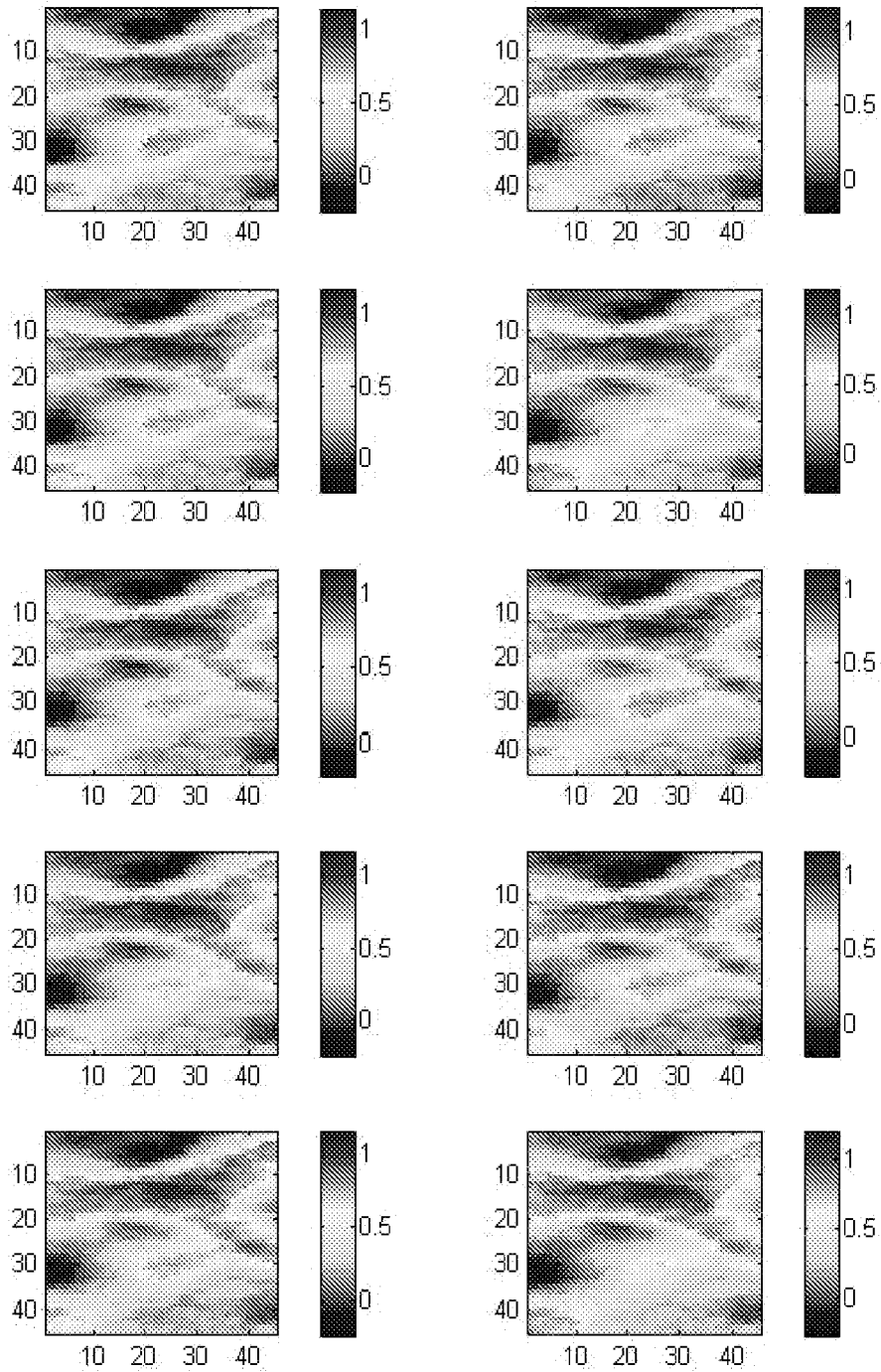
**FIG. 23**



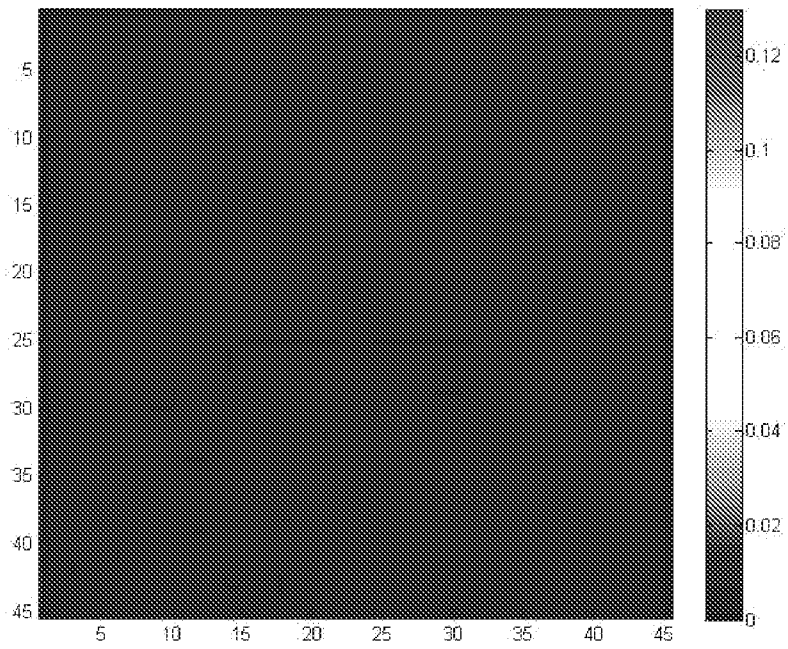
*FIG. 24*



**FIG. 25**



*FIG. 26*



*FIG. 27*

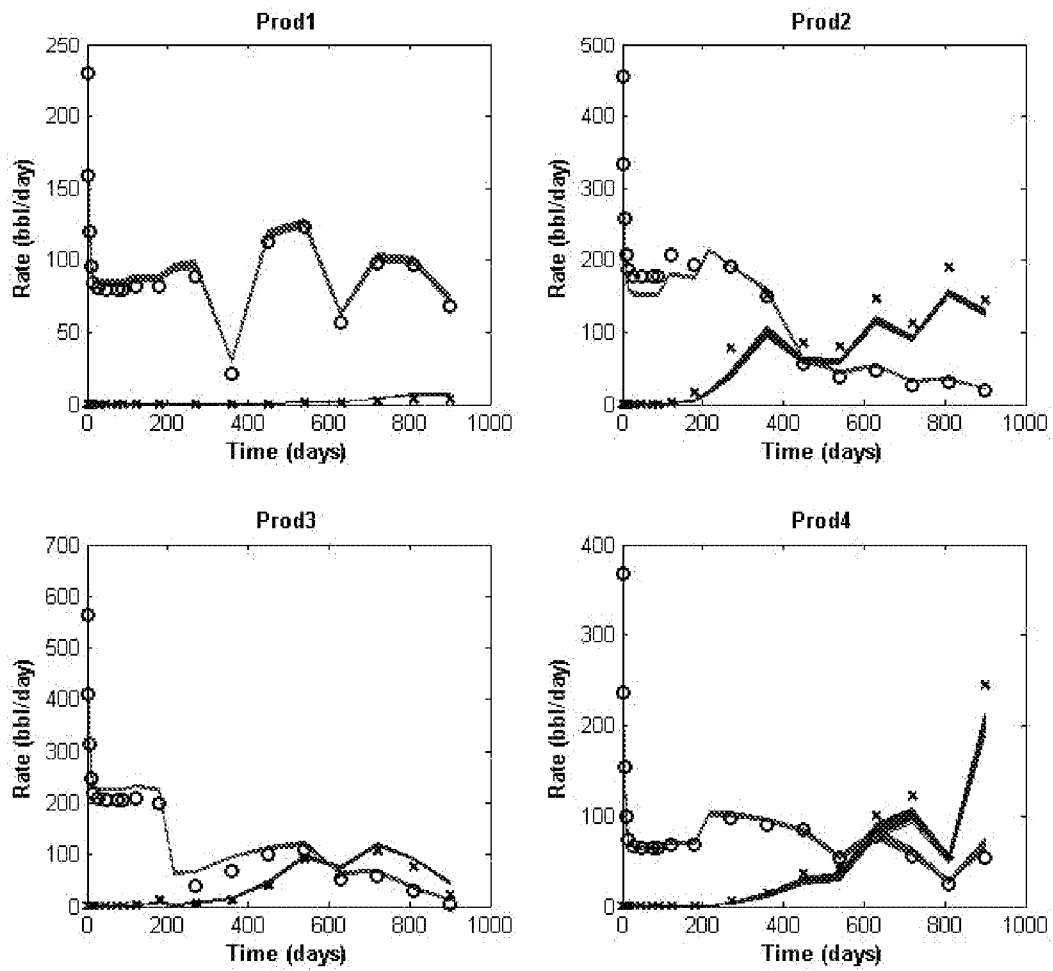
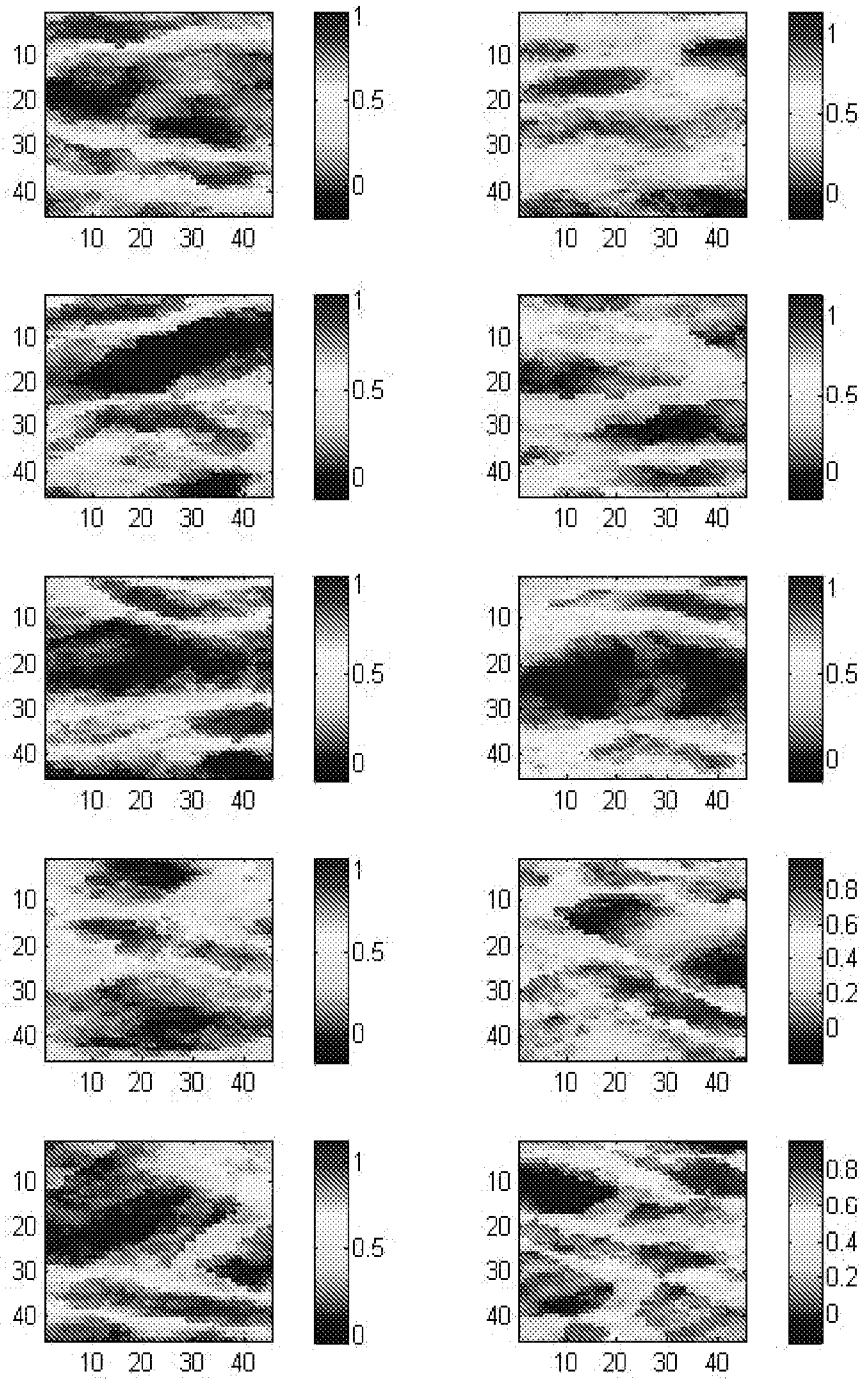
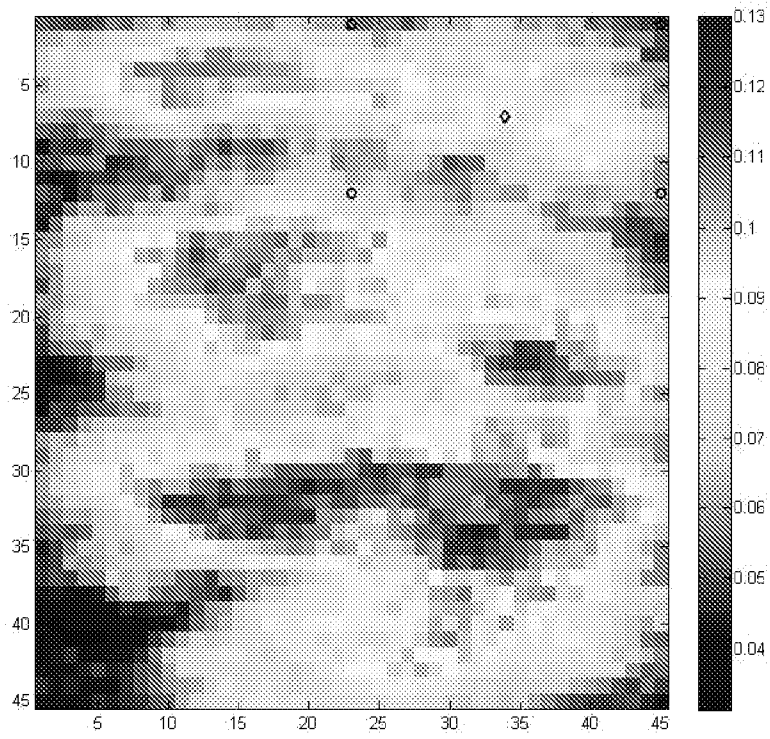


FIG. 28



**FIG. 29**

30/35



*FIG. 30*

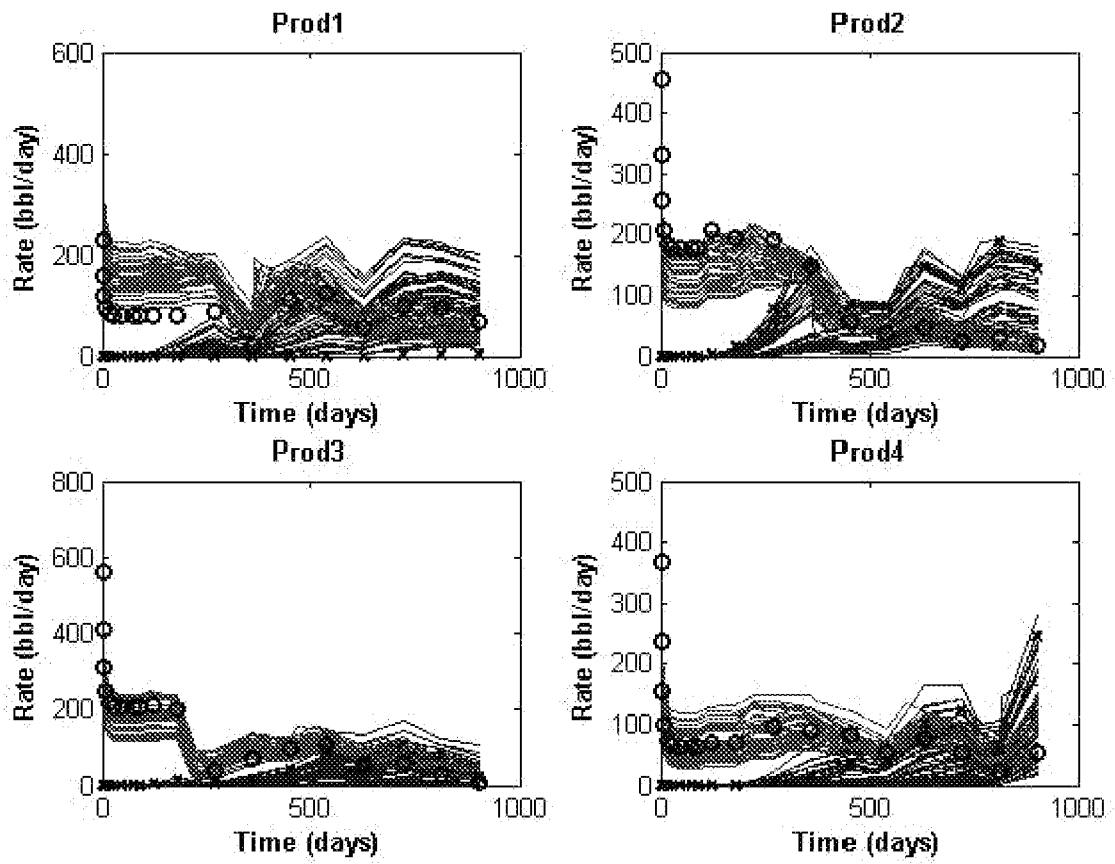
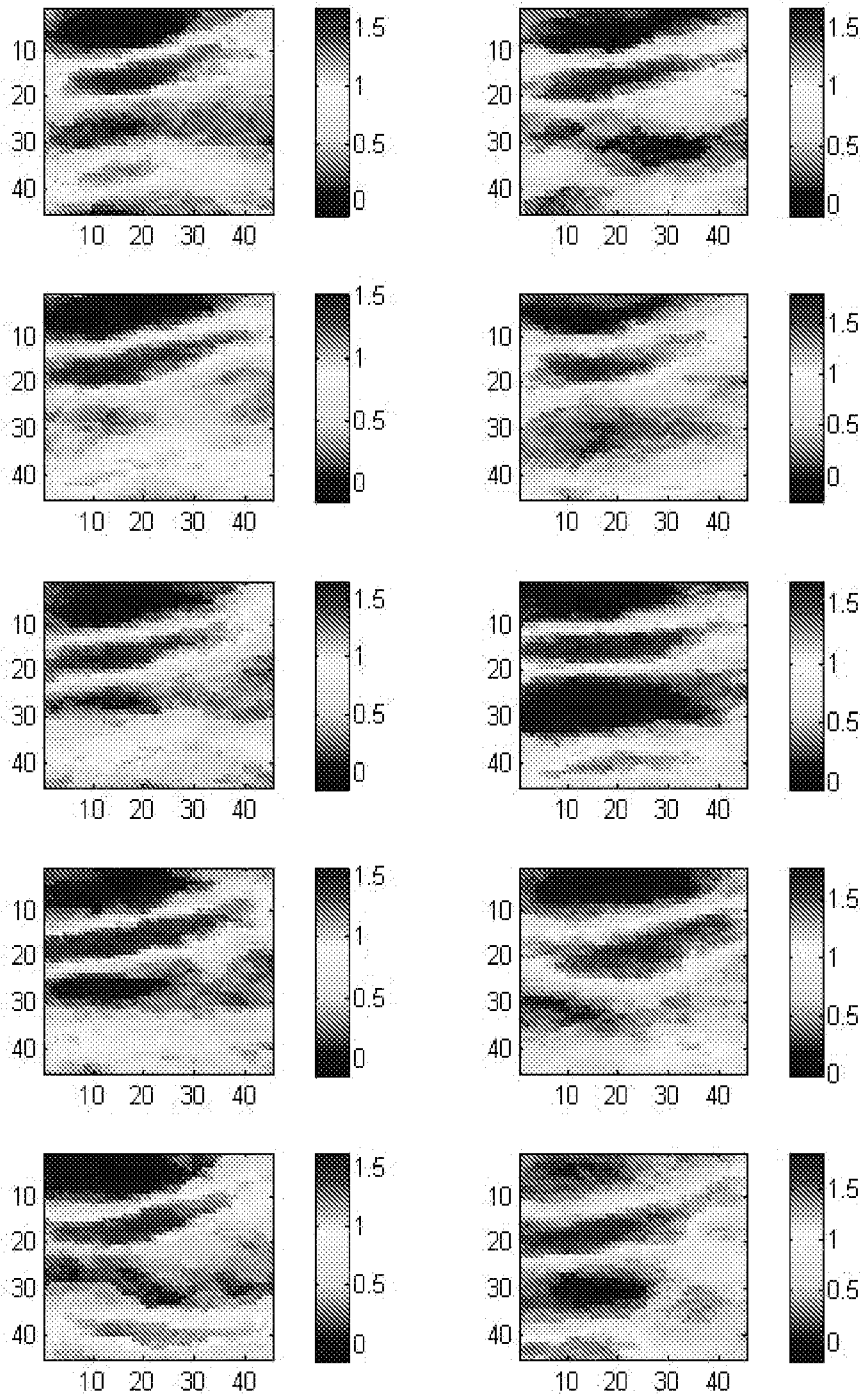
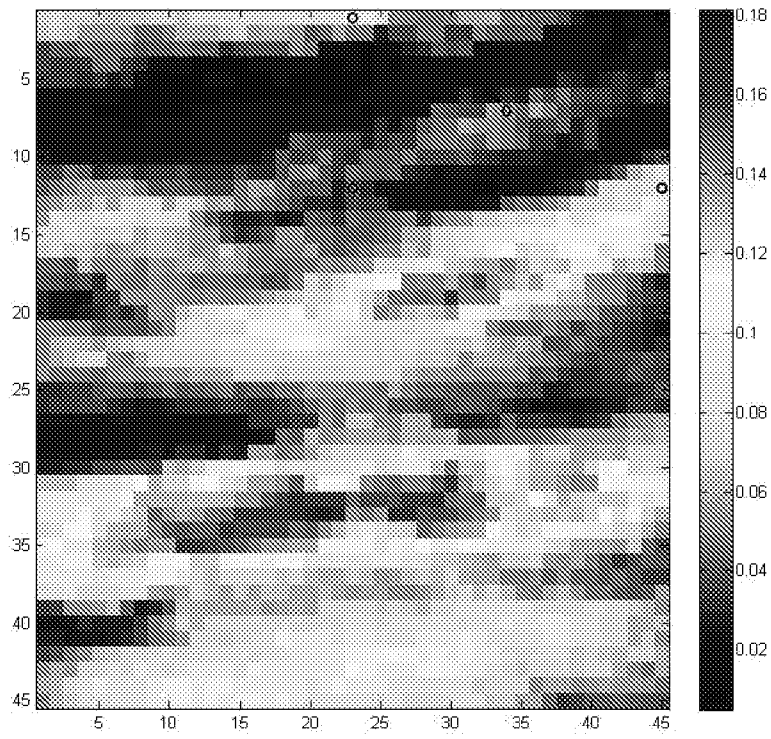


FIG. 31



**FIG. 32**



*FIG. 33*

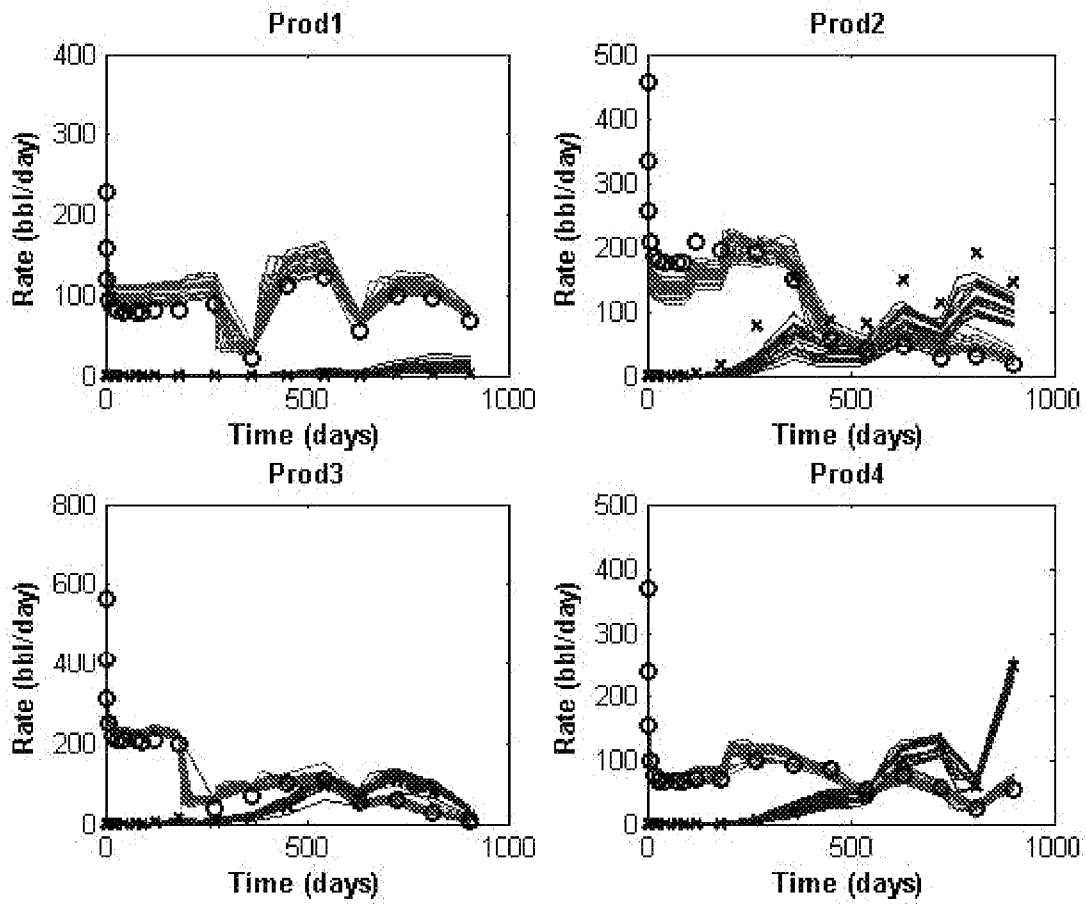
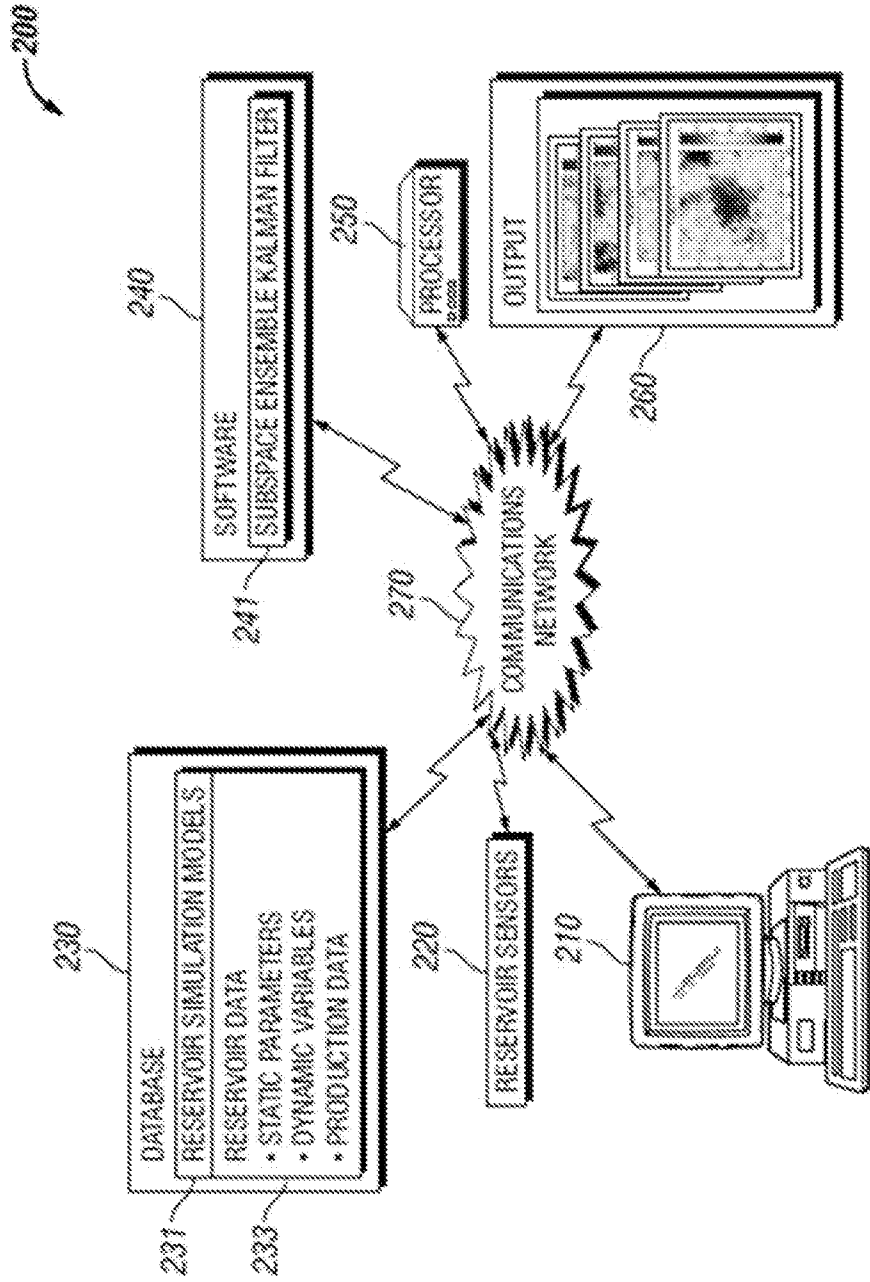


FIG. 34



**FIG. 35**