Computational Geosciences (2006) 10:37–60
DOI: 10.1007/s10596-005-9010-6

Waterflooding using closed-loop control

Geir Nævdal[a], D. Roald Brouwer[b] and Jan-Dirk Jansen[b,c]

[a]*RF-Rogaland Research, Thormøhlensgate 55, N-5008 Bergen, Norway.*
E-mail: geir.naevdal@rf.no
[b]*Exploratory Research, Shell International Exploration and Production,*
PO Box 60, 2280 AB Rijswijk, The Netherlands.
E-mail: roald.bouwer@shell.com
[c]*Department of Geotechnology, Delft University of Technology,*
PO Box 5028, 2600 GA, Delft, The Netherlands.
E-mail: j.d.jansen@citg.tudelft.nl

Accepted 8 August 2005

To fully exploit the possibilities of "smart" wells containing both measurement and control equipment, one can envision a system where the measurements are used for frequent updating of a reservoir model, and an optimal control strategy is computed based on this continuously updated model. We developed such a closed-loop control approach using an ensemble Kalman filter to obtain frequent updates of a reservoir model. Based on the most recent update of the reservoir model, the optimal control strategy is computed with the aid of an adjoint formulation. The objective is to maximize the economic value over the life of the reservoir. We demonstrate the methodology on a simple waterflooding example using one injector and one producer, each equipped with several individually controllable inflow control valves (ICVs). The parameters (permeabilities) and dynamic states (pressures and saturations) of the reservoir model are updated from pressure measurements in the wells. The control of the ICVs is rate-constrained, but the methodology is also applicable to a pressure-constrained situation. Furthermore, the methodology is not restricted to use with "smart" wells with down-hole control, but could also be used for flooding control with conventional wells, provided the wells are equipped with controllable chokes and with sensors for measurement of (wellhead or down hole) pressures and total flow rates. As the ensemble Kalman filter is a Monte Carlo approach, the final results will vary for each run. We studied the robustness of the methodology, starting from different initial ensembles. Moreover, we made a comparison of a case with low measurement noise to one with significantly higher measurement noise. In all examples considered, the resulting ultimate recovery was significantly higher than for the case of waterflooding using conventional wells. Furthermore, the results obtained using closed-loop control, starting from an unknown permeability field, were almost as good as those obtained assuming a priori knowledge of the permeability field.

Keywords: closed loop, real time, control, reservoir management, ensemble Kalman filter, reservoir simulation, permeability estimation, optimal control, adjoint equation, automatic history matching

1. Introduction

In the last few years, the need to produce cheaper and to produce more oil from a reservoir has resulted in the development of a variety of technologies to better measure and control the production process through the wells. Typically, these technologies are installed within the well and can be operated remotely. A well equipped with this type of technology is generally referred to as a smart, intelligent, or instrumented well. Sensors have been developed for permanent down-hole measurement of, for instance, temperature, pressure, resistivity, fluid composition and acoustic velocities. With these sensors, more and more detailed information on conditions in the well and the near-well region can be obtained than with conventional wells. Apart from measurement equipment, major progress has been made in the development of technology to control the production process. This includes down-hole valves, which enable the splitting up of wells into a number of segments that can be controlled individually and remotely. The information obtained from the down-hole measurements can be used to update the reservoir model. Traditionally, this has been performed by "history matching," where one tries to fit uncertain parameters, such as permeability and porosity, in the reservoir model such that the predicted results approach the measured production data. However, the traditional history matching suffers from a number of drawbacks:

- It is usually only performed on a campaign basis, typically after periods of years.
- The matching techniques are usually ad hoc and involve manual adjustment of model parameters, instead of systematic parameter updating.
- Uncertainties in the state variables, model parameters and measured data are usually not explicitly taken into account.
- The resulting history-matched model often violates essential geological constraints.
- Worst of all, the updated model may reproduce the production data perfectly but have little or no predictive capacity because it may have been overfitted by adjusting a large number of unknown parameters using a much smaller number of measurements.

Various techniques for "automated history matching" have been developed over the past decade to address these issues. A recent approach is to update the reservoir model using an ensemble Kalman filter. This method has been first used to continuously update near-well reservoir models [16,17]. Recent work has shown that it is a promising alternative to "traditional" automatic history-matching approaches [11,12,18].

The outcome of the ensemble Kalman filter is an ensemble of updated models taking the recent measurement into account. The mean of the updated models, which can be considered as the most likely model at the current time, can then be used to compute a strategy for further control of the valves. The injection and production rates that maximize the ultimate oil recovery or the economic value over the life of the reservoir can be computed using optimal control theory. For cases where the reservoir description is known, it has been shown that in certain cases, there may be a significant benefit in using a smart well compared to a conventional well [5,6,20]. In these studies, the optimal rates were computed using a gradient-based optimization routine where the derivative information was obtained through the use of an adjoint equation. These results are in line with those from earlier studies that addressed the optimization of waterflooding using optimal control of conventional wells [3,23,24]. For a review of other applications of optimal control in reservoir simulation, we refer to [5].

Figure 1 shows how reservoir management may be considered as a model-based closed-loop control process [13]. The system we consider is the reservoir with wells and other facilities. With the aid of the sensors, we are able to make observations of (a part of) the state of the reservoir. These measurements are affected by noise. The flow in the reservoir can be managed by controlling the well rates or the bottom hole pressures. The optimal well rates or pressures are calculated using an optimal control algorithm, which, in our case, is based on a model of the system. Usually, the reservoir model is based on information obtained from all sources, including seismics, well logs, well tests, fluid properties and geological prior information. In the closed-loop approach, the model is regularly updated based on information from the production measurements (i.e., the sensors). Here, we use the ensemble Kalman filter for continuous model updating. To compare the performance of a closed-loop system using smart wells with an approach using conventional wells, it is convenient to use a synthetic reservoir model, as it is difficult to conduct such a comparison using a real reservoir. Such a comparison was first presented in [7]. In a follow-up study [19], the

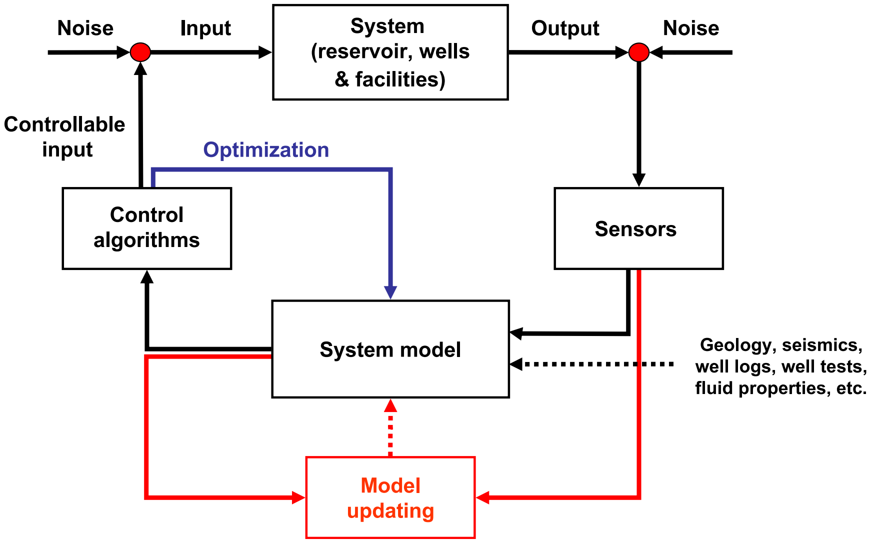
Noise
Input
System (reservoir, wells & facilities)
Output
Noise
Controllable input
Optimization
Control algorithms
Sensors
System model
Geology, seismics, well logs, well tests, fluid properties, etc.
Model updating

Figure 1. Reservoir management represented as a model-based closed-loop controlled process [7].

system (i.e., the real reservoir) was represented with a 110 × 110 × 1 grid, whereas the system model was simplified to a 10 × 10 × 1 grid. The results of these earlier studies were promising both for optimizing the NPV and estimating the dominating features of the permeability field. Two other studies, using different history-matching techniques and, in one case, also a different optimization technique, confirm the scope for improved reservoir management using a closed-loop approach [21]. One of the shortcomings of our study reported in [7] was that the measurement errors used in pressure were very small. Here, we will study the impact of measurement uncertainties and show that the method is robust with respect to the initial ensemble. We will briefly present the reservoir model in Section 2, the theory of optimal control in Section 3, the ensemble Kalman filter in Section 4 and our results for the examples considered in Section 5.

2. The reservoir model

In this study, we restrict ourselves to a very simple horizontal, square, two-dimensional, two-phase (oil–water) reservoir, as depicted in figure 2, which we also used in earlier studies [6,7]. The reservoir is water-flooded with two horizontal wells, each divided into a large number of segments of which the flow rate can be individually controlled with inflow control valves (ICVs). Since only one phase (water) is injected, the injection rates can be controlled completely. For the producers, the total rate is controlled, but the produced fluid composition is determined by the phase mobilities directly around the well. We assume that both pressures and liquid rates can be measured. Alternatively, the configuration can be interpreted as two rows of conventional vertical wells, with controllable surface chokes and measurements of pressures and phase rates. Although highly idealized, we believe that this model reflects the essential features required to demonstrate the scope for closed-loop reservoir management. The large number of ICVs in each well is, at present, unrealistic, but our earlier work on open-loop optimization [6] indicates that a much smaller number may be sufficient to achieve efficient dynamic control of the reservoir flow. In most circumstances, also the assumption of unconstrained rate control in the segments is not realistic because lift constraints or injection pressure constraints may necessitate the control of pressures instead of rates, or a combination of both. However, as demonstrated earlier, also when more realistic well constraints are employed, there remains significant scope for waterflooding optimization [5,6,20]. In all cases, the scope depends on the reservoir heterogeneity and the economic assumptions that we will address below.

We used a simple reservoir simulator, identical to the one described in detail in [5] and [6]. The simulator is based on the usual equations for conservation of mass and Darcy's law as applied to oil and water and the black-oil model to represent the fluid properties [4]. It uses a standard five-point finite difference discretization in space and a semi-implicit Euler discretization in time. We note that the results of our study are

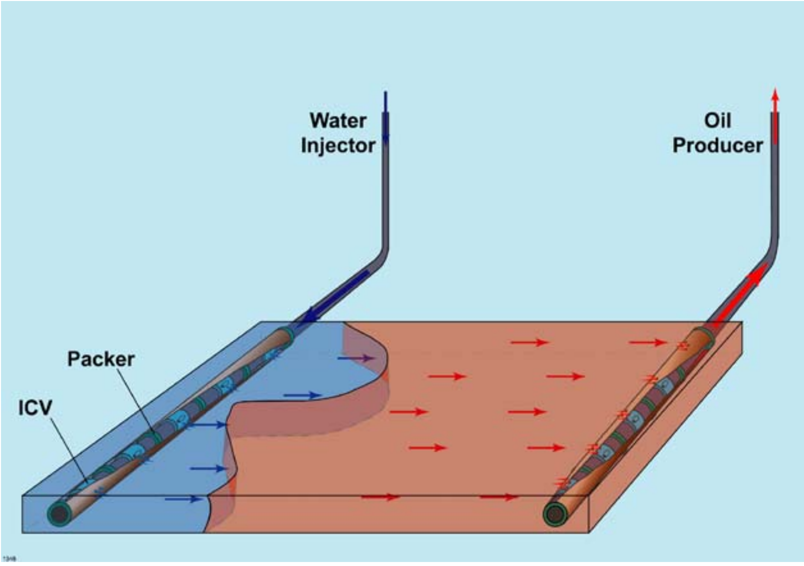

Figure 2. Reservoir model [6].

not dependent on the particular choice of discretization, although the implementation of adjoint-based optimal control is easier when using a fully implicit formulation [20]. We did not take into account pressure drop along the horizontal wells to simplify the analysis. This assumption is justified for wells with a low to medium flow rate and medium to high drawdowns. Moreover, the assumption is not of importance to our objective to demonstrate the scope for closed-loop reservoir management, which does not rely on the use of horizontal "smart wells," but could equally well be obtained by the "smart" use of vertical wells [7].

3. Optimal control of the waterflooding process

To optimize the waterflooding process, we maximize an objective function J, defined as a simple net present value (NPV):

$$J = \sum_{k=0}^{K-1} \sum_{n=1}^{N_{pr}} \frac{-r_w q_{w,n}(k) - r_o q_{o,n}(k)}{(1+b)^a} \Delta t(k). \qquad (1)$$

Here, k is the time step counter, K is the total number of time steps in the simulation, n is the well segment counter, N_{pr} is the number of well segments in the producer, r_o is the oil price, r_w is the cost of water production, $q_{o,n}(k)$ and $q_{w,n}(k)$ are the oil and water rates at surface conditions at time step k in segment n, respectively, taken negative for production, $\Delta t(k)$ is the time step size, b is the discount rate, expressed as a fraction per year, and a is the number of years passed since the start of production. As a simplification, we ignored the cost of injecting water since, in our examples, the water injection rates are kept constant.

As the objective function (1) indicates, we solve the optimal control problem in a discretized version, i.e., after discretization of the reservoir model. The discretized version of the reservoir model may be represented as a discrete dynamic system

$$\mathbf{g}[\mathbf{x}(k+1), \mathbf{x}(k), \mathbf{u}(k)] = \mathbf{0}, \tag{2}$$

where $\mathbf{x}$ denotes the state vector (containing pressures and saturations in each grid block), $\mathbf{u}$ is the control term (the injection and production rates in the grid blocks penetrated by a well), and $\mathbf{g}$ is a vector-valued function representing the reservoir simulator. We will denote the initial state with $\mathbf{x}_0$. Equation (2) is implicit in time, but we remark that the derivation of the optimal control formulation may equally well be obtained using an explicit formulation. Moreover, the theory of optimal control is quite analogous in the discrete and continuous time versions [15].

According to equation (1), the NPV at each time step k can be computed from the state vector $\mathbf{x}$ and control vector $\mathbf{u}$. In our application, $\mathbf{u}$ consists of the total rates in the producer segments and the water rates in the injector segments. The oil and water rates in the producer segments, as needed in equation (1), depend on the water saturation of the corresponding grid blocks. This implies that $q_o(k)$ and $q_w(k)$ in equation (1) are functions of both $\mathbf{u}$ and $\mathbf{x}$. The terms r_o and r_w are constants, and the remaining terms depend only on the time index k. This means that the objective function (1) can be written in the form

$$J = \sum_{k=0}^{K-1} J_k[\mathbf{x}(k), \mathbf{u}(k)], \tag{3}$$

where J_k represents the contribution to J in each time step. Constraints can be expressed in terms of the state variables or the input variables and may be equality or inequality constraints, which we represent in a general form as

$$\mathbf{c}[\mathbf{x}(k), \mathbf{u}(k)] \leq \mathbf{0}. \tag{4}$$

In our implementation, we impose the constraint that the total injection rate (i.e., the sum over all segments in the injector) is equal to the total production rate and remains constant over time.

The control problem can now be formulated as finding the control vector $\mathbf{u}(k)$ that maximizes J over the time interval $k = 0, \ldots, K-1$, subject to system equations (2), initial conditions $\mathbf{x}_0$, and constraints (4). Many numerical techniques are available to solve this optimization problem and to find an optimum solution $\mathbf{u}(k)$, $k = 0, \ldots, K-1$. We use a steepest descend technique, as described in more detail in [5], which requires the sensitivities of the objective function, i.e., the changes δJ_{ik} in J caused by perturbations $\delta u_i(k)$ of the input, to guide the iteration process. Here, $u_i(k)$ is a single element i of vector $\mathbf{u}(k)$ at time k. An efficient way to compute the sensitivities is through the use of an adjoint model. The adjoint model can be found using standard theory of optimal control [15,22].

The derivation of the adjoint model is closely connected to the classic way to solve constrained optimization problems, through the use of Lagrange multipliers (see, e.g., [2]). Setting aside, for the moment, the "ordinary constraints" $\mathbf{c}$, and only addressing the "system constraints" $\mathbf{g}$, we can define a modified objective function

$$\overline{J} = \sum_{k=0}^{K-1} \left\{ J_k[\mathbf{x}(k), \mathbf{u}(k)] + \boldsymbol{\lambda}(k+1)^T \mathbf{g}[\mathbf{x}(k+1), \mathbf{x}(k), \mathbf{u}(k)] \right\}, \tag{5}$$

where the constraint $\mathbf{g}$ has been added to J_k with the aid of a vector of Lagrange multipliers $\boldsymbol{\lambda}$. Next, we define an auxiliary function, the Lagrangian:

$$\begin{aligned} \mathcal{L}(k) \equiv \mathcal{L}[\mathbf{x}(k+1), \mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\lambda}(k+1)] \overset{\Delta}{=} \\ J_k[\mathbf{x}(k), \mathbf{u}(k)] + \boldsymbol{\lambda}(k+1)^T \mathbf{g}[\mathbf{x}(k+1), \mathbf{x}(k), \mathbf{u}(k)], \end{aligned} \tag{6}$$

with which we can rewrite equation (5) as

$$\overline{J} = \sum_{k=0}^{K-1} \mathcal{L}[\mathbf{x}(k+1), \mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\lambda}(k+1)]. \tag{7}$$

We can obtain a first-order description of the effect of changing $\mathbf{u}$ on the magnitude of $\overline{J}$, through taking the first variation of equation (7):

$$\begin{aligned} \delta\overline{J} = \sum_{k=0}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k)}\right] \delta\mathbf{x}(k) + \sum_{k=0}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k+1)}\right] \delta\mathbf{x}(k+1) + \sum_{k=0}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{u}(k)}\right] \delta\mathbf{u}(k) \\ + \sum_{k=0}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\boldsymbol{\lambda}(k+1)}\right] \delta\boldsymbol{\lambda}(k+1). \end{aligned} \tag{8}$$

By changing the index of summation, the first term at the right-hand side of equation (8) can be rewritten as

$$\sum_{k=0}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k)}\right] \delta\mathbf{x}(k) = \left[\frac{\partial\mathcal{L}(0)}{\partial\mathbf{x}(0)}\right] \delta\mathbf{x}(0) + \sum_{k=1}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k)}\right] \delta\mathbf{x}(k). \tag{9}$$

while the second term in equation (8) can be rewritten as

$$\begin{aligned} \sum_{k=0}^{K-1} \left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k+1)}\right] \delta\mathbf{x}(k+1) = \sum_{k=1}^{K} \left[\frac{\partial\mathcal{L}(k-1)}{\partial\mathbf{x}(k)}\right] \delta\mathbf{x}(k) = \\ \sum_{k=1}^{K-1} \left[\frac{\partial\mathcal{L}(k-1)}{\partial\mathbf{x}(k)}\right] \delta\mathbf{x}(k) + \left[\frac{\partial\mathcal{L}(K-1)}{\partial\mathbf{x}(K)}\right] \delta\mathbf{x}(K), \end{aligned} \tag{10}$$

Substitution of results (9) and (10) in equation (8) and reordering the results gives

$$\delta\overline{J} = \left[\frac{\partial\mathcal{L}(0)}{\partial\mathbf{x}(0)}\right]\delta\mathbf{x}(0) + \sum_{k=1}^{K-1}\left[\frac{\partial\mathcal{L}(k-1)}{\partial\mathbf{x}(k)} + \frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k)}\right]\delta\mathbf{x}(k) + \left[\frac{\partial\mathcal{L}(K-1)}{\partial\mathbf{x}(K)}\right]\delta\mathbf{x}(K)$$
$$+ \sum_{k=0}^{K-1}\left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{u}(k)}\right]\delta\mathbf{u}(k) + \sum_{k=0}^{K-1}\left[\frac{\partial\mathcal{L}(k)}{\partial\boldsymbol{\lambda}(k+1)}\right]\delta\boldsymbol{\lambda}(k+1). \qquad (11)$$

The first term in equation (11) is zero because a change in the input vector does not affect the initial conditions, i.e., $\delta\mathbf{x}(\mathbf{0}) = \mathbf{0}$. The last term also vanishes because $\partial\mathcal{L}(k)/\partial\boldsymbol{\lambda}(k+1) = \mathbf{g}(k) = \mathbf{0}$. A further simplification of equation (11) can be obtained through putting a restriction on the Lagrange multipliers as follows. The third term in equation (11) becomes zero if we impose that

$$\frac{\partial\mathcal{L}(K-1)}{\partial\mathbf{x}(K)} = \mathbf{0}, \qquad (12)$$

which, after substitution of equation (7), can be reduced to the condition

$$\boldsymbol{\lambda}(K)^T = \mathbf{0}^T. \qquad (13)$$

The second term becomes zero if for every time step k

$$\frac{\partial\mathcal{L}(k-1)}{\partial\mathbf{x}(k)} + \frac{\partial\mathcal{L}(k)}{\partial\mathbf{x}(k)} = \mathbf{0}, \qquad (14)$$

which, after substitution of equation (6), can also be written as

$$\boldsymbol{\lambda}(k)^T\frac{\partial\mathbf{g}(k-1)}{\partial\mathbf{x}(k)} + \frac{\partial J_k(k)}{\partial\mathbf{x}(k)} + \boldsymbol{\lambda}(k+1)^T\frac{\partial\mathbf{g}(k)}{\partial\mathbf{x}(k)} = \mathbf{0}, \qquad (15)$$

or

$$\boldsymbol{\lambda}(k)^T = -\left[\boldsymbol{\lambda}(k+1)^T\frac{\partial\mathbf{g}(k)}{\partial\mathbf{x}(k)} + \frac{\partial J_k(k)}{\partial\mathbf{x}(k)}\right]\left[\frac{\partial\mathbf{g}(k-1)}{\partial\mathbf{x}(k)}\right]^{-1}. \qquad (16)$$

Equation (16) is known as the *adjoint* or *costate equation*. Starting from final condition (13), it can be integrated backward in time. With these simplifications, equation (11) can now be written as:

$$\delta\overline{J} = \sum_{k=0}^{K-1}\left[\frac{\partial\mathcal{L}(k)}{\partial\mathbf{u}(k)}\right]\delta\mathbf{u}(k) = \sum_{k=0}^{K-1}\left[\frac{\partial J_k(k)}{\partial\mathbf{u}(k)} + \boldsymbol{\lambda}(k+1)^T\frac{\partial\mathbf{g}(k)}{\partial\mathbf{u}(k)}\right]\delta\mathbf{u}(k). \qquad (17)$$

Equation (17) represents the first-order change in the objective function resulting from a change in the input vector $\mathbf{u}(k)$. With the aid of the adjoint formulation, we obtain all sensitivities in the form of the matrix $\partial\bar{J}/\partial\mathbf{u}(k) = \sum_{k=0}^{K-1} \partial\mathcal{L}(k)/\partial\mathbf{u}(k)$, through a single forward simulation of the system equations (2) and a subsequent single "backward" simulation of the adjoint equations (16). In case of an optimum, these first-order variations are zero if the controls are unconstrained.

The optimization problem is now transferred to a *two-point boundary problem*, because initial condition for the dynamic system and the adjoint are specified at opposite sides of the time interval (cf., e.g., [22]). Solution of the optimization problem consists of repeating the following steps until the optimal control vector $\mathbf{u}(k)$ has been found for each time step:

1. Numerical simulation of the dynamic system behavior by solution of equation (2) from time interval 0 to $K - 1$, with the current estimate of the optimal control vector $\mathbf{u}(k)$, using $\mathbf{x}(0)$ as initial conditions. This corresponds to running the reservoir simulator with given injection and production rates. At the first iteration, one must start with an initial choice of the optimal control vector $\mathbf{u}(k)$. As the method is gradient-based and local optima may exist, the selection of the initial $\mathbf{u}(k)$ may influence the final solution. The initial $\mathbf{u}(k)$ would, if possible, be selected using some ideas about the optimal solution. In our case, we selected the initial rates to be equal for each of the zones, which corresponds to a "naive" optimization approach as will be discussed below. While the optimal control algorithm is run in a closed-loop setting, the initial choice of $\mathbf{u}(k)$ is selected as the solution obtained at the previous time step.
2. Evaluation of the objective function (1).
3. Calculation of the Lagrange multipliers by backward numerical solution of the adjoint equation (16), starting from the final condition (13).
4. Computation of the gradients $\frac{\partial\mathcal{L}(k)}{\partial\mathbf{u}(k)} = \frac{\partial J_k(k)}{\partial\mathbf{u}(k)} + \boldsymbol{\lambda}(k+1)^T \frac{\partial\mathbf{g}(k)}{\partial\mathbf{u}(k)}$.
5. Computation of an improved control vector $\mathbf{u}$, using the gradients $\partial\mathcal{L}(k)/\partial\mathbf{u}(k)$. This is performed in our case by a steepest descent-based method. In addition, some actions must be taken to meet the constraints on the input. In our case, this corresponds to constraining the total production and injection rates. The constraints may be incorporated as additional terms in the modified objective function (5), or they can be treated in a more ad hoc manner. In our implementation, we used an ad hoc solution (see [5] and [6]). For a more formal treatment of the constraint problem, see [20].

Because the process is gradient-based, a local optimum may be computed and the results may depend on the initial choice of $\mathbf{u}(k)$. For more details on the implementation and references to related literature, see [5].

4. Ensemble Kalman filter and estimation of permeability fields

The Kalman filter was initially developed for discrete-data linear filtering problems [14], but was later extended for use with nonlinear systems by introduction of the *extended Kalman filter* (see, e.g., [22]). In the Kalman filtering approach, measurements are used as soon as they become available to estimate the state of the process. For models with uncertain parameters, adaptive filters are available, in which the uncertain parameters are estimated together with the states. This is performed through a straightforward extension of the Kalman filter [22], namely, through augmenting the state vector with the model parameters that should be estimated. The inclusion of model parameters in the state introduces nonlinearities in the estimation process, which puts additional challenges to the implementation of the filter. For an introduction to the Kalman filter, see, e.g., [22].

The extended Kalman filter has shortcomings both when applied to large-scale models and strong nonlinearities. A more recent approach that has been applied successfully to large-scale nonlinear models within oceanographic sciences is the ensemble Kalman filter, first introduced in [9]. A recent overview of the ensemble Kalman filter and its applications within atmospheric and oceanographic sciences can be found in [10]. The first use of the ensemble Kalman filter for updating of the permeability field of a reservoir model was reported in [16]. Further work in this direction has been reported in [17] and [18]. In [11] and [12], both the porosity and permeability of the reservoir were estimated using the ensemble Kalman filter.

The filter can be divided in two steps, a forecast step and an analysis step. In the forecast step, the state of the system is updated by solving the reservoir model as represented by equation (2) numerically. In the analysis step, the state of the model is updated to take measurements into account. The state vector after running the forecast step is denoted by $\widehat{\mathbf{x}}(\kappa)^f$, and the state vector after the assimilation step by $\widehat{\mathbf{x}}(\kappa)^a$. Here, the index κ refers to the specific time steps when the measurements are assimilated. The "extended" state vector $\widehat{\mathbf{x}}$ is an augmented version of $\mathbf{x}$, to which a vector $\mathbf{k}$, reflecting the permeabilities of all simulator grid blocks, is added, i.e.;

$$\widehat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{k} \end{bmatrix}. \tag{18}$$

The ensemble Kalman filter is based on a Monte Carlo approach, using an ensemble of models to represent the necessary statistics. Based on experience from previous work [16–18], we use 100 members in the ensemble. In [12], it was found that only 40 ensemble members were sufficient to get reasonable results. The influence of the size of the ensemble is a topic for further research.

The forecast step consists of running the model (i.e., the reservoir simulator) from the current time to the next point in time where measurements are going to be assimilated for each member of the ensemble. The filter is initiated by generating an initial ensemble. In our application, we assume that the initial pressure and saturations are in a known equilibrium state, but we generate different permeability fields for each of the ensemble members using a mixture distribution, as described in [18]. The permeability fields are generated by first drawing a correlation length l (expressed in grid block lengths, assuming square grid blocks) from a normal distribution $N(\mu_l, \sigma_l^2)$, where μ_l is a specified mean correlation length and σ_1^2 is the variance of the correlation length. After generating the correlation length, a covariance matrix $\mathbf{Q}$ is generated with entries Q_{pq} defined as

$$Q_{pq} = \sigma_k^2 \exp\left[-\frac{\left(i_p - i_q\right)^2 + \left(j_p - j_q\right)^2}{l} \right], \tag{19}$$

where i_p and j_p are the grid block coordinates of grid block p, with similar definitions for grid block q, and σ_k^2 is the variance of the permeability field. The permeability of an ensemble member is then obtained by drawing from a multinormal distribution, $N(\mu_k \mathbf{1}, \mathbf{Q})$, where μ_k is the initial mean of the permeability (identical for all the grid blocks), and $\mathbf{1}$ is a unit vector with a length equal to the number of grid blocks used in the reservoir simulator.

The assimilated state at time κ is computed by taking into account the measurement vector $\mathbf{y}(\kappa)$. We assume that there is a linear relationship between the measurements $\mathbf{y}(\kappa)$ and the states $\mathbf{x}(\kappa)$, as expressed by the equation

$$\mathbf{y}(\kappa) = \mathbf{C}\widehat{\mathbf{x}}(\kappa). \tag{20}$$

Equation (20) refers to an idealized situation without any measurement errors. We assume that the measurement errors can be expressed by a Gaussian random variable with zero mean and covariance $\mathbf{R}$. If there is a nonlinear relationship between the states and the measurements, modifications are required. A possible modification is to extend the state vector with the measurements that have a nonlinear relationship to the (original) states. This was performed, for instance, in [18].

As pointed out in [8], it is necessary to add noise to the actual measurement for proper error propagation in the ensemble Kalman filter. This is performed by letting the actual measurement $\mathbf{y}(\kappa)$ serve as a reference observation and generating an ensemble of observations. For each member of the ensemble, a measurement vector $\mathbf{y}(\kappa)$ is generated randomly as $\mathbf{y}(\kappa, i) = \mathbf{y}(\kappa) + \mathbf{e}(\kappa, i)$, where $\mathbf{e}(\kappa, i) \sim N(\mathbf{0}, \mathbf{R})$.

To apply the Kalman filter, the error covariance matrix $\mathbf{P}(\kappa)$ for the model is needed. In the Kalman filter, this is defined in terms of the true state $\widehat{\mathbf{x}}(\kappa)^t$ as the expectation

$$\mathbf{P}(\kappa)^t = E\left\{ \left[\widehat{\mathbf{x}}(\kappa)^f - \widehat{\mathbf{x}}(\kappa)^t\right] \left[\widehat{\mathbf{x}}(\kappa)^f - \widehat{\mathbf{x}}(\kappa)^t\right]^T \right\}. \tag{21}$$

Since the true state is not known, we approximate the true state by the mean of the ensemble:

$$\widehat{\mathbf{x}}(\kappa)^t \approx \overline{\widehat{\mathbf{x}}(\kappa)^f} = \frac{1}{N_{ens}} \sum_{i=1}^{N_{ens}} \widehat{\mathbf{x}}(\kappa, i)^f, \tag{22}$$

where N_{ens} is the size of the ensemble, and $\widehat{\mathbf{x}}(\kappa, i)^f$ indicates ensemble i at time κ. With this approximation of the true state, an approximation of a left factor of the error covariance of the model is

$$\mathbf{L}(\kappa)^f = \frac{1}{\sqrt{N_{ens}-1}} \left(\left[\widehat{\mathbf{x}}(\kappa, 1)^f - \overline{\widehat{\mathbf{x}}(\kappa)^f} \right] \left[\widehat{\mathbf{x}}(\kappa, 2)^f - \overline{\widehat{\mathbf{x}}(\kappa)^f} \right] \right. \\ \left. \ldots \left[\widehat{\mathbf{x}}(\kappa, N_{ens})^f - \overline{\widehat{\mathbf{x}}(\kappa)^f} \right] \right). \tag{23}$$

The approximation of the model error covariance then becomes

$$\mathbf{P}(\kappa)^f = \mathbf{L}(\kappa)^f \mathbf{L}^{\mathrm{T}}(\kappa)^f. \tag{24}$$

The analyzed state of each member of the ensemble is now computed as

$$\widehat{\mathbf{x}}(\kappa, i)^a = \widehat{\mathbf{x}}(\kappa, i)^{\mathrm{f}} + \mathbf{K}\left[\mathbf{y}(\kappa, i) - \mathbf{C}\widehat{\mathbf{x}}(\kappa, i)^{\mathrm{f}} \right], \tag{25}$$

where the Kalman gain matrix $\mathbf{K}$ is given by

$$\mathbf{K}(\kappa) = \mathbf{P}(\kappa)^f \mathbf{C}^T \left[\mathbf{C}\mathbf{P}(\kappa)^f \mathbf{C}^{\mathrm{T}} + \mathbf{R} \right]^{-1}. \tag{26}$$

The analyzed error covariance matrix $\mathbf{P}(\kappa)^a$ of the model can be computed along the same lines as $\mathbf{P}(\kappa)^f$. Since the updating of the ensemble is linear, the new estimate of the true state, based on the ensemble after the analysis step, is

$$\widehat{\mathbf{x}}(\kappa)^t \approx \overline{\widehat{\mathbf{x}}(\kappa)^a} + \mathbf{K}\left[\mathbf{y}(\kappa) - \mathbf{C}\overline{\widehat{\mathbf{x}}(\kappa)^f} \right], \tag{27}$$

and the model error covariance matrix after the analysis step is

$$\mathbf{P}(\kappa)^a = [\mathbf{I} - \mathbf{K}(\kappa)\mathbf{C}]\mathbf{P}(\kappa)^f. \tag{28}$$

For the practical implementation of the filter, it is important to bear in mind that while evaluating the Kalman gain matrix, $\mathbf{P}(\kappa)^f$ should be entered factorized as in equation (24), and the products should be evaluated in an order such that the dimensions of temporary matrices are kept as low as possible.

5. Examples

The example used in this study has been used before in [5–7]. It represents a two-dimensional, horizontal, two-phase (oil–water) reservoir modeled using a 45 × 45 grid. The permeability field is shown in figure 3 (left). The total dimension of the reservoir is 450 × 450 × 10 m, and each grid block is 10 × 10 × 10 m. An array of 45 vertical injectors, mimicking one "smart" horizontal injector, is located along the left edge. Similarly, an array of vertical producers, mimicking one "smart" horizontal producer, is located along the right edge. The true permeability field is used to generate production data during the closed-loop approach, but otherwise assumed unknown. The reservoir parameters and PVT properties are given in table 1. The relative permeability functions are linear with minimum and maximum values of 0 and 1, respectively, and further details are given in table 1. We note that the choice of linear relative permeabilities removes a significant nonlinearity from the system equations, and further research is required to assess the effect of more realistic, nonlinear, relative permeability curves. The NPV is evaluated according to equation (1). We assume an oil price of \$80/m^3; the cost of the produced water is \$20/m^3. We use an annual interest rate of zero (b = 0); thus, the results are nondiscounted NPVs, which implies that we are optimizing the ultimate recovery of the reservoir for a given time interval. The restriction to a zero discount rate does not have a major influence on our results, in particular, because we consider rate-constrained controls. For a discussion on the effects of discounting on the optimal control of waterflooding, which are, in particular, relevant for pressure-constrained scenarios, we refer to our earlier publications on open-loop control [5,6].

We compared five different scenarios: three different reference cases and two cases using the closed-loop approach, each with a different level of measurement noise. In all cases, the reservoir was steered on voidage replacement with constant field rates of

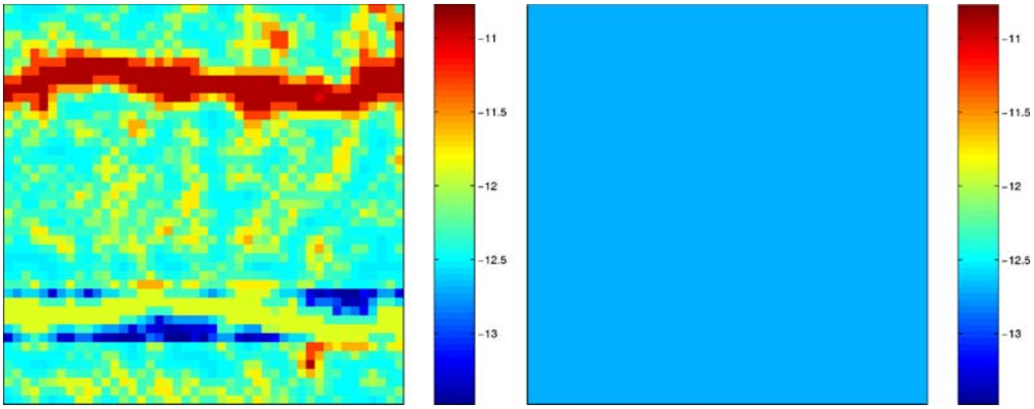

Figure 3. The true permeability (*left*) and initial estimate of permeability (*right*). Permeability units: 10-based log m^2.

Table 1
Reservoir parameters.

Porosity	ϕ	[–]	0.2
Permeability	k	[m^2]	Figure 3
Oil density	ρ_o	[kg/m^3]	1000
Water density	ρ_w	[kg/m^3]	1000
Initial water saturation	$S_{w,ini}$	[–]	0.1
Connate water saturation	S_{wc}	[–]	0.1
Residual oil saturation	S_{or}	[–]	0.1
Capillary pressure function	P_{cow}	[Pa]	0
Oil viscosity	μ_o	[Pa s]	1×10^{-3}
Water viscosity	μ_w	[Pa s]	1×10^{-3}

428.5 m^3/day. The first reference case (the "known" case) represents drainage using optimal control of the well rates, while assuming perfect knowledge of the reservoir. In the second reference case (the "naive" case), we also have smart wells, but we apply a naive control approach where the injection and production rates are identical in all the segments and constant through the entire production history. This control strategy coincides with the optimal control for a homogeneous reservoir. The third reference case (the "conventional" case) is introduced to mimic the use of conventional wells with a constant bottom hole pressure along the well. In this case, the injection/production rates in the well segments are distributed according to the permeabilities of the corresponding grid blocks, and the total rates are chosen to be equal to the rates for the first two cases. (Recall that we have simplified the well model by mimicking one horizontal well with 45 individual wells, one in each perforated segment.)

In both closed-loop examples, we use pressure measurements in each segment of both the injector and producer. We assume two different levels of pressure measurement errors. In Example 1, the standard deviation of the measurement uncertainty for the pressure measurement is 0.02 bar; in Example 2, it is 1 bar. Example 1 corresponds to Example 1 in [7]. Furthermore, we assume that we are able to inject/produce exactly the designed rates from each well segment. Errors in the rate control are implicitly accounted for through the model error, i.e., the uncertainty in the states of the reservoir. While generating the initial ensemble, we assume an initial mean of the permeability of 200 mD $\approx 10^{-13}$ m^2 (see figure 3, right). The permeability fields are generated as described in section 4 (equation 19), with a mean correlation length of 20 grid blocks and a standard deviation of 1 grid block.

The measurements are assimilated at about 2, 5, 7, 9, 12, 23, 46, 69, 93, 116, 174, 231, 289, 347, 405, 463, 521, 579, 637, 694, 752, 810, 868, and 926 days. The total simulation time is 946 days. A new control is computed each time measurements are assimilated. Since we do not believe that the first estimates of the permeability are very accurate, we do not strive for convergence while computing the optimal control. Therefore, we perform only three iterations of the search algorithm described at the end of section 3. As an initial guess, we use the control computed at the previous step, and before the first measurements are assimilated, we control the wells assuming a homogeneous permeability field.

The work flow for all the 12 runs in Examples 1 and 2 is as follows:

1. Generate an initial ensemble for the Kalman filter.
2. Generate the first measurements by running the simulator over the measurement interval with the true permeability and equal production/injection rates in each well segment.
3. Run the reservoir simulator for each of the members of the ensemble Kalman filter over the measurement time interval with the same production/injection rates as used for the true permeability.
4. Update all the ensemble members by taking into account the recent measurements.
5. Compute the mean of the ensemble members and use that as the current estimate of the reservoir. Compute an optimal production/injection strategy by running the reservoir simulator and the adjoint model over the entire reservoir life several times, while iteratively updating the control strategy.
6. Apply the production/injection strategy computed above and run the simulator with the true permeability up to the next point in time where measurements are going to be assimilated, using the dynamic states from the previous run as initial conditions. Save the dynamic states at this point. Return to step 3.

For both the examples, we performed six runs, each time using a new initial ensemble. The initial ensembles, as well as the measurement noise, were generated randomly. In this way, we were able to study the robustness of the approach. The resulting NPVs for each of the examples and some statistics are shown in table 2. Comparing the two examples, one can conclude that there is no significant difference in the average performance. This implies that for our present implementation of the Kalman filter and for the very simple example considered, there is no gain in reducing the measurement uncertainty of the pressure measurement.

Table 2
NPV for each of the six runs of Examples 1 and 2.

Run	Example 1	Example 2
1	\$2.0117 × 10^7	\$2.0355 × 10^7
2	\$2.0073 × 10^7	\$1.9835 × 10^7
3	\$2.0036 × 10^7	\$1.9719 × 10^7
4	\$1.9489 × 10^7	\$1.9668 × 10^7
5	\$1.9338 × 10^7	\$1.9477 × 10^7
6	\$1.8959 × 10^7	\$1.9107 × 10^7
Mean	\$1.9669 × 10^7	\$1.9694 × 10^7
Standard deviation	\$4.7843 × 10^7	\$4.1217 × 10^7

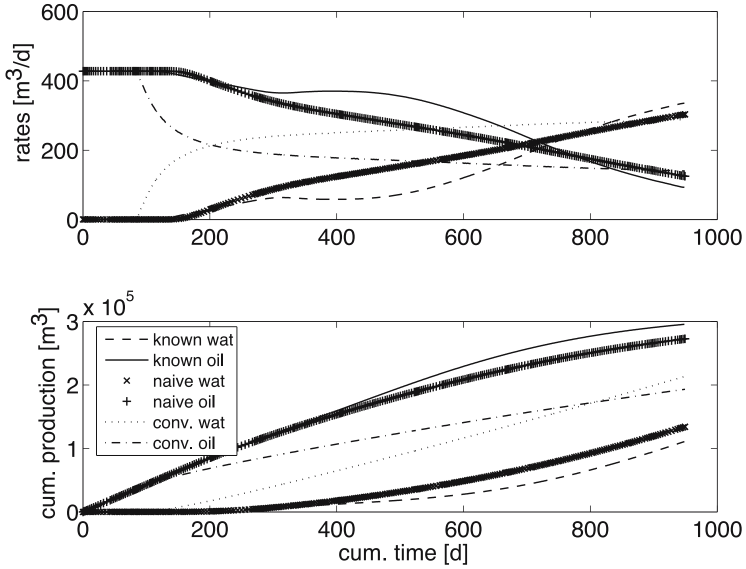

Figure 4. The rates for the three reference cases. The "known" case refers to optimal control based on full knowledge of the true permeability field. The "naive" case refers to optimal control based on a homogeneous permeability field (equal rates in all segments). The "conv." case refers to using conventional horizontal wells (rates approximately proportional to the grid-block permeabilities).

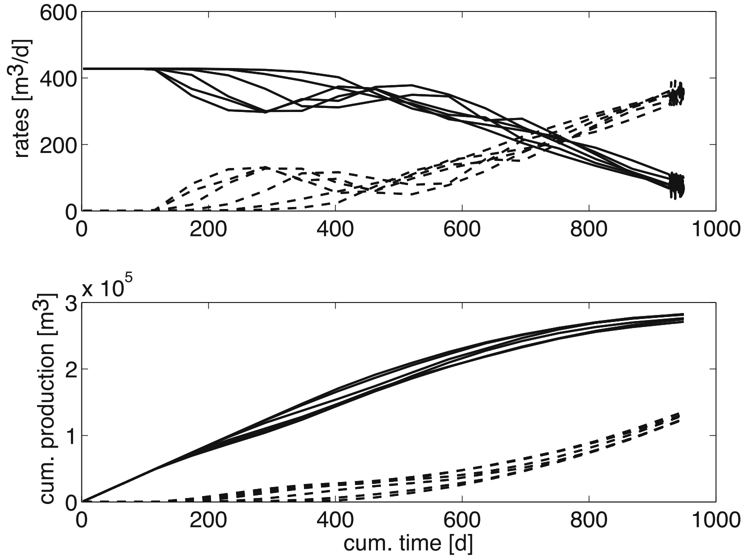

Figure 5. Example 1 (0.02-bar pressure measurement uncertainty): production and cumulative production rates.

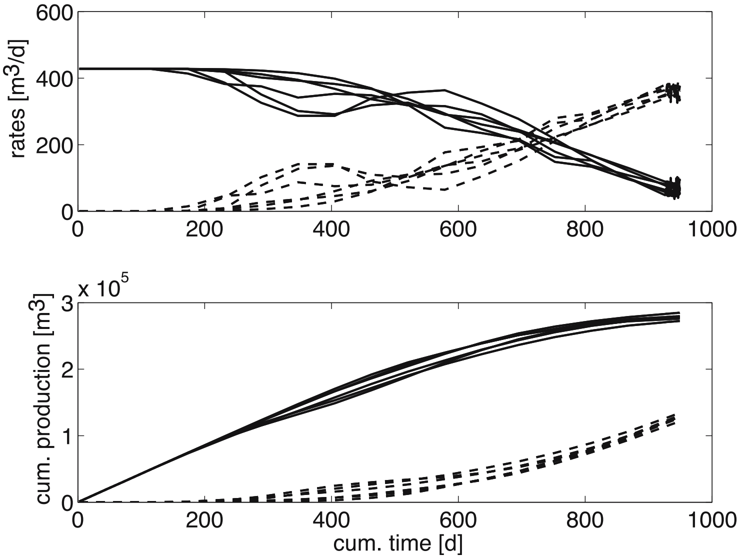

Figure 6. Example 2 (1-bar pressure measurement uncertainty): production and cumulative production rates.

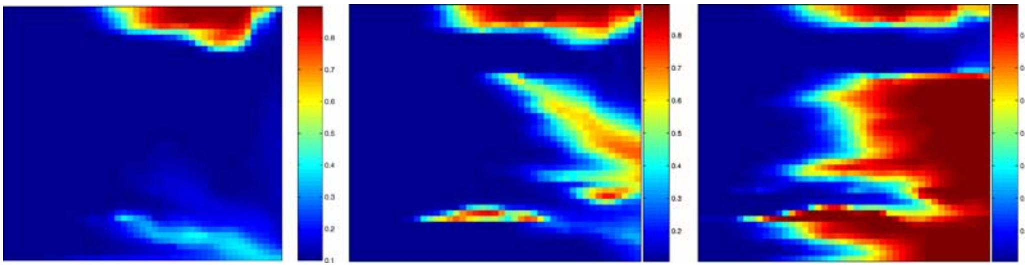

Figure 7. Final oil saturations. *Left*: “known case”; *middle*: “naive case”; *right*: “conventional case.” See also key for figure 4.

The production rates and the cumulative rates for the three reference cases are shown in figure 4. The rates from the six runs from Example 1 are shown in figure 5 and those from Example 2 are shown in figure 6. The results of the closed-loop approach are generally better than those for the naive case (i.e., assuming a homogeneous field). This conclusion is based on comparing the mean NPVs of the closed-loop runs with those of the corresponding naive cases. However, there is a stochastic element in the closed-loop results, and in one of the six closed-loop runs, the NPV turned out to be lower than the NPV for the naive reference case. Furthermore, we did consider neither the costs of implementing the closed-loop approach nor the reliability aspects of the required hardware. The performance of our closed-loop approach depends to a large extent on the performance of the ensemble

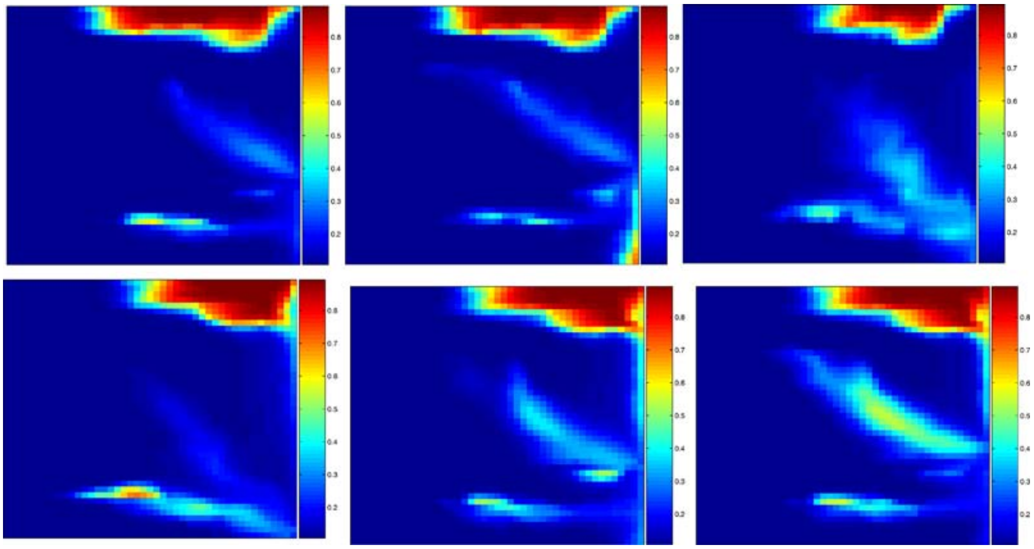

Figure 8. Example 1: water saturation after 949 days. In the *upper row* from *left* to *right* are runs 1, 2, and 3; in the *lower row* are runs 4, 5, and 6.

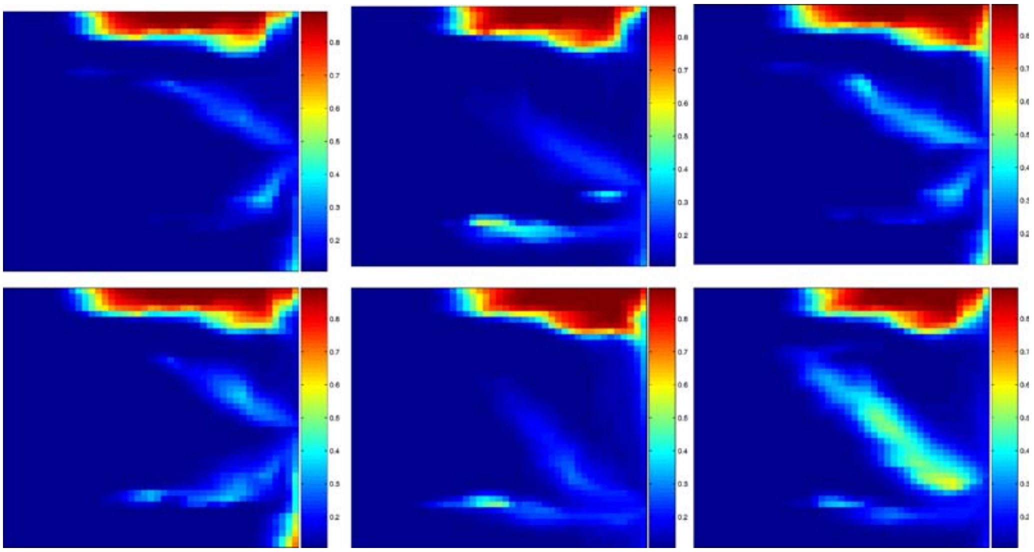

Figure 9. Example 2: water saturation after 949 days. In the *upper row* from *left* to *right* are runs 1, 2, and 3; in the *lower row* are runs 4, 5, and 6.

Kalman filter, and improvements of the filter and making it suitable for use in real fields are important topics of further research.

To further illustrate the performance of the closed-loop approach, we include some figures. In figure 7, we show the final oil saturation for the known, the naive,

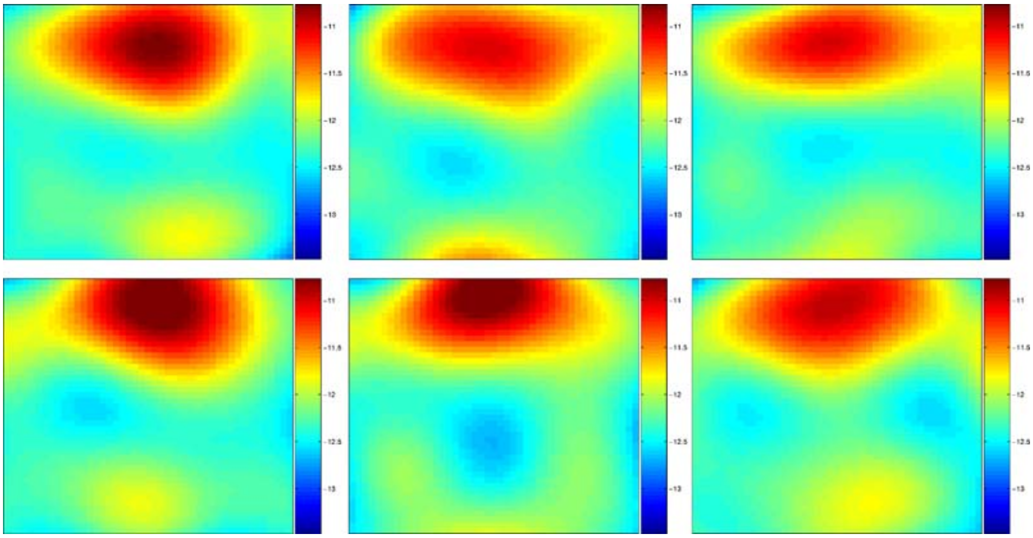

Figure 10. Example 1: estimated permeabilities after 949 days. In the *upper row* from *left* to *right* are runs 1, 2, and 3; in the *lower row* are runs 4, 5, and 6. Permeability units: 10-based log m^2.

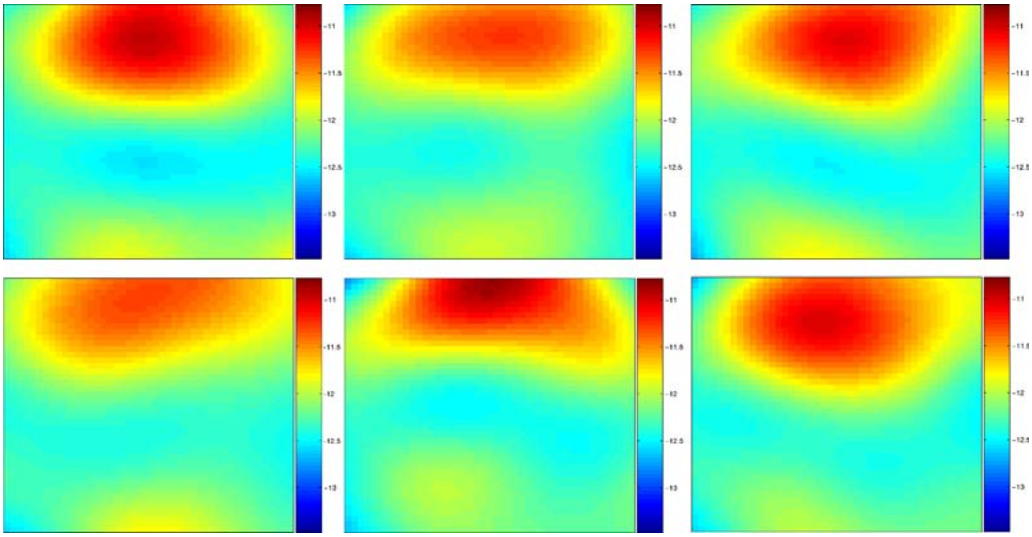

Figure 11. Example 2: estimated permeabilities after 949 days. In the *upper row* from *left* to *right* are runs 1, 2, and 3; in the *lower row* are runs 4, 5, and 6. Permeability units: 10-based log m^2.

and the conventional cases. This should be compared to the final oil saturations using closed-loop control presented in figures 8 and 9. In figures 10 and 11, we show the final estimates of the permeabilities for the six runs of each of the examples. Large-scale trends are identified, but small-scale variation is not matched. The selection of the fields that initiated the ensembles does not contain any small-scale variation, and it is therefore not possible to match such a variation either. (Recall that we use a mean correlation length of 20 grid blocks.) The final permeability estimates using the highest measurement noise are smoother than those obtained with very accurate measurements. This is because of the fact that increasing the measurement noise reduces the magnitude of the updates; that is, in the updating step of the Kalman filter, more weight will be put on the a priori assumptions of the model. Here we assume a constant permeability as the initial mean.

In figures 12 and 13, we show the estimated mean permeability for a run using very accurate measurements (figure 12) and a run using less accurate pressure measurements (figure 13). Here we can see that the permeability is changed much more at an early stage using the accurate measurements. The figures show the first 9 of 18 updates. With reference to the final permeability estimates, shown in the upper right plots in figures 10 and 11, it can be seen that most of the changes in the permeability fields occur during the first nine assimilation steps, i.e., within the first

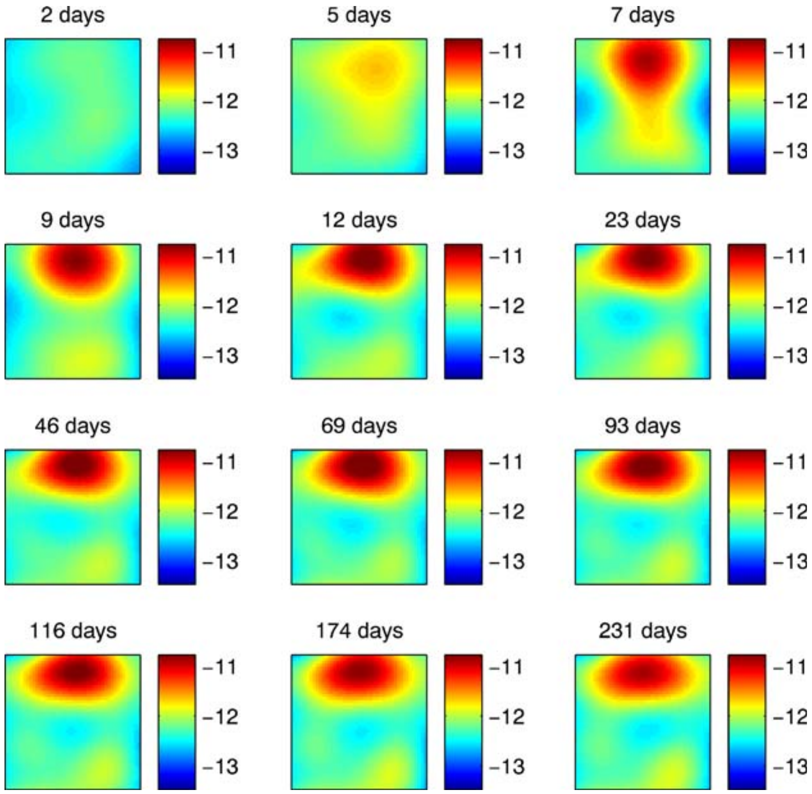

Figure 12. Example 1: estimated permeabilities of run 3. Permeability units: 10-based log m^2.

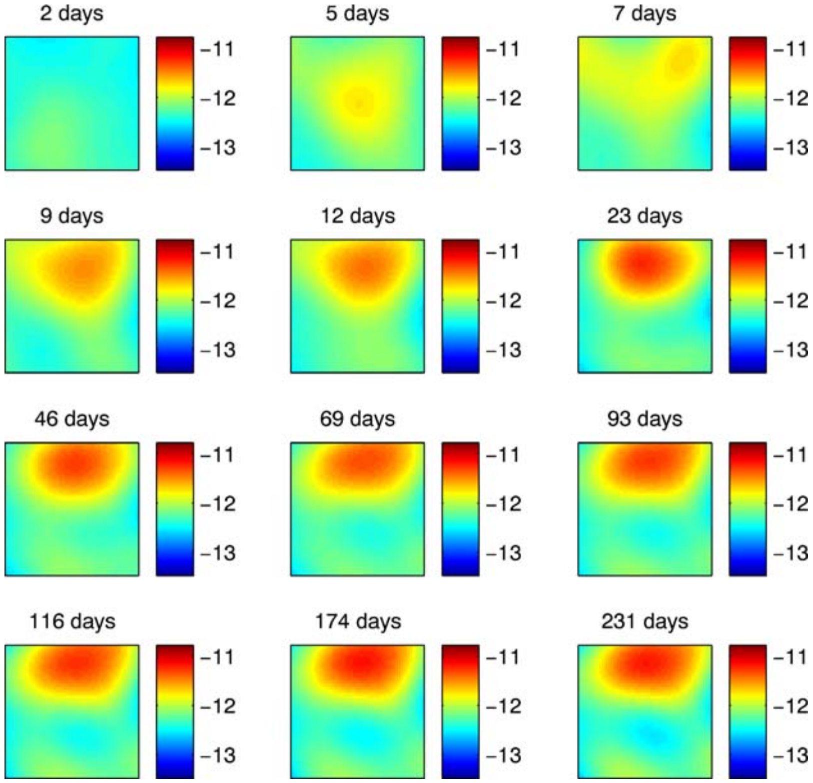

Figure 13. Example 2: estimated permeabilities of run 3. Permeability units: 10-based log m^2.

93 days. Water breakthrough only occurs after approximately 300 days, and the early permeability updates could therefore never have been obtained from phase rate measurements.

6. Conclusion

For the simple examples considered, the use of smart wells gave much better results (i.e., a much higher NPV excluding the added costs of the closed-loop approach) than the use of conventional wells. A closed-loop optimization approach using an ensemble Kalman filter for updating the reservoir model and adjoint-based optimal control of the well rates gave, on average, better results than steering the smart wells "naively" assuming a homogeneous reservoir. An early, although very approximate, estimate of the permeability field can be obtained from pressure measurements before water breakthrough has occurred. For the current implementation, there was no gain in using very accurate pressure measurements.

Nomenclature

a	number of years since the start production (years)
b	discount rate (1/year)
$\mathbf{c}$	constraint vector
$\mathbf{C}$	output matrix (measurement matrix)
$\mathbf{e}$	error vector
E	expectation operator
$\mathbf{g}$	nonlinear system function vector
$\mathbf{I}$	unit matrix
J	objective function (\$)
$\bar{J}$	modified objective function (\$)
k	discrete time step counter
K	total number of time steps
$\mathbf{K}$	Kalman gain matrix
l	correlation length
$\mathbf{L}$	left factor of $\mathbf{P}$
$\mathcal{L}$	Lagrangian (\$)
n	counter
N	number of wells
p	pressure (Pa)
$\mathbf{P}$	model error covariance matrix
q	flow rate (m^3/s)
$\mathbf{Q}$	correlation length covariance matrix
r	price per unit volume (\$/m^3)
$\mathbf{R}$	measurement error covariance matrix
S	saturation
t	time (s)
$\mathbf{u}$	input vector (control vector)
$\mathbf{x}$	state vector
$\hat{\mathbf{x}}$	extended state vector

$\mathbf{y}$	output vector (measurement vector)
ε	error
κ	specific time step, e.g. corresponding to a measurement
$\boldsymbol{\lambda}$	vector of Lagrange multipliers
μ	mean
σ	standard deviation
$\mathbf{1}$	unit vector

Subscripts

ens	ensemble
k	permeability
l	correlation length
o	oil
pr	producer
w	water

Superscripts

a	analyzed
f	forecasted
t	true

References

[1] I. Aitokhuehi, L.J. Durlofsky, V. Artus, B. Yeten and K. Aziz, Optimization of advanced well type and performance, in: *Proc. 9th European Conference on the Mathematics of Oil Recovery*, Cannes, France (2004).

[2] T.M. Apostol, *Calculus*, vol. 2, 2nd Edition (Wiley, New York, 1969).

[3] H. Asheim, Maximization of water sweep efficiency by controlling production and injection rates, paper SPE 18365, in: *Proc. SPE European Petroleum Conference*, London, UK (1988).

[4] K. Aziz and A. Settari, *Petroleum Reservoir Simulation* (Applied Science Publishers, London, UK, 1979).

[5] D.R. Brouwer, *Dynamic Water Flood Optimization with Smart Wells Using Optimal Control Theory*, Ph.D. thesis, Delft University of Technology (2004).

[6] D.R. Brouwer and J.D. Jansen, Dynamic optimisation of water flooding with smart wells using optimal control theory, SPE J. December (2004) 391–402.

[7] D.R. Brouwer, G. Nævdal, J.D. Jansen, E.H. Vefring and C.P.J.W. van Kruijsdijk, Improved reservoir management through optimal control and continuous model updating. Paper SPE 90149, in: Proc. SPE Annual Technical Conference and Exhibition, Houston, TX, USA (2004).

[8] G. Burgers, P.J. van Leeuwen and G. Evensen, On the analysis scheme in the ensemble Kalman filter, Mon. Weather Rev. 126 (1998) 1719–1724.

[9] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res. 99(C5) (1994) 10143–10162.

[10] G. Evensen, The ensemble Kalman filter: Theoretical formulation and practical implementation, Ocean Dyn. 53 (2003) 343–367.

[11] G. Gao, M. Zafari and A.C. Reynolds, Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF. Paper 93324, in: *Proc. SPE Reservoir Simulation Symposium*, Houston, TX, USA (2005).

[12] Y. Gu and D.S. Oliver, History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. Paper SPE 89942, in: *Proc. SPE Annual Technical Conference and Exhibition*, Houston, TX, USA (2004).

[13] J.D. Jansen, D.R. Brouwer, G. Nævdal and C.P.J.W. van Kruijsdijk, Closed-loop reservoir management, First Break 23 (2005) 43–48.

[14] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME, J. Basic Eng. 82(Series D) (1960) 34–45.

[15] D.G. Luenberger, *Introduction to Dynamic Systems* (Wiley, New York, 1979).

[16] G. Nævdal, T. Mannseth and E.H. Vefring, Near-well reservoir monitoring through ensemble Kalman filter. Paper SPE 75235, in: *Proc. SPE/DOE Thirteenth Symposium on Improved Oil Recovery*, Tulsa, OK (2002).

[17] G. Nævdal, T. Mannseth and E.H. Vefring, Instrumented wells and near well reservoir monitoring through ensemble Kalman filter, in: *Proc. 8th European Conf. on the Mathematics of Oil Recovery*, Freiburg, Germany (2002).

[18] G. Nævdal, L.M. Johnsen, S.L. Aanonsen and E.H. Vefring, Reservoir monitoring and continuous model updating using ensemble Kalman filter, SPE J. 10(1) (2005) 66–74.

[19] K.M. Overbeek, D.R. Brouwer, G. Nævdal and C.P.J.W. van Kruijsdijk, Closed-loop waterflooding, in: *Proc. 9th European Conference on the Mathematics of Oil Recovery*, Cannes, France (2004).

[20] P. Sarma, K. Aziz and L.J. Durlofsky, Implementation of adjoint solution for optimal control of smart wells. Paper SPE 92864, in: *Proc. SPE Reservoir Simulation Symposium*, Houston, TX, USA (2005).

[21] P. Sarma, L.J. Durlofsky, K. Aziz and W.H. Chen, Efficient real-time reservoir management using adjoint-based optimal control and model updating, Comput. Geosci. (2005), DOI: 10.1007/s10596-005-9009-z.

[22] R.F. Stengel, *Optimal Control and Estimation* (Dover, New York, 1994).

[23] B. Sudaryanto and Y.C. Yortsos, Optimization of fluid front dynamics in porous media using rate control. I. Equal mobility fluids, Phys. Fluids 12(7) (2000) 1656–1670.

[24] G.A. Virnovski, Water flooding strategy design using optimal control theory, in: *Proc. 6th European Symposium on IOR*, Stavanger, Norway (1991) pp. 437–446.