Bayesian state and parameter estimation of uncertain dynamical systems

Jianye Ching a, James L. Beck b,*, Keith A. Porter c

a Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC
b Department of Applied Mechanics and Civil Engineering, California Institute of Technology, Mail Code 104-44, Pasadena, CA 91125, USA
c George W. Housner Senior Researcher in Civil Engineering, Mail Code 104-44, California Institute of Technology, Pasadena, CA 91125, USA

Received 4 August 2004; received in revised form 16 May 2005; accepted 6 August 2005
Available online 4 November 2005

Abstract

The focus of this paper is Bayesian state and parameter estimation using nonlinear models. A recently developed method, the particle filter, is studied that is based on stochastic simulation. Unlike the well-known extended Kalman filter, the particle filter is applicable to highly nonlinear models with non-Gaussian uncertainties. Recently developed techniques that improve the convergence of the particle filter simulations are introduced and discussed. Comparisons between the particle filter and the extended Kalman filter are made using several numerical examples of nonlinear systems. The results indicate that the particle filter provides consistent state and parameter estimates for highly nonlinear models, while the extended Kalman filter does not.

Keywords: Bayesian analysis; State estimation; Parameter estimation; Dynamical systems; Monte Carlo simulation; Importance sampling; Particle filter; Stochastic simulation; Extended Kalman filter

1. Introduction

1.1. Applications of state estimation in civil engineering

State estimation is the process of using dynamic data from a system to estimate quantities that give a complete description of the state of the system according to some representative model of it. State estimation has the potential to be widely applied in civil engineering. For instance, it can be used in structural health monitoring to detect changes of dynamical properties of structural systems during earthquakes and, more generally, it can be used for system identification to better understand the nonlinear behavior of structures subject to seismic loading. For structural control, the ability to estimate system states in real time may help to accomplish an efficient control strategy. For performance-based earthquake engineering, state estimation can provide crucial information to assess the seismic performance of an instrumented structure in terms of repair costs, casualties and repair duration (dollars, deaths and downtime) shortly after the cessation of strong motion.


1.2. Development of Bayesian state-estimation algorithms

Among state-estimation methodologies, those founded on the Bayesian framework are powerful because: (a) they are rigorously based on the probability axioms and therefore preserve information; and (b) they give the probability density function (PDF) of the model state conditioned on the available information, which may then be used for any probability-based structural health monitoring, system identification, reliability
assessment and control techniques. With the PDF available, we can not only estimate the state but also give a description of the associated uncertainties. The most well known Bayesian state-estimation algorithm is the Kalman filter (KF) [9,10], which is applicable for linear models with Gaussian uncertainties. Later, KF was modified to give the extended Kalman filter (EKF) [11] to accommodate lightly nonlinear systems, and, despite several noteworthy contributions since that time [12–18], this is basically the dominant Bayesian state-estimation algorithm for nonlinear models and non-Gaussian uncertainties for the last 30 years.

Although EKF has been widely used, it is only reliable for models that are almost linear on the time scale of the updating intervals [12,13]. However, civil-engineering systems are often highly nonlinear when subject to severe loading events, in which case the applicability of the KF and EKF is often questionable. The two equations in (1) are called, from left to right, state transition (or evolution) and observation (or output) equations, respectively. The values of the variables \( x_k, y_k, w_k \) and \( v_k \) are uncertain, while \( u_k \) is assumed to be known excitation.

For each time \( k \), the dynamical system input \( u_k \) and output \( \hat{y}_k \) are measured. Note that \( \hat{y}_k \) denotes the uncertain value of the observed output (before it is measured) whereas \( \hat{y}_k \) denotes the measured value of \( y_k \) that becomes available at time \( k \). Also, \( \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_k\} \) and \( \{u_1, u_2, \ldots, u_k\} \) are denoted by \( \hat{Y}_k \) and \( U_k \), respectively. Our goal is to evaluate sequentially the PDF \( p(x_k|\hat{Y}_k) \) for the state \( x_k \) at every time \( k \), i.e. to perform a sequential update of this conditional PDF using the measured system input and output up to the current time, based on prescribed probabilistic models for \( w_k \) and \( v_k \). Note that the conditioning of every PDF on \( U_k \) and the chosen model in (1) is left implicit.

The basic equations for updating \( p(x_{k-1}|\hat{Y}_{k-1}) \) to \( p(x_k|\hat{Y}_k) \) are the predictor and updater (or corrector) equations that follow from the Theorem of Total Probability and Bayes Theorem, respectively:
\[
p(x_k|\hat{Y}_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|\hat{Y}_{k-1})dx_{k-1}
\]
\[
p(x_k|\hat{Y}_k) = \frac{p(\hat{y}_k|x_k)p(x_k|\hat{Y}_{k-1})}{\int p(\hat{y}_k|x_k)p(x_k|\hat{Y}_{k-1})dx_k} = p(\hat{Y}_k|x_k) \frac{p(x_k|\hat{Y}_{k-1})}{p(\hat{Y}_{k-1})}
\]
where \( \hat{Y}_{k-1} \) is dropped in \( p(x_k|x_{k-1}) \) and \( p(\hat{y}_k|x_k) \) because the models for the state transition and observation PDFs make it irrelevant.

An alternative formulation of the basic equations will prove useful in the next section. If the history of the state up to time \( k \) is denoted by \( X_k = \{x_0, x_1, \ldots, x_k\} \), then from Bayes Theorem:
\[
p(X_k|\hat{Y}_k) = \frac{p(X_k|\hat{Y}_{k-1})p(\hat{y}_k|X_k, \hat{Y}_{k-1})}{p(\hat{y}_k|\hat{Y}_{k-1})}
\]
\[
= p(X_{k-1}|\hat{Y}_{k-1})p(x_k|X_{k-1}, \hat{Y}_{k-1})p(\hat{y}_k|X_k, \hat{Y}_{k-1})
\]
\[
= p(X_{k-1}|\hat{Y}_{k-1})p(x_k|x_{k-1})p(\hat{y}_k|x_k)\frac{p(x_k|\hat{Y}_{k-1})}{p(\hat{Y}_{k-1})}
\]
where we have used the fact that \( p(\hat{y}_k|X_k, \hat{Y}_{k-1}) = p(\hat{y}_k|x_k) \) and that \( p(x_k|X_{k-1}, \hat{Y}_{k-1}) = p(x_k|x_{k-1}) \) based on (1) and the fact that the PDFs for \( v_k \) and \( w_k \) are prescribed. Evaluating the recursive
equation in (3), it is clear that

\[
p(X_k|\hat{Y}_k) = p(\xi_0) \cdot \prod_{m=1}^{k} \frac{p(\hat{\xi}_m|x_m) \cdot p(x_m|x_{m-1})}{p(\hat{\xi}_m|\hat{Y}_{m-1})} = \frac{p(\xi_0)}{p(\hat{Y}_k)} \cdot \prod_{m=1}^{k} p(\hat{\xi}_m|x_m) \cdot p(x_m|x_{m-1})
\]

(4)

The main challenge in Bayesian state estimation for nonlinear models is that these basic equations cannot be readily evaluated because they involve high-dimensional integrations. When \( f_k \) and \( h_k \) in (1) are linear in \( u_k, x_k, w_k \) and \( v_k \) and all uncertainties are Gaussian, the conditional PDF \( p(x_k|\hat{Y}_k) \) is also Gaussian and it can be updated analytically in a sequential manner. The updating algorithm is the well-known Kalman filter (KF). In the case that the adopted dynamical model is slightly nonlinear, an approximation for KF can be derived by linearizing the model. The resulting filter is the well-known extended Kalman filter (EKF). However, both KF and EKF can only update the first two moments of the conditional PDF for the state. For nonlinear models with non-Gaussian uncertainties, it is desirable to update the conditional PDF itself because the first two moments may not give an adequate description of the state; however, doing so completely requires an infinite number of parameters to represent the functional form of the conditional PDF.

An alternative approach is to conduct stochastic simulation by drawing samples from the conditional PDF so that the conditional expectation of any function of \( x_k \) can be consistently estimated. In the next section, we focus on stochastic simulation techniques for state estimation and use the term particle filters (PF) to denote the resulting algorithms (following the terminology of [18]), although similar algorithms have been called Monte Carlo filters [16] and sequential Monte Carlo Bayesian filters [17,23].

3. Particle filter

3.1. Stochastic simulation for state estimation

Our interest is to adopt a stochastic simulation algorithm for the conditional PDF \( p(X_k|\hat{Y}_k) \) that is Markovian, in that information is required only from time steps \( k-1 \) and \( k \), and the earlier state and observation data can be forgotten. In other words, if \( \hat{X}_{k-1} \) is a sample from \( p(X_{k-1}|\hat{Y}_{k-1}) \), the sample from \( p(X_k|\hat{Y}_k) \) must have the form \( \hat{X}_k = \{\hat{X}_{k-1}, \hat{x}_k\} \), where \( \hat{x}_k \) is the new sample. However, such a stochastic simulation algorithm cannot be directly implemented. This is because \( p(X_k|\hat{Y}_{k-1}) \) is different from \( p(X_{k-1}|\hat{Y}_{k-1}) \).

The basic idea of PF is to draw \( N \) samples \( \{\hat{X}_k^i : i = 1, \ldots, N\} \) randomly from a chosen importance sampling PDF \( q(X_k|\hat{Y}_k) \) where \( q(X_k|\hat{Y}_k) \) is chosen so that it can be readily sampled, then the expectation of any function \( r(x_k) \) conditioned on \( \hat{Y}_k \) can be estimated using the importance sampling technique as follows:

\[
E[r(X_k)|\hat{Y}_k] = \left[ \sum_{j=1}^{N} \beta_j \cdot r(\hat{X}_k^j) \right] / \sum_{j=1}^{N} \beta_j
\]

(5)

where

\[
\beta_j = \frac{p(\hat{X}_0^j) \cdot \prod_{m=1}^{k} p(\hat{\xi}_m^j|x_m^j) \cdot p(x_m^j|x_{m-1}^j)}{q(\hat{X}_k^j|\hat{Y}_k)}
\]

(6)

is the non-normalized importance weight of the \( j \)th sample. Note that the likelihood functions \( p(\hat{\xi}_m^j|x_m^j) \) and \( p(x_m^j|x_{m-1}^j) \) can be readily evaluated using the prescribed PDFs for \( v_m \) and \( w_m \) if the mappings in (1) uniquely specify \( v_m \) and \( w_m \), given \( y_m, x_m \) and \( x_{m-1} \). It is shown in [24] that this estimator is asymptotically unbiased if the support region for \( p(X_k|\hat{Y}_k) \) is a subset of that for \( q(X_k|\hat{Y}_k) \). Any quantity of interest can be estimated with the appropriate \( r(\cdot) \) function in (5); for instance, if \( r(X_k) = X_k \), \( E[r(X_k)|\hat{Y}_k] \) is simply the conditional expectation \( E[X_k|\hat{Y}_k] \); if \( r(X_k) = X_kX_k^T \), \( E[r(X_k)|\hat{Y}_k] \) is the conditional second moment \( E[X_kX_k^T|\hat{Y}_k] \). In practice, the quantity of interest might be any facility performance metric such as peak interstory drift, repair cost, repair duration, casualties, etc.

The chosen \( q(X_k|\hat{Y}_k) \) must also be Markovian since a Markovian PF algorithm is desired. In other words, the structure of \( q(X_k|\hat{Y}_k) \) must be such that \( x_{k-1} \) is independent of \( y_k \) conditioned on \( Y_{k-1} \). The selection of such an importance sampling PDF is discussed in [23,25]. The conclusion is that the following importance sampling PDF is optimal:

\[
q(X_k|\hat{Y}_k) = p(\xi_0) \cdot \prod_{m=1}^{k} p(x_m|x_{m-1}, \hat{y}_m)
\]

(7)

With this choice, one can see that the importance weights can be computed sequentially:

\[
\tilde{\beta}_k = \tilde{\beta}_{k-1} \cdot \frac{p(\hat{\xi}_m^j|x_m^j) \cdot p(\hat{\xi}_m^j|x_{m-1}^j)}{p(\hat{\xi}_m^j|\hat{y}_m)}
\]

(8)

Note that sampling from \( q(X_k|\hat{Y}_k) \) requires the ability to sample from \( p(x_m|x_{m-1}, \hat{y}_m) \), which is generally not Gaussian. A solution is to find a Gaussian PDF \( p_G(x_m|x_{m-1}, \hat{y}_m) \) whose first two moments are close to this PDF. Such a task can be accomplished by using a single-time-step EKF algorithm [18,23].

Because of the structure of the algorithm, at any time \( k \), we are only required to store the sampled states and weights in time steps \( k \) and \( k-1 \), if the quantity of interest is \( r(x_k) \) and so depends only on the current state (although, clearly, additional dependence on the previous state \( x_{k-1} \) can also be treated). As a result, the following recursive algorithm can be used:

**Algorithm 3.1**: basic PF algorithm

1. Initialize the \( N \) samples: Draw \( \hat{x}_i^j \) from \( p(\xi_0) \) and set \( \beta_i = 1/N, i = 1, \ldots, N \).
2. At time $k$, store the previous samples and weights
\[ \tilde{x}^i = \tilde{x}^i, \quad \tilde{\beta}_i = \beta_i \] (9)

For $i = 1, \ldots, N$, draw $\tilde{x}^i$ from $p_C(x_k|x_{k-1} = \tilde{x}^i, \tilde{y}_k)$ and update the importance weight
\[ \beta^i = \frac{p(\tilde{y}_k|x_k = \tilde{x}^i) \cdot p(x_k = \tilde{x}^i|x_{k-1} = \tilde{x}^i)}{p_C(x_k = \tilde{x}^i|x_{k-1} = \tilde{x}^i, \tilde{y}_k)} \] (10)

3. For $i = 1, \ldots, N$, $E[r(x_k)|\hat{Y}_k]$ can be approximated by:
\[ E[r(x_k)|\hat{Y}_k] \approx \sum_{i=1}^{N} \left[ \frac{\beta^i}{\sum_{j=1}^{N} \beta^j} \right] \cdot r(\tilde{x}^i) \] (11)

where $r(\cdot)$ is a function that maps from $x_k$ to the quantity of interest.

4. Do Steps 2 and 3 for time steps $k = 1, \ldots, T$.

3.2. Reducing sample degradation: recursive resampling and parallel particle filters

Note that it is desirable to have the importance weights $\{\beta^i: i = 1, 2, \ldots, N\}$ be approximately uniform so that all samples contribute significantly in (5), but they become far from uniform as $k$ grows, which is due to the recursion in (8) and the fact that $q(x_k|\hat{Y}_k) \neq p(x_k|\hat{Y}_k)$. Ultimately, a few weights become much larger than the rest, so the effective number of samples is small. Nevertheless, this degradation can be reduced, as described next.

Instead of letting the $N$ samples evolve through time independently (Algorithm 3.1), one can resample the samples when the importance weights become highly non-uniform [16,23,25]. After the resampling, the importance weights become uniform, therefore the degradation problem is alleviated. The resampling step tends to terminate small-weight samples and duplicate large-weight samples and, therefore, forces the $N$ samples to concentrate in the high-probability region of $p(x_k|\hat{Y}_k)$.

Although the resampling step sets the weights back to uniform, the price to pay is that the samples become dependent and therefore collectively carry less information about the state. As a result, the resampling procedure should only be executed when the importance weights become highly non-uniform. This paper proposes the following approach: monitor the coefficient of variation (c.o.v.) of the importance weights, and the resampling procedure is executed only when this c.o.v. exceeds a certain threshold, indicating that the variability in the importance weights is large.

Another novel way proposed by this research to alleviate the dependency induced by the resampling step is to conduct several independent PF algorithms and combine all of the obtained samples. Although the samples obtained in a single algorithm can be highly dependent, the samples from different algorithms are completely independent. The resulting algorithm is as follows:

Algorithm 3.2: parallel PF algorithm with resampling

1. Initialize $N$ samples for each of the $L$ parallel PFs: Draw $\tilde{x}^i$ from $p(x_0)$ and set $\tilde{\beta}^i = 1/N$ for $i = 1, \ldots, N, j = 1, \ldots, L$.
2. Perform the following steps 3 – 4 for $j = 1, \ldots, L$ independently. Since the processes are completely independent, they can be conducted in parallel.
3. At time $k$, store the previous samples and weights
\[ \tilde{x}^{i,j} = \tilde{x}^i, \quad \tilde{\beta}^{i,j} = \beta^j \] (12)

For $i = 1, \ldots, N$, draw $\tilde{x}^{i,j}$ from $p_C(x_k|x_{k-1} = \tilde{x}^{i,j}, \tilde{y}_k)$ and update the importance weight
\[ \beta^{i,j} = \frac{p(\tilde{y}_k|x_k = \tilde{x}^{i,j}) \cdot p(x_k = \tilde{x}^{i,j}|x_{k-1} = \tilde{x}^{i,j})}{p_C(x_k = \tilde{x}^{i,j}|x_{k-1} = \tilde{x}^{i,j}, \tilde{y}_k)} \] (13)

4. Compute the c.o.v. of $\{\tilde{\beta}^{i,j}: i = 1, \ldots, N\}$. If the c.o.v. is larger than the prescribed threshold, then execute the resampling step for $i = 1, \ldots, N$:
\[ \tilde{x}^{i,j} = \tilde{x}^{i,j} \text{ w.p. } \frac{\tilde{\beta}^{i,j}}{\sum_{i=1}^{N} \tilde{\beta}^{i,j}} \] (14)

and set $\beta^{i,j} = 1/N$ for $i = 1, \ldots, N$. Otherwise, for $i = 1, \ldots, N$:
\[ \tilde{x}^{i,j} = \tilde{x}^i, \quad \beta^{i,j} = \frac{\tilde{\beta}^{i,j}}{\sum_{i=1}^{N} \tilde{\beta}^{i,j}} \] (15)

Store $\tilde{\beta}^{i,j}_{k,N} = \sum_{j=1}^{L} r(\tilde{x}^{i,j}) \cdot \tilde{\beta}^{i,j}$

6. $E[r(x_k)|\hat{Y}_k]$ can be then approximated by
\[ E[r(x_k)|\hat{Y}_k] \approx \frac{\sum_{j=1}^{L} \tilde{\beta}^{i,j}_{k,N}}{L} \] (16)

7. Do Steps 2 – 5 for $k = 1, \ldots, T$.

3.3. Advantages and disadvantages of the PF technique

The advantages of the PF technique include: (a) as $N$ (the number of samples per algorithm) approaches infinity, the value of any function of the state $x_k$ estimated by PF converges to its expected value; therefore, the PF technique can be used to validate other methodologies; and (b) parallel computations are possible for PF algorithms. A disadvantage of the PF technique is that it is computationally expensive, especially when the degradation is severe so that we need large $N$ and $L$ to have the algorithm converge. In general, the required $N$ and $L$ grow with the size of the effective support region of $p(x_k|\hat{Y}_k)$. A simple test for convergence is to add parallel particle filters until the estimated quantity of interest, $r(x_k)$, does not significantly change. It is noted that for linear models with time-varying unknown parameters, a more efficient PF algorithm can be derived, as shown in [24].
4. Examples to compare EKF and PF methods

We present three examples in this section. We generate data that is contaminated by noise for three simulated dynamical systems. With the simulated data, we use identification models that are derived from these dynamical systems to conduct EKF and PF and we compare their performance. The goal of these examples is to see if these techniques produce consistent results.

4.1. Example 1: planar four-story shear building with time-varying system parameters

4.1.1. Data generation

We first describe the system that generates the simulated data. Consider an idealized planar four-degrees-of-freedom (DOF) shear-building system with known time-invariant masses equal to \( m_1 = m_2 = m_3 = m_4 = 250,000 \text{ kg} \) (subscript denotes the story number). The inter-story stiffnesses \( k_1, k_2, k_3 \) and \( k_4 \) change through time as shown in Fig. 1 (\( k_1, k_3 \) and \( k_4 \), drift around certain values, while \( k_2 \) significantly decreases and then partially recovers). The inter-story viscous damping coefficients are \( c_1, c_2, c_3 \) and \( c_4 \) and are also time-varying (Fig. 1). The time evolutions of \( k_1, k_3, k_4, c_1, c_2, c_3 \) and \( c_4 \) are Brownian motions with standard deviation of the drift equal to 2% of their mean values during each sampling interval described later. The governing equation of this system subject to base excitation is

\[
M\ddot{x}_t + C_t x_t + K_t x_t = F_{u_t},
\]

where

\[
x_t = \begin{bmatrix} x_{4,t} \\ x_{3,t} \\ x_{2,t} \\ x_{1,t} \end{bmatrix}, \quad M = \begin{bmatrix} m_4 & 0 & 0 & 0 \\ 0 & m_3 & 0 & 0 \\ 0 & 0 & m_2 & 0 \\ 0 & 0 & 0 & m_1 \end{bmatrix},
\]

\[
F_t = \begin{bmatrix} -m_4 \\ -m_3 \\ -m_2 \\ -m_1 \end{bmatrix}, \quad C_t = \begin{bmatrix} c_{4,t} & -c_{4,t} & 0 & 0 \\ -c_{4,t} & c_{4,t} + c_{3,t} & -c_{3,t} & 0 \\ 0 & c_{3,t} & c_{3,t} + c_{2,t} & -c_{2,t} \\ 0 & 0 & -c_{2,t} & c_{2,t} + c_{1,t} \end{bmatrix},
\]

\[
K_t = \begin{bmatrix} k_{4,t} & -k_{4,t} & 0 & 0 \\ -k_{4,t} & k_{4,t} + k_{3,t} & -k_{3,t} & 0 \\ 0 & -k_{3,t} & k_{3,t} + k_{2,t} & -k_{2,t} \\ 0 & 0 & -k_{2,t} & k_{2,t} + k_{1,t} \end{bmatrix}
\]

Fig. 1. Time evolutions of the actual inter-story stiffnesses and dampings (example 1).
x_{it} is the displacement of the \((i+1)\)-th floor (fifth floor is the roof) relative to the ground at time \(t\); \(u_t\) is the acceleration at the base (first floor of the building); and \(c_{it}\) and \(k_{it}\) are the inter-story damping coefficient and stiffness of the \(i\)-th story, respectively, at time \(t\).

We generate the data using white-noise for the excitation \(u_t\).

The observed output \(y_t\) is the absolute acceleration time histories at the four stories:

\[
y_t = \begin{bmatrix} x_{1t} + u_t \\
       x_{2t} + u_t \\
       x_{3t} + u_t \\
       x_{4t} + u_t 
\end{bmatrix} + \Gamma \cdot v_t = -M^{-1} [C_t \dot{x}_t + K_t x_t] + \Gamma \cdot v_t \tag{19}
\]

where \(v_t \in \mathbb{R}^4 \sim N(0, I)\) are the (stationary) measurement uncertainties for \(y_t\); \(\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_4)\) is such that the overall signal/noise rms amplitude ratios for each channel is roughly equal to 10. Both the excitation and observation are sampled at an interval of 0.02 s and are shown in Fig. 2. With the excitation and observation of the system, i.e. \(\{u_t; t=1, \ldots, T\}\) and \(\{y_t; t=1, \ldots, T\}\), the goal is to estimate the system states (displacements and velocities) as well as the system parameters (dampings, stiffnesses and the uncertainty parameters) in real time.

4.1.2. Identification model

The identification model is now described. The following time-varying linear state-space model is used as the identification model:

\[
d \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} \overline{= \begin{bmatrix} \dot{x}_t \\
0 
\end{bmatrix} + \begin{bmatrix} -M^{-1} K_{1t} x_t - M^{-1} C_{1t} \dot{x}_t \\
0 
\end{bmatrix} u_t \\
0 \\
G \end{bmatrix} u_t + \begin{bmatrix} 0 \\
0 \end{bmatrix} w_t + \begin{bmatrix} 0 \\
G \end{bmatrix} \theta_t \tag{20}
\]

where \(w_t \sim N(0, I); \quad v_t \sim N(0, I); \quad G \in \mathbb{R}^{12 \times 12}\) is a diagonal matrix whose diagonals must be specified (the reason will be explained later); \(H_t = \text{diag}(h_{1t}, h_{2t}, h_{3t}, h_{4t})\) where the diagonals are unknown parameters (i.e. the four uncertainty parameters); and \(\theta_t \in \mathbb{R}^{12}\) is the vector containing system parameters (including four stiffnesses, four dampings and four uncertainty parameters). The dimension of the state of the identification model is 20 (four displacements, four velocities, four stiffnesses, four dampings and four uncertainty parameters), whereas the dimension of the state in (17) is only eight since it consists of \(x_t\) and \(\dot{x}_t\).

To complete the probabilistic identification model, one must also specify the prior PDF for the entire (augmented) state trajectory \(\{x_t, \dot{x}_t; t=0, \ldots, T\}\). More specifically, for the model in (20), one must specify the following: the prior PDF of \(x_0\) and \(\dot{x}_0\); the prior PDF of \(\theta_0\); and the diagonals of the \(G\) matrix. Note that the identification model in (20) uses a Brownian-motion prior PDF for the parameter evolution \(\{\theta_t; t=0, \ldots, T\}\) due to the following dynamical equation in (20):

\[
\theta_t = G \cdot w_t \tag{21}
\]

A diagonal \(G\) matrix means that all system parameters are known a priori to drift independently. The effect of the \(G\) matrix is similar to the forgetting factor often used in adaptive filtering [26]. When the entries of \(G\) are large, the system parameters are allowed to drift more freely, relaxing the dependency between parameter values of adjacent time steps; therefore, the identified parameters only reflect most recent data. The converse is true when the entries of \(G\) are small; in this case, the identified parameters can reflect data from the remote past.

In this example, the prior PDF for \(x_0\) and \(\dot{x}_0\) is taken to be zero-mean Gaussian with large variances; the prior PDF of \(\theta_0\) is taken to be Gaussian with mean equal to the actual value of \(\theta_0\) and large variances; the diagonal entries of \(G\) are chosen such that in each time step, each parameter drifts with a coefficient of variation (c.o.v., defined by the standard deviation divided by the mean value) equal to 2%. Recall that for \(k_1, k_3, k_4, c_1, c_2, c_3\) and \(c_4\), their actual evolutions (see Fig. 1) are Brownian motions with the same 2% drift c.o.v., i.e. there is no modeling error for the evolutions of \(k_1, k_3, k_4, c_1, c_2, c_3\) and \(c_4\). But for \(k_2\) and the four uncertainty parameters, the actual evolutions are not Brownian motions (the actual evolution of \(k_2\) is shown in Fig. 1; the four uncertainty parameters are actually constant),

![Fig. 2. The simulated excitation and observation data (example 1).](image-url)
while the identification model uses a Brownian-motion prior on their evolutions.

Before we can proceed, we first convert (20) to the following discrete-time system using numerical integration (integrate over time step):

\[
\begin{bmatrix}
  x_k \\
  \dot{x}_k \\
  \theta_k 
\end{bmatrix} = f_{k-1}\left(\begin{bmatrix}
  x_{k-1} \\
  \dot{x}_{k-1} \\
  \theta_{k-1}
\end{bmatrix}, u_{k-1}, w_{k-1}\right)
\]

\[
y_k = h_k\left(\begin{bmatrix}
  x_k \\
  \dot{x}_k \\
  \theta_k 
\end{bmatrix}, u_k, v_k\right)
\]

(22)

where \(f_{k-1}\) is evaluated using Matlab command ODE23.

4.1.3. Results
The EKF and PF methods are applied to the generated simulated data. The stiffness, damping and uncertainty parameter estimates, and the associated 95\% confidence intervals from EKF and PF (with \(N=200\) samples for each of \(L=10\) parallel PF and the importance weight c.o.v. threshold = 200\% using Algorithm 3.2) are shown in Figs. 3–8. (EKF does not provide confidence intervals for the uncertainty parameter estimates.) For this example, using more samples in PF than \(NL = 2000\) gives little improvement in the convergence, indicating that the results are close to convergence. One can treat the results from PF as a comparison standard since it gives asymptotically consistent estimates for the conditional means and variances.

In Figs. 3–8, the thick lines indicate the actual parameter evolutions while the thin dashed lines are the conditional means of the identified system parameters and the thin dotted lines indicate the 95\% confidence intervals. The results from EKF are similar to those of PF. Both algorithms successfully track the system parameters. For most parameters, the actual parameter evolutions lie within the 95\% confidence bounds. Notice that although the Brownian-motion prior for \(k_2\) and the uncertainty parameters do not exactly match their actual evolutions, both Bayesian algorithms can still appropriately track \(k_2\) and the uncertainty parameters. Compared to the accuracy of the stiffnesses, the estimates of the damping and uncertainty parameters are worse and the associated uncertainties are larger. Although EKF and PF perform roughly equally in this example, there is a noticeable difference in the variances of the identified damping from PF, which are slightly larger than those from EKF. For this example, EKF and PF perform approximately equally because there is no highly nonlinear behavior.

4.2. Example 2: nonlinear hysteretic damping system with unknown system parameters

4.2.1. Data generation
The previous example is a time-varying linear system. In the current example, consider a time-varying nonlinear system consisting of a single DOF (SOF) Bouc-Wen hysteretic damping system [27]. The purpose of this example is to
Fig. 4. The PF estimates of inter-story stiffness (example 1).

Fig. 5. The EKF estimates of inter-story damping (example 1).
Fig. 6. The PF estimates of inter-story damping (example 1).

Fig. 7. The EKF estimates of uncertainty parameters $h_{1,t}$, $h_{2,t}$, $h_{3,t}$, $h_{4,t}$ (example 1).
compare the performances of different methods for tracking the state and unknown parameters of a nonlinear system. The system that generates the data can be described by the following governing equation:

\[
\begin{bmatrix}
\frac{d}{dr} x_t \\
\frac{d}{dr} \dot{x}_t \\
\frac{d}{dr} r_t
\end{bmatrix}
= \begin{bmatrix}
\dot{x}_t \\
-1/m \cdot r_t + 1/m \cdot u_t \\
\theta_{1,t} \cdot \dot{x}_t - \theta_{2,t} \cdot |\dot{x}_t| r_t^{\theta_{3,t}} - r_t + \theta_{3,t} \cdot \dot{x}_t |r_t|^{\theta_{4,t}}
\end{bmatrix}
\]

\[
y_t = -1/m \cdot r_t + 1/m \cdot u_t + v_t
\]

where \( r_t \) is the restoring force of the SDOF system; \( m \) is the mass, which is set to unity during the data generation; \( u_t \) is a white-noise excitation force on the mass; \( y_t \) is the acceleration measured on the mass; \( v_t \) is stationary such that the overall signal/noise amplitude ratio is 10; \( \theta_{1,t} \), \( \theta_{2,t} \), \( \theta_{3,t} \), \( \theta_{4,t} \) are time-varying system parameters (their actual fluctuations are shown in Figs. 10 and 11, and they are Brownian motions with drift c.o.v. equal to 2\% during each sampling interval); \( \theta_{1,t} \) is the stiffness, \( \theta_{2,t} \), \( \theta_{3,t} \) and \( \theta_{4,t} \) are parameters that fine tune the shape of the hysteretic loop. Note that Bouc-Wen hysteretic damping system is Markovian in the sense that one can define a finite-dimensional state such that the current system status is completely characterized by this state. Both the excitation \( u_t \) and observation \( \dot{y}_t \) (shown in Fig. 9) are sampled at an interval of 0.5 s (roughly five sample points per oscillation cycle of the system).

Fig. 8. The PF estimates of uncertainty parameters \( h_{1,t}, h_{2,t}, h_{3,t}, h_{4,t} \) (example 1).

Fig. 9. The excitation force \( u_t \) and the observed acceleration \( \dot{y}_t \) (example 2).
4.2.2. Identification model

Given the data \( u_t \) and \( y_t \), the goal is to estimate the means and variances of the identified system state and system parameters using the following identification model:

\[
\frac{dx_t}{dt} = \begin{bmatrix}
\dot{x}_t \\
\dot{r}_t \\
\dot{\theta}_{1,t} \\
\dot{\theta}_{2,t} \\
\dot{\theta}_{3,t} \\
\dot{\theta}_{4,t} \\
\dot{h}_t
\end{bmatrix} = \begin{bmatrix}
-1/m \cdot r_t + 1/m \cdot u_t \\
\theta_{1,t} \cdot \dot{x}_t - \theta_{2,t} \cdot |x_t| r_t^{\theta_{3,t} - 1} r_t + \theta_{3,t} \cdot \dot{x}_t |r_t|^{\theta_{4,t}} \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[
y_t = -1/m \cdot r_t + 1/m \cdot u_t + h_t \cdot v_t + w_t
\]

where \( w_t \sim \mathcal{N}(0, I) \); \( v_t \sim \mathcal{N}(0, 1) \); and \( m \) is assumed to be known so it is not considered one of the uncertain parameters.

The prior PDF for \( x_0 \) and \( \dot{x}_0 \) is taken to be zero-mean Gaussian with large variances; the prior PDF of \( \theta_{1,0}, \theta_{2,0}, \theta_{3,0}, \theta_{4,0}, h_0 \) is assumed to be Gaussian with mean equal to their actual value at time zero and large variances; the \( 5 \times 5 \) matrix \( G \) in (24) is taken to be diagonal. The diagonals of \( G \) are chosen such that in each time step, each parameter is allowed to drift with a c.o.v. equal to 2%, i.e. there is no modeling error for the evolutions of \( \theta_{1,t}, \theta_{2,t}, \theta_{3,t}, \theta_{4,t} \), but modeling error exists for the evolution of \( h_t \) in (24) (the actual \( h_t \) is constant instead of a Brownian motion). The continuous-time model is numerically integrated to get the discrete-time version of this model similar to (22) with sampling rate equal to 0.5 s.

4.2.3. Results

Figs. 10–15 show the results of identification of EKF and PF (as before, \( N = 200 \) and \( L = 10 \) and the importance weight c.o.v. threshold \( = 200\% \) using Algorithm 3.2). For this example, using more samples in PF than \( NL = 2000 \) gives little improvement in the convergence, indicating that the results are close to convergence. The results from PF are treated as a comparison standard. As before, the 95% confidence intervals on the parameters and states are indicated by thin dotted lines in Figs. 10–13 and 15.

It is found that EKF performs less effectively than PF: at some time instants, the EKF estimates of the stiffness parameter \( \theta_{1,t} \) oscillate around the actual evolution (Fig. 10), while this is not seen for PF (Fig. 11). Also, the EKF estimates for \( \theta_{2,t} \) (Fig. 10) significantly deviate from those of PF (Fig. 11). For the estimation of displacement, velocity and restoring force, the performances from the three methods are similar (Figs. 12 and 13). However, PF estimates
Fig. 11. The PF estimates of the system parameters (example 2).

Fig. 12. The EKF estimates of the system states (example 2).
4.3. Example 3: Lorenz chaotic system

The Lorenz system exhibits chaotic behavior, as discovered by Lorenz [28] when he solved a simplified Rayleigh-Bernard problem modeling two-dimensional fluid motion driven by buoyancy due to a temperature difference across its height. The resulting simplified set of differential equations that consider the first few modes of the system is

$$\begin{align*}
\dot{x}_1 &= -\sigma(x_1 - x_2) \\
\dot{x}_2 &= r \cdot x_1 - x_2 - x_1 x_3 \\
\dot{x}_3 &= x_1 x_2 - b x_3
\end{align*}$$

(25)

where $x_1,t$ specifies the time evolution of the stream-function of the first mode, whose contours are the streamlines; $x_2,t$ and $x_3,t$ specify the time evolutions of the temperature of the first two modes of the system; the parameter $\sigma$ depends on the properties of the fluid (for water the value is typically between 1 and 4);
the number \( b \) depends on the scales of the modes; \( r \) is the temperature difference: for small \( r \), the system is asymptotically stable, i.e. \( x_1, t \to 0 \) as \( t \to \infty \). For large \( r \), chaos occurs with the so-called butterfly attractor where the ultimate fate of a trajectory of the system is to wander around two unstable equilibrium points and the trajectory is extremely sensitive to its initial condition.

### 4.3.1. Data generation

In this example, the values of \( \sigma, b \) and \( r \) are set to 3, 1 and 26 (\( r \) is large so that the butterfly attractor occurs), and one observes \( x_1, t \) (contaminated by noise) with sampling interval of 0.5 s:

\[
y_t = x_{1,t} + h \cdot v_t
\]  

(Eq. 26)

where \( h \) is chosen such that the overall signal/noise ratio is 10. Fig. 16 shows the observed value \( \hat{y}_t \), which clearly shows that the trajectory of \( x_{1,t} \) switches several times between the two unstable equilibrium points at \( x_1 = -5 \) and \( x_1 = 5 \) (especially during 0–25 s).

### 4.3.2. Identification model

The goal is to estimate the trajectory of the three system states based on \( \hat{y}_t \) using EKF and PF. When applying EKF and PF, it is assumed that one is very uncertain about the position of the initial state (i.e. large variances for the prior PDF of the three states); it is also assumed that \( \sigma, b, r \) and \( h \) are known, and their actual values are used during the identification. Eqs. (25) and (26) are directly used in the identification model.

### 4.3.3. Results

Figs. 17 and 18 show the estimates made by EKF and PF (with \( N = 200 \) and \( L = 10 \) and the importance weight c.o.v. threshold = 200% using Algorithm 3.2). For this example, using \( NL = 2000 \) samples in PF is found to be sufficient for convergence, so once again, one can treat the results from PF as a comparison standard. Also, as before, the 95% confidence intervals on the states are indicated by thin dotted lines in Figs. 17 and 18.

It is clear that PF can successfully track all three system states, while EKF can only reliably track the observed state variable \( x_{1,t} \) (since \( \hat{y}_t \) directly measures \( x_{1,t} \), it is possible that an inappropriate filtering algorithm can still track \( x_{1,t} \) perfectly.) EKF can track some parts of \( x_{2,t} \) and \( x_{3,t} \), but performs poorly in other parts, especially in the beginning portions of \( x_{2,t} \) and \( x_{3,t} \) where the system switches between the two equilibrium points.

![Fig. 16. The plot for \( \hat{y}_t \) (example 3).](image)

![Fig. 17. The EKF estimates of the system states (example 3).](image)
5. Conclusion

Two Bayesian state-estimation algorithms are examined: the extended Kalman filter (EKF) and the newer particle filter (PF), a stochastic simulation approach. Their performance is compared using three numerical examples, which show that PF is the better one to use, since EKF can sometimes create misleading results. Basically, the three examples represent three different classes of dynamical systems: a linear model with time-varying system parameters (Section 4.1), the nonlinear hysteretic model (Section 4.2) that can be considered to give moderately nonlinear behavior, and the Lorenz chaotic model (Section 4.3) that gives highly nonlinear behavior.

It is believed that PF has performed satisfactorily for all examples, judging from the fact that it always provide estimates for the system state and unknown parameters with associated confidence intervals that are consistent with their actual values. In theory, PF should provide estimates that asymptotically converge to the expected values. It turns out that EKF can only track the system state and unknown parameters for the first example, its performance for the second example is only fair, and it performs poorly for the Lorenz chaotic example. This is consistent with the expectation that EKF is not suitable for highly nonlinear models.

References


