Kalman filtering techniques for parameter estimation

Jared Barber
Department of Mathematics, University of Pittsburgh
Work with Ivan Yotov and Mark Tronzo
March 17, 2011
Summary from last time

• Kalman filter
  – Combines measurements and model information to produce a best estimate of the state of the system at each time step at which measurements are available
  – Can be viewed as using measurements to, every once in a while, improve model’s estimates of the state of the system

• Parameter estimation
  – Unknown parameter values can be included as components in the state vector and data used to adjust the parameters so the model produces results closer to the measured data
Summary from last time

• Kalman Filter Parameter estimation
  – Unknown parameter values can be included as components in the state vector and data used to adjust the parameters so the model produces results closer to the measured data

• KF Parameter estimation benefits
  – Sequential parameter estimation
  – Best estimates of parameters and uncertainty associated with those parameters both tracked
  – Handle nonlinearities without Jacobian
  – Easily parallelized
Summary of Linear Kalman Filter

• Forecast Step

\[ x_k^f = A_k x_{k-1} \]
\[ P_{xx,k}^f = A_k P_{xx,k-1}^a A_k^T + P_{xx,k}^{model} \]
\[ P_{xd,k}^f = P_{xx,k}^f H^T \]
\[ P_{dd,k}^f = H P_{xx,k}^f H^T + P_{dd,k}^{meas} \]

• Adjustment/Analysis Step

\[ d_k^m \]
\[ K_k = P_{xd,k}^f (P_{dd,k}^f)^{-1} \]
\[ x_k^a = x_k^f + K_k (d_k^m - H x_k^f) \]
\[ P_{xx,k}^a = (I - K_k H) P_{xx,k}^f \]
Summary Ensemble Kalman Filter

**Forecast step**

\[ x_{k,\text{en},i}^f = f(x_{k-1,\text{en},i}^a) + w_{k,\text{en},i} \]

\[ d_{k,\text{en},i}^f = h(x_{k,\text{en},i}^f) \]

\[ \bar{x}_{k,\text{en}}^f = \frac{1}{(q-1)} \sum_{i=1}^{q} x_{k,\text{en},i}^f \]

\[ \bar{d}_{k,\text{en}}^f = \frac{1}{(q-1)} \sum_{i=1}^{q} d_{k,\text{en},i}^f \]

\[ P_{\text{xd},k,i,j}^f = \sum_{l=1}^{q} \frac{(x_{k,\text{en},l,i}^f - \bar{x}_{k,\text{en},i}^f)(d_{k,\text{en},l,j}^f - \bar{d}_{k,\text{en},j}^f)}{q-1} \]

\[ P_{\text{dd},k,i,j}^f = \sum_{l=1}^{q} \frac{(d_{k,\text{en},l,i}^f - \bar{d}_{k,\text{en},i}^f)(d_{k,\text{en},l,j}^f - \bar{d}_{k,\text{en},j}^f)}{q-1} \]

**Adjustment/Analysis Step**

\[ d_{k,\text{en},i}^m = d_{k,\text{en},i}^m + v_{k,\text{en},i} \]

\[ K_k = P_{\text{xd},k}^f (P_{\text{dd},k}^f)^{-1} \]

\[ x_{k,\text{en},i}^a = x_{k,\text{en},i}^f + K_k (d_{k,\text{en},i}^m - h(x_{k,\text{en},i}^f)) \]
Ensemble KF
Advantages/Disadvantages

• Handles nonlinearity without Jacobian
• For huge state vectors, avoids tracking huge covariances and associated Jacobian
• Need a good number of points to get decent results...depends on application
• Time Kalman Filter takes is proportional to number of calls to the model made...hence with an ensemble of points we call the model many times and En KF takes long time
Stochastic Collocation

• Assume we have a random/stochastic variable $z$ defined on $\mathbb{R}^n$ with a gaussian distribution (so $z$ is a vector)

• The random variable has a mean $\bar{z}$ and covariance matrix $P_{zz}$

• Consider a general function of the random variable $g(z)$ (may or may note be a vector). It’s expected value can be found by evaluating the integral:

$$E[g(z)] = \int_{\mathbb{R}^n} g(z) \left( \frac{1}{2\pi} \right)^{n/2} \frac{e^{-(z-\bar{z})^T P_{zz}^{-1} (z-\bar{z})}}{\det(P_{zz})} \, dz$$
Stochastic Collocation

• This integral can sometimes be hard to evaluate, especially in multiple dimensions. One way to obtain an estimate of the integral is to use Monte Carlo integration:

\[
E[g(z)] = \int g(z) \left( \frac{1}{2\pi} \right)^{n/2} \frac{e^{-(z-\bar{z})P_{zz}^{-1}(z-\bar{z})^T/2}}{\det(P_{zz})} \, dz \approx \frac{1}{(q-1)} \sum_{i=1}^{q} g(z_i)
\]

• Here \( q \) is a large number corresponding to the number of random samples \( z_i \) that we chose to obtain the estimate of the expected value.
Stochastic Collocation

• When the number of dimensions isn’t too terribly large, the integral can also be approximated using quadrature rules:

• 1d:

\[ E[g(z)] = \int_{-\infty}^{\infty} g(z) \frac{e^{-z^2/2\sigma^2}}{\sigma \sqrt{2\pi}} \, dz \approx \left( g(-\sigma \sqrt{3}) + 4g(0) + g(\sigma \sqrt{3}) \right) / 6 \]

• \( z_i = -\sigma \sqrt{3}, 0, \sigma \sqrt{3} \) are called collocation points

• \( w_i = \) are called collocation weights
Stochastic Collocation

- In n-dimensional stochastic space the expected value can be approximated by choosing appropriate collocation points and weights:

\[
E[g(z)] = \int_{\mathbb{R}^n} g(z) \left( \frac{1}{2\pi} \right)^{n/2} e^{-\frac{(z-z_\text{ref})^T P_{zz}^{-1} (z-z_\text{ref})}{2}} \frac{1}{\det(P_{zz})} \, dz \\
\approx \sum_{i=1}^{2n+1} w_i g(z_i)
\]

- Exact if \( g(z) \) is a polynomial of degree 2 or less
Obtaining mean value and covariance estimates

• Let \( g(z) = z \), the expected value of the underlying distribution can be recovered by either

\[
E[z] \approx \frac{1}{n} \sum_{i=1}^{q} z_i \quad \text{Monte Carlo}
\]

\[
E[z] \approx \sum_{i=1}^{2n+1} w_i z_i \quad \text{Stochastic Collocation}
\]
Obtaining mean value and covariance estimates

- Similarly let \( g(z) = (z_i - \mathbb{E}(z_i))(z_j - \mathbb{E}(z_j)) \), the covariance of the underlying distribution can be recovered by either

\[
P_{zz,i,j} \approx \frac{1}{(q-1)} \sum_{i=1}^{q} (z_i - \bar{z}_i)(z_j - \bar{z}_j)
\]

Monte Carlo

\[
P_{zz,i,j} \approx \sum_{i=1}^{2n+1} w_i (z_i - \bar{z}_i)(z_j - \bar{z}_j)
\]

Stochastic Collocation

- Ensemble Kalman Filter uses first method
- Stochastic Collocation Kalman Filter uses second (and is usually more precise)
Stochastic Collocation Kalman Filter

- **Forecast step**

\[
x_{k,\text{en},i}^f = f(x_{k-1,\text{en},i}^a)
\]

\[
d_{k,\text{en},i}^f = h(x_{k,\text{en},i}^f)
\]

\[
\bar{x}_{k,\text{en}}^f = \sum_{i=1}^{2n+1} w_i x_{k,\text{en},i}^f
\]

\[
\bar{d}_{k,\text{en}}^f = \sum_{i=1}^{2n+1} w_i d_{k,\text{en},i}^f
\]

\[
P_{xx,k,i,j}^f = \sum_{l=1}^{2n+1} w_i (x_{k,\text{en},l,i}^f - \bar{x}_{k,\text{en},i}^f)(x_{k,\text{en},l,j}^f - \bar{x}_{k,\text{en},j}^f) + P_{\text{model}}
\]

\[
P_{xd,k,i,j}^f = \sum_{l=1}^{2n+1} w_i (x_{k,\text{en},l,i}^f - \bar{x}_{k,\text{en},i}^f)(d_{k,\text{en},l,j}^f - \bar{d}_{k,\text{en},j}^f)
\]

\[
P_{dd,k,i,j}^f = \sum_{l=1}^{2n+1} w_i (d_{k,\text{en},l,i}^f - \bar{d}_{k,\text{en},i}^f)(d_{k,\text{en},l,j}^f - \bar{d}_{k,\text{en},j}^f) + P_{\text{meas}}
\]

- **Adjustment/Analysis Step**

\[
K_k = P_{xd,k}^f (P_{dd,k}^f)^{-1}
\]

\[
\bar{x}_k^a = \bar{x}_k^f + K_k (d_{k}^m - h(\bar{x}_k^f))
\]

\[
P_{xx,k}^a = (I - K_k H) P_{xx,k}^f
\]

\[
\chi_{k,\text{en},i}^a = \bar{x}_k^a + \left( \sqrt{P_{xx,k}^a} \right) z_{en,i}
\]
Ensemble and Stochastic Collocation Comparison

- En mean \((q\approx 100 \text{ pts})\)
- SC mean \((N\approx 6 \text{ pts})\)
Stochastic Collocation Advantages and Disadvantages

- Faster than En KF for small numbers
- Slower than En KF for large numbers
- Usually more accurate than En KF
- Can handle nonlinearities
- Curse of dimensionality for pdes
  - 20x20x20 grid needs 16001 collocation points
- Is there any way to get around this?
Stochastic Collocation Kalman Filter with Karhunen-Loève Expansion

- To make the Kalman Filter work with pdes on 1-d, 2-d, or 3-d grids, the values in the grid are transferred into a state vector

\[
\begin{pmatrix}
  x_{1,1,1} \\
  x_{2,1,1} \\
  x_{3,1,1} \\
  \vdots \\
  p_1 \\
  \vdots \\
\end{pmatrix}
\]
Stochastic Collocation Kalman Filter with Karhunen-Loeve Expansion

• Usually, proceed by assuming that applying the model or measuring data produces uncorrelated errors between state components/measurement locations.

• In general, not true. Errors at adjacent points on a grid are often correlated.

• How do we represent correlated noise?
Stochastic Collocation Kalman Filter with Karhunen-Loeve Expansion

- Oftentimes people use polynomial chaos expansion or the Karhunen-Loeve Expansion to represent correlated noise. In such cases, covariances may be represented by eigenfunction-like expansions:

\[
\text{cov}(x, y) = \sum_{j=1}^{\infty} \lambda_j f_j(x) f_j(y)
\]

\[
\approx \sum_{j=1}^{\text{small}} \lambda_j f_j(x) f_j(y)
\]
Stochastic Collocation Kalman Filter with Karhunen-Loeve Expansion

• For us, the discretized version is (for the state-state covariance matrix):

\[ P_{xx,i,j} = \sum_{l=1}^{8000} \lambda_l f_l(location(x_i)) f_l(location(x_j)) \]

\[ \approx \sum_{l=1}^{100} \lambda_l f_l(location(x_i)) f_l(location(x_j)) \]

• Can now represent covariance using an organized ensemble:

\[ \chi_{k,\text{en},i,j}^a = \bar{\chi}_{k,j}^a + z_{\text{en},i,j} (f_{k,i,j} P_{xx,k}^a f_{k,i,j}) f_{k,i,j} \]
Stochastic Collocation Kalman Filter

• Forecast step

\[ x_{k, \text{en}, i}^f = f(x_{k-1, \text{en}, i}^a) \]

\[ d_{k, \text{en}, i}^f = h(x_k^f) \]

\[ \bar{x}^f_{k, \text{en}} = \sum_{i=1}^{2n+1} w_i x_{k, \text{en}, i}^f \]

\[ \bar{d}^f_{k, \text{en}} = \sum_{i=1}^{2n+1} w_i d_{k, \text{en}, i}^f \]

\[ P_{xx, k, i, j}^f = \sum_{l=1}^{2n+1} w_i (x_{k, \text{en}, l, i}^f - \bar{x}_{k, \text{en}, i}^f)(x_{k, \text{en}, l, j}^f - \bar{x}_{k, \text{en}, j}^f) + P_{\text{model}} \]

\[ P_{xd, k, i, j}^f = \sum_{l=1}^{2n+1} w_i (x_{k, \text{en}, l, i}^f - \bar{x}_{k, \text{en}, i}^f)(d_{k, \text{en}, l, j}^f - \bar{d}_{k, \text{en}, j}^f) \]

\[ P_{dd, k, i, j}^f = \sum_{l=1}^{2n+1} w_i (d_{k, \text{en}, l, i}^f - \bar{d}_{k, \text{en}, i}^f)(d_{k, \text{en}, l, j}^f - \bar{d}_{k, \text{en}, j}^f) + P_{\text{meas}} \]

• Adjustment/Analysis Step

\[ K_k = P_{xd, k}^f (P_{dd, k}^f)^{-1} \]

\[ \bar{x}_k^a = \bar{x}_k^f + K_k (d_k^m - h(\bar{x}_k^f)) \]

\[ P_{xx, k}^{a} = (I - K_k H) P_{xx, k}^f \]

\[ x_{k, \text{en}, i, j}^a = \bar{x}_{k, j}^a + z_{en, i, j} (f_{k, i, j} P_{xx, k}^{a} f_{k, i, j}) f_{k, i, j} \]
Karhunen-Loeve Advantages and Disadvantages

• Number of ensemble members very small-> fast parameter estimation (e.g. 201 vs 16001 for 20x20x20 grid)

• Doesn’t capture all info...may not always work (at least how we’ve implemented it)
Simplified Necrotizing Enterocolitis Model: The experiment

Create wound with pipette

Epithelial Layer

Wound

Epithelial Layer

\[ \approx 150 \ \mu m \]
Simplified Necrotizing Enterocolitis Model: The Model and Equation

\[ \frac{\partial e_c}{\partial t} = D \nabla \cdot \left( \left( \frac{e_c^2}{e_c^2 + (1-e_c)^2} \right) \nabla e_c \right) + k_p e_c (1-e_c) \]
Simplified Necrotizing Enterocolitis: Perfect Simulated Data

\[ k_p, D \text{ Estimate} \]

\[ \text{Time (hrs)} \]

\[ 0, 1, 2, 3 \]

\[ 1e-10, 1e-6 \]

\[ k_p, D \text{ Estimate Error} \]

\[ \text{Time (hrs)} \]

\[ 0.0001, 0.1 \]

\[ 0.6, 0.8, 1 \]

En KF; SC KF; KL SC KF.
Simplified Necrotizing Enterocolitis: Imperfect Simulated Data

$\text{En KF; SC KF; KL SC KF.}$
Simplified Necrotizing Enterocolitis: Real Data

En KF; SC KF; KL SC KF.
Are parameter estimates good?

- Produce qualitatively correct results
Are parameter estimates good?

- Produce qualitatively correct results
Are parameter estimates good?

• Produce qualitatively correct results
Are parameter estimates good?

- Produce qualitatively correct results
Are parameter estimates good?

- Produce qualitatively correct results
Are parameter estimates good?

- Produce qualitatively correct results
Are parameter estimates good?

• Produce qualitatively correct results
Are parameter estimates good?

• Produce qualitatively correct results
Conclusions/Future directions

• With perfect measurements and a pretty good model, SC does best, then KL, then En
• With imperfect measurements, all are comparable
• With real data, KL fails. Why? Guess: Too much error associated with $D$/results not sensitive to $D$
• Additional real data info
  – Gives temporal information about the parameters
  – Gives uncertainty estimates
  – Reveals info about suitability of the model
• Possibility for hybrid methods?
• All run significantly faster than the direct optimization method used