Data Assimilation Tutorial

A. M. Jorgensen
Electrical Engineering Department
New Mexico Tech
USA

Outline

- Bayesian formulation
- The original linear Kalman filter
- Example: Vehicle position determination
- Non-linear filtering
- Ensemble filtering
- Example: 1-dimensional model
- Ensemble Smoothing
- Example: 1-dimensional model
- Particle Filter
- Summary

What is Data Assimilation?

- We want to know the “state” of a “system.”
- The state could be time-evolving or not
- Could be a complex system:
  - The global weather pattern: winds, temperatures, pressures, etc.
- .. or it could be a simple system

The Use of Data Assimilation In Weather Forecasting

- To create a forecast we need:
  - Current state of the system
  - The expected evolution of drivers during the forecast period
  - A model for the time-evolution of the system based on the drivers
- Drivers might include
  - Solar power input
  - Tidal forces
  - .... others .... ?
- Drivers and the numerical model are "easy."
- The current state is "difficult" because it requires complete knowledge of all aspects of the system – or does it require that??

How to Get Current State of the System?

- Measurements
  - Usually incomplete
  - Can interpolate between measurement points
- A model
  - Run the model from far enough in the past such that initial conditions are not important
  - Use measurements of the drivers to run the model
  - “Butterfly” effect
  - Ignores knowledge about past state of the system which could actually constrain what the current state is, if we knew how to do it.
- The optimal is to use both observations and the model, and “tweak” the model – within limits – to make it agree with the observations

Firm Mathematical Footing

- We begin with a model

$$\psi = \{\psi_0, \ldots, \psi_k\}$$

- ... which evolves

$$\psi_i = f\left(\psi_{i-1}\right)$$

- .. or if we want to call the drivers out explicitly

$$\psi_i = f\left(\psi_{i-1}, q_i\right)$$

- It's the same thing, but sometimes it is easier to think of the drivers as external and sometimes as internal to the model.

What is ψ_i?

- For weather modeling it is grids of temperature, pressure, wind velocities, humidity, etc etc.
- Perhaps also an array of the drivers (which are inserted by the function f()), although that is not strictly necessary.

Bayesian Likelihood

(from Evensen, *Ocean Dynamics*, 53, 343-367, 2003)

- The probability distribution of the model given the (uncertain) observations

$$f\left(\psi|d\right) = \frac{f\left(d|\psi\right) f\left(\psi\right)}{\int \left(\ldots\right)\, d\psi}$$

- But where did the prior probability distribution of ψ come from?
- It comes from the fact that we acknowledge that the model is not exact and that some "tweaking" which is not described by the numerical equations of the model is allowed.
- We will return to this later.

Sequential Likelihood Evaluation

- Remember that

$$\psi = \{\psi_0, \ldots, \psi_k\}$$

- So we can re-write the previous equation as

$$f(\psi_0, \ldots, \psi_k | d_0, \ldots, d_k) = \frac{f(\psi_0, \ldots, \psi_k)\, f(d_1, \ldots, d_k | \psi_0, \ldots, \psi_k)}{\int (\ldots)\, d\psi}$$

- ... or as

$$f(\psi_0, \ldots, \psi_k | d_0, \ldots, d_k) = \frac{f(\psi_0)\, f(\psi_1 | \psi_0) \ldots f(\psi_k | \psi_{k-1}) f(d_1 | \psi_1) \ldots f(d_k | \psi_k)}{\int (\ldots)\, d\psi}$$

- ... when assuming independent measurement measurements

Sequential Likelihood Evaluation

- ... which can be decomposed sequentially like this

$$f\left(\psi_0, \psi_1 | d_1\right) = \frac{f(\psi_0) f(\psi_1|\psi_0) f(d_1|\psi_i)}{\int \left(\ldots\right) \, d\psi}$$

$$f\left(\psi_0, \psi_1, \psi_2 | d_1, d_2\right) = \frac{f(\psi_0) f(\psi_1|\psi_0) f(\psi_2|\psi_1) f(d_1|\psi_1) f(d_2|\psi_2)}{\int \left(\ldots\right) \, d\psi}$$

$$= \frac{f\left(\psi_0, \psi_1 | d_1\right) f(\psi_2|\psi_1) f(d_2|\psi_2)}{\int \left(\ldots\right) \, d\psi}$$

Sequential Likelihood Evaluation

- ... and in general

$$f\left(\psi_0, \ldots, \psi_k\right) = \frac{f(\psi_0, \ldots, \psi_{k-1}) f(\psi_k|\psi_{k-1}) f(d_k|\psi_k)}{\displaystyle\int (\ldots)\; d\psi}$$

- The conclusion is that Bayesian estimation can be decomposed sequentially.
  - We do not need to fit the entire time-sequence simultaneously.
  - Instead we can run the model forward one step at a time and incorporate observations multiplicatively.
  - The latter is much simpler.
  - Step-at-a-time assimilation is MUCH simpler than assimilating to an entire time sequence at once.

The State Transition Probabilities

- We still need to understand the state transition probabilities

$$f(\psi_k|\psi_{k-1})$$

- ... because the model is supposed to be exact, so why are there probabilities involved?
- Because instead of the model evolution

$$d\psi = f(\psi)dt$$

  we should imagine a stochastic evolution - because the model is not exact -

$$d\psi = f(\psi)dt + g(\psi)dq$$

  where dq is a random variable
- Obviously the model probability will diverge in this case unless we apply constraints – from observations.

The Original Kalman Filter

- Swerling (1958) and Kalman (1960) developed a formulation for linear systems. Estimate state based on measurements of observables, allowing for inaccurate model and measurement noise

Driver effect

External driver

System evolution

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Model noise

System state

$$y_k = Cx_k + z_k$$

Measurement noise

Measurement

Measurement operator

The Original Kalman Filter

- We can form estimates $\hat{x}_k$ and P_k of the mean and covariance of the state.

$$K_k = AP_kC^T\left(CP_kC^T + S_z\right)^{-1}$$

$$\rightarrow \quad \hat{x}_{k+1} = \left(A\hat{x}_k + Bu_k\right) + K_k\left(y_{k+1} - C\hat{x}_k\right)$$

$$\rightarrow \quad P_{k+1} = AP_kA^T + S_w - AP_kC^TS_z^{-1}CP_kA^T$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \qquad P = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1m} \\ \sigma_{21} & \sigma_2^2 & & \sigma_{2m} \\ \vdots & & \vdots & \\ \sigma_{m1} & \sigma_{m2} & \dots & \sigma_m^2 \end{bmatrix}$$

Example: vehicle position estimation

(adapted from D. Simon, *Embedded Systems Programming*, June 2001)

- Estimate "state" of a vehicle. State is 1-dimensional [position,velocity]
- We know the accelerator position, but external factors like wind, potholes, and other factors modify the actual effect
- We measure position, but it is not accurate
- If we use only measured position and ignore model we get positions, but with large uncertainties
- If we use only model and ignore measured position it will diverge because the model is not accurate (butterfly effect)
- The Kalman filter combines observations and model in an optimal way under certain assumptions

Example: vehicle position estimation

(adapted from D. Simon, *Embedded Systems Programming*, June 2001)

- Equation for velocity, including model noise

$$v_{k+1} = v_k + Ta + \tilde{v}_k$$

Model noise

Time step

Acceleration

- Equation for position including model noise

$$p_{k+1} = p_k + Tv_k + \frac{1}{2}T^2a + \tilde{p}_k$$

Model noise

State Equations

$$x_k = \left[\begin{array}{c} p_k \\ v_k \end{array} \right] \qquad w_k = \left[\begin{array}{c} \tilde{p}_k \\ \tilde{v}_k \end{array} \right]$$

$$x_{k+1} = \left[\begin{array}{cc} 1 & T \\ 0 & 1 \end{array} \right] x_k + \left[\begin{array}{c} \frac{1}{2}T^2 \\ T \end{array} \right] a + w_k$$

$$y_k = \left[\begin{array}{cc} 1 & 0 \end{array} \right] x_k + z_k$$

Pick Some Numbers

- Measurements uncertainty
  - Position measurements uncertainty: 10
- Model noise
  - Assume constant acceleration of 1 with noise of 0.2. that means, we are confident in the model to within an acceleration of 0.2, or 20%. That means that if we don't adjust the difference between model and reality will grow very big very quickly. After 10 s the model error will exceed the measurement error.
- Based on the above we can compute what should be the the covariance of model and measurement noise.

$$S_w = \begin{bmatrix} 1 \times 10^{-6} & 2 \times 10^{-5} \\ 2 \times 10^{-5} & 4 \times 10^{-4} \end{bmatrix} \qquad S_z = [100]$$

Kalman Filter Equations

$$S_z = [100]$$

$$K_k = AP_kC^T\left(CP_kC^T + S_z\right)^{-1}$$

$$\hat{x}_{k+1} = (A\hat{x}_k + Bu_k) + K_k\left(y_{k+1} - C\hat{x}_k\right)$$

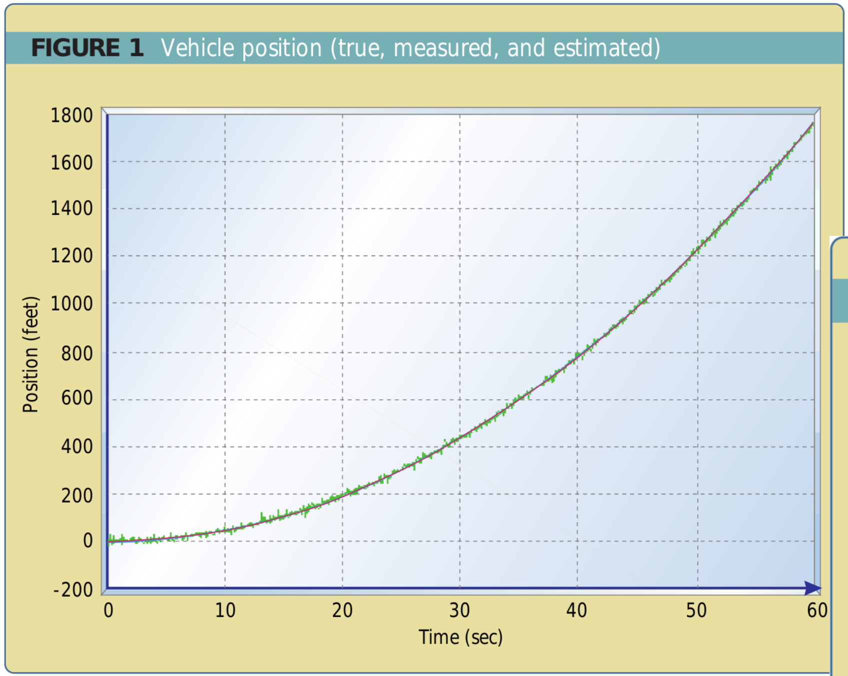
$$P_{k+1} = AP_kA^T + S_w - AP_kC^TS_z^{-1}CP_kA^T$$

$$S_w = \begin{bmatrix} 1\times10^{-6} & 2\times10^{-5} \\ 2\times10^{-5} & 4\times10^{-4} \end{bmatrix}$$

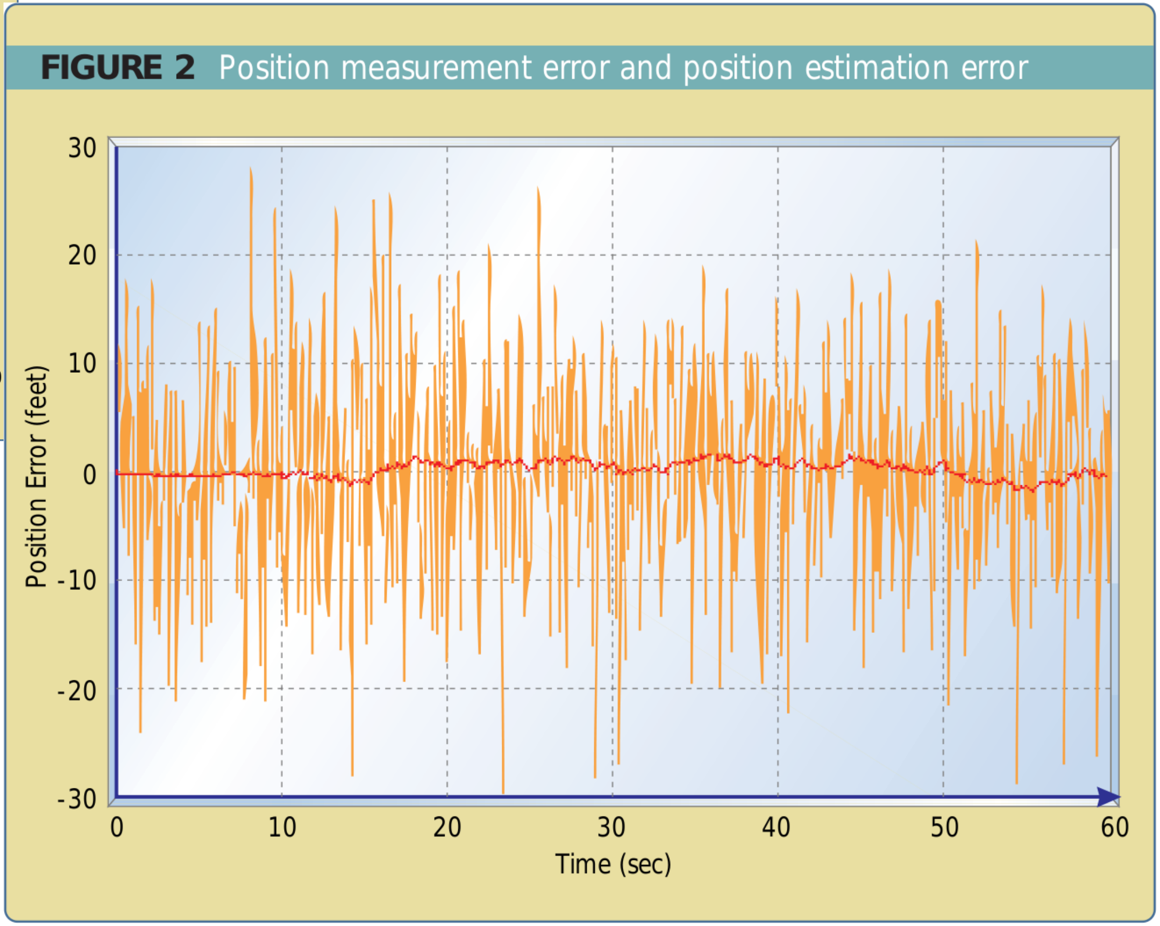
If S_z is large then K_k is small, and model evolution dominates. Also, uncertainty will grow.

If S_z is small then K_k is large and measurement dominates. Also, uncertainty decreases.

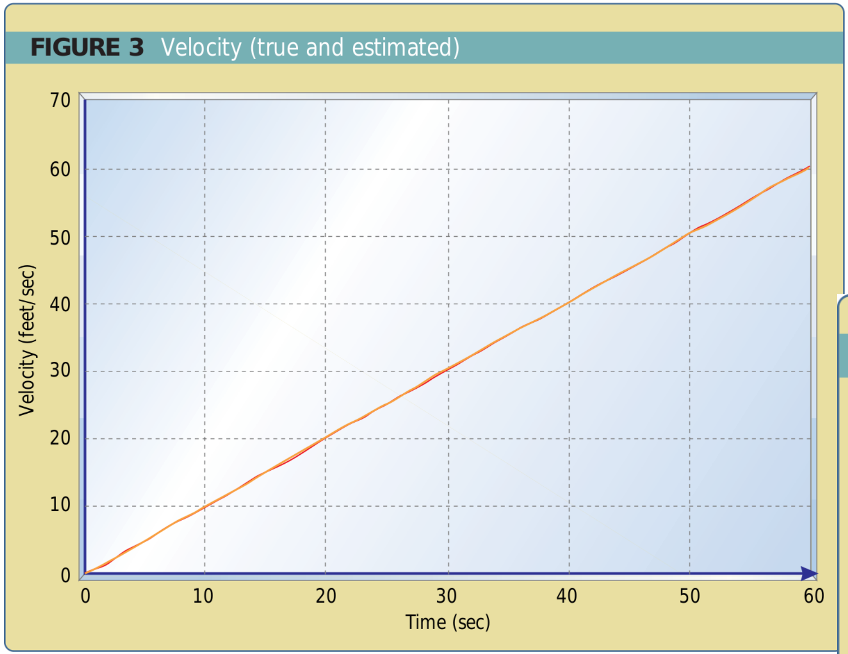
Position Estimate

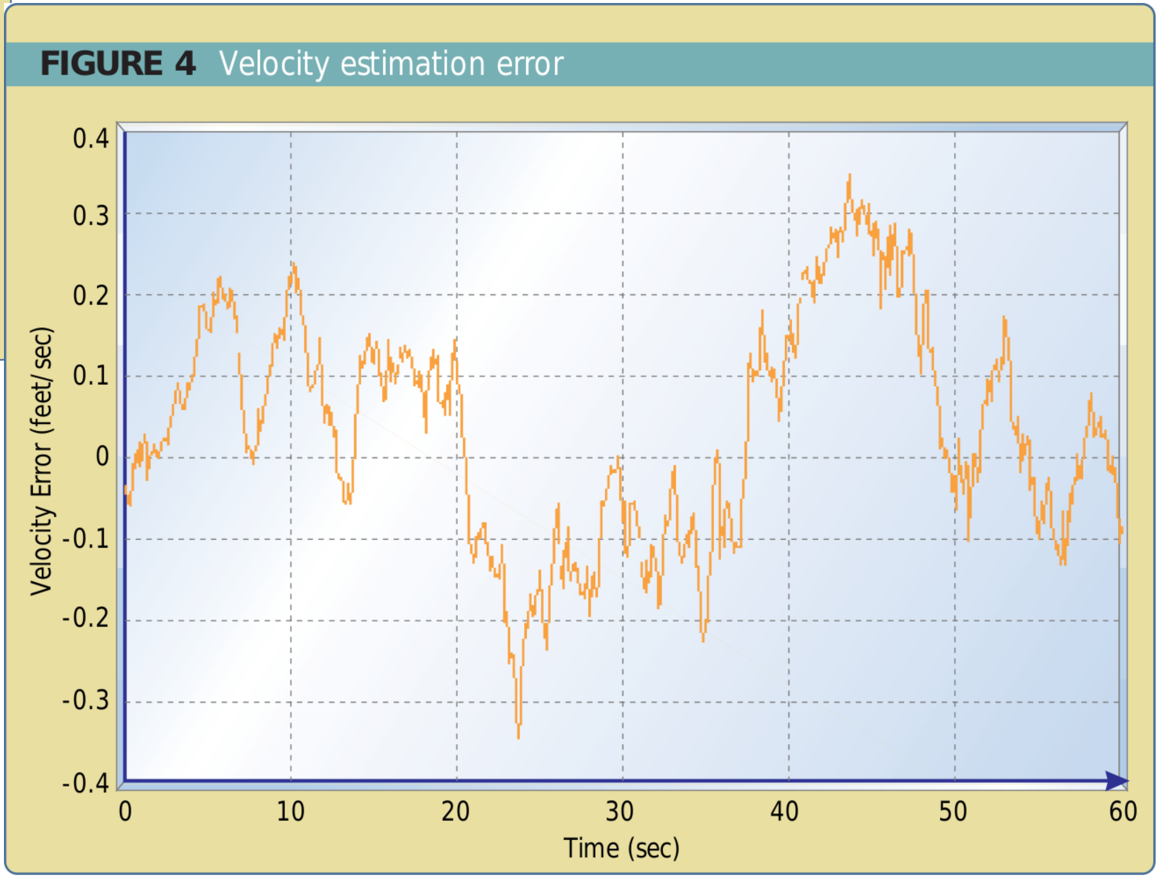


The estimation error is much smaller than the measurement error because of the use of the model.



Velocity Estimate





What's the point?

- The point is that if
  - You trust the model fairly well short-term but worry about “butterfly” effect
  - You have uncertain and incomplete measurements
- Then
  - Driving the model with observations improves the estimate of the state
- But isn't that just the same as smoothing?
  - Yes
  - ... smart smoothing and interpolation!
  - When you smooth you assume that all consecutive values are the same
  - With data assimilation you can apply nearly arbitrarily smart smoothing/interpolation algorithms

Non-Linear Filtering

- Advance model according to actual equations, and linearize around the state.
- The linearization is done by replacing, in the coavariance matrix estimation, A, with the Jacobian of the state advance function f.

$$P_{k+1} = AP_kA^T + S_w - AP_kC^TS_z^{-1}CP_kA^T$$

$$A = \begin{bmatrix} \frac{\partial x_1^+}{\partial x_1} & \cdots & \frac{\partial x_1^+}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_m^+}{\partial x_1} & \cdots & \frac{\partial x_m^+}{\partial x_m} \end{bmatrix}$$

- This is called Extended Kalman Filtering, EKF
- Useful because most interesting models are non-linear
- Requires that the variances are small enough that the linear approximation of the Jacobian is valid.

Problem: Size and Computation

- For a small state of two numbers, like velocity and position in the problem we just saw, the Kalman filter, or even Extended (linearized) Kalman filtter, work well.
- For large problems like the plasmasphere they do not. The state for the default version of DGCPM is 40000 grid points.
- This results in a covariance matrix with 1.6 x 10^9 elements.
- Storage is not as much the issue as is the computational burden.
- And do we really need ALL covariance terms?

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$P = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1m} \\ \sigma_{21} & \sigma_2^2 & & \sigma_{2m} \\ \vdots & & \vdots & \\ \sigma_{m1} & \sigma_{m2} & \dots & \sigma_m^2 \end{bmatrix}$$

Ensemble Filtering

- There is a better way. Run a smaller number of models in parallel, each evolving with some random contribution while preserving mean and covariance of the ensemble.
- If we chose a ensemble size of 1000, then instead of a 40000 x 40000 covariance matrix we have a 40000 x 1000 ensemble matrix.

$$\boldsymbol{A} = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1N} \\ \psi_{21} & \psi_{22} & & \psi_{2N} \\ \vdots & & \ddots & \vdots \\ \psi_{m1} & \psi_{m2} & \cdots & \psi_{mN} \end{bmatrix}$$

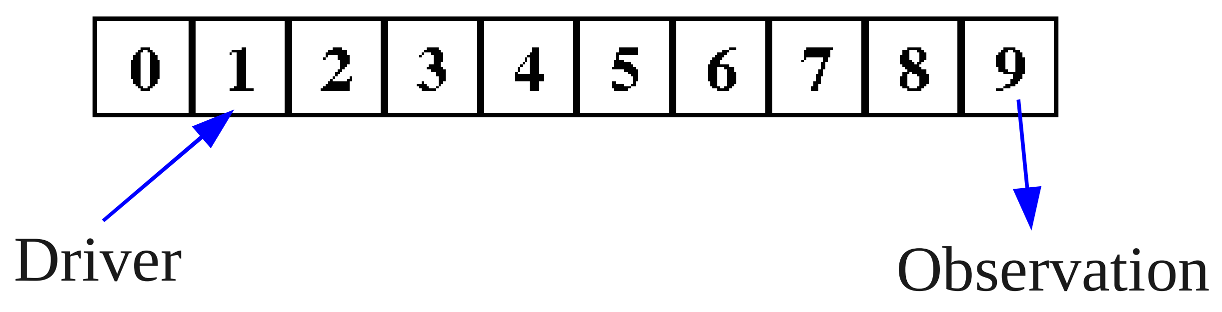
- The posterior ensemble is now a linear combination of the prior ensemble, again such that it preserves the mean and variance of the original Kalman Filter

$$\boldsymbol{A}_{\text{posterior}} = \boldsymbol{A}_{\text{prior}} \times \boldsymbol{X}$$

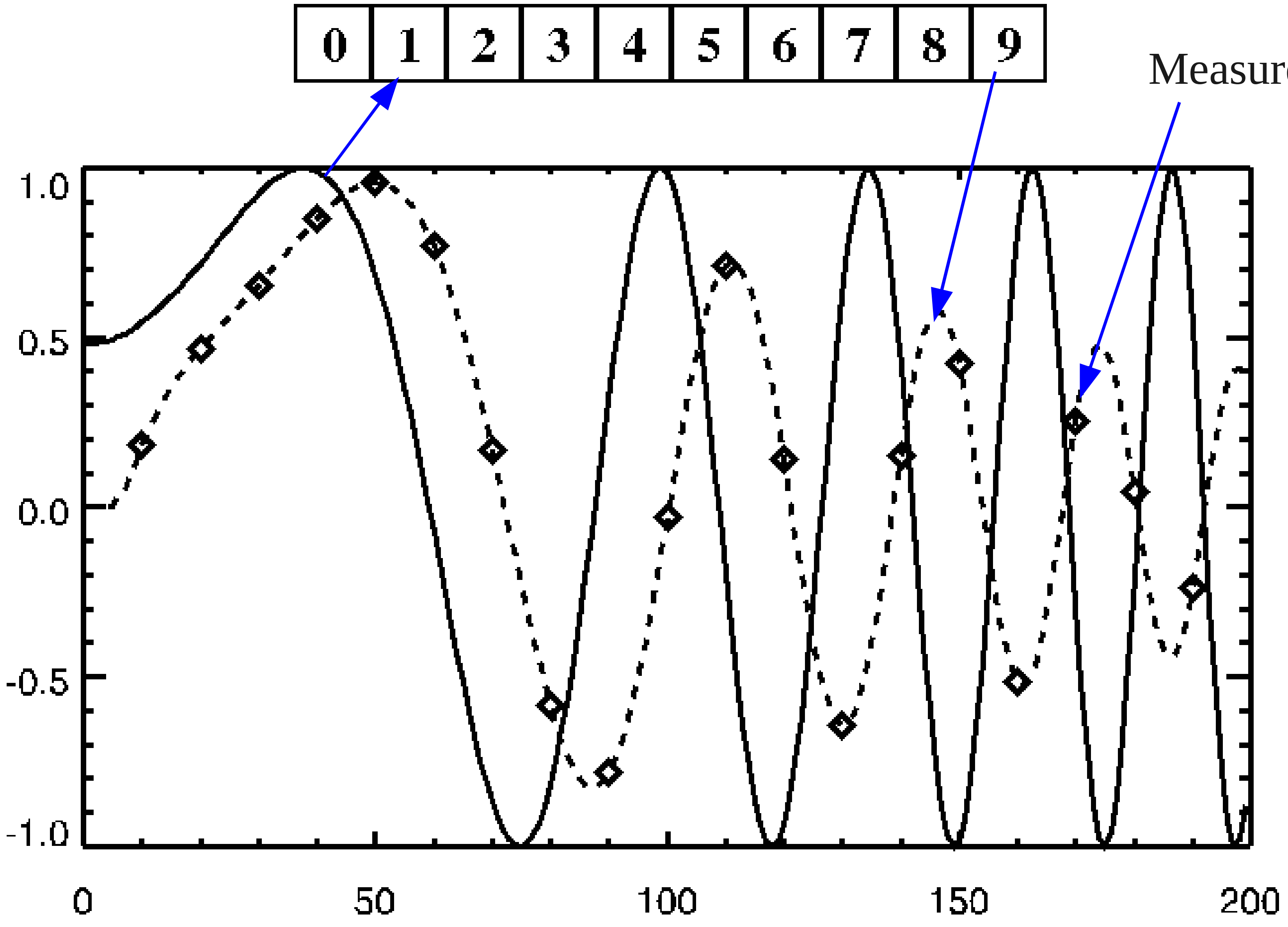
- The computational burden is reduced by orders of magnitude.
- But we need to understand how to generate the randomness.

Example: 1-dimensional model

- A convection-like model:
- Model state $\bar{x}_k = [x_{k,0}, x_{k,1}, \ldots, x_{k,9}]$
- Model equations: $\bar{x}_{k+1} = f(\bar{x}_k) \quad x_{k+1,i} = \dfrac{3\, x_{k,i-1} + x_{k,i+1}}{4}$
- Driver: $x_{k+1,1} = q_{k+1}$
- Observation: $x_{k,9}$



1D Model Simulation



What is the Question?

- We can now ask several questions:
  - Given that we know the model equations, how well can we determine the state of the system based on only the noisy intermittent observations at x_9?

$$x_{k+1,i} = \frac{3\,x_{k,i-1} + x_{k,i+1}}{4}$$

  - .... or... Given that we know the model equations, how well can we determine the driver based only on the noisy intermittent observations at x_9?

$$x_{k+1,1} = q_{k+1}$$

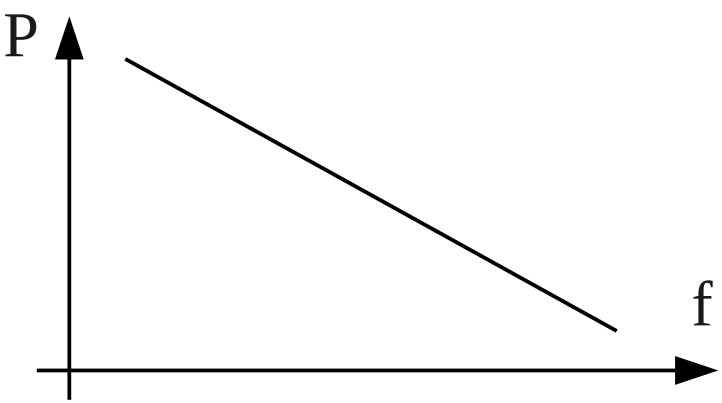
Model Noise

- This is what makes the models diverge to explore parameter space (parameter space is the driver)
- Parameter space is in this case just the behavior of the driver as a function of time
- Red noise: $$q_{k+1} = \alpha\, q_k + \sqrt{1-\alpha^2}\, w_k$$

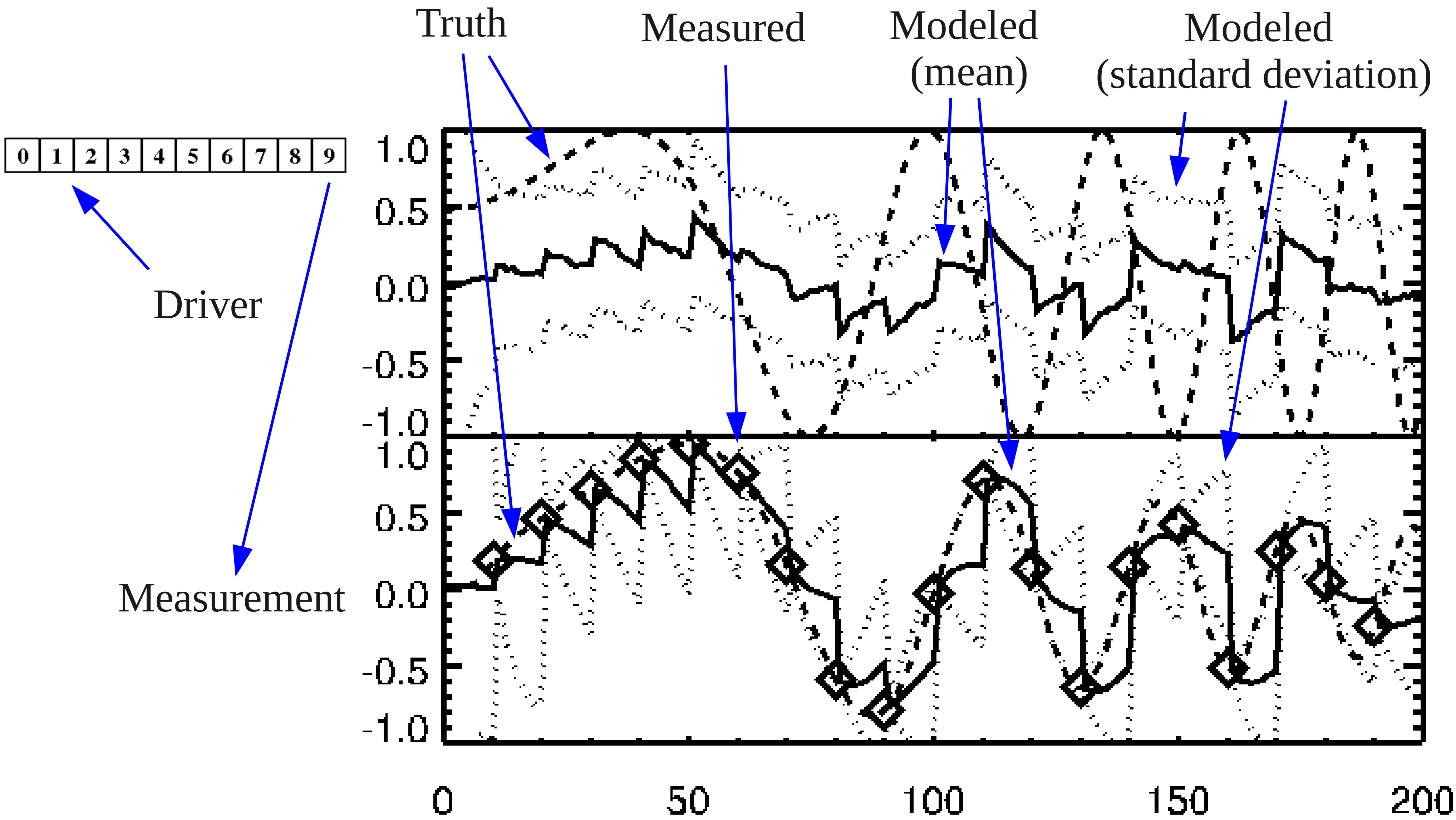
$$\alpha = \frac{1}{\tau}$$

Time constant

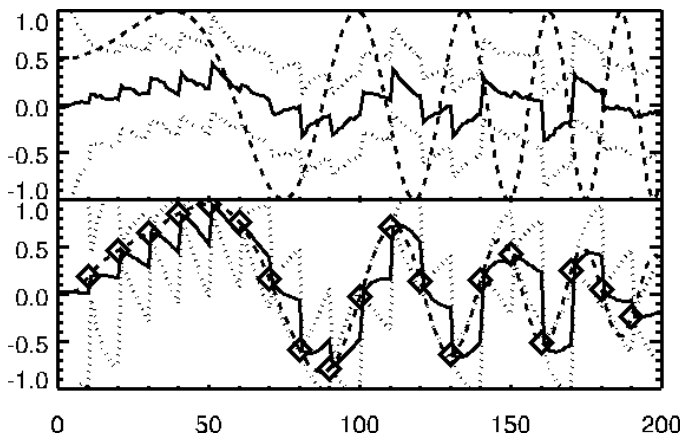
Random noise



Assimilation



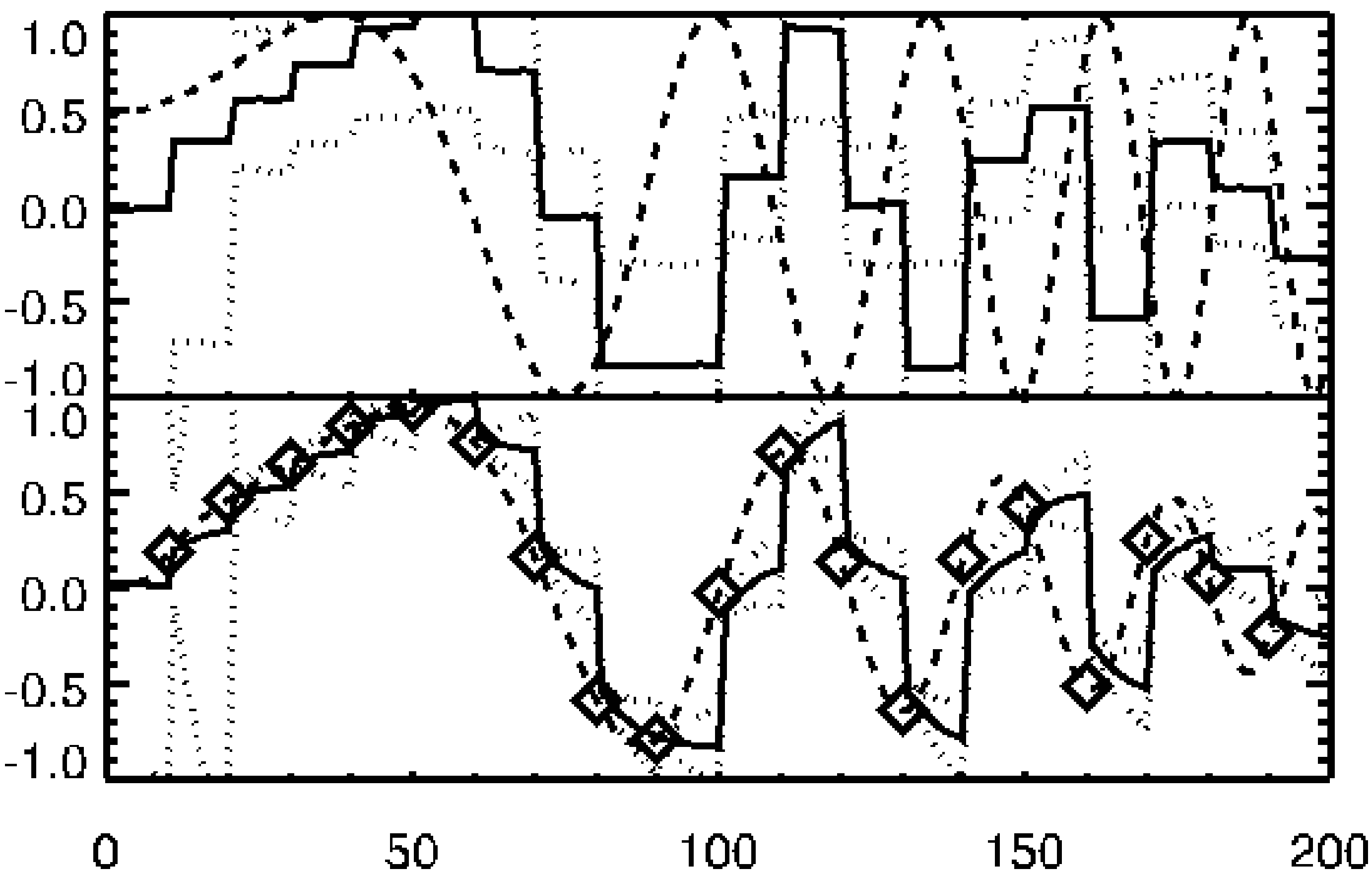
Discussion

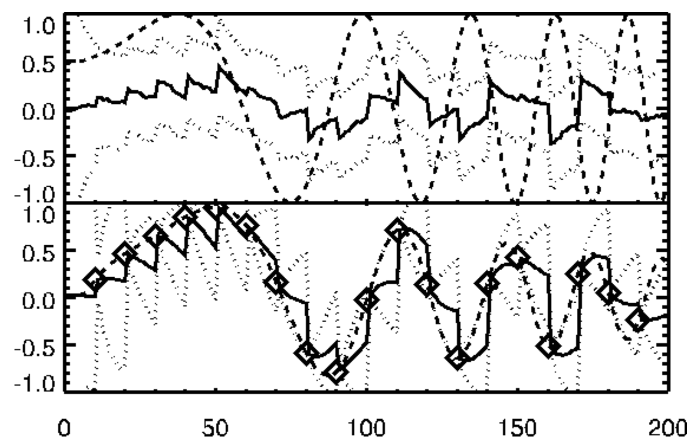


- Between measurements the model ensemble diverges.
- Measurements noise is much smaller than the chosen model noise.
- At the time of measurement the ensemble is reset tightly around the measurement, thus giving rise to discontinuities.
- But the extrapolation beyond the last data point is still “optimal” in the absence of future measurements.
- In a moment we will return to using future measurements to do even better.
- But first let's try a different model noise....

A Different Model Noise

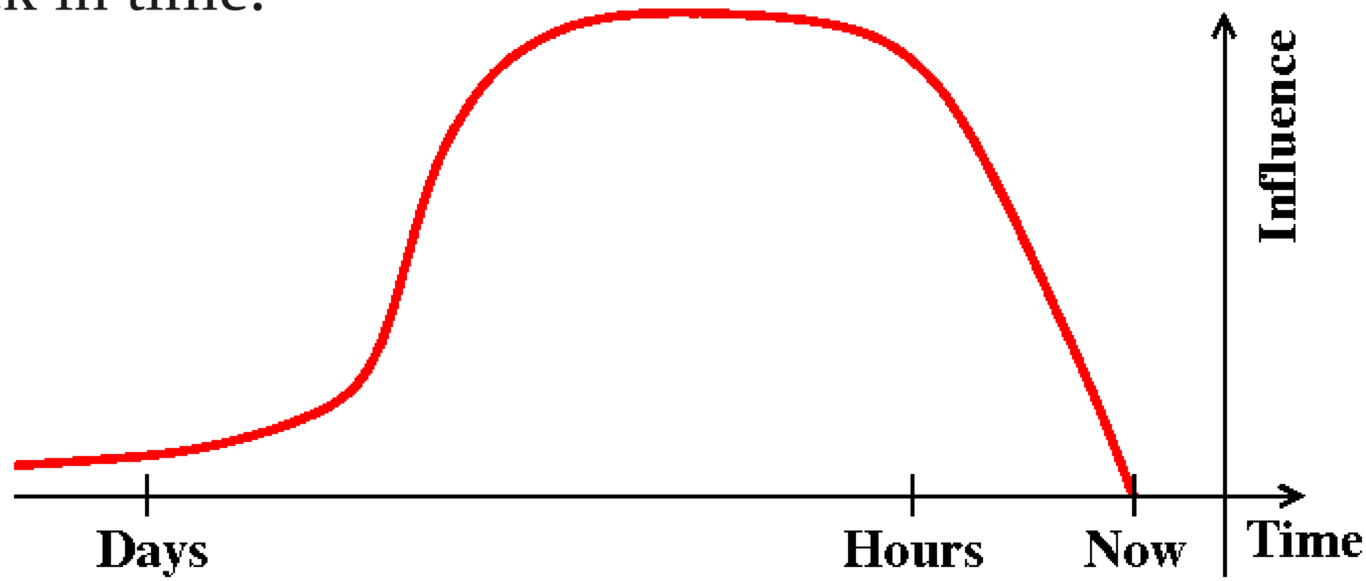
$$\sout{q_{k+1} = \alpha\, q_k + \sqrt{1-\alpha^2}\, w_k} \;\rightarrow\; q_{k+1} = q_k + \alpha\, w_k$$





Ensemble Smoothing

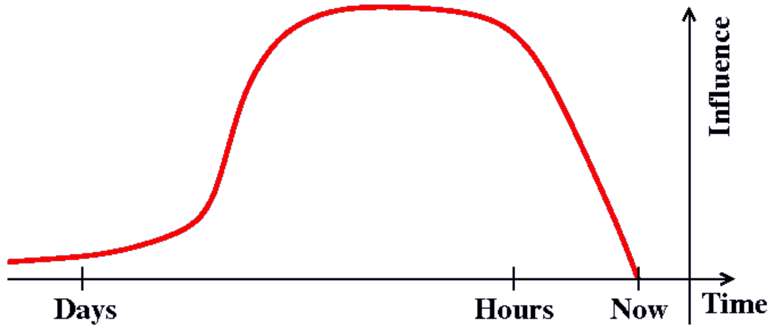
- One thing missing from the KF and EnKF formulations (but not from the original Bayesian formulation) is inclusion of future data.
- Data constrain the model best a bit back in time, but not too far back in time.



- The two KF and EnKF do not take advantage of that

Ensemble Smoothing

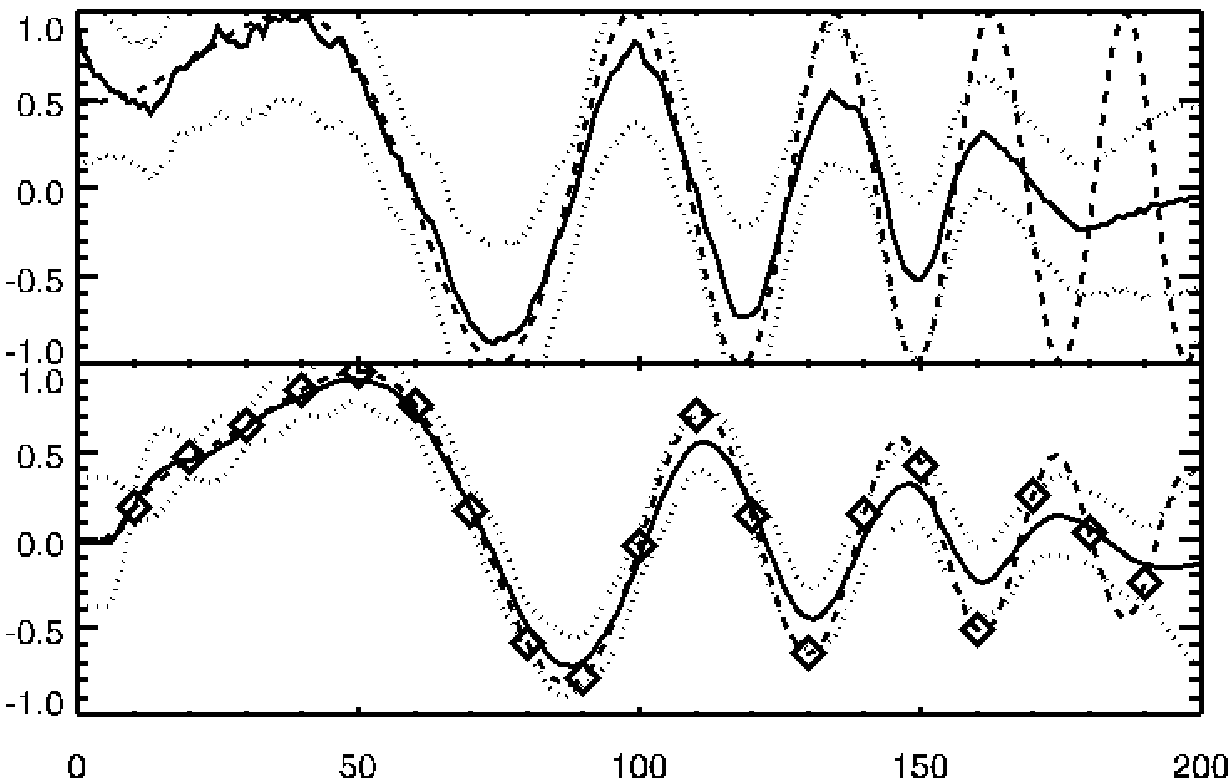
- Ensemble Kalman Smoother – EnKS
- The transformation we apply each time data are available should also be applied to past states. We can use present data to improve the prediction from the distant past up to the present.
- But as we go very far back in time applying the transformation will not change the mean and covariance of the state.

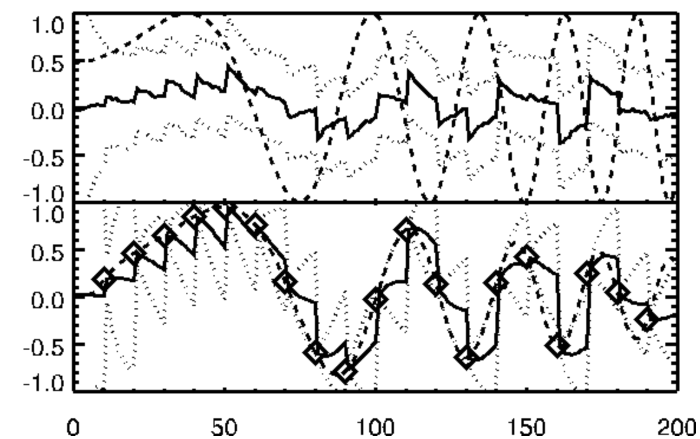


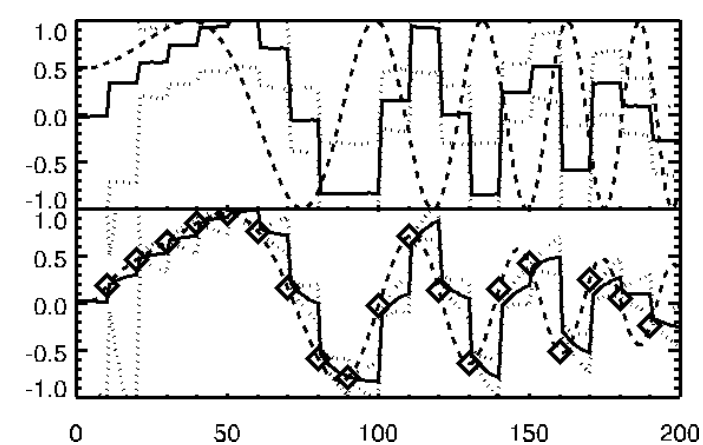
- The distant past states are random with respect to the current transformation, so no change.
- Here is how the EnKS looks......

Example: 1-dimensional model

- (WOW!) Note that present observations can be used to improve past states, but that this does not change/improve present states – for that future data are needed







Particle Filter

- In the KF, EnKF, and EnKS, data are assimilated by forming new states which are linear combinations of the states before data are introduced.
- This assumes linearity which is not usually correct.

$$\psi_c = \alpha \, \psi_a + \beta \, \psi_b$$

- The particle filter does not assume linearity. Here new states are created by statistically picking the best of the existing states to continue to run and letting the worst states die.
- But.. in order to have enough states to pick from it becomes necessary to have a far larger ensemble – perhaps 10-100 times larger(?)

Summary

- We started by formulating a Bayesian approach to combining model and observations.
- Then we looked at the Kalman Filter which is based on it, and found that it works best for small problems
- Then we briefly looked at dealilng with non-linearities (the Extended Kalman Filter)
- Next, we looked at how to deal with large problems with the Ensemble Kalman Filter
- ... and how the smoothing filter improves retrospective analysis
- And finally we touched briefly on the particle filter, which can be fully non-linear and non-Gaussian.