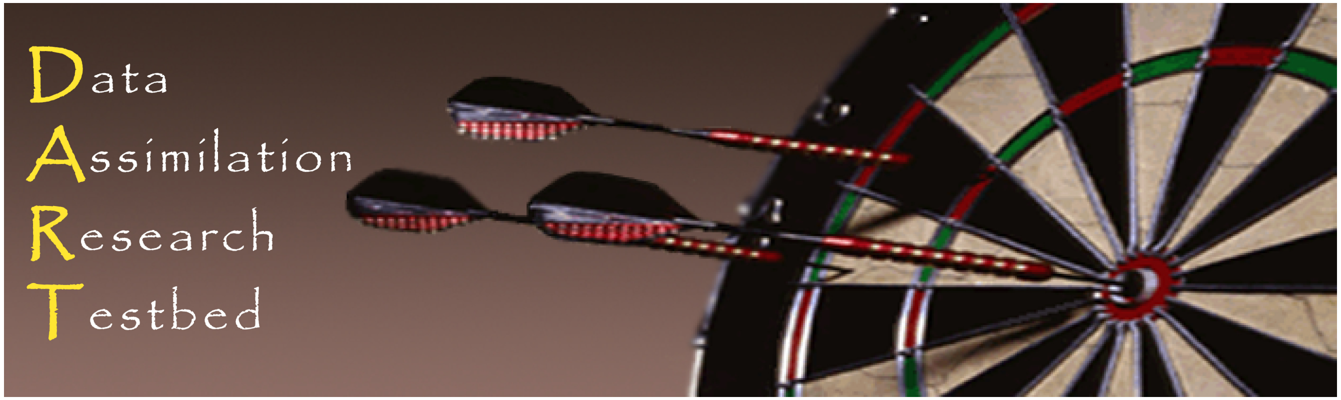
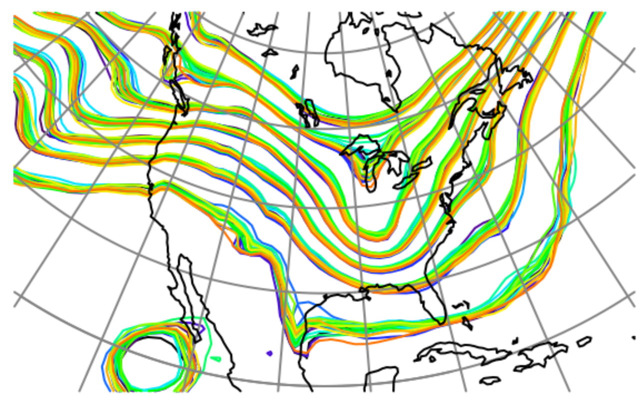


Introduction to Data Assimilation

Jeff Anderson representing the NCAR Data Assimilation Research Section



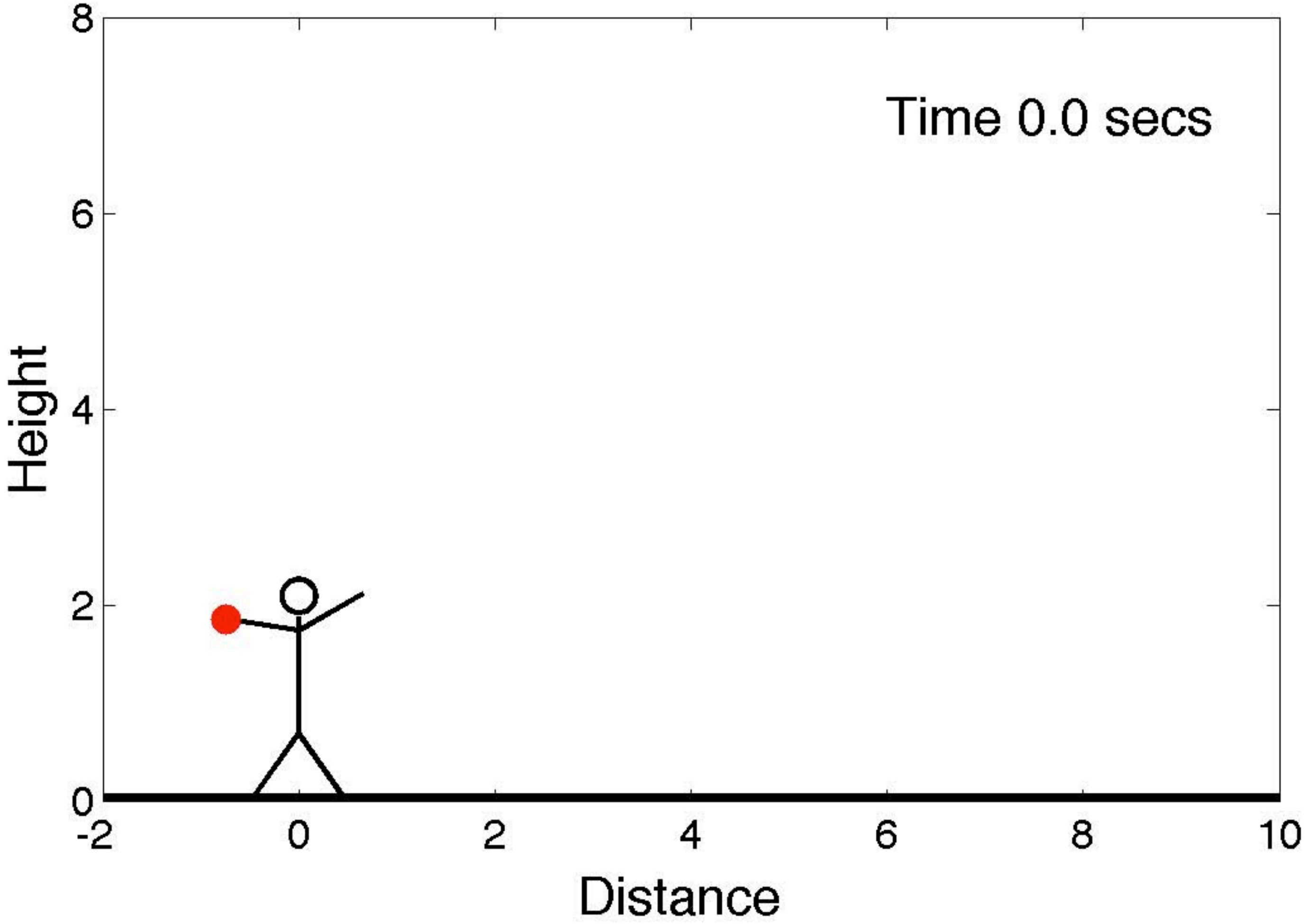
The National Center for Atmospheric Research is sponsored by the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

NCAR | National Center for
UCAR | Atmospheric Research

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

Want to predict where the ball will land.



Data Assimilation: Building a simple forecast system

Prediction Model

Data Assimilation: Building a simple forecast system

Prediction Model

For the ball this is simple:

$$x = x_{initial} + u_{initial} t$$

$$y = y_{initial} + v_{initial} t - 1/2\, g t^2$$

Data Assimilation: Building a simple forecast system

Prediction Model

For the ball this is simple:

$$x = x_{initial} + u_{initial} t$$

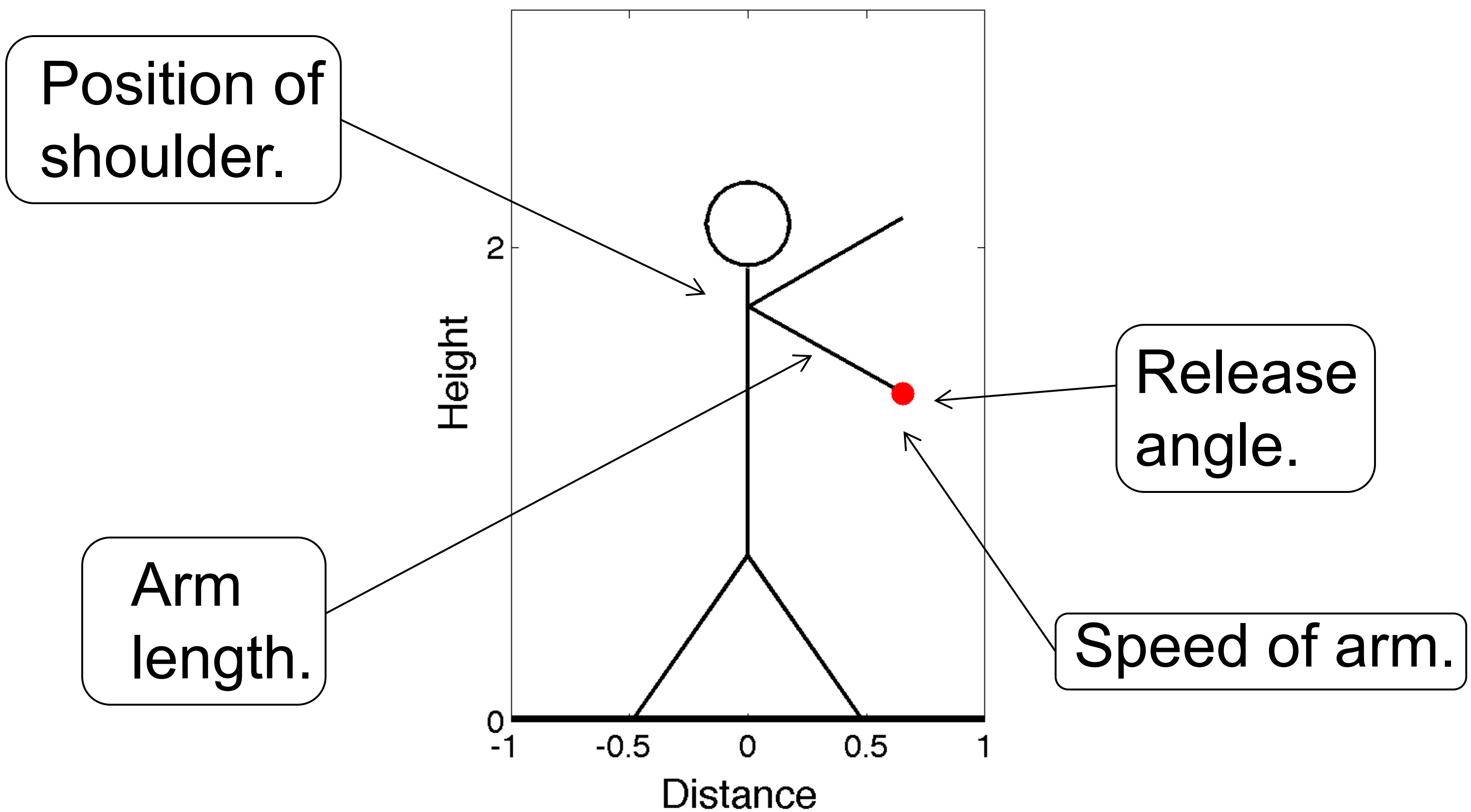
$$y = y_{initial} + v_{initial} t - 1/2\, g t^2$$

However, may be uncertain about initial conditions:

$x_{initial}$, $y_{initial}$, $u_{initial}$ and $v_{initial}$

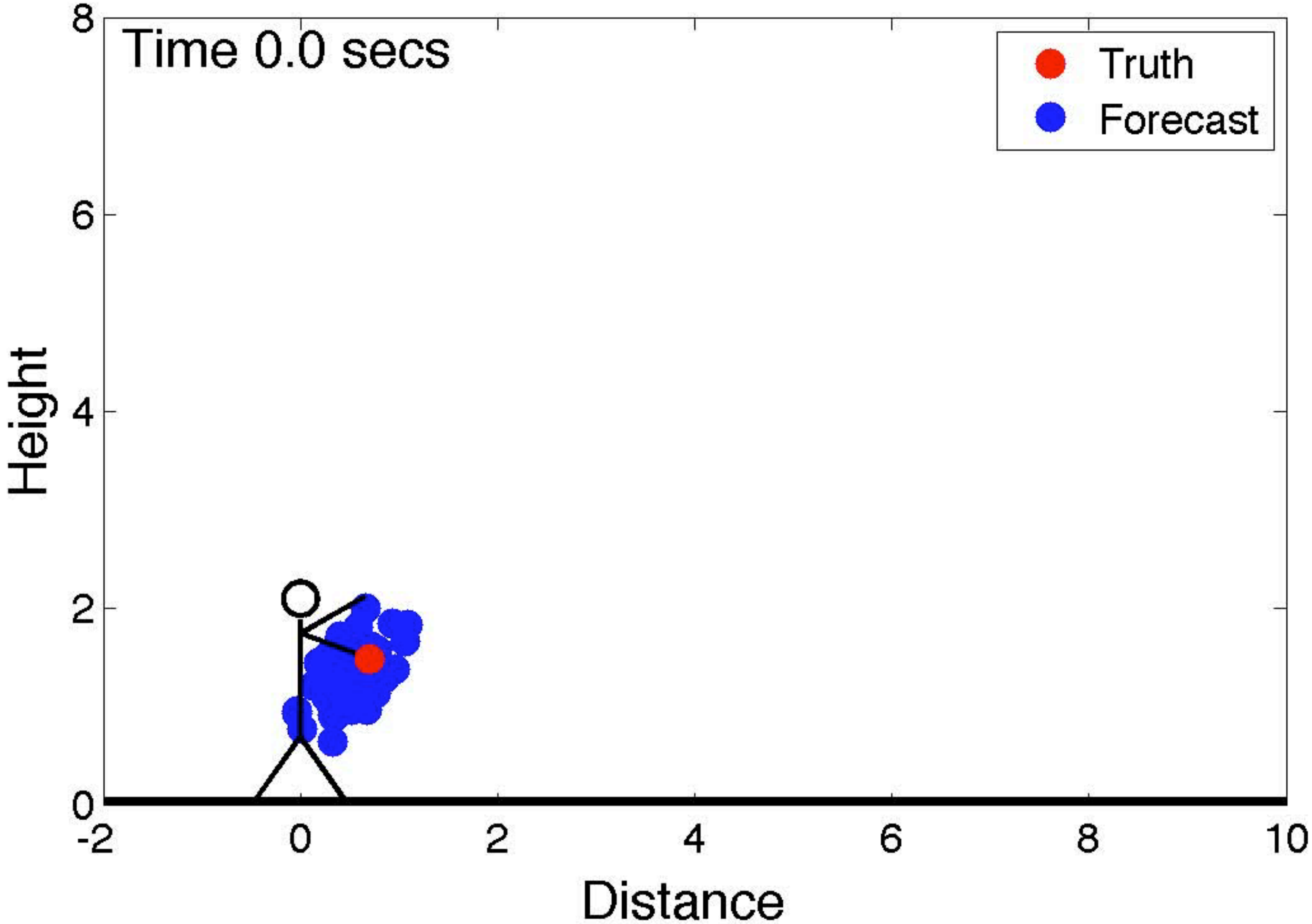
Data Assimilation: Building a simple forecast system

Only know probability distribution about the throw.
Uncertainty in these leads to uncertainty in x, y, u, v.



Data Assimilation: Building a simple forecast system

Sample this with an ‘ensemble’ of balls with random draws from the thrower.



Data Assimilation: Building a simple forecast system

Prediction Model

Observing System

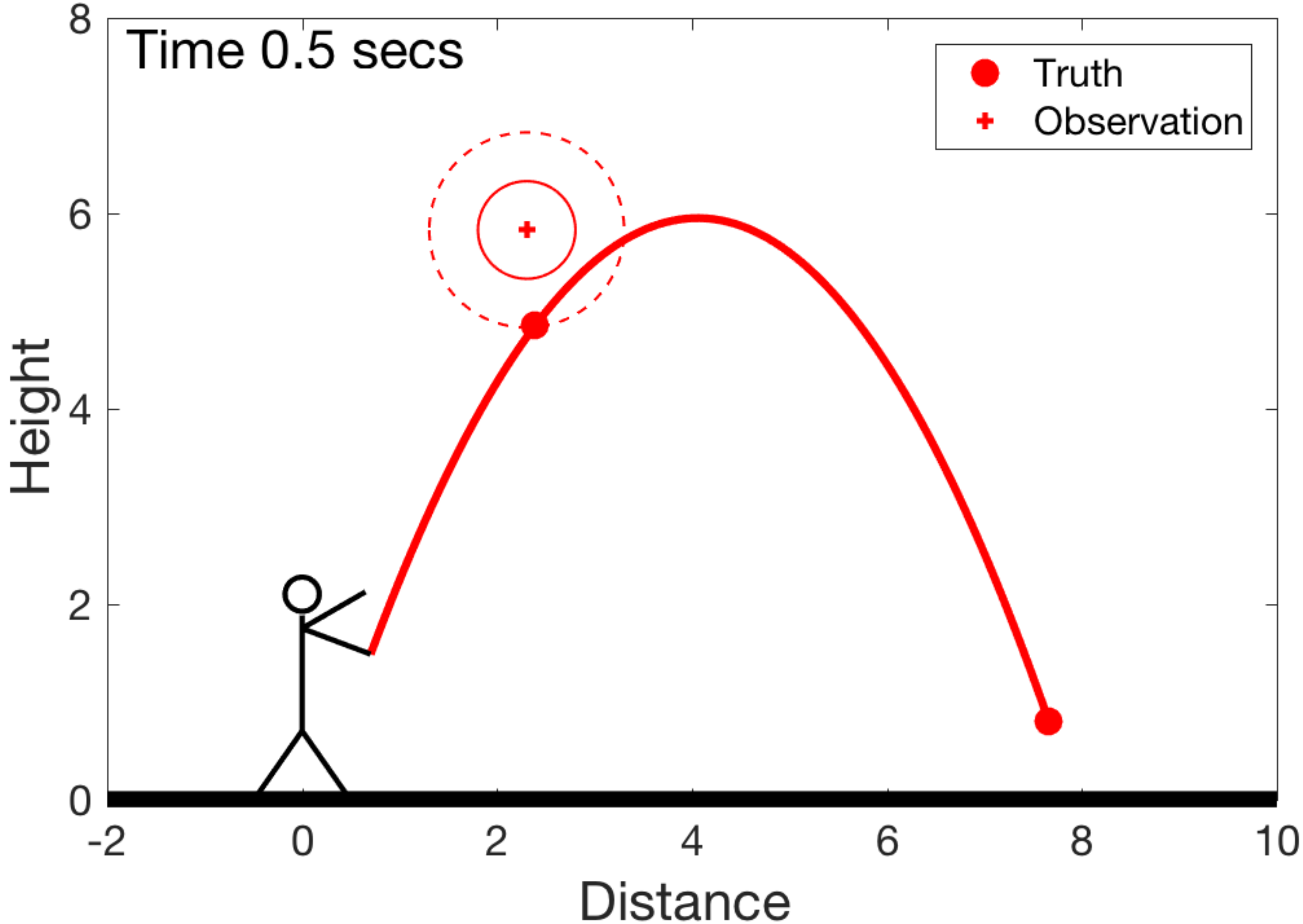
Need observations (measurements) of the red ball.

All observations have errors.

Observe position of ball every half second after throw.

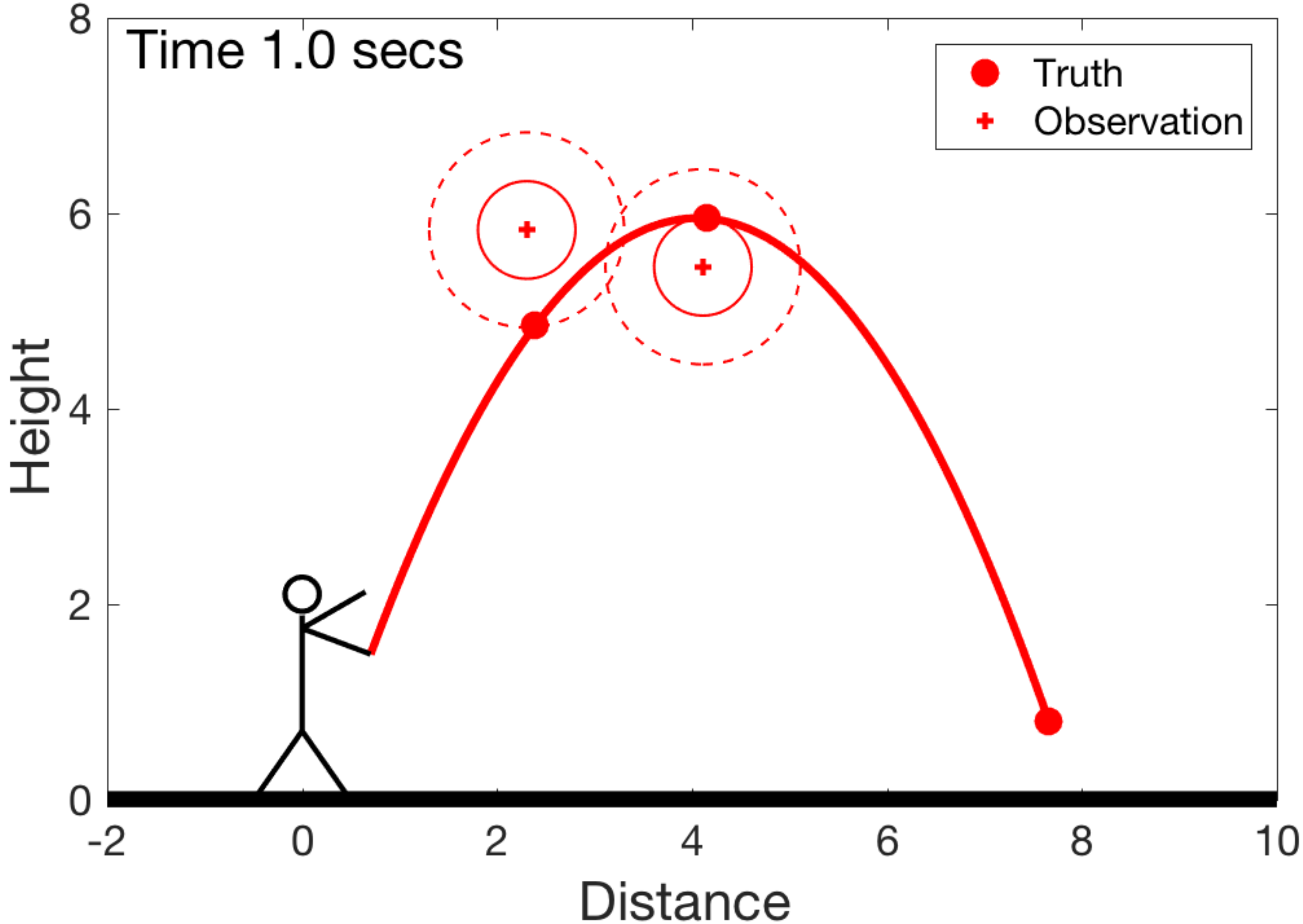
Data Assimilation: Building a simple forecast system

Observations of the red ball.



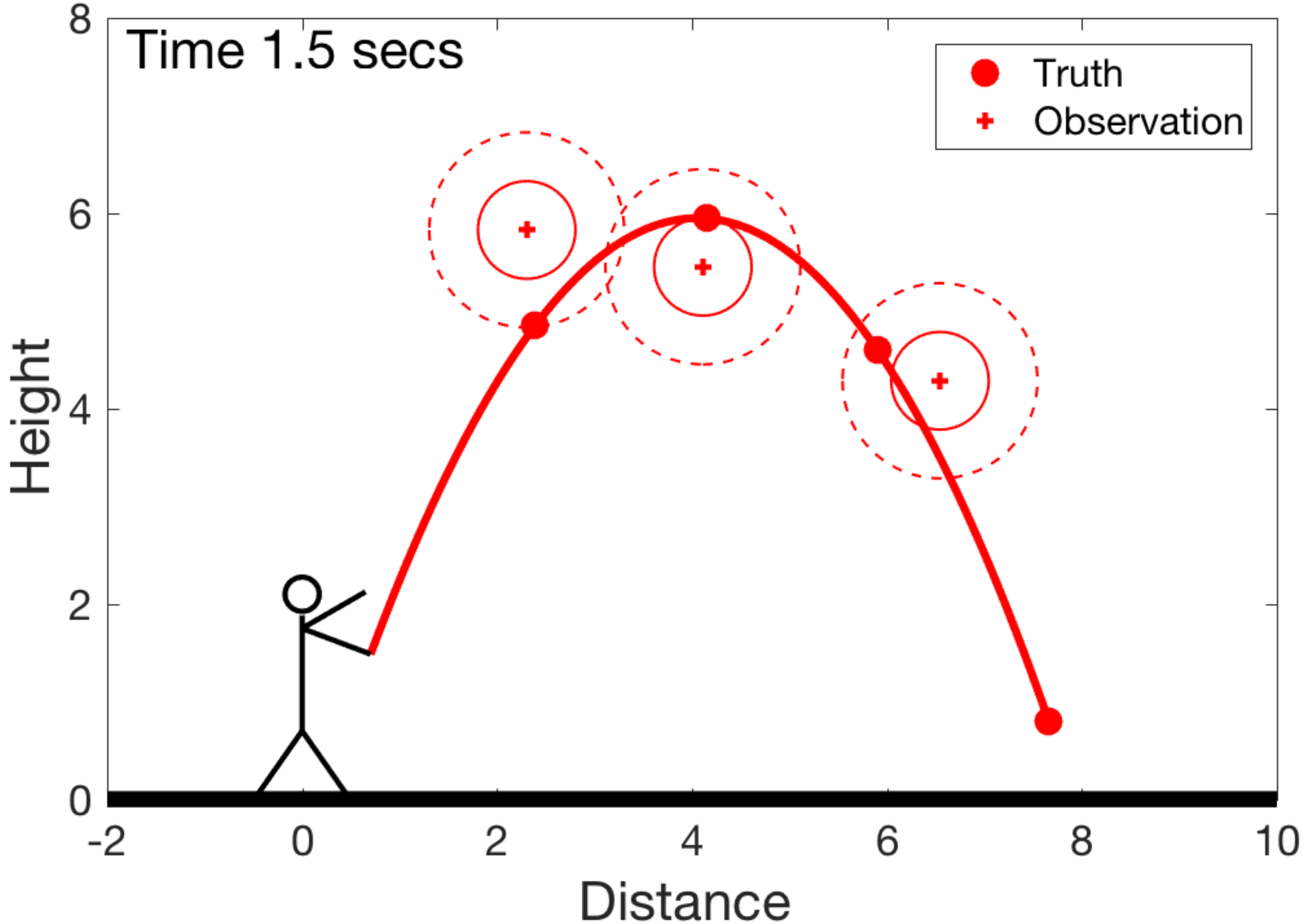
Data Assimilation: Building a simple forecast system

Observations of the red ball.



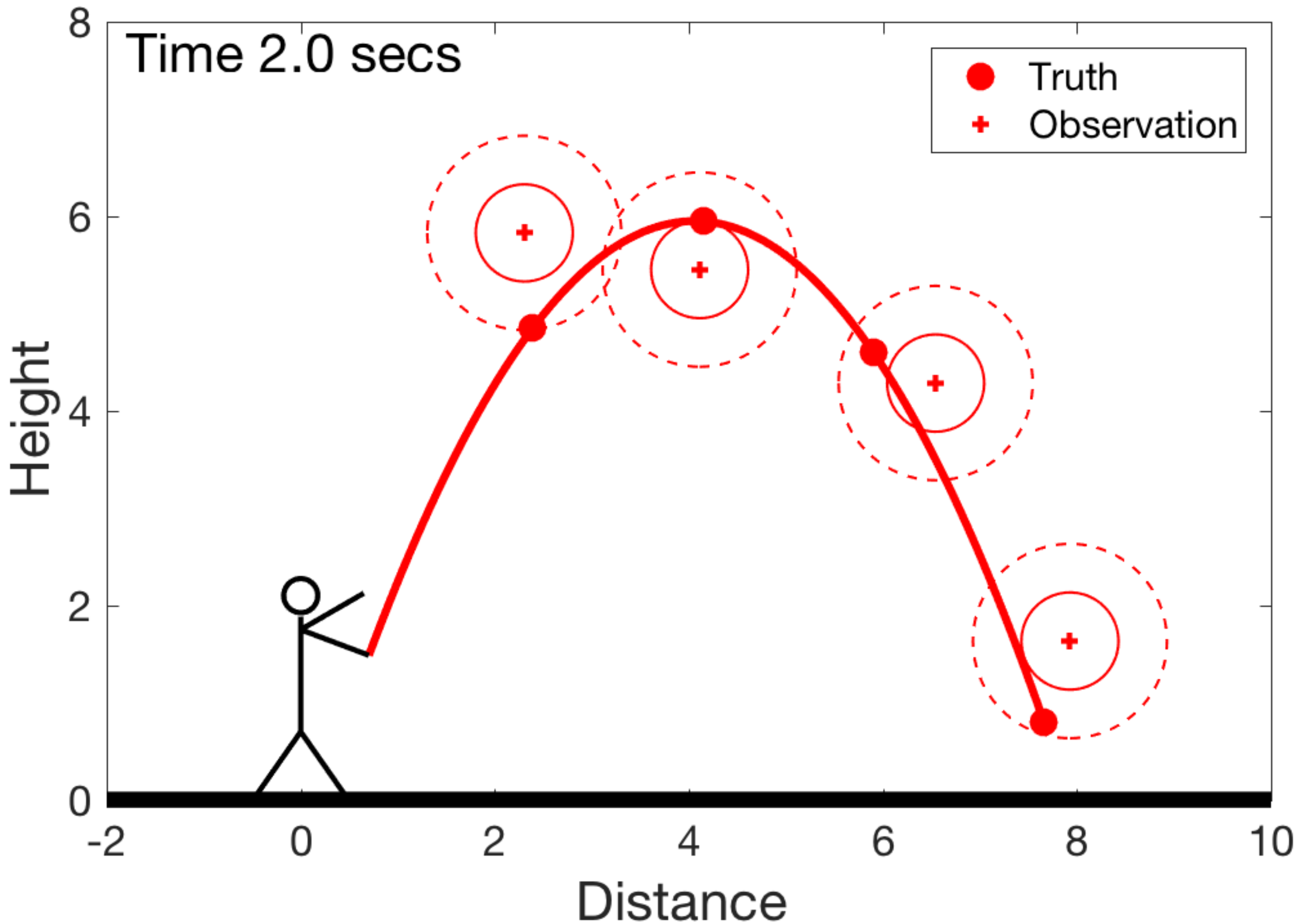
Data Assimilation: Building a simple forecast system

Observations of the red ball.



Data Assimilation: Building a simple forecast system

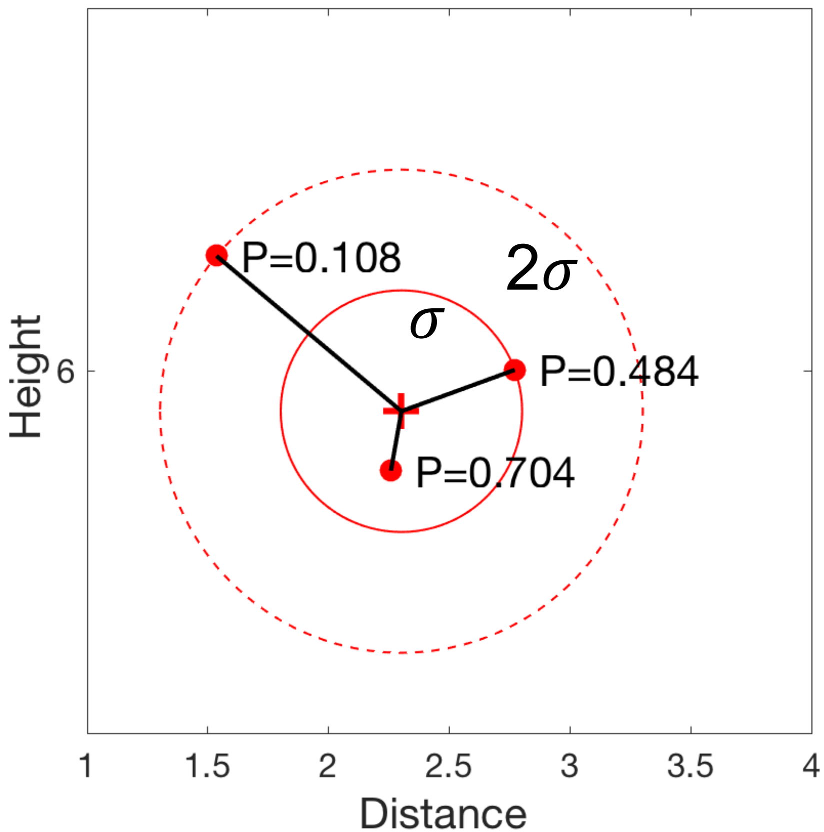
Observations of the red ball.



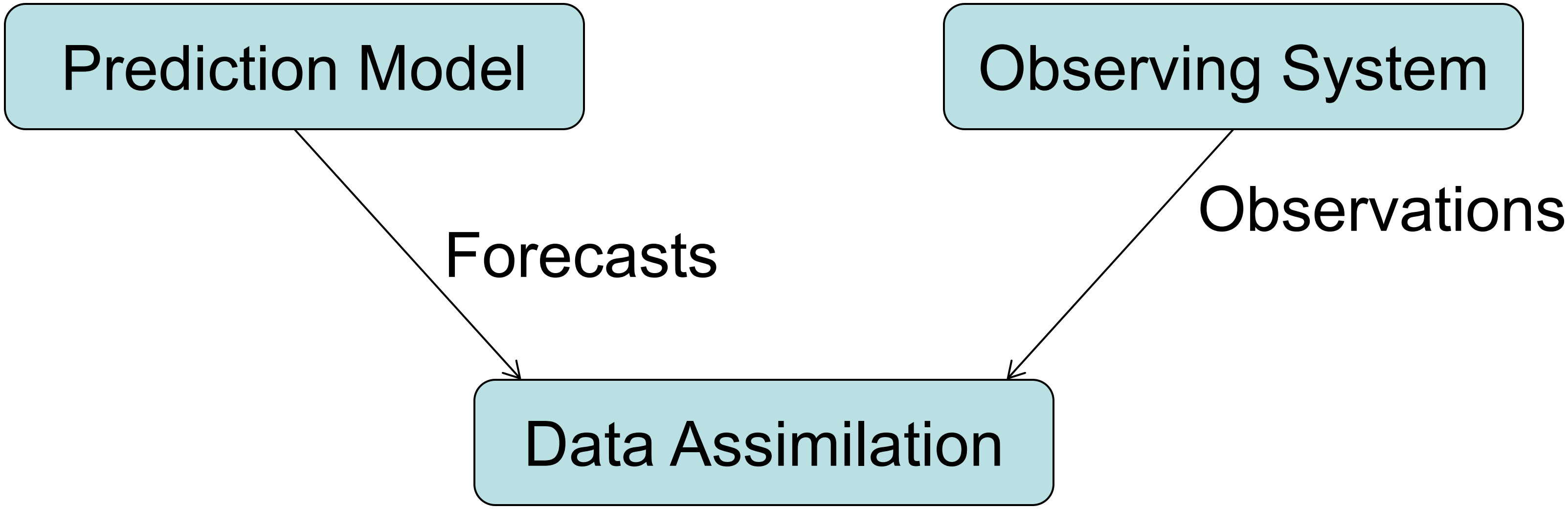
Observation error is normal (gaussian).
Probability (likelihood) of sample states shown.

$$P = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-d^2}{2\sigma^2}}$$ (from definition of normal).

Solid circle is 1 standard deviation, σ.
Dashed circle is 2 standard deviations, 2σ.
(x and y errors uncorrelated)



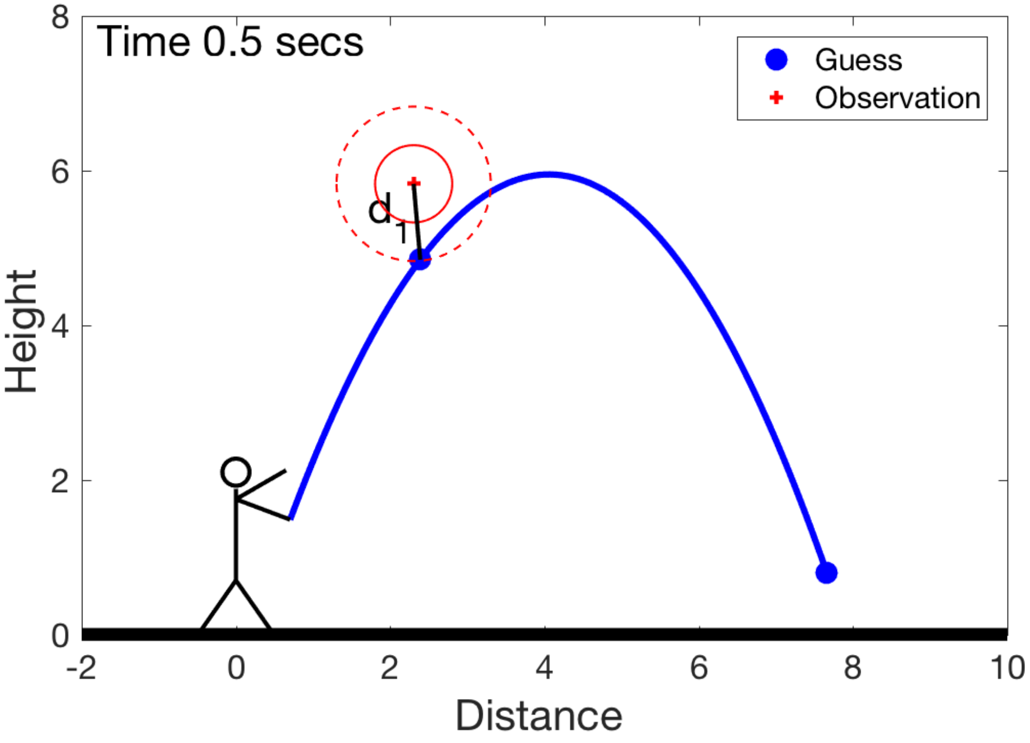
Data Assimilation: Building a simple forecast system



Data Assimilation: Building a simple forecast system

For one obesrvation, likelihood of trajectory is:

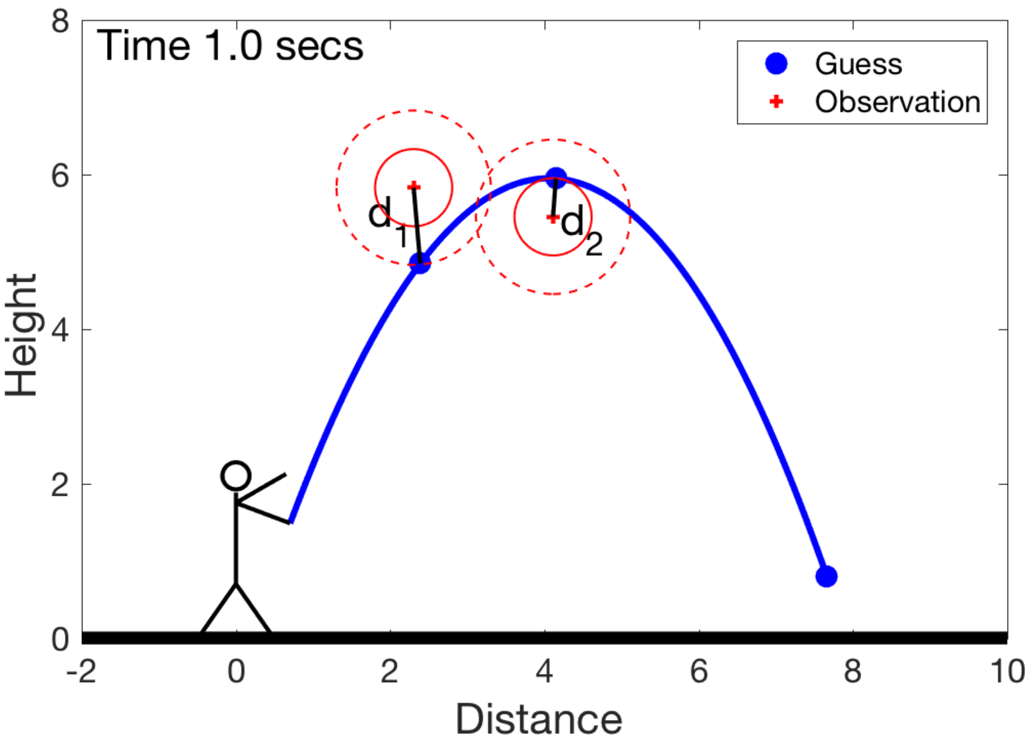
$$P = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{\frac{-d_1^2}{2\sigma_1^2}}$$



Data Assimilation: Building a simple forecast system

For many observations, product of likelihoods

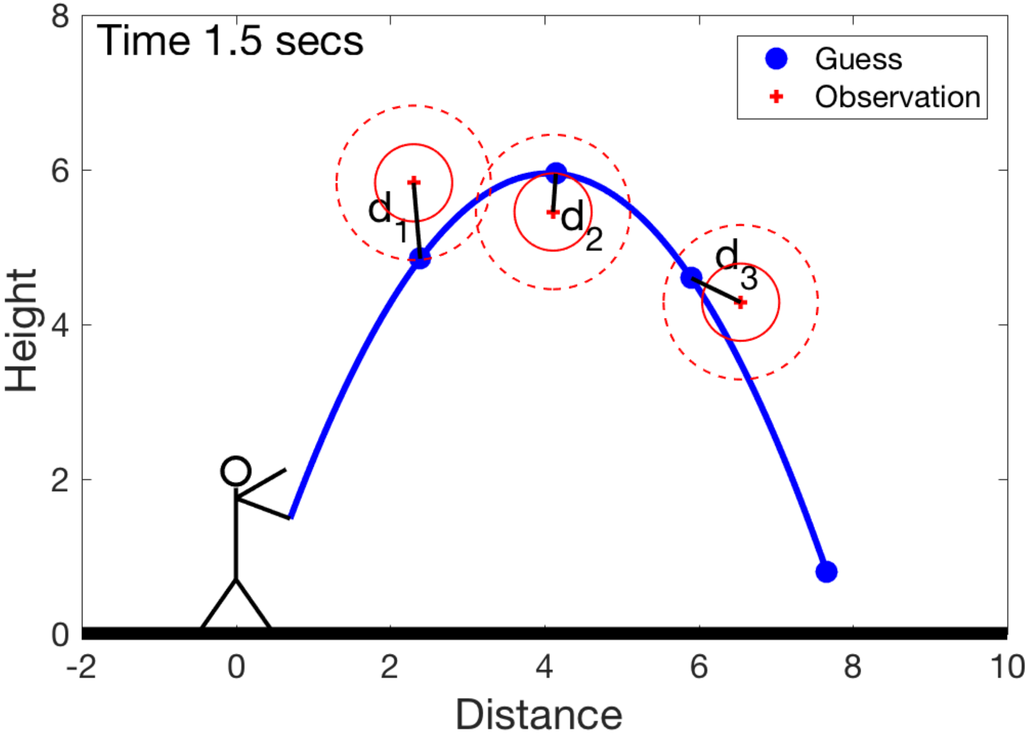
$$P = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{\frac{-d_1^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi{\sigma_2}^2}} e^{\frac{-d_2^2}{2\sigma_2^2}}$$



Data Assimilation: Building a simple forecast system

For many observations, product of likelihoods

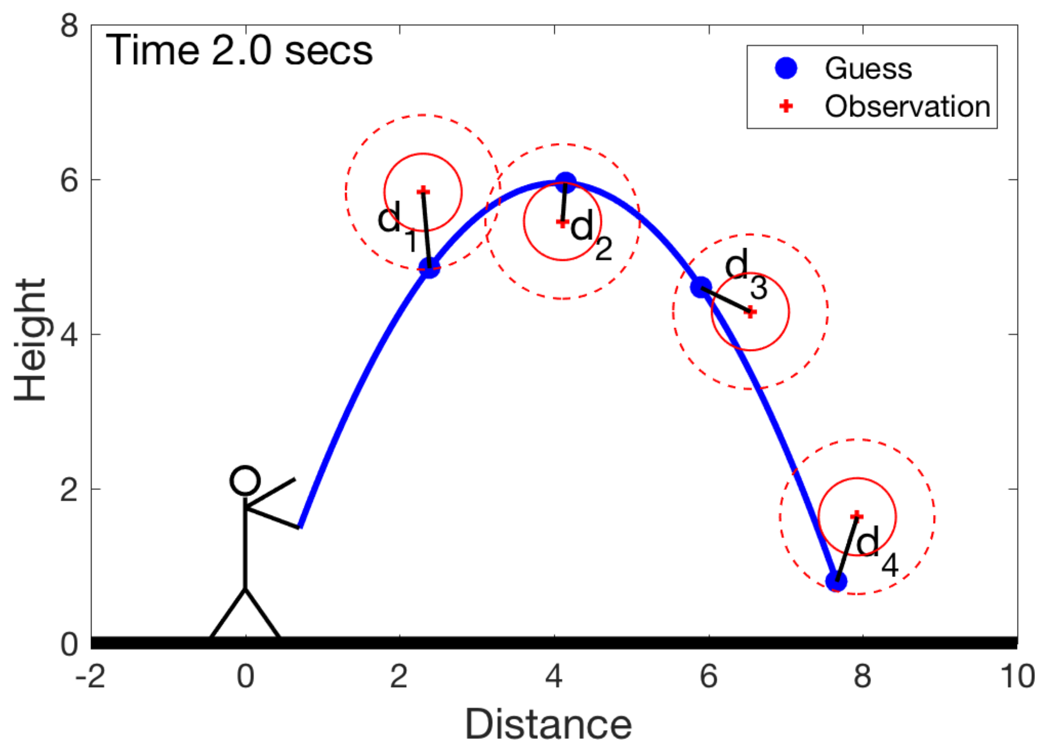
$$P = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{\frac{-d_1^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi{\sigma_2}^2}} e^{\frac{-d_2^2}{2\sigma_2^2}} \frac{1}{\sqrt{2\pi{\sigma_3}^2}} e^{\frac{-d_3^2}{2\sigma_3^2}}$$



Data Assimilation: Building a simple forecast system

For many observations, product of likelihoods

$$P = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{\frac{-d_1^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi{\sigma_2}^2}} e^{\frac{-d_2^2}{2\sigma_2^2}} \frac{1}{\sqrt{2\pi{\sigma_3}^2}} e^{\frac{-d_3^2}{2\sigma_3^2}} \frac{1}{\sqrt{2\pi{\sigma_4}^2}} e^{\frac{-d_4^2}{2\sigma_4^2}}$$



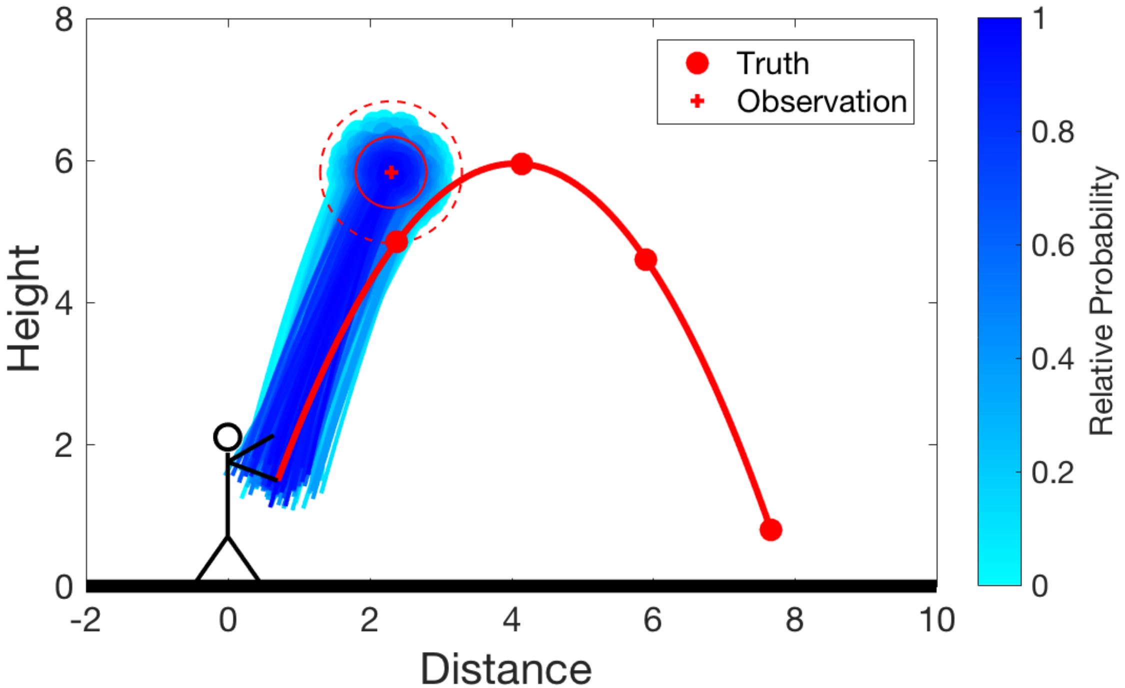
Data Assimilation: Building a simple forecast system

Assimilating the first observation:

Make large ensemble of forecasts.

Use first observation to assign relative probability.

500 most likely of 10000 (darker blue => more likely).

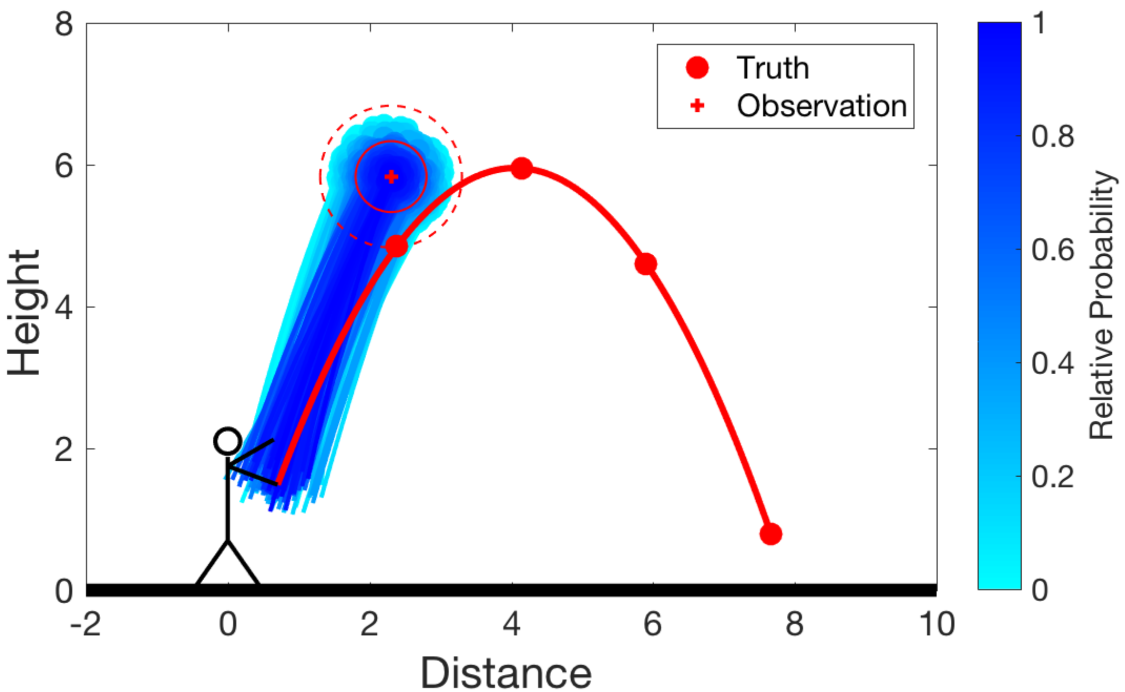


Data Assimilation: Building a simple forecast system

Assimilating the first observation:

500 balls at time 0.5 are an ensemble analysis.

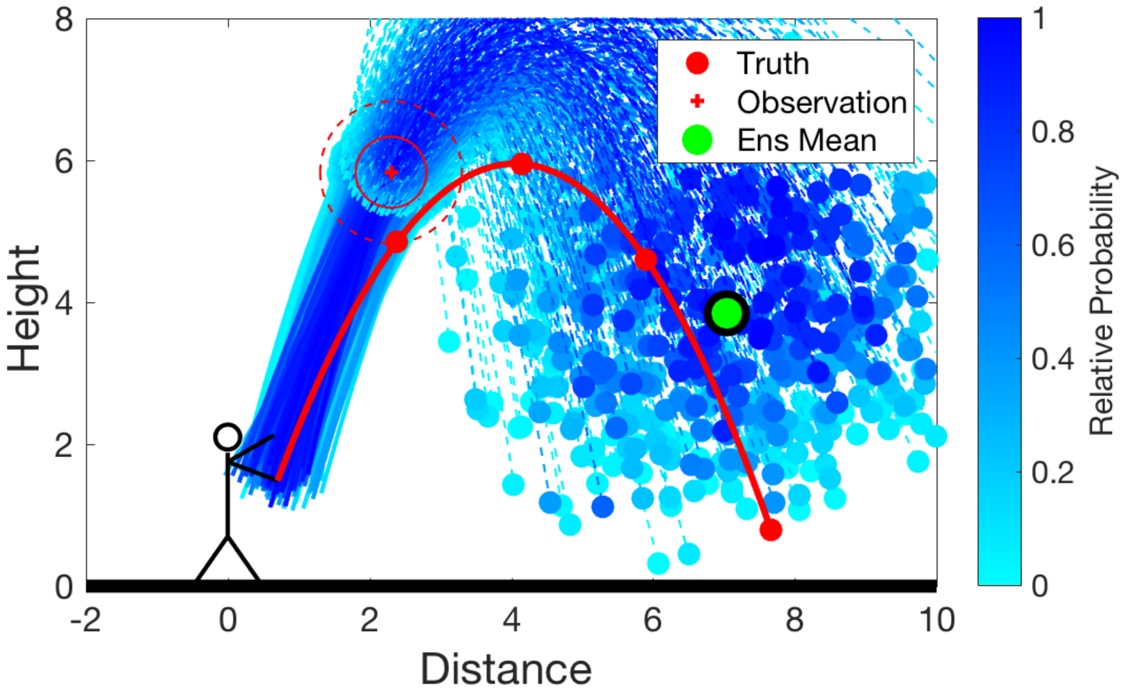
Show uncertainty of best estimate of red ball's location.



Data Assimilation: Building a simple forecast system

Assimilating the first observation:

- Can make a forecast to later times.
- Green is probability weighted ensemble mean.
- Best estimate of the ball's position after 2 seconds.

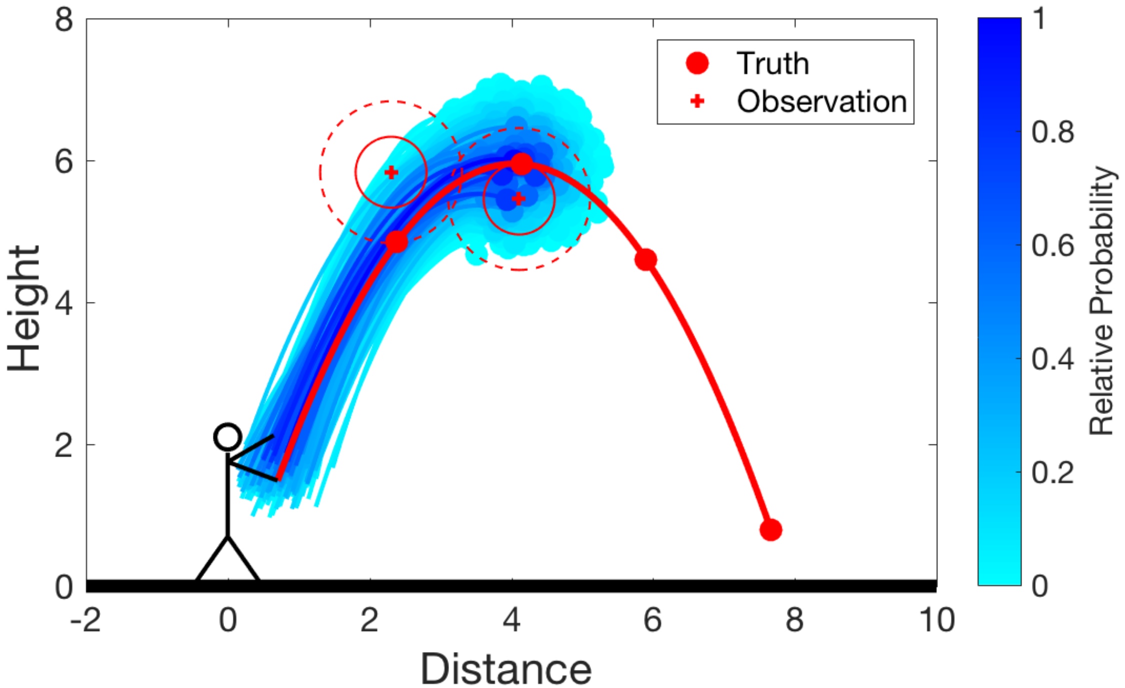


Data Assimilation: Building a simple forecast system

Assimilating the second observation:

Can include observations from subsequent times.

Analysis using observations at 0.5 and 1 seconds shown.

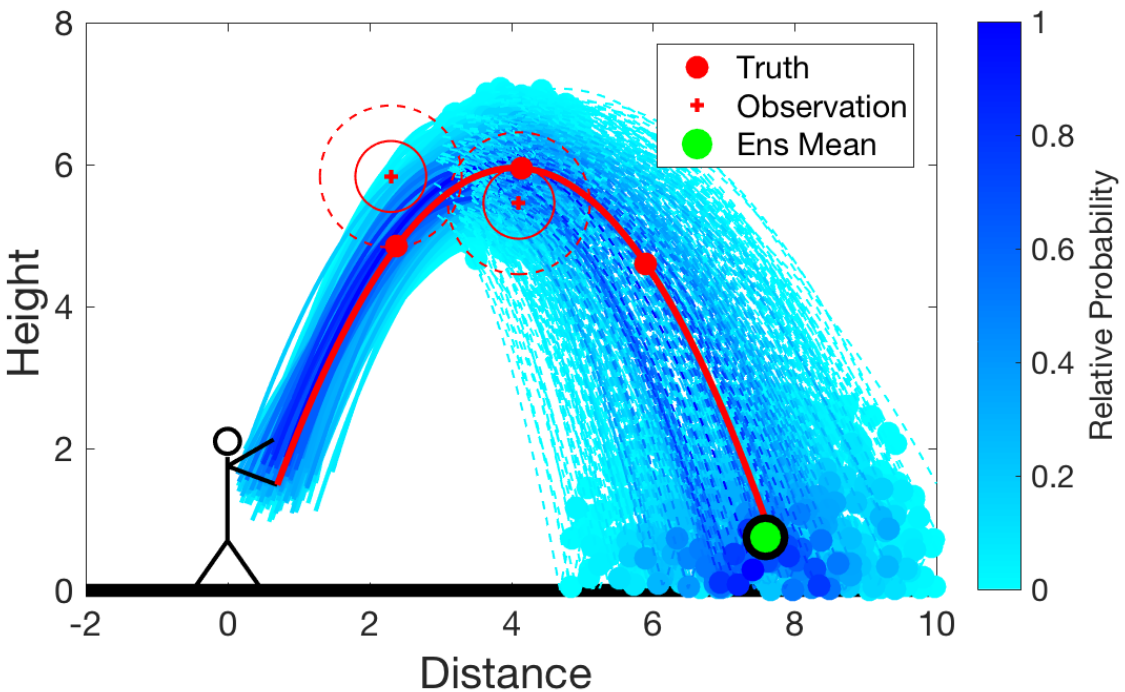


Data Assimilation: Building a simple forecast system

Assimilating the second observation:

Can include observations from subsequent times.

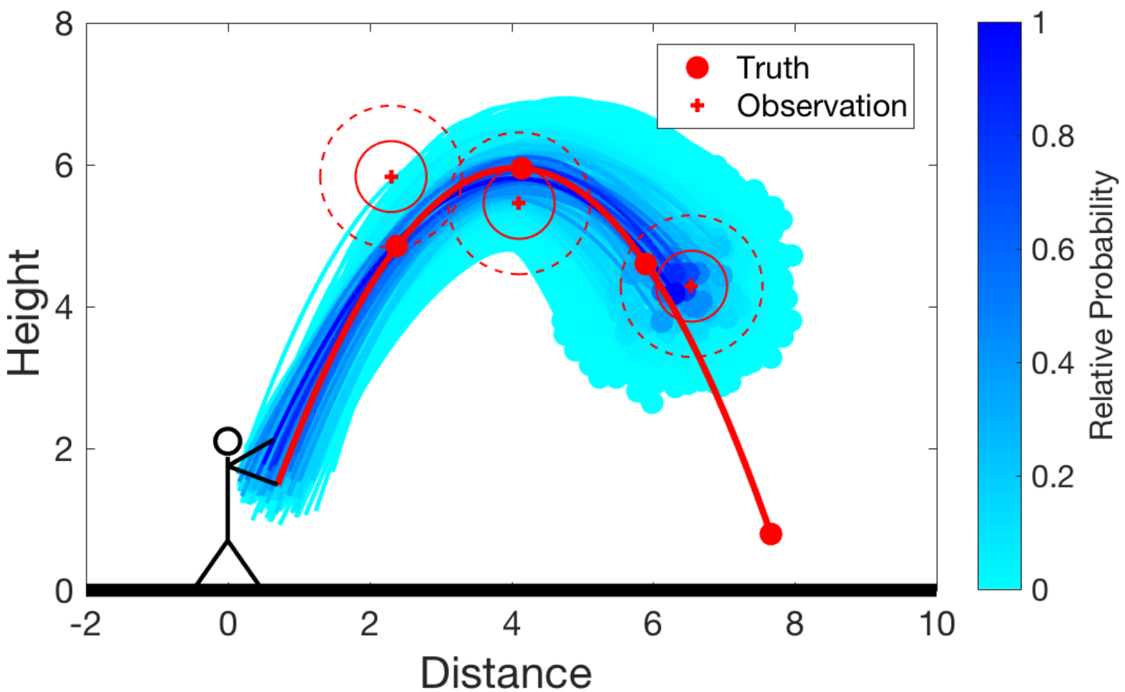
Forecast using observations at 0.5 and 1 seconds shown.



Data Assimilation: Building a simple forecast system

Assimilating the third observation:

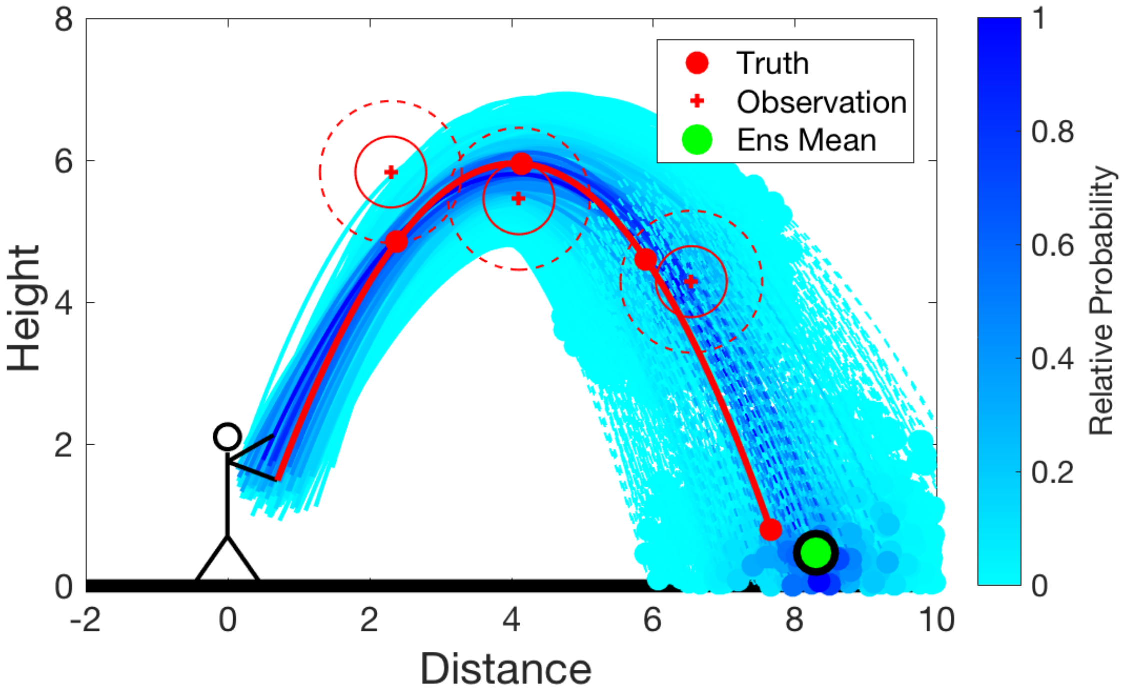
Analysis using observations at 0.5, 1, 1.5 seconds shown.



Data Assimilation: Building a simple forecast system

Assimilating the third observation:

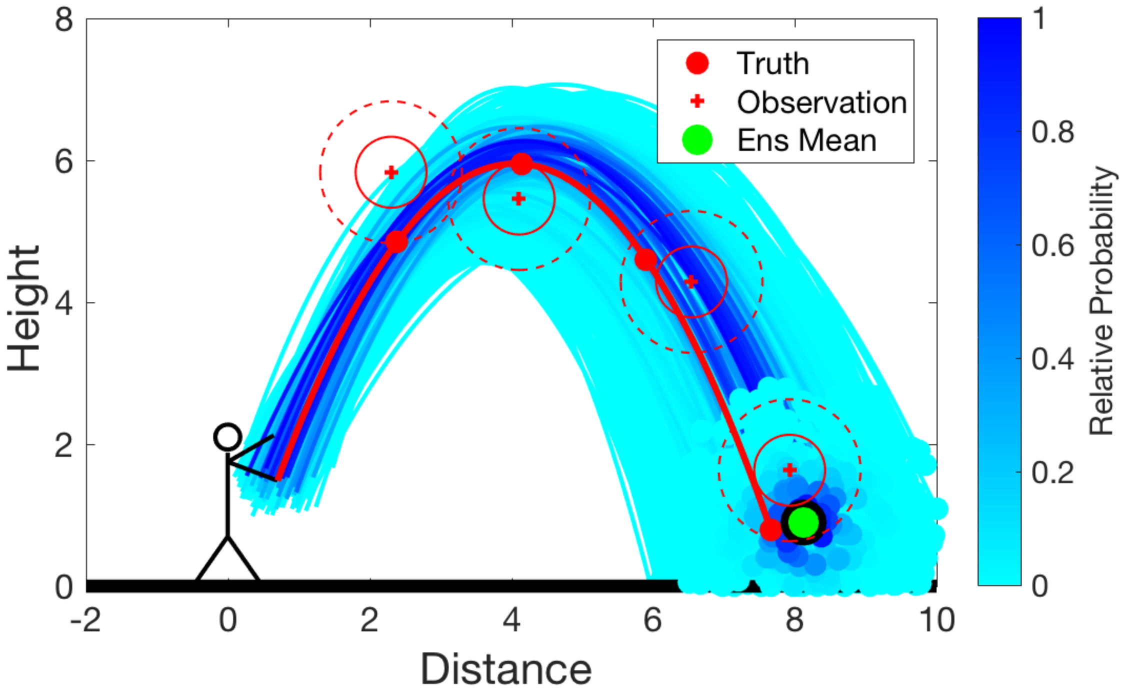
Forecast using observations at 0.5, 1, 1.5 seconds shown.



Data Assimilation: Building a simple forecast system

Assimilating the final observation:

Analysis using observations at 0.5, 1, 1.5, 2 seconds.



Data Assimilation: Building a simple forecast system

Data assimilation combines model and observations.

Analyses and forecasts are always uncertain.

Analyses become more accurate, more certain with more observations.

Shorter lead forecasts are more accurate, more certain.

Quantifying this uncertainty is important.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

Data Assimilation: What can it do?

Analyses more accurate, more certain with more observations.

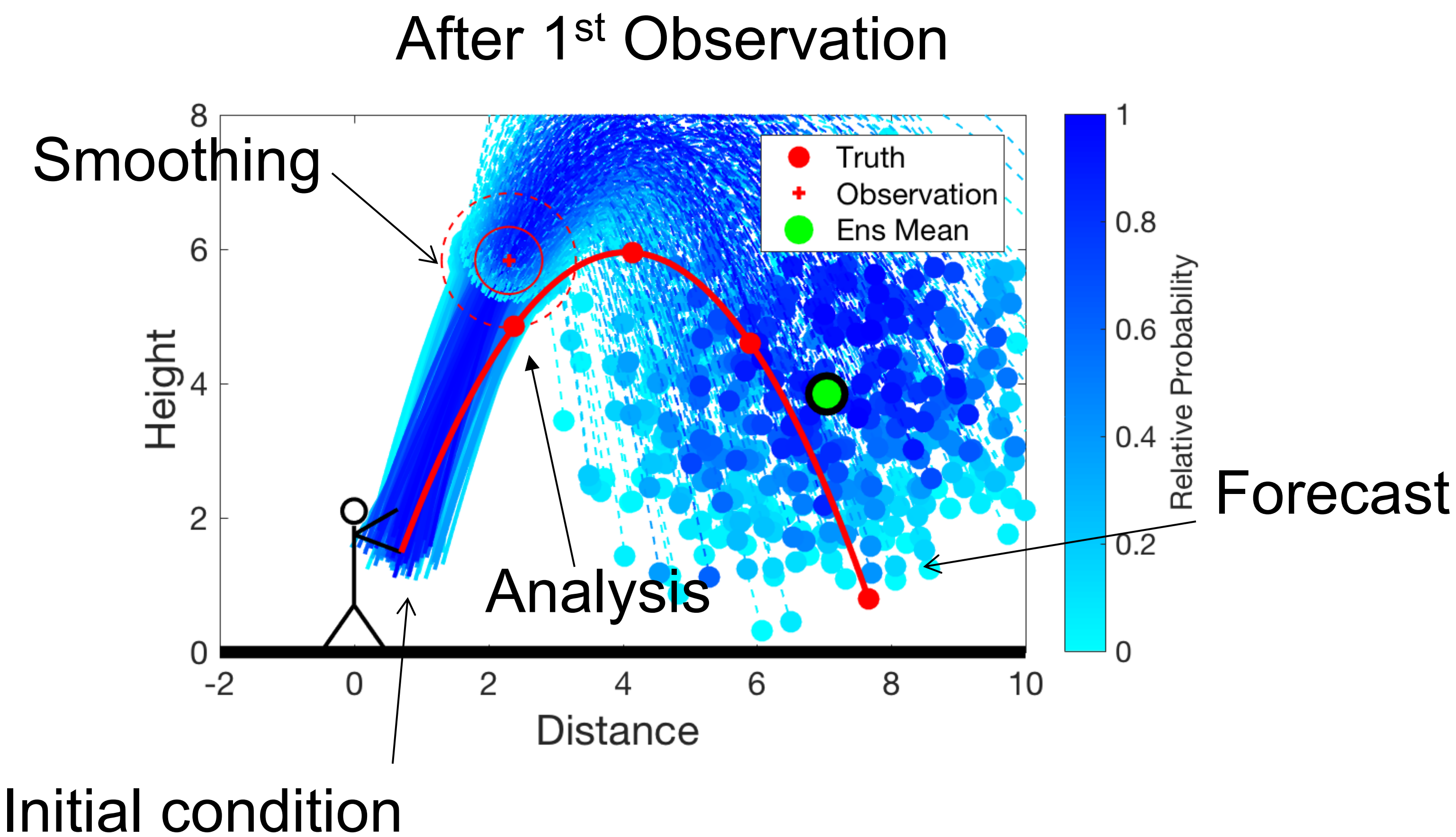
Shorter lead forecasts are more accurate, more certain.

Can also improve estimate using future observations; this is called smoothing.

Information about the thrower can also be obtained; this is called initial condition estimation.

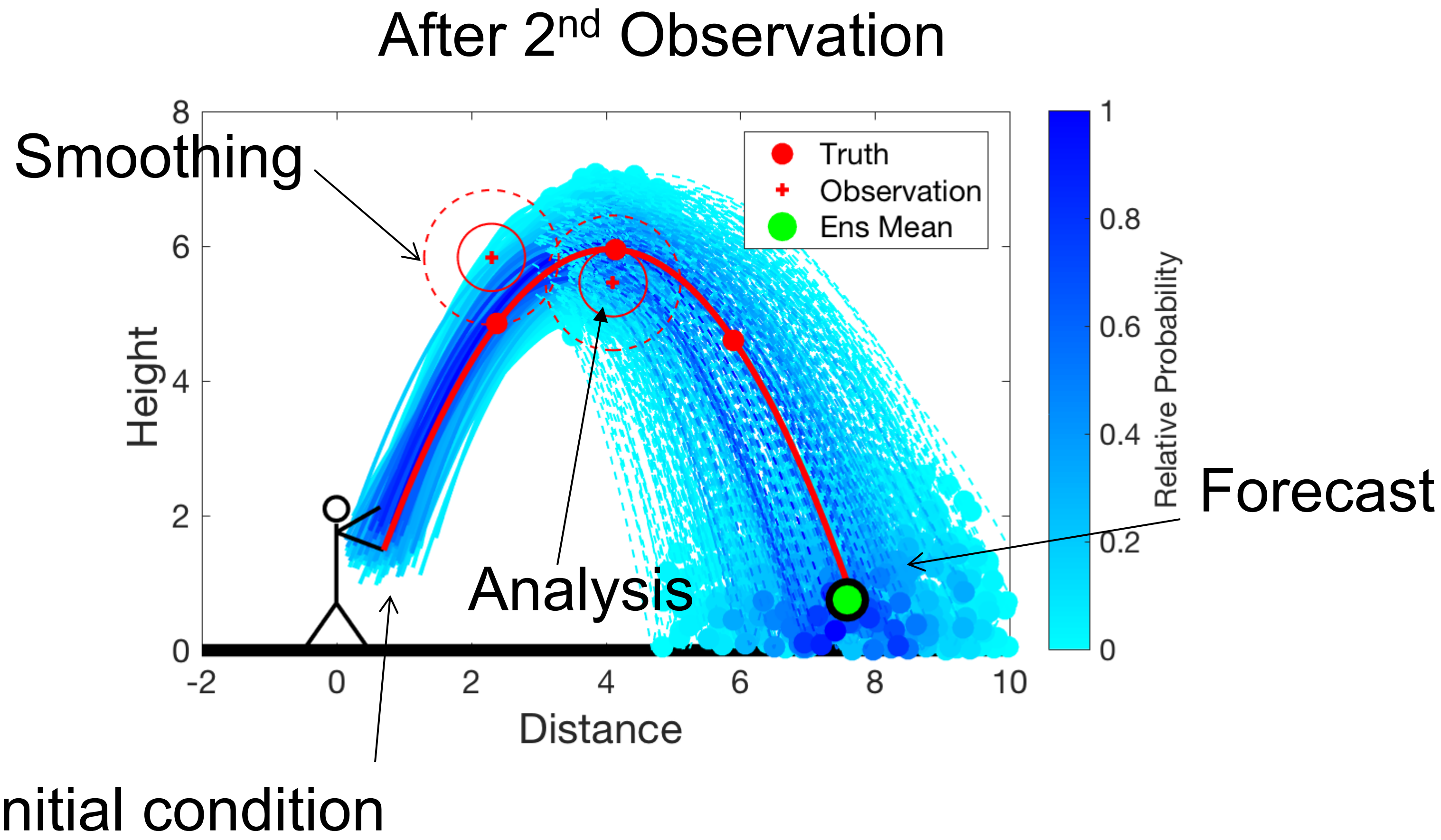
Data Assimilation: What can it do?

As more observations are used, forecast for time two seconds generally improves and has less uncertainty.



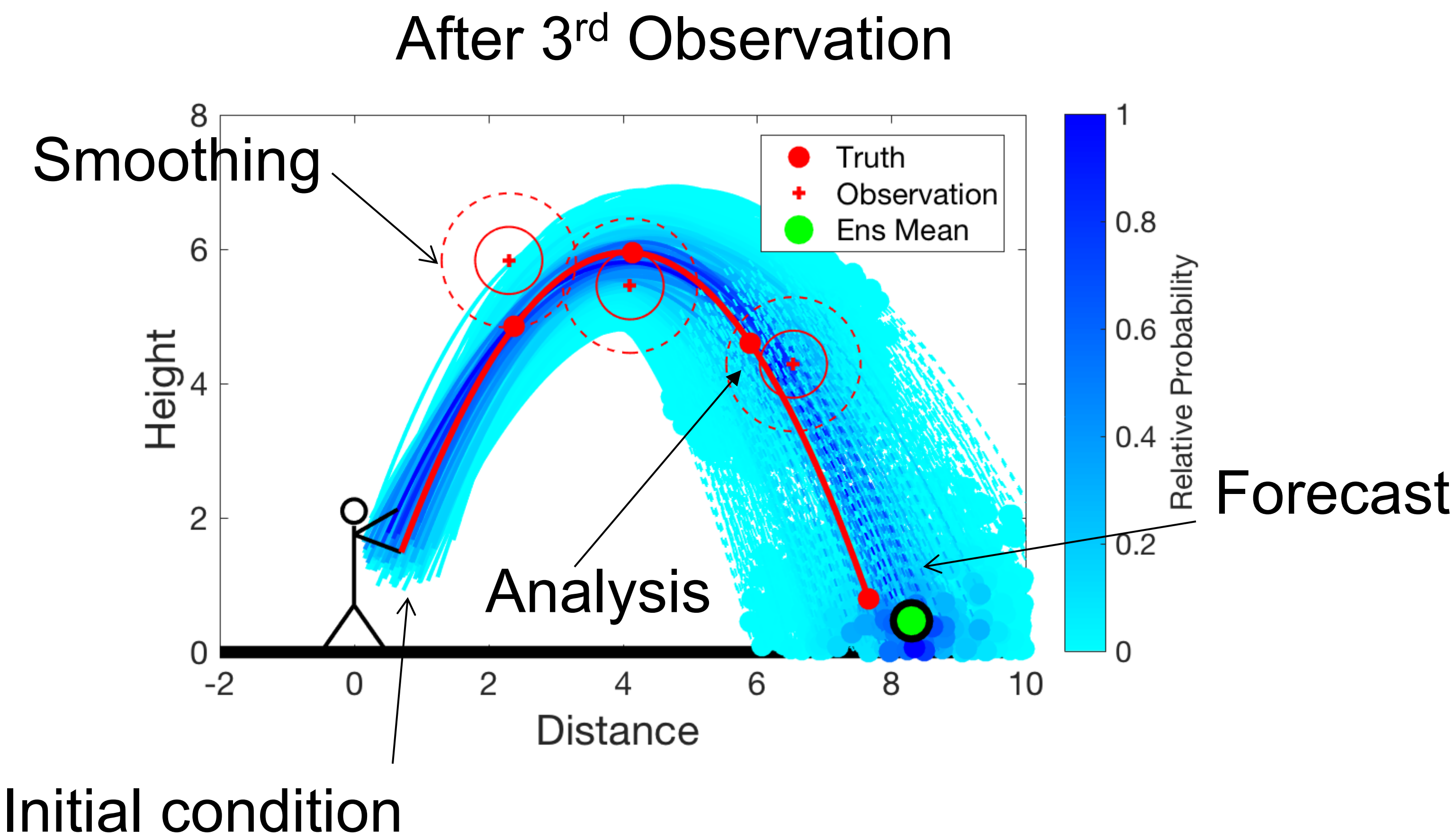
Data Assimilation: What can it do?

As more observations are used, forecast for time two seconds generally improves and has less uncertainty.



Data Assimilation: What can it do?

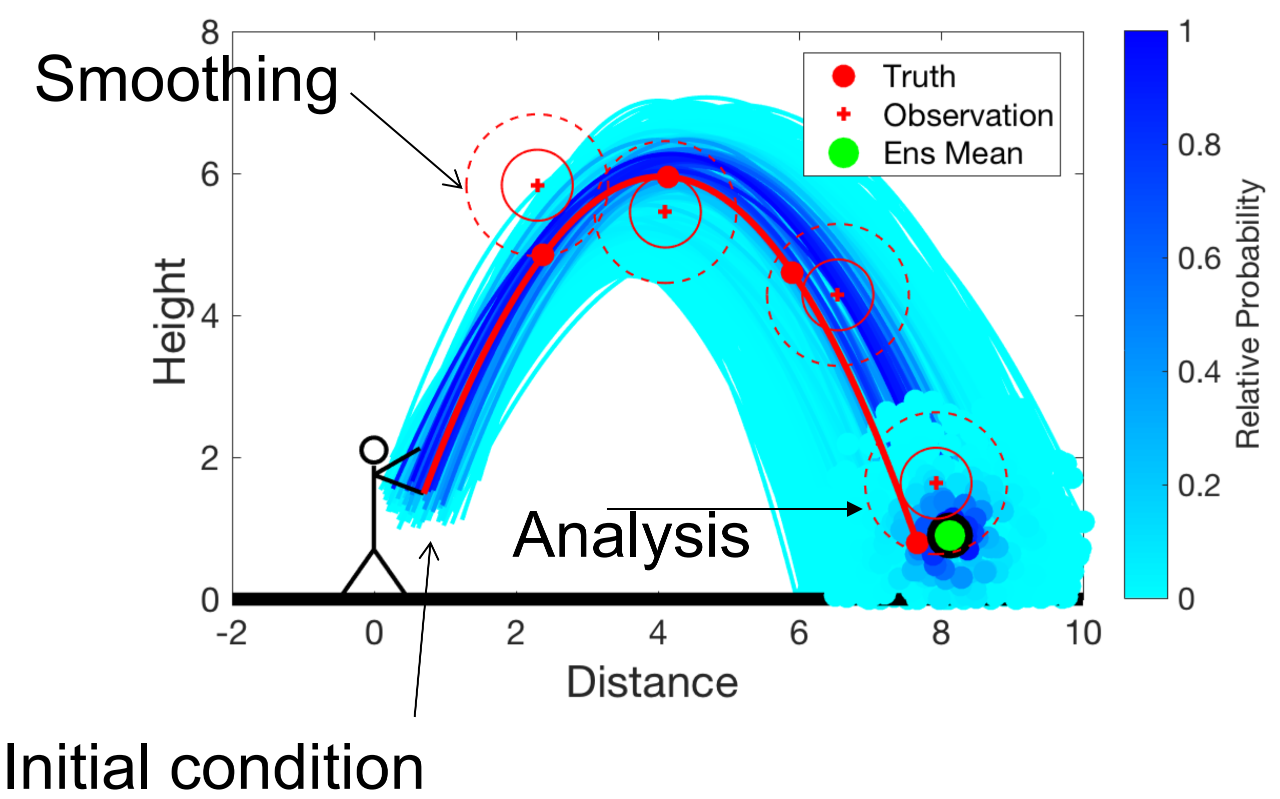
As more observations are used, forecast for time two seconds generally improves and has less uncertainty.



Data Assimilation: What can it do?

As more observations are used, forecast for time two seconds generally improves and has less uncertainty.

After 4th Observation



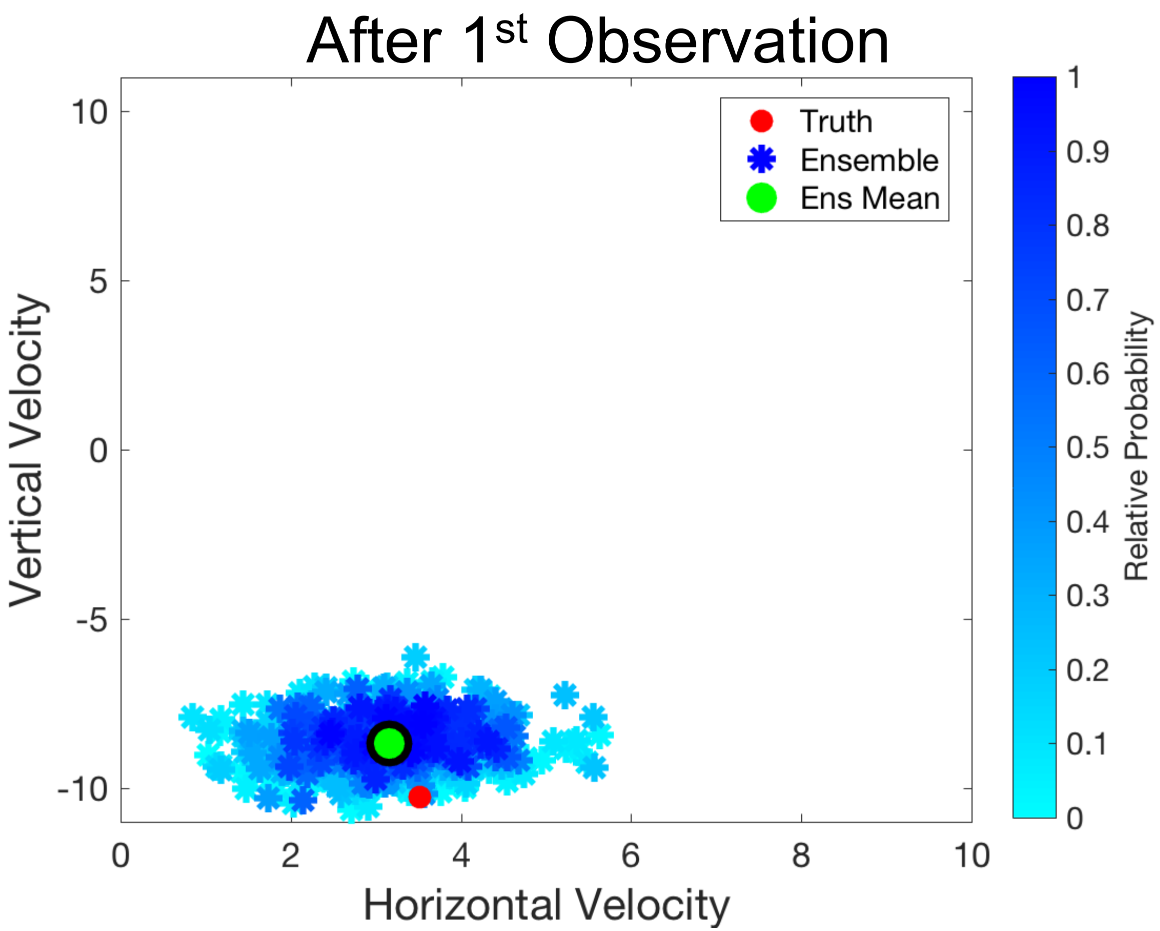
Data Assimilation: What can it do?

Impact on unobserved (hidden) variables.

Have been looking at position x, y.
Velocity components u and v are also part of model.
Estimates of u and v are improved by observations of x, y.

Data Assimilation: What can it do?

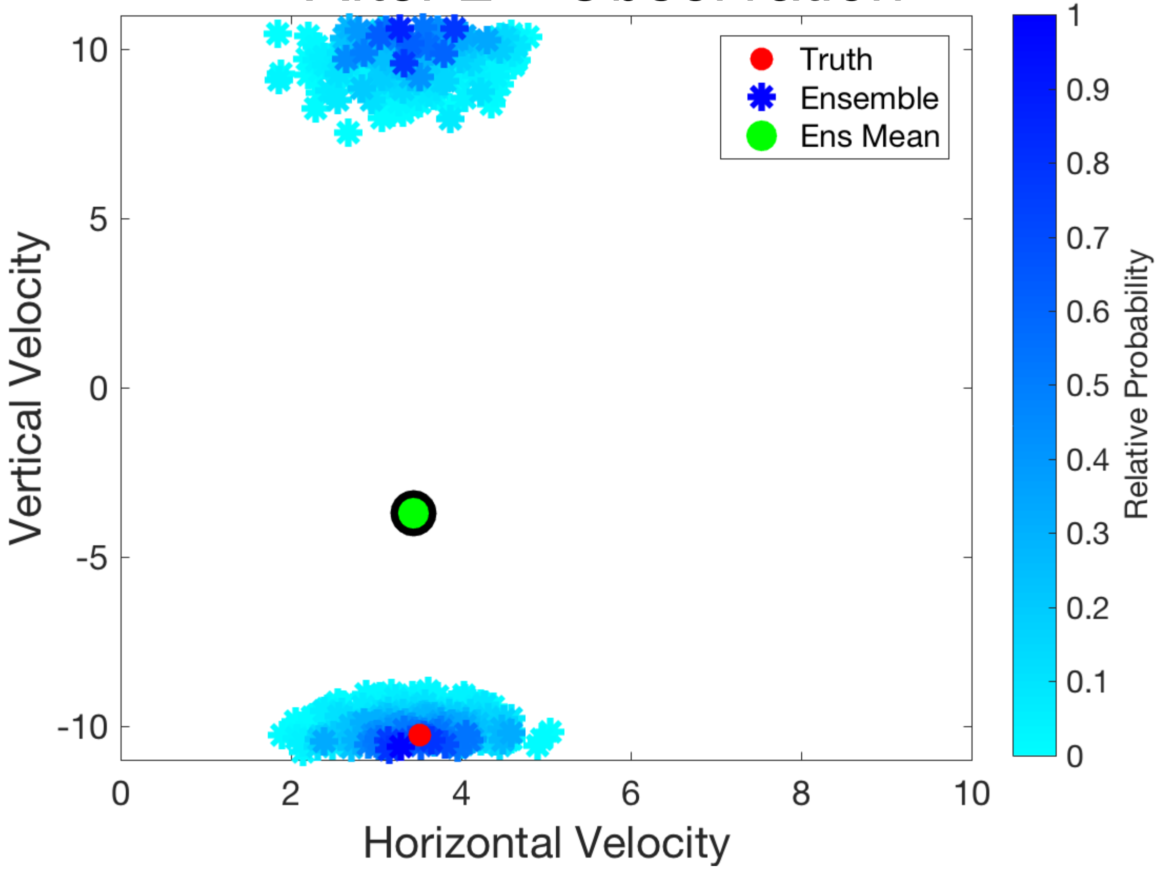
Forecast of velocity as lead time decreases.
As more observations are used, forecast for time 2 seconds generally improves, gets more certain.



Data Assimilation: What can it do?

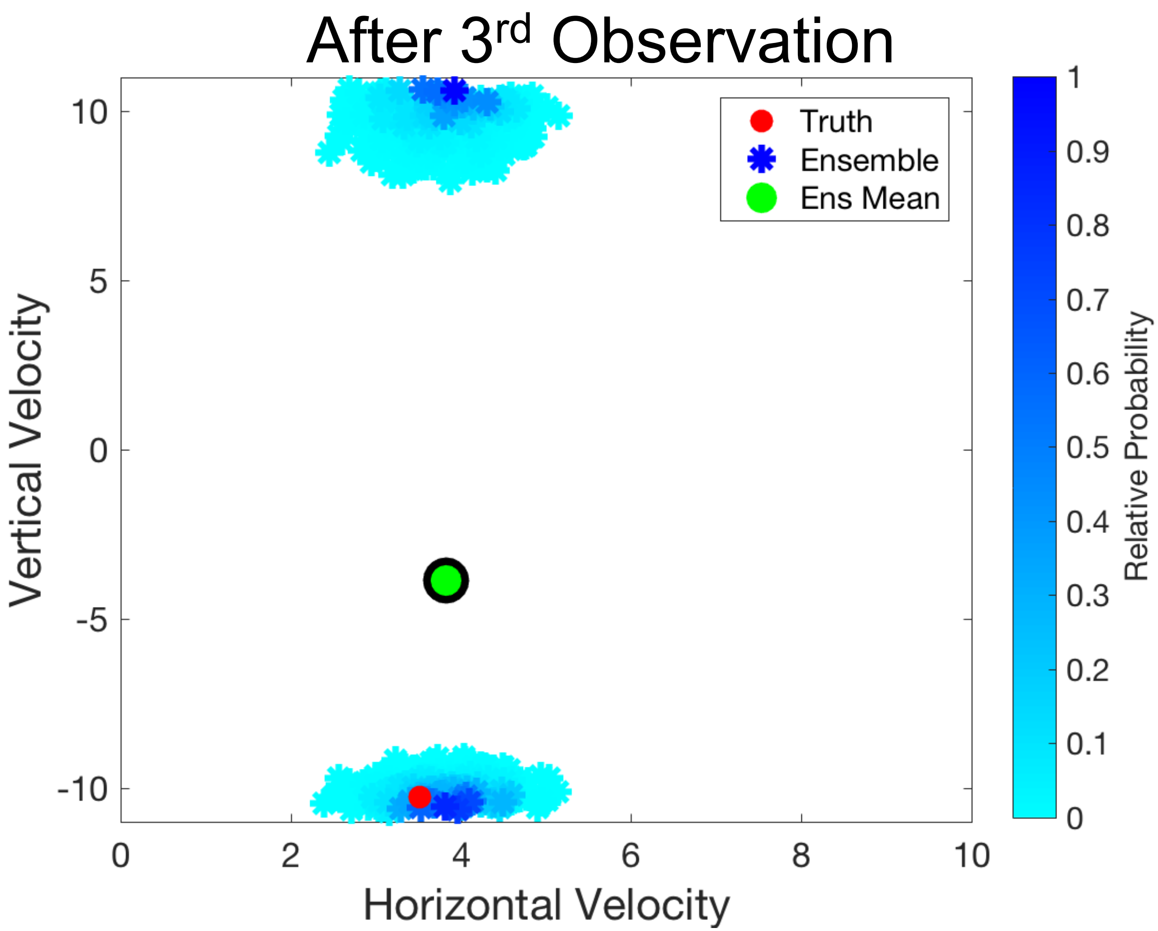
Forecast of velocity as lead time decreases.
As more observations are used, forecast for time 2 seconds generally improves, gets more certain.

After 2nd Observation



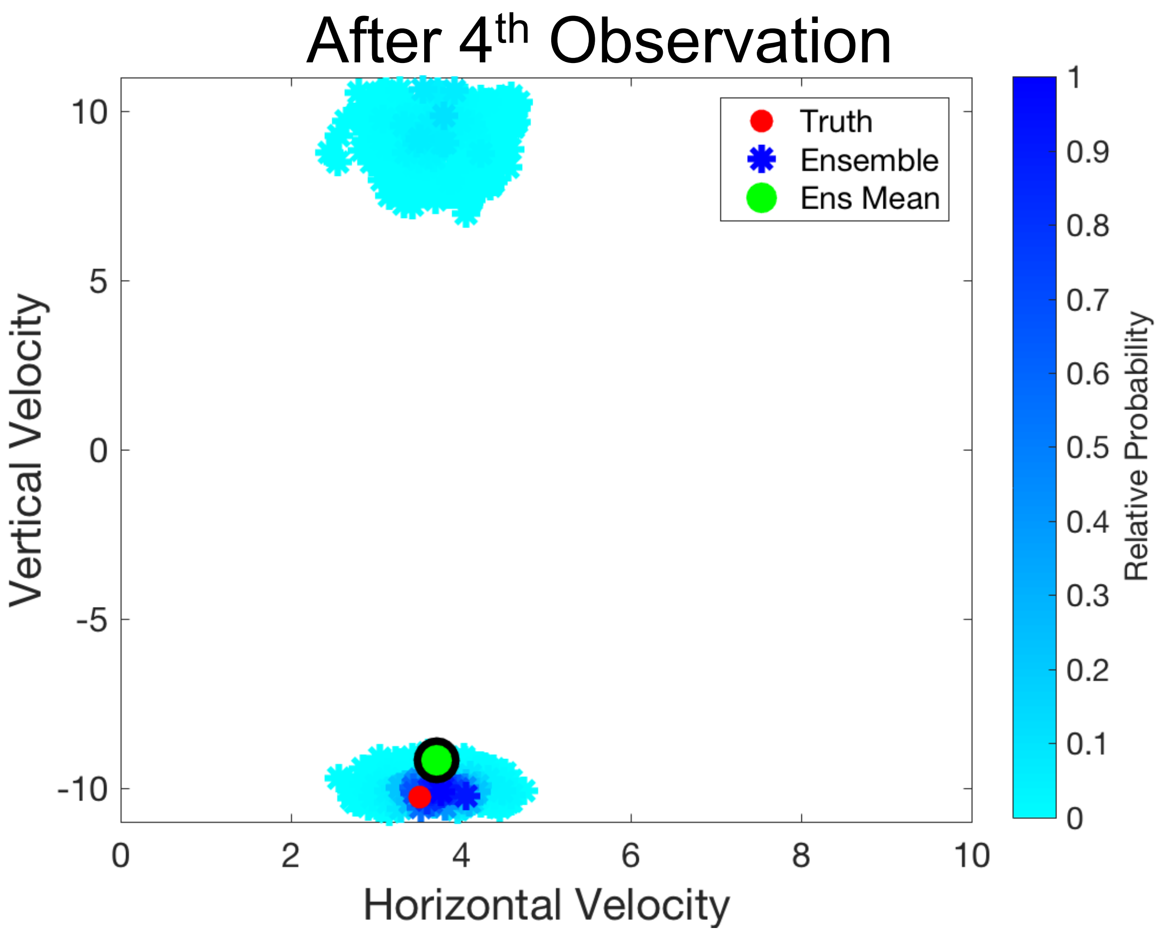
Data Assimilation: What can it do?

Forecast of velocity as lead time decreases.
As more observations are used, forecast for time 2 seconds generally improves, gets more certain.



Data Assimilation: What can it do?

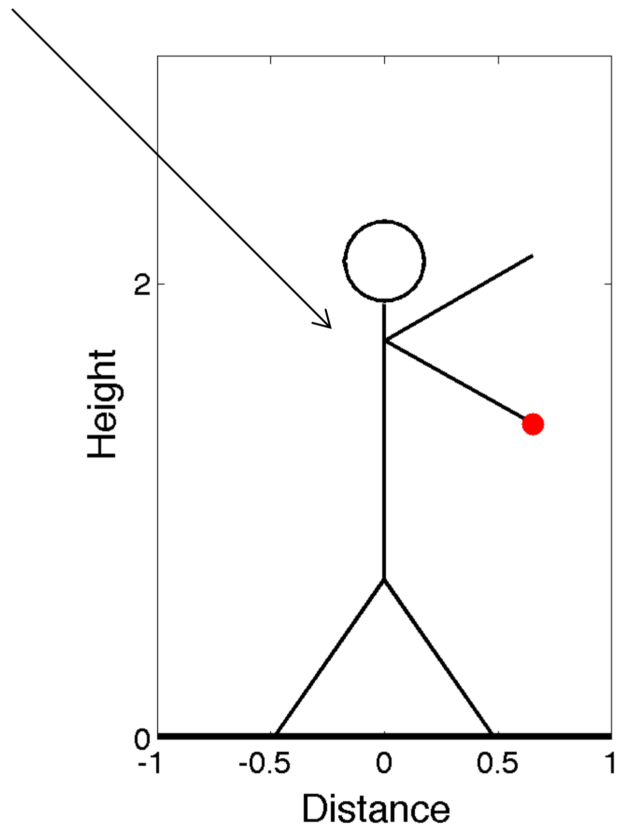
Forecast of velocity as lead time decreases.
As more observations are used, forecast for time 2 seconds generally improves, gets more certain.



Data Assimilation: What can it do?

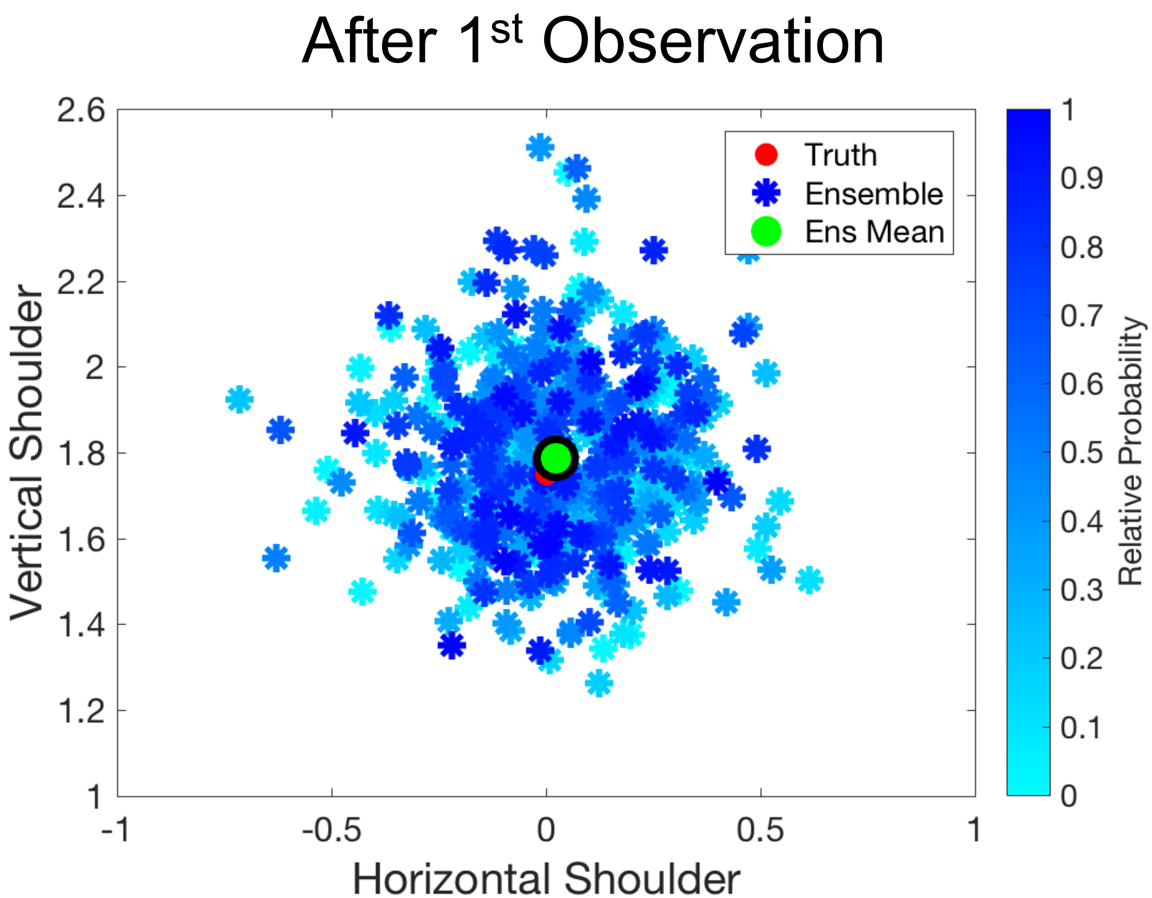
Estimating initial conditions.

Example: estimate of shoulder position.



Data Assimilation: What can it do?

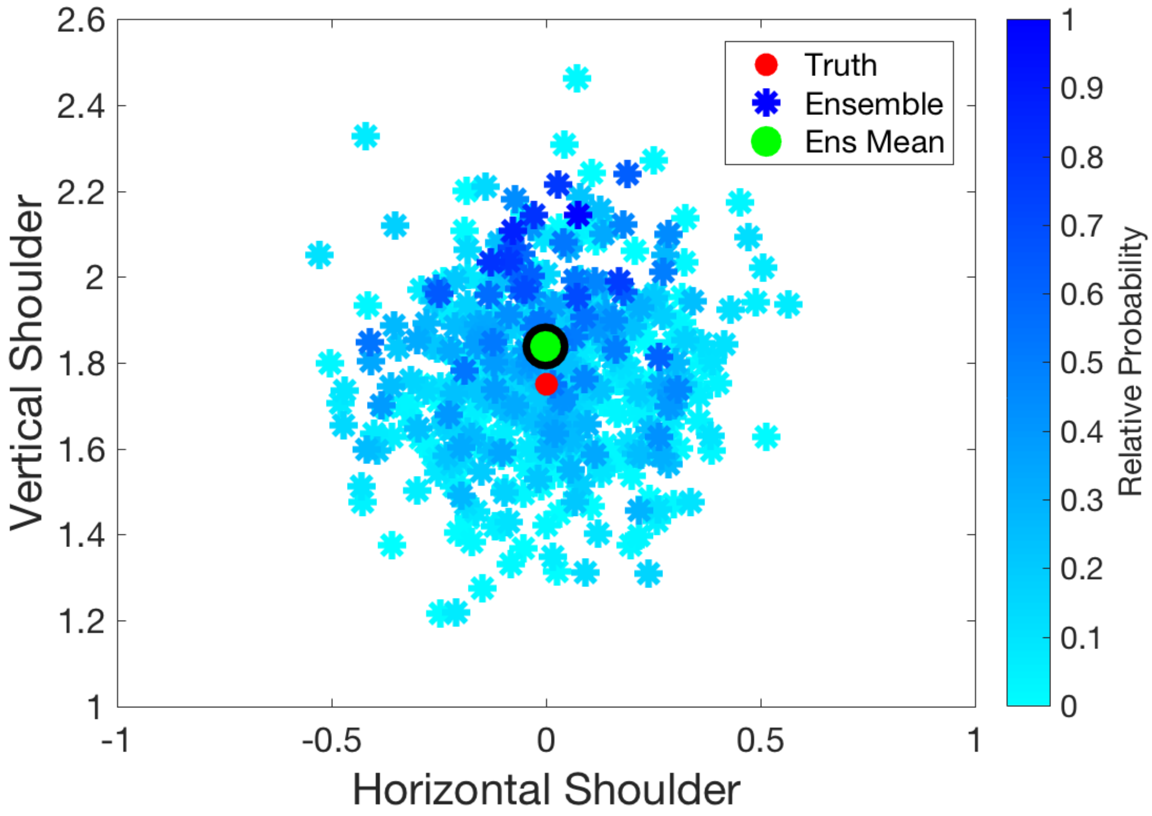
Estimating initial conditions: Shoulder position.
Estimate improves, gets more certain with more observations.



Data Assimilation: What can it do?

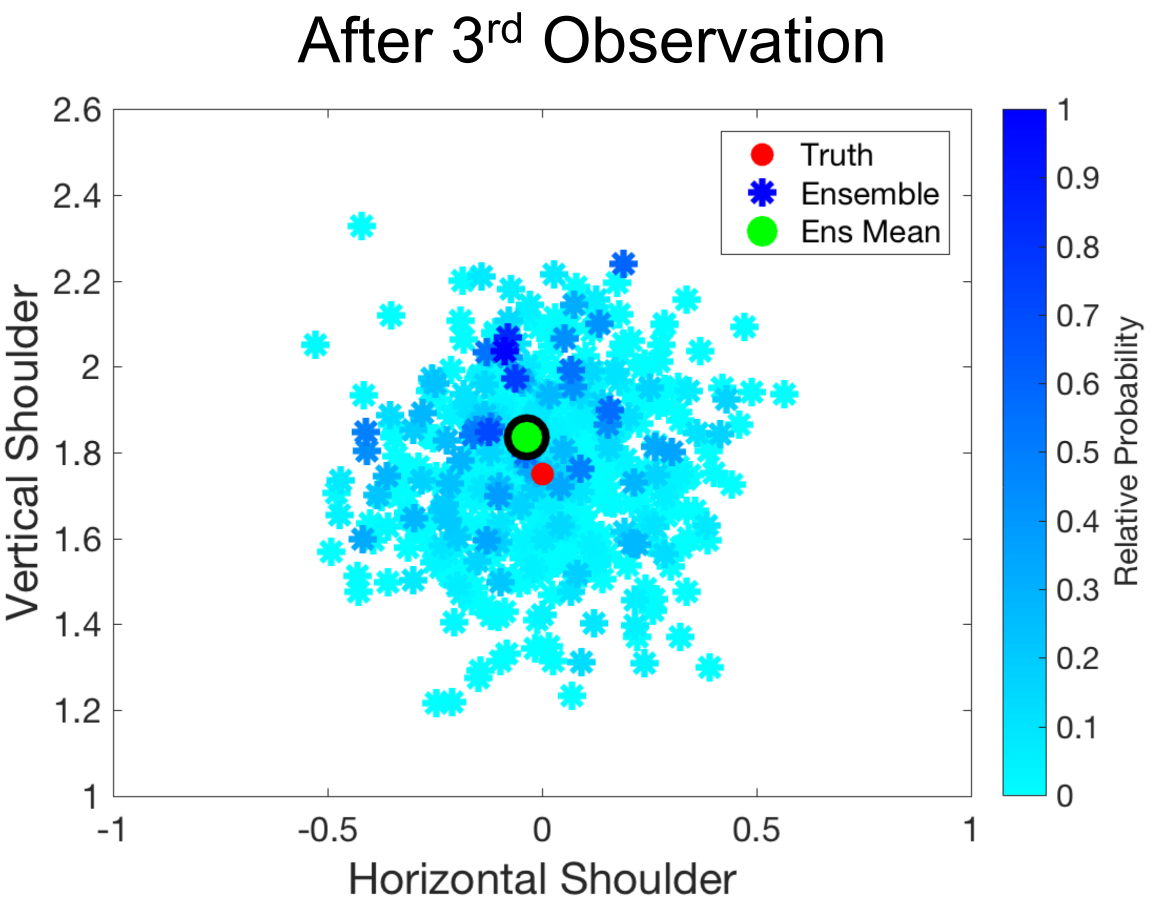
Estimating initial conditions: Shoulder position.
Estimate improves, gets more certain with more observations.

After 2nd Observation



Data Assimilation: What can it do?

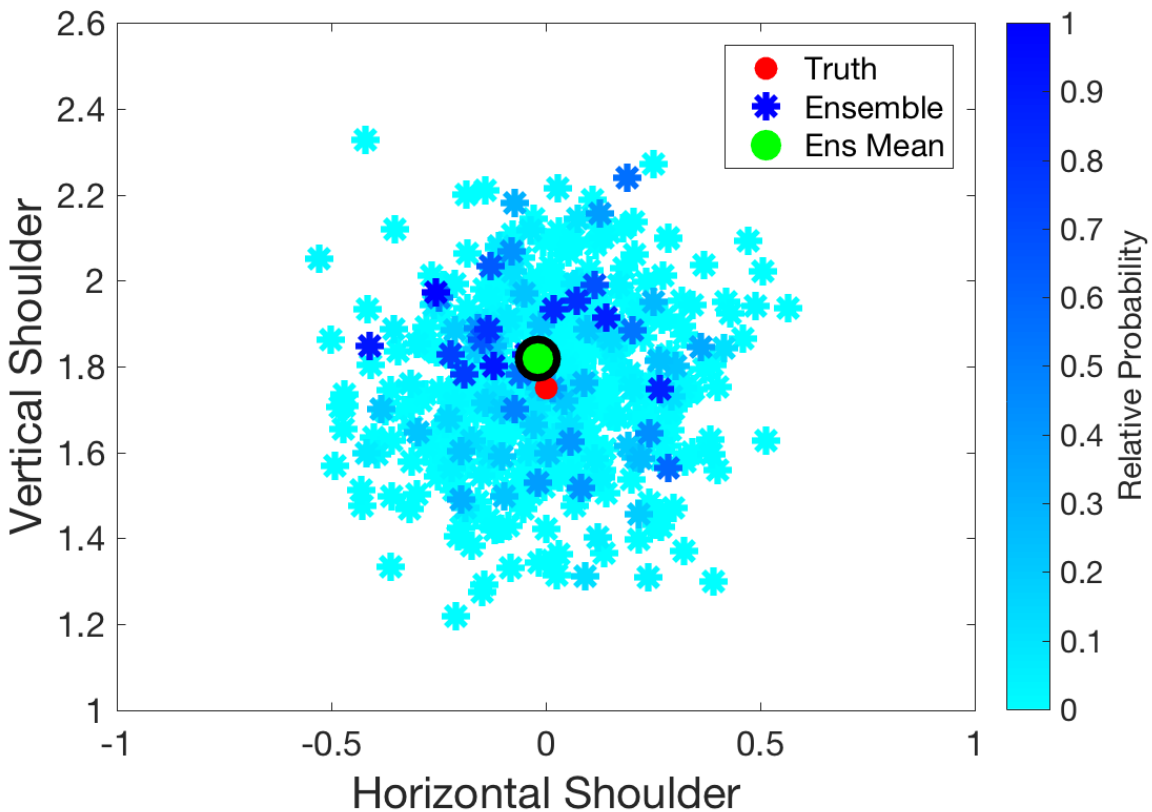
Estimating initial conditions: Shoulder position.
Estimate improves, gets more certain with more observations.



Data Assimilation: What can it do?

Estimating initial conditions: Shoulder position.
Estimate improves, gets more certain with more observations.

After 4th Observation



Data Assimilation: What can it do?

Estimating a model parameter:

Suppose we don't know g exactly.

Put a sample of possible g values into model ensemble.
Each forecast has its own g.

Look at probability of resulting trajectories as function of g.

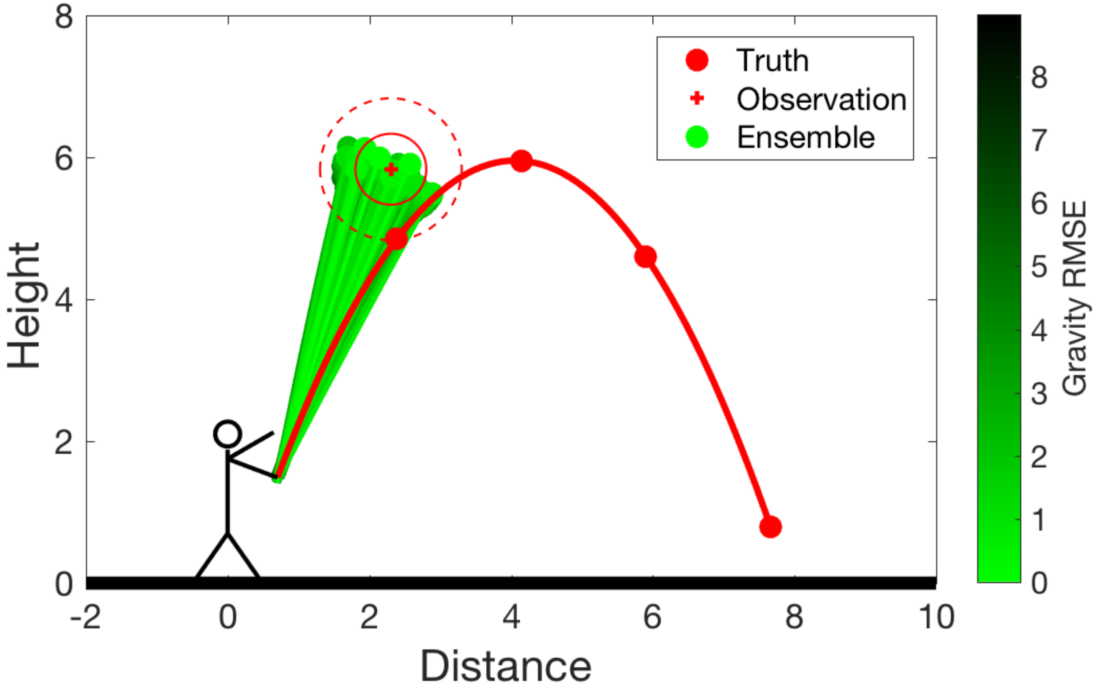
Display 500 highest probability trajectories out of 10,000.

Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.

After 1st Observation

Brighter green is closer to -9.8 m/s^2

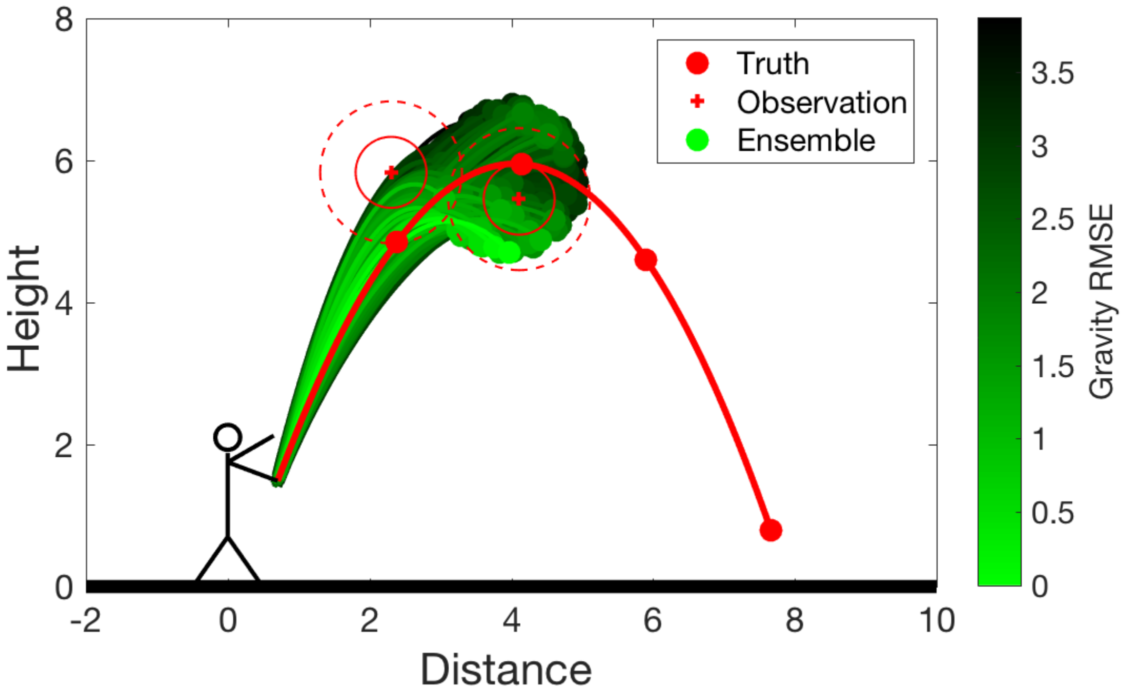


Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.

After 2^{nd} Observation

Brighter green is closer to -9.8 m/s^2

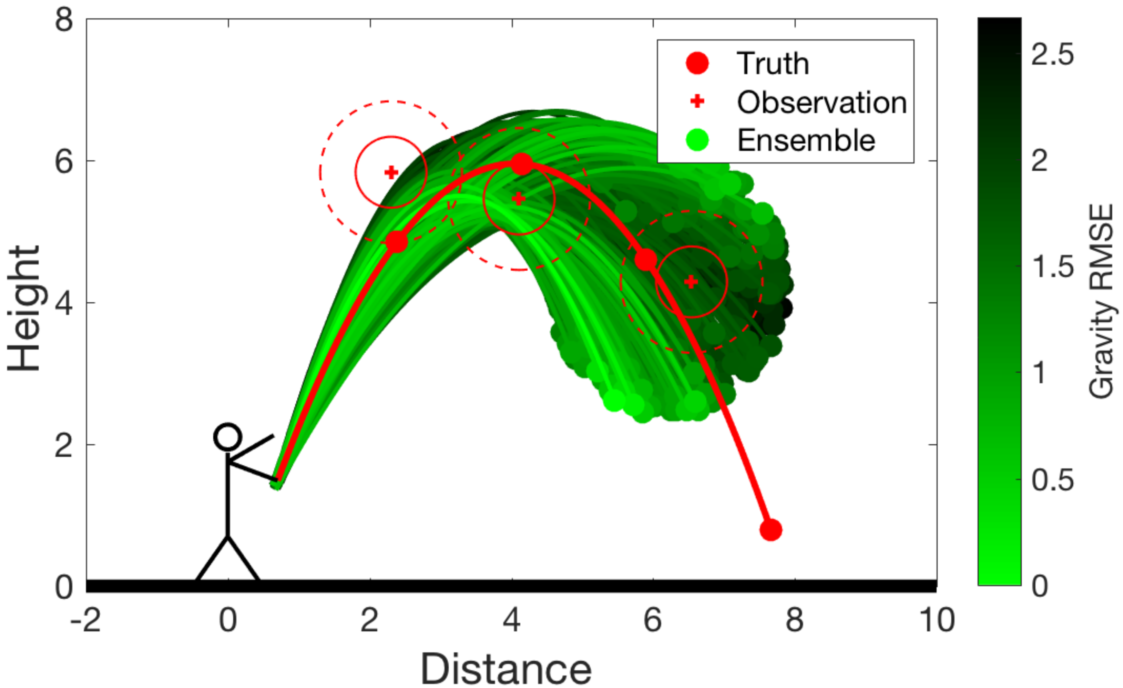


Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.

After 3rd Observation

Brighter green is closer to -9.8 m/s^2

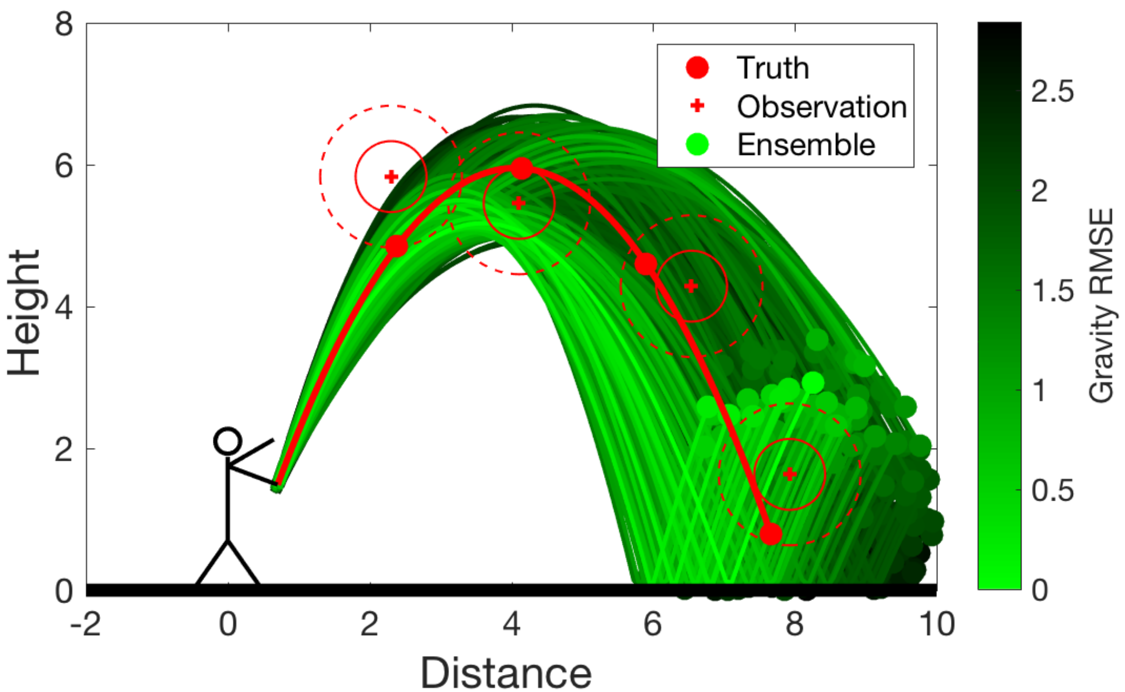


Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.

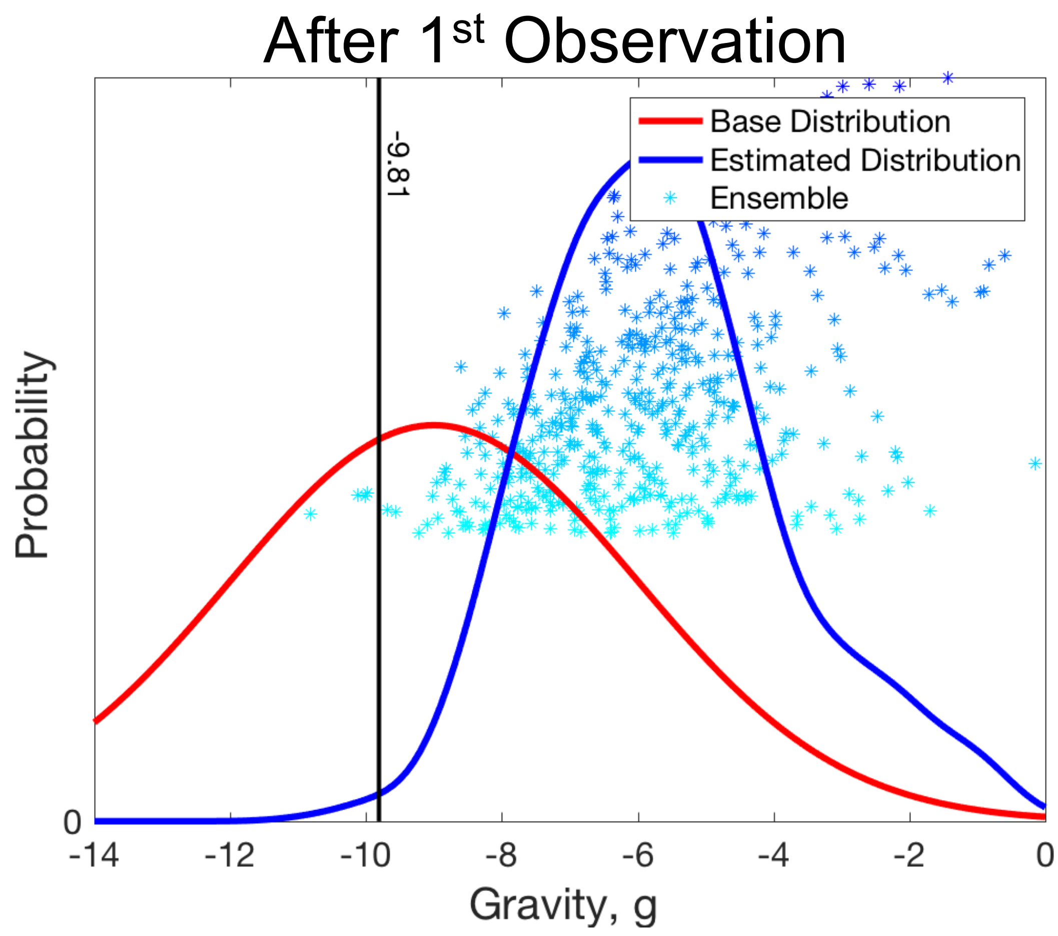
After 4th Observation

Brighter green is closer to - 9.8 m/s^2



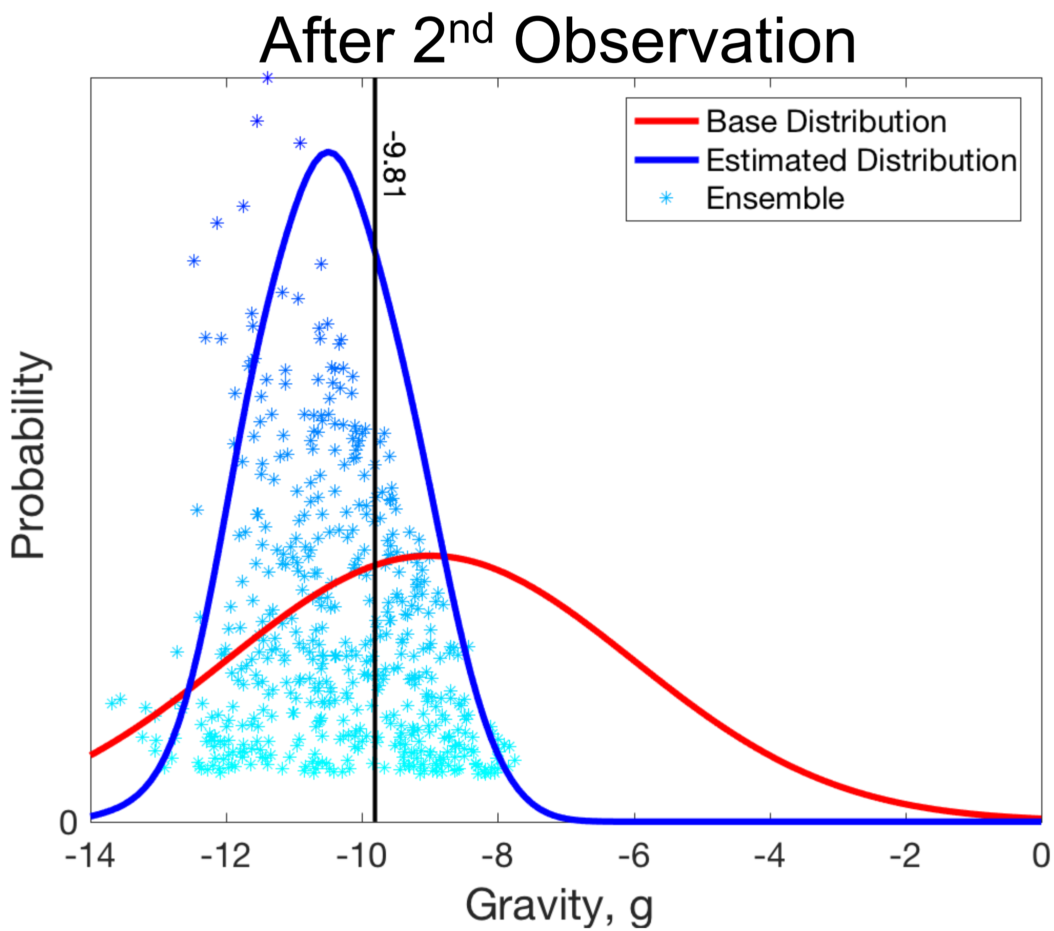
Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.



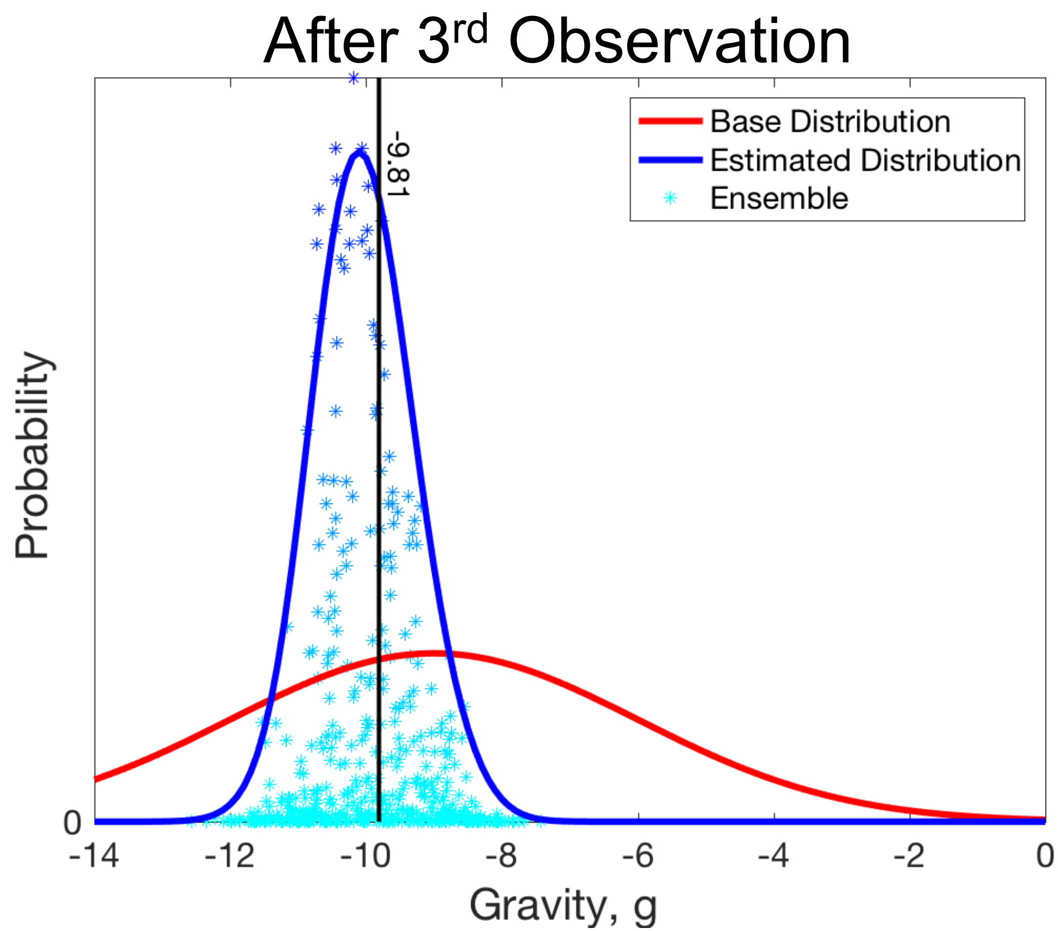
Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.



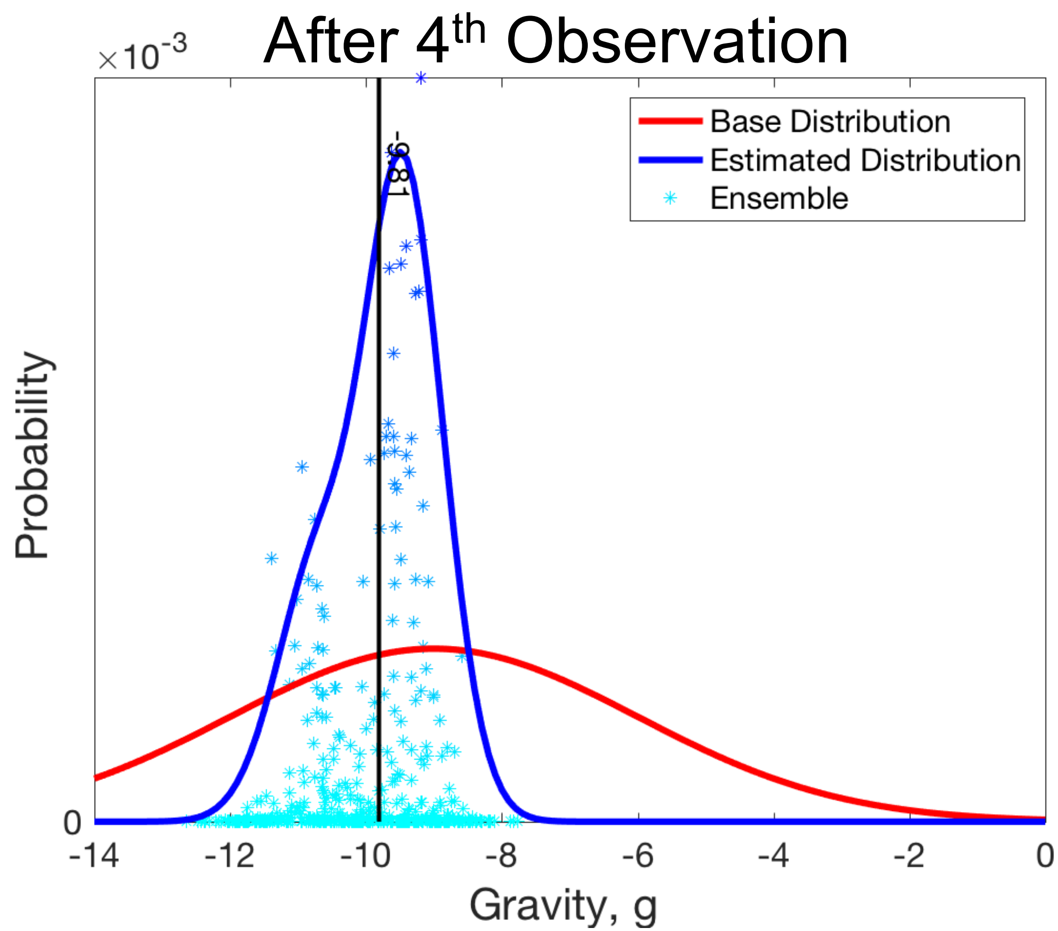
Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.



Data Assimilation: What can it do?

Estimating model parameter: gravity.
Estimate improves, gets more certain with more observations.



Data Assimilation: What can it do?

Learning about the model:

- Is the model estimate of uncertainty accurate?
- What is bias (mean error) of model forecasts?

Data Assimilation: What can it do?

Learning about the observations:

- What is observation error variance?
- What is bias (mean error) of observations?
- How valuable is each observation?
- Designing observation system:
  - Better to observe u and v?
  - Two observations at time 1, none at time 2?
  - One good instrument or two bad ones?

Data Assimilation: What can it do?

Learning about the ’external forcing’:

Example: Suppose there is a strong wind blowing.
We don’t have a (dynamical) forecast model for the wind.
Wind has strong influence on trajectory.

Can estimate wind from observations of ball.

Geophysical examples:

- Soil moisture model forced by precipitation.
- Atmosphere model forced by sea surface temperature.
- Upper atmosphere forced by solar inputs.

Data Assimilation: What can it do?

- Analyses and forecasts of state variables.
- Smoothing estimates of state variables.
- Estimate model parameters.
- Estimate initial conditions.
- Estimate model errors.
- Estimate observing system errors.
- Quantitatively design observing systems.
- Estimate external forcing.
- Estimate anything correlated with model/observations.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

Data Assimilation: A general description.

Ball example is in a 2-dimensional space, easy to visualize.

Really a 4-dimensional ‘phase’ space including velocity: u, v.

Atmosphere, ocean, land, coupled models are BIG.

But, still just a ‘ball’ moving in a HUGE phase space.

All data assimilation capabilities will still work.

Data Assimilation: A general description.

A time-varying state-vector $\mathbf{x}_t$,

Data Assimilation: A general description.

A time-varying state-vector $\mathbf{x}_t$,

Times t_k with observations: $k = 1, 2, \ldots;$ $\quad t_{k+1} > t_k \geq t_0,$

Data Assimilation: A general description.

A time-varying state-vector $\mathbf{x}_t$,

Times t_k with observations: $k = 1, 2, \ldots;\quad t_{k+1} > t_k \geq t_0$,

Observations at t_k related to $\mathbf{x}_{t_k}$; $\quad \mathbf{y}_k = h_k\left(\mathbf{x}_{t_k}\right) + \nu_k, \qquad (1)$

Observations

Forward Operator

Observation Error

Data Assimilation: A general description.

A time-varying state-vector $\mathbf{x}_t$,

Times t_k with observations: $k = 1, 2, \ldots ;\qquad t_{k+1} > t_k \geq t_0,$

Observations at t_k related to $\mathbf{x}_{t_k}$; $\quad \mathbf{y}_k = h_k\left(\mathbf{x}_{t_k}\right) + \nu_k, \qquad (1)$

Observation error is zero mean, normal, $\nu_k = N(0, \mathbf{R}_k), \qquad (2)$

Observation
Error
Covariance

Data Assimilation: A general description.

A time-varying state-vector $\mathbf{x}_t$,

Times t_k with observations: $k = 1, 2, \dots;$ $\quad t_{k+1} > t_k \geq t_0,$

Observations at t_k related to $\mathbf{x}_{t_k}$; $\quad \mathbf{y}_k = h_k(\mathbf{x}_{t_k}) + \nu_k,$ (1)

Observation error is zero mean, normal, $\nu_k = N(0, \mathbf{R}_k),$ (2)

A forecast model m for the state-vector; $\mathbf{x}_{t_{k+1}} = m_{k:k+1}(\mathbf{x}_{t_k})$ (3)

Data Assimilation: A general description.

A time-varying state-vector $\mathbf{x}_t$,

Times t_k with observations: $k = 1, 2, \ldots$; $\quad t_{k+1} > t_k \geq t_0$,

Observations at t_k related to $\mathbf{x}_{t_k}$; $\quad \mathbf{y}_k = h_k(\mathbf{x}_{t_k}) + \nu_k, \qquad (1)$

Observation error is zero mean, normal, $\nu_k = N(0, \mathbf{R}_k), \qquad (2)$

A forecast model m for the state-vector; $\mathbf{x}_{t_{k+1}} = m_{k:k+1}(\mathbf{x}_{t_k}) \qquad (3)$

m can have deterministic and stochastic parts;

$$m_{k:k+1}(\mathbf{x}_{t_k}) = f_{k:k+1}(\mathbf{x}_{t_k}) + g_{k:k+1}(\mathbf{x}_{t_k}). \qquad (4)$$

Data Assimilation: A general description.

Define the set of all observations taken no later than time t_k:

$$\mathbf{Y}_{t_k} = \{\mathbf{y}_t; t \leq t_k\} \qquad (5)$$

Problems of interest are:

Analysis: $P(\mathbf{x}_t|\mathbf{Y}_{t_k}), \quad t = t_k \qquad (6)$

Forecast: $P(\mathbf{x}_t|\mathbf{Y}_{t_k}), \quad t > t_k \qquad (7)$

Smoother: $P(\mathbf{x}_t|\mathbf{Y}_{t_k}), \quad t < t_k \qquad (8)$

Data Assimilation: A general description.

Define the set of all observations taken no later than time t_k:

$$\mathbf{Y}_{t_k} = \{\mathbf{y}_t; t \leq t_k\} \tag{5}$$

Problems of interest are:

Analysis: $P(\mathbf{x}_t|\mathbf{Y}_{t_k}), \quad t = t_k$ (6)

Forecast: $P(\mathbf{x}_t|\mathbf{Y}_{t_k}), \quad t > t_k$ (7)

Smoother: $P(\mathbf{x}_t|\mathbf{Y}_{t_k}), \quad t < t_k$ (8)

Note: could also replace $\mathbf{x}_t$ with any of the other things data assimilation can estimate: parameters, initial conditions, …

Data Assimilation: A general description.

Forecasts of state, $\mathbf{x}$ are obtained from model.

Need to update forecast state given new observations:

$$P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_k}\right) = \mathrm{P}\left(\mathbf{x}_{t_k} | \mathbf{y}_k, \mathbf{Y}_{t_{k-1}}\right)$$

Bayes' rule:

$$P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_k}\right) = \frac{P\left(\mathbf{y}_k | \mathbf{x}_{t_k}, \mathbf{Y}_{t_{k-1}}\right) P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_{k-1}}\right)}{P\left(\mathbf{y}_k | \mathbf{Y}_{t_{k-1}}\right)} \qquad (9)$$

Observation errors uncorrelated in time:

$$P\left(\mathbf{y}_k | \mathbf{x}_{t_k}, \mathbf{Y}_{t_{k-1}}\right) = P\left(\mathbf{y}_k | \mathbf{x}_{t_k}\right)$$

Denominator in (9) is normalization, makes update a pdf.

Data Assimilation: A general description.

Probability after new observation:

Prior (forecast)

Likelihood

$$P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k | \mathbf{x}) P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_{k-1}}\right)}{Normalization} \tag{10}$$

Posterior (analysis).

Data Assimilation: A general description.

Probability after new observation:

Prior (forecast)

Likelihood

$$P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k|\mathbf{x})P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{Normalization} \tag{10}$$

Posterior (analysis).

Forecasts produced by applying model to analysis.

Smoother can be derived from a similar Bayesian analysis.

Data Assimilation: A general description.

Aside: Derivation of generalized Bayes:

$$P(A,B) = P(A|B)\,P(B) = P(B|A)P(A) \quad \text{(a1)}$$

Data Assimilation: A general description.

Aside: Derivation of generalized Bayes:

$$P(A,B) = P(A|B)\,P(B) = P(B|A)P(A) \quad \text{(a1)}$$

$$P(A,B,C) = P\big(A,(B,C)\big) = P(A|B,C)P(B,C) \quad \text{(a2)}$$

$$P(A,B,C) = P\big(B,(A,C)\big) = P(B|A,C)P(A,C) \quad \text{(a3)}$$

Extend (a1) to get (a2) and (a3).

Data Assimilation: A general description.

Aside: Derivation of generalized Bayes:

$$P(A,B) = P(A|B)\;P(B) = P(B|A)P(A) \quad \text{(a1)}$$

$$P(A,B,C) = P\big(A,(B,C)\big) = P(A|B,C)P(B,C) \quad \text{(a2)}$$

$$P(A,B,C) = P\big(B,(A,C)\big) = P(B|A,C)P(A,C) \quad \text{(a3)}$$

Solve from (a2), (a3).

$$P(A|B,C) = \frac{P(B|A,C)P(A,C)}{P(B,C)} \quad \text{(a4)}$$

Ratio from (a1).

$$\frac{P(A,C)}{P(B,C)} = \frac{P(A|C)P(C)}{P(B|C)P(C)} = \frac{P(A|C)}{P(B|C)} \quad \text{(a5)}$$

Substitute (a4) in (a5).

$$P(A|B,C) = \frac{P(B|A,C)P(A|C)}{P(B|C)}$$

Data Assimilation: A general description.

Aside: Derivation of generalized Bayes:

$$P(A|B,C) = \frac{P(B|A,C)P(A|C)}{P(B|C)}$$

$$P\left(\mathbf{x}_{t_k}|\mathbf{y}_k,\mathbf{Y}_{t_{k-1}}\right) = \frac{P\left(\mathbf{y}_k|\mathbf{x}_{t_k},\mathbf{Y}_{t_{k-1}}\right)P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{P\left(\mathbf{y}_k|\mathbf{Y}_{t_{k-1}}\right)}$$

Data Assimilation: A general description.

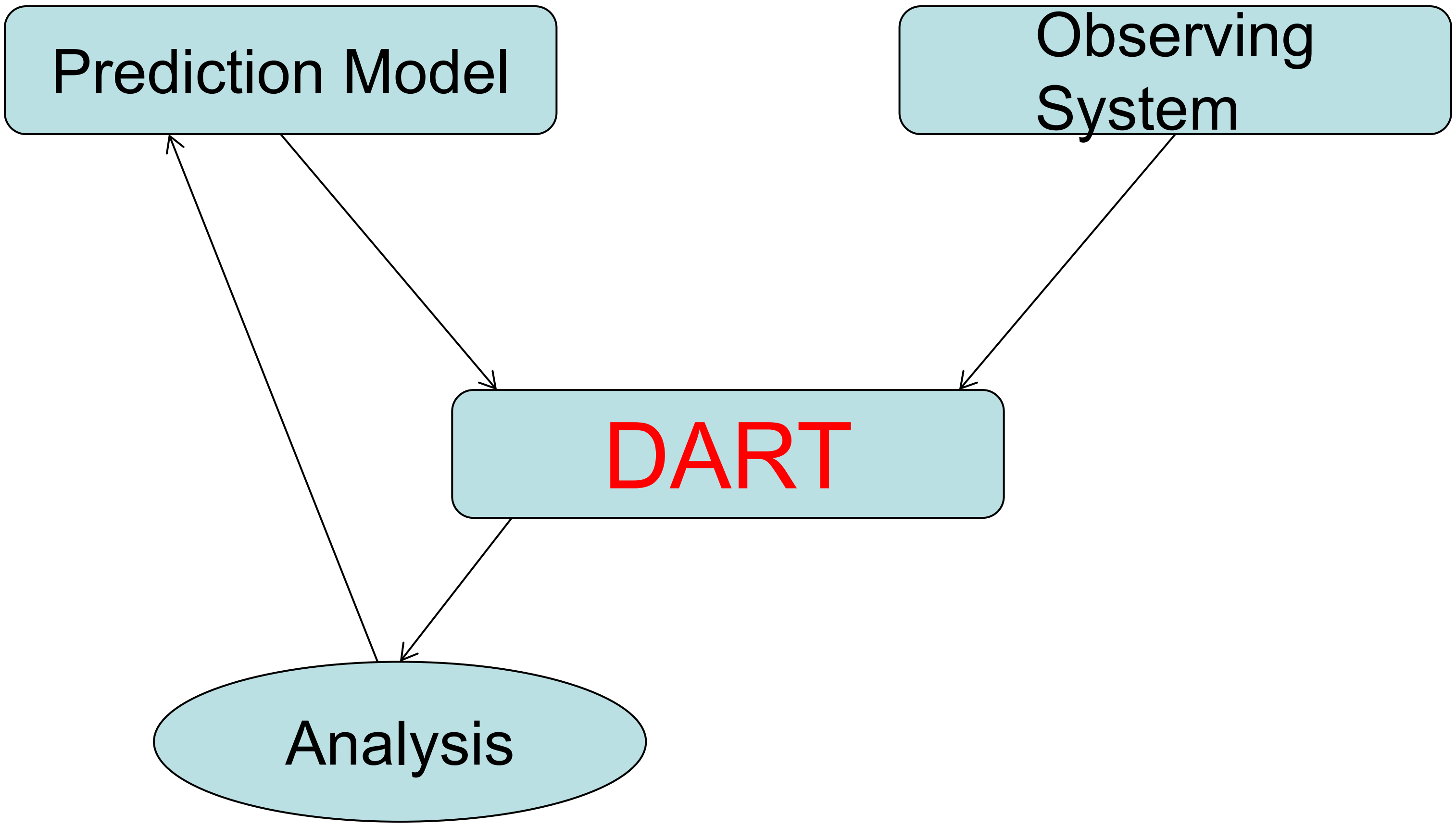
- Data Assimilation is stochastic.
- Bayes can be used to define the problem.
- General, but still some assumptions:
  - Observation error has zero mean.
  - Observation error is gaussian.
  - Observation error uncorrelated in time.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

The Data Assimilation Research Testbed (DART)

DART provides data assimilation 'glue' to build state-of-the-art ensemble forecast systems for even the largest models.



DART Goals

Provide State-of-the-Art Data Assimilation capability to:

- Prediction research scientists,
- Model developers,
- Observation system developers,

Who may not have any assimilation expertise.

DART Design Constraints

- Models small to huge.
- Few or many observations.
- Tiny to huge computational resources.
- Entry cost must be low.
- Competitive with existing methods for weather prediction:

  Scientific quality of results,

  Total computational effort.

Methods: Particle filter.

The method we have been using is too expensive.
Had to do 10,000 sample forecasts for this simple model.

This method scales horribly when the model size gets bigger.

Not practical for most geophysical applications.

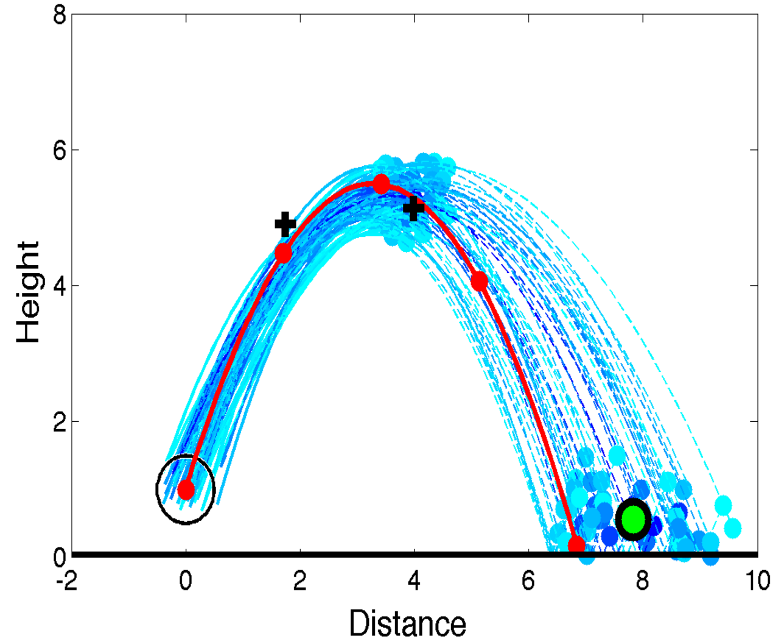
Methods: Particle filter.

The naïve method used here so far has:

- Independent evolving estimates,
- Associate probability with each estimate given observations.

Particle filter adds:

- Periodically eliminate some unlikely estimates,
- Duplicate likely estimates.



Methods: Particle filter.

Capabilities:

- Can represent arbitrary probability distribution.
- Trivial to implement.

Challenges:

- Needs many model forecasts even for small models.
- Scales poorly for large models.

Prospects:

- Research on improving scaling underway.
- Hybrid methods with ensemble filters promising.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

Methods: Variational

Simplify problem by only trying to find most likely trajectory.

This is the trajectory that maximizes the probability.

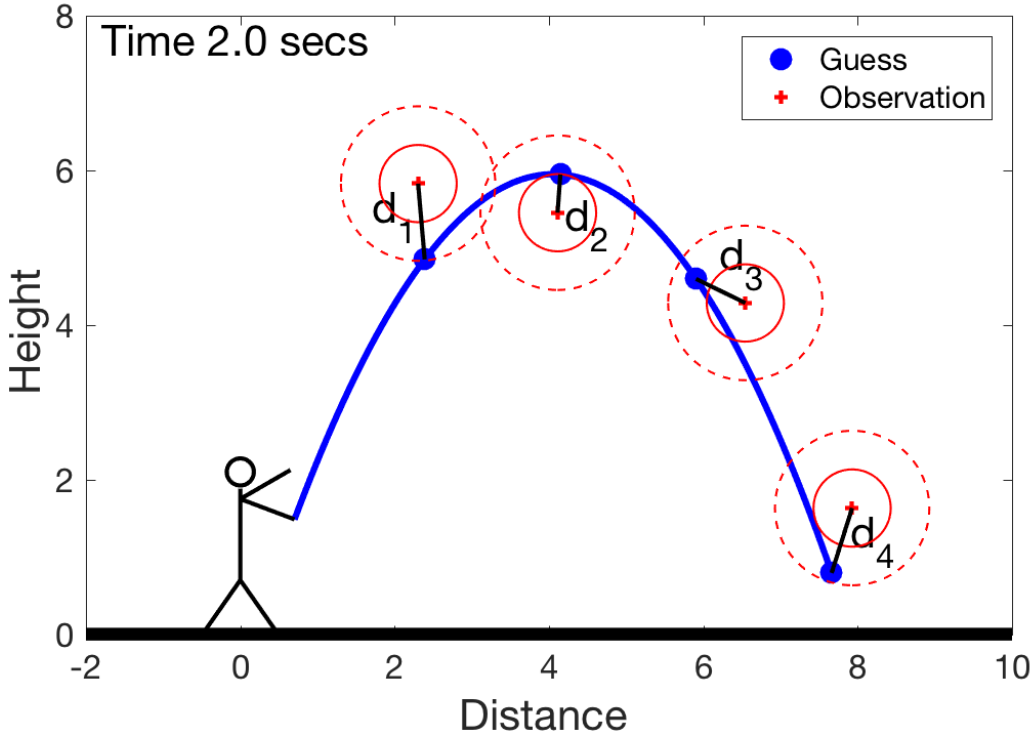
Same as trajectory that minimizes distance from observations.

(If the distance is normalized by observation error variances).

Find single most probable trajectory.

Maximize P:

$$P = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{\frac{-d_1^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi{\sigma_2}^2}} e^{\frac{-d_2^2}{2\sigma_2^2}} \frac{1}{\sqrt{2\pi{\sigma_3}^2}} e^{\frac{-d_3^2}{2\sigma_3^2}} \frac{1}{\sqrt{2\pi{\sigma_4}^2}} e^{\frac{-d_4^2}{2\sigma_4^2}}$$



Methods: Variational

Maximize P:

$$P = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{\frac{-d_1^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi{\sigma_2}^2}} e^{\frac{-d_2^2}{2\sigma_2^2}} \frac{1}{\sqrt{2\pi{\sigma_3}^2}} e^{\frac{-d_3^2}{2\sigma_3^2}} \frac{1}{\sqrt{2\pi{\sigma_4}^2}} e^{\frac{-d_4^2}{2\sigma_4^2}}$$

$\ln P$ will also have its maximum for this trajectory.

$-2\ln P$ will have its minimum for this trajectory.

$$-2\ln P = -2\sum_{i=1}^{4} \frac{1}{\sqrt{2\pi\sigma_i^2}} + \sum_{i=1}^{4} \frac{d_i^2}{\sigma_i^2}$$

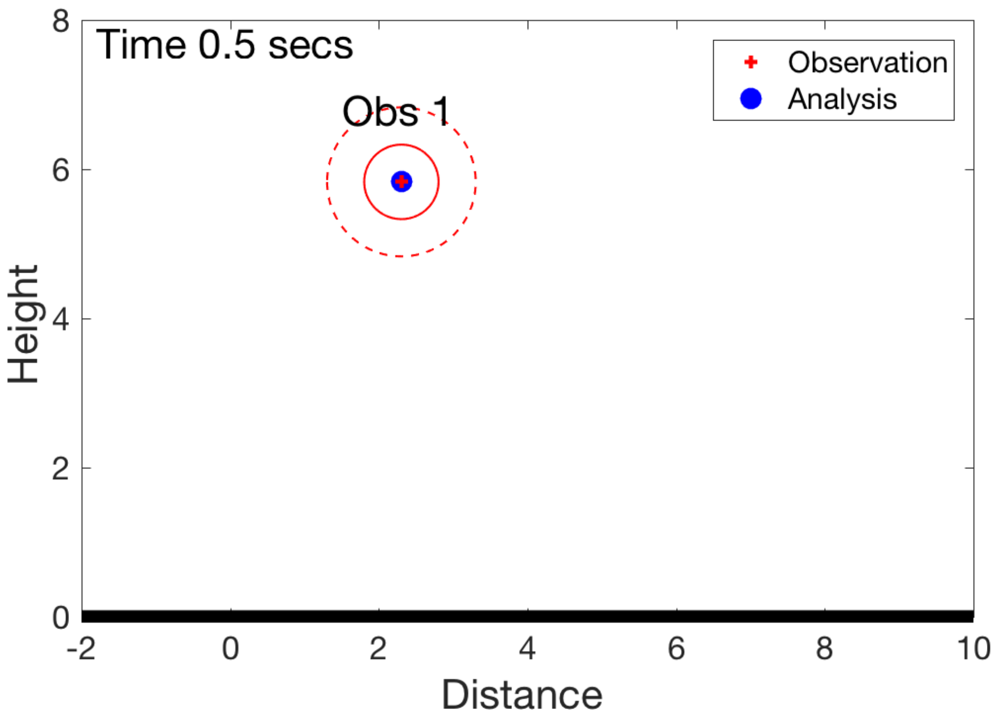
First term is constant, so minimize $\sum_{i=1}^{4} \frac{d_i^2}{\sigma_i^2}$

Methods: Variational

3D Variational assimilation.

Just consider observations at a single time.

Find state that minimizes distance to observations at this time.



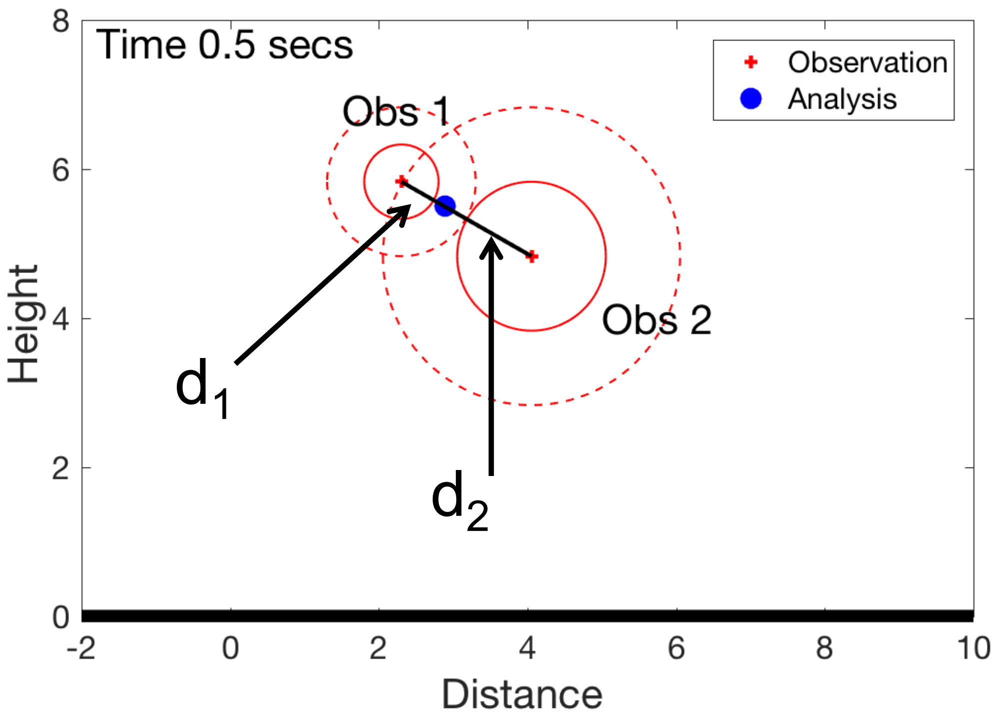
For single x, y observation at time 0.5, solution is just the observation itself (not very exciting).

Methods: Variational

3D Variational assimilation.

Two observations of x, y at time 0.5.

One has error variance that is twice as large as the other.



Best analysis minimizes:

$$d_1^2/\sigma_1^2 + d_2^2/\sigma_2^2$$

Where σ^2 is observation error variance.

Methods: Variational

3D Variational assimilation.

Cost function measures distance between a model state and the observations (normalized by the observation error).

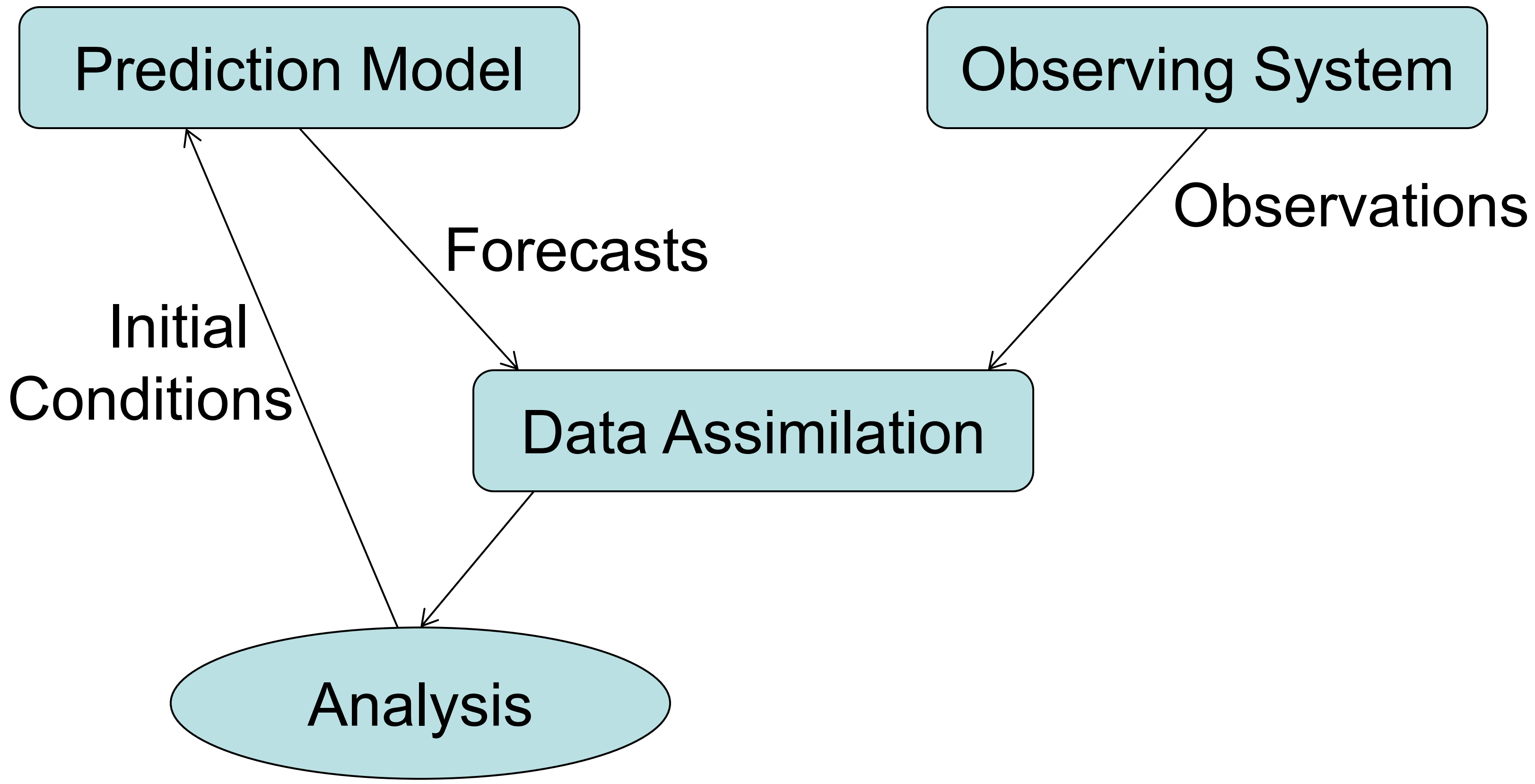
Generalized cost function consistent with earlier equations.

Minimize J to get the analysis vector x_a

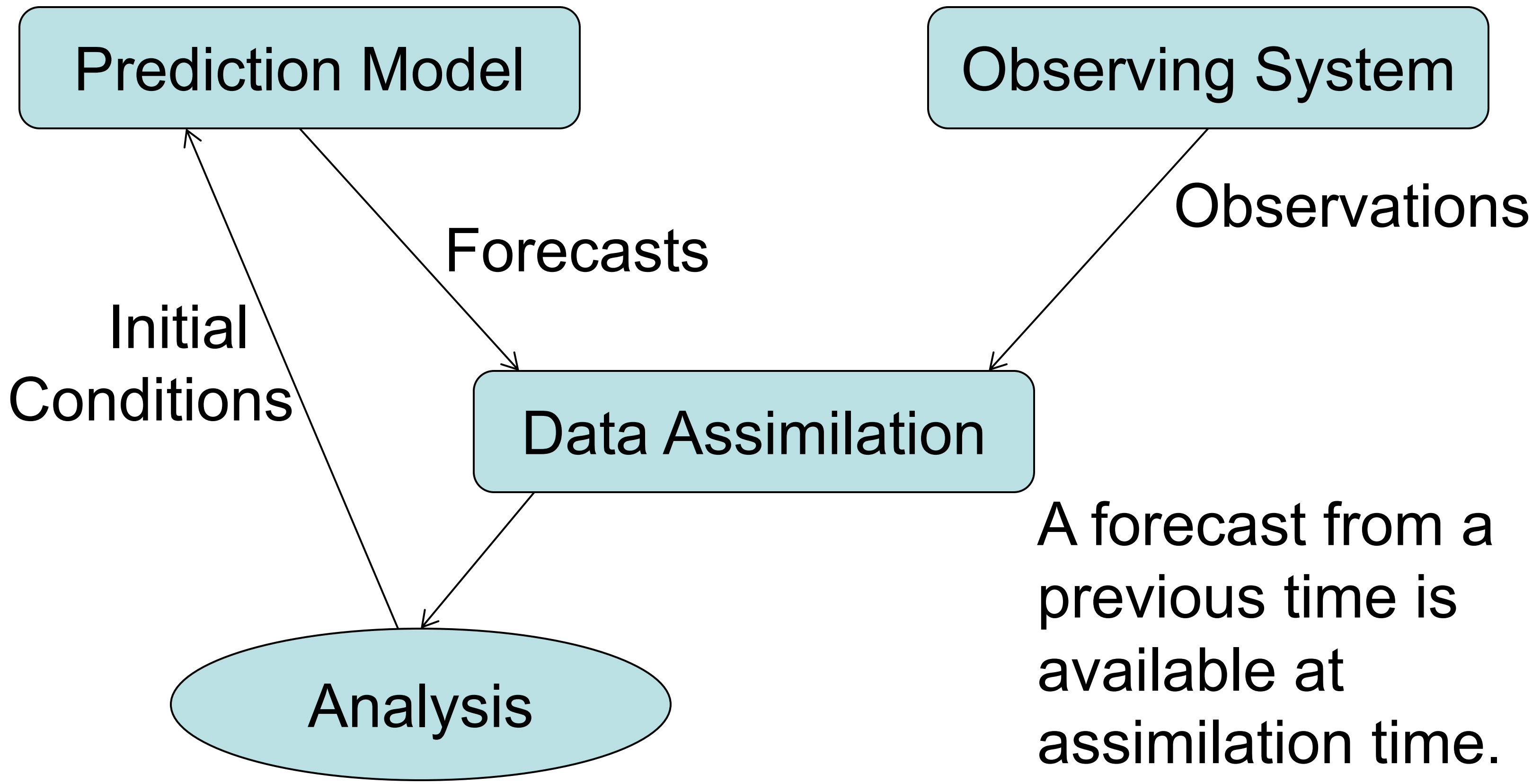
$$J(\mathbf{x}_a) = [h(\mathbf{x}_a) - \mathbf{y}\ ]^T \mathbf{R}^{-1} [h(\mathbf{x}_a) - \mathbf{y}\ ]$$

Methods: Variational

Sequential Data Assimilation Framework



3D Variational is done sequentially.



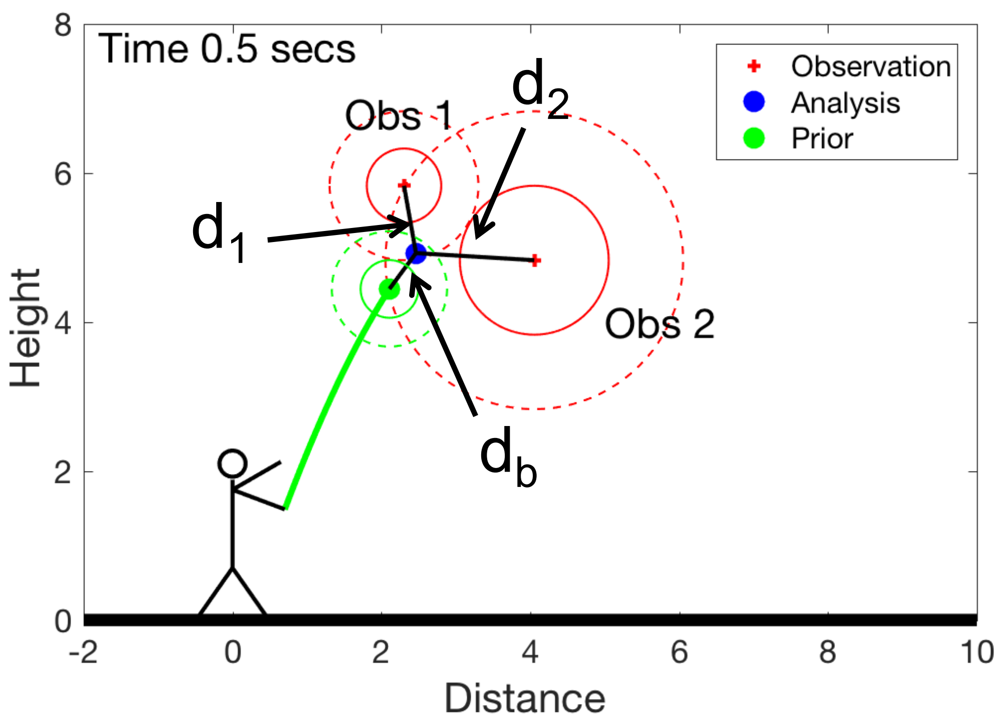
A forecast from a previous time is available at assimilation time.

Methods: Variational

3D Variational assimilation.

Two observations of x, y at time 0.5.

Also have a forecast from an earlier time, prior called $\mathbf{x}_f$.



Best analysis minimizes:

$$d_1^2/\sigma_1^2 + d_2^2/\sigma_2^2 + d_b^2/\sigma_b^2$$

Challenge: What is σ_b^2 the 'background error covariance' (green circles)? In other words, how much do we trust the forecast compared to observations?

Methods: Variational

3-Dimensional Variational Method:

Cost function measures distance between a model state, the observations, and a background (forecast) state.

Generalized cost function consistent with earlier equations.

Minimize J to get the analysis vector x_a

$$J(\mathbf{x}_a) = \left(\mathbf{x}_a - \mathbf{x}_f\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_a - \mathbf{x}_f\right) + [h(\mathbf{x}_a) - \mathbf{y}\ ]^T \mathbf{R}^{-1}[h(\mathbf{x}_a) - \mathbf{y}\ ]$$

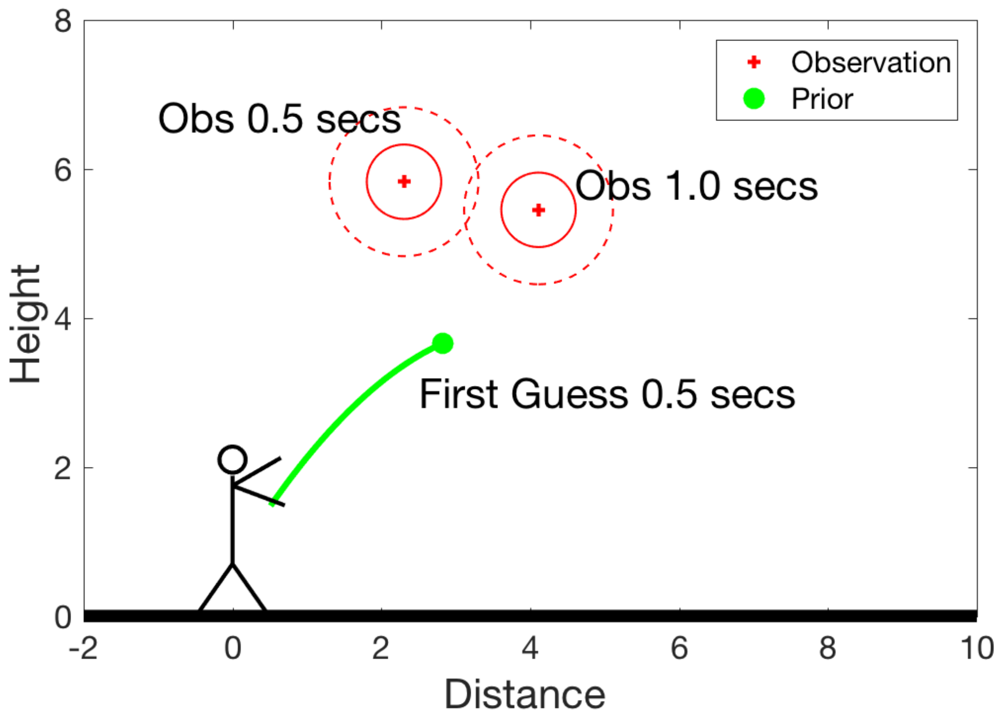
Challenge: What is background error covariance matrix, B?

Methods: Variational

4D Variational assimilation.

Two observations of x, y at time 0.5 and 1.0.

Also have a forecast from an earlier time valid at time 0.5.

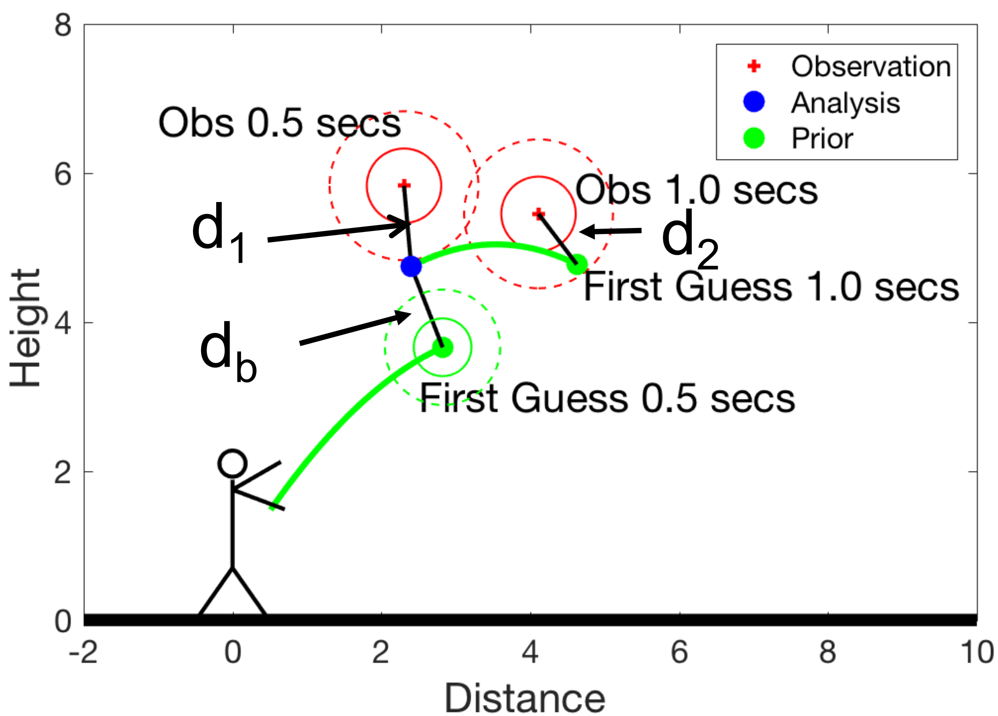


Methods: Variational

4D Variational assimilation.

Two observations of x, y at time 0.5 and 1.0.

Also have a forecast from an earlier time valid at time 0.5.



Find model state at time 0.5 that gives green forecast to time 1.0 and minimizes cost function:

$$d_1^2/\sigma_1^2 + d_2^2/\sigma_2^2 + d_b^2/\sigma_b^2$$

Methods: Variational

4-Dimensional Variational Method:

Cost function measures distance between a model trajectory, the observations, and a background (forecast) state.

Generalized cost function consistent with earlier equations.

Minimize J to get the analysis vector x_a

$J(\mathbf{x}_a) = \left(\mathbf{x}_a - \mathbf{x}_f\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_a - \mathbf{x}_f\right) +$	Background
$[h_1(\mathbf{x}_a) - \mathbf{y}_1]^T \mathbf{R}_1^{-1}[h_1(\mathbf{x}_a) - \mathbf{y}_1] +$	1st time observations
$\left[h_2\left(m_{1:2}(\mathbf{x}_a)\right) - \mathbf{y}_2\right]^T \mathbf{R}_2^{-1}\left[h_2\left(m_{1:2}(\mathbf{x}_\mathbf{a})\right) - \mathbf{y}_2\right]$	2nd time observations

Methods: Variational

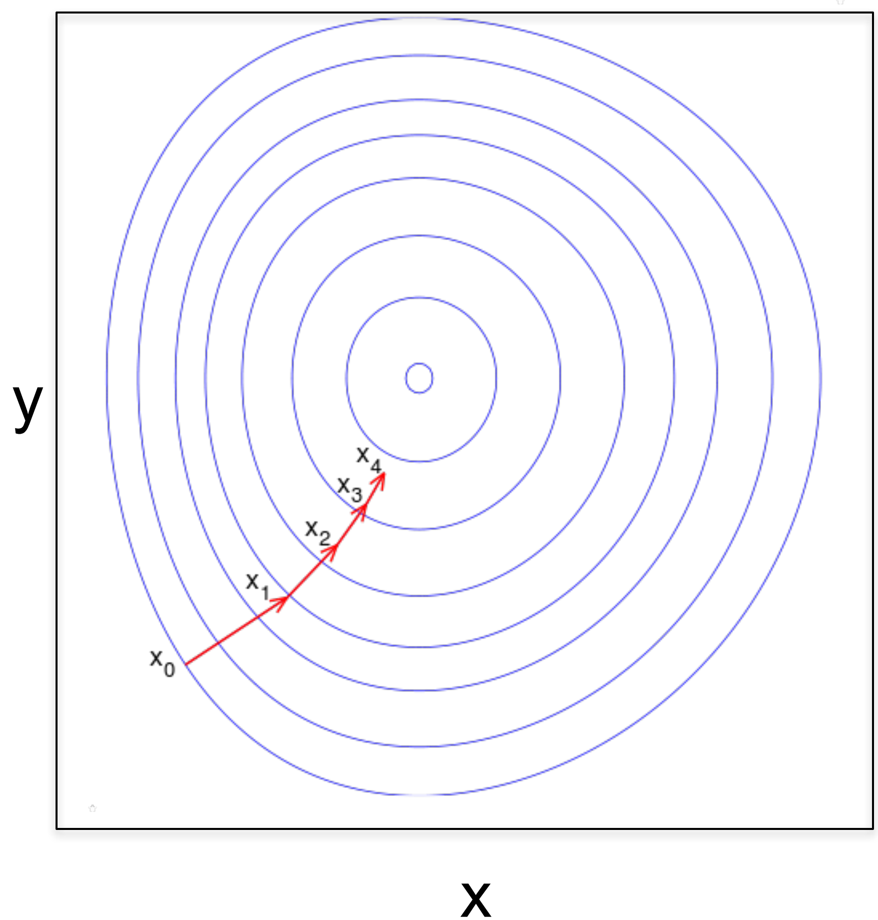
How to minimize these cost functions?

Standard gradient descent.

How to get the gradient for
large problems?

Must be very efficient.

There is a fast way
using linear tangent operator
of the model and its transpose
(or adjoint).



Methods: Variational

Gradient of J for 3D Variational:

$$J(\mathbf{x}_a) = \left(\mathbf{x}_a - \mathbf{x}_f\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_a - \mathbf{x}_f\right) + [h(\mathbf{x}_a) - \mathbf{y}\,]^T \mathbf{R}^{-1}[h(\mathbf{x}_a) - \mathbf{y}\,]$$

Gradient of J is:

$$\nabla J(\mathbf{x}_a) = 2\mathbf{B}^{-1}\left(\mathbf{x}_a - \mathbf{x}_f\right) + 2\mathbf{H}^T \mathbf{R}^{-1}[h(\mathbf{x}_a) - \mathbf{y}]$$

$\mathbf{H}$ is the Jacobian (called the tangent linear) of forward operator h.

The ith row and jth column of **H** is: $\mathbf{H}_{i,j}(\mathbf{x}) = \dfrac{\partial h_i}{\partial x_j}$

Where h_i is the forward operator for the ith observation and x_j is the jth component of the state vector.

The transpose $\mathbf{H}^T$ is called the adjoint of $\mathbf{H}$.

Methods: Variational

Gradient of J for 3D Variational:

$$\nabla J(\mathbf{x}_a) = 2\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_f) + 2\mathbf{H}^T\mathbf{R}^{-1}[h(\mathbf{x}_a) - \mathbf{y}]$$

Once the adjoint $\mathbf{H}^T$ is computed, gradient is straightfoward.

In trivial example here, h is linear:

$\mathbf{H}$ is just matrix representation of h.

Methods: Variational

Gradient of J for 4D Variational:

$$J(\mathbf{x}_a) = \left(\mathbf{x}_a - \mathbf{x}_f\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_a - \mathbf{x}_f\right) + [h_1(\mathbf{x}_a) - \mathbf{y}_1]^T \mathbf{R}_1^{-1}[h_1(\mathbf{x}_a) - \mathbf{y}_1] + \left[h_2\left(m_{1:2}(\mathbf{x}_a)\right) - \mathbf{y}_2\right]^T \mathbf{R}_2^{-1}\left[h_2\left(m_{1:2}(\mathbf{x}_\mathbf{a})\right) - \mathbf{y}_2\right]$$

Gradient of first line is same as 3D Variational.

Last line leads to extra gradient term:

$$2\mathbf{M}_{1:2}{}^T \mathbf{H}_2^T \mathbf{R}_2^{-1}\left[h_2\left(m_{1:2}(\mathbf{x}_a)\right) - \mathbf{y}_2\right]$$

Methods: Variational

Gradient of J for 4D Variational:

$$J(\mathbf{x}_a) = \left(\mathbf{x}_a - \mathbf{x}_f\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_a - \mathbf{x}_f\right) + [h_1(\mathbf{x}_a) - \mathbf{y}_1]^T \mathbf{R}_1^{-1}[h_1(\mathbf{x}_a) - \mathbf{y}_1] +$$
$$\left[h_2\left(m_{1:2}(\mathbf{x}_a)\right) - \mathbf{y}_2\right]^T \mathbf{R}_2^{-1}\left[h_2\left(m_{1:2}(\mathbf{x}_\mathbf{a})\right) - \mathbf{y}_2\right]$$

Gradient of first line is same as 3D Variational.

Last line leads to extra gradient term:

$$2\mathbf{M}_{1:2}{}^T \mathbf{H}_2^T \mathbf{R}_2^{-1}\left[h_2\left(m_{1:2}(\mathbf{x}_a)\right) - \mathbf{y}_2\right]$$

$\mathbf{M}_{1:2}$ is the linear tangent of the forecast model, $m_{1:2}$.

$\mathbf{M}_{1:2}{}^T$ is the adjoint of the forecast model.

Methods: Variational

Ball example: state vector is $x, y, u, v;$ $\mathbf{x} \equiv \{x, y, u, v\}$

Model equations for one timestep advance.

$$\mathbf{x}_{t+\Delta t} = m(\mathbf{x}_t)$$

$$\begin{aligned} x_{t+\Delta t} &= x_t + u_t \Delta t \\ y_{t+\Delta t} &= y_t + v_t \Delta t + 0.5 g (\Delta t)^2 \\ u_{t+\Delta t} &= u_t \\ v_{t+\Delta t} &= v_t + g \Delta t \end{aligned}$$

Tangent linear $\left[\frac{\partial m_i}{\partial x_j}\right]$

$$\mathbf{M} = \begin{vmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Adjoint

$$\mathbf{M}^T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \Delta t & 0 & 1 & 0 \\ 0 & \Delta t & 0 & 1 \end{vmatrix}$$

Methods: Variational

Enhancing performance and speed, 3D and 4D Variational:

Make B matrix smaller and better conditioned:
- Transform model variables to minimize off-diagonal terms.
- Transform to make some variables unimportant, truncate.

Incremental variational:
- Minimize $J(\delta\mathbf{x})$where $\delta\mathbf{x} = \mathbf{x}_a - \mathbf{x}_f$
- Makes problem quadratic (like our ball example).
- Size of $\delta\mathbf{x}$ may be smaller than $\mathbf{x}$ after transforms.

Methods: Variational

Enhancing performance and speed, 4D Variational:

Tangent linear is for a given nonlinear trajectory.
Increments from optimization may violate linear assumption.
Outer/inner loop:

After some number of gradient descent steps (inner loop),
rerun nonlinear trajectory (outer loop).

Use reduced resolution/accuracy forecast model.

Lots of other cool numerical tricks and empirical accelerations.

Methods: Variational

Capabilities:

- Estimate maximum likelihood solution only.
- With enhancements works well for huge models.
- Has been state-of-the-art for weather prediction.

Challenges:

- Coding of tangent linear/adjoint can be time consuming.
- Good convergence of optimization may be challenging.
- Good B matrices may be hard to find.

Prospects:

- Weak constraint 4Dvar allows for model error.
- Hybrid methods with ensemble filters promising.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

Methods: Kalman Filter

Assumes:

linear model

Gaussian noise

$$m_{k:k+1}(\mathbf{x}_{t_k}) = f_{k:k+1}(\mathbf{x}_{t_k}) + g_{k:k+1}(\mathbf{x}_{t_k}).$$

Gaussian state

linear forward operator

$$\mathbf{y}_k = h_k(\mathbf{x}_{t_k}) + \nu_k$$

Gaussian observation error

Methods: Kalman Filter

Product of d-dimensional normals (gaussians) with means μ_1 and μ_2 and covariance matrics Σ_1 and Σ_1 is normal.

$$N(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = cN(\mu, \Sigma)$$

Methods: Kalman Filter

Product of d-dimensional normals (gaussians) with means μ_1 and μ_2 and covariance matrics Σ_1 and Σ_1 is normal.

$$N(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = cN(\mu, \Sigma)$$

Covariance: $\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$

Mean: $\mu = \Sigma(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)$

Methods: Kalman Filter

Product of d-dimensional normals (gaussians) with means μ_1 and μ_2 and covariance matrics Σ_1 and Σ_1 is normal.

$$N(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = cN(\mu, \Sigma) \qquad (11)$$

Covariance: $$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \qquad (12)$$

Mean: $$\mu = \Sigma(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2) \qquad (13)$$

Weight: $$c = \frac{1}{(2\pi)^{d/2}|\Sigma_1+\Sigma_2|^{1/2}} \exp\left\{-\frac{1}{2}[(\mu_2 - \mu_1)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_2 - \mu_1)]\right\}$$

We'll ignore the weight since we immediately normalize products to be PDFs.

Methods: Kalman Filter

Recall our earlier assimilation update equation.

$$P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k|\mathbf{x})P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{Normalization} \tag{10}$$

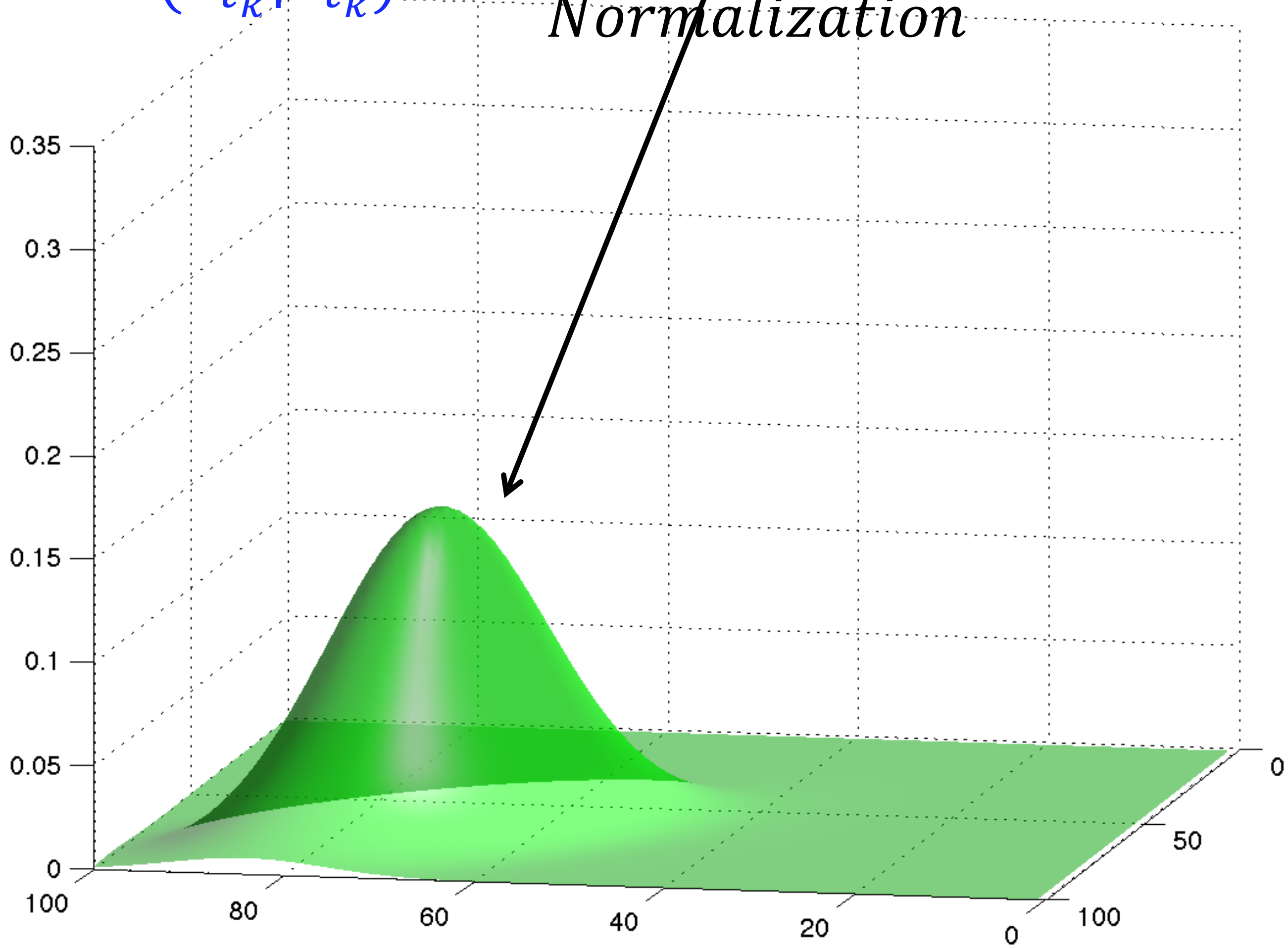
Numerator is just product of two gaussians.

Denominator just normalizes posterior to be a PDF.

Methods: Kalman Filter

Prior is gaussian, comes from forecast model.

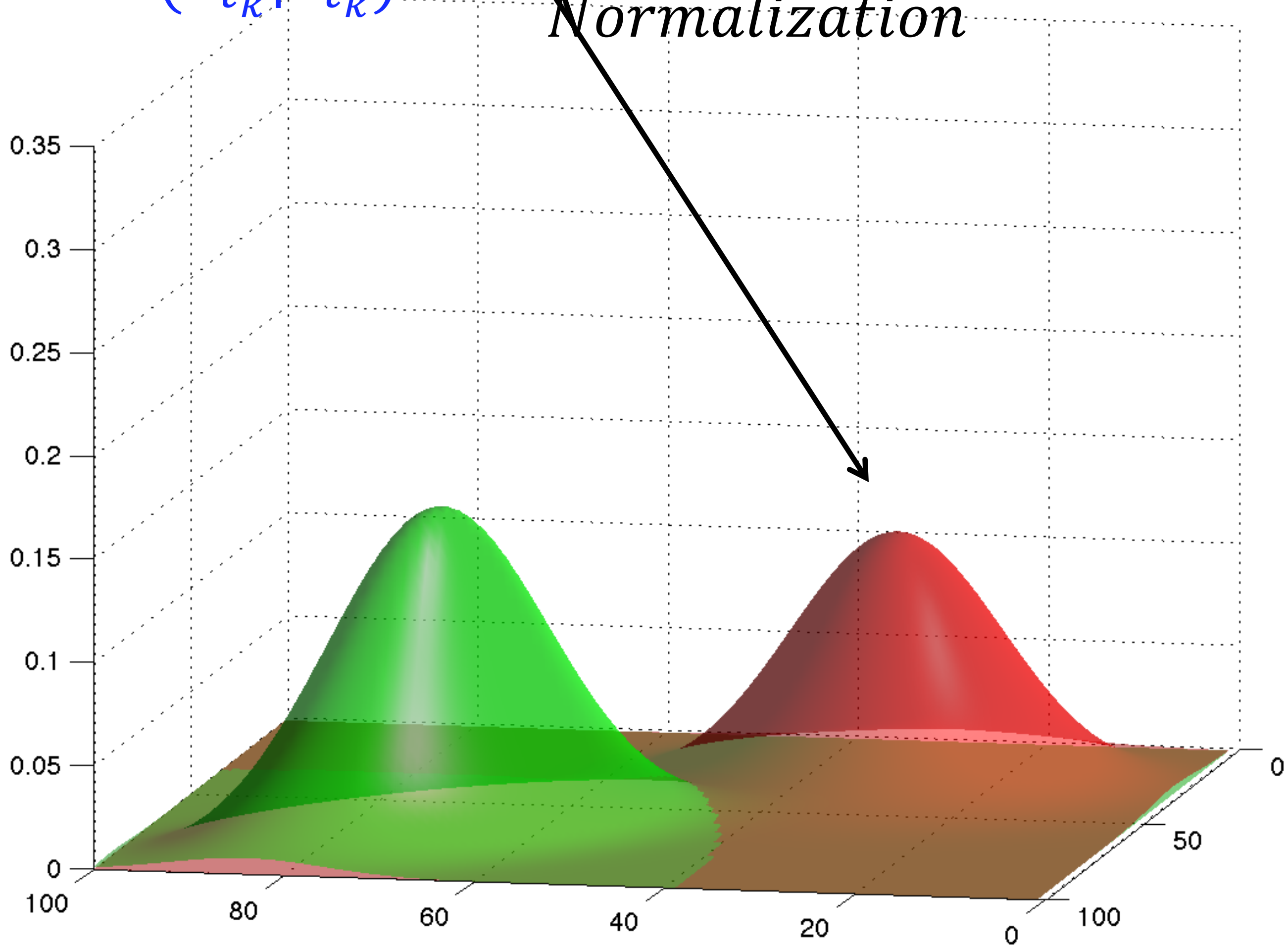
$$P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k | \mathbf{x}) P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_{k-1}}\right)}{Normalization} \quad (10)$$



Methods: Kalman Filter

Likelihood gaussian: mean is measured, covariance from designer.

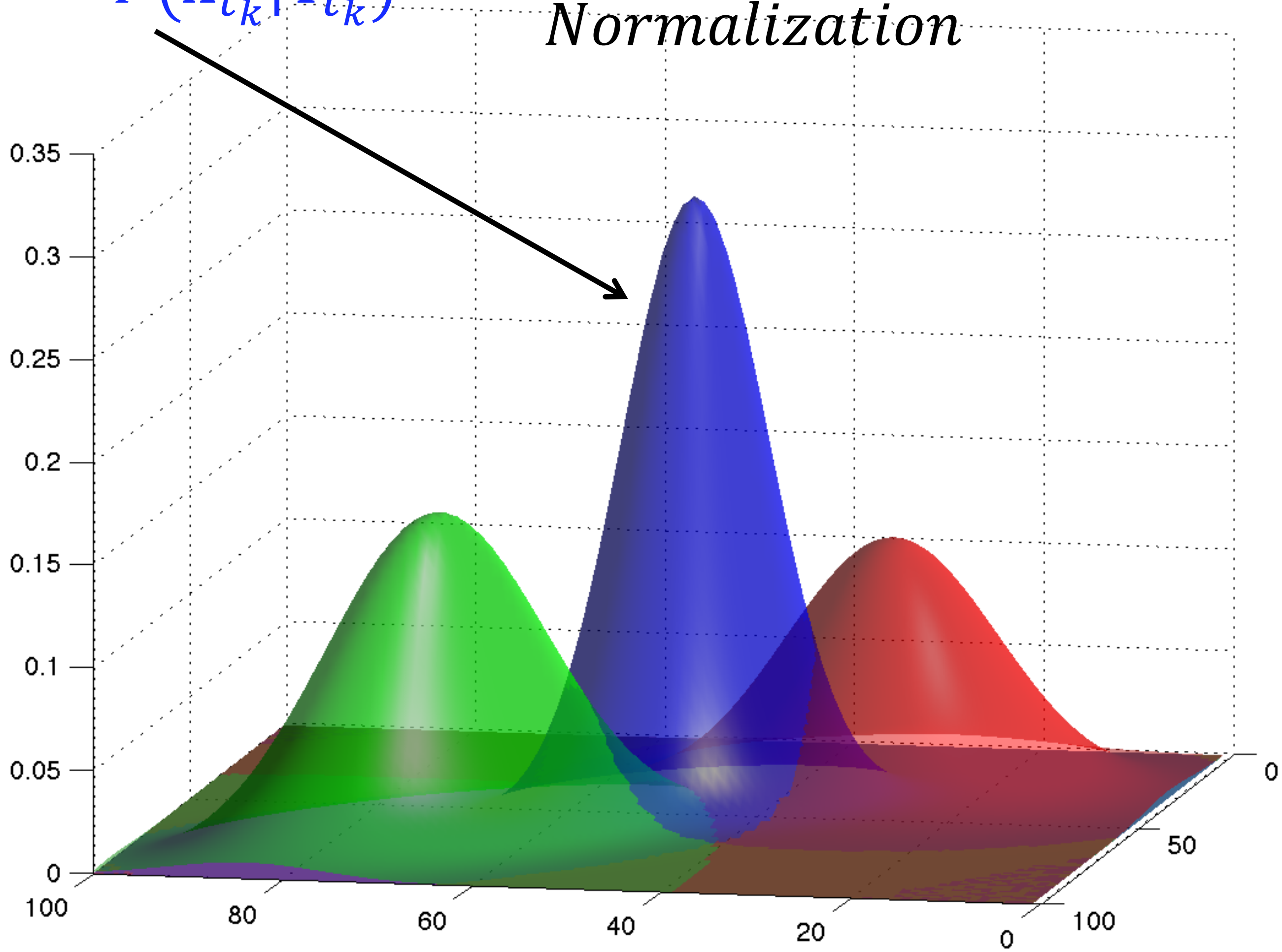
$$P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k | \mathbf{x}) P\left(\mathbf{x}_{t_k} | \mathbf{Y}_{t_{k-1}}\right)}{Normalization} \quad (10)$$



Methods: Kalman Filter

Posterior is gaussian, from (11).

$$P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k|\mathbf{x})P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{Normalization} \quad (10)$$



Methods: Kalman Filter

Kalman filter cost challenges:

Product of d-dimensional normals (gaussians) with means μ_1 and μ_2 and covariance matrics Σ_1 and Σ_1 is normal.

$$N(\mu_1, \Sigma_1) N(\mu_2, \Sigma_2) = cN(\mu, \Sigma) \qquad (11)$$

Covariance: $$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \qquad (12)$$

Mean: $$\mu = \Sigma(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2) \qquad (13)$$

Must store and invert covariance matrices.
Too big to store for large problems.
Too costly to invert, > $O(n^2)$.

Methods: Kalman Filter

Capabilities:

- Estimates normal approximation to probability density.
- Easy to apply with linear models.
- Huge literature with many extensions.

Challenges:

- Scales poorly for large models.
- Requires extensions for use with nonlinear models or h.

Prospects:

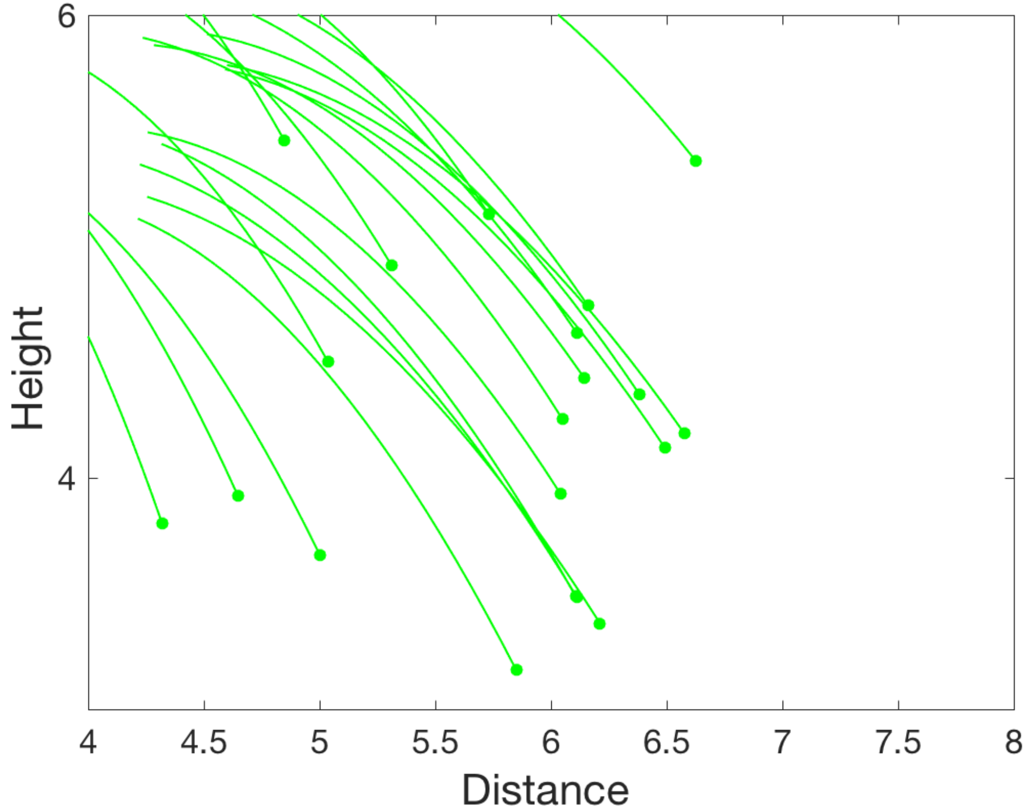
- Ensemble approximations avoid challenges.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

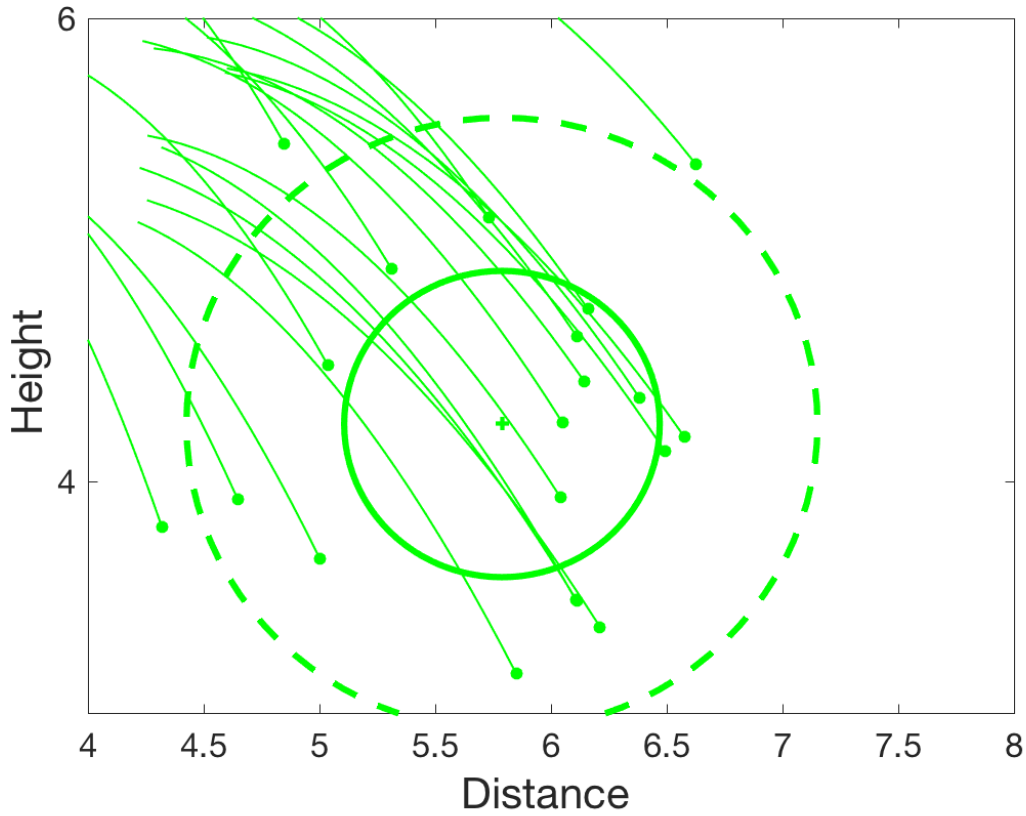
Methods: Ensemble Kalman Filter

1. Prior ensemble:



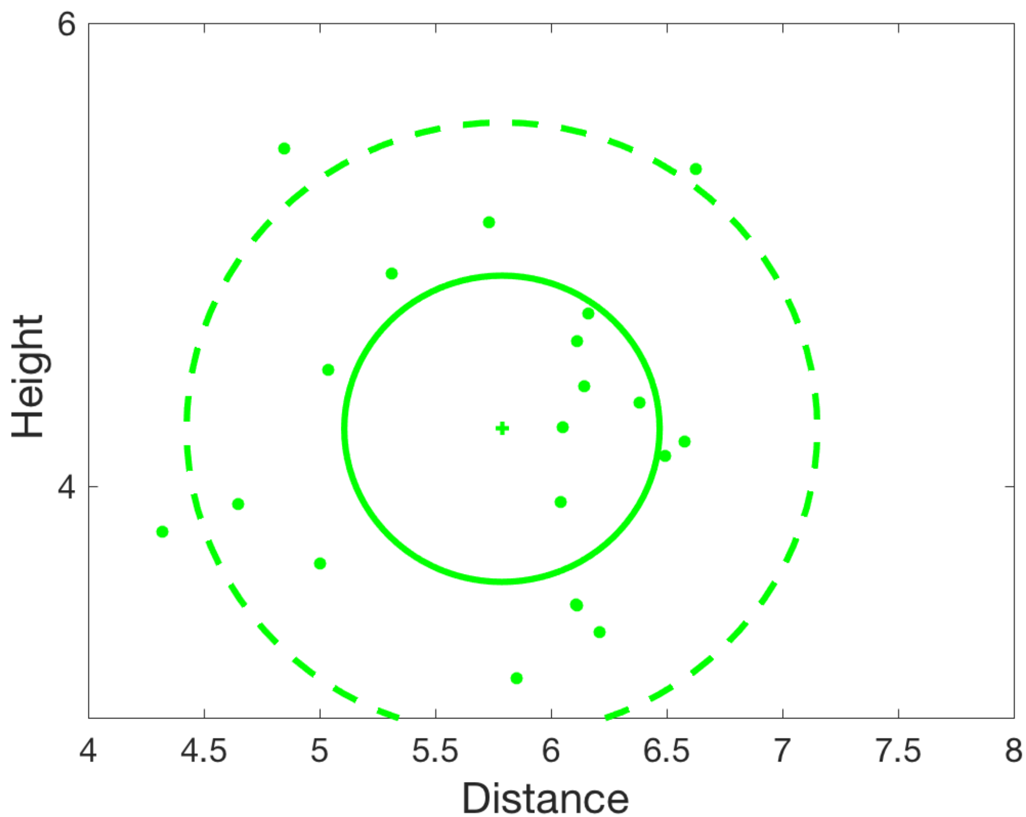
Methods: Ensemble Kalman Filter

1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).



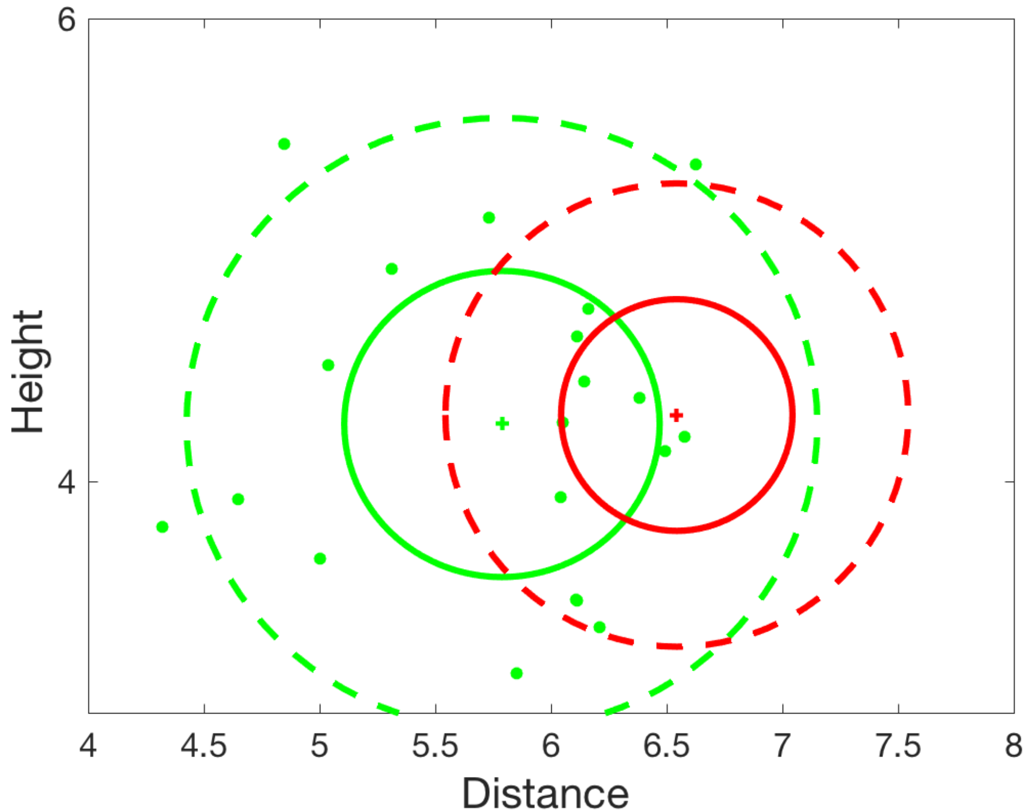
Methods: Ensemble Kalman Filter

1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).



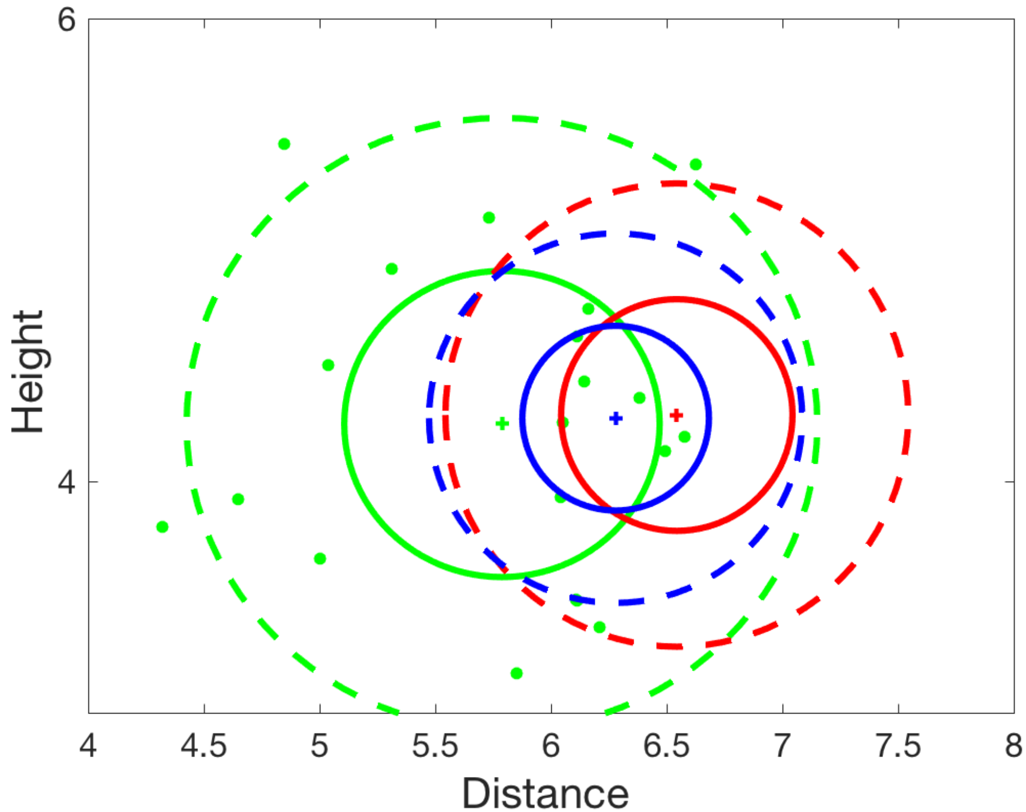
Methods: Ensemble Kalman Filter

1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).
2. Kalman filter product of observation likelihood (red) with prior (green)



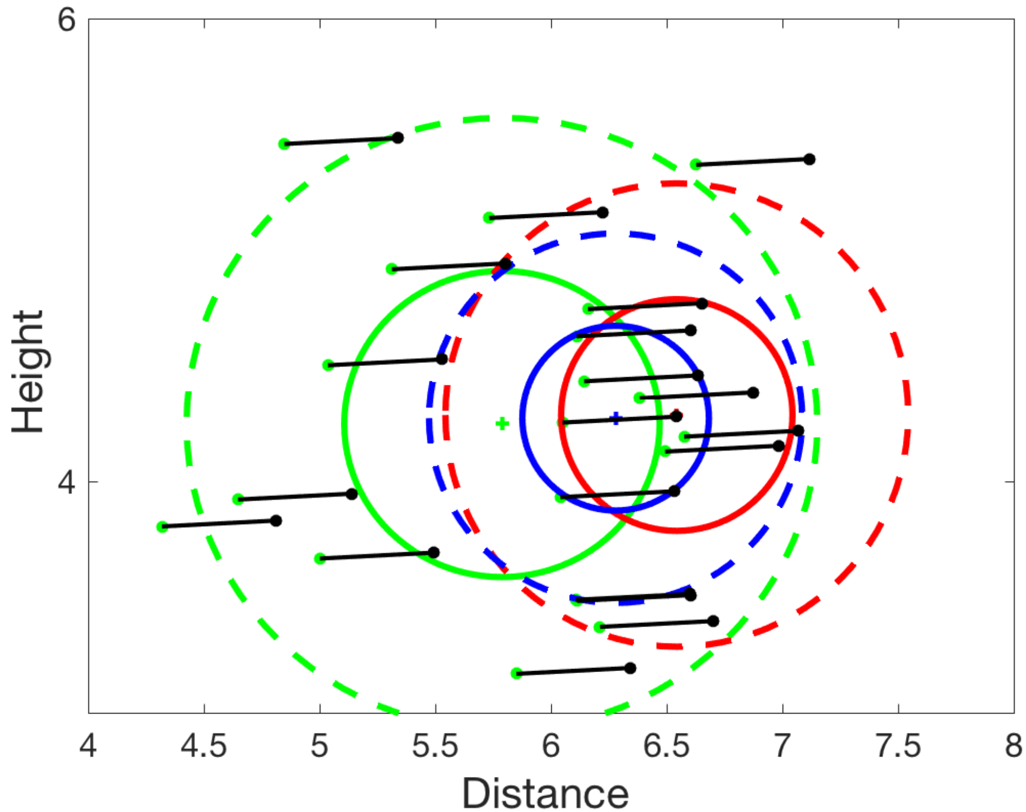
Methods: Ensemble Kalman Filter

1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).
2. Kalman filter product of observation likelihood (red) with prior (green) to get gaussian posterior (blue).



Methods: Ensemble Kalman Filter

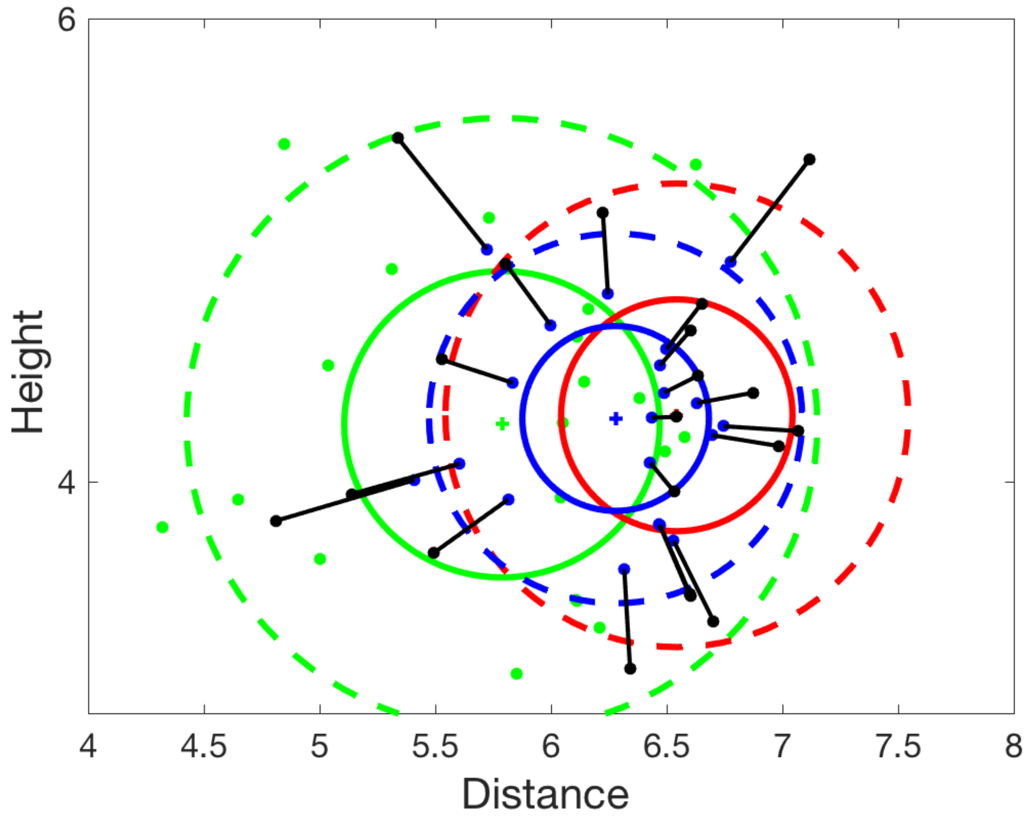
1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).
2. Kalman filter product of observations (red) with prior (green) to get gaussian posterior (blue).



3. Shift ensemble members to have posterior mean.

Methods: Ensemble Kalman Filter

1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).
2. Kalman filter product of observations (red) with prior (green) to get gaussian posterior (blue).

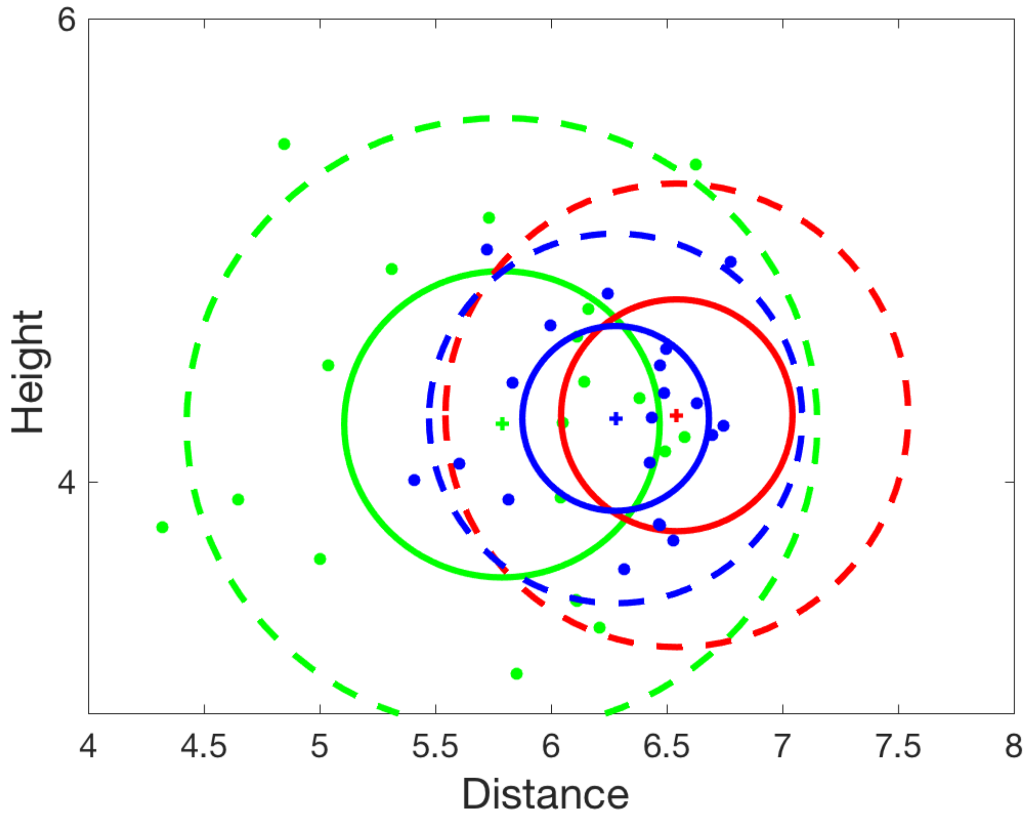


3. Shift ensemble members to have posterior mean.

4. Compact ensemble to have posterior covariance.

Methods: Ensemble Kalman Filter

1. Prior ensemble: fit a gaussian, sample mean and covariance (pdf is green contours).
2. Kalman filter product of observations (red) with prior (green) to get gaussian posterior (blue).

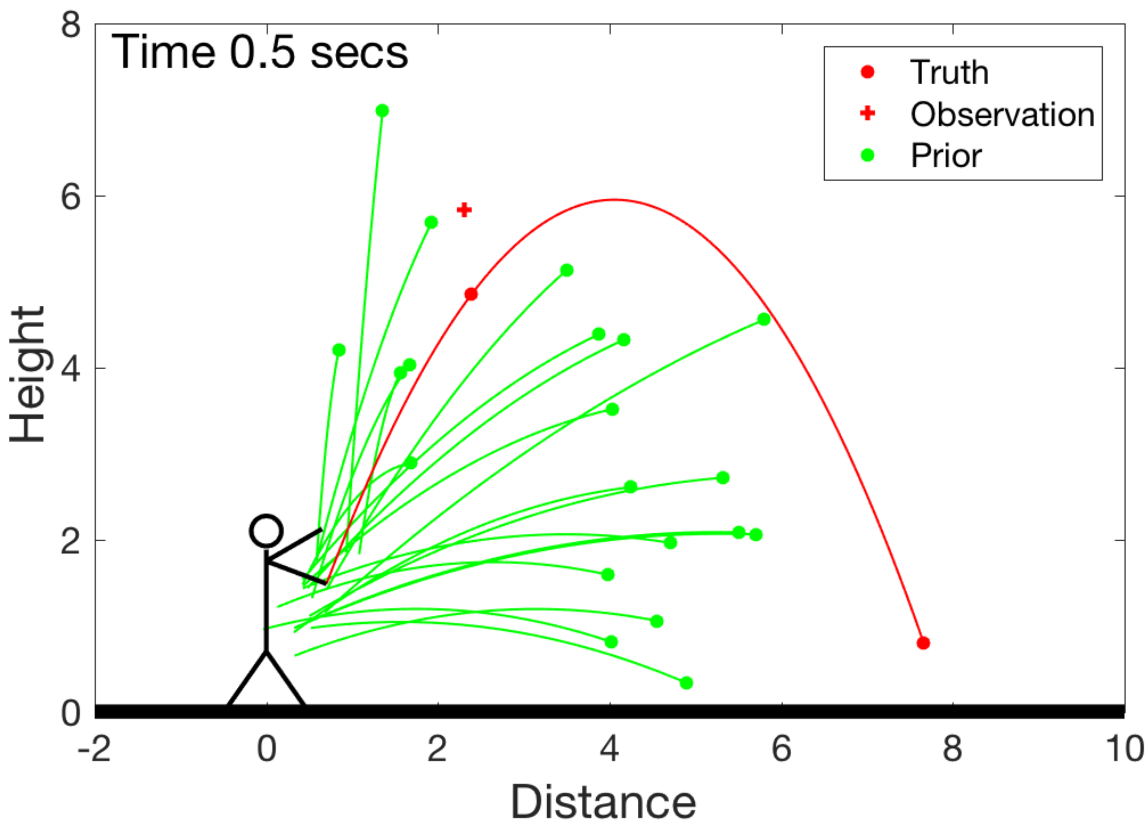


3. Shift ensemble members to have posterior mean.

4. Compact ensemble to have posterior covariance.

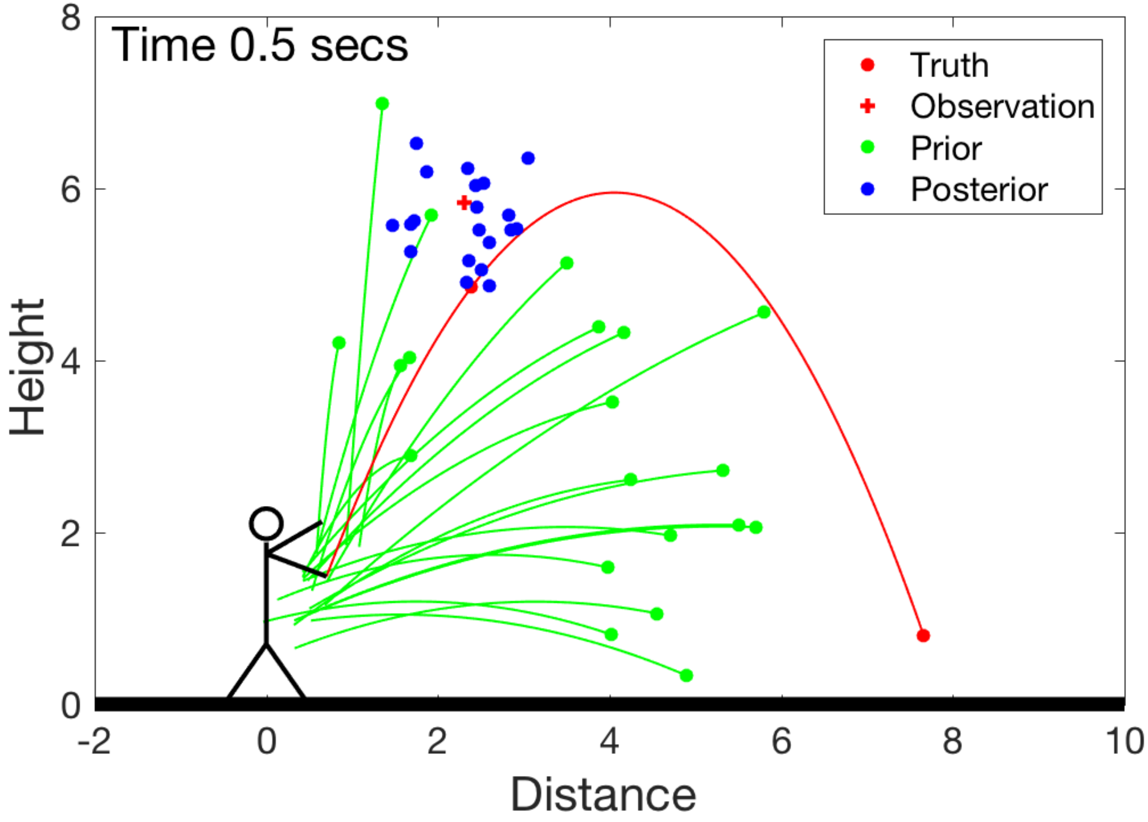
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



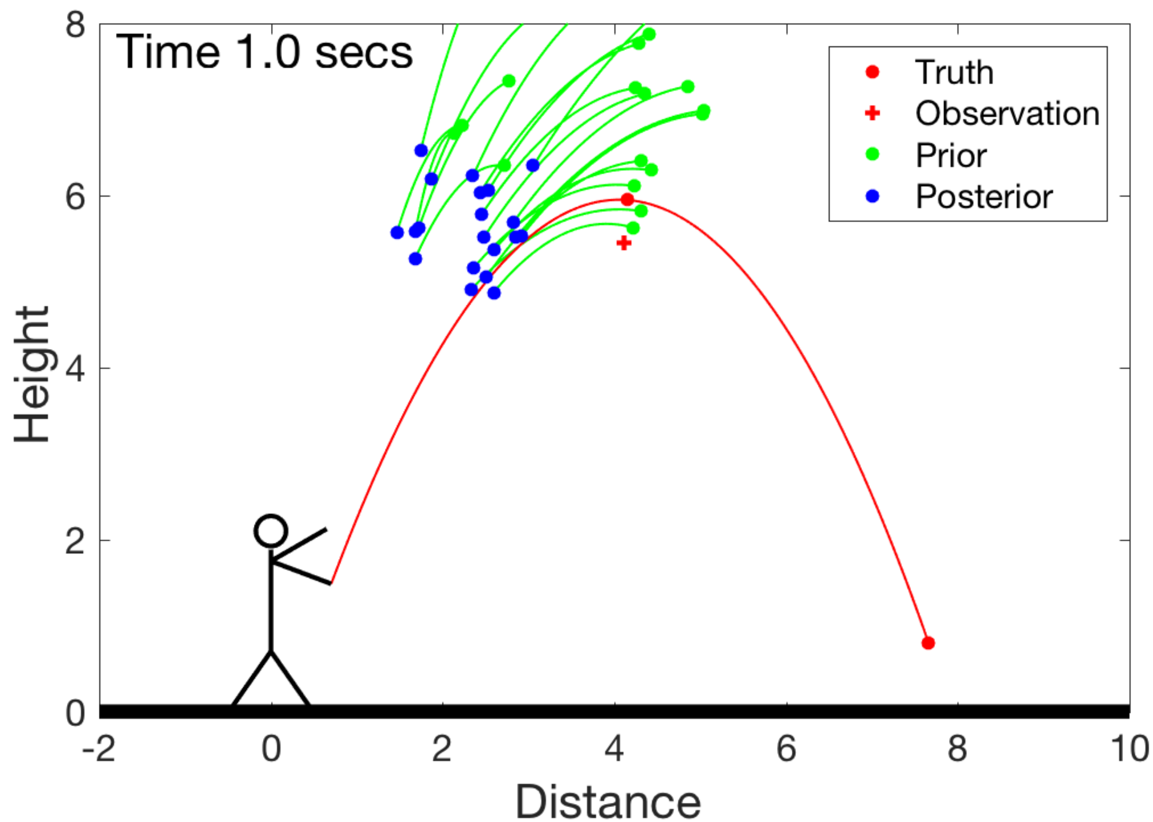
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



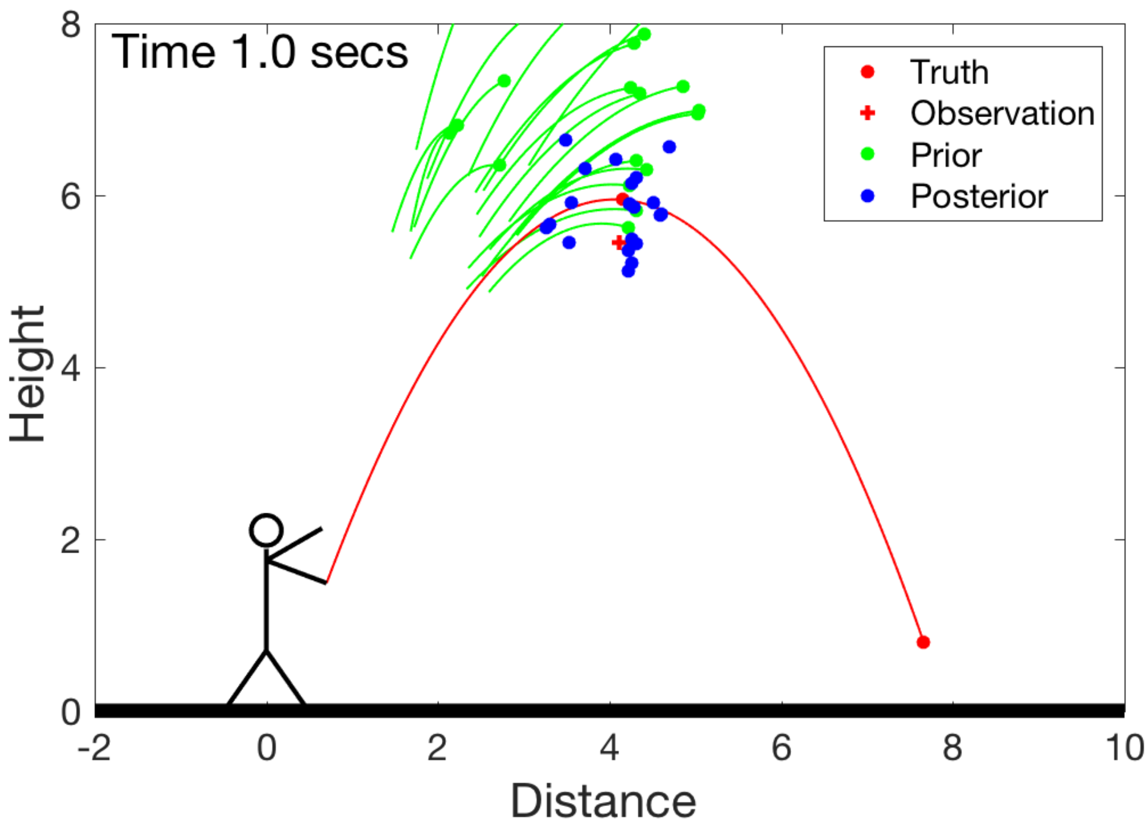
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



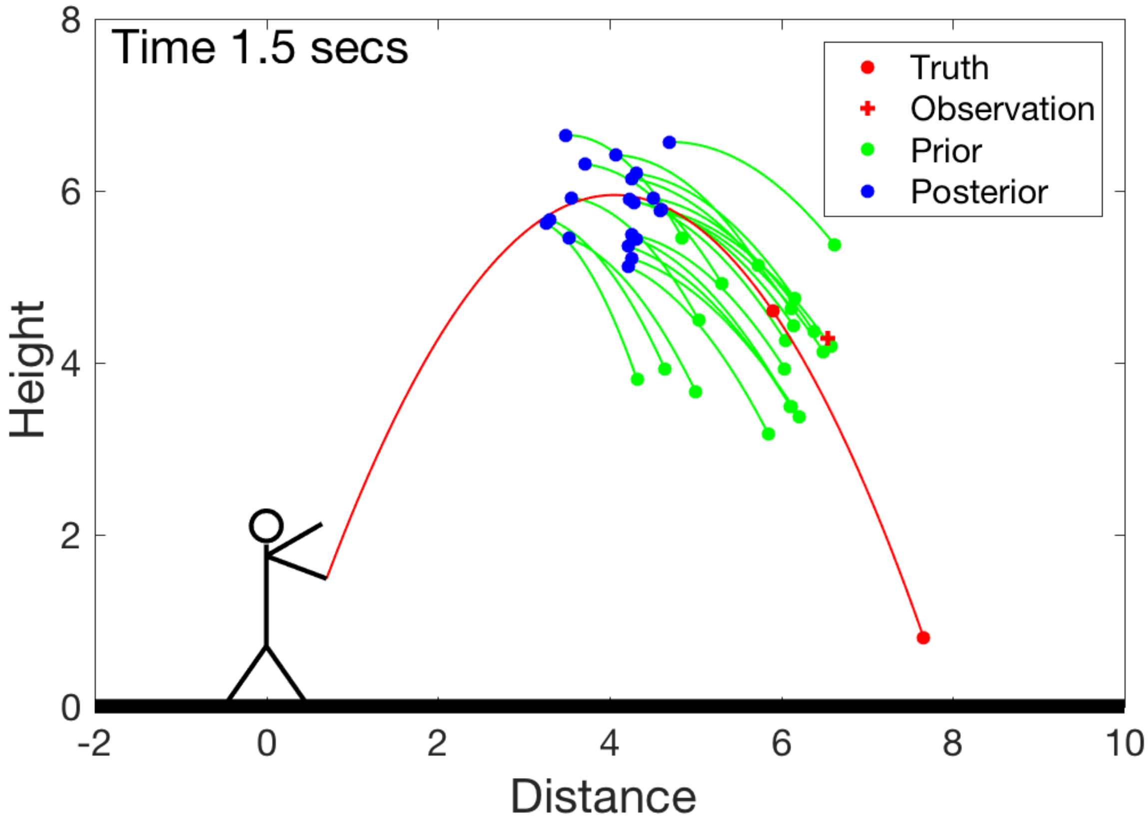
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



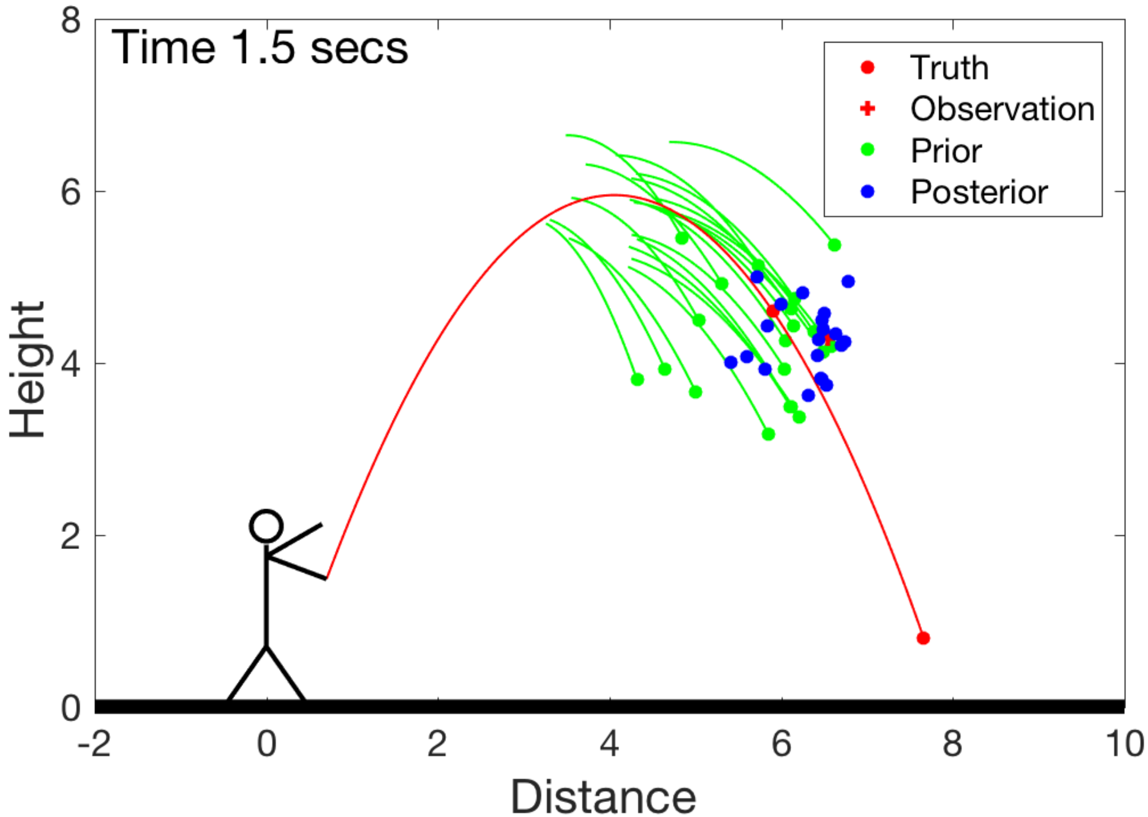
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



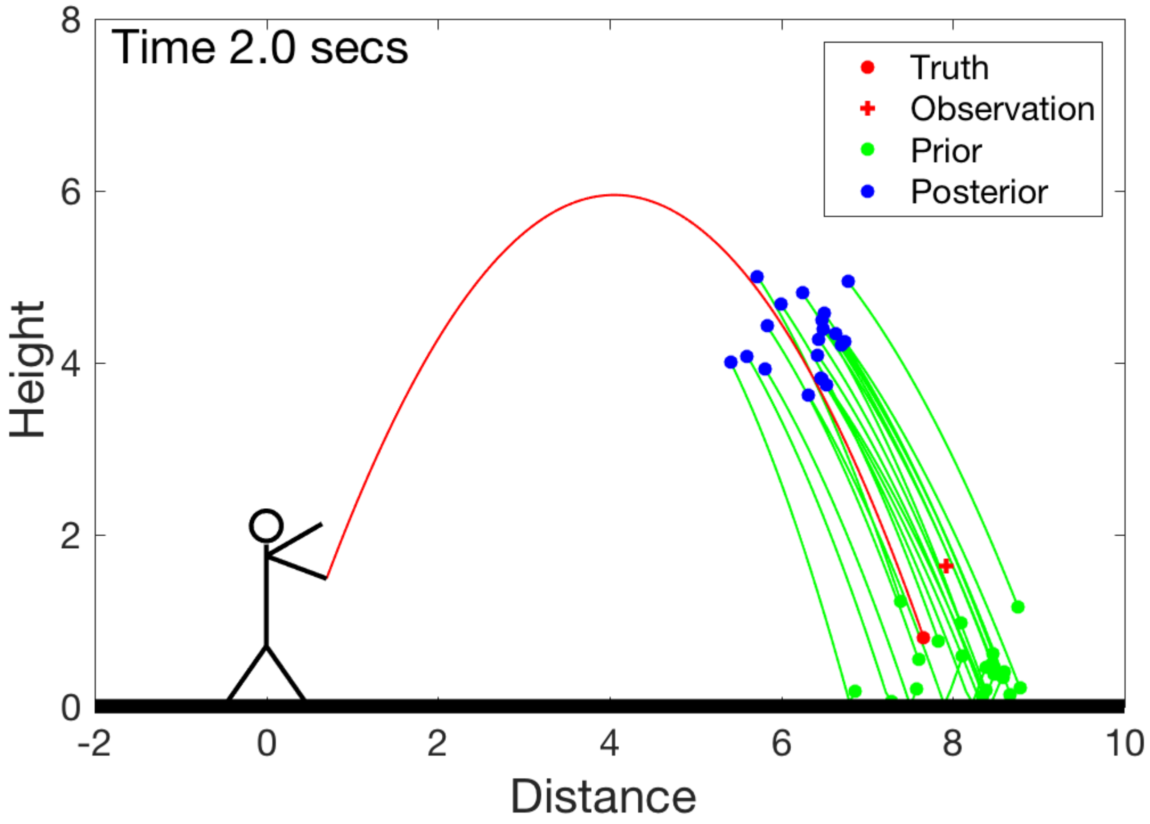
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



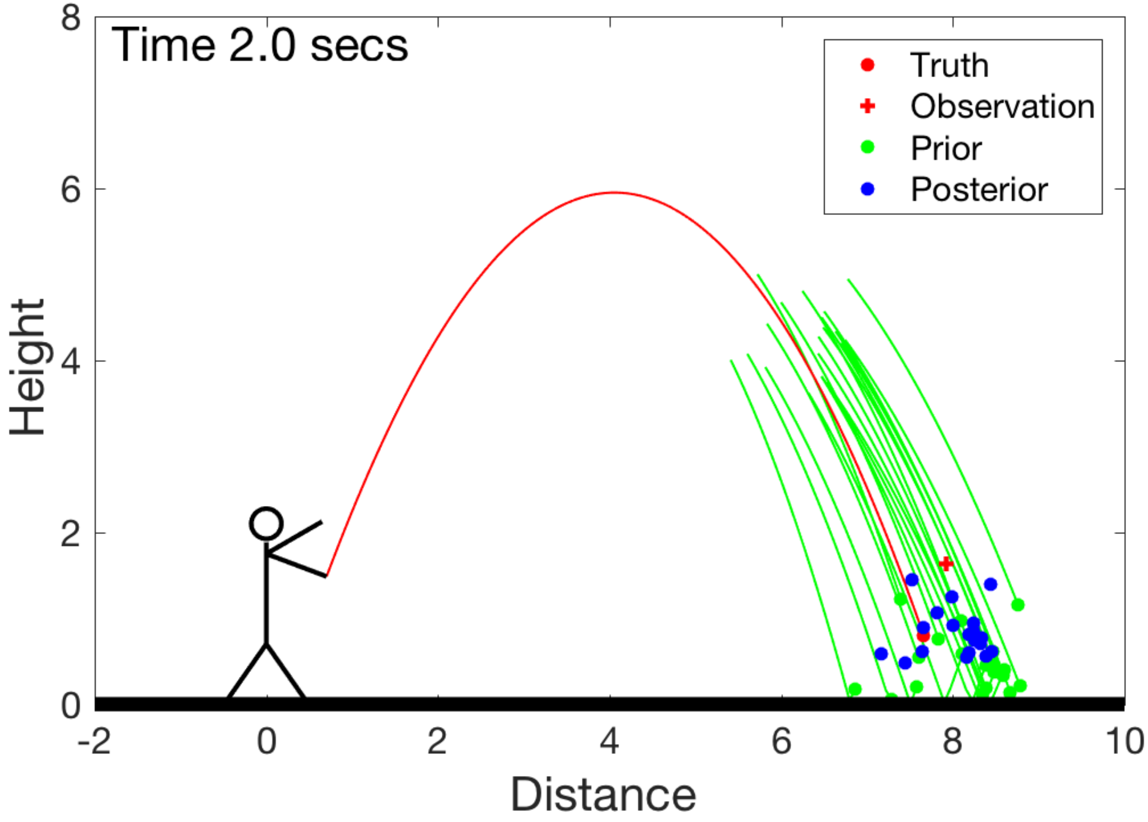
Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



Methods: Ensemble Kalman Filter

20-member ensemble example.
Starts with random draw from uncertain thrower.
Quickly converges towards truth.



Methods: Ensemble Kalman Filter

Can be implemented with no matrix inversions.
Very fast compared to Kalman filter.

Fails for large models with small ensembles.
    Can be fixed by ’localizing’ impact of observations.

Small ensembles have too little variance.
    Can be fixed by ‘inflating’ ensembles.

Methods: Ensemble Kalman Filter

Capabilities:

- Estimates some aspects of arbitrary probability distribution.
- Easy to apply to any model, observation operators.
- Works with huge models (see caveats below).

Challenges:

- Sampling error leads to covariance errors.
- Needs localization/inflation to work with
  small ensembles/large models.

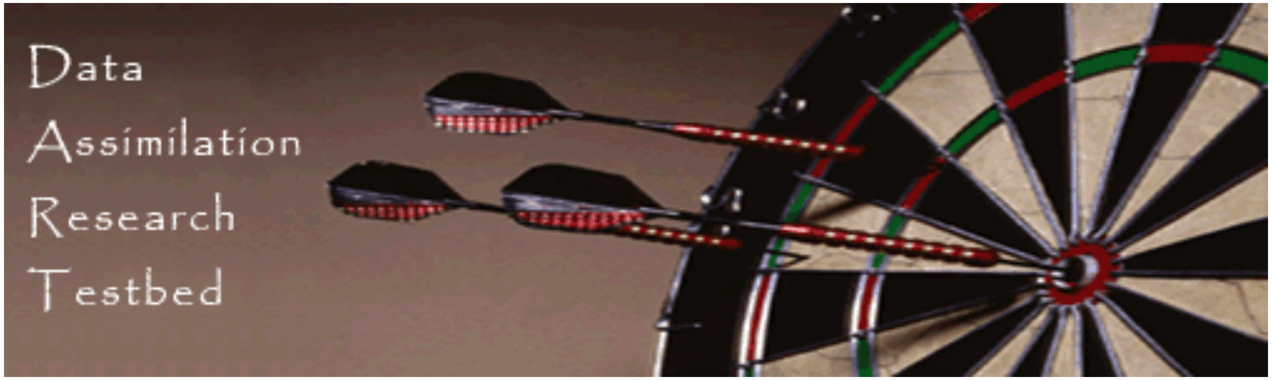
Prospects:

- Hybrids with variational may have advantages.
- Hybrids with particle filters may be better for non-gaussian.

Outline

1. Data Assimilation: Building a simple forecast system.
2. Data Assimilation: What can it do?
3. Data Assimilaton: A general description.
4. Methods: Particle filter.
5. Methods: Variational.
6. Methods: Kalman filter.
7. Methods: Ensemble Kalman filter.
8. Additional topics.

Learn more about DART at:



www.image.ucar.edu/DAReS/DART

dart@ucar.edu

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., Arellano, A., 2009: *The Data Assimilation Research Testbed: A community facility.* BAMS, **90**, 1283—1296, doi: 10.1175/2009BAMS2618.1