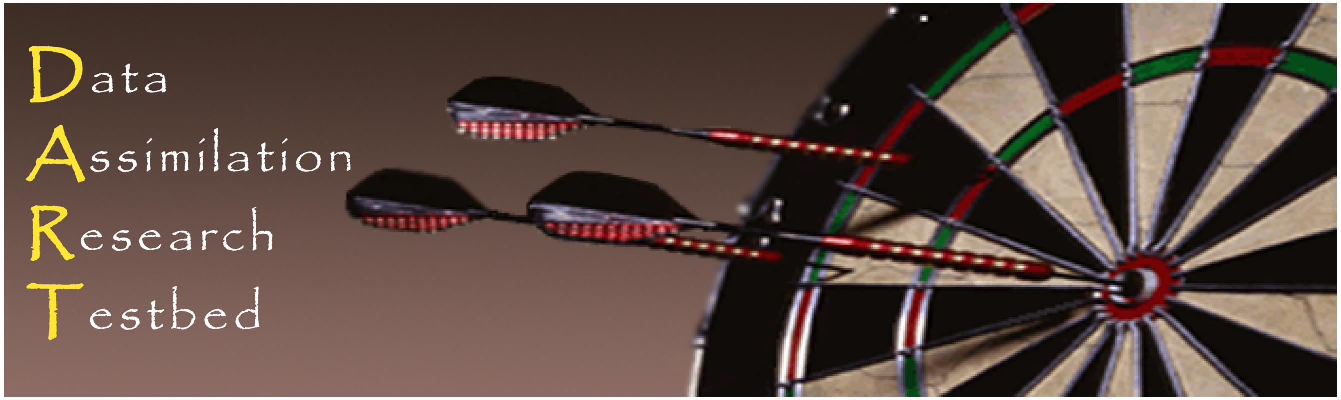
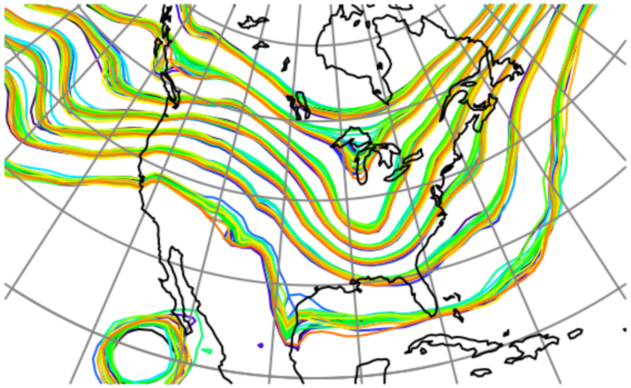


DART_LAB Tutorial Section 2: How should observations impact an unobserved state variable? Multivariate assimilation.

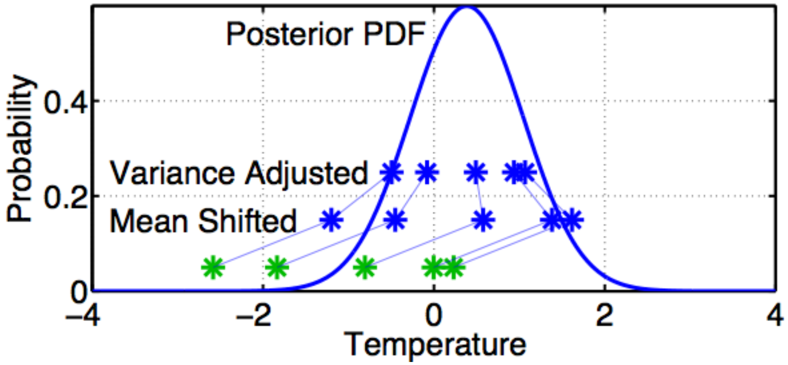


The National Center for Atmospheric Research is sponsored by the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

NCAR UCAR | National Center for Atmospheric Research

Single observed variable, single unobserved variable.

So far, we have a known likelihood for a single variable.

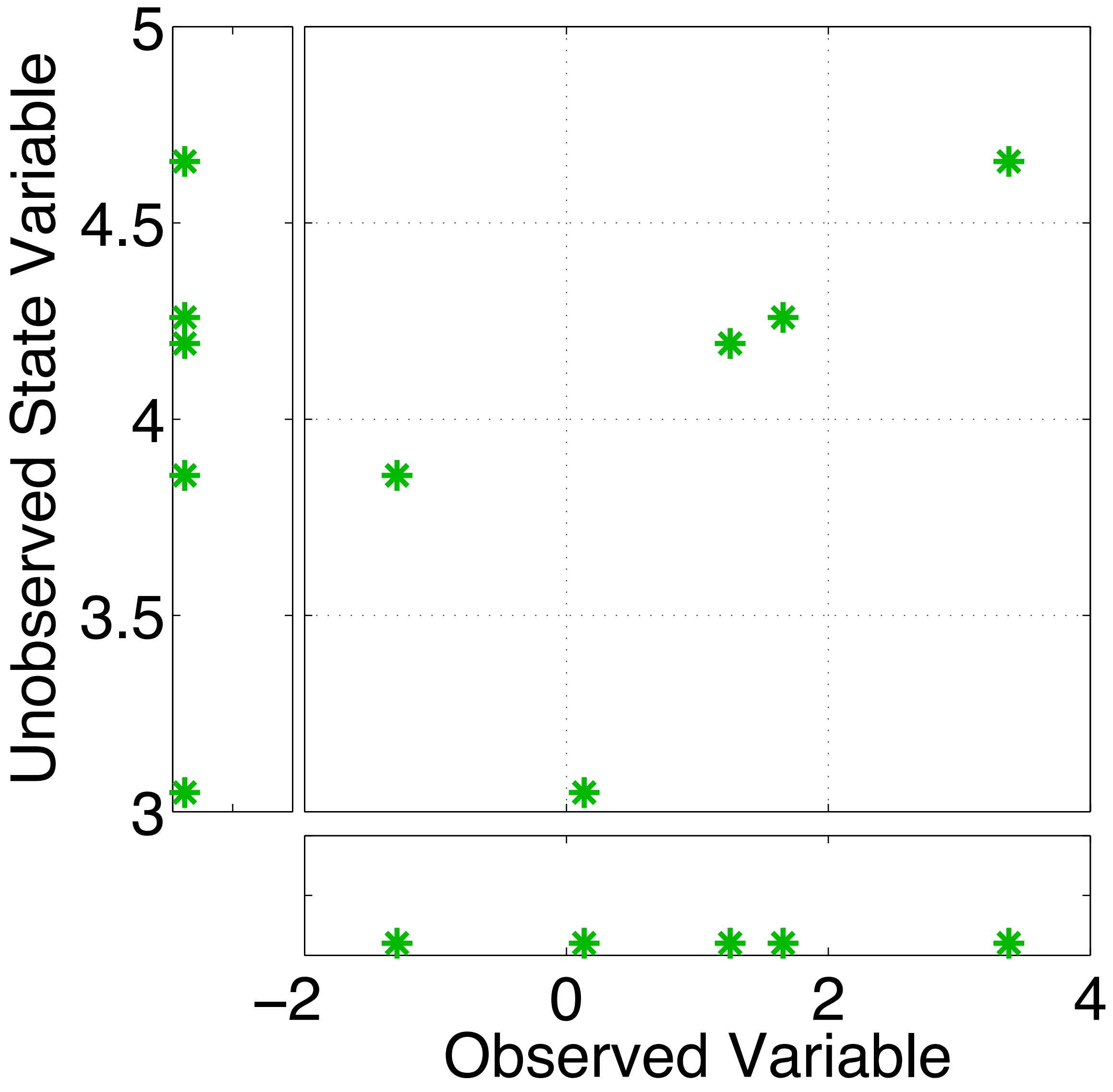


Now, suppose the prior has an additional variable …

We will examine how ensemble members update the additional variable.

Basic method generalizes to any number of additional variables.

Ensemble filters: Updating additional prior state variables

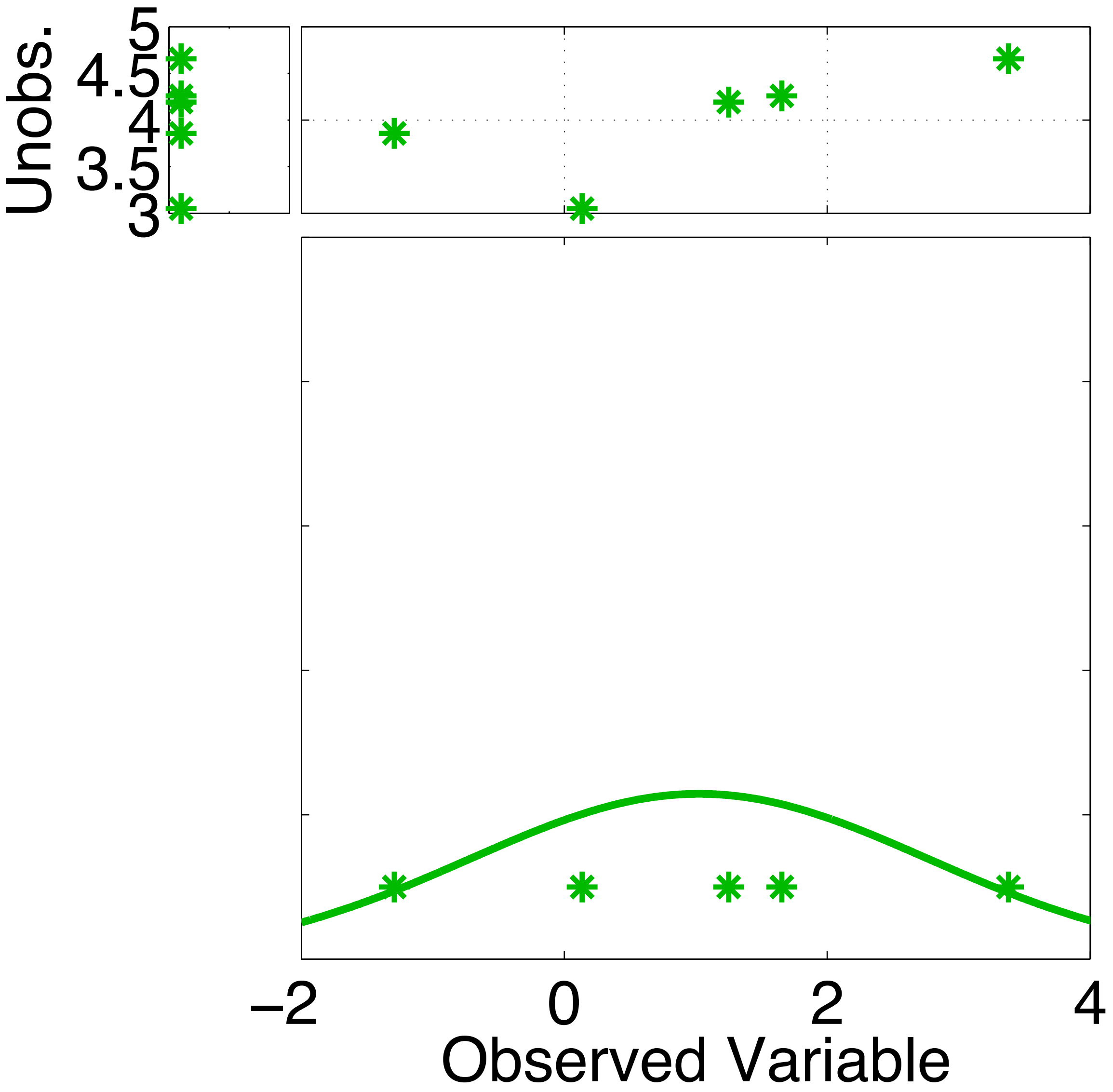


Assume that all we know is the prior joint distribution.

One variable is observed.

What should happen to the unobserved variable?

Ensemble filters: Updating additional prior state variables

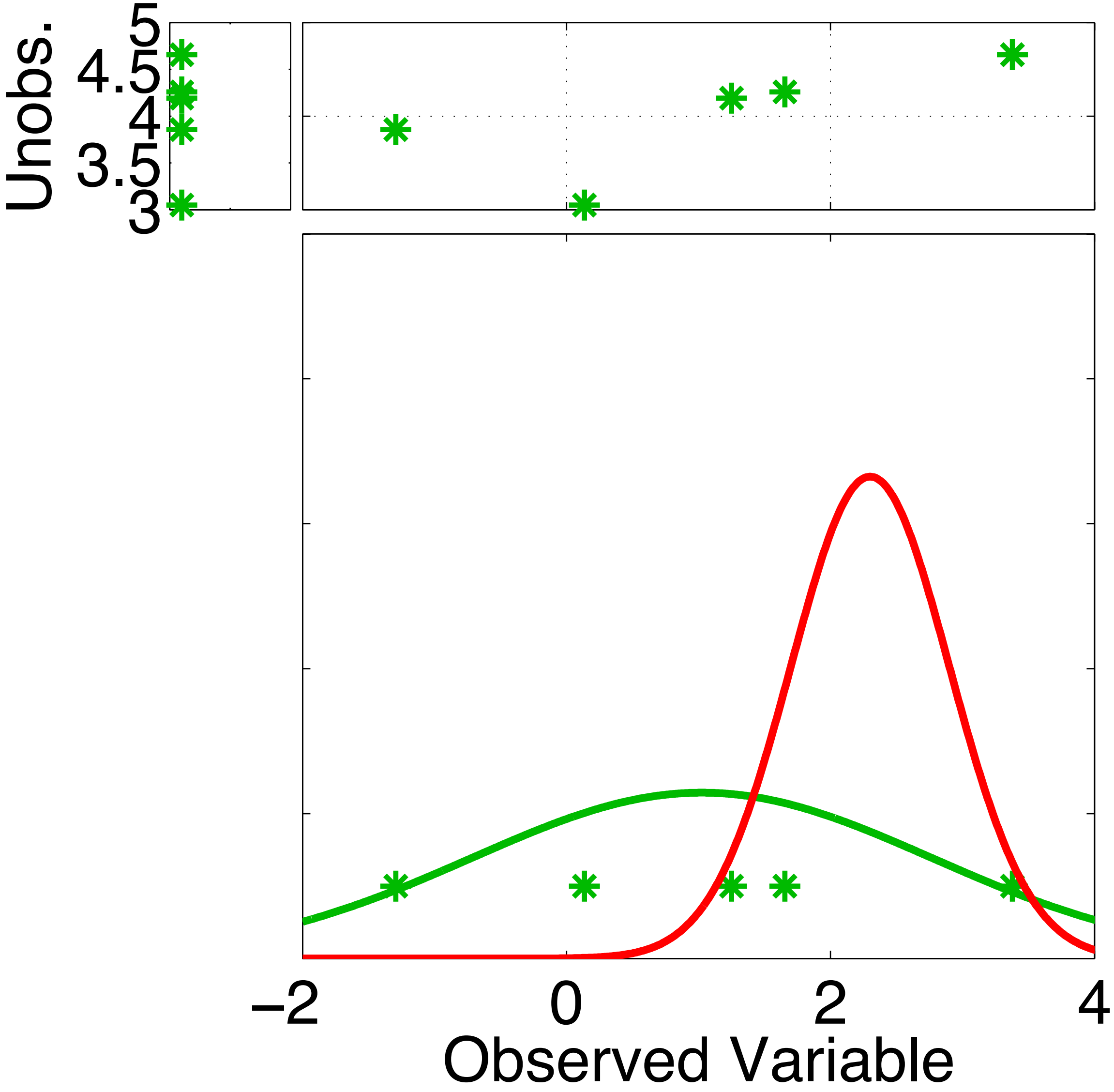


Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with one of the previous methods.

Ensemble filters: Updating additional prior state variables

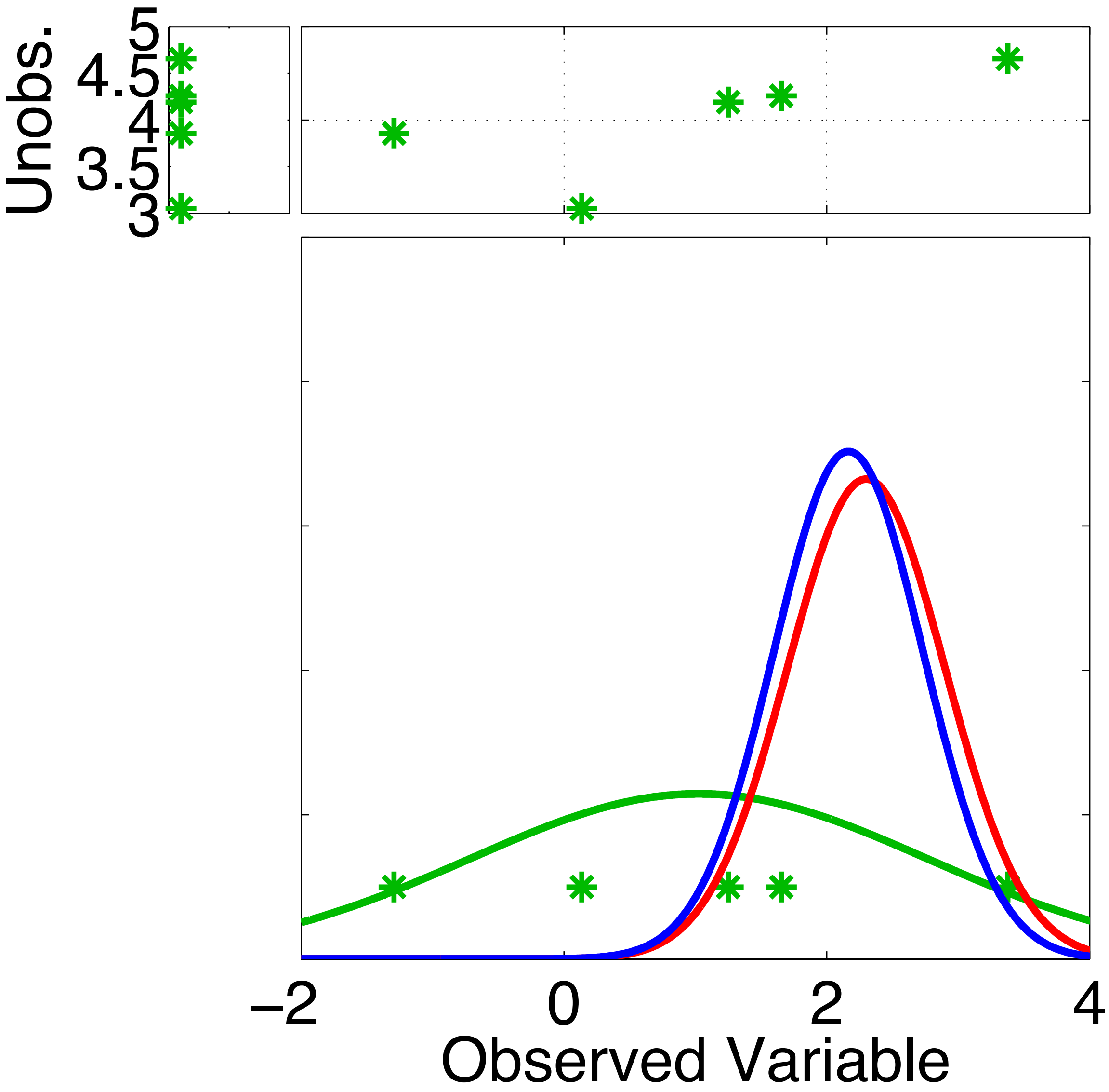


Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with one of the previous methods.

Ensemble filters: Updating additional prior state variables

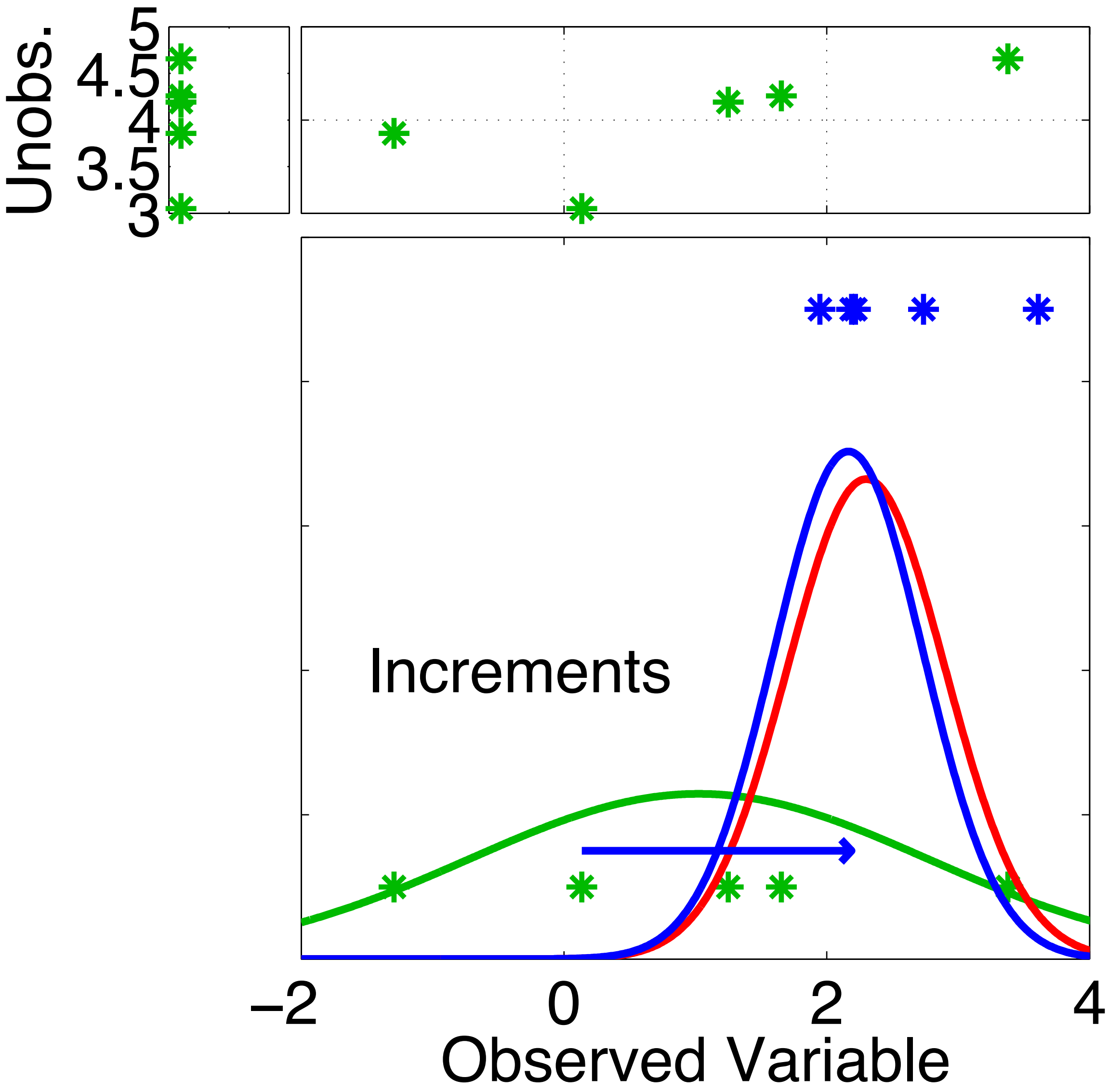


Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with one of the previous methods.

Ensemble filters: Updating additional prior state variables

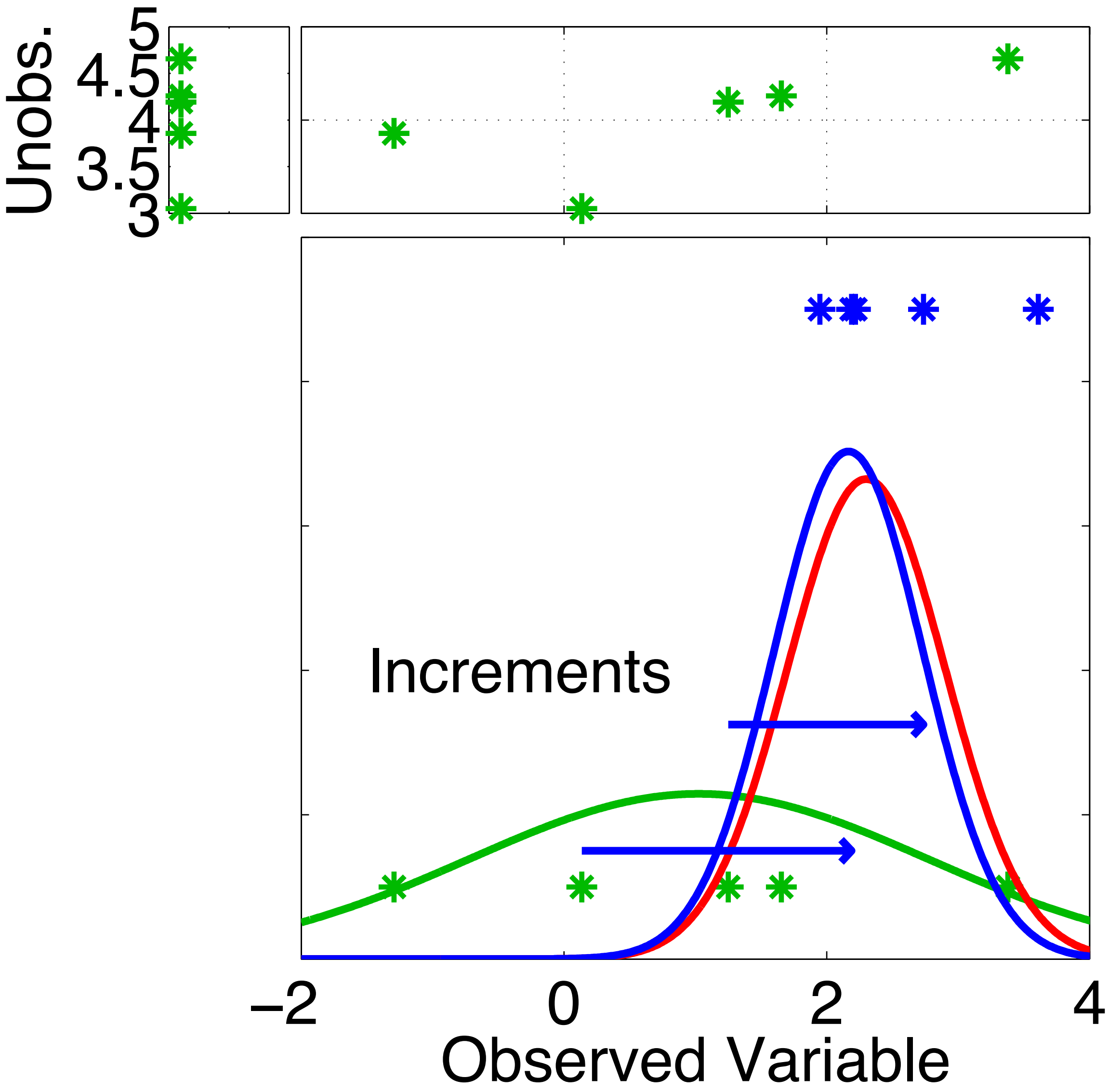


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

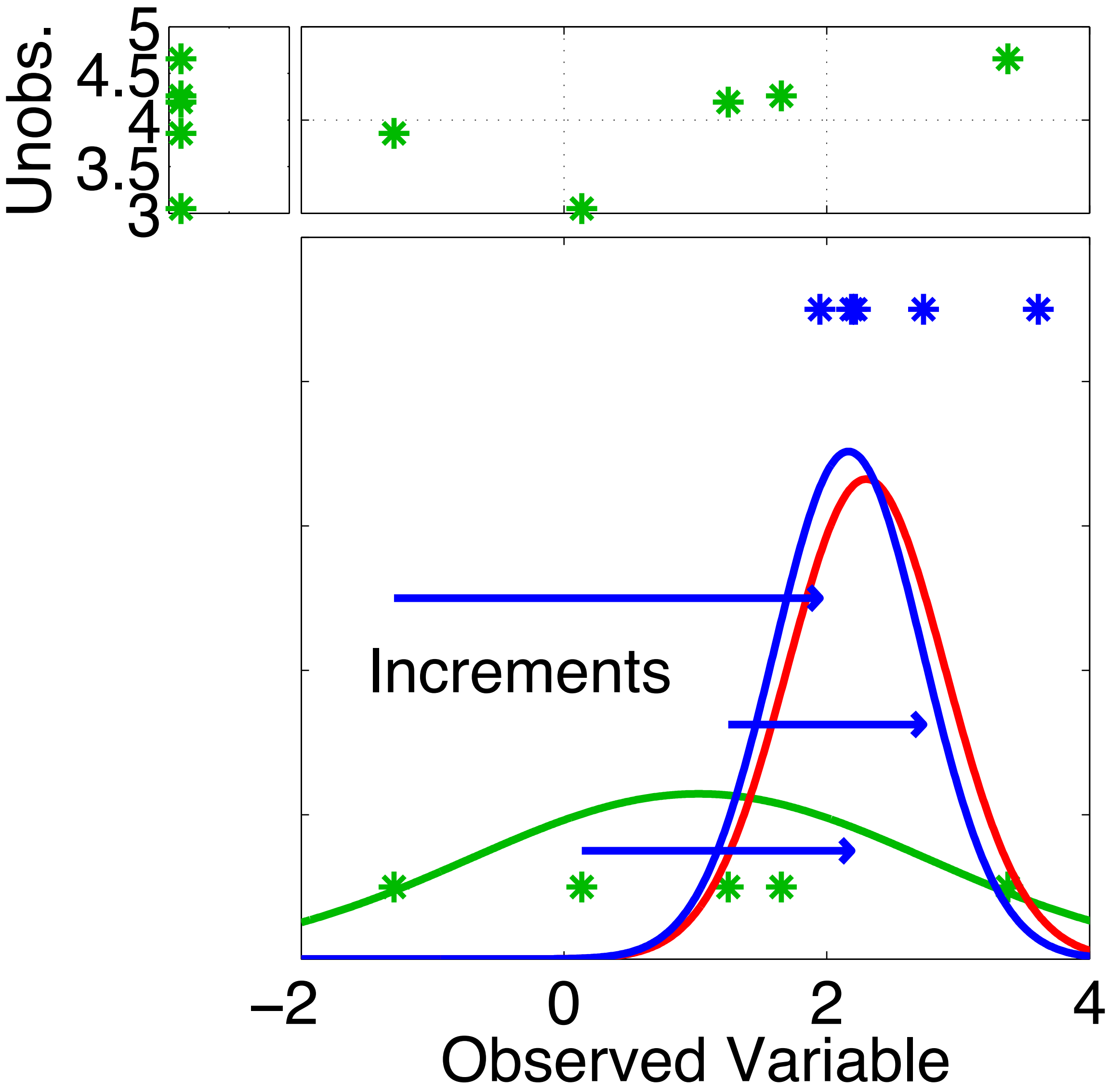


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

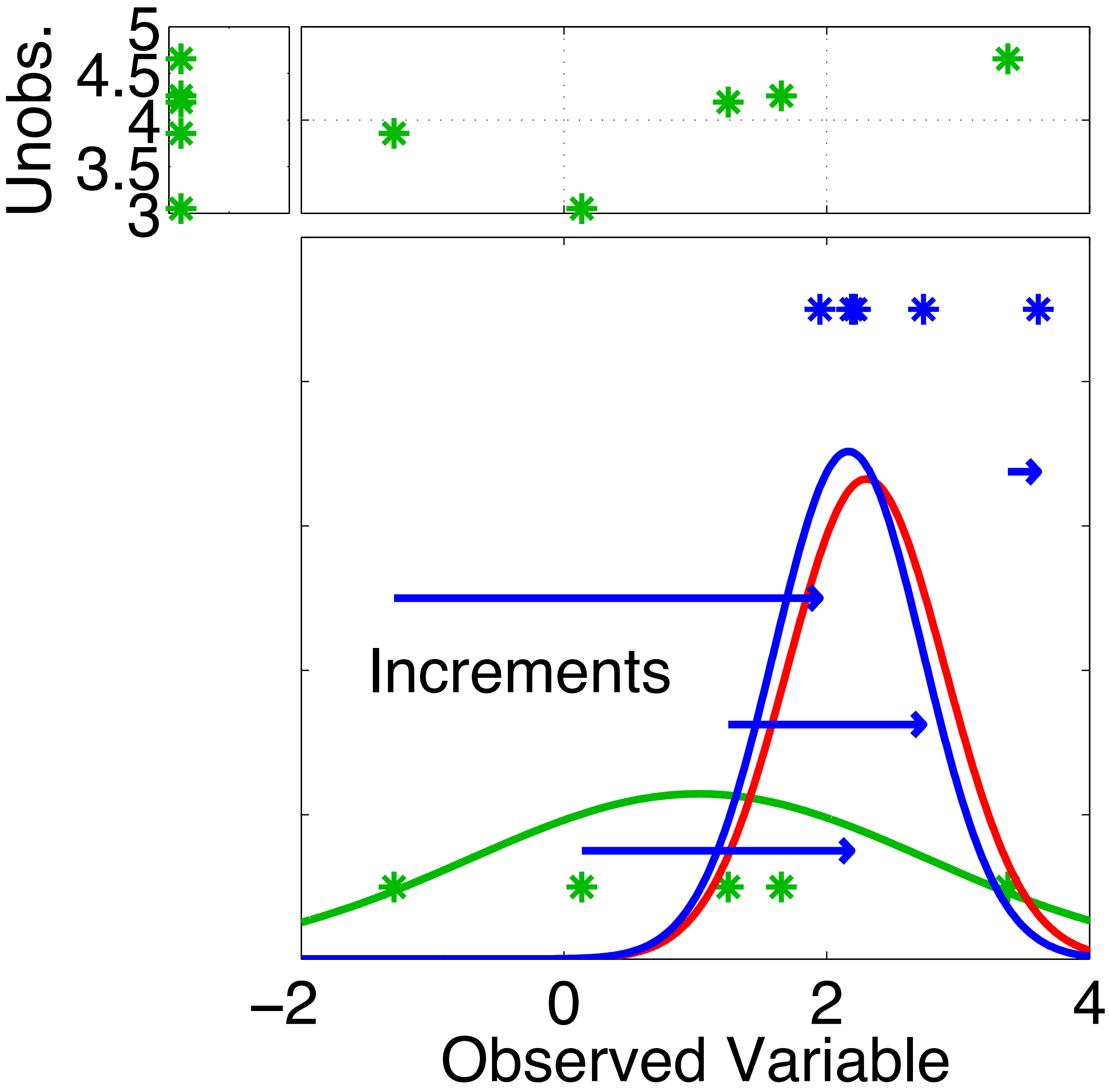


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

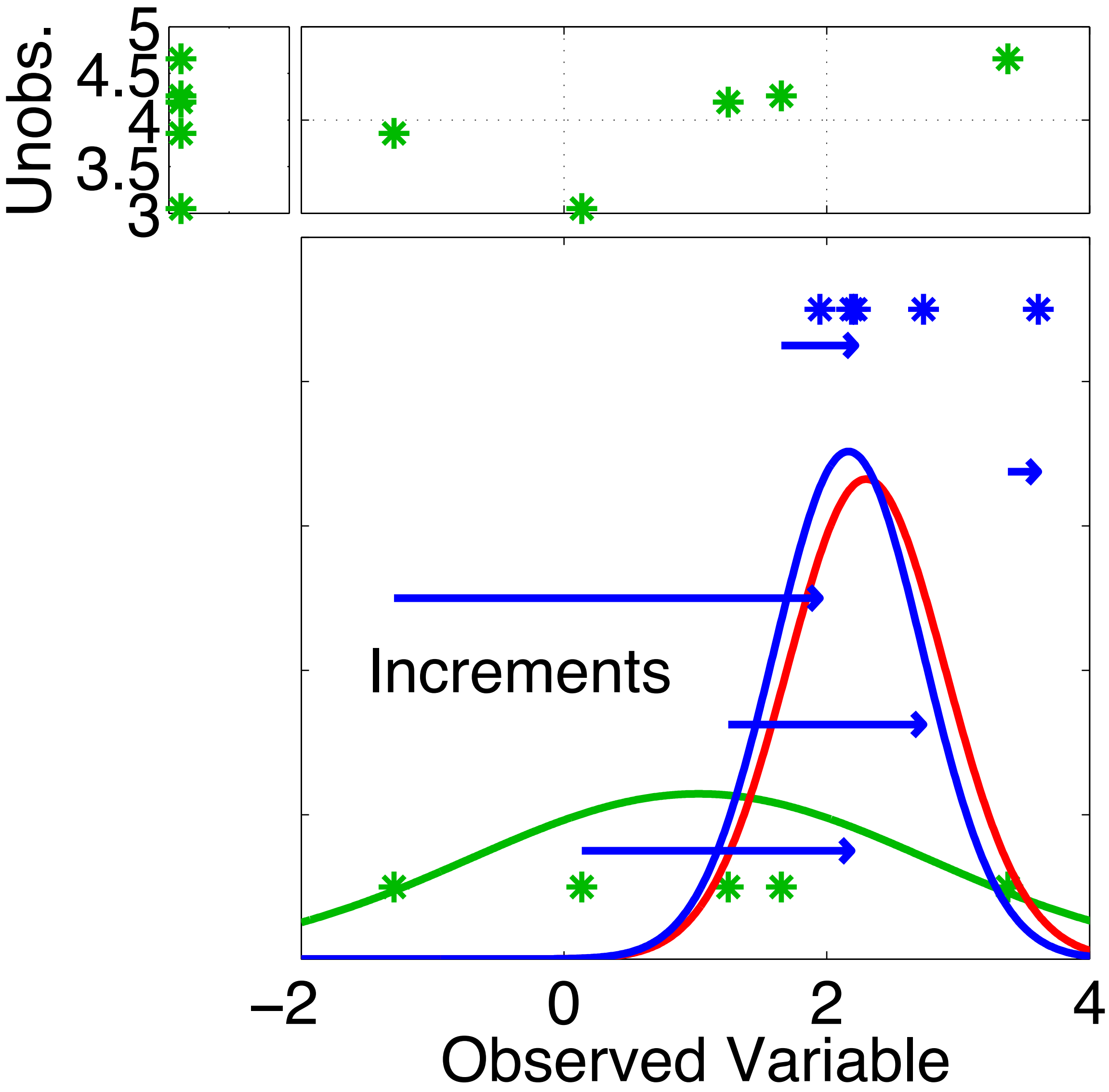


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

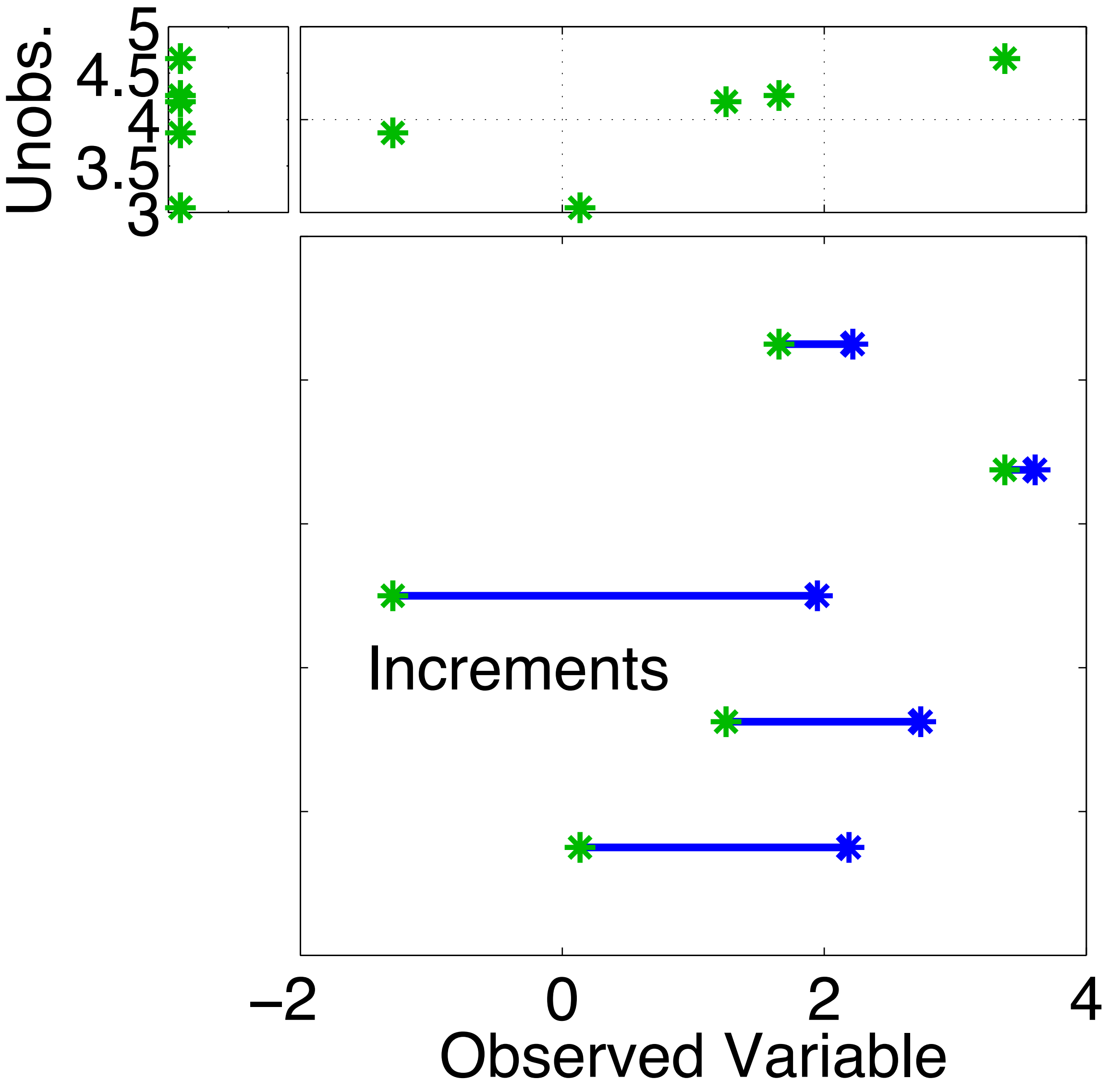


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

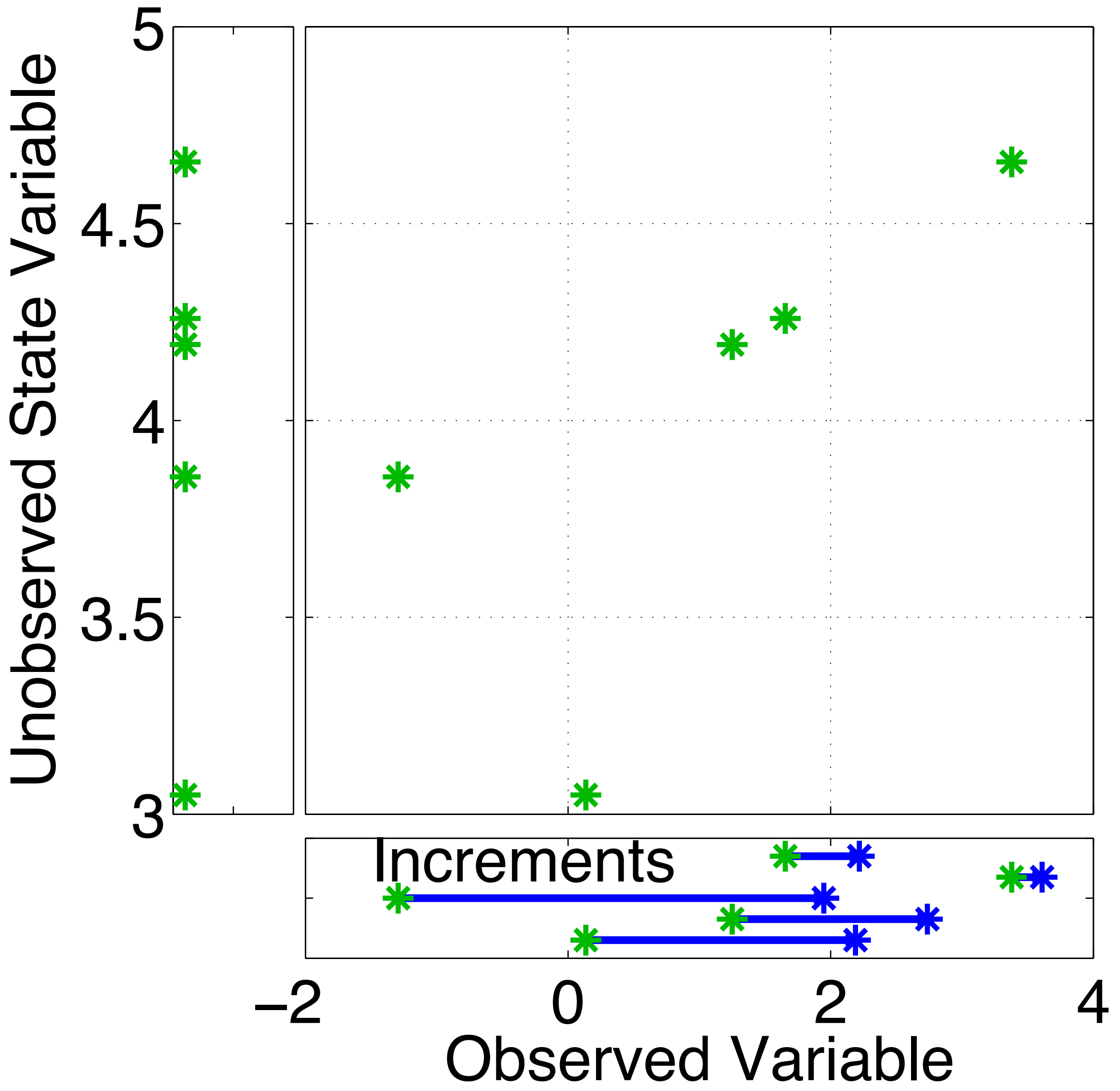
Ensemble filters: Updating additional prior state variables



Using only increments guarantees that if observation had no impact on observed variable, the unobserved variable is unchanged.

Highly desirable!

Ensemble filters: Updating additional prior state variables



Assume that all we know is the prior joint distribution.

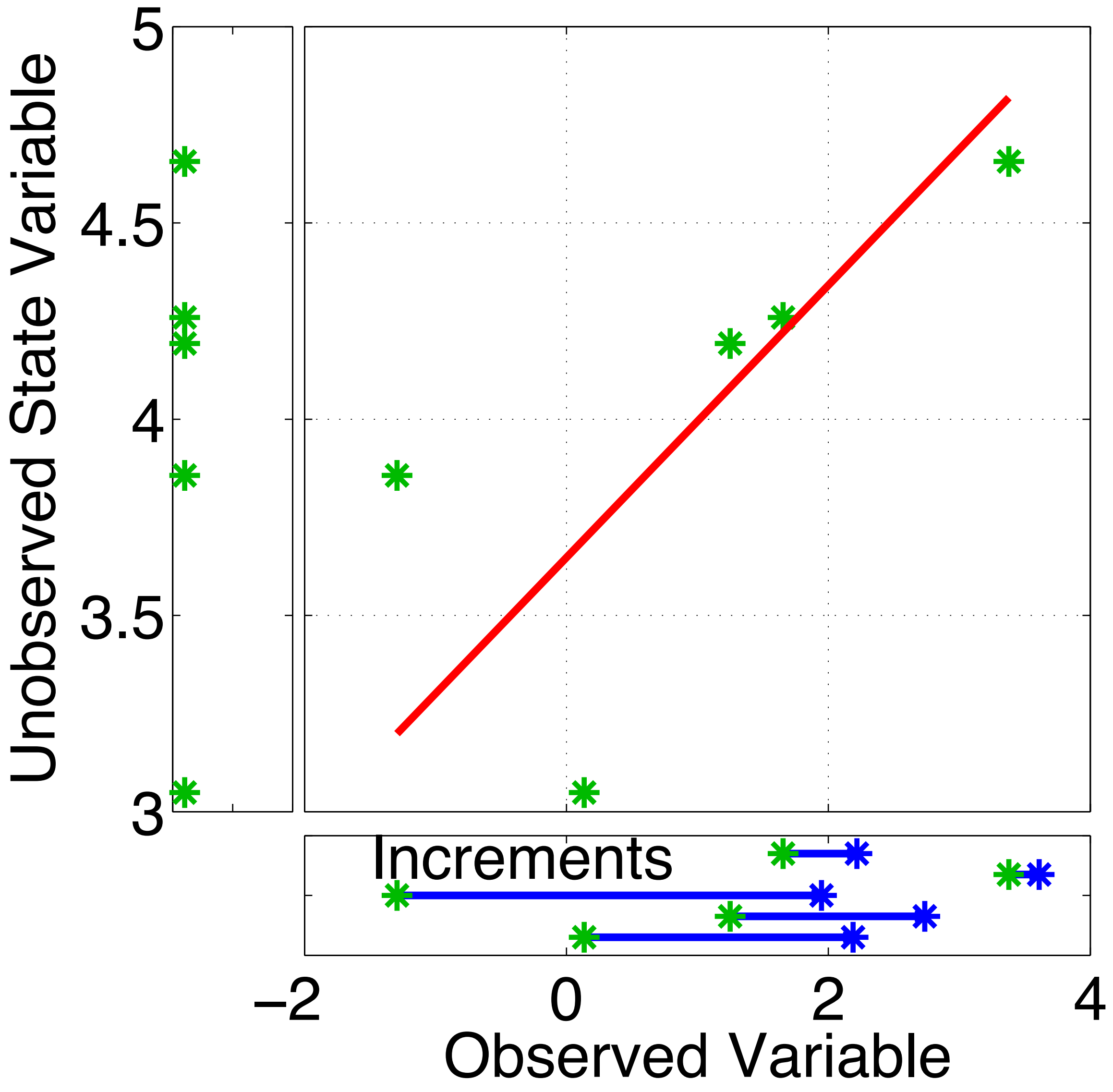
How should the unobserved variable be impacted?

1st choice: least squares

Equivalent to linear regression.

Same as assuming binormal prior.

Ensemble filters: Updating additional prior state variables



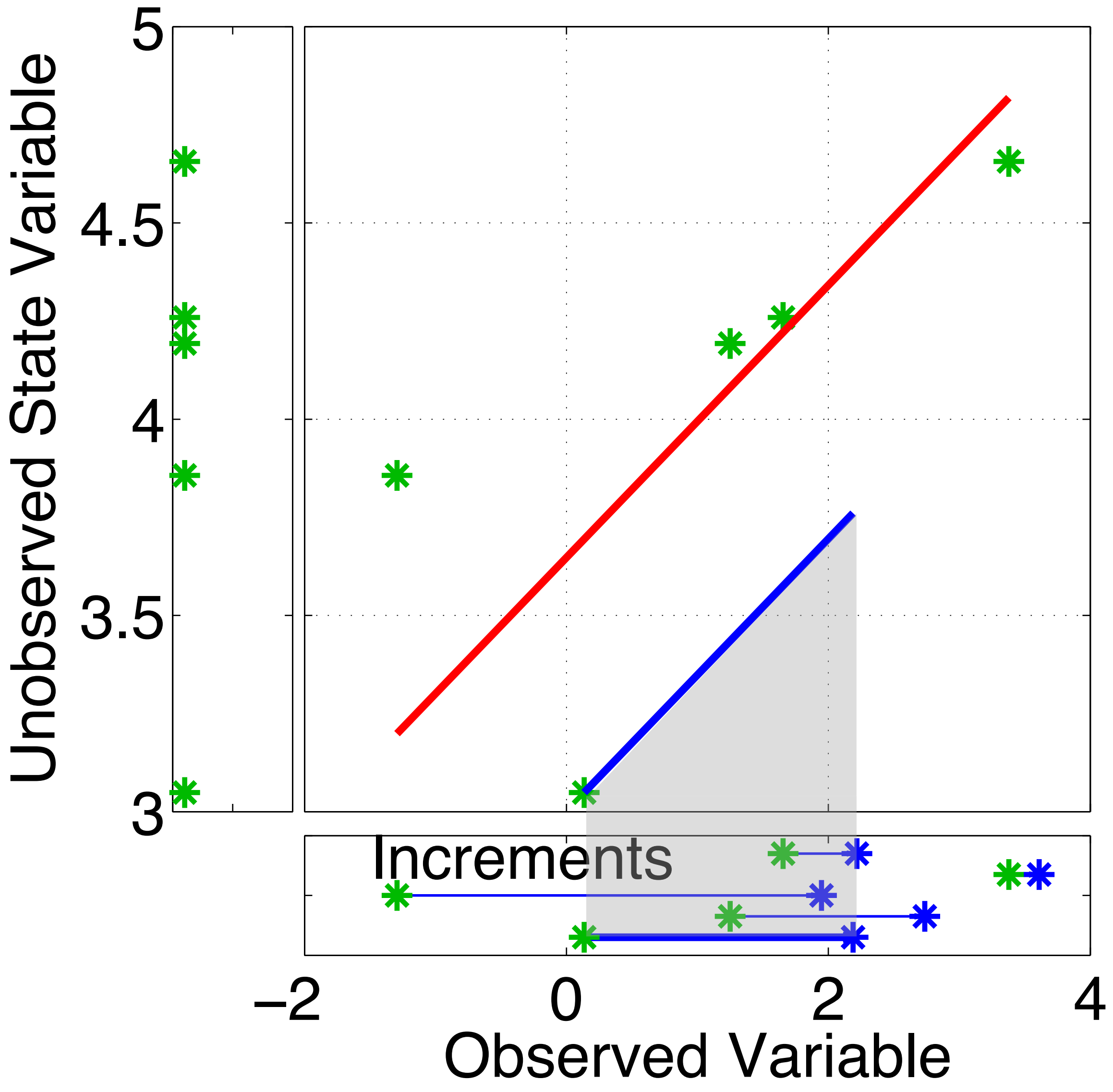
Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

1st choice: least squares

Begin by finding least squares fit.

Ensemble filters: Updating additional prior state variables

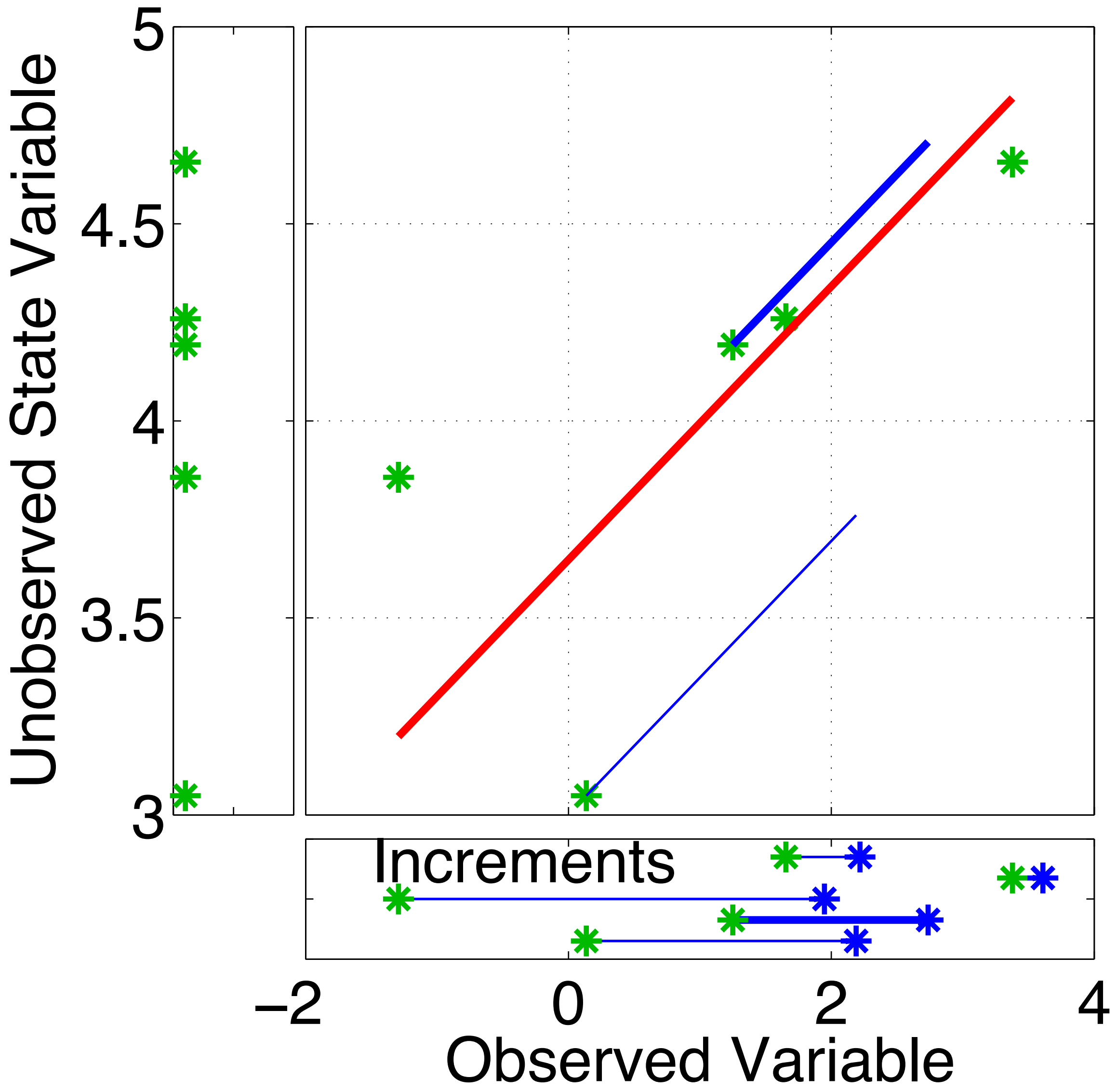


Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.

Ensemble filters: Updating additional prior state variables

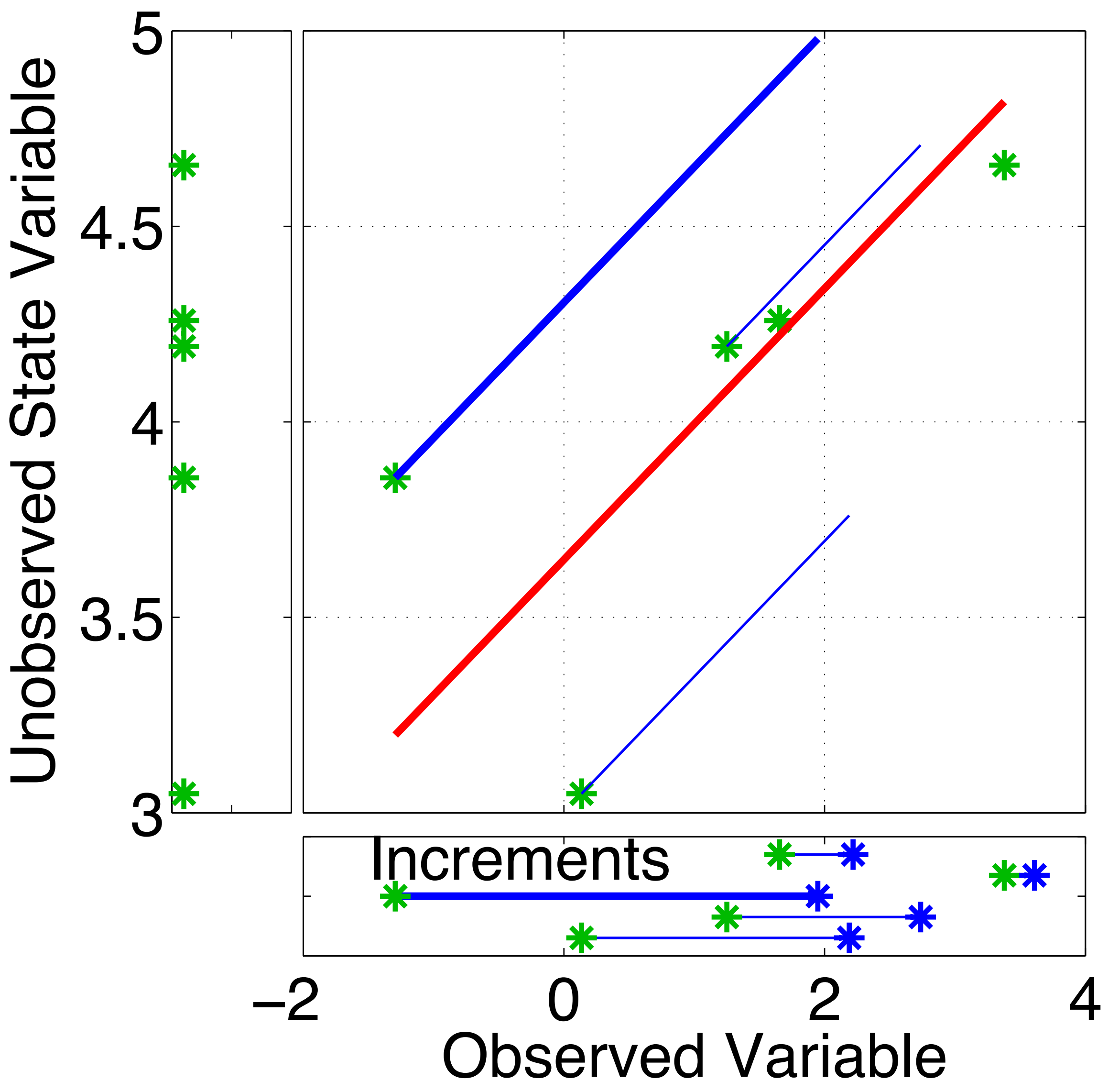


Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.

Ensemble filters: Updating additional prior state variables

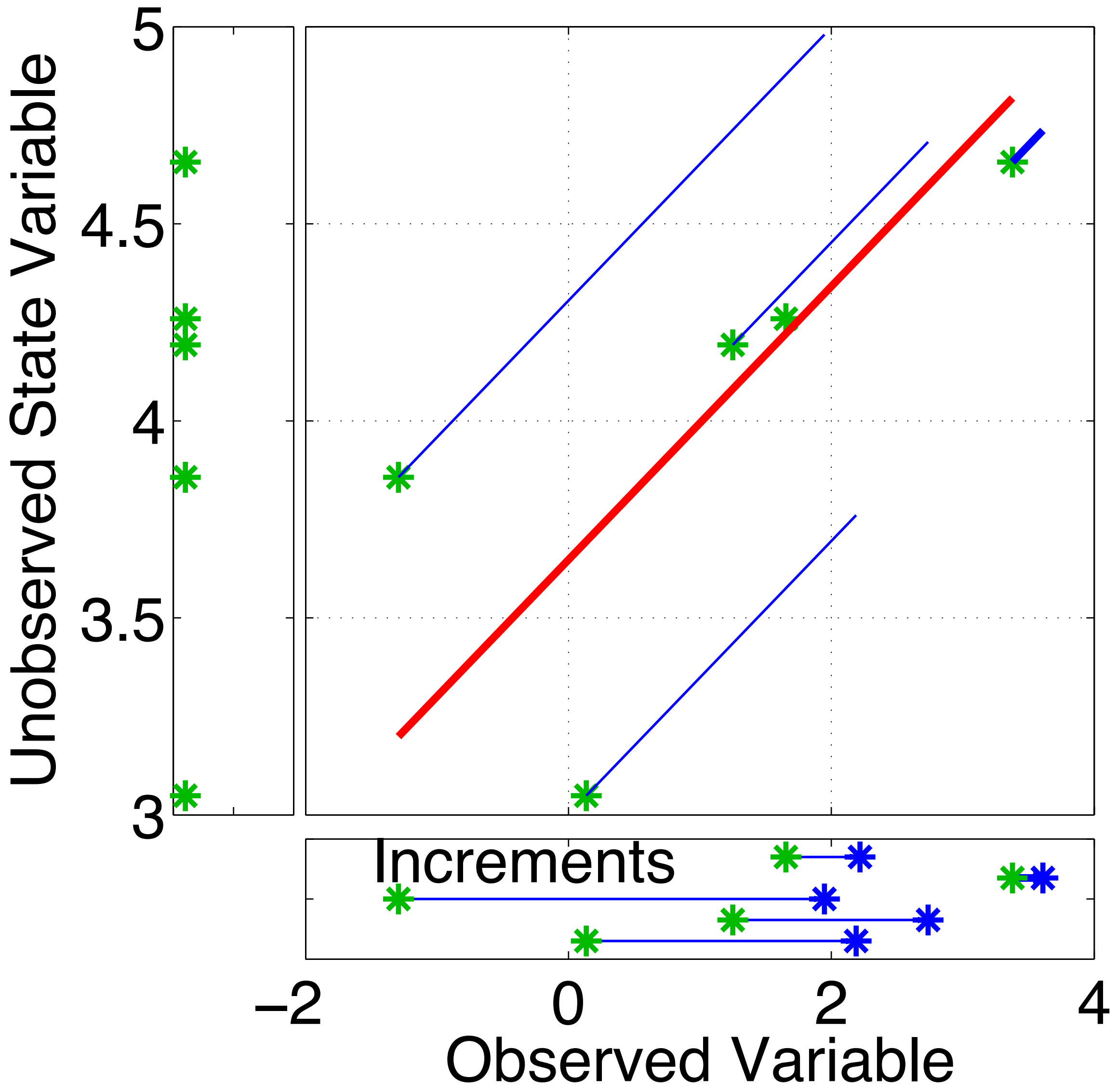


Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.

Ensemble filters: Updating additional prior state variables

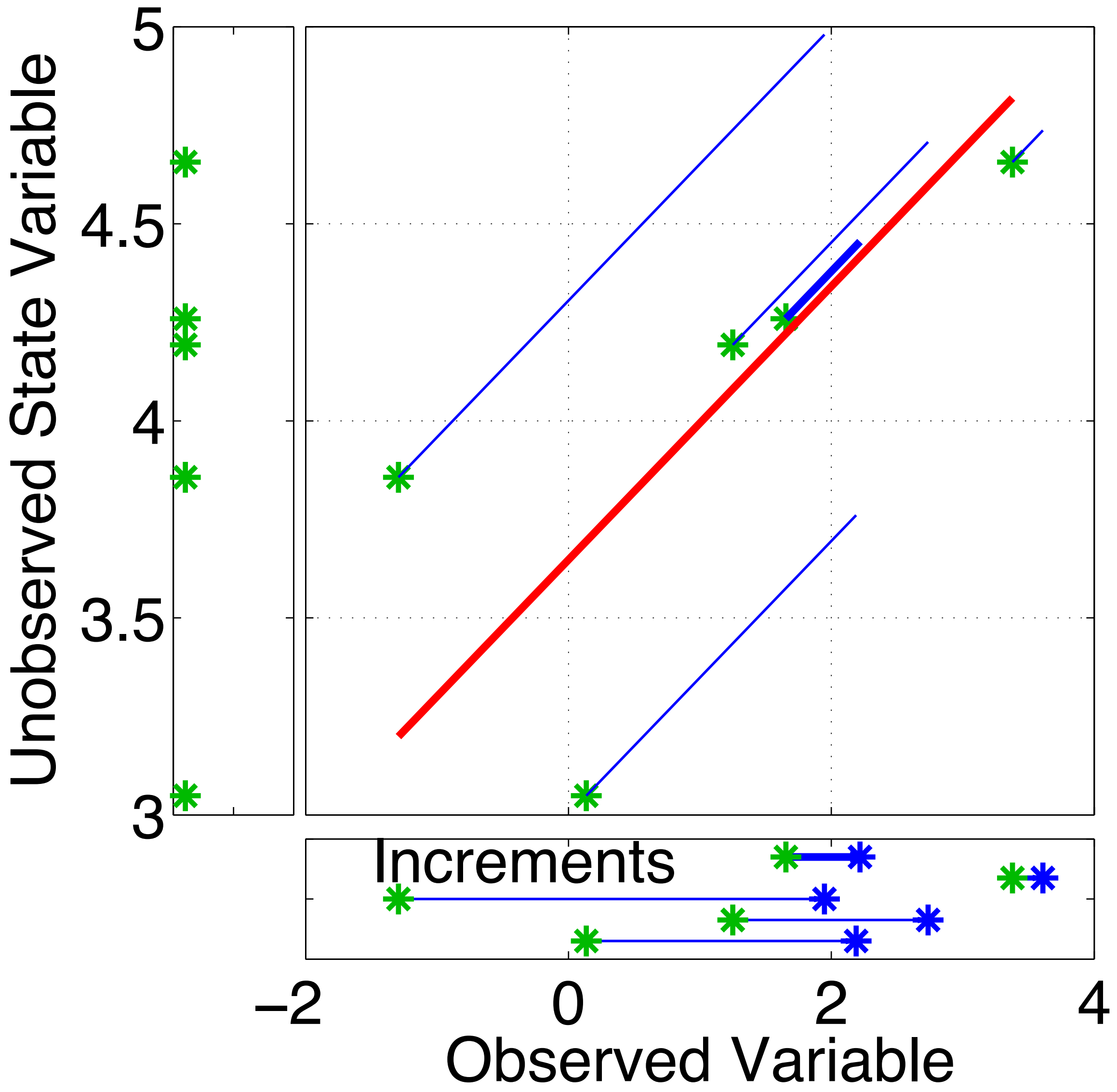


Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.

Ensemble filters: Updating additional prior state variables

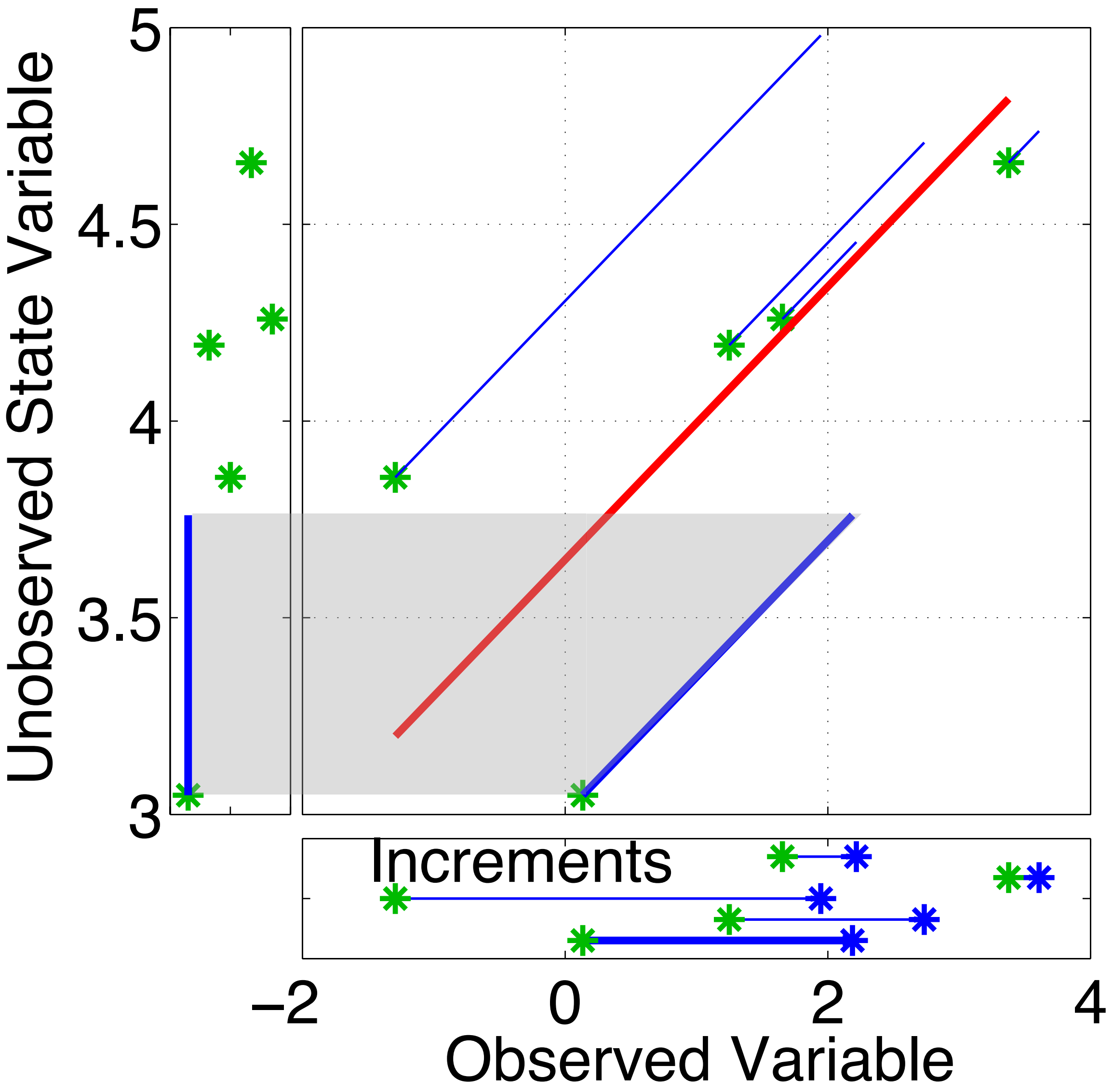


Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.

Ensemble filters: Updating additional prior state variables

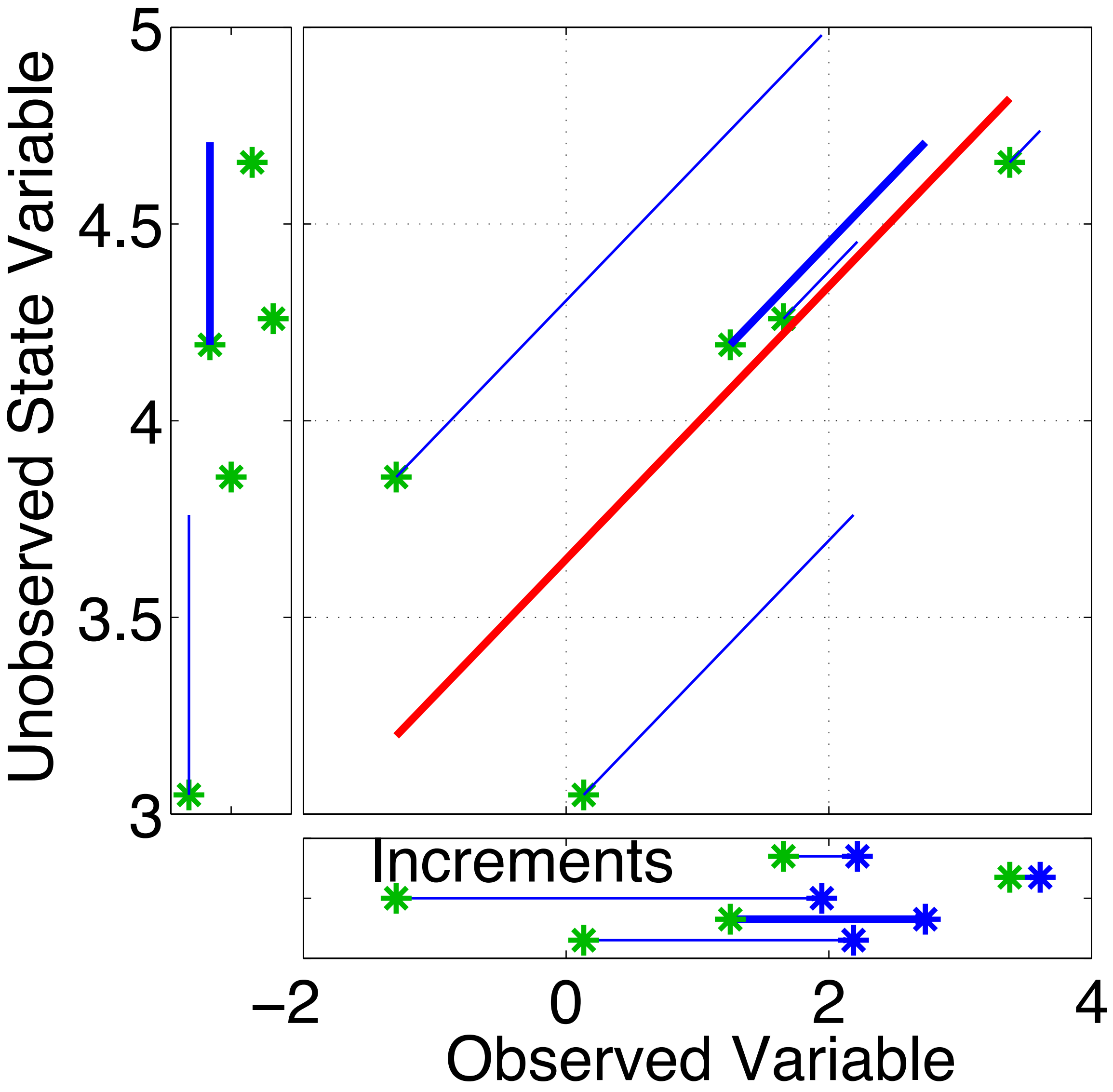


Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.

Ensemble filters: Updating additional prior state variables

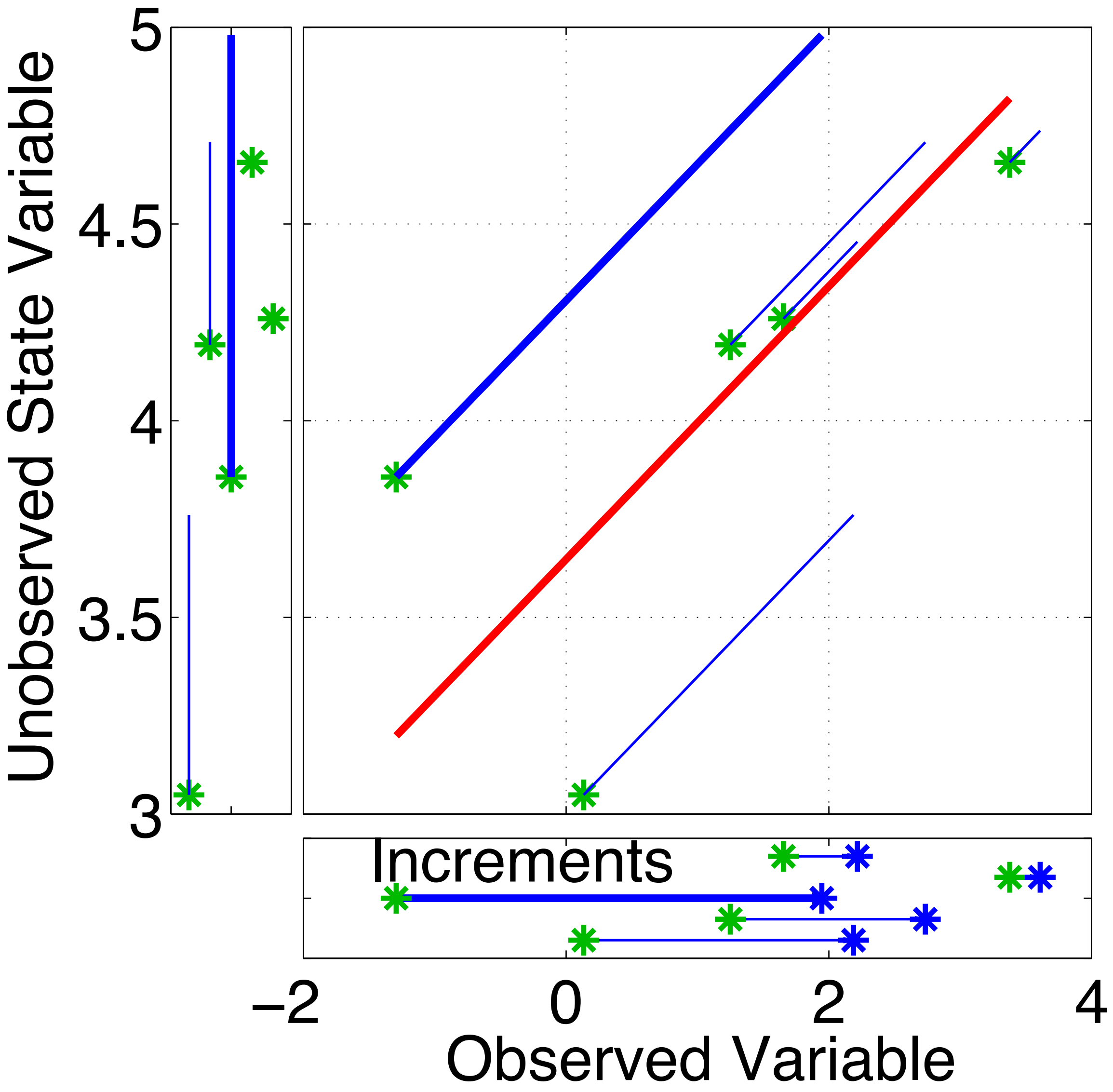


Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.

Ensemble filters: Updating additional prior state variables

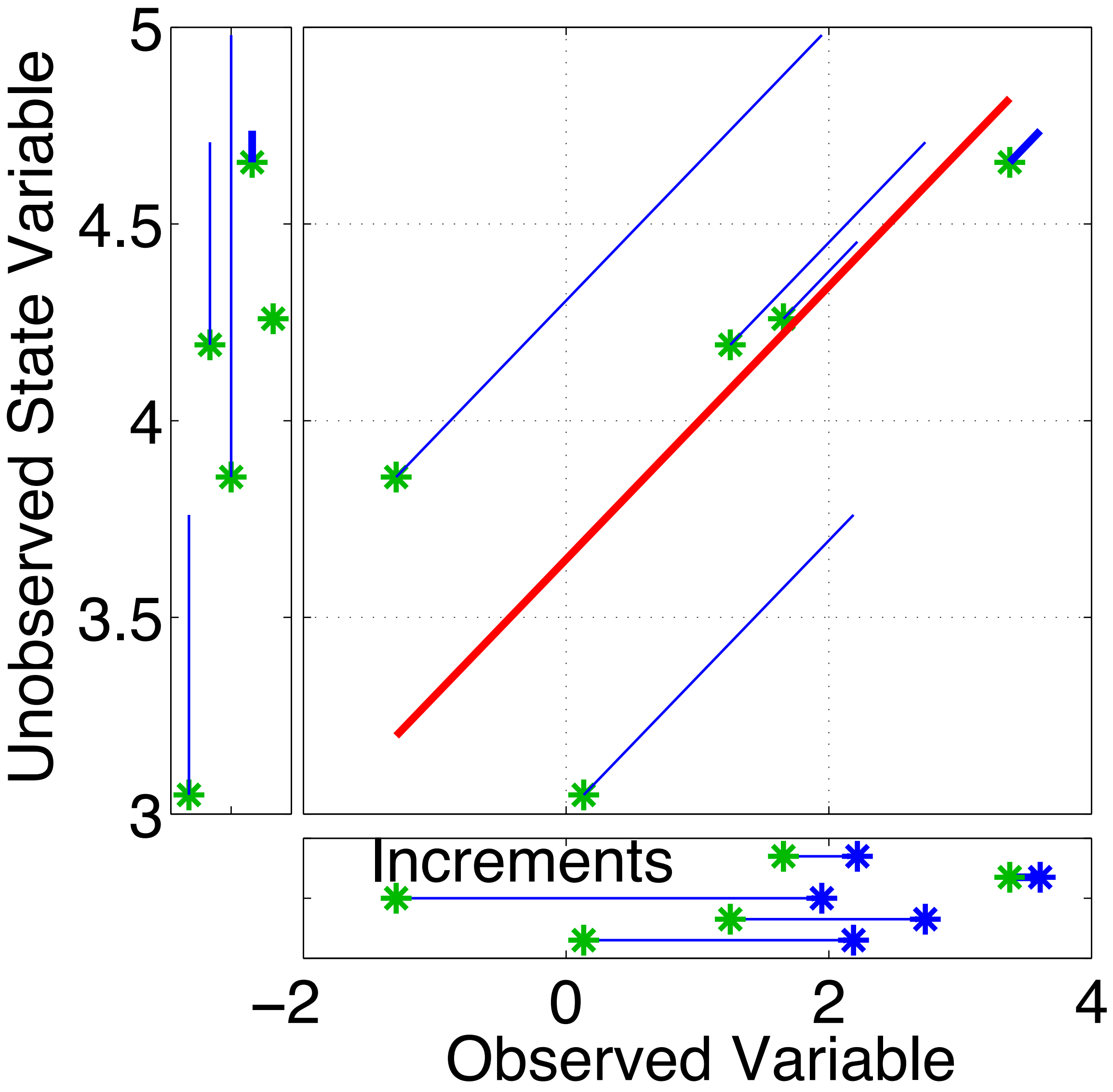


Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.

Ensemble filters: Updating additional prior state variables

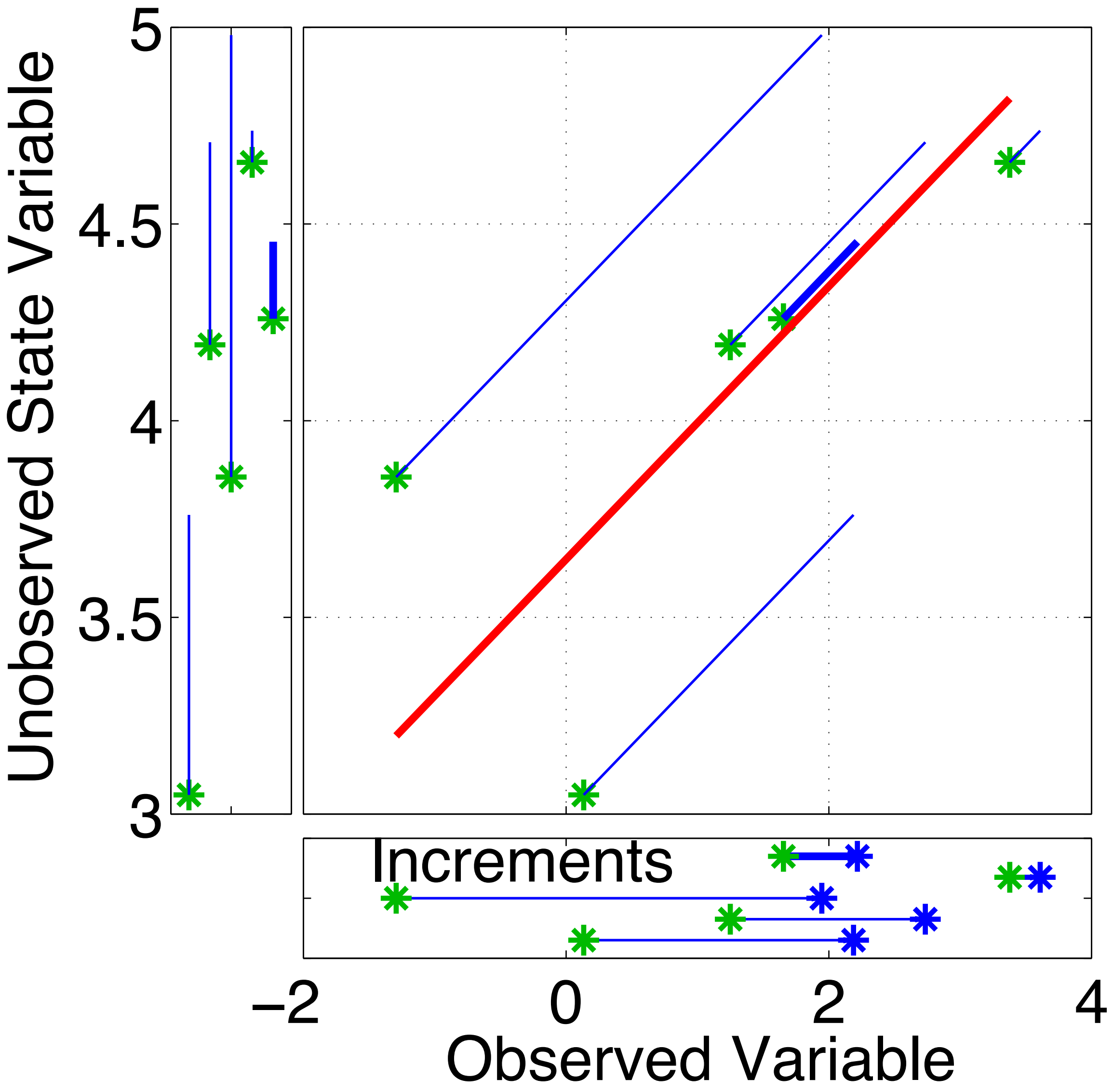


Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.

Ensemble filters: Updating additional prior state variables

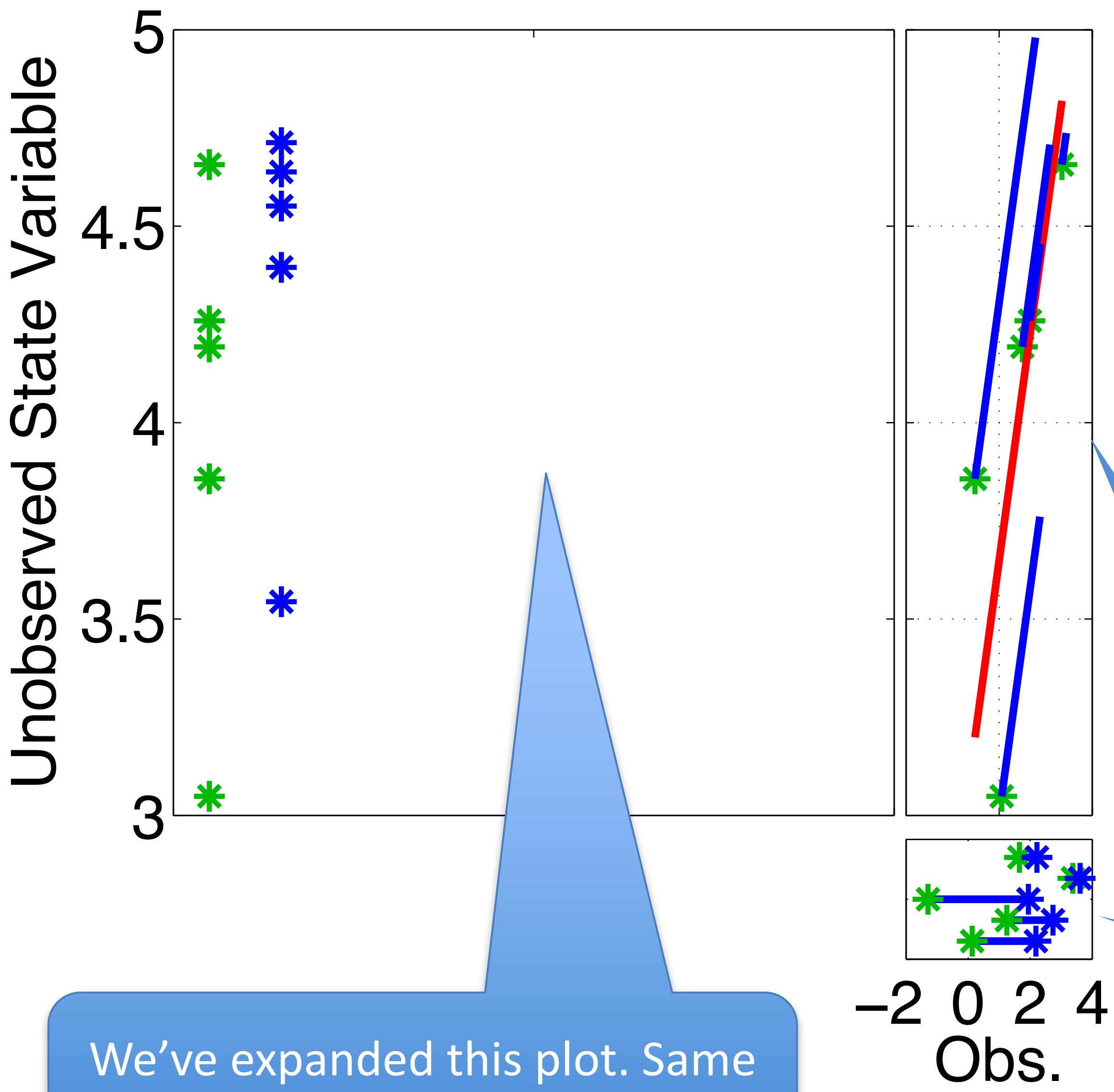


Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.

Ensemble filters: Updating additional prior state variables

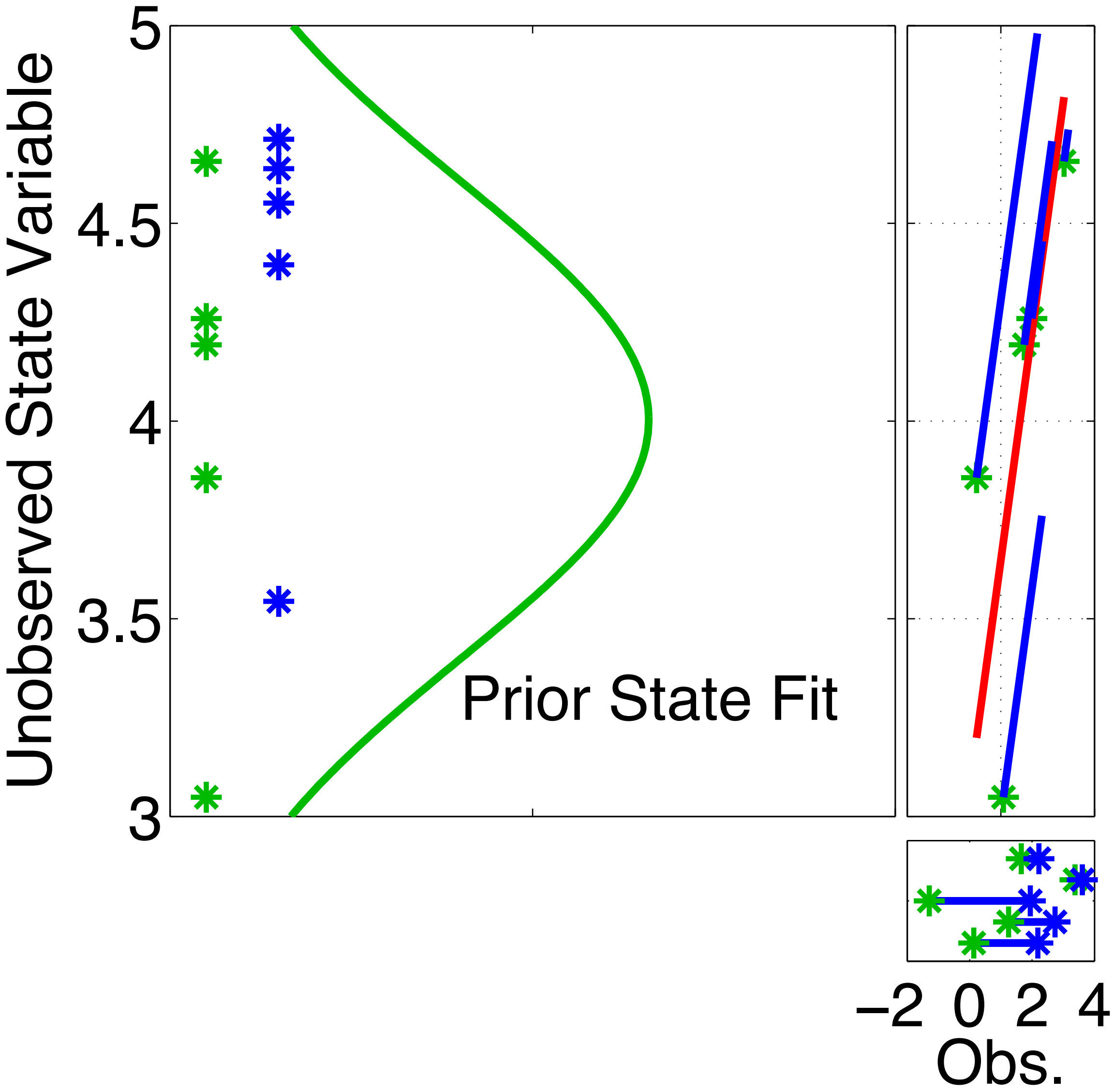


Now have an updated (posterior) ensemble for the unobserved variable.

We’ve expanded this plot. Same information as previous slides.

Compressed these two.

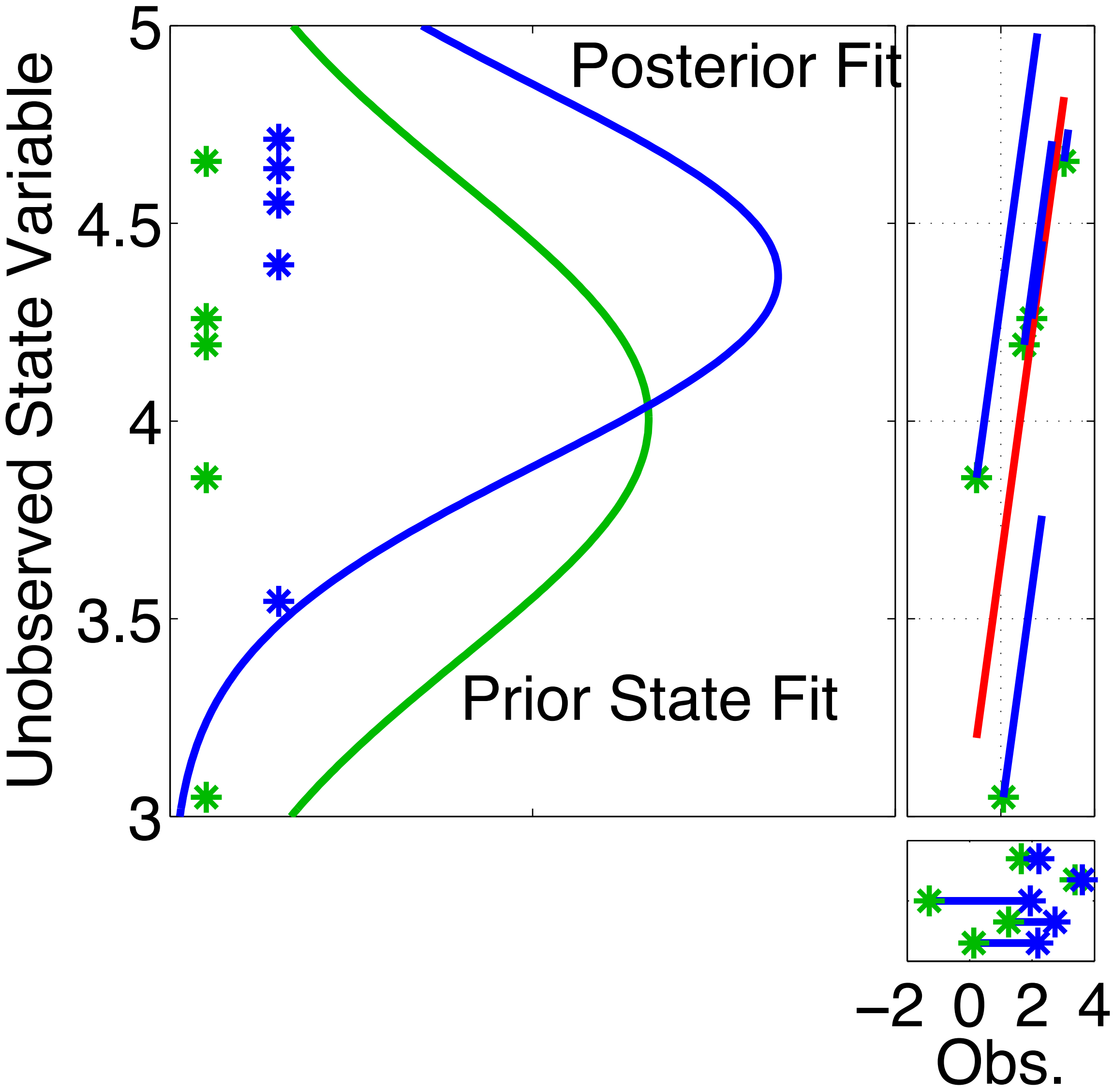
Ensemble filters: Updating additional prior state variables



Now have an updated (posterior) ensemble for the unobserved variable.

Fitting Gaussians shows that mean and variance have changed.

Ensemble filters: Updating additional prior state variables

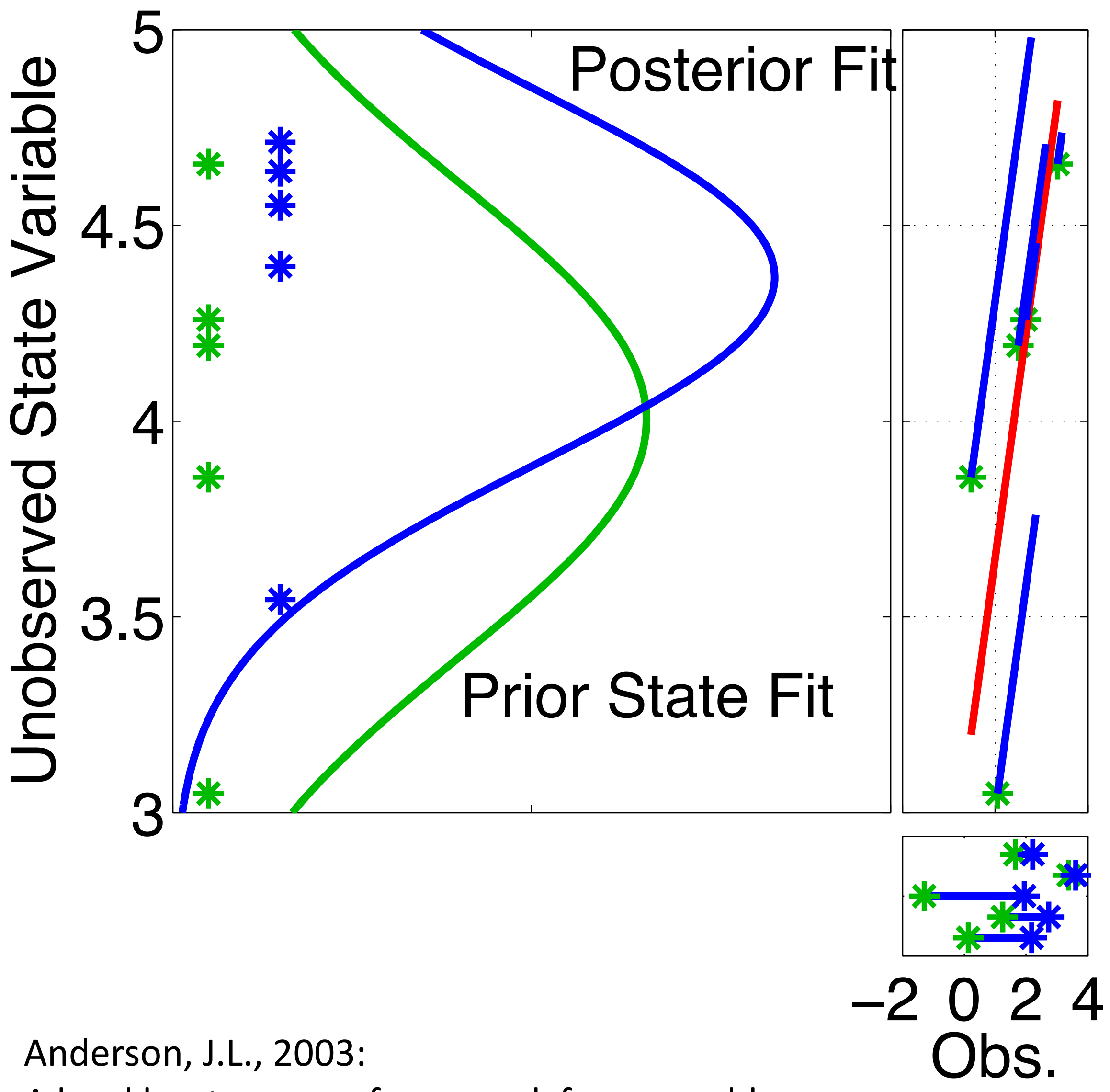


Now have an updated (posterior) ensemble for the unobserved variable.

Fitting Gaussians shows that mean and variance have changed.

Other features of the prior distribution may also have changed.

Ensemble filters: Updating additional prior state variables



CRITICAL POINT:

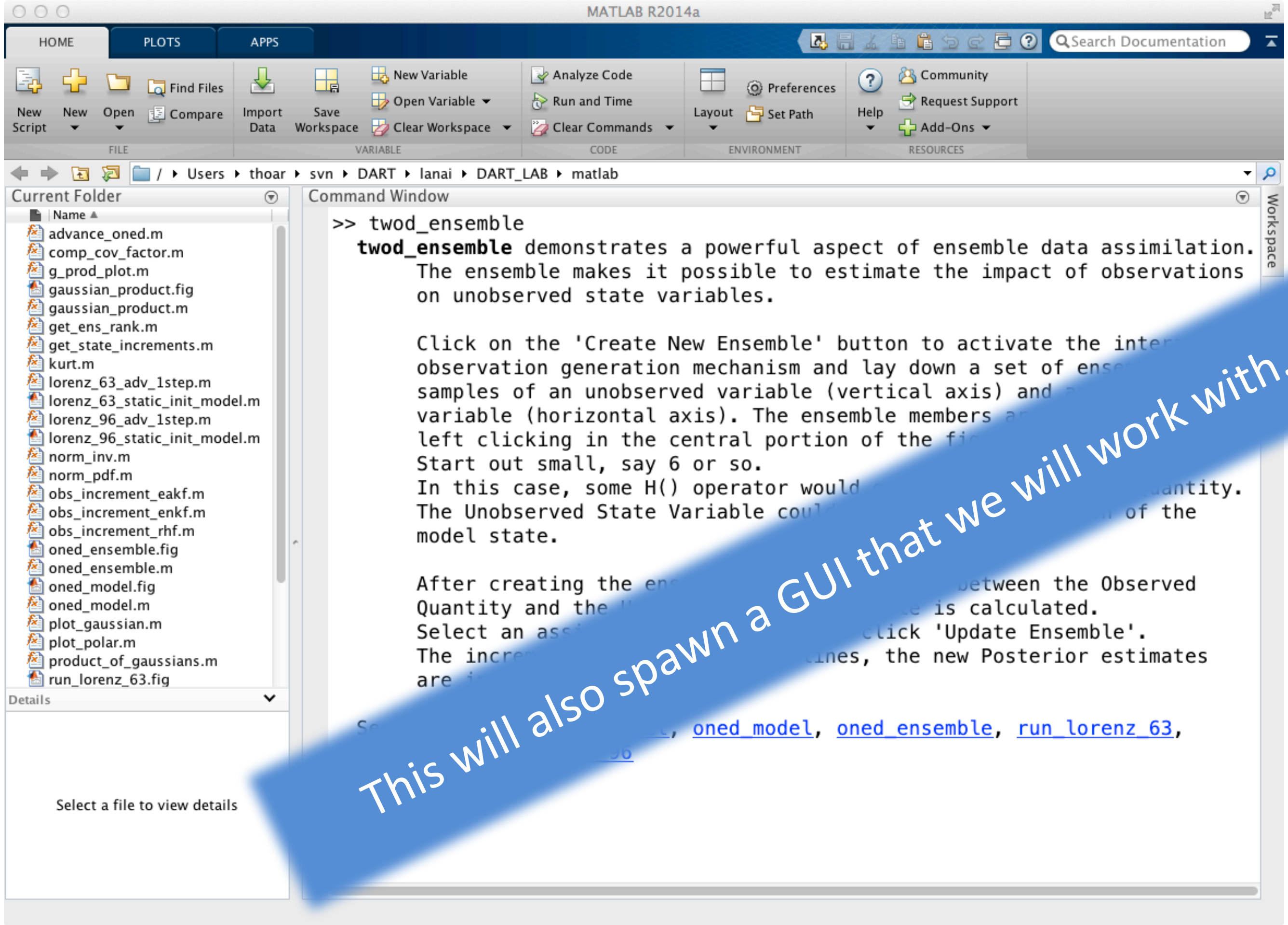
Since impact on unobserved variable is simply a linear regression, can do this INDEPENDENTLY for any number of unobserved variables!

Could also do many at once using matrix algebra as in traditional Kalman Filter.

Anderson, J.L., 2003:
A local least squares framework for ensemble filtering. *Mon. Wea. Rev.*, **131**, 634-642

Matlab Hands-On: **twod_ensemble**

Purpose: Explore how an unobserved state variable is updated by an observation of another state variable.



advance_oned.m
comp_cov_factor.m
g_prod_plot.m
gaussian_product.fig
gaussian_product.m
get_ens_rank.m
get_state_increments.m
kurt.m
lorenz_63_adv_1step.m
lorenz_63_static_init_model.m
lorenz_96_adv_1step.m
lorenz_96_static_init_model.m
norm_inv.m
norm_pdf.m
obs_increment_eakf.m
obs_increment_enkf.m
obs_increment_rhf.m
oned_ensemble.fig
oned_ensemble.m
oned_model.fig
oned_model.m
plot_gaussian.m
plot_polar.m
product_of_gaussians.m
run_lorenz_63.fig

```
>> twod_ensemble
  twod_ensemble demonstrates a powerful aspect of ensemble data assimilation.
      The ensemble makes it possible to estimate the impact of observations
      on unobserved state variables.

      Click on the 'Create New Ensemble' button to activate the inte
      observation generation mechanism and lay down a set of ense
      samples of an unobserved variable (vertical axis) and a
      variable (horizontal axis). The ensemble members ar
      left clicking in the central portion of the fi
      Start out small, say 6 or so.
      In this case, some H() operator would                  antity.
      The Unobserved State Variable coul                   of the
      model state.

      After creating the en                   etween the Observed
      Quantity and the                     is calculated.
      Select an ass                   click 'Update Ensemble'.
      The incre                 ines, the new Posterior estimates
      are

  Se                 , oned_model, oned_ensemble, run_lorenz_63,
                  96
```

This will also spawn a GUI that we will work with.

Matlab Hands-On: twod_ensemble

Bivariate ensemble plot with projected marginals for observed, unobserved variables.

Start creating an ensemble. See next slide.

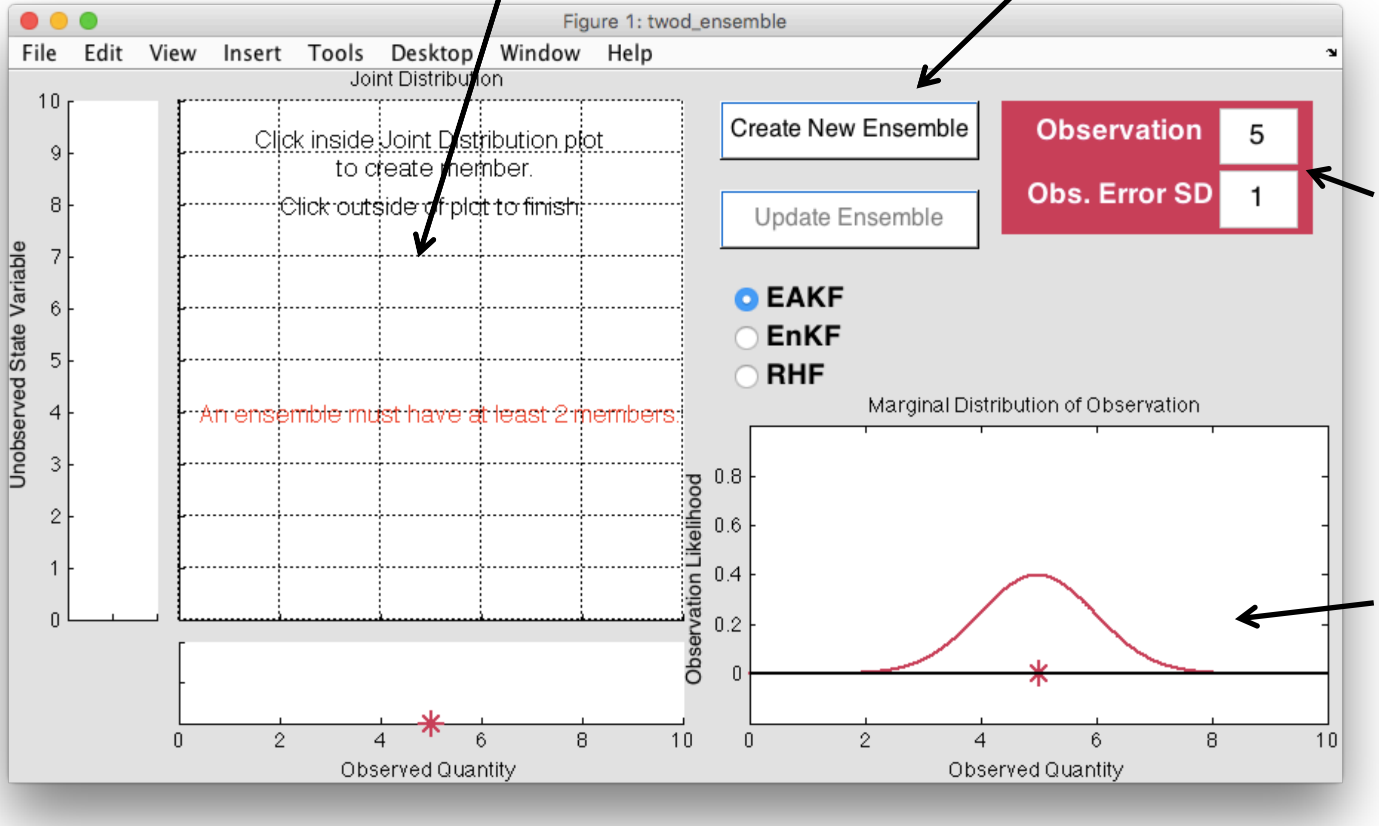


Figure 1: twod_ensemble
File Edit View Insert Tools Desktop Window Help
Joint Distribution
Click inside Joint Distribution plot to create member.
Click outside of plot to finish.
An ensemble must have at least 2 members.
Unobserved State Variable
Observed Quantity
Create New Ensemble
Update Ensemble
Observation 5
Obs. Error SD 1
EAKF
EnKF
RHF
Marginal Distribution of Observation
Observation Likelihood
Observed Quantity

Selected: ● EAKF ○ EnKF ○ RHF; Observation: 5; Obs. Error SD: 1

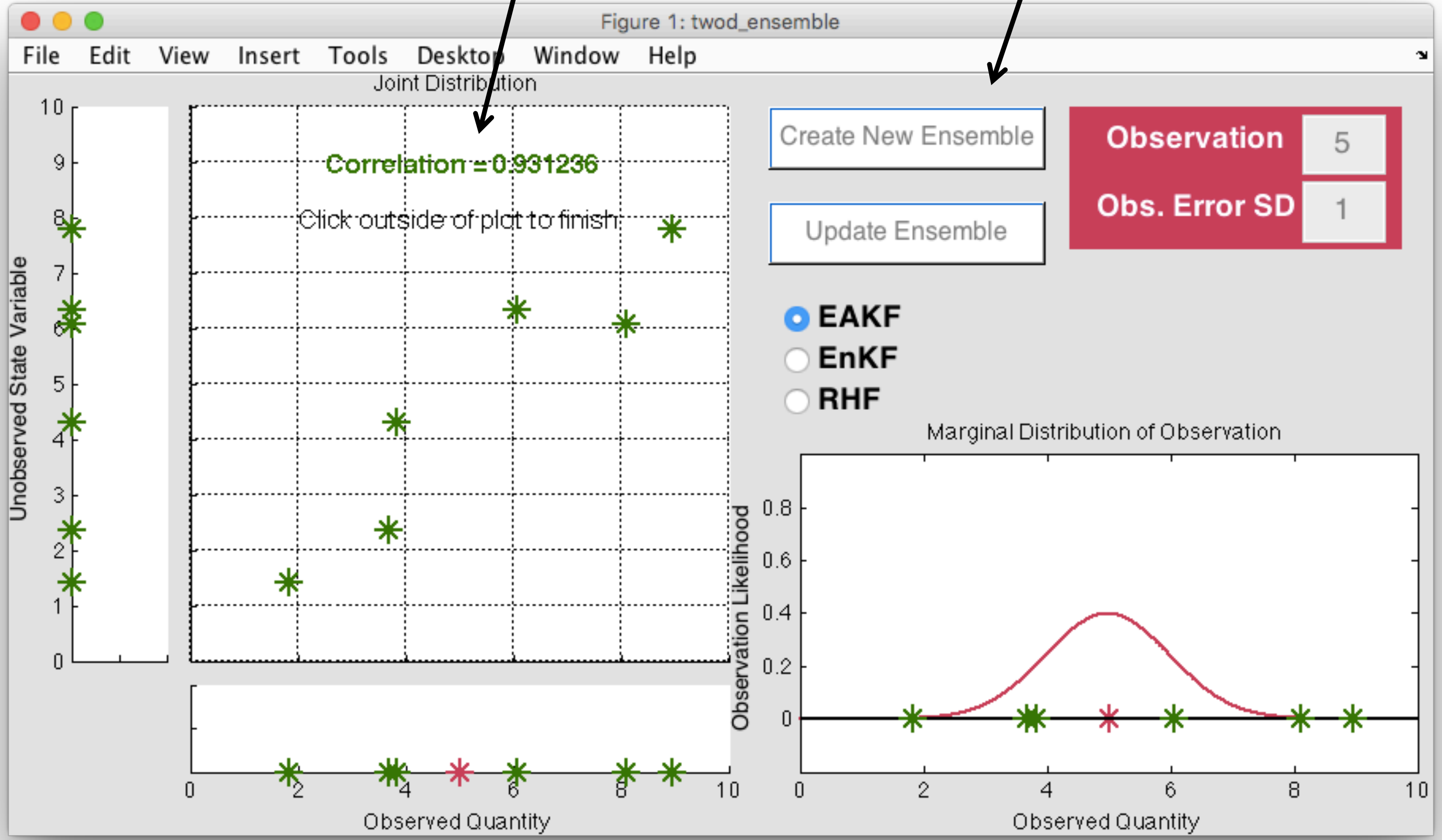
Control observation value and error; same as oned_ensemble.

Detailed plot for observed variable; same as oned_ensemble.

Matlab Hands-On: twod_ensemble

Move cursor and click in this frame to create ensemble members. Click outside this frame when all members are created.

Start creating an ensemble.



Matlab Hands-On: twod_ensemble

Explorations:

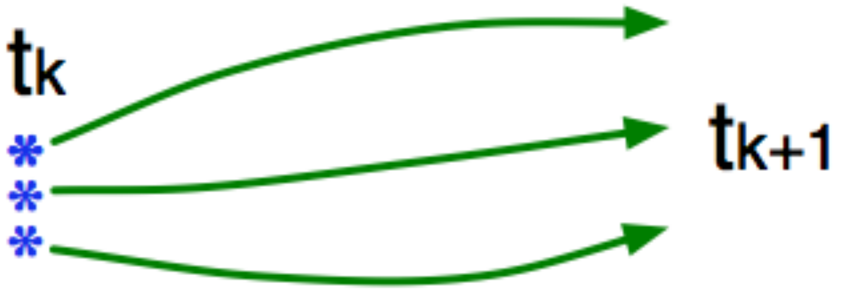
- Create ensemble members that are nearly on a line. Explore how the unobserved variable is updated.
- What happens for nearly uncorrelated observed and unobserved variables? Create a roundish cloud of points for the prior.
- What happens with a two-dimensional bimodal distribution?
- Try prior ensembles with various types of outliers.

Schematic of an Ensemble Filter for Geophysical Data Assimilation

1. Use model to advance ensemble (3 members here) to time at which next observation becomes available.

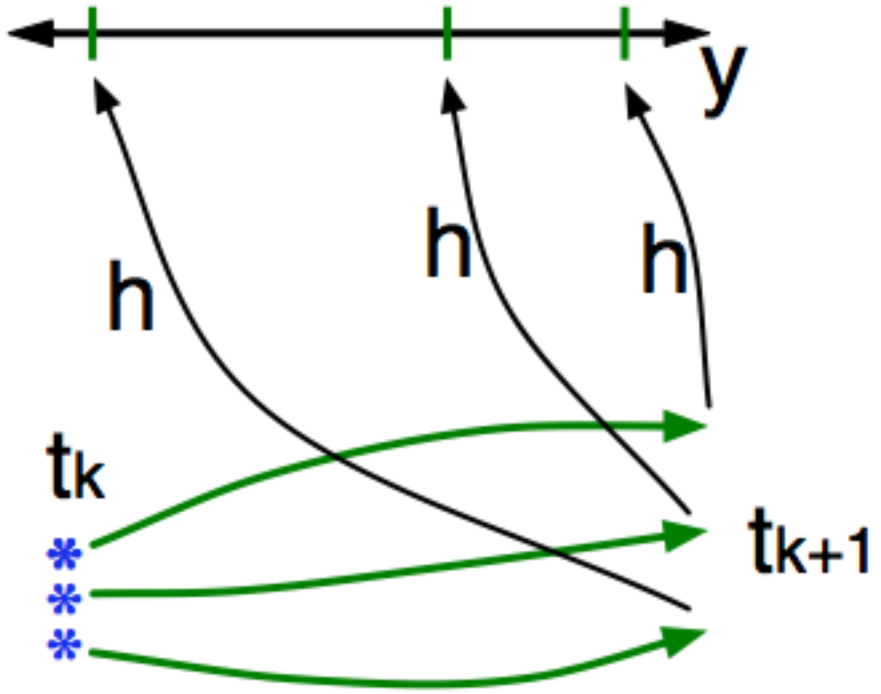
Ensemble state estimate after using previous observation (analysis)

Ensemble state at time of next observation (prior)



How an Ensemble Filter Works for Geophysical Data Assimilation

2. Get prior ensemble sample of observation, $y = h(x)$, by applying forward operator **h** to each ensemble member.

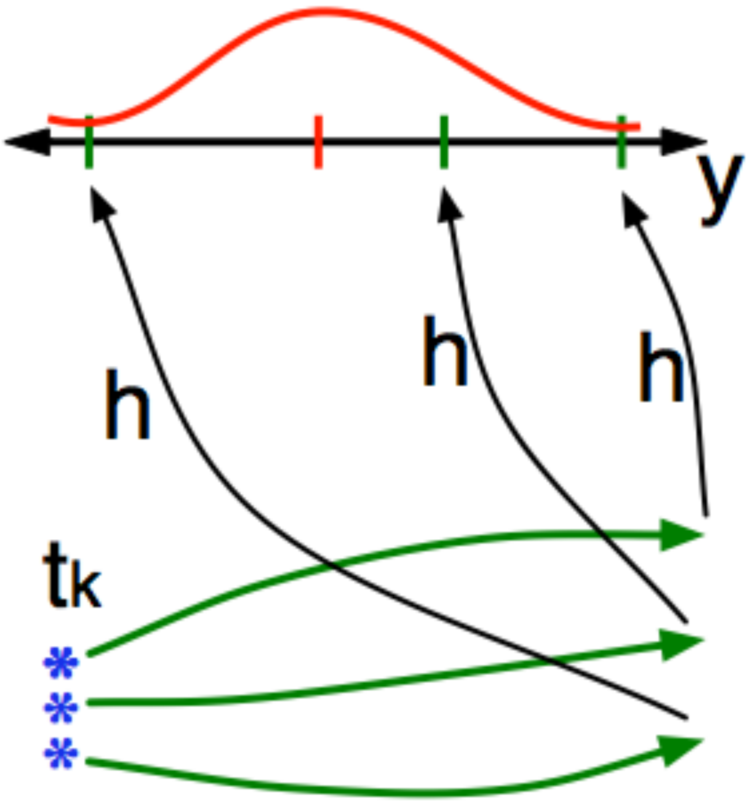


Theory: observations from instruments with uncorrelated errors can be done sequentially.

Houtekamer, P.L. and H.L. Mitchell, 2001:
A sequential ensemble Kalman filter for atmospheric data assimilation.
Mon. Wea. Rev., **129**, 123-137

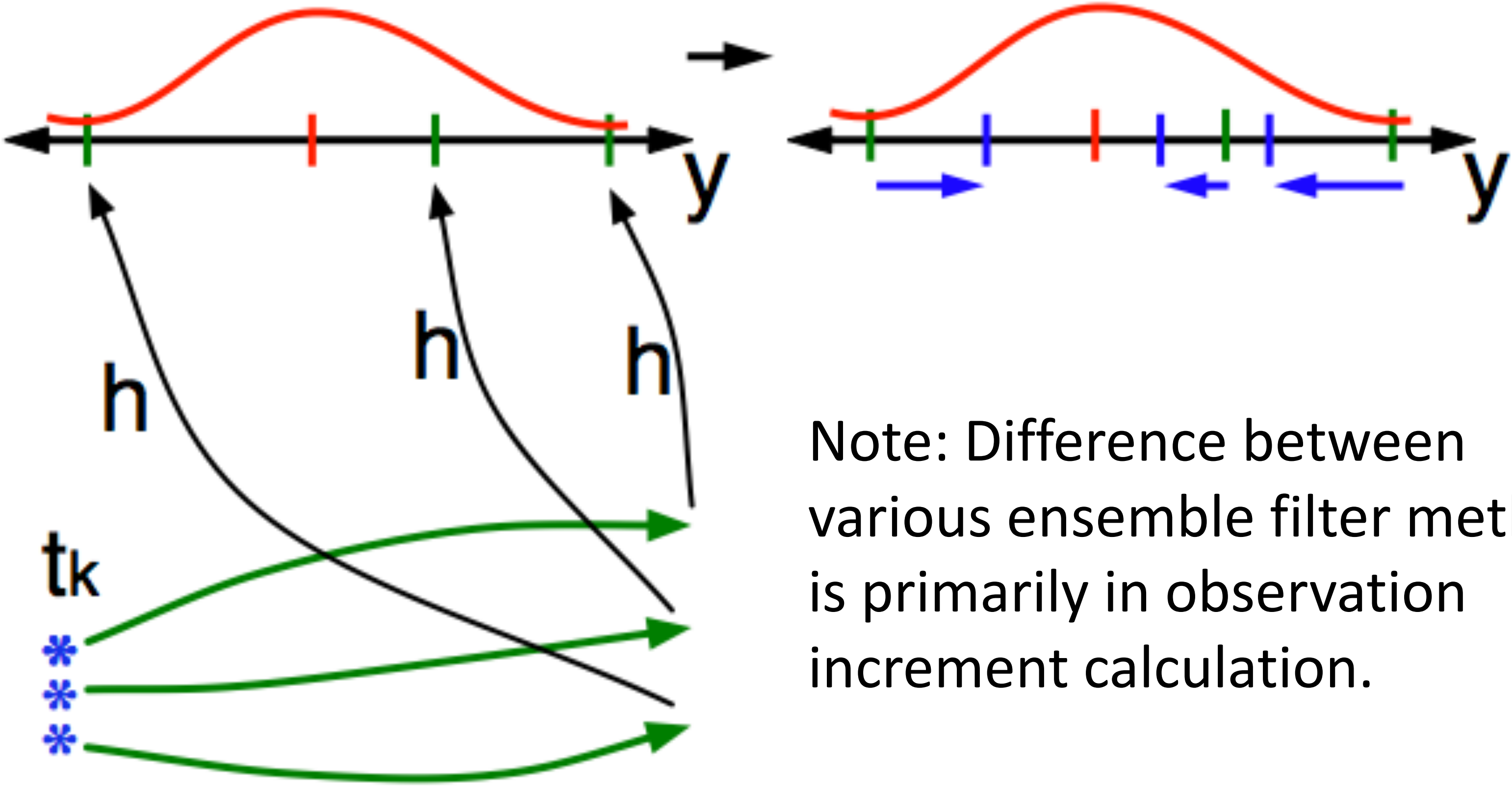
How an Ensemble Filter Works for Geophysical Data Assimilation

3. Get observed value and observational error distribution from observing system.



How an Ensemble Filter Works for Geophysical Data Assimilation

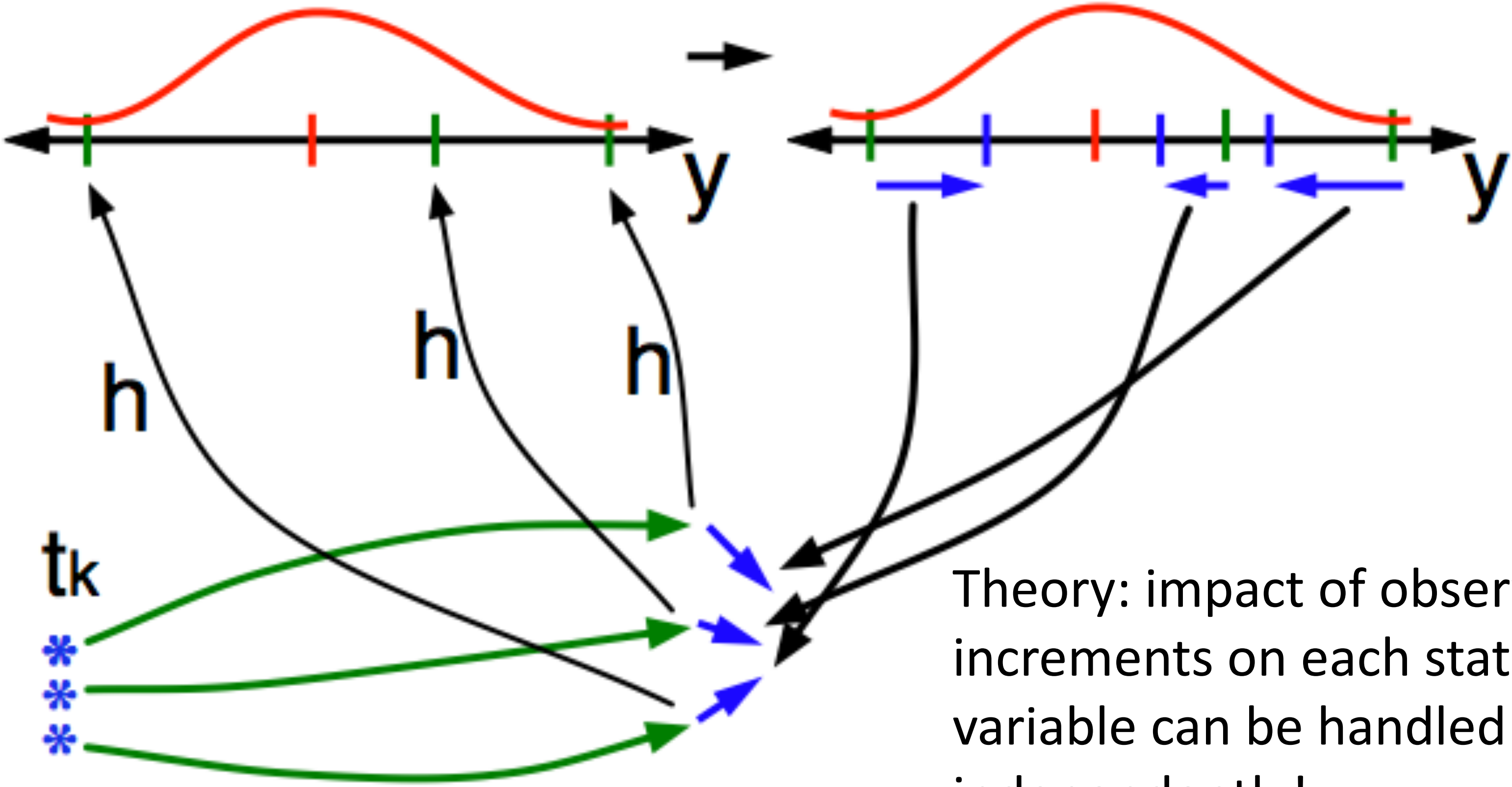
4. Find the increments for the prior observation ensemble (this is a scalar problem for uncorrelated observation errors).



Note: Difference between various ensemble filter methods is primarily in observation increment calculation.

How an Ensemble Filter Works for Geophysical Data Assimilation

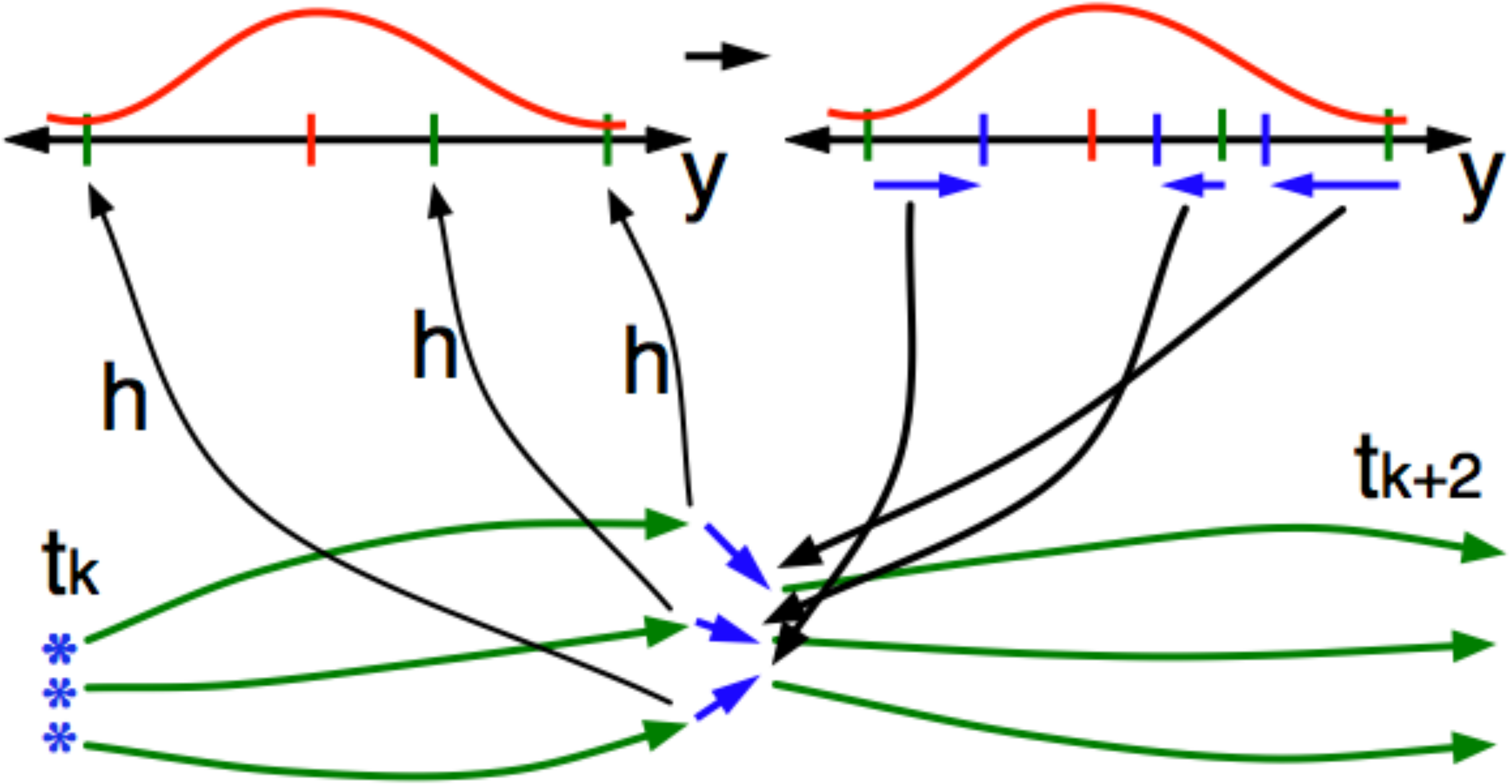
5. Use ensemble samples of y and each state variable to linearly regress observation increments onto state variable increments.



Theory: impact of observation increments on each state variable can be handled independently!

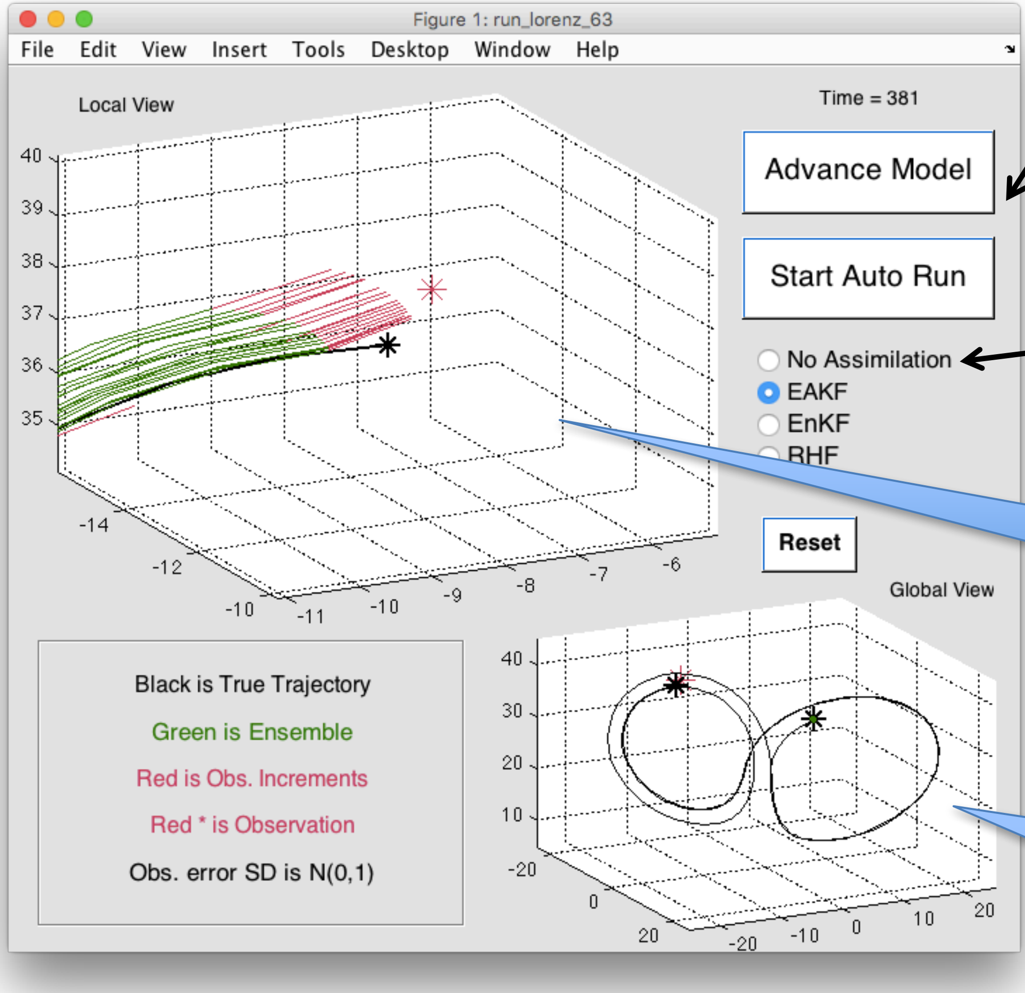
How an Ensemble Filter Works for Geophysical Data Assimilation

6. When all ensemble members for each state variable are updated, there is a new analysis. Integrate to time of next observation …



Matlab Hands-On: **run_lorenz_63**

Purpose: Explore behavior of ensemble Kalman filters in a low-order, chaotic dynamical system, the 3-variable Lorenz 1963 model.



○ No Assimilation
● EAKF
○ EnKF
○ RHF

Black is True Trajectory

Green is Ensemble

Red is Obs. Increments

Red * is Observation

Obs. error SD is N(0,1)

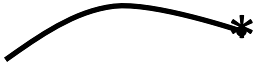
These controls work the same as for oned_model.

Assimilation can be turned off, just does model advances.

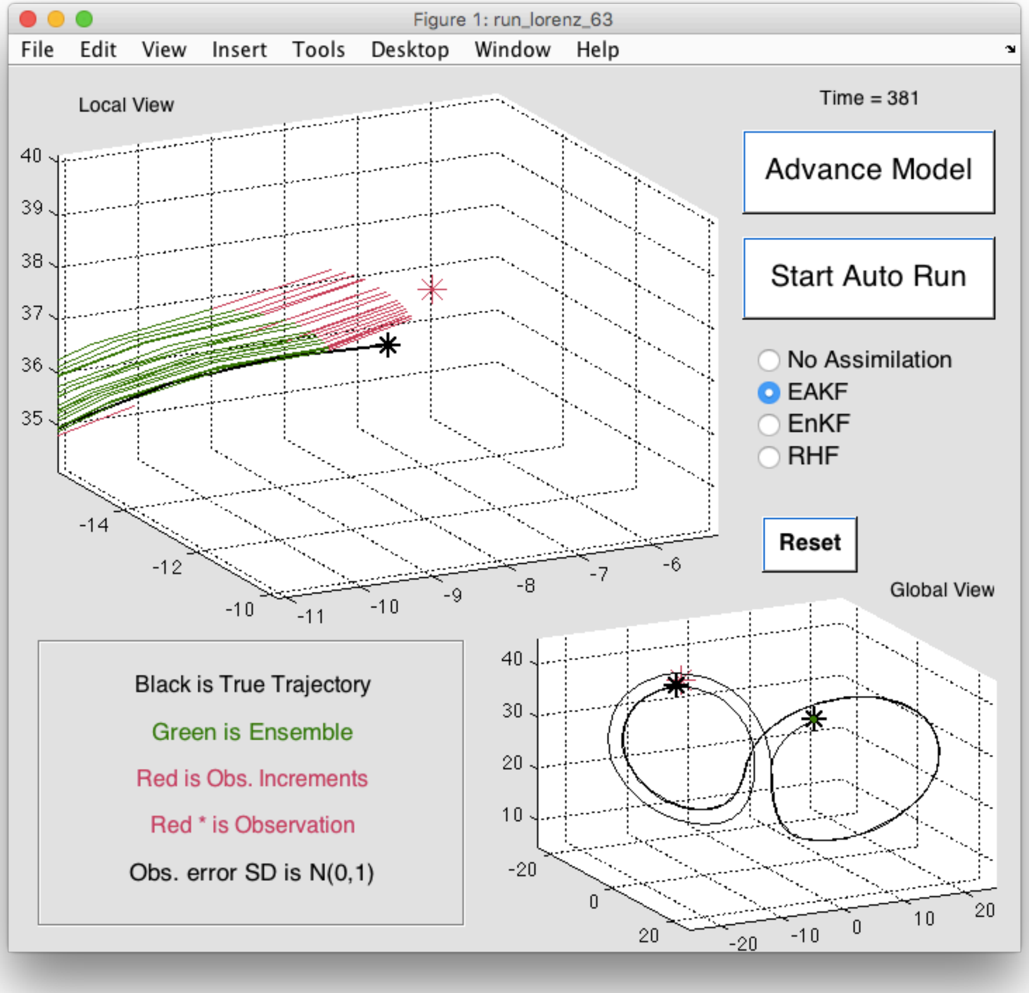
‘Local’ domain, local timeframe

Full domain, full timeframe.

Matlab Hands-On: run_lorenz_63

Both panels show time evolution of true state (black). 

20 ensemble members are shown in green in top window.



○ No Assimilation
● EAKF
○ EnKF
○ RHF

Black is True Trajectory

Green is Ensemble

Red is Obs. Increments

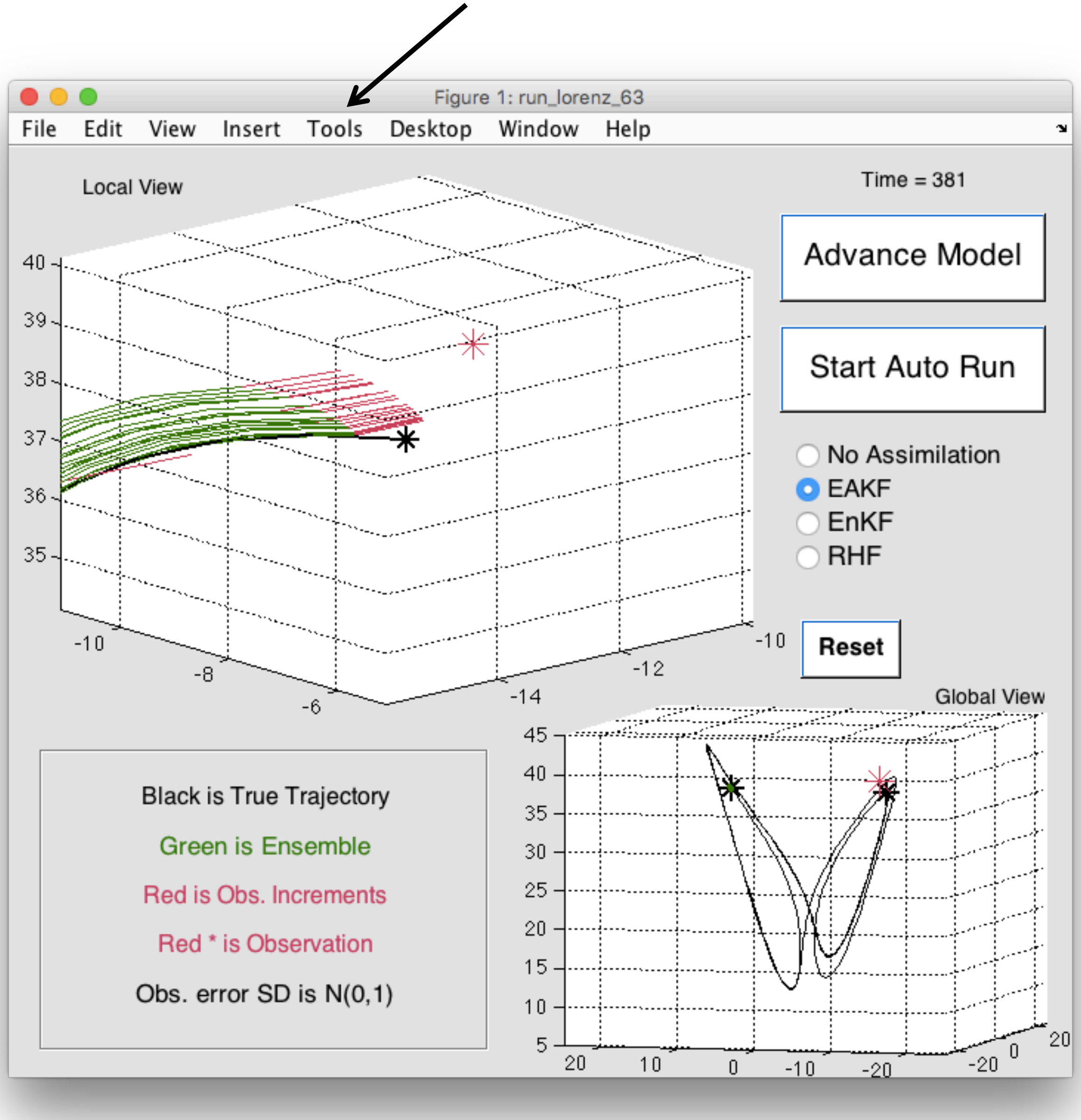
Red * is Observation

Obs. error SD is N(0,1)

At each observation time, the three components of the truth are 'observed’ by adding a random draw from a standard normal distribution to the true value.

Matlab Hands-On: run_lorenz_63

You can use matlab tools to modify plots.



○ No Assimilation
● EAKF
○ EnKF
○ RHF

Black is True Trajectory

Green is Ensemble

Red is Obs. Increments

Red * is Observation

Obs. error SD is N(0,1)

Here, the Rotate 3D tools has been used to change the angle of view of both the local and global views of the assimilation.

Matlab Hands-On: run_lorenz_63

Explorations:

- Select Start Auto Run and watch the evolution of the ensemble. Try to understand how the ensemble spreads out.

- Restart the GUI and select EAKF. Do individual advances and assimilations and observe the behavior.

- Do some auto runs with assimilation turned on.

- Explore how different areas of the attractor have different assimilation behavior.

Matlab Hands-On: **run_lorenz_96**

Purpose: Explore the behavior of ensemble filters in a 40-variable chaotic dynamical system; the Lorenz 1996 model.

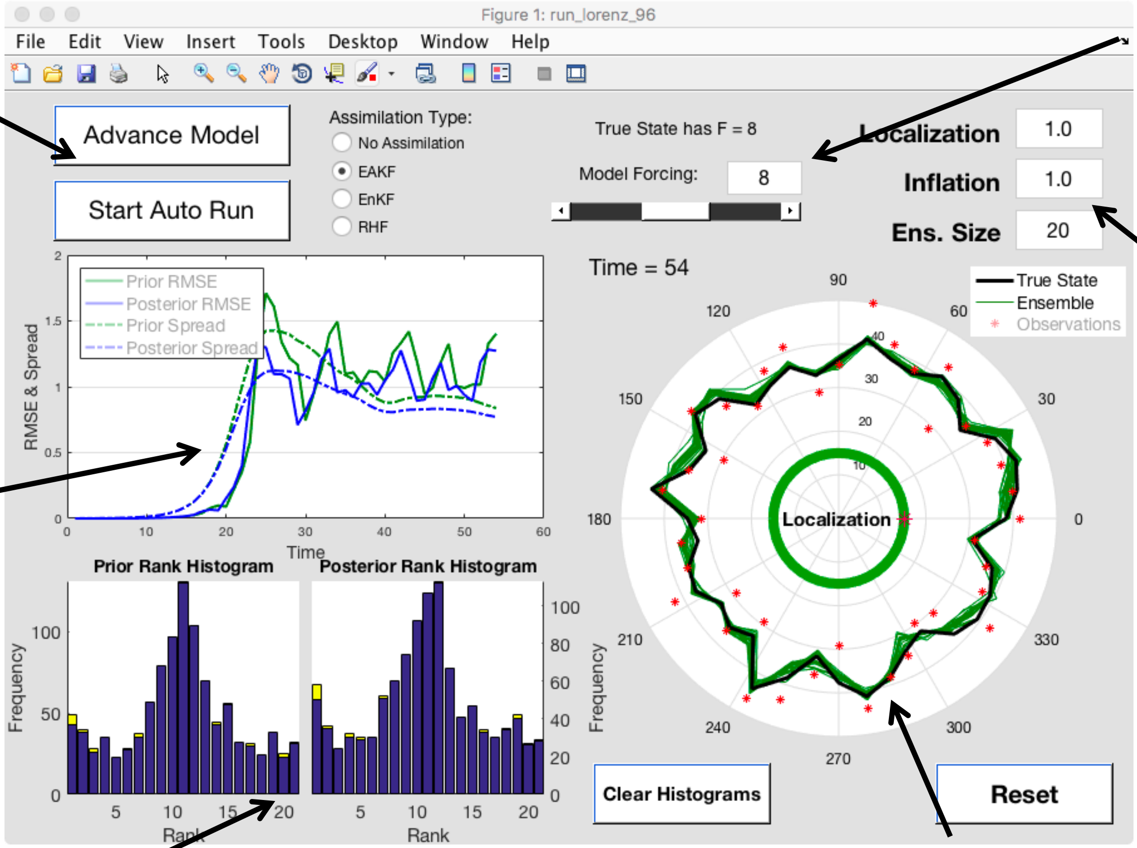


Figure 1: run_lorenz_96

Assimilation Type:

○ No Assimilation
● EAKF
○ EnKF
○ RHF

True State has F = 8

Model Forcing: 8

Localization 1.0

Inflation 1.0

Ens. Size 20

Time = 54

These controls work the same as lorenz_63.

Root mean square error from truth and ensemble spread as function of time.

Prior and posterior rank histograms.

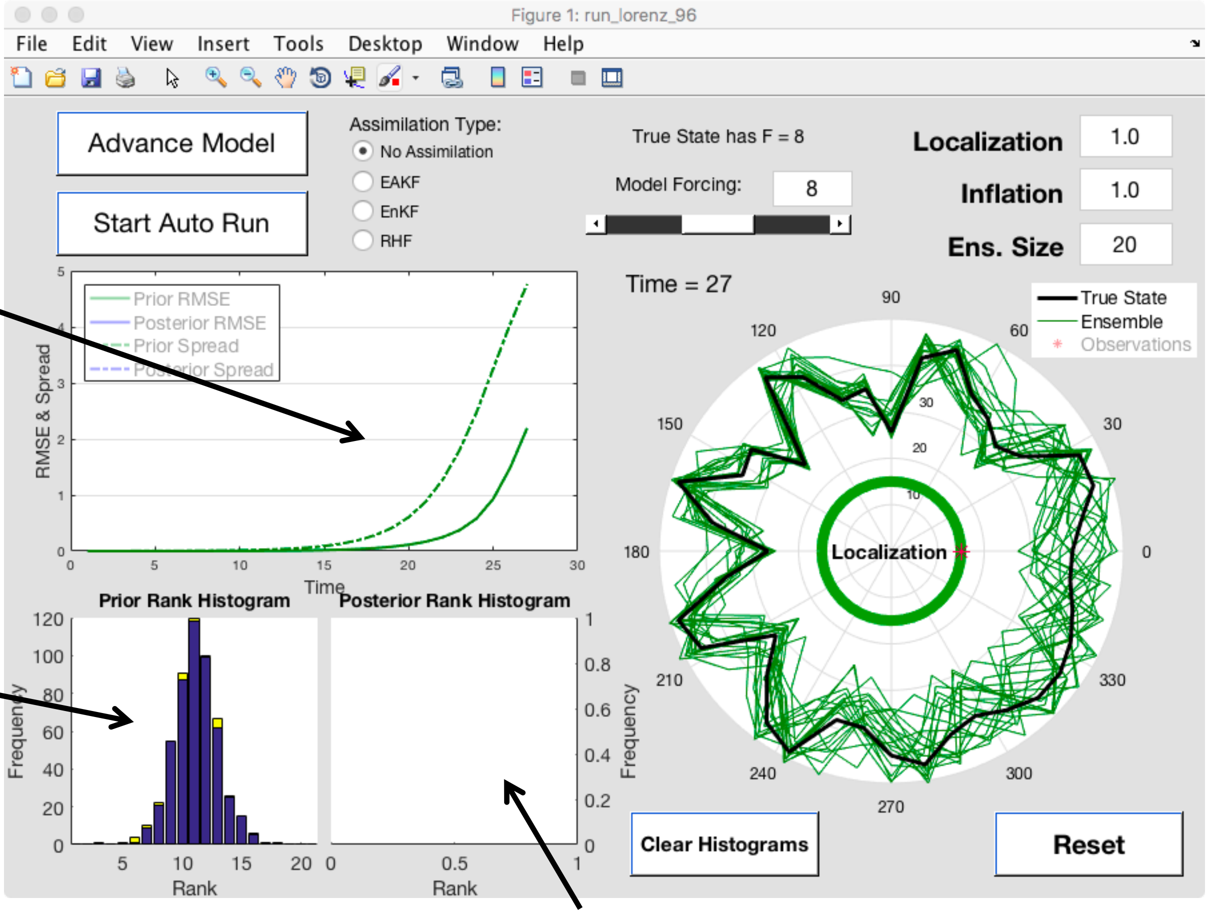
Model forcing.

Parameters for ensemble filter.

Ensemble of model contours (spaghetti plot).

Matlab Hands-On: run_lorenz_96

Start a Free Run of the ensemble (No Assimilation). After some time, the minute perturbations in the original states lead to visibly different model states.



Assimilation Type:
● No Assimilation
○ EAKF
○ EnKF
○ RHF

True State has F = 8

Model Forcing: 8

Localization 1.0

Inflation 1.0

Ens. Size 20

Time = 27

Takes a while for the small initial perturbations to grow.

Ensemble size is 20, so there are 21 bins in the rank histogram.

No posterior in a free run.

Matlab Hands-On: run_lorenz_96

1) Stop the free run after some time.
2) Turn on the EAKF
3) Advance model, assimilate…

Note: All 40 state variables are observed. Observation error standard deviation is 4.0

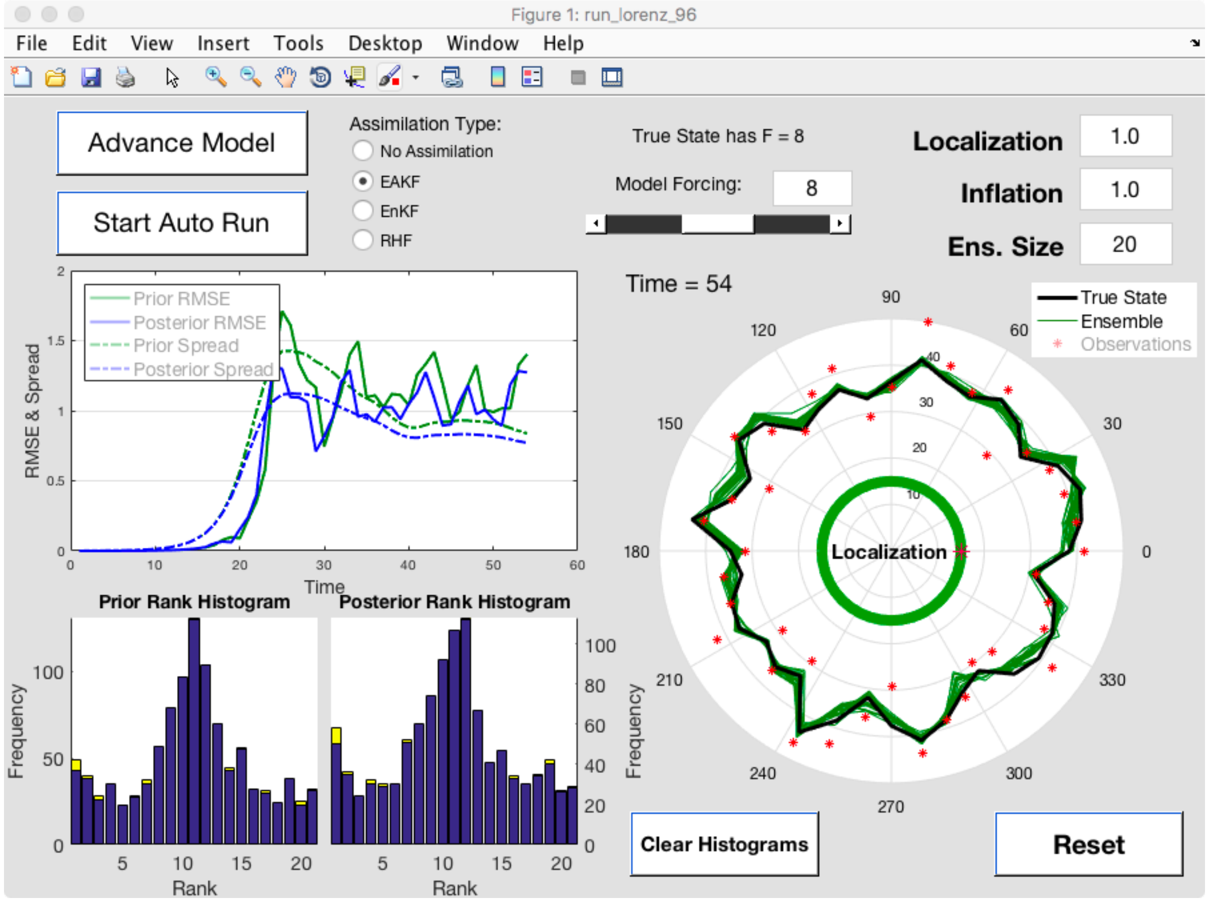


Figure 1: run_lorenz_96

Assimilation Type:
○ No Assimilation
● EAKF
○ EnKF
○ RHF

True State has F = 8

Model Forcing: 8

Localization 1.0

Inflation 1.0

Ens. Size 20

Time = 54

Your figures will be different depending on your settings. That's OK.

Matlab Hands-On: run_lorenz_96

Explorations:

- Do an extended free run to see error growth in the ensemble. How long does it take to saturate?
- Select EAKF and explore how the assimilation works.
- Try adding inflation (maybe 1.4) and repeat.