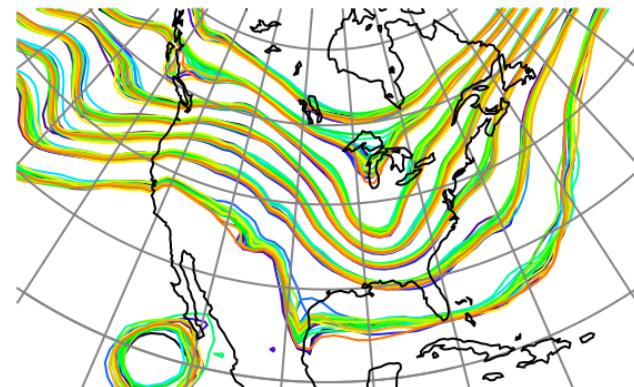


D  
A  
R  
T

ata  
ssimilation  
esearch  
estbed



## DART Tutorial Section 4: How should observations impact an unobserved state variable? Multivariate assimilation.



©UCAR 2014

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



# Single observed variable, single unobserved variable.

So far, have known observation likelihood for single variable.

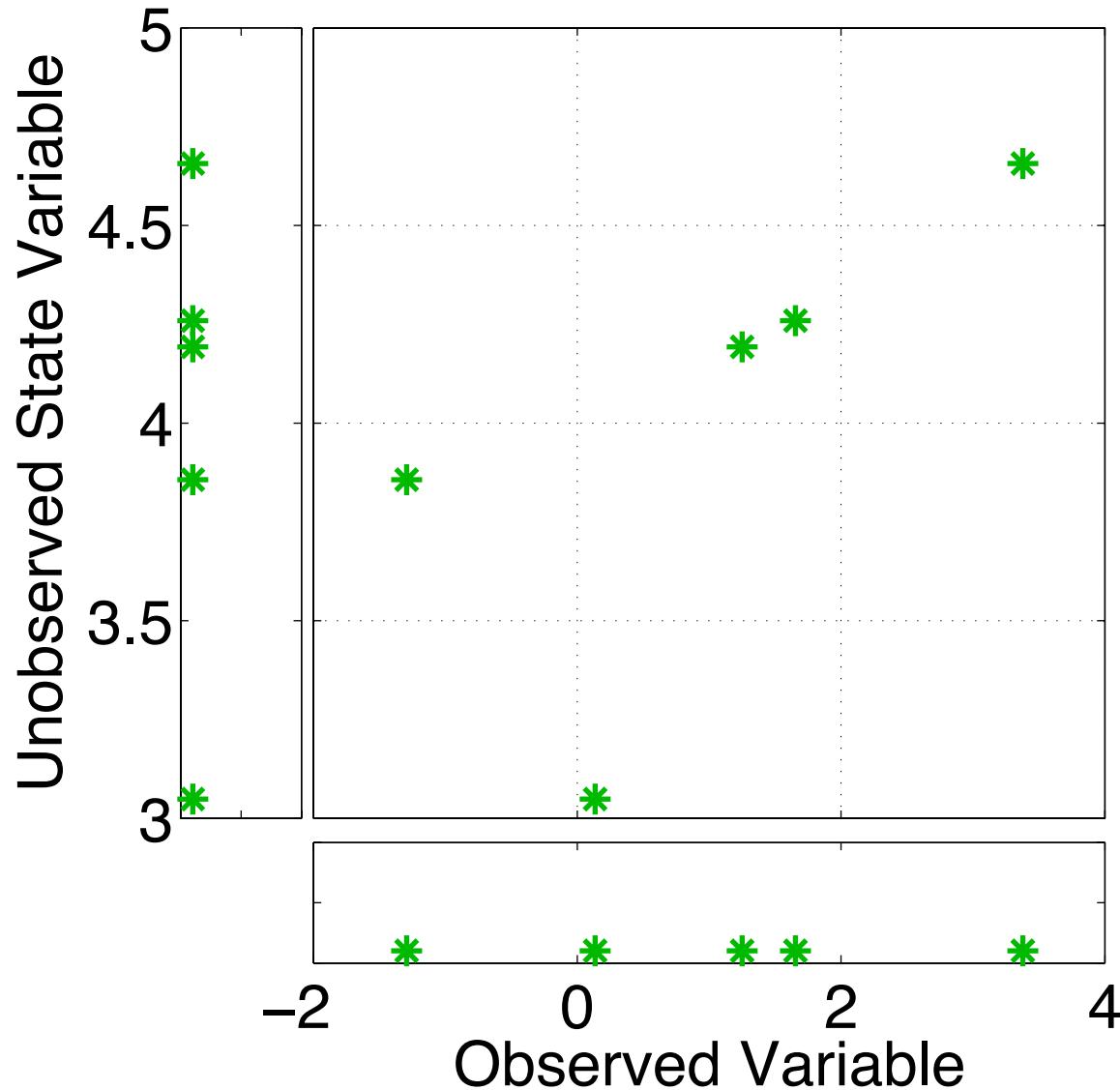
Now, suppose prior has an additional variable.

Will examine how ensemble methods update additional variable.

Basic method generalizes to any number of additional variables.

Methods related to Kalman filter in some sense, but not done here.

# Ensemble filters: Updating additional prior state variables

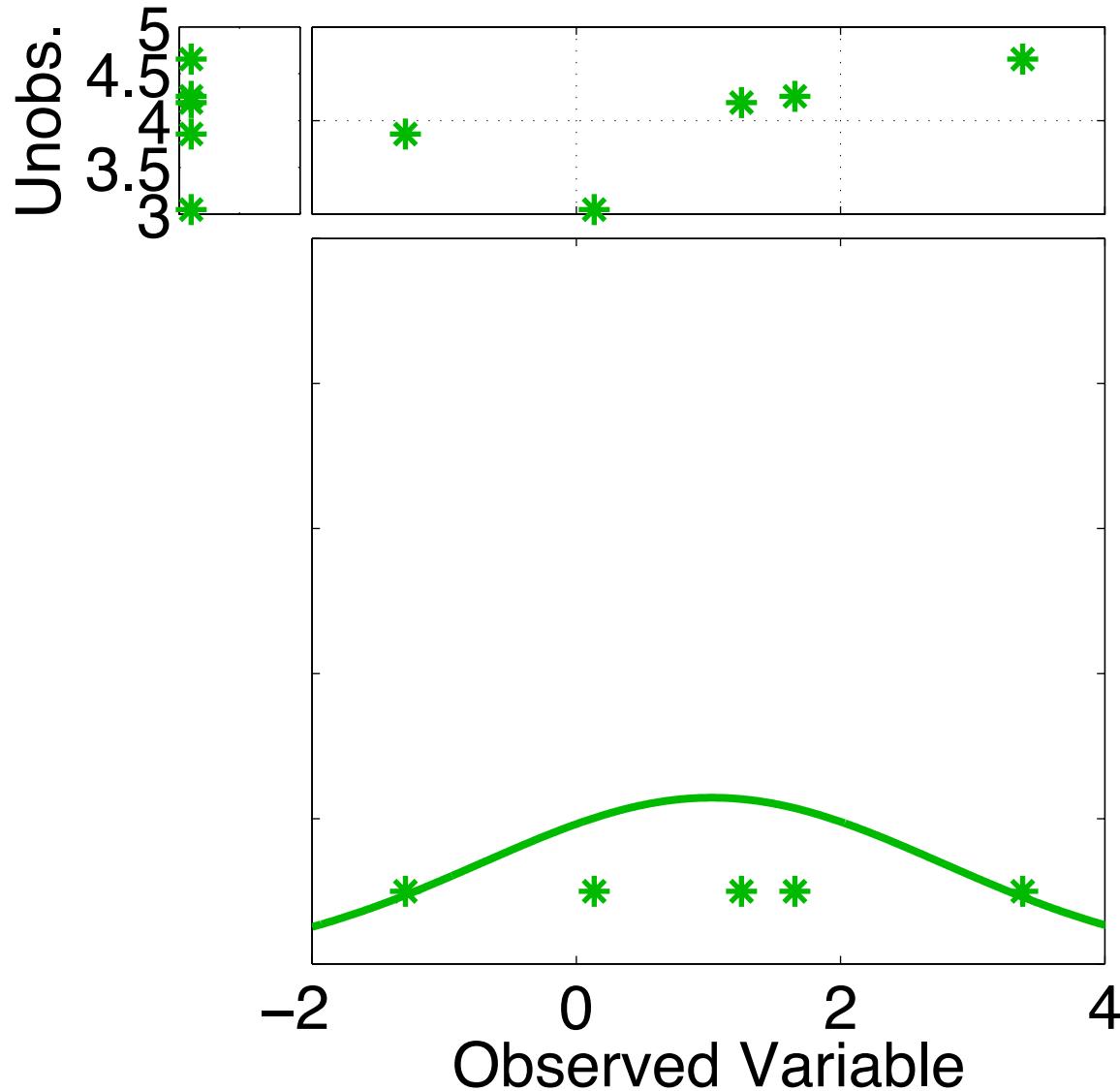


Assume that all we know is the prior joint distribution.

One variable is observed.

What should happen to the unobserved variable?

# Ensemble filters: Updating additional prior state variables

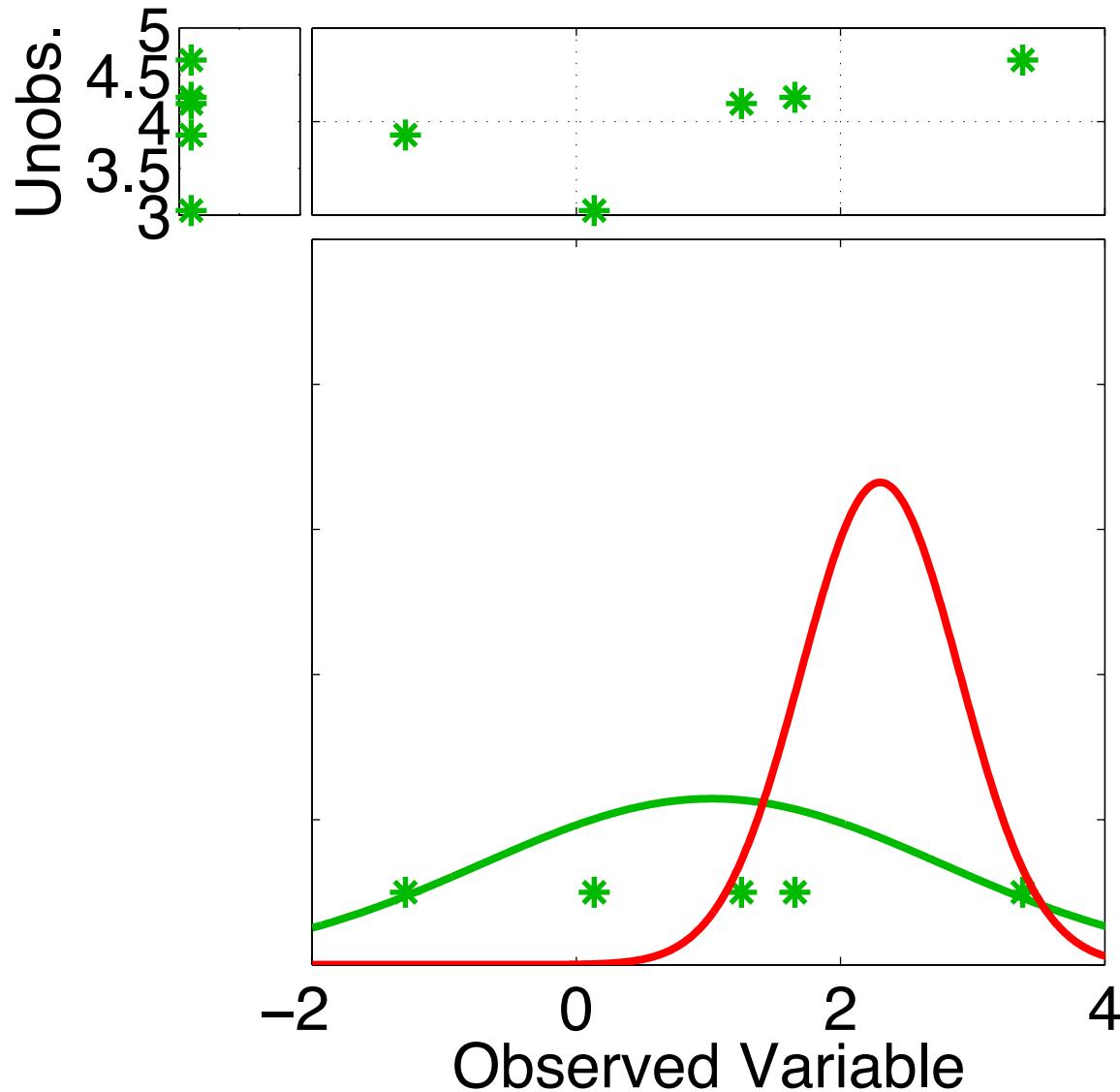


Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with one of the previous methods.

# Ensemble filters: Updating additional prior state variables

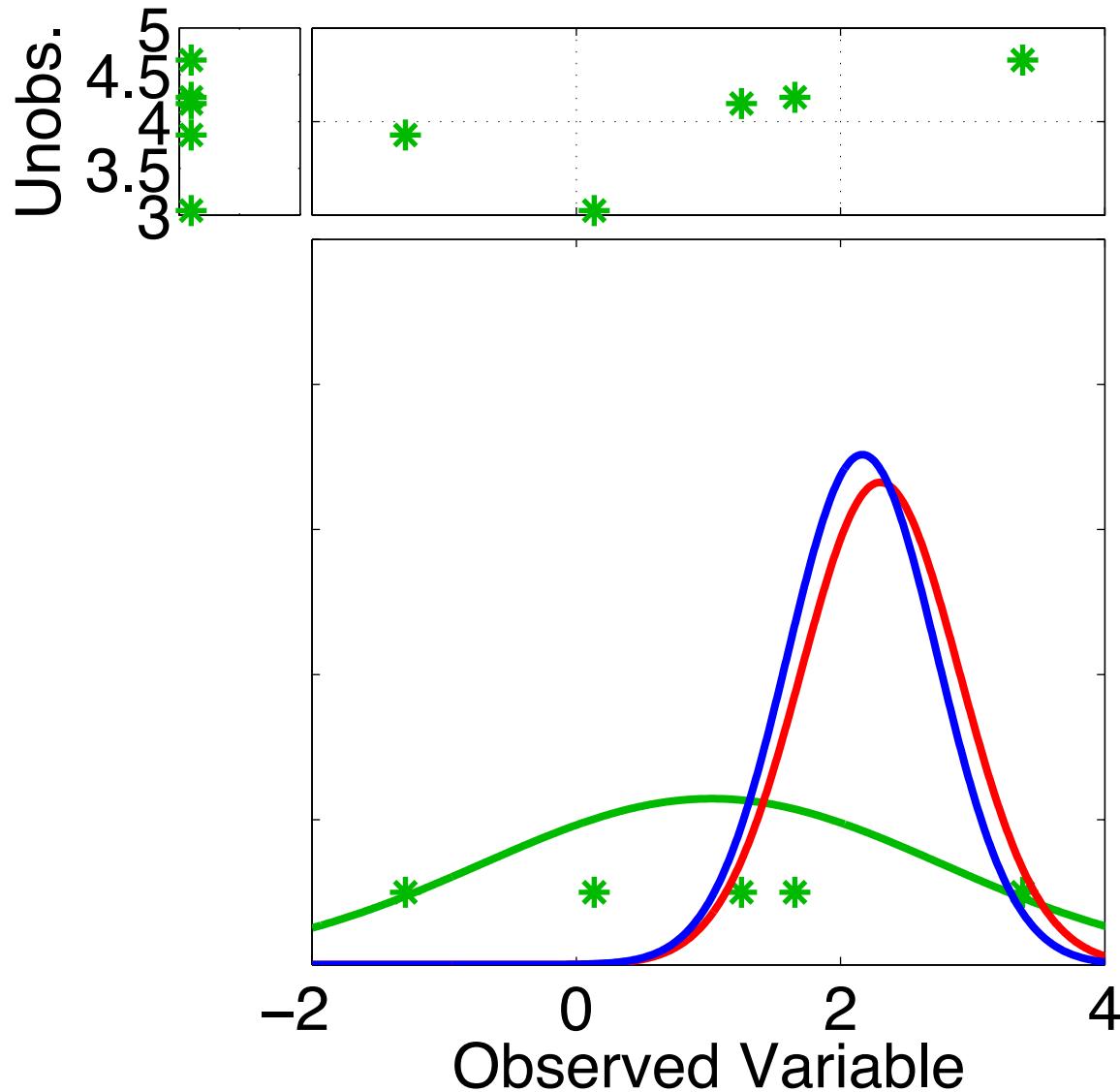


Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with one of the previous methods.

# Ensemble filters: Updating additional prior state variables

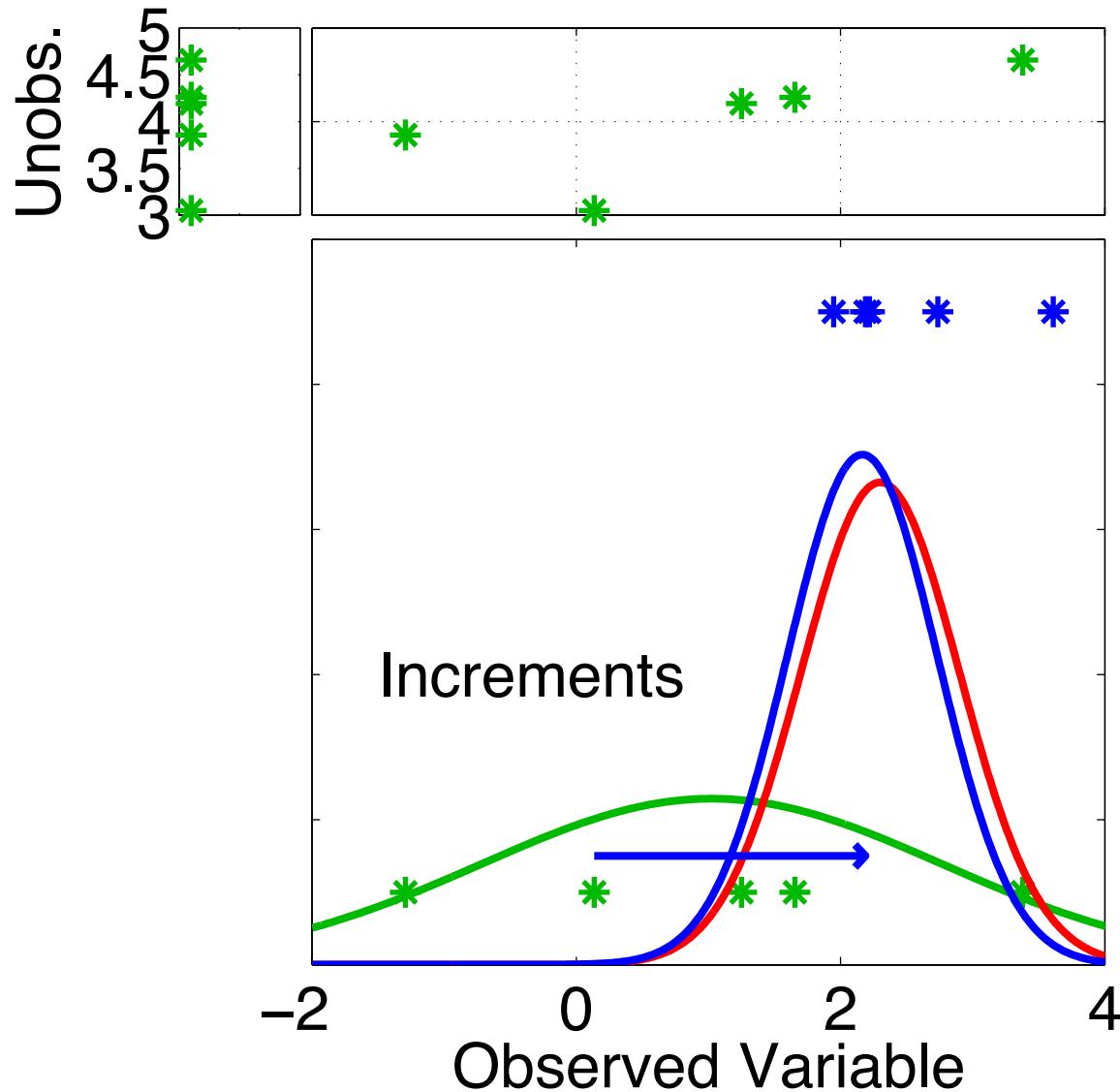


Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with one of the previous methods.

# Ensemble filters: Updating additional prior state variables

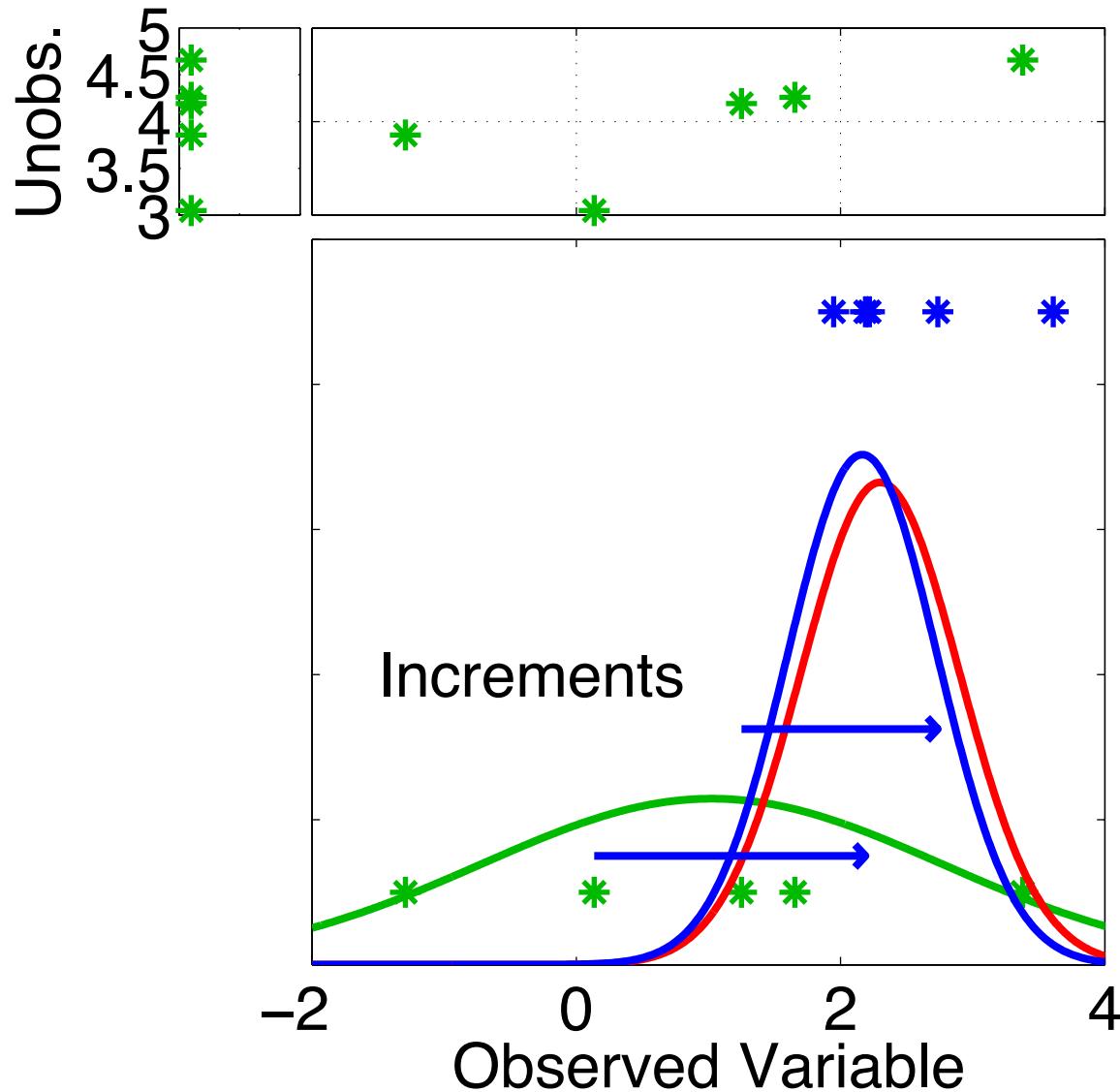


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

# Ensemble filters: Updating additional prior state variables

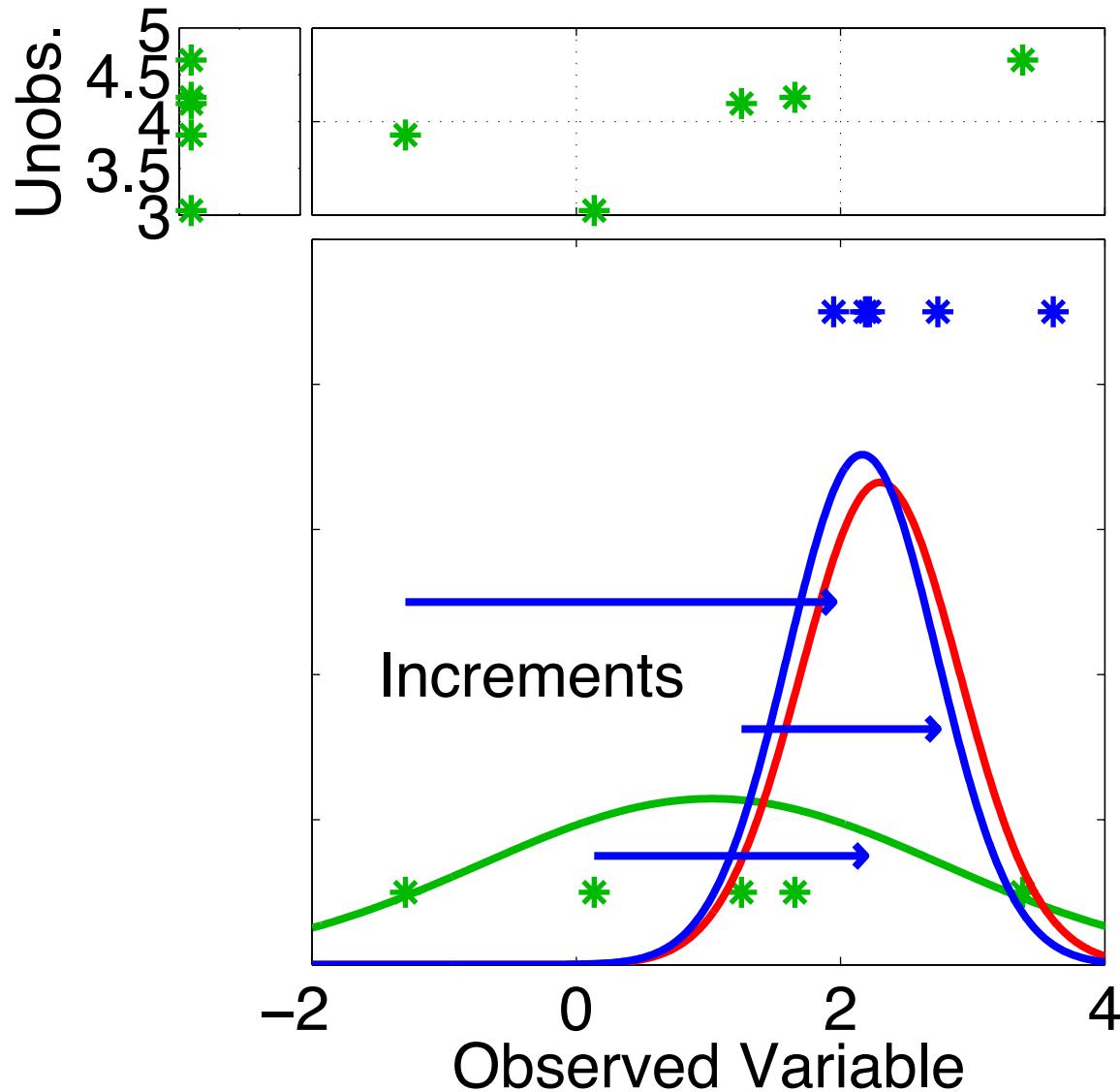


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

# Ensemble filters: Updating additional prior state variables

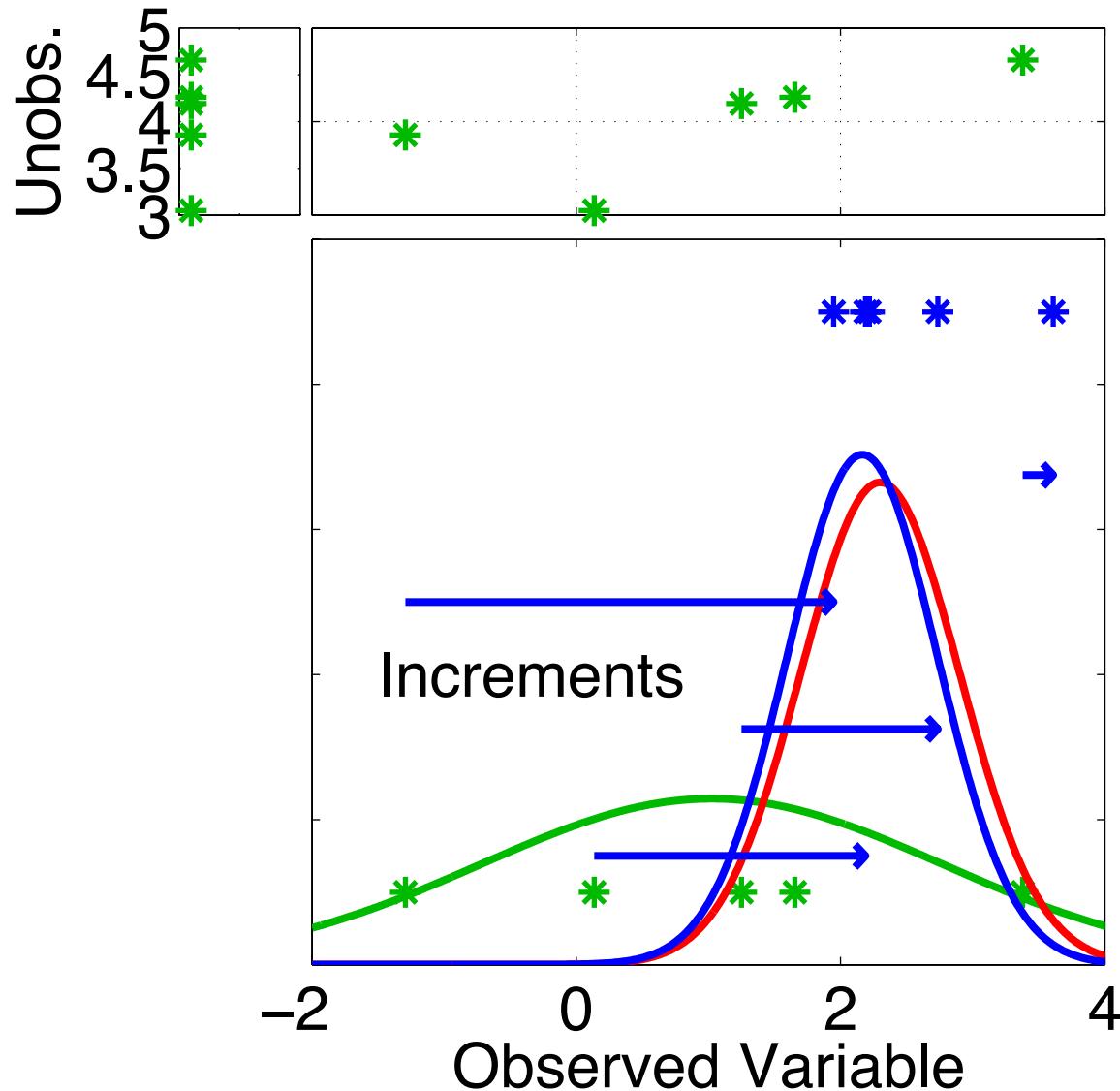


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

# Ensemble filters: Updating additional prior state variables

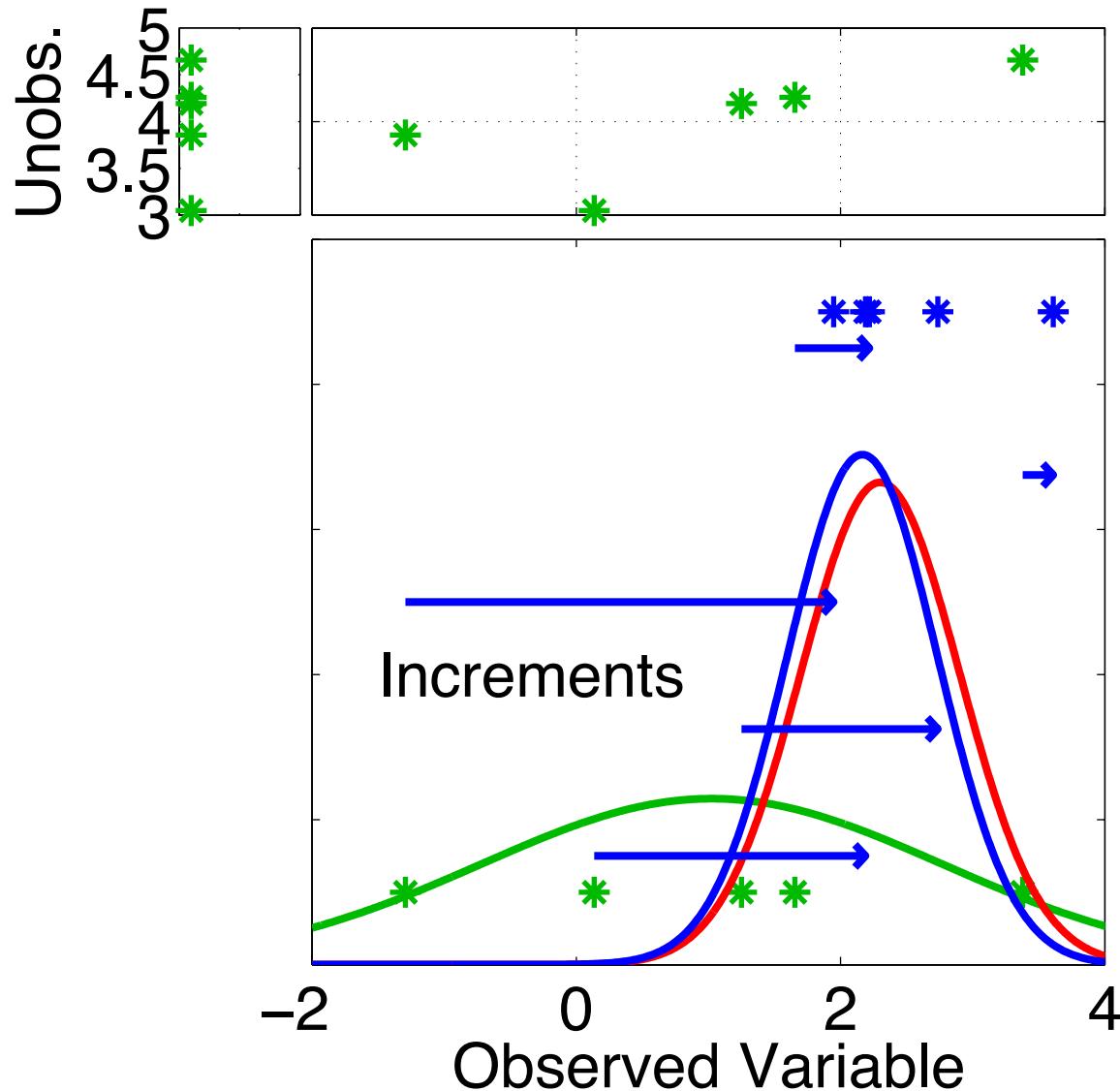


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

# Ensemble filters: Updating additional prior state variables

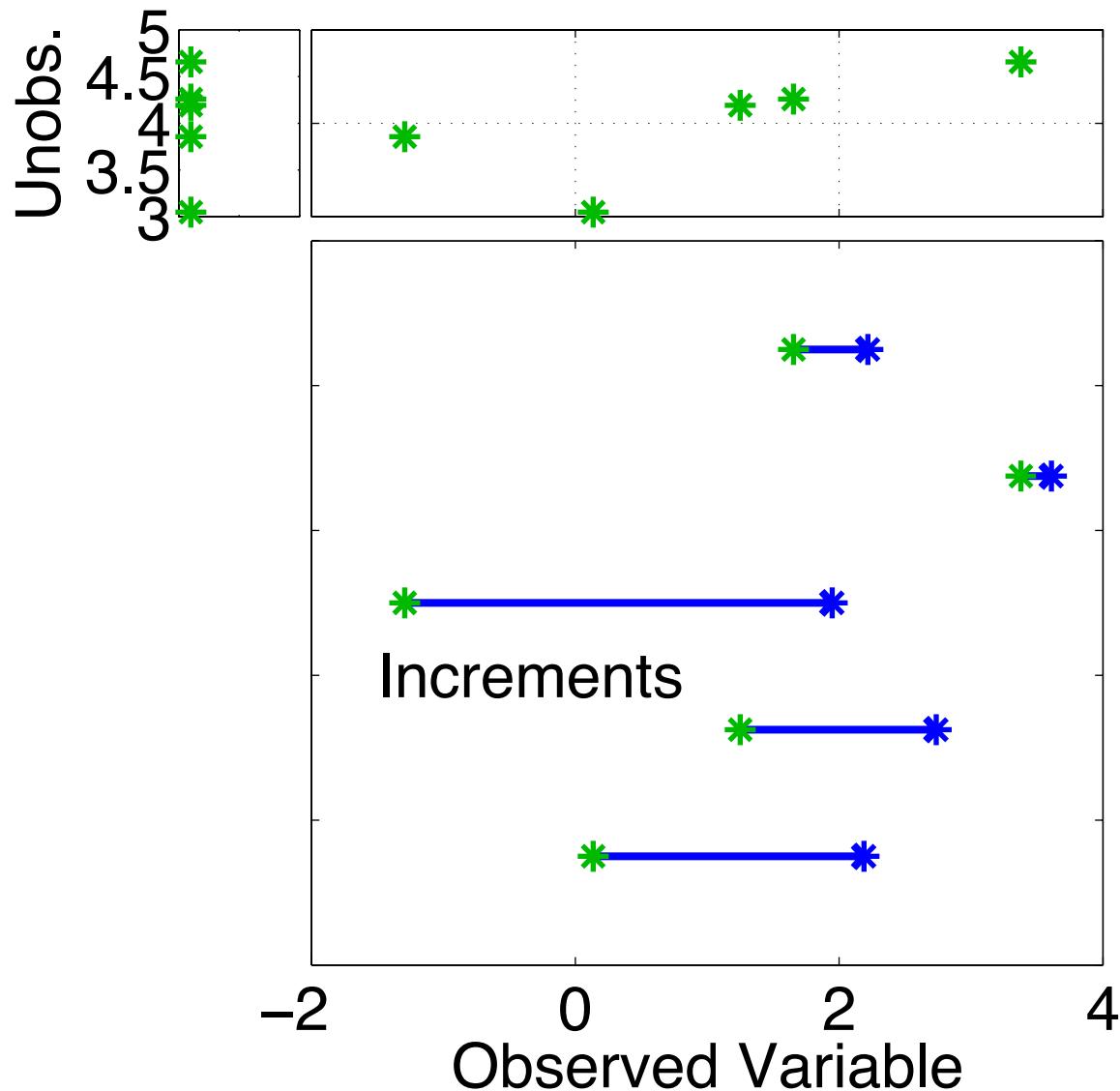


Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

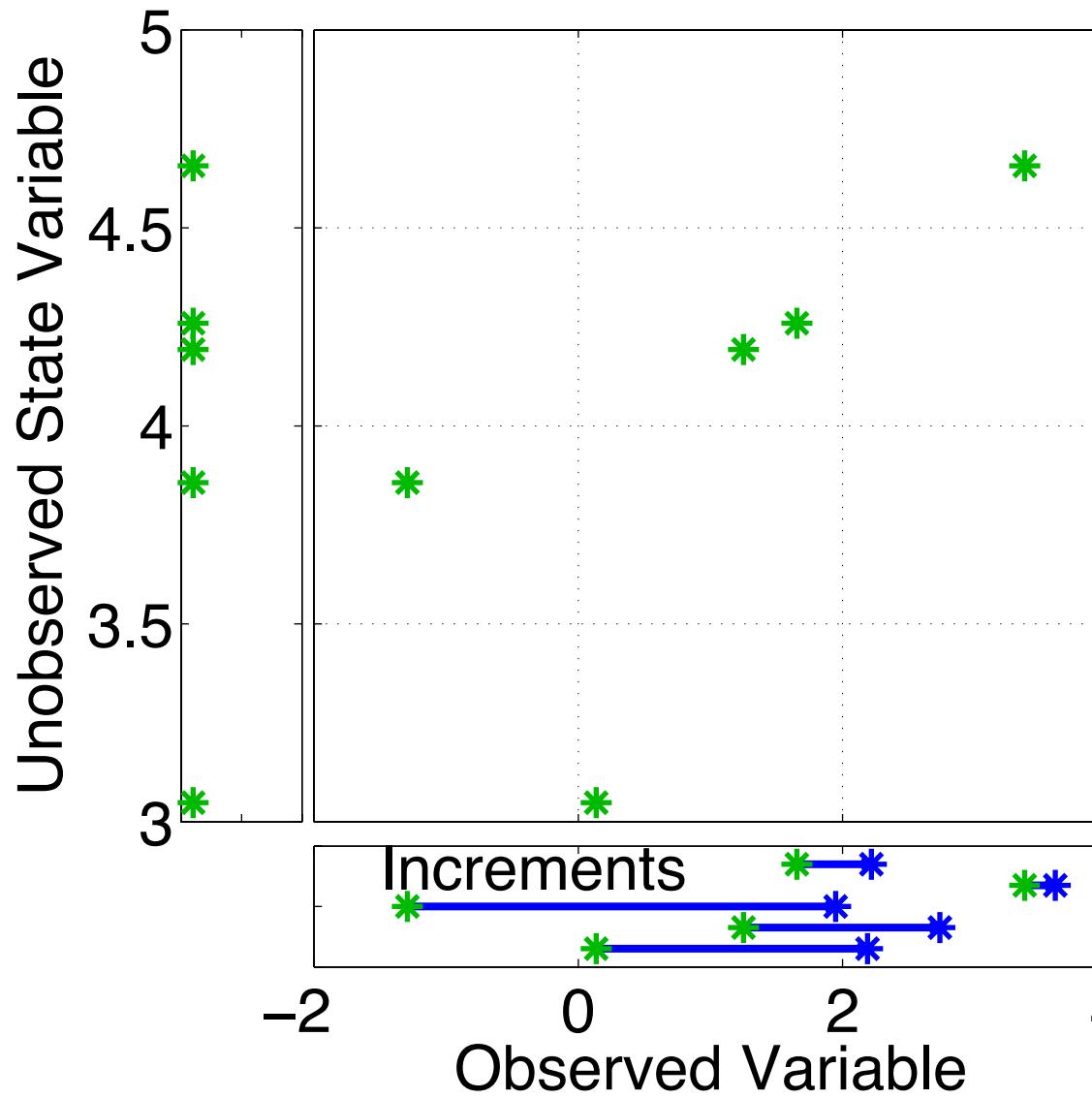
# Ensemble filters: Updating additional prior state variables



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.

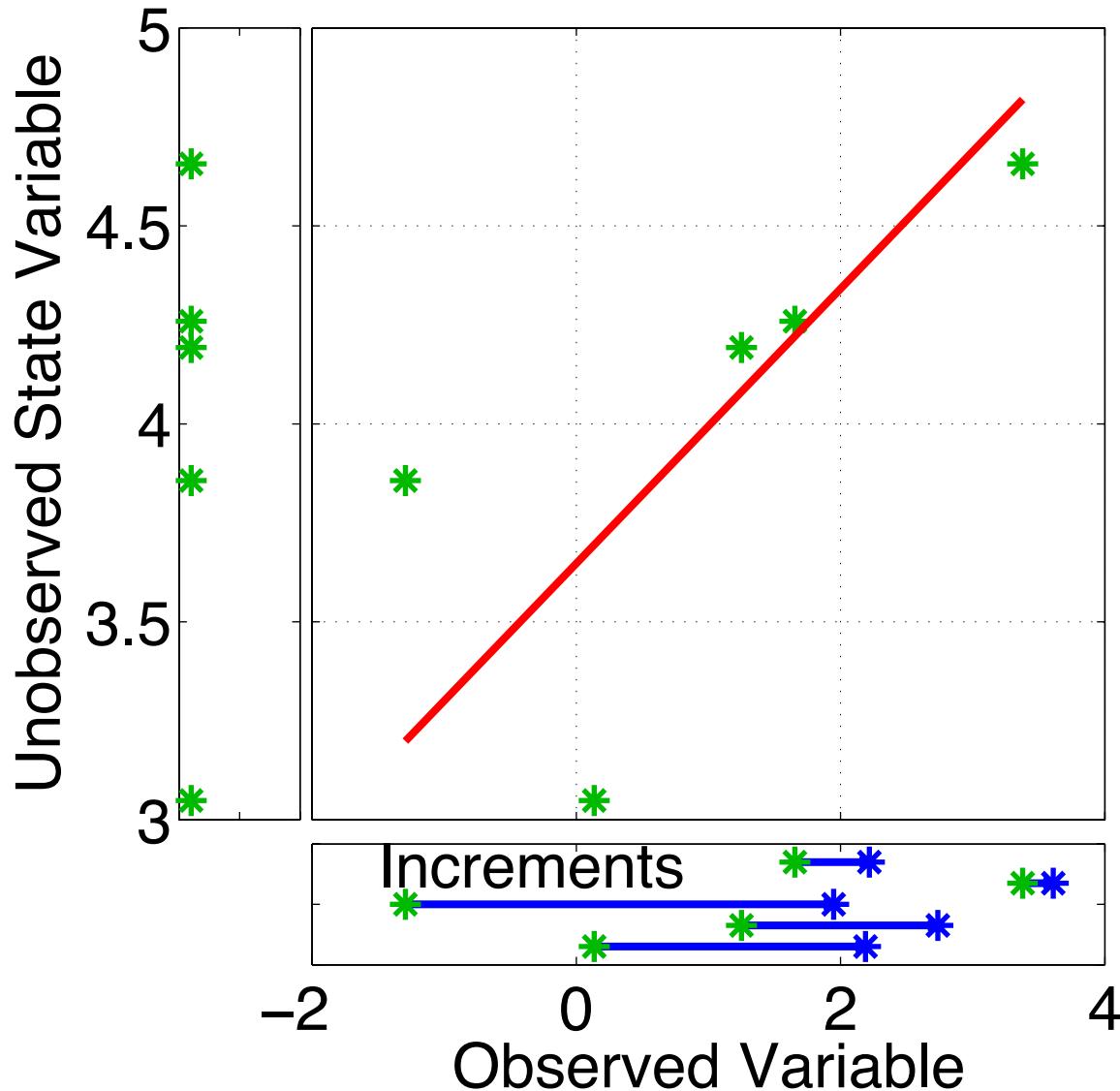
How should the unobserved variable be impacted?

1<sup>st</sup> choice: least squares

Equivalent to linear regression.

Same as assuming binormal prior.

# Ensemble filters: Updating additional prior state variables



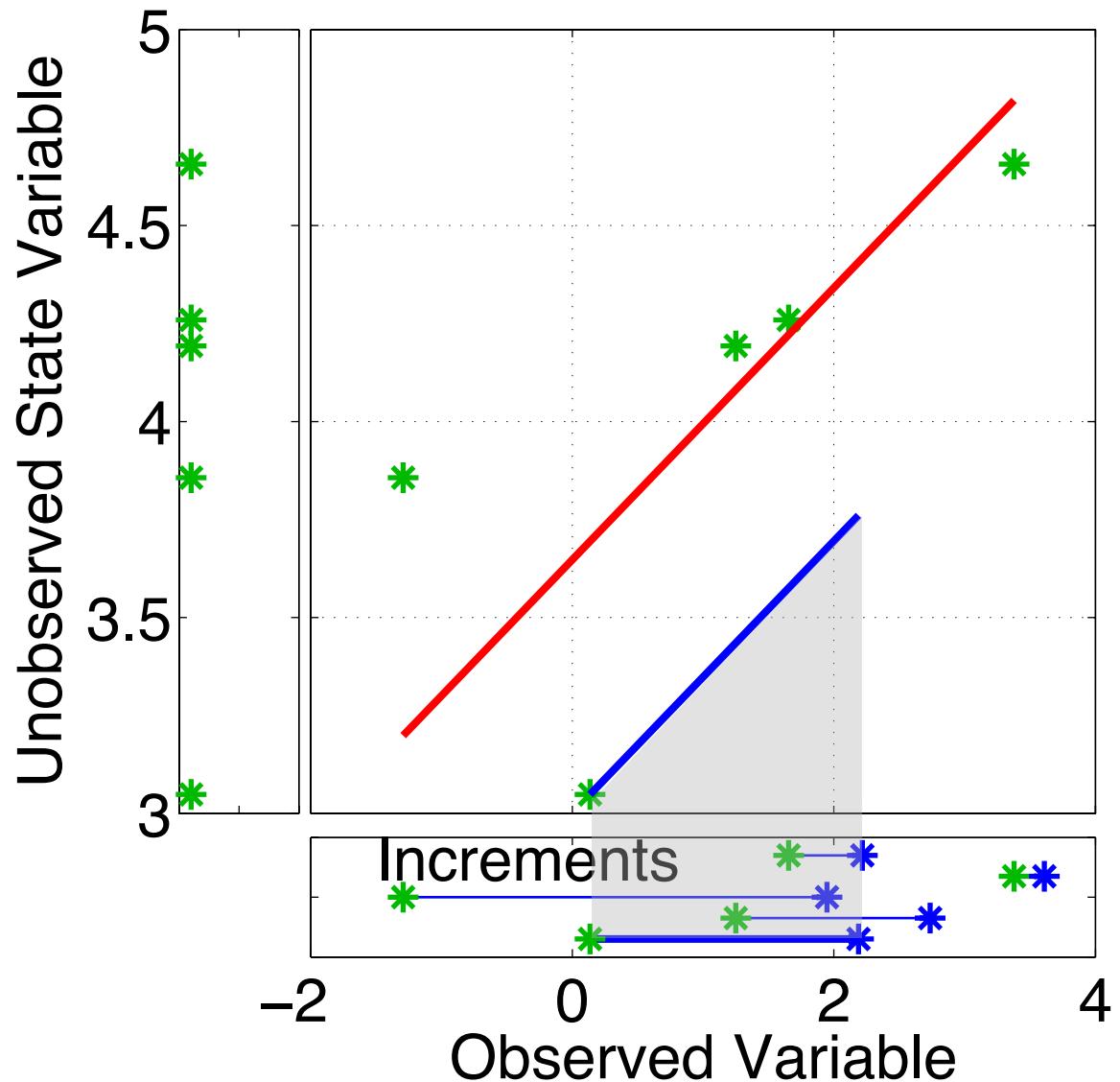
Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

1<sup>st</sup> choice: least squares

Begin by finding **least squares fit**.

# Ensemble filters: Updating additional prior state variables

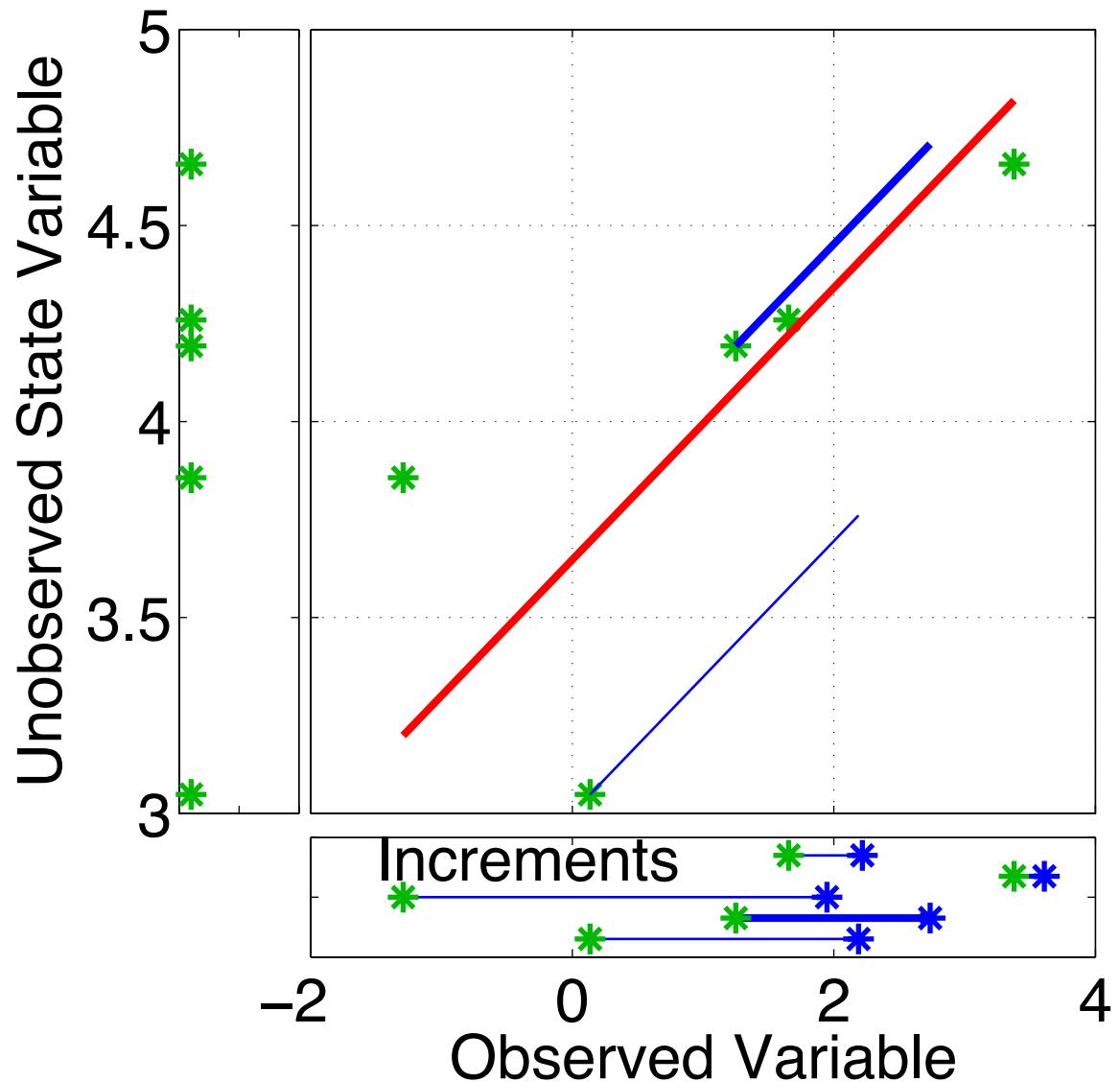


Have joint prior distribution of two variables.

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

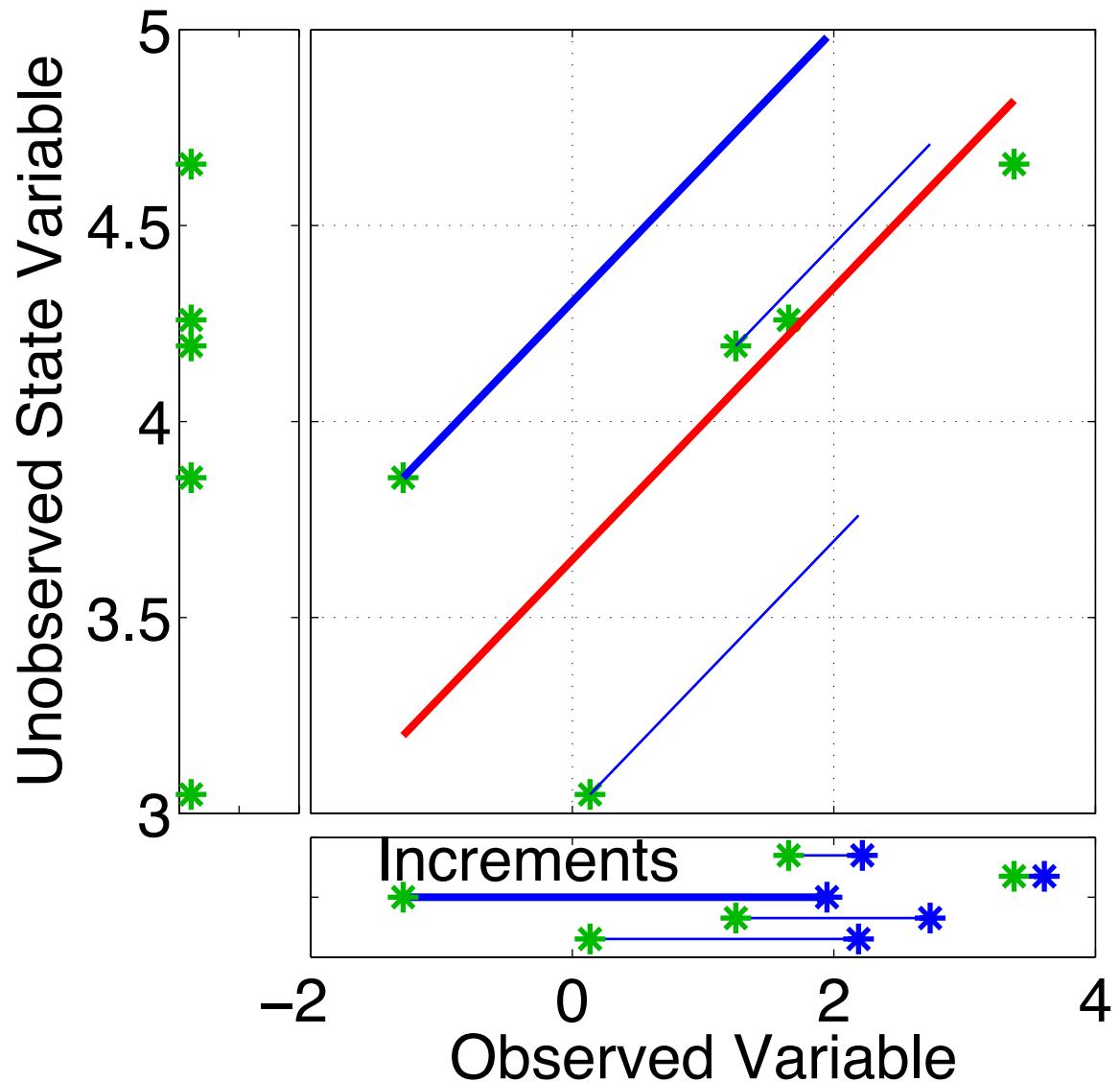


Have joint prior distribution of two variables.

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

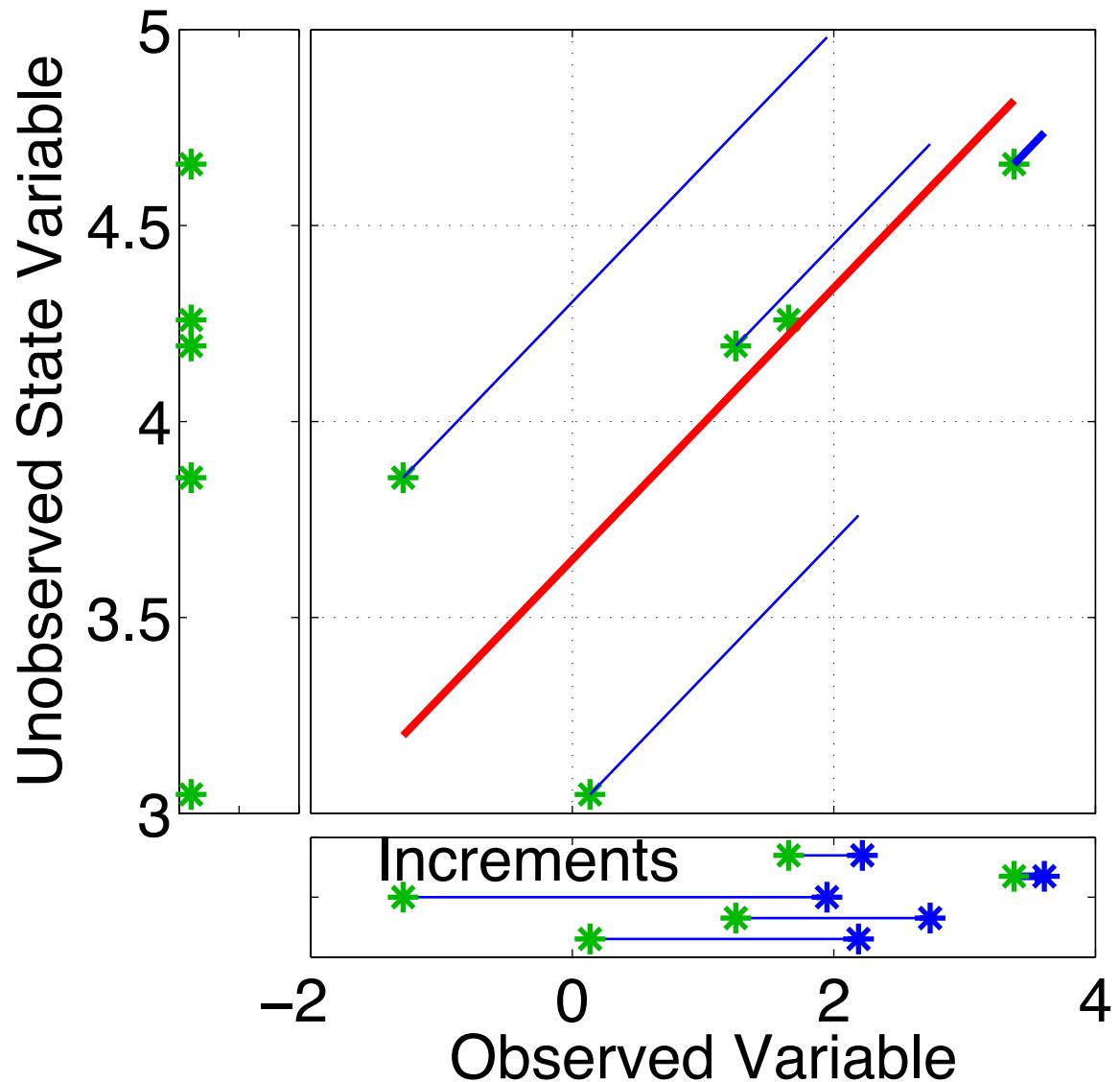


Have joint prior distribution of two variables.

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

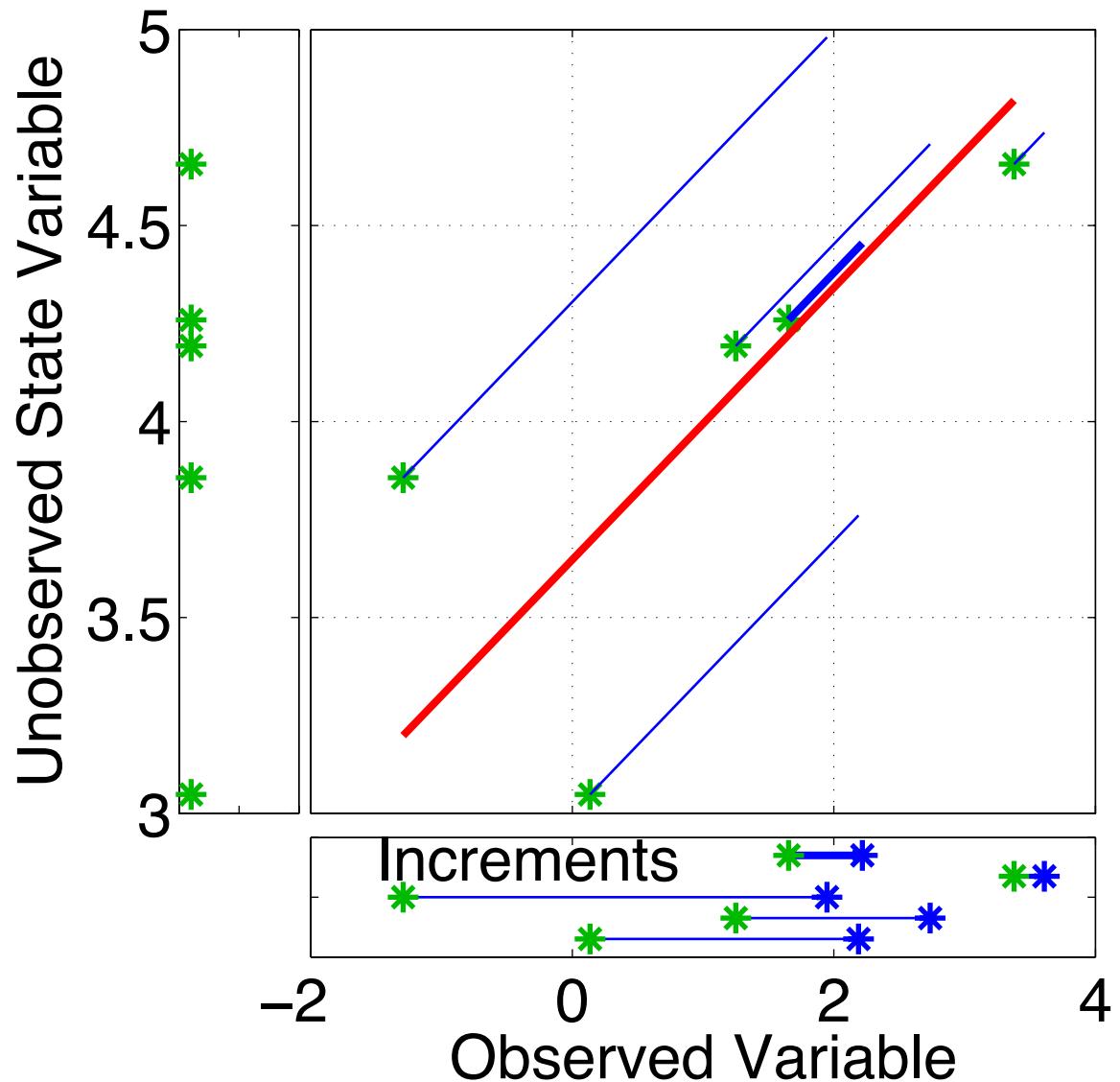


Have joint prior distribution of two variables.

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

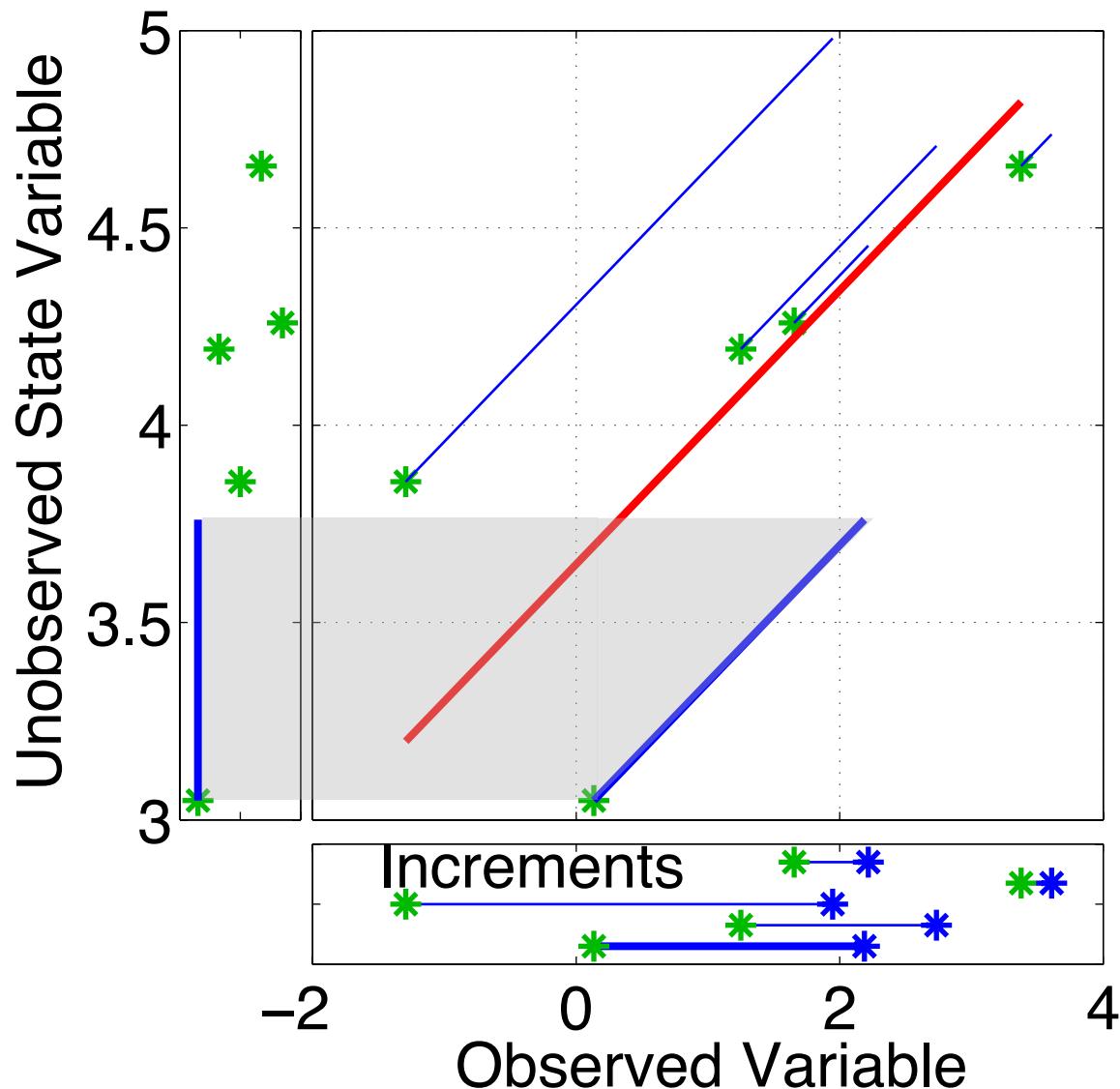


Have joint prior distribution of two variables.

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

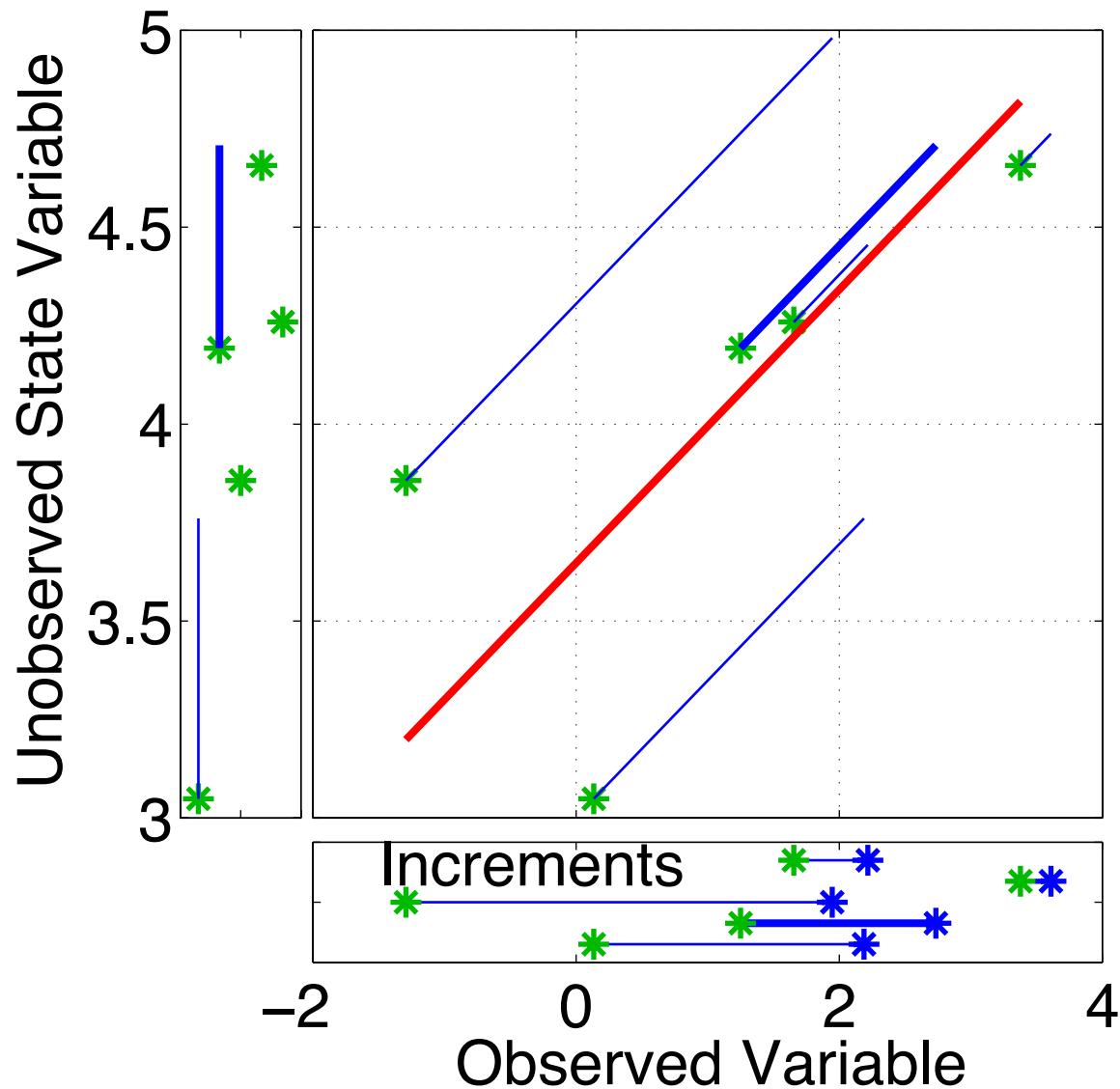


Have joint prior distribution of two variables.

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

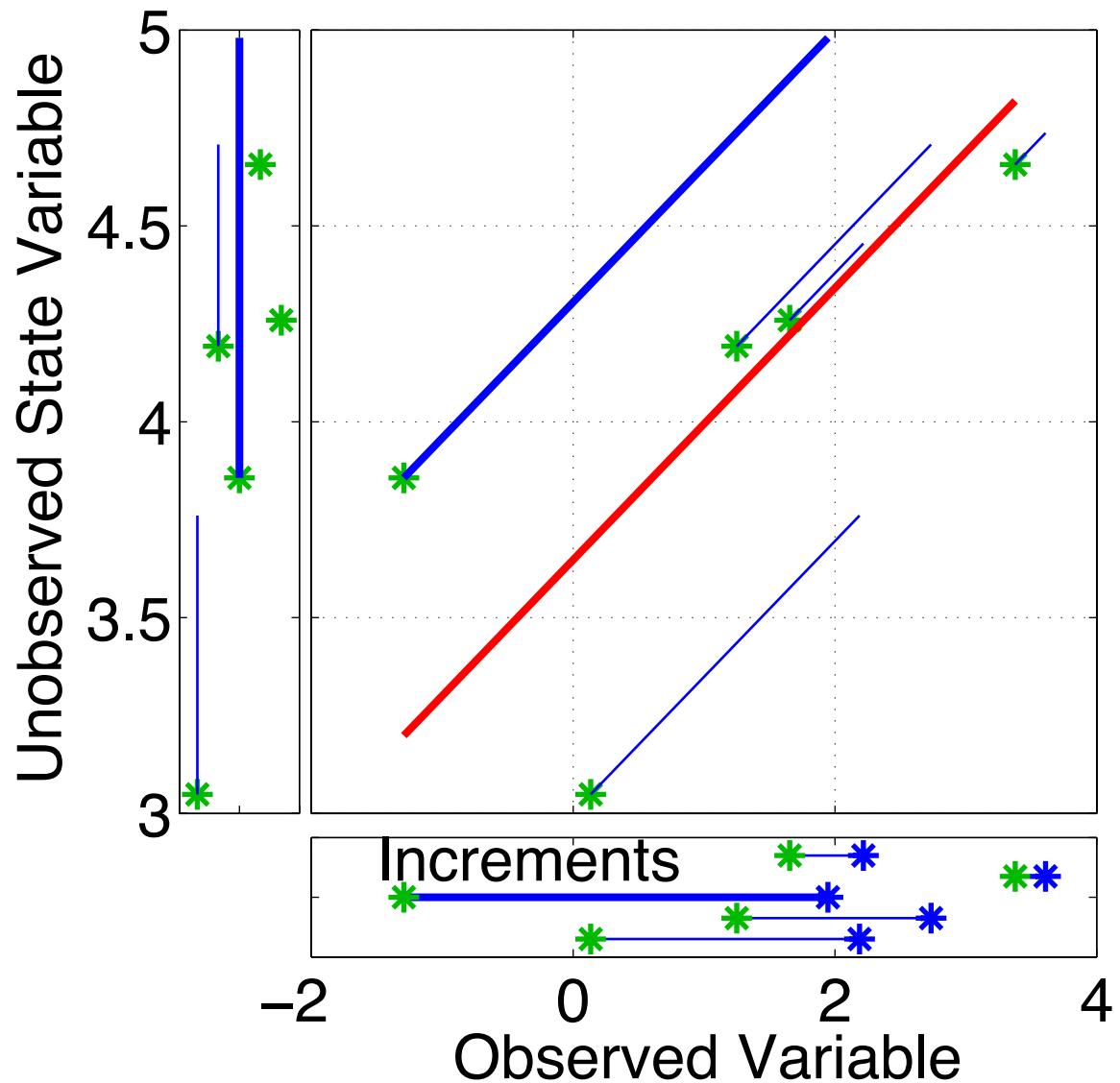


Have joint prior distribution of two variables.

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

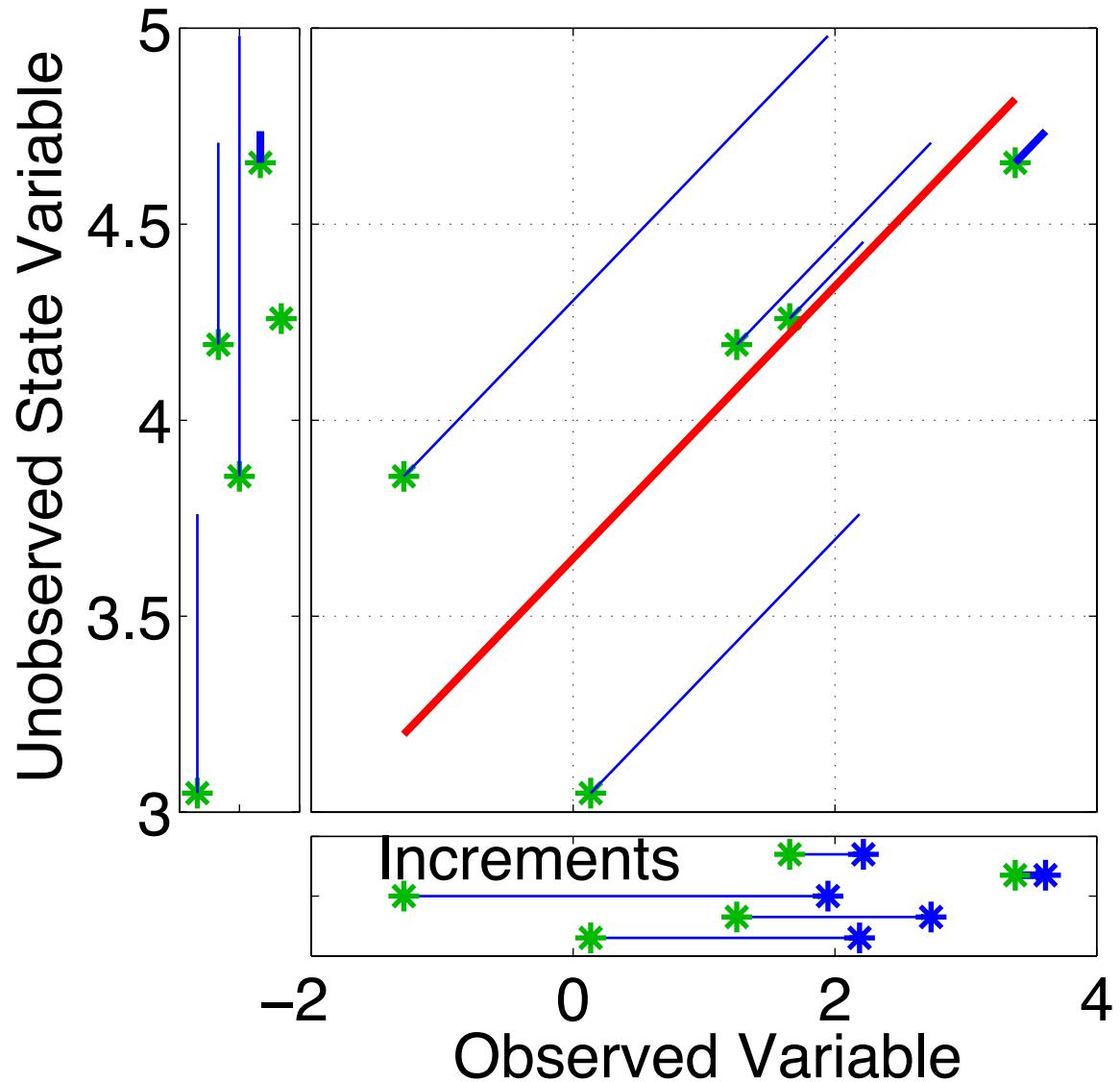


Have joint prior distribution of two variables.

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

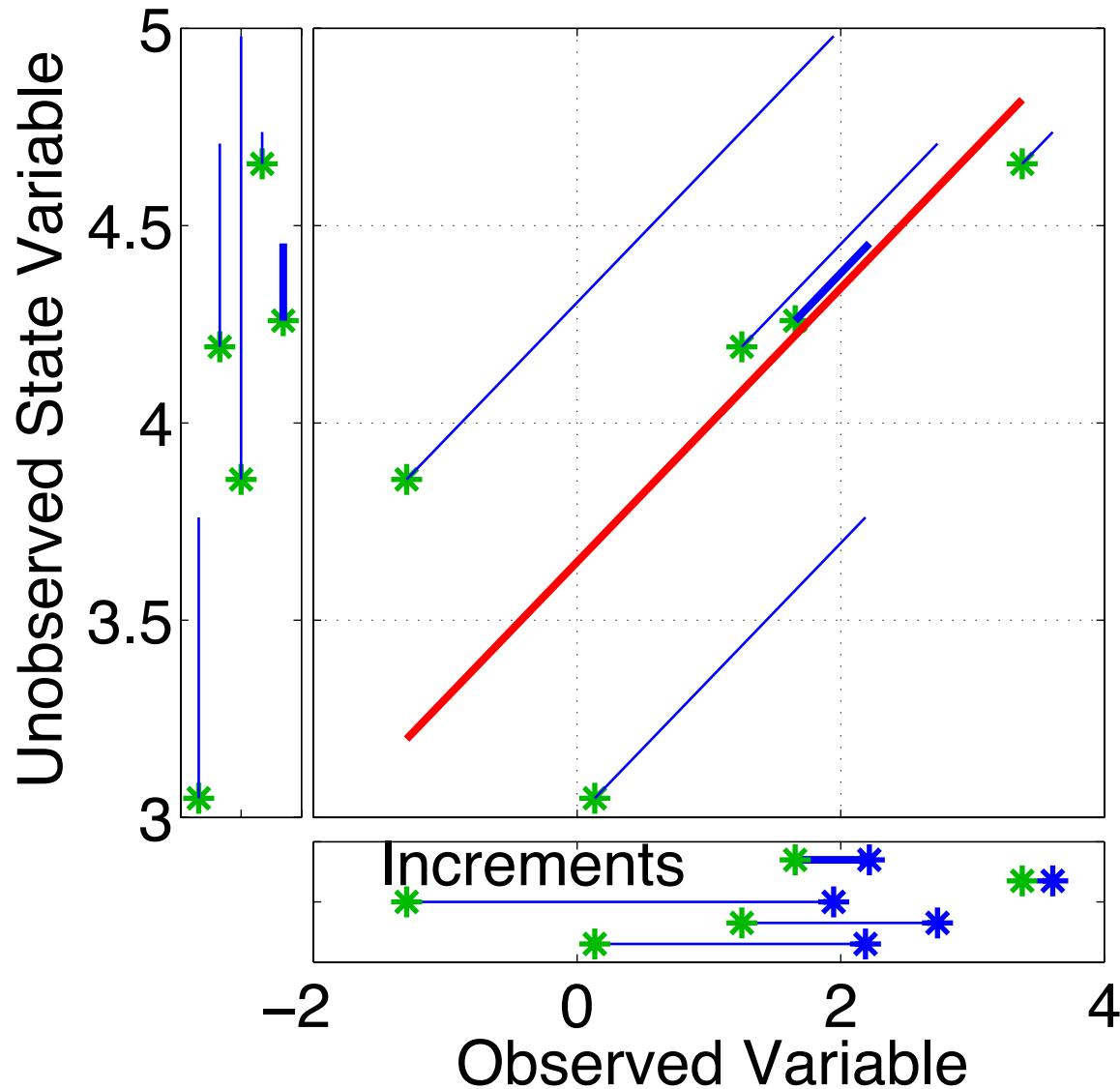


Have joint prior distribution of two variables.

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

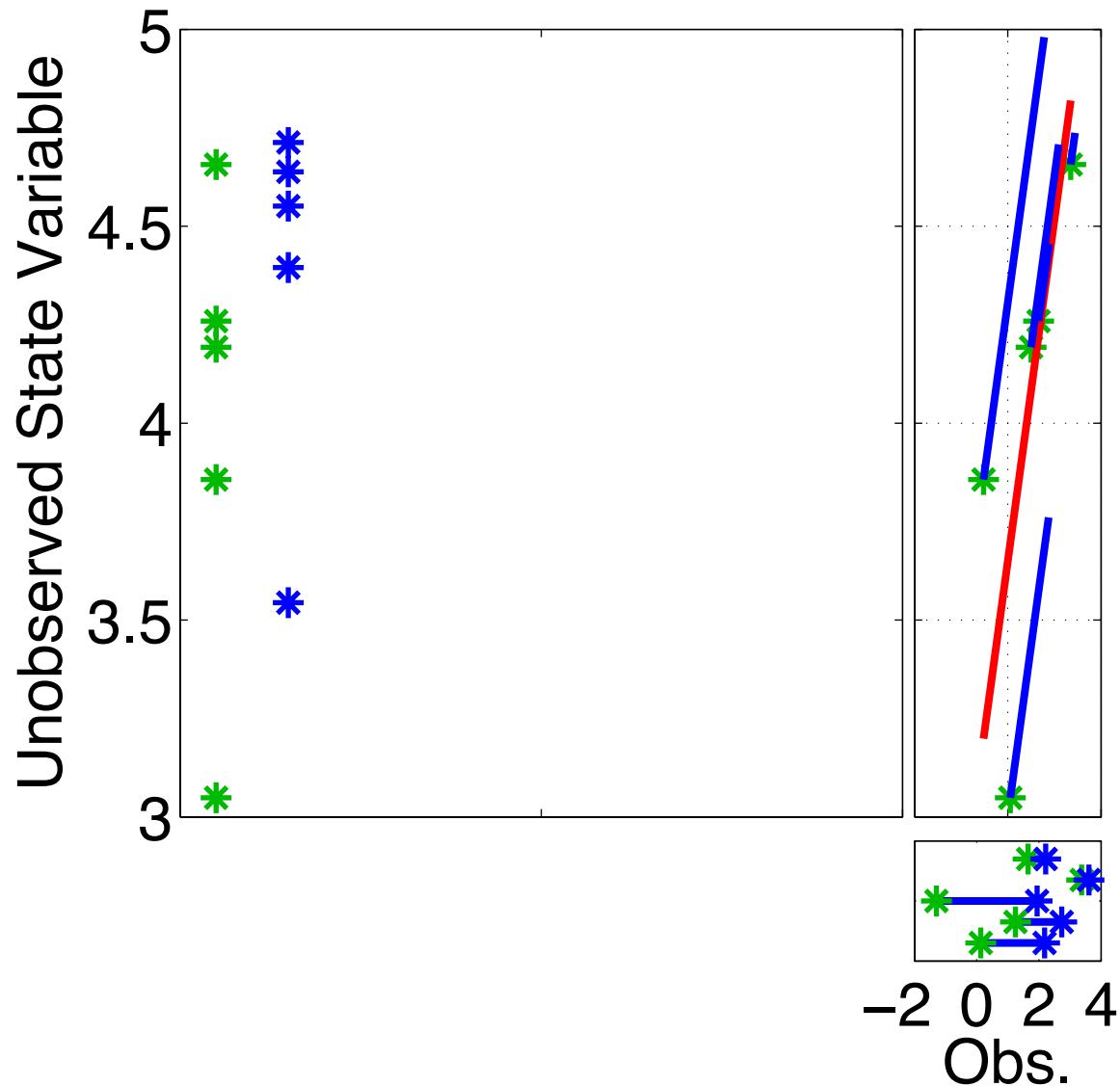


Have joint prior distribution of two variables.

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

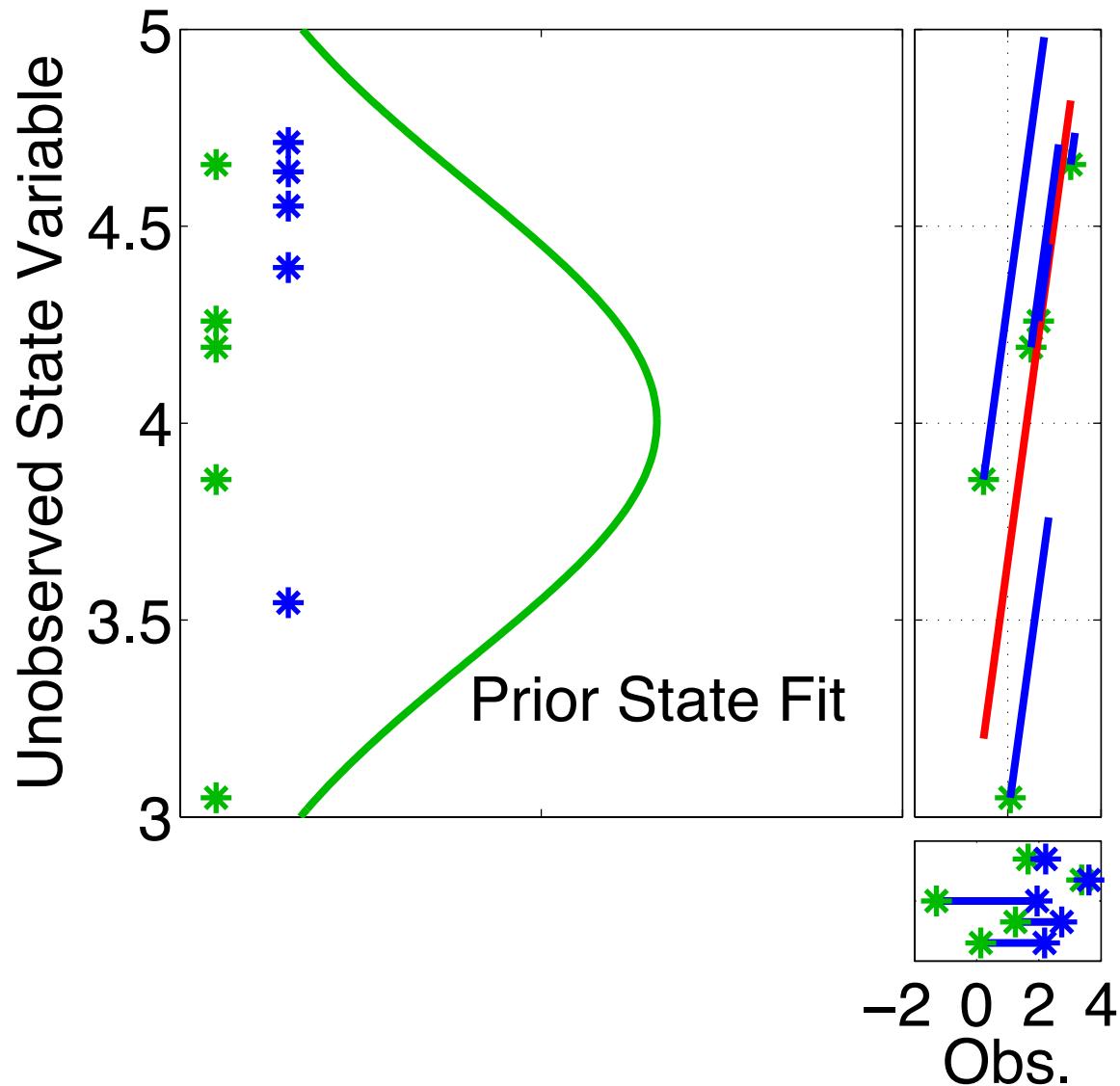
Then projecting from joint space onto unobserved priors.

# Ensemble filters: Updating additional prior state variables



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

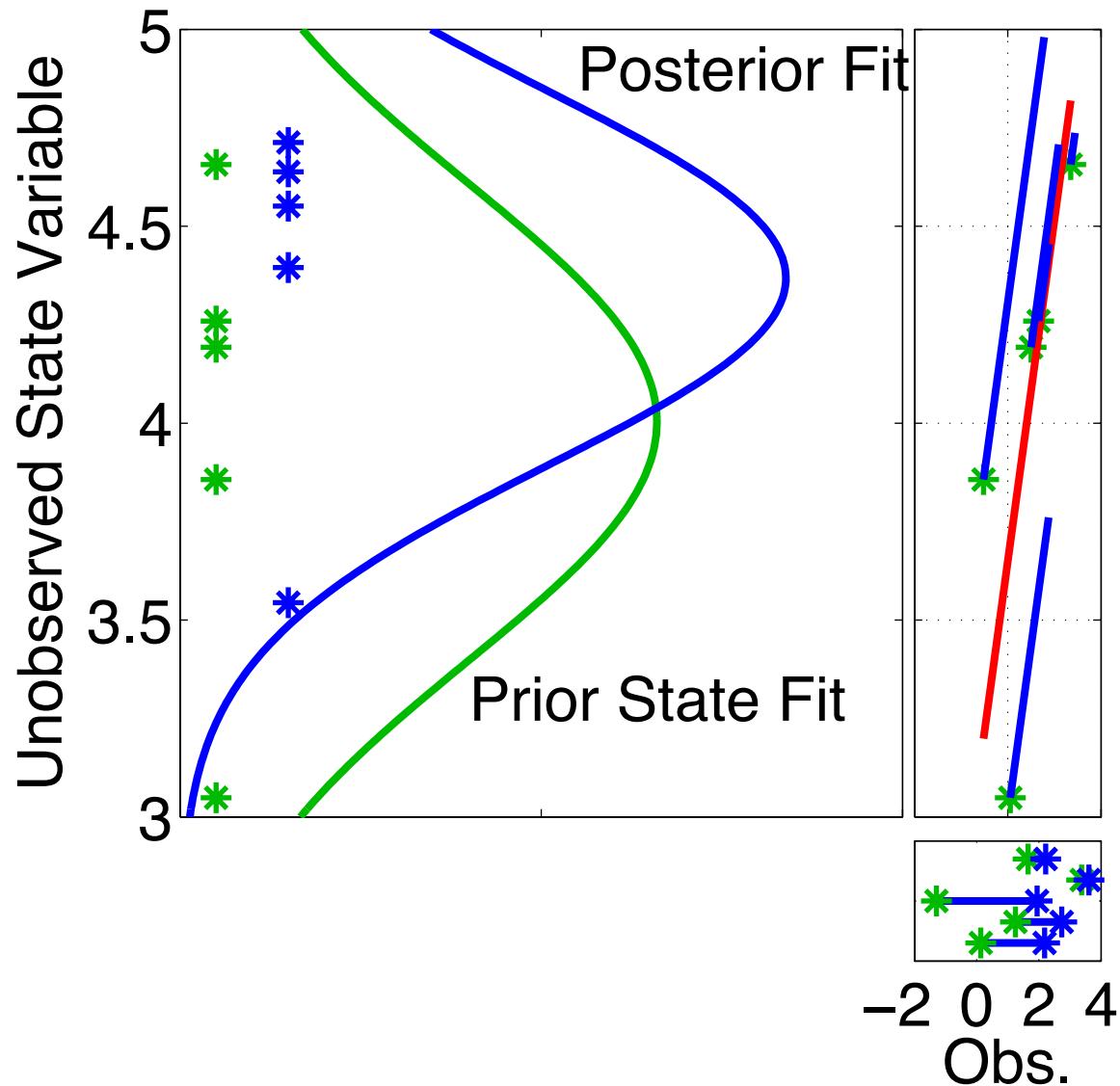
# Ensemble filters: Updating additional prior state variables



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

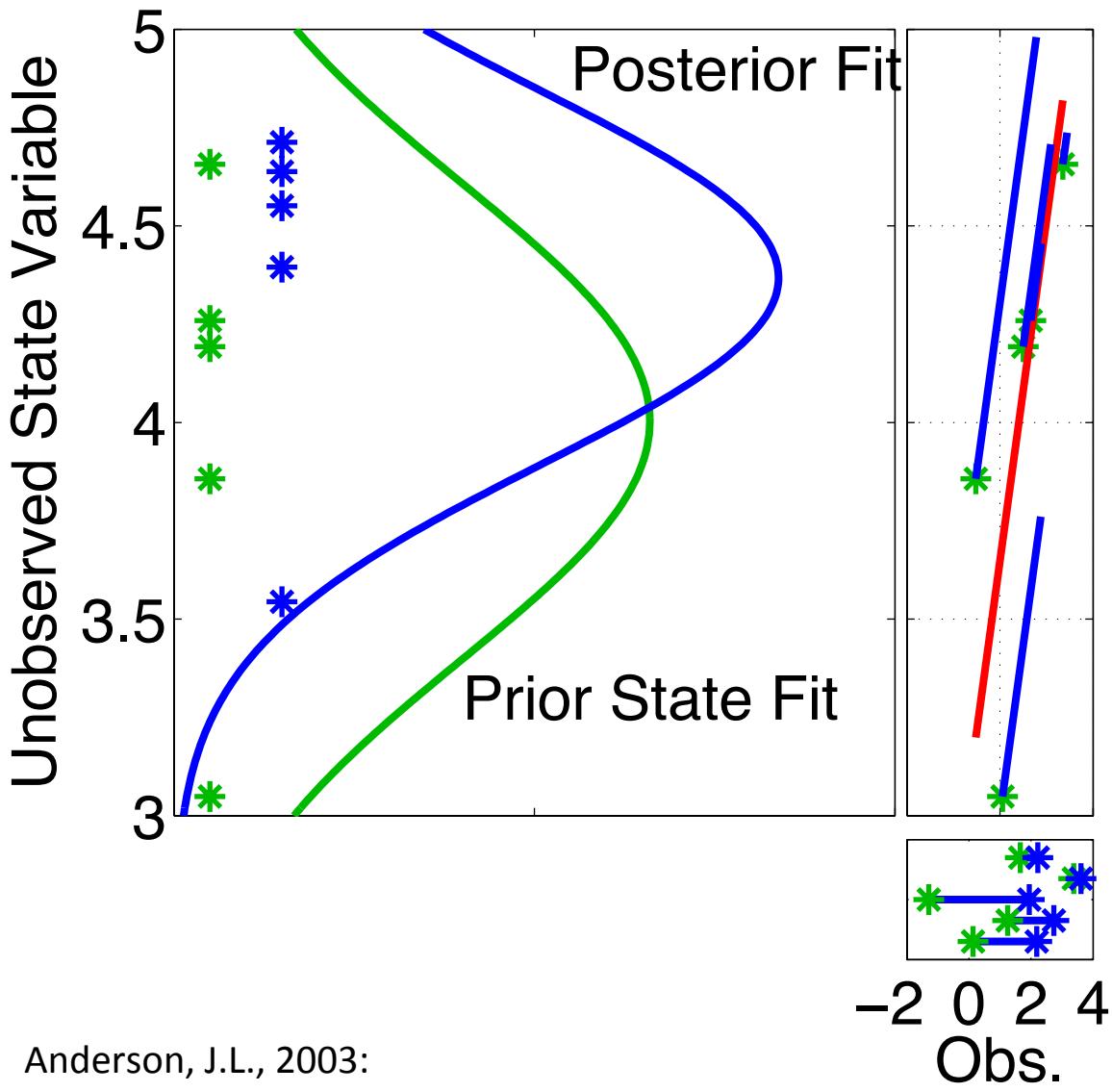


Now have an updated (posterior) ensemble for the unobserved variable.

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



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

**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.

# Multivariate assimilation with DART:

The basic regression code is trivial:

However, DART advanced options can obscure the code.

See *assimilation\_code/modules/assimilation/assim\_tools\_mod.f90*  
*subroutine update\_from\_obs\_inc.*

To generate output from a multivariate Lorenz\_63 experiment:

Run *./filter* in *models/lorenz\_63/work*

(the value of *cutoff* is presumed to be large – from Section 3)

Now do matlab diagnostics.

- Does multivariate do better?
- Be sure to record the error values for comparison.
- Can you identify any obvious performance differences?

# Multivariate assimilation in Lorenz\_63:

What happens if not all state variables are observed?

1. Try observing only  $x$  and  $y$  (ignore  $z$  observations from above).

In *models/lorenz\_63/work* edit *input.nml*

```
&filter_nml
...
async                      = 0,
adv_ens_command            = "./advance_model.csh",
obs_sequence_in_name       = "obs_seq.out",
obs_sequence_out_name      = "obs_seq.final",
...

```

*Change to obs\_seq.out.xy* ←

Execute *./filter* to produce new assimilation.

Look at the error statistics and time series with matlab.

Record the error and spread values and compare to univariate case.

# Multivariate assimilation in Lorenz\_63:

What happens if not all state variables are observed?

2. Try observing only  $x$  (ignore  $y$  and  $z$  observations from above).

In *models/lorenz\_63/work* edit *input.nml*

```
&filter_nml
...
obs_sequence_in_name      = "obs_seq.out.xy"    ← Change to Obs_seq.out.x
...
...
```

Execute *./filter* to produce new assimilation.

Look at the error statistics and time series with matlab.

Record the error and spread values and compare to univariate case.

What would have happened if we made this into a univariate assimilation?

```
&assim_tools_nml
filter_kind              = 1
cutoff                   = 1000000.0    ← run a test with a small value
                                                ← Change to 0.00001
```

# Multivariate assimilation in Lorenz\_63:

What happens if not all state variables are observed?

3. Try observing only  $z$  (ignore  $x$  and  $y$  observations from above).

In *models/lorenz\_63/work* edit *input.nml*

```
&filter_nml
...
obs_sequence_in_name      = "obs_seq.out.x"           Change to Obs_seq.out.z
...
&assim_tools_nml
...
cutoff                   = 0.001                  Change to 1000000.0
```

Execute *./filter* to produce new assimilation;  
look at the error statistics and time series with matlab.

Record the error and spread values and compare to univariate case.  
Dynamics for  $x$  and  $y$  are symmetric;  $z$  can NOT distinguish them.  
How do we want filter to handle this?  
Does it do what we want in this case?

# DART Tutorial Index to Sections

1. Filtering For a One Variable System
2. The DART Directory Tree
3. DART Runtime Control and Documentation
4. How should observations of a state variable impact an unobserved state variable?  
Multivariate assimilation.
5. Comprehensive Filtering Theory: Non-Identity Observations and the Joint Phase Space
6. Other Updates for An Observed Variable
7. Some Additional Low-Order Models
8. Dealing with Sampling Error
9. More on Dealing with Error; Inflation
10. Regression and Nonlinear Effects
11. Creating DART Executables
12. Adaptive Inflation
13. Hierarchical Group Filters and Localization
14. Quality control
15. DART Experiments: Control and Design
16. Diagnostic Output
17. Creating Observation Sequences
18. Lost in Phase Space: The Challenge of Not Knowing the Truth
19. DART-Compliant Models and Making Models Compliant
20. Model Parameter Estimation
21. Observation Types and Observing System Design
22. Parallel Algorithm Implementation
23. Location module design (not available)
24. Fixed lag smoother (not available)