Data assimilation

How to merge models and data optimally

Axel Hutt, September 2021

motivation

basic methods

Kalman filter

prediction and verification

motivation

ideas cycling

basic methods

Kalman filter

prediction and verification

Imperial Trans-Antarctic Expedition (1914-1917)

Ernest Shackleton

(Anglo-Irish Explorer)

Imperial Trans-Antarctic Expedition (1914-1917)

Ernest Shackleton

(Anglo-Irish Explorer)

his ship *Endurance*

Imperial Trans-Antarctic Expedition (1914-1917)

Ernest Shackleton

his ship *Endurance*

Imperial Trans-Antarctic Expedition (1914-1917)

Ernest Shackleton

his ship *Endurance*

- trip of 800 miles = 1290 km

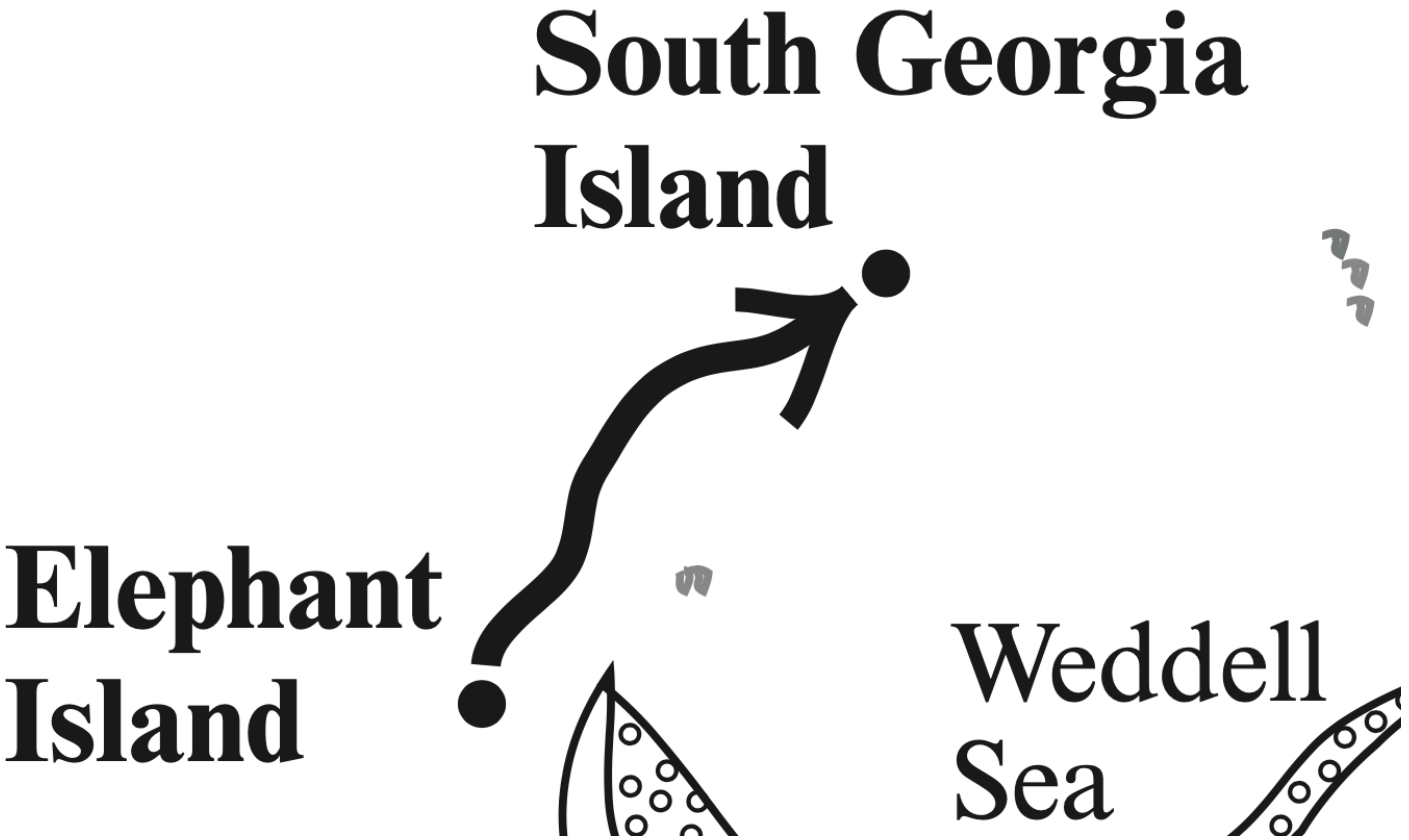
boat for 800mile - trip to South Georgia

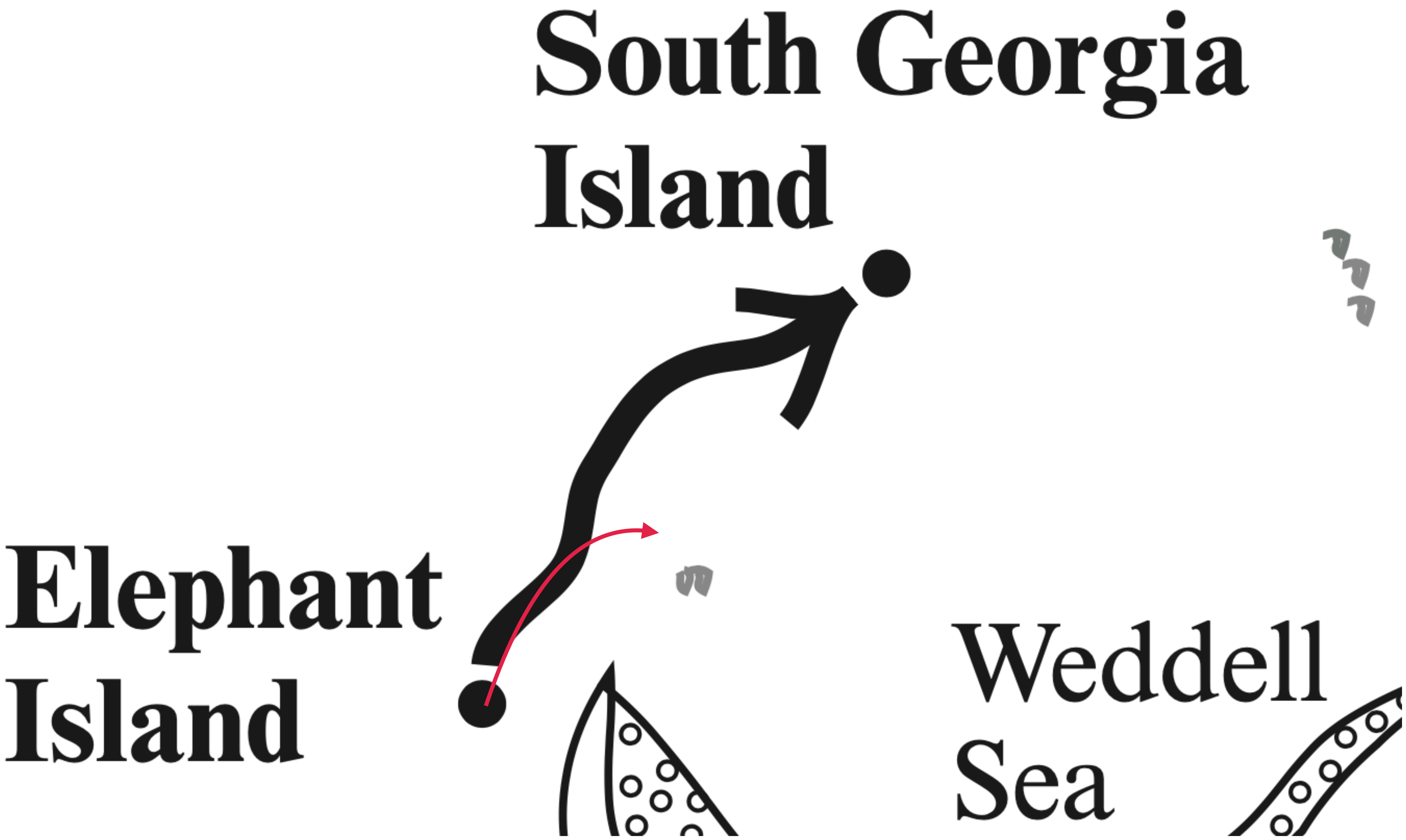
- trip of 800 miles = 1290 km
- navigation equipment : map for orientation

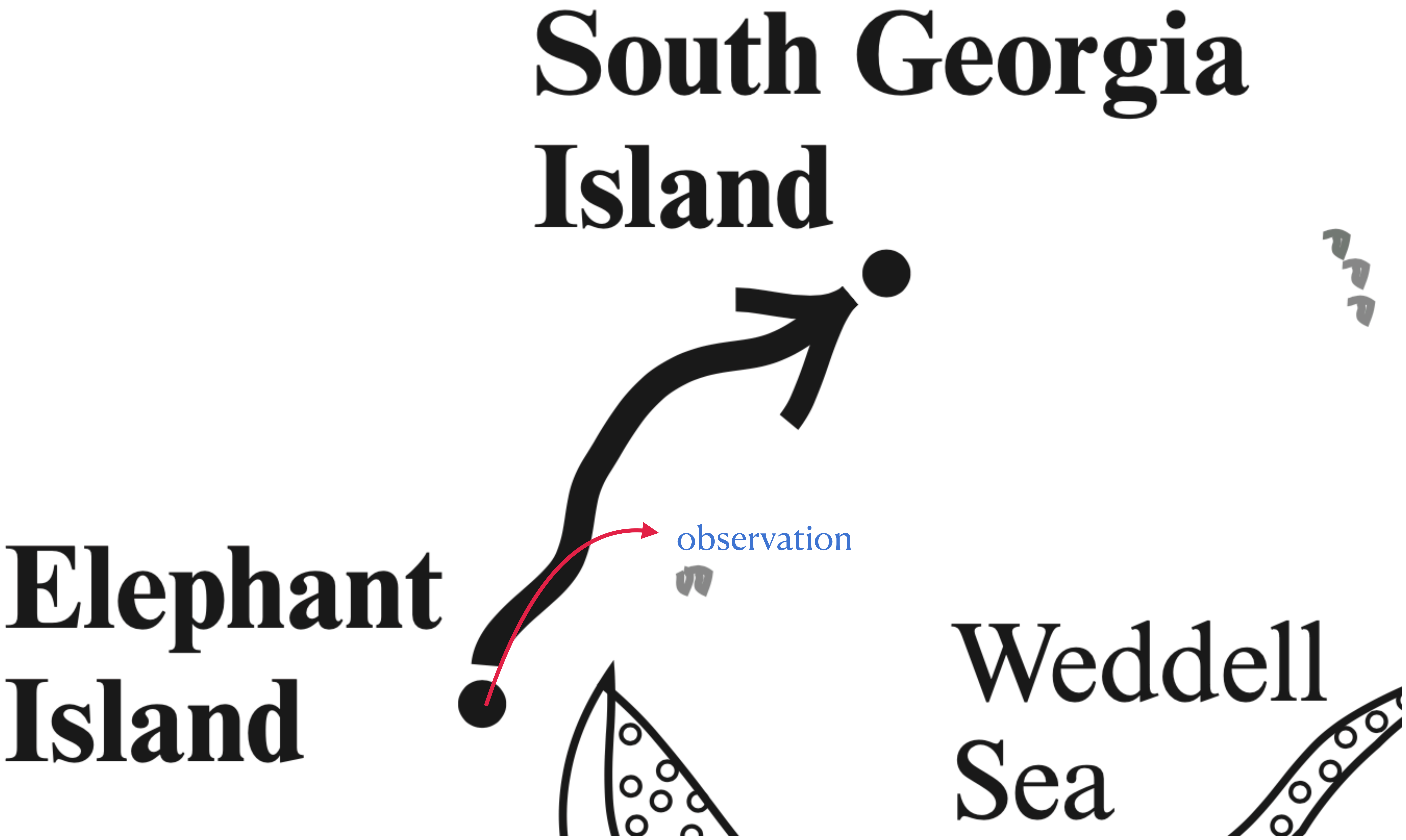
boat for 800mile - trip to South Georgia

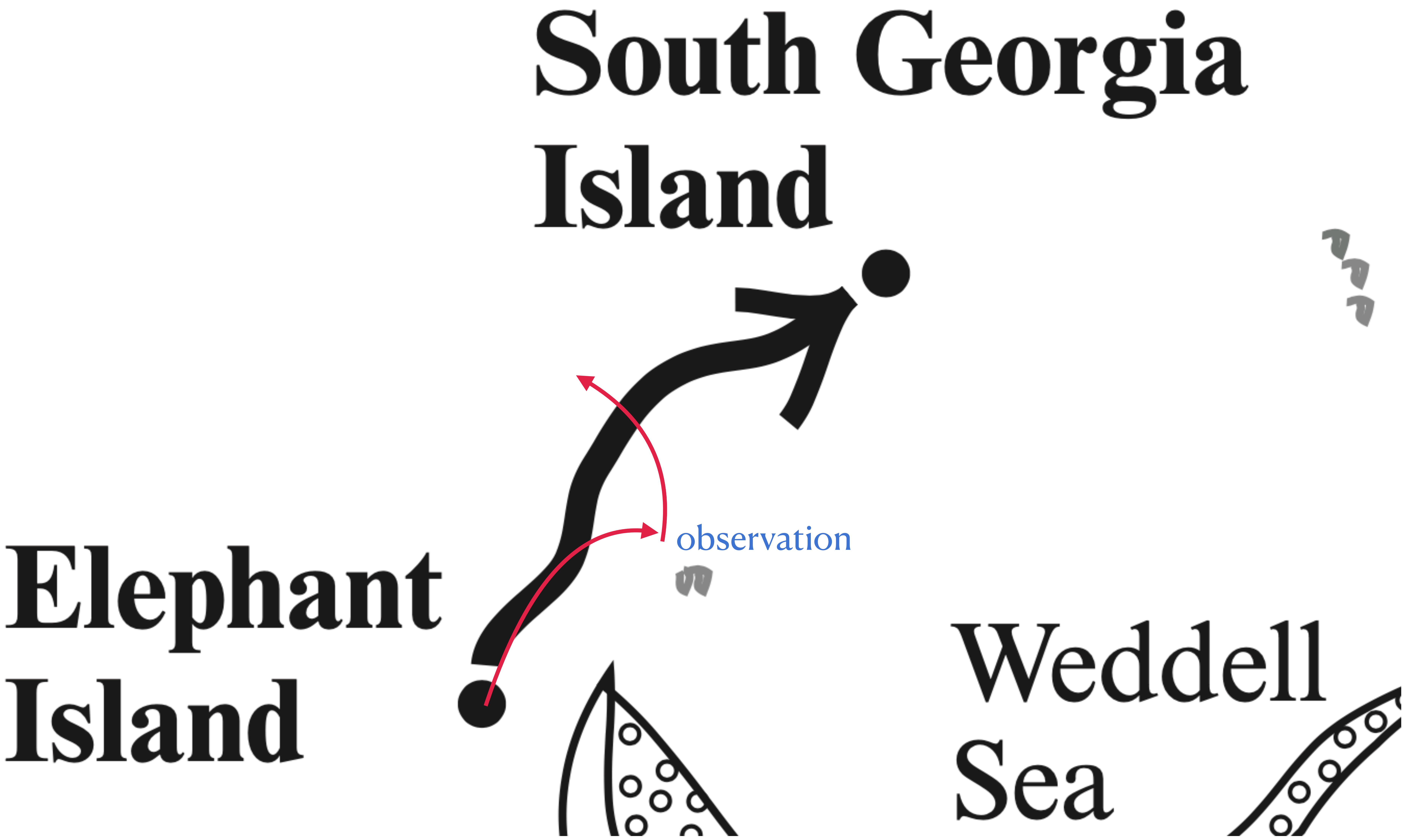
- trip of 800 miles = 1290 km
- navigation equipment : map for orientation
- equipment : sextant for position observation

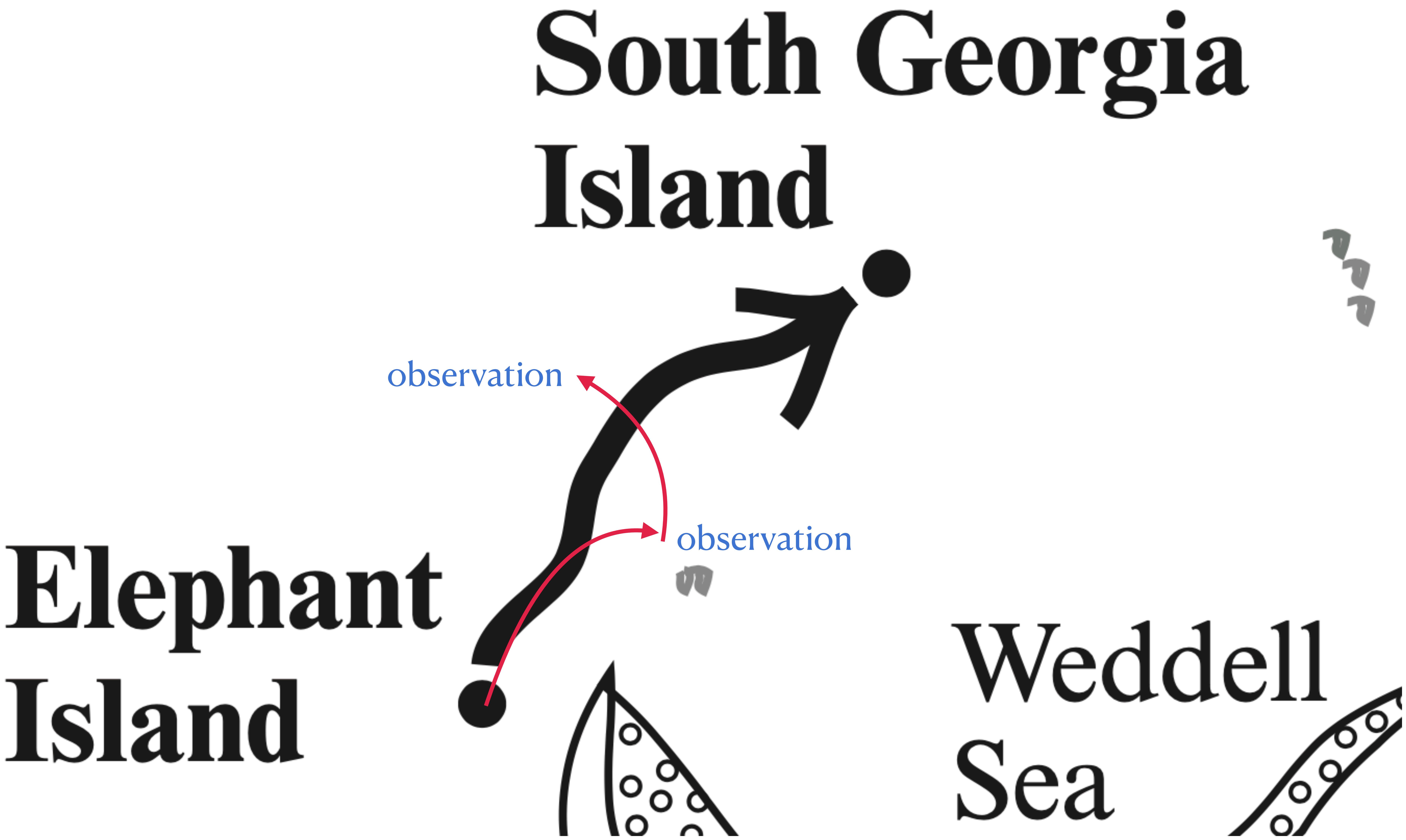
boat for 800mile - trip to South Georgia


South Georgia Island
Elephant Island
Weddell Sea

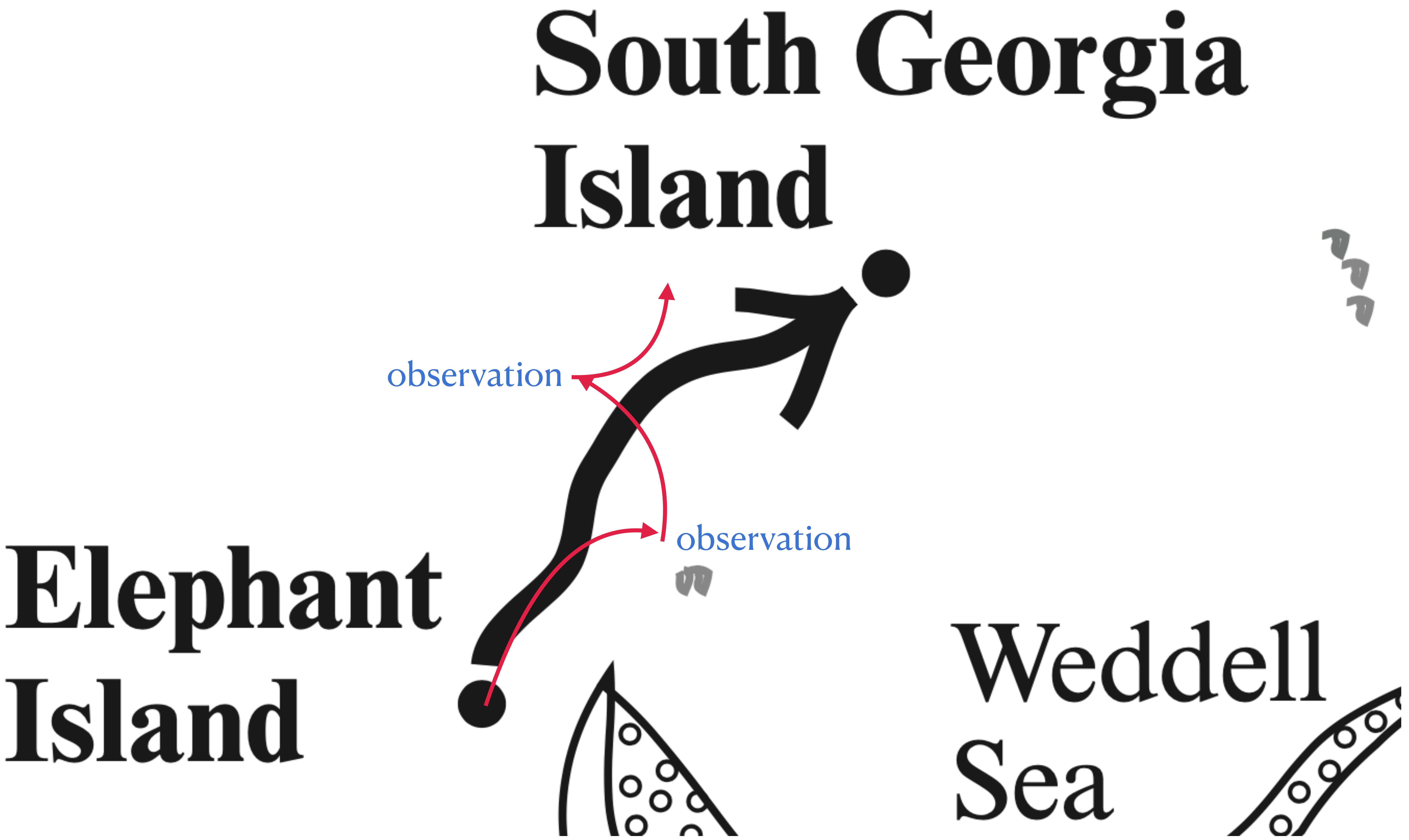

South Georgia Island
Elephant Island
Weddell Sea

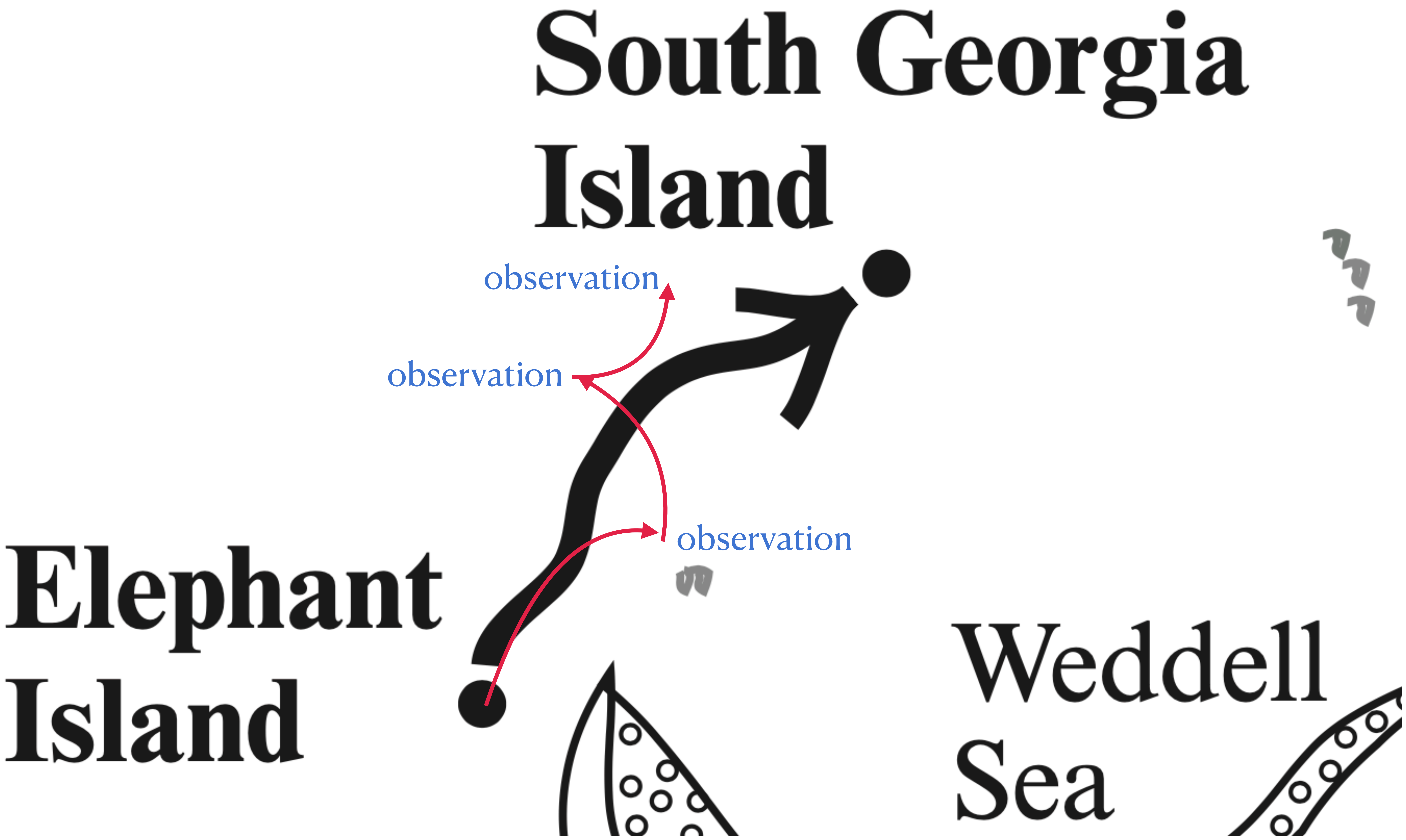

South Georgia Island
Elephant Island
observation
Weddell Sea

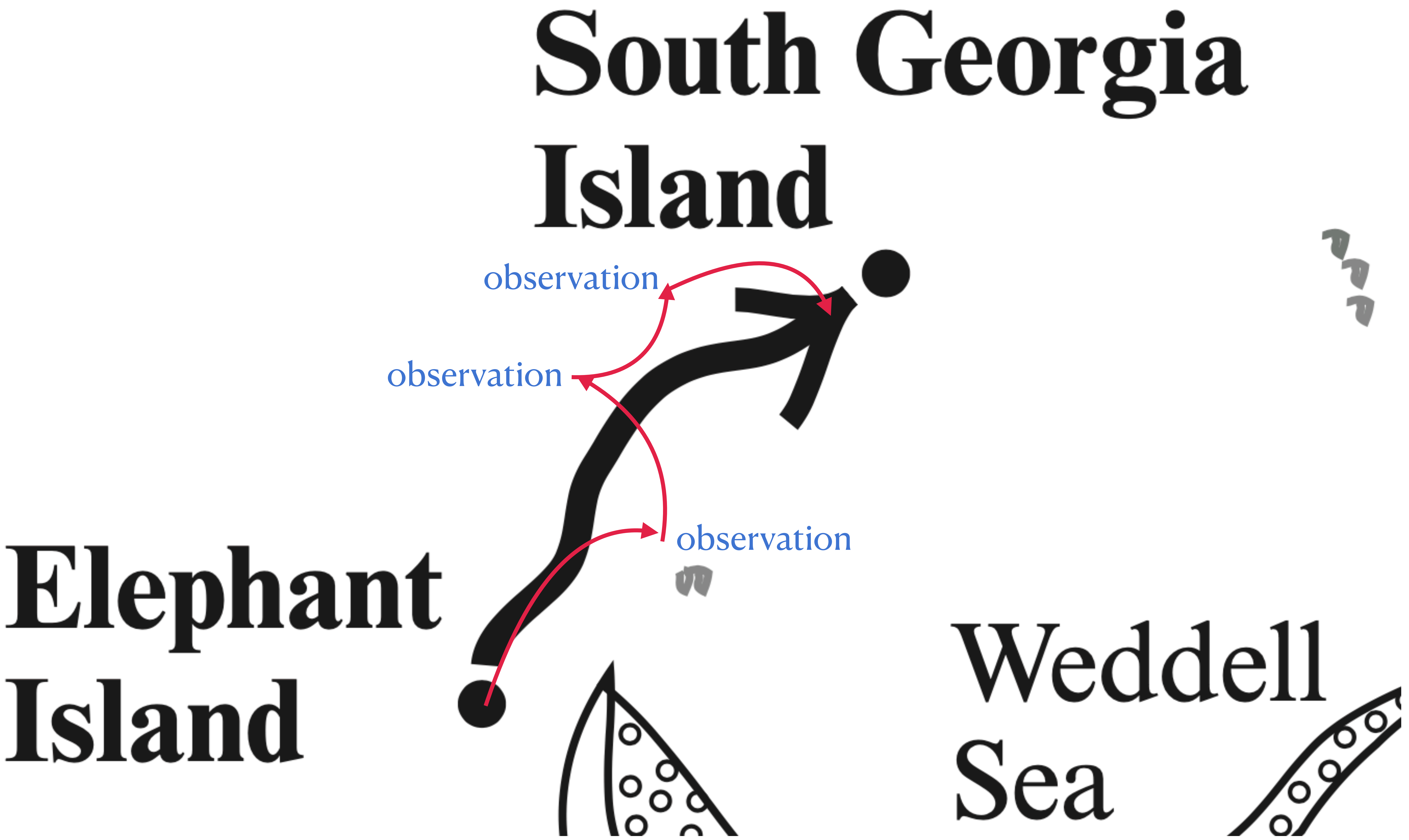

South Georgia Island
observation
Elephant Island
Weddell Sea

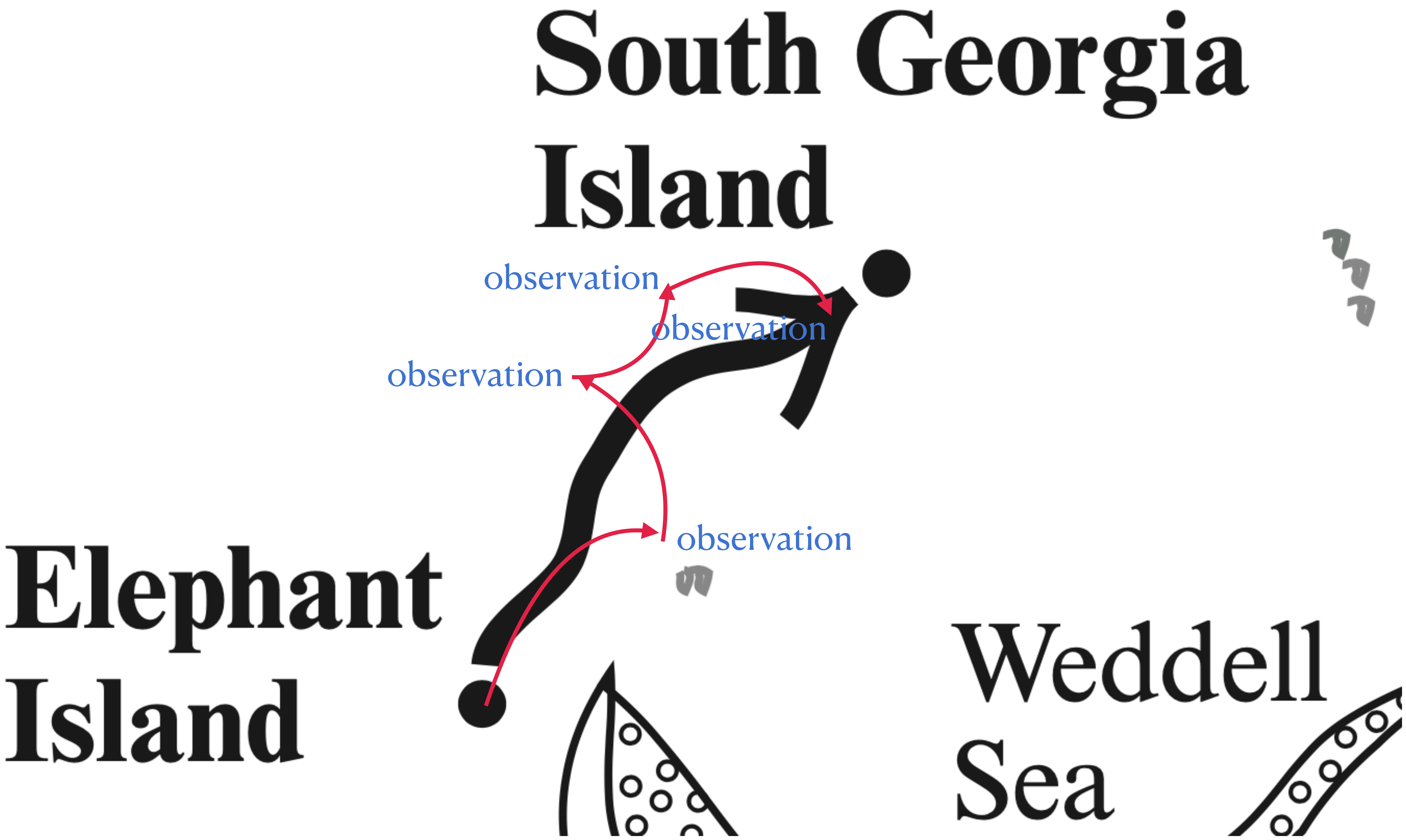


South Georgia Island
observation
observation
Elephant Island
Weddell Sea

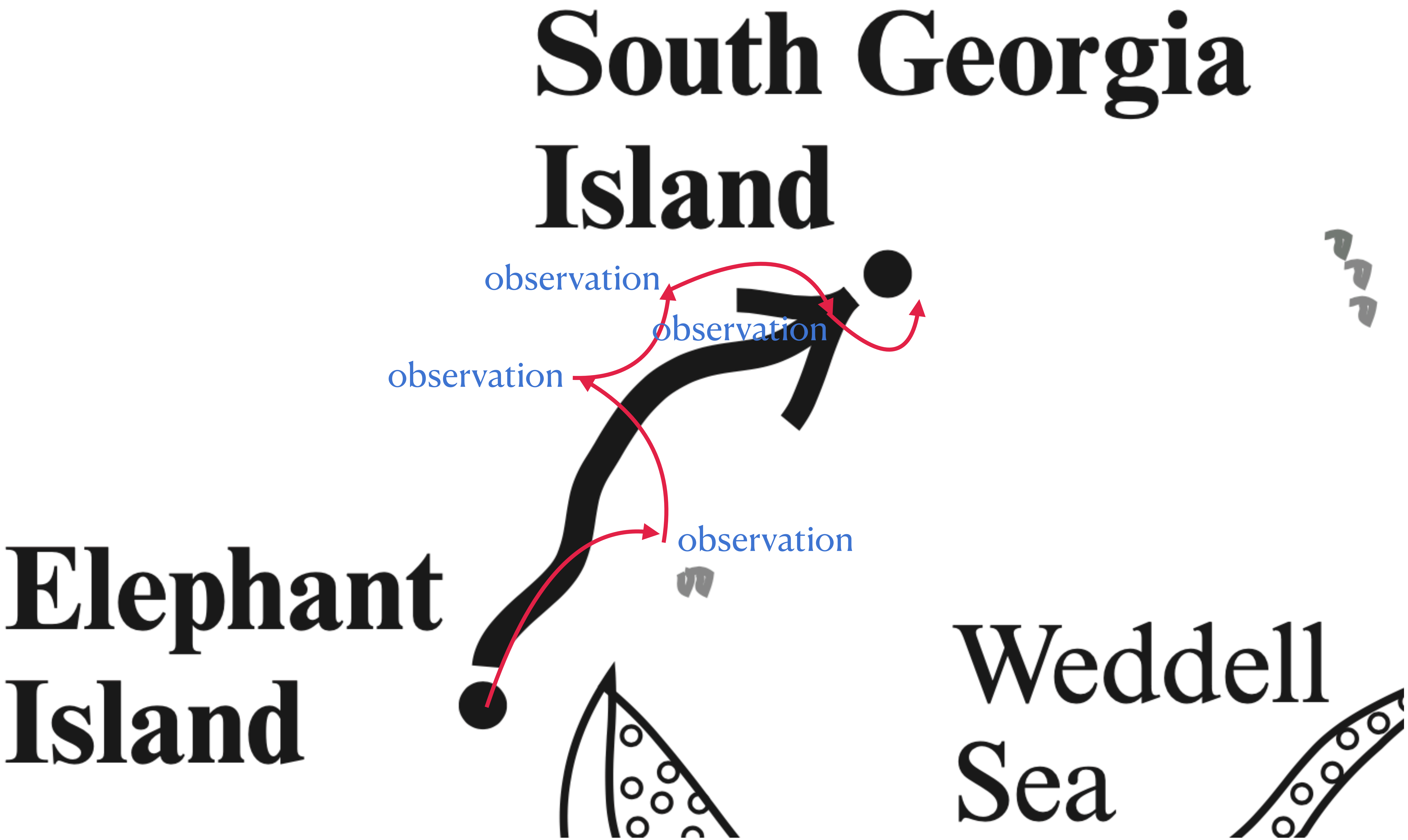

South Georgia Island
observation
observation
Elephant Island
Weddell Sea

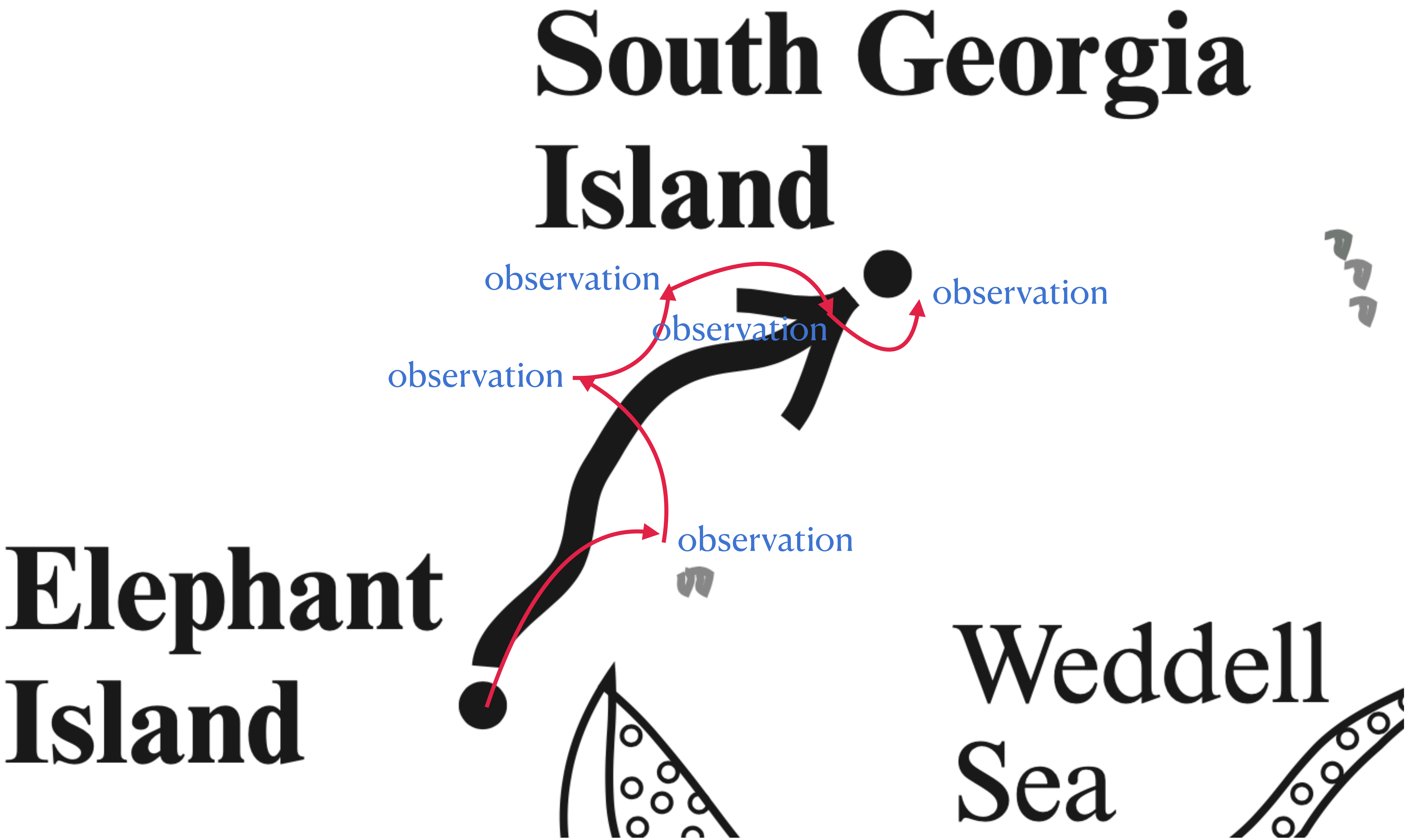

South Georgia
Island
observation
observation
observation
Elephant
Island
Weddell
Sea

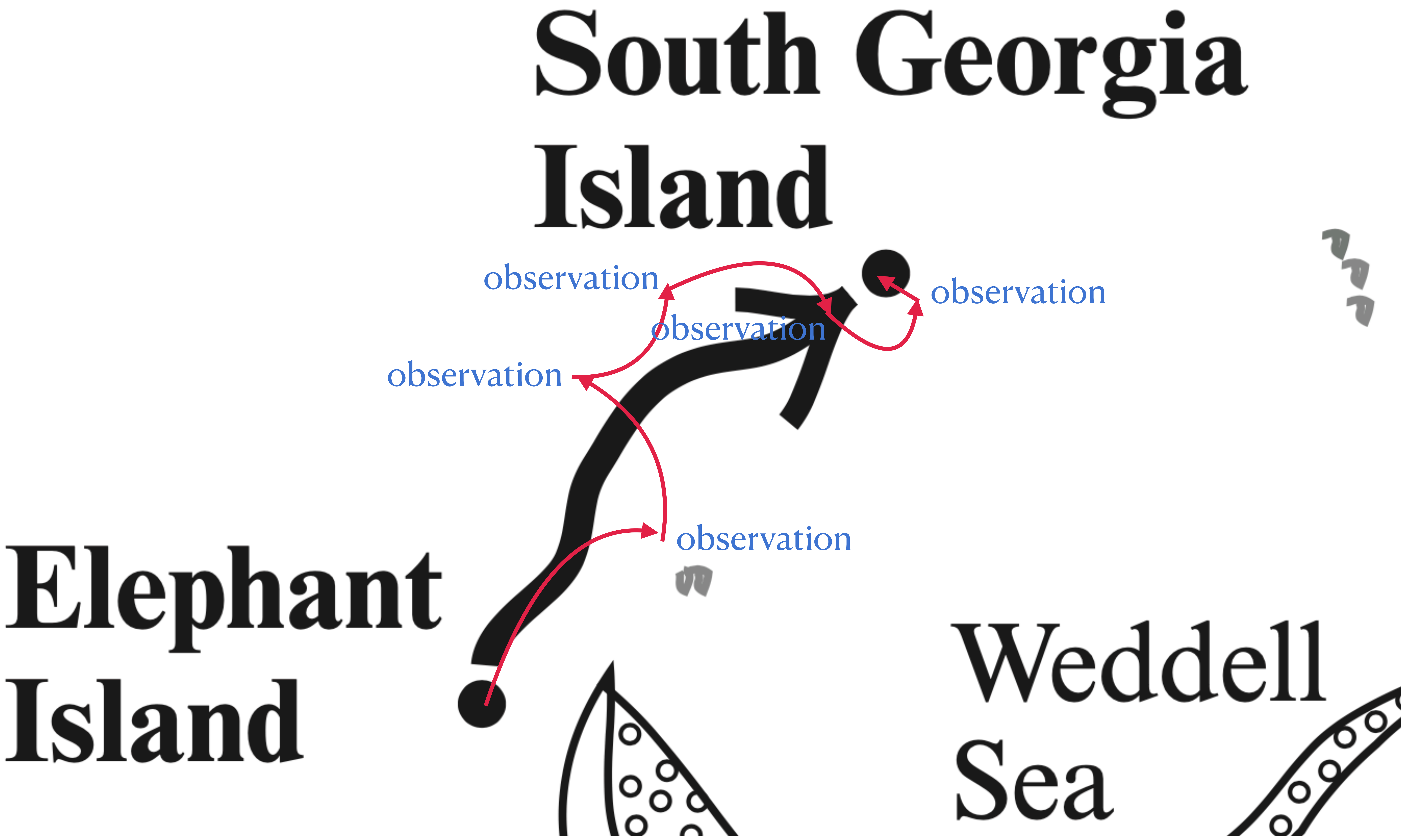

South Georgia Island
observation
observation
observation
Elephant Island
Weddell Sea

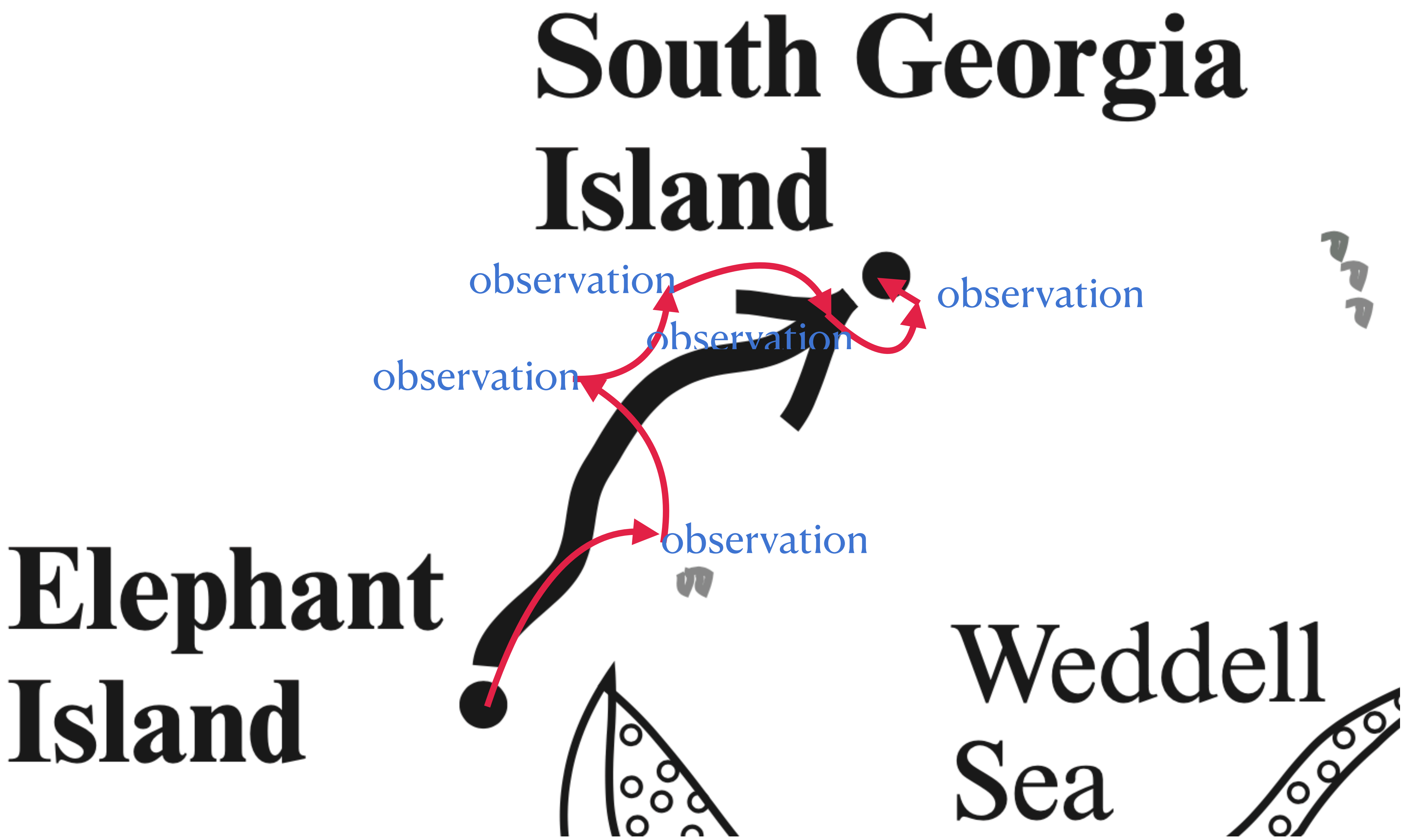


South Georgia Island
observation
observation
observation
observation
Elephant Island
Weddell Sea

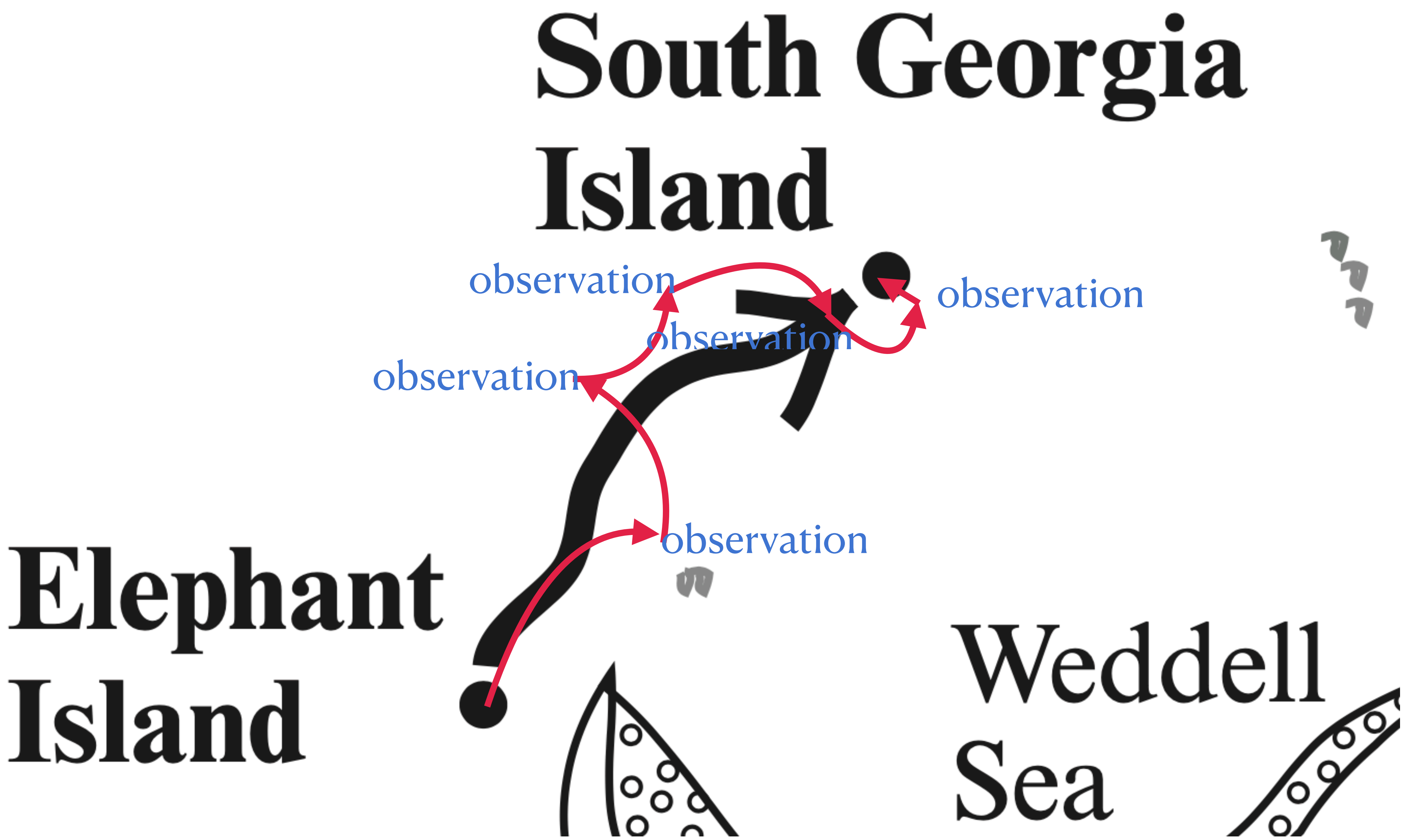

South Georgia Island
observation
observation
observation
observation
Elephant Island
Weddell Sea


South Georgia Island
observation
observation
observation
observation
observation
Elephant Island
Weddell Sea


South Georgia Island
observation
observation
observation
observation
observation
Elephant Island
Weddell Sea

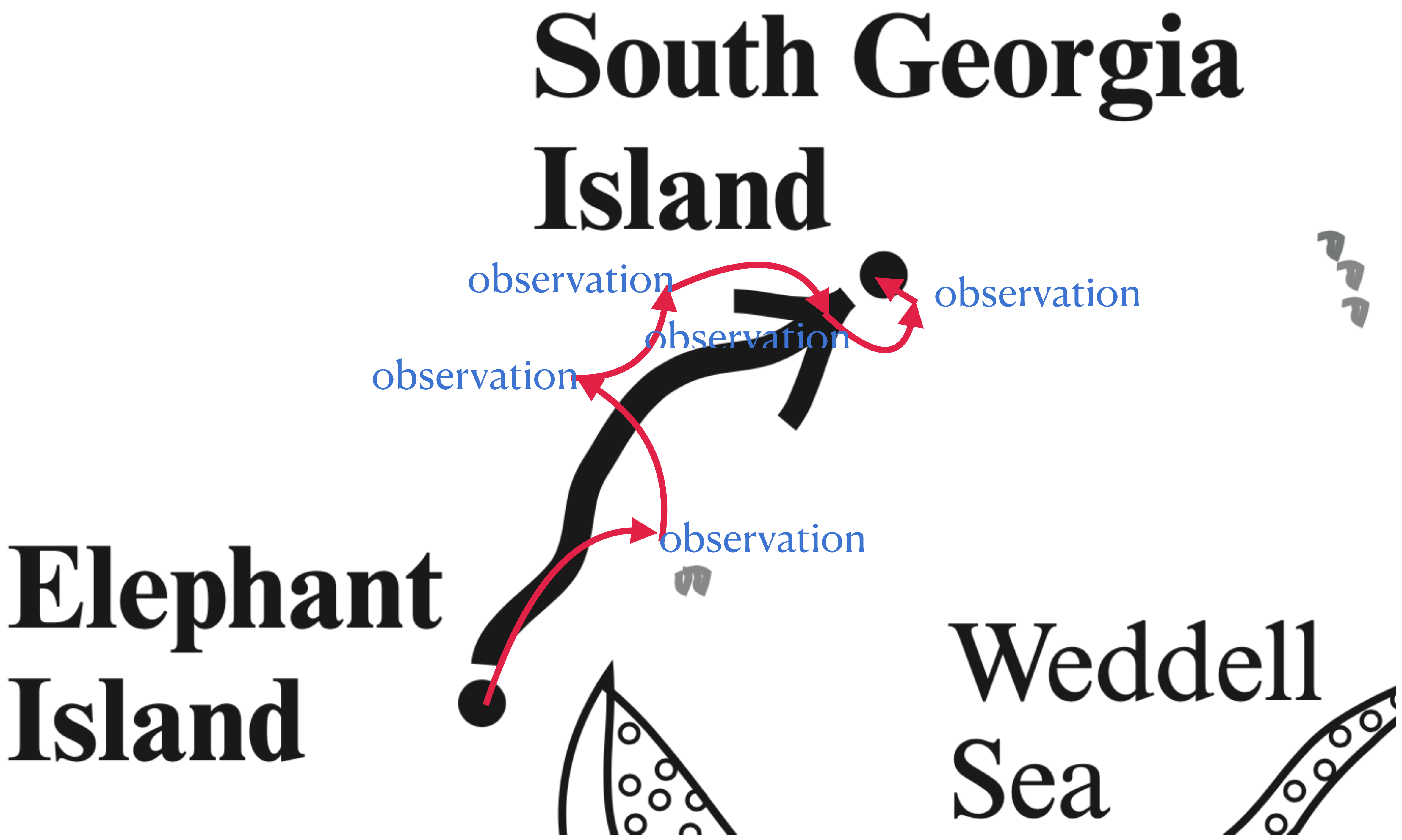


boat movement according to map —> observation of position and correction —>



boat movement according to map —> observation of position and correction —>

boat movement according to map —> observation of position and correction —>



boat movement according to map —> observation of position and correction —>

boat movement according to map —> observation of position and correction —>

boat movement according to map —> observation of position and correction —>

....

motivation

ideas cycling

basic methods

Kalman filter

prediction and verification

generalisation: system behaviour

generalisation: system behaviour —> correction by observation

generalisation: system behaviour —> correction by observation

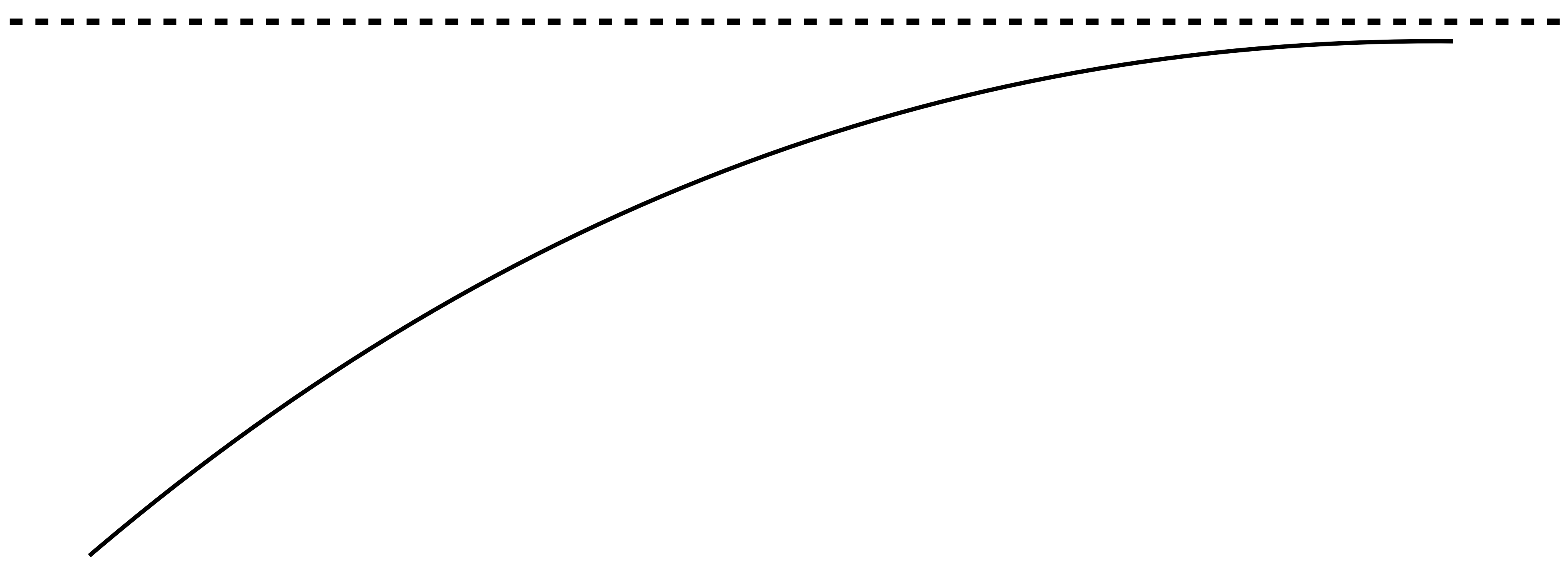
generalisation: system behaviour —> correction by observation

data assimilation cycle

generalisation: system behaviour —> correction by observation

data assimilation cycle

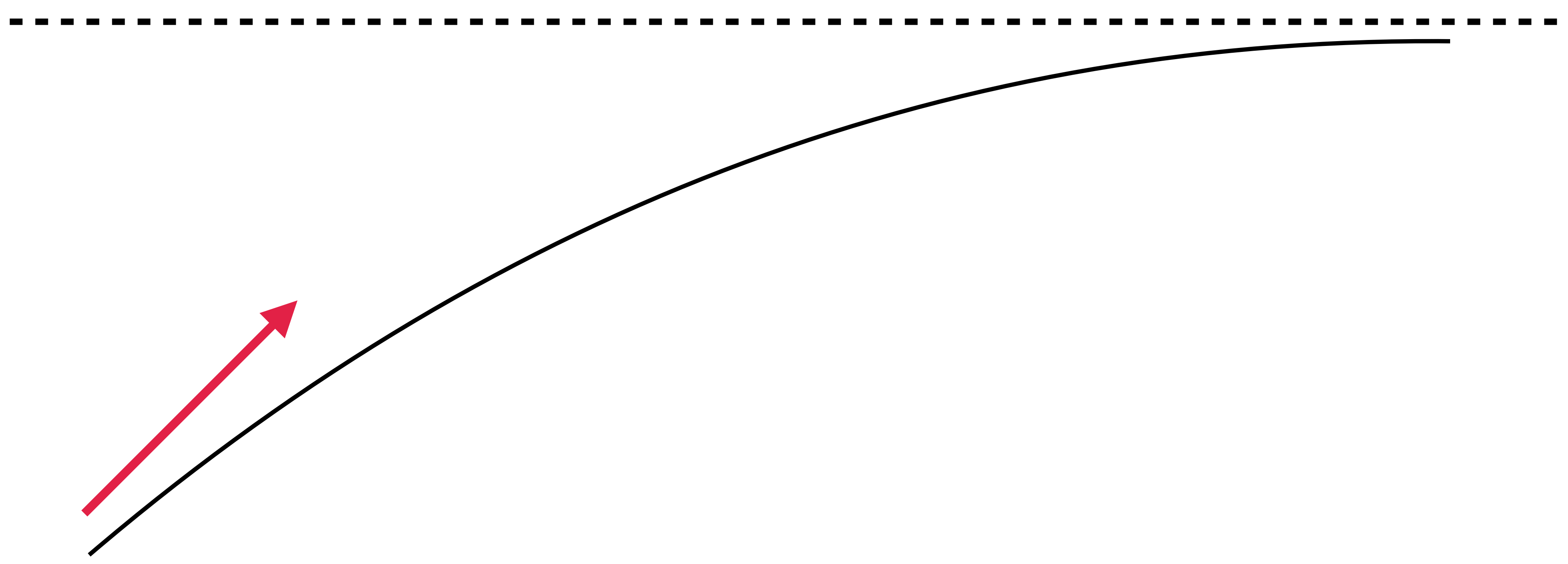
<u>other example:</u> **control of rocket trajectory**



generalisation: system behaviour —> correction by observation

data assimilation cycle

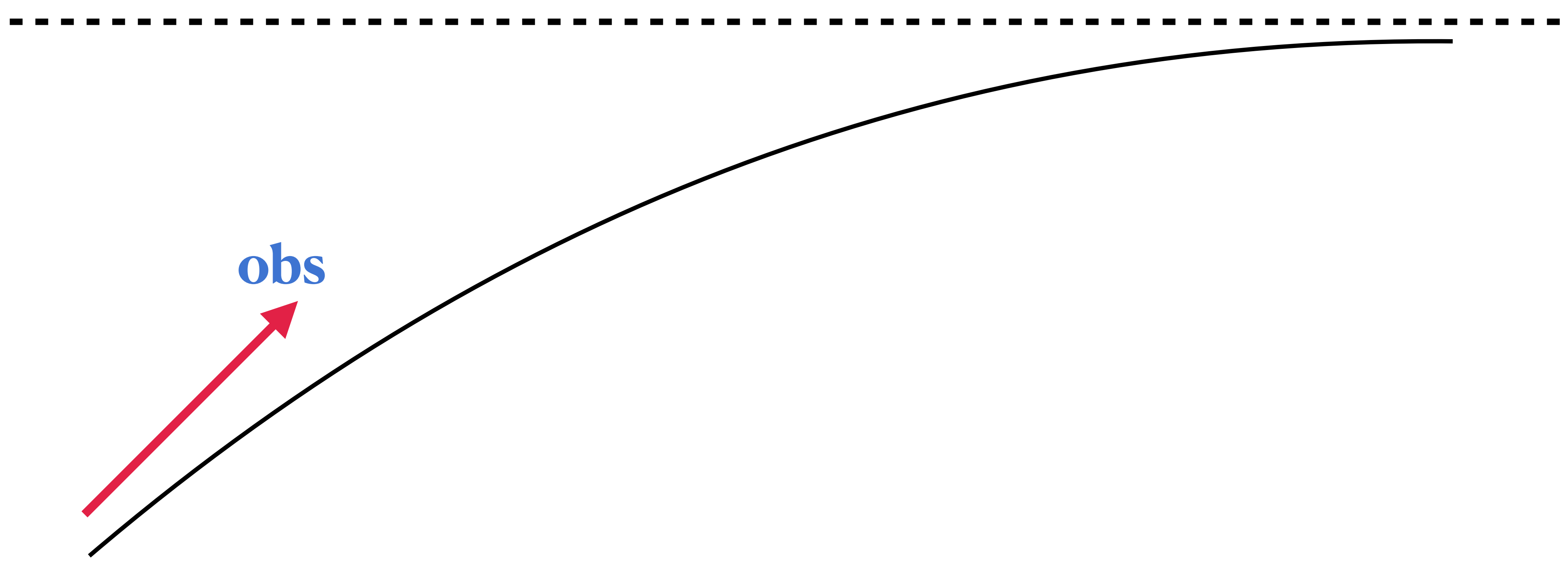
<u>other example:</u> **control of rocket trajectory**



generalisation: system behaviour —> correction by observation

data assimilation cycle

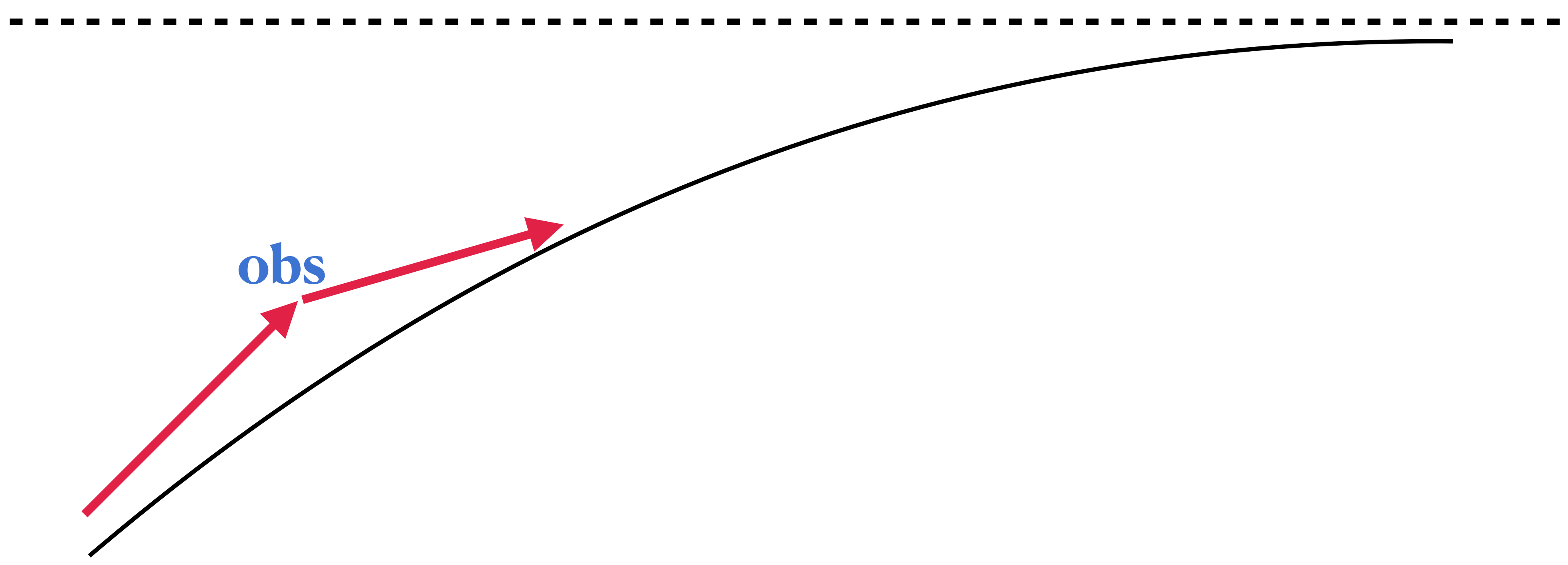
<u>other example:</u> **control of rocket trajectory**



generalisation: system behaviour —> correction by observation

data assimilation cycle

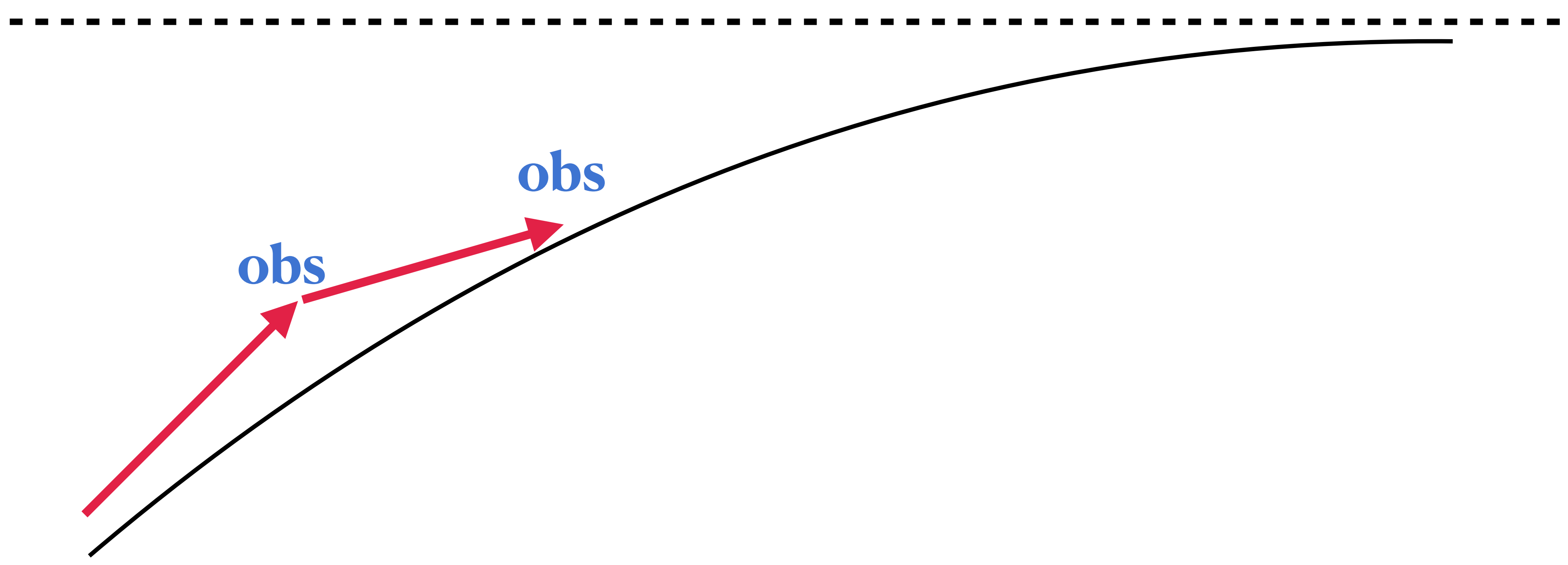
other example: **control of rocket trajectory**



generalisation: system behaviour —> correction by observation

data assimilation cycle

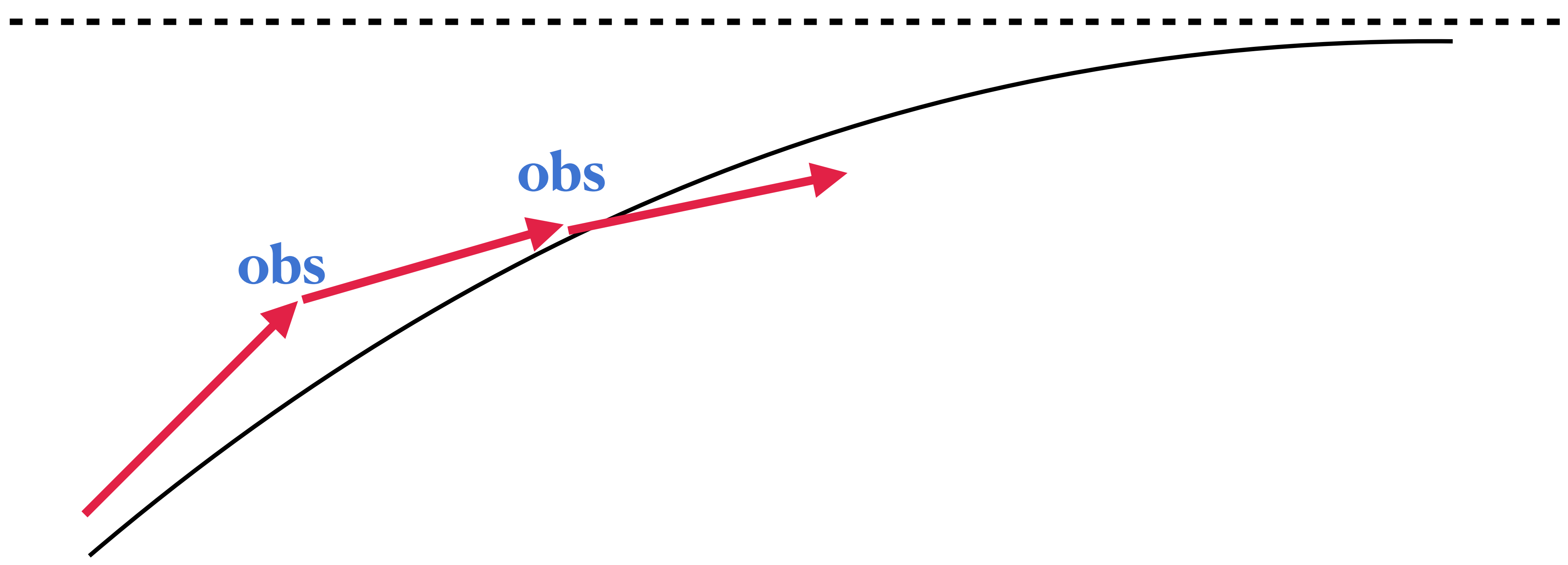
other example: **control of rocket trajectory**



generalisation: system behaviour —> correction by observation

data assimilation cycle

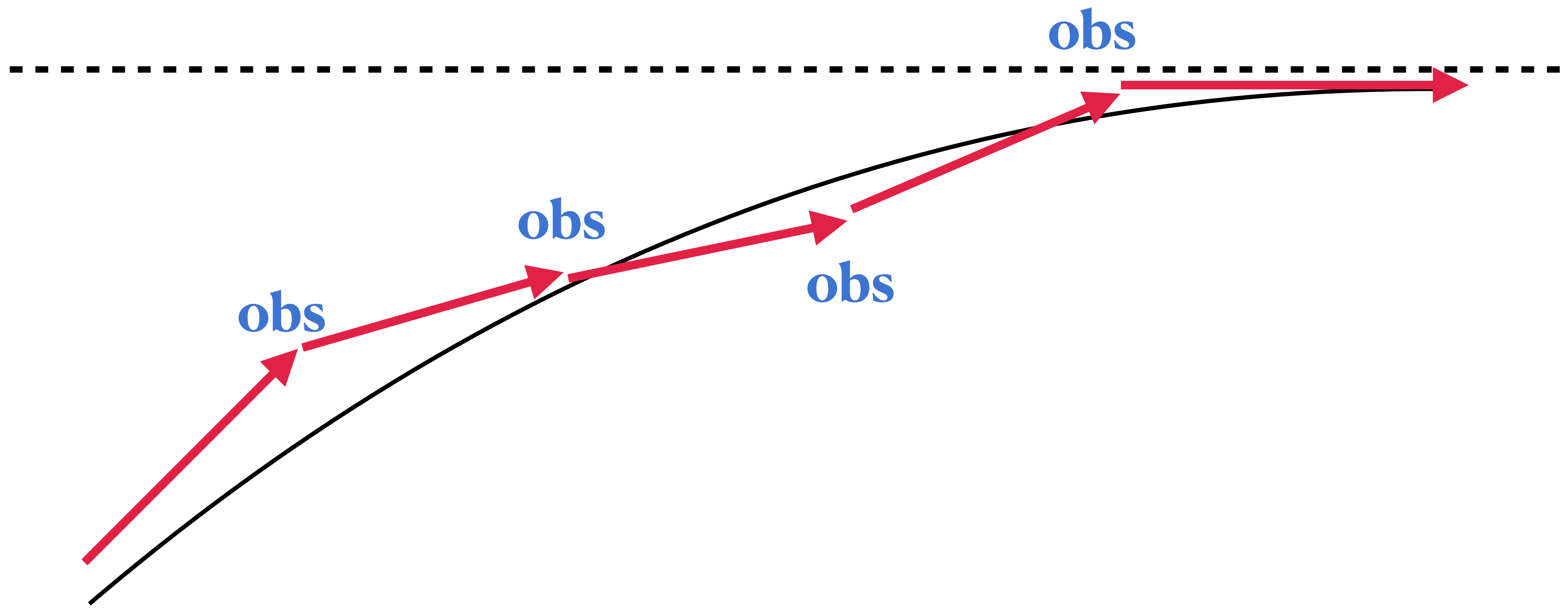
<u>other example:</u> **control of rocket trajectory**



generalisation: system behaviour —> correction by observation

data assimilation cycle

<u>other example:</u> **control of rocket trajectory**



Very simple data assimilation scheme : the **Cressman scheme** (or **nudging**)

weight factor

observation

$$\mathbf{x}_a(t) = \mathbf{x}_b(t) + \mathbf{K}(\mathbf{y} - \mathbf{x}_b(t))$$

analysis

background

Very simple data assimilation scheme : the **Cressman scheme** (or **nudging**)

weight factor

observation

$$\mathbf{x}_a(t) = \mathbf{x}_b(t) + \mathbf{K}(\mathbf{y} - \mathbf{x}_b(t))$$

analysis

background

$\mathbf{K} > 0$ if observation is close to system point

$\mathbf{K} = 0$ if observation is far from system point

Very simple data assimilation scheme : the **Cressman scheme** (or **nudging**)

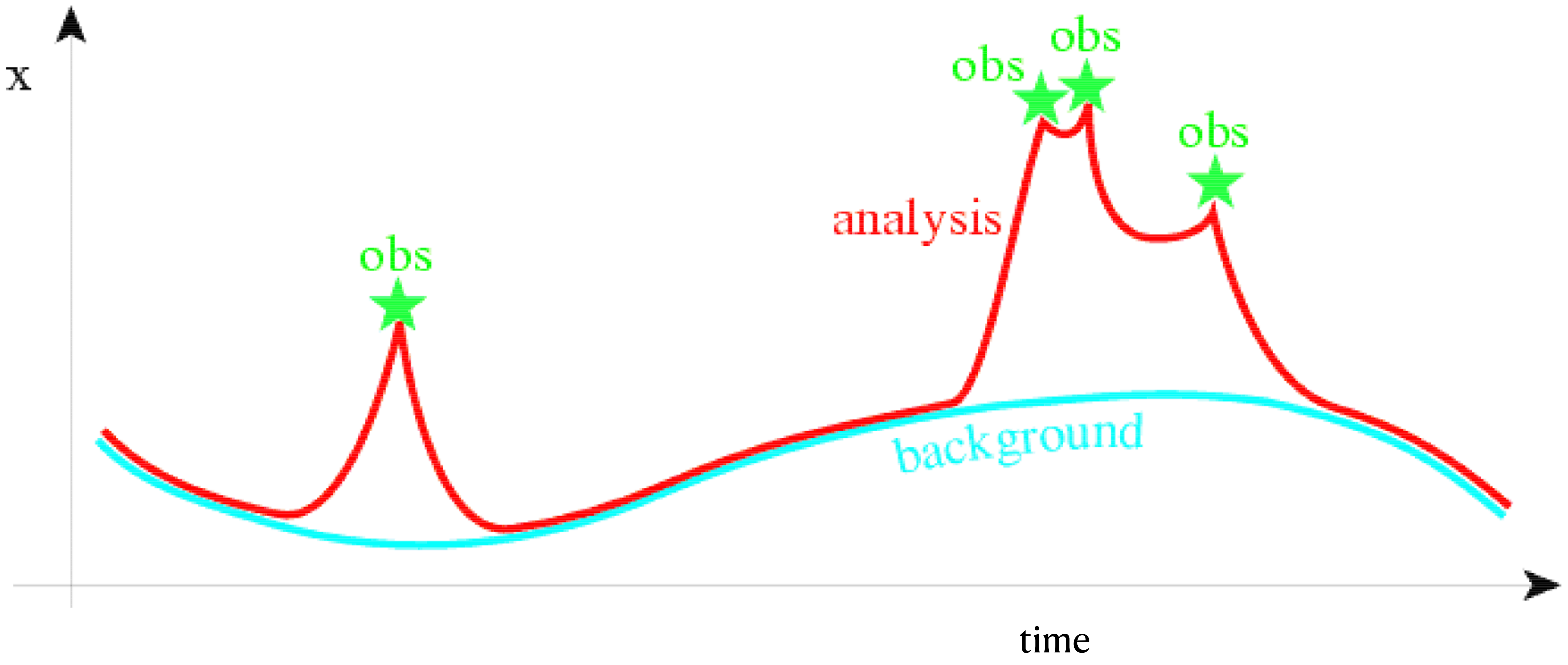
weight factor

observation

$$\mathbf{x}_a(t) = \mathbf{x}_b(t) + \mathbf{K}(\mathbf{y} - \mathbf{x}_b(t))$$

analysis

background



$\mathbf{K} > 0$ if observation is close to system point

$\mathbf{K} = 0$ if observation is far from system point stable equilibrium is reached if $\quad \mathbf{x}_a(t) = \mathbf{x}_b(t) + \mathbf{K}(\mathbf{y} - \mathbf{x}_b(t)) = \mathbf{x}_b(t)$

$\rightarrow \mathbf{y} = \mathbf{x}_b$ the system with activity $\mathbf{x}_b$ attempts to reach $\mathbf{y}$

stable equilibrium is reached if $\mathbf{x}_a(t) = \mathbf{x}_b(t) + \mathbf{K}(\mathbf{y} - \mathbf{x}_b(t)) = \mathbf{x}_b(t)$

$\rightarrow \mathbf{y} = \mathbf{x}_b$ the system with activity $\mathbf{x}_b$ attempts to reach $\mathbf{y}$

what happens in the system when no observation is present ?

$$\mathbf{x}_b(t) = \underbrace{\mathcal{M}}_{\text{model}}\mathbf{x}_b(t - \Delta t)$$

stable equilibrium is reached if $\mathbf{x}_a(t) = \mathbf{x}_b(t) + \mathbf{K}(\mathbf{y} - \mathbf{x}_b(t)) = \mathbf{x}_b(t)$

$\rightarrow \mathbf{y} = \mathbf{x}_b$ the system with activity $\mathbf{x}_b$ attempts to reach $\mathbf{y}$

what happens in the system when no observation is present ?

$$\mathbf{x}_b(t) = \underbrace{\mathcal{M}}_{\text{model}}\mathbf{x}_b(t - \Delta t)$$

system can be observed: implementation by the observation operator H

$$\mathbf{y}_b(t) = \mathbf{H}(\mathbf{x}_b(t))$$

Data Assimilation

system under study

state evolution:

$$x_n = F(x_{n-1})$$

Data Assimilation

system
under
study

state evolution:

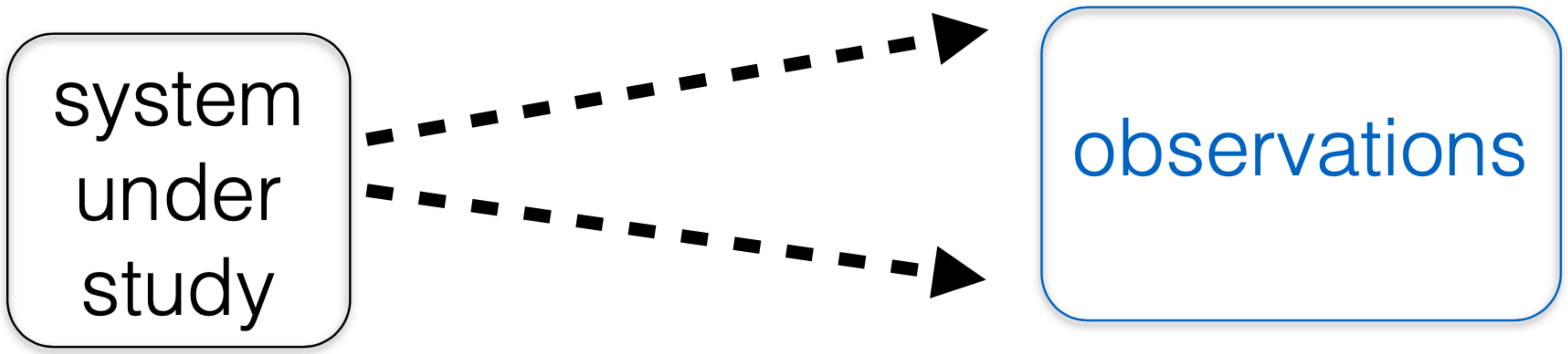
$$x_n = F(x_{n-1})$$

model evolution yields forecast:

$$x^f_n = M(x^a_{n-1}) + \varepsilon_n$$

model   error

Data Assimilation



state evolution:

$$x_n = F(x_{n-1})$$

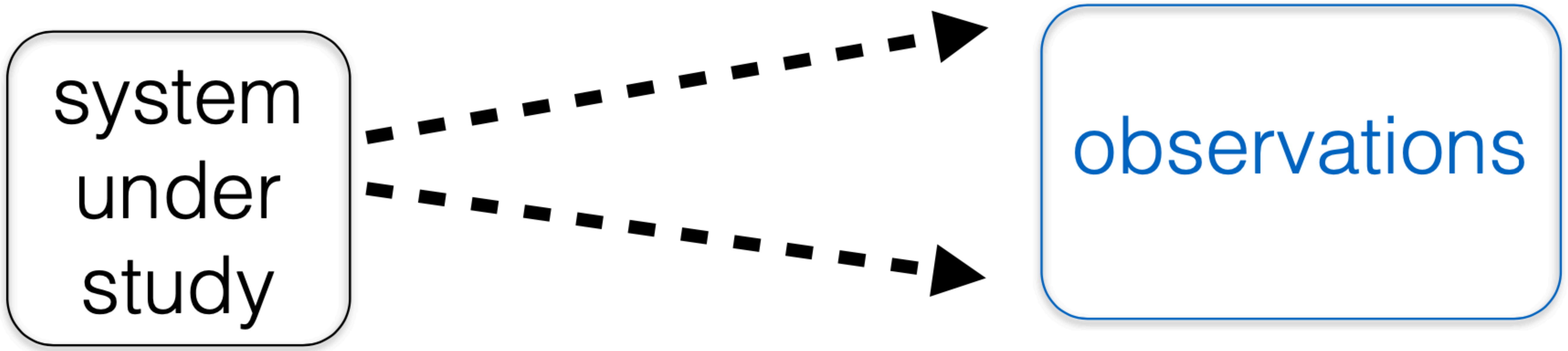
model evolution yields forecast:

$$x^f_n = M(x^a_{n-1}) + \varepsilon_n$$

model

error

Data Assimilation



state evolution:

$$x_n = F(x_{n-1})$$

$$\underbrace{y_n}_{\text{observations}} = \underbrace{H}_{\text{observation operator}}(x^f_n) + \overset{\text{error}}{\eta_n}$$

model evolution yields forecast:

$$x^f_n = \underbrace{M}_{\text{model}}(x^a_{n-1}) + \underbrace{\varepsilon_n}_{\text{error}}$$

Data Assimilation

system under study

observations

state evolution:

$$x_n = F(x_{n-1})$$

error

$$y_n = H(x^f_n) + \eta_n$$

observations   observation operator

model evolution yields forecast:

$$x^f_n = M(x^a_{n-1}) + \varepsilon_n$$

model   error

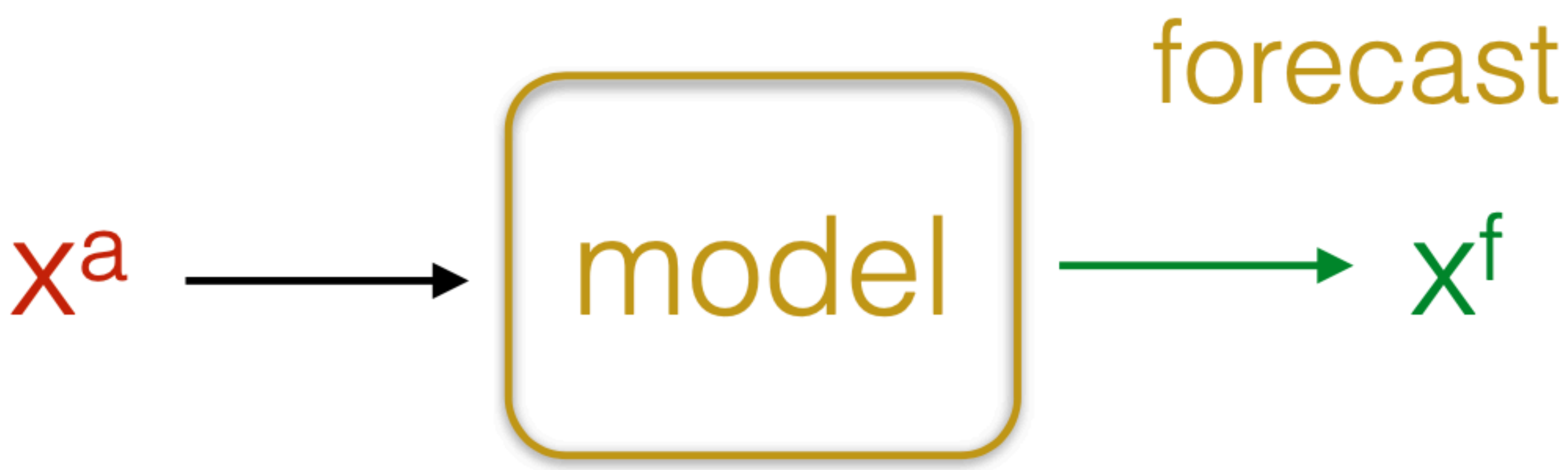
$$\boxed{x^a_n = x^f_n + \mathbf{K}\,(y_n - H(x^f_n))}$$

weight matrix / Kalman matrix

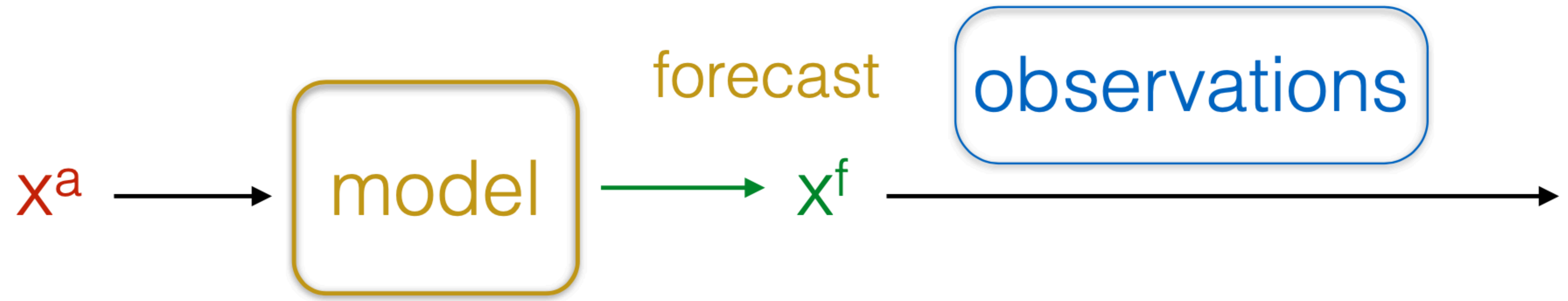
cycled Data Assimilation

x^a

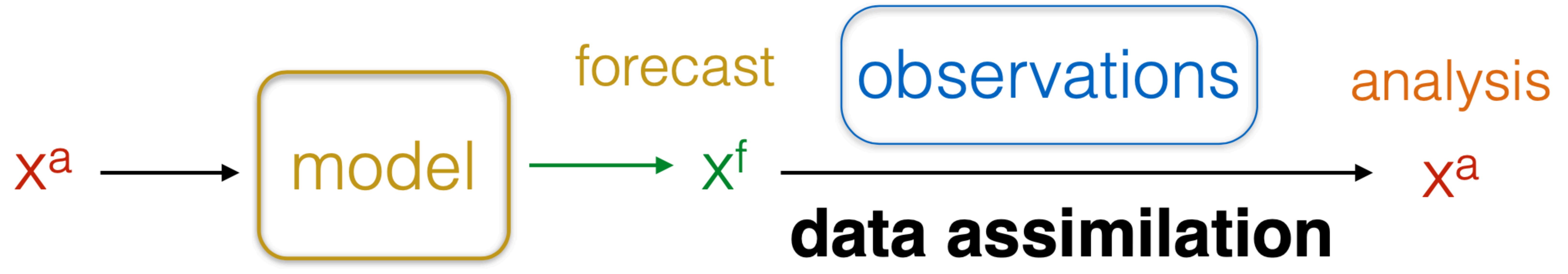
cycled Data Assimilation



cycled Data Assimilation



cycled Data Assimilation



cycled Data Assimilation

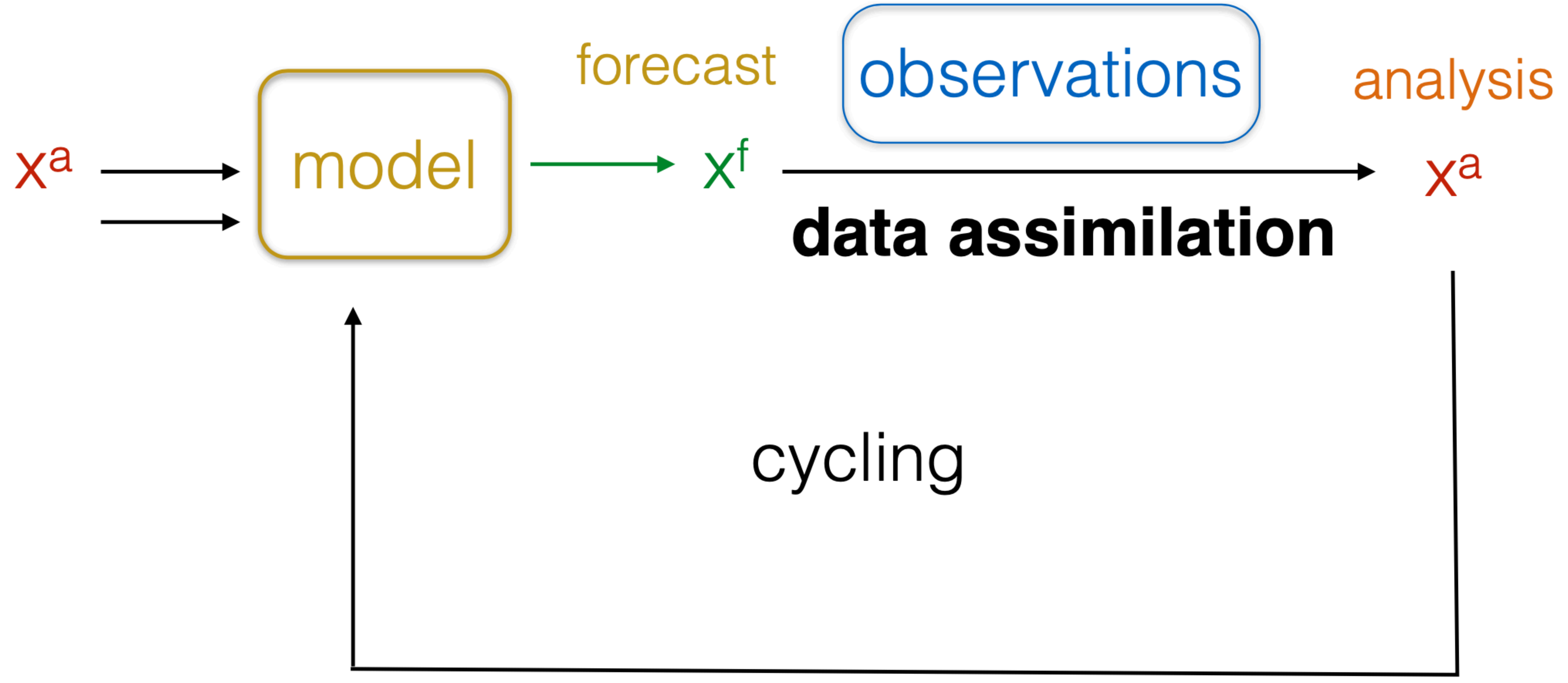
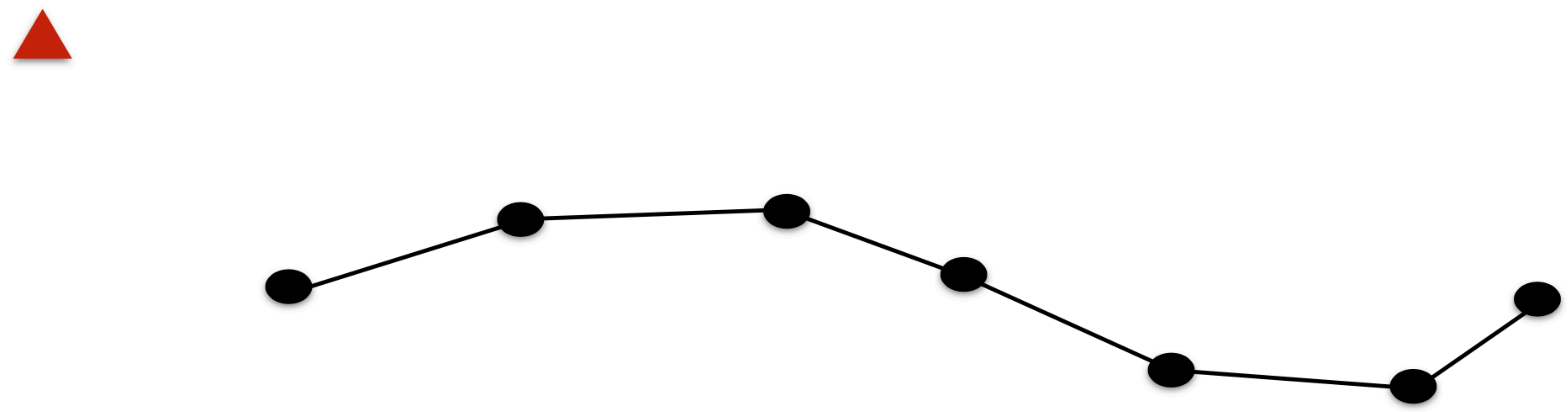


Illustration of cycling:



Data Assimilation

Illustration of cycling:

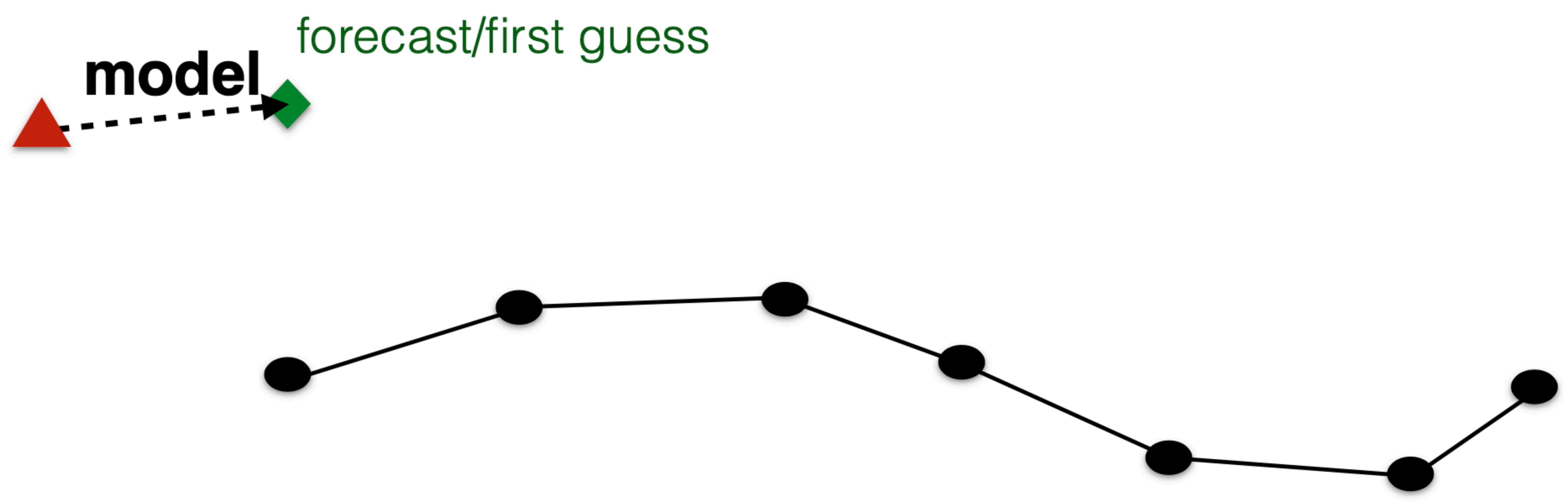


Illustration of cycling:

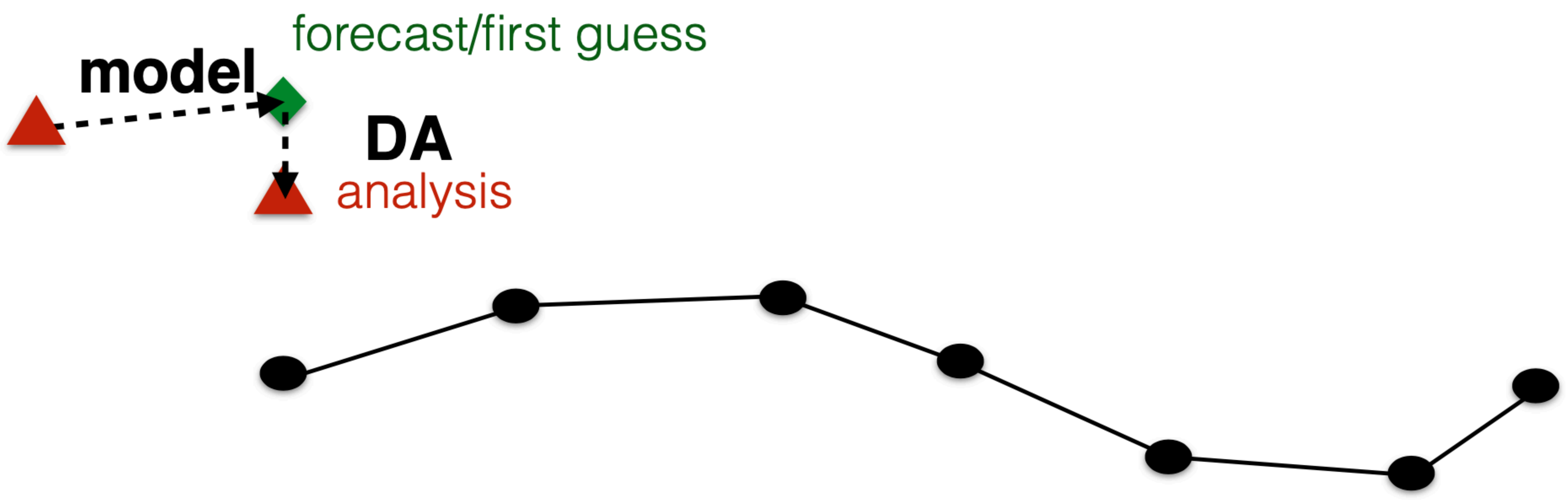


Illustration of cycling:

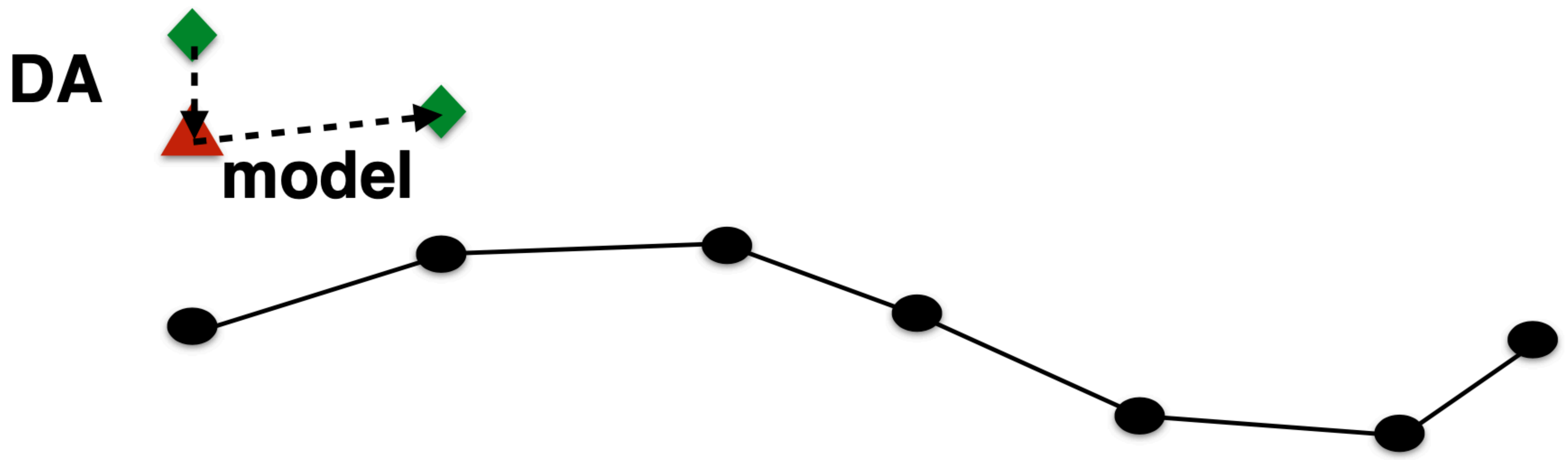


Illustration of cycling:

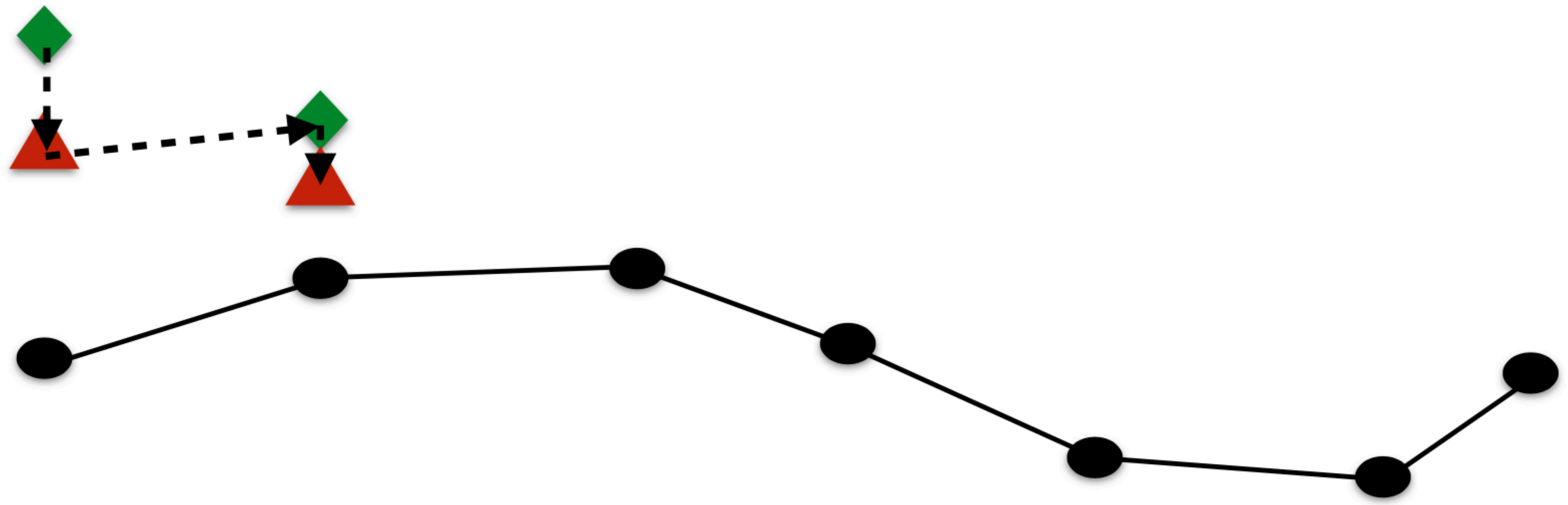
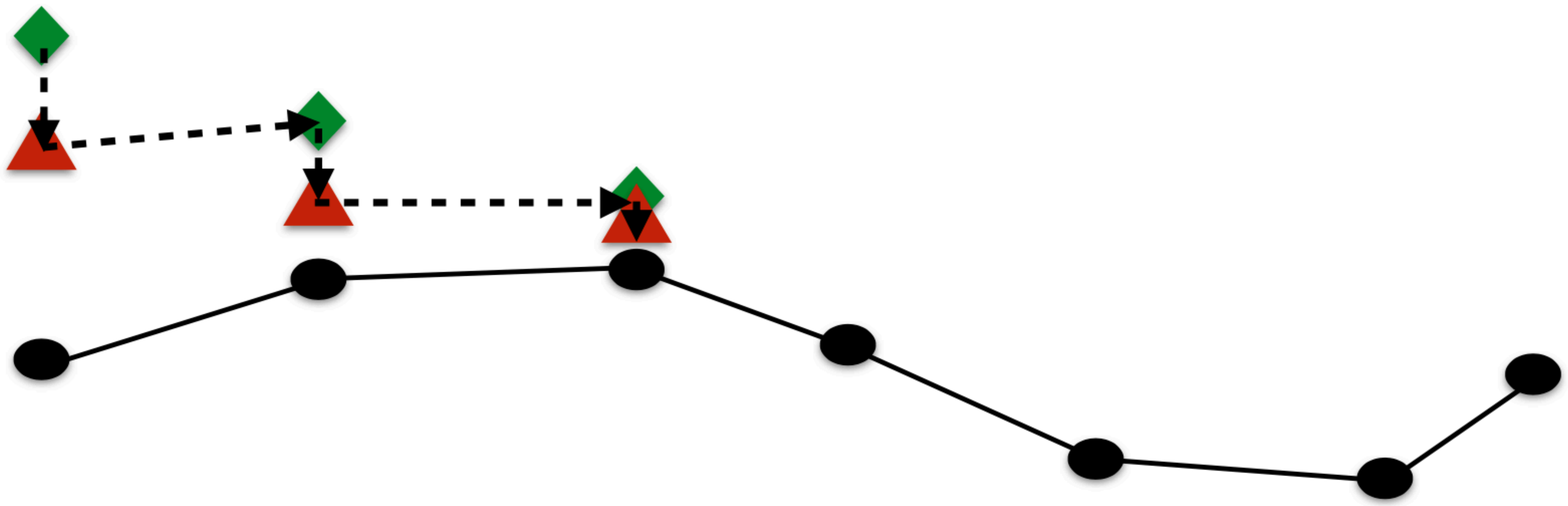
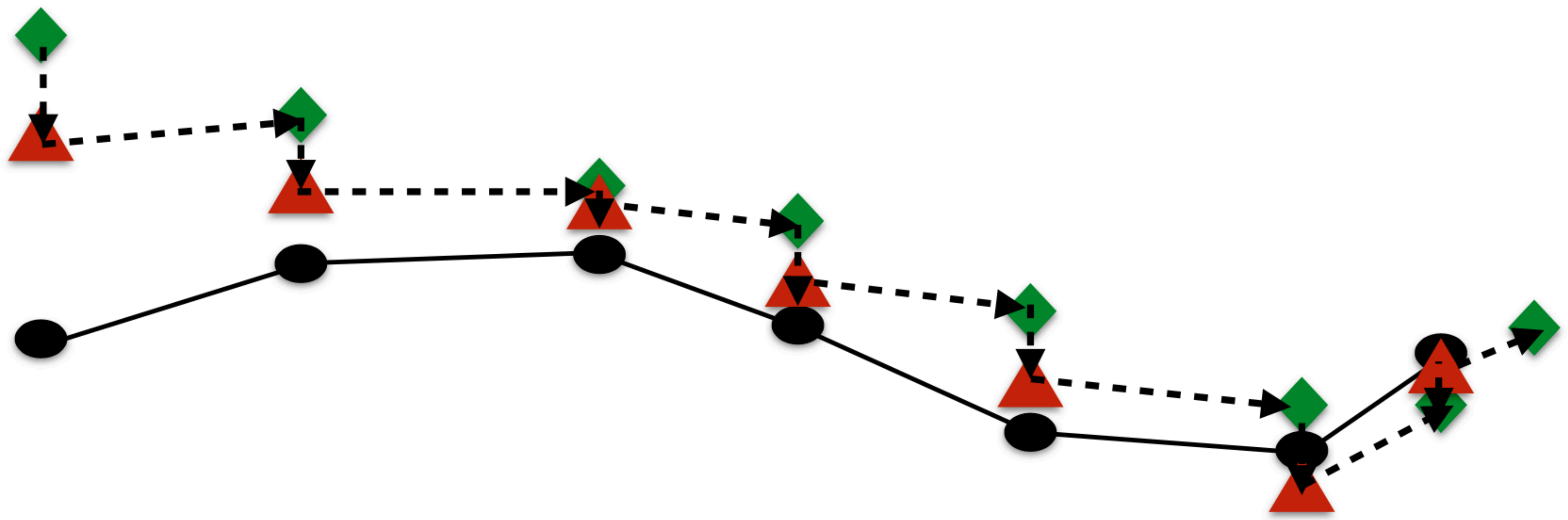


Illustration of cycling:



Data Assimilation

Illustration of cycling:



data assimilation:

data assimilation:

- it estimates time-dependant state variables

data assimilation:

- it estimates time-dependant state variables
    - parameter estimation

data assimilation:

- it estimates time-dependant state variables
  - parameter estimation
  - estimation of unobserved variables

data assimilation:

- it estimates time-dependant state variables
  - parameter estimation
  - estimation of unobserved variables
- the estimates improve model output / forecasts
  - improved forecasts

data assimilation:

- **it estimates time-dependant state variables**
  - **parameter estimation**
  - **estimation of unobserved variables**
- **the estimates improve model output / forecasts**
  - improved forecasts

motivation

basic methods

Kalman filter

prediction and verification

motivation

basic methods

BLUE 3DVar

Kalman filter

prediction and verification

Best Linear Unbiased Estimator (BLUE)

assume a linear model $$\mathbf{y} = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \ldots + \mathbf{x}^{(M)}\beta_M + \varepsilon$$

$\mathbf{y}$: N-dimensional observation

$\mathbf{x}^{(j)}$: N-dimensional state variable

β_j : state variable of type j

M : number of state variable types

Best Linear Unbiased Estimator (BLUE)

assume a linear model $$\mathbf{y} = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \ldots + \mathbf{x}^{(M)}\beta_M + \varepsilon$$

$\mathbf{y}$: N-dimensional observation

$\mathbf{x}^{(j)}$: N-dimensional state variable

β_j : state variable of type j

M : number of state variable types

ε : observation error $\quad E[\varepsilon] = 0 \quad E[\varepsilon\varepsilon^t] = \mathbf{R}$

$$R_{ii} \neq 0 \;,\; R_{ij} = 0 (i \neq j)$$

Best Linear Unbiased Estimator (BLUE)

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

Best Linear Unbiased Estimator (BLUE)

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

given: observations $\mathbf{y}$, state variables $\mathbf{x}$, observation error R_{ii}

to be estimated: model parameters β

Best Linear Unbiased Estimator (BLUE)

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

given: observations $\mathbf{y}$, state variables $\mathbf{x}$, observation error R_{ii}

to be estimated: model parameters β

cost function $C = (\mathbf{y} - \mathbf{X}\beta)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{X}\beta)$ has to be minimised

Best Linear Unbiased Estimator (BLUE)

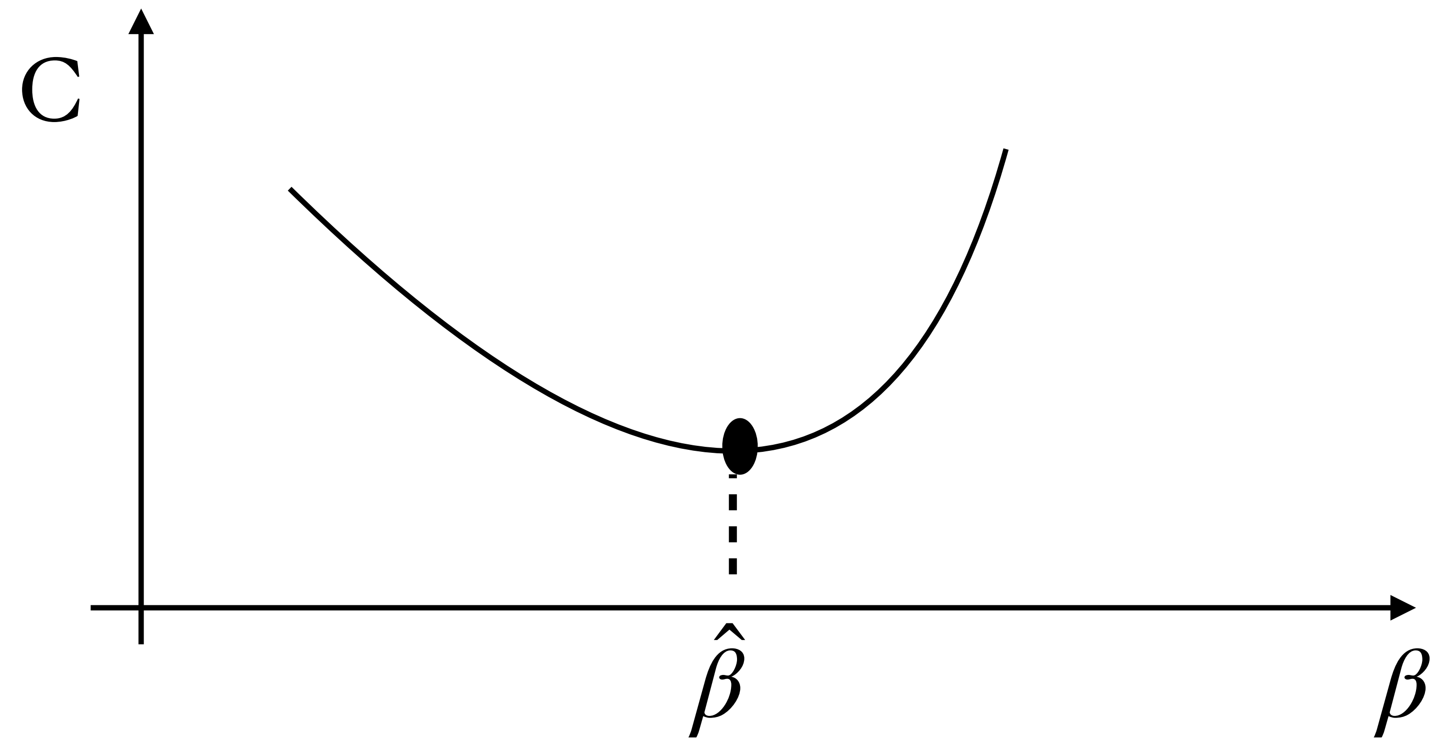
$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

given: observations $\mathbf{y}$, state variables $\mathbf{x}$, observation error R_{ii}

to be estimated: model parameters β

cost function $\boxed{C = (\mathbf{y} - \mathbf{X}\beta)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{X}\beta)}$ has to be minimised

$$C = \min \quad \Rightarrow \quad \hat{\beta} = (\mathbf{X}^t \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{W} \mathbf{y}$$



motivation

basic methods

BLUE **3DVar**

Kalman filter

prediction and verification

Variational method: 3DVar

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{B} \qquad\qquad E[\varepsilon] = 0$$

$$\mathbf{y}(t_n) = \mathbf{H}\mathbf{x}_b(t_n) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R} \qquad\qquad E[\eta] = 0$$

Variational method: 3DVar

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{B} \qquad E[\varepsilon] = 0$$

$$\mathbf{y}(t_n) = \mathbf{H}\mathbf{x}_b(t_n) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R} \qquad E[\eta] = 0$$

$\mathbf{B}$: covariance of model error ε

$\mathbf{R}$: covariance of observation error η

$\mathbf{H}$: linear observation operator matrix

Variational method: 3DVar

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{B} \qquad\qquad E[\varepsilon] = 0$$

$$\mathbf{y}(t_n) = \mathbf{H}\mathbf{x}_b(t_n) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R} \qquad\qquad E[\eta] = 0$$

$\mathbf{B}$: covariance of model error ε

$\mathbf{R}$: covariance of observation error η

$\mathbf{H}$: linear observation operator matrix

best estimate of state $\mathbf{x}_a$ when background $\mathbf{x}_b$ and observations $\mathbf{y}$ are given if

$$C = (\mathbf{x} - \mathbf{x}_b)^t\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + (\mathbf{y} - \mathbf{H}\mathbf{x})^t\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x})$$

$$C = \min \rightarrow \mathbf{x}_a$$

$$C(\mathbf{x}_a) = (\mathbf{x}_a - \mathbf{x}_b)^t \mathbf{B}^{-1} (\mathbf{x}_a - \mathbf{x}_b) + (\mathbf{y} - \mathbf{H}\mathbf{x}_a)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x}_a) = \min$$

$\mathbf{x}_a$ close to $\mathbf{x}_b$

$\mathbf{H}\mathbf{x}_a$ close to $\mathbf{y}$

$$C(\mathbf{x}_a) = (\mathbf{x}_a - \mathbf{x}_b)^t\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) + (\mathbf{y} - \mathbf{H}\mathbf{x}_a)^t\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}_a) = \min$$

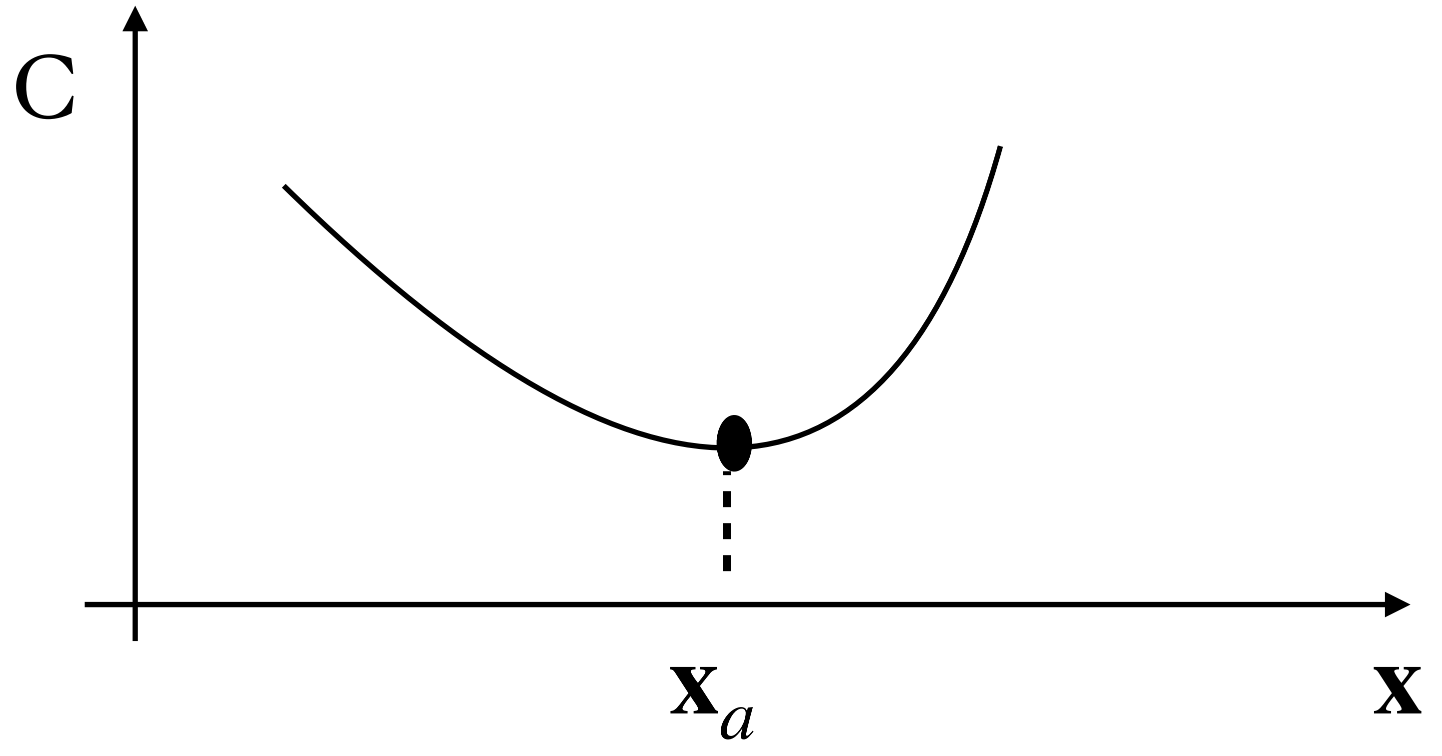
$\mathbf{x}_a$ close to $\mathbf{x}_b$ $\mathbf{H}\mathbf{x}_a$ close to $\mathbf{y}$

result:

$$\boxed{\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_\mathbf{b})}$$

$$\boxed{\mathbf{K} = \mathbf{B}\mathbf{H}^t(\mathbf{H}\mathbf{B}\mathbf{H}^t + \mathbf{R})^{-1}}$$

BLUE



$$C(\mathbf{x}_a) = (\mathbf{x}_a - \mathbf{x}_b)^t \mathbf{B}^{-1} (\mathbf{x}_a - \mathbf{x}_b) + (\mathbf{y} - \mathbf{H}\mathbf{x}_a)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x}_a) = \min$$

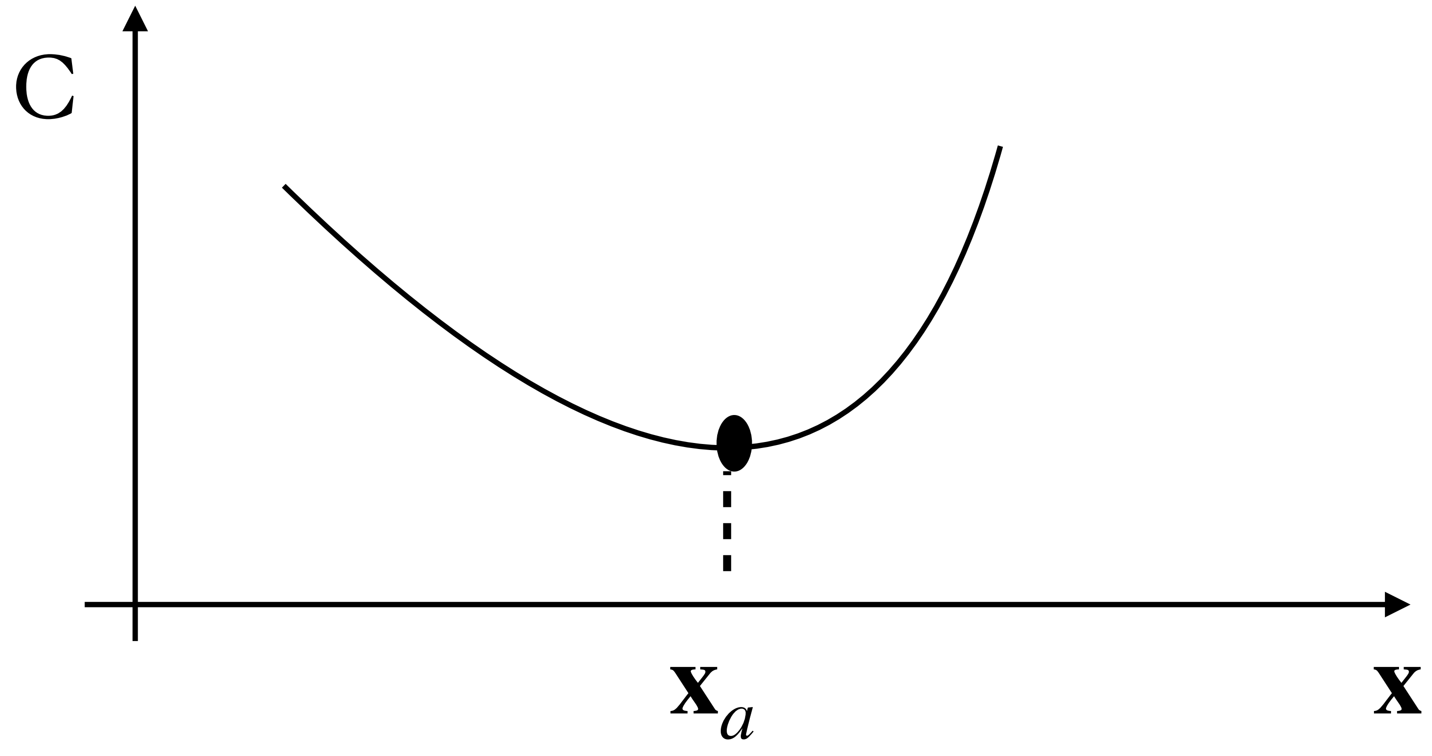
$\mathbf{x}_a$ close to $\mathbf{x}_b$ $\qquad$ $\mathbf{H}\mathbf{x}_a$ close to $\mathbf{y}$

result:

$$\boxed{\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_\mathbf{b})}$$

$$\boxed{\mathbf{K} = \mathbf{B}\mathbf{H}^t (\mathbf{H}\mathbf{B}\mathbf{H}^t + \mathbf{R})^{-1}}$$

BLUE



for linear observation operator $H(\mathbf{x}_b) = \mathbf{H}\mathbf{x}_b$ only

Data Assimilation

system under study

observations

re-call:

state evolution:

$$x_n = F(x_{n-1})$$

error

$$y_n = H(x^f_n) + \eta_n$$

observations observation operator

model evolution yields forecast:

$$x^f_n = M(x^a_{n-1}) + \varepsilon_n$$

model error

$$\boxed{x^a_n = x^f_n + \mathbf{K}\,(y_n - H(x^f_n))}$$

weight matrix / Kalman matrix

re-call:

Data Assimilation

system under study

observations

state evolution:

$x_n=F(x_{n-1})$

error

$y_n=H(x^f_n)+\eta_n$

observations observation operator

model evolution yields forecast:

$x^f_n=M(x^a_{n-1})+\varepsilon_n$

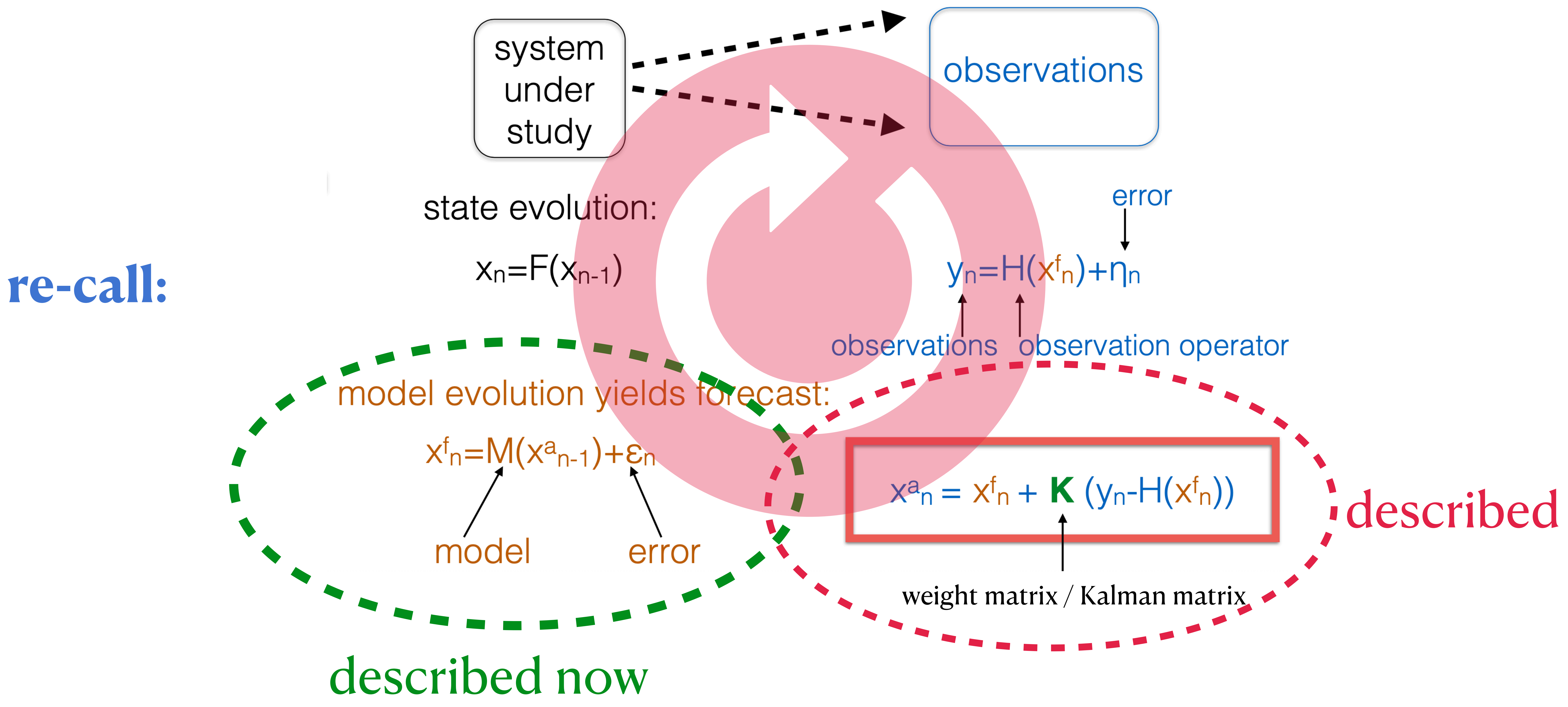
model error

$$\boxed{x^a_n = x^f_n + \mathbf{K}\,(y_n-H(x^f_n))}$$

weight matrix / Kalman matrix

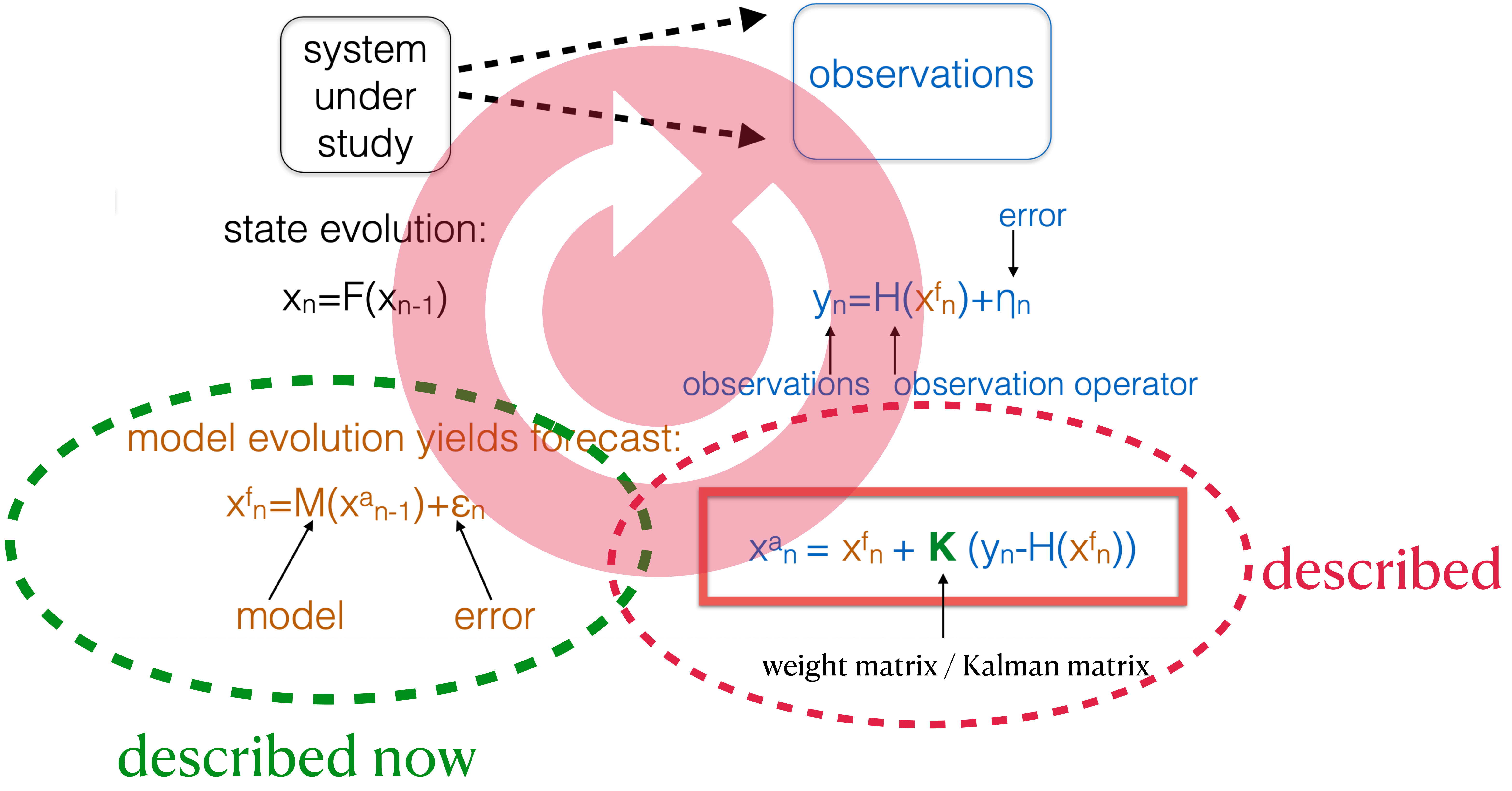
described

Data Assimilation



Data Assimilation

re-call:



$$\boxed{\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_a(t_{n-1})}$$

background covariance **B** is constant in time

all together:

all together:

- observation operator is linear

all together:

- observation operator is linear

at time $t_{n-1} \rightarrow t_n$ prediction step

$$\boxed{\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_a(t_{n-1})}$$

all together:

- observation operator is linear

at time $t_{n-1} \rightarrow t_n$ prediction step

$$\boxed{\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_a(t_{n-1})}$$

at time t_n analysis step

$$\boxed{\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x_b})}$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^t(\mathbf{H}\mathbf{B}\mathbf{H}^t + \mathbf{R})^{-1}$$

all together:

- observation operator is linear

at time $t_{n-1} \rightarrow t_n$ prediction step

$$\boxed{\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_a(t_{n-1})}$$

cycle step

at time t_n analysis step

$$\boxed{\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_\mathbf{b})}$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^t(\mathbf{H}\mathbf{B}\mathbf{H}^t + \mathbf{R})^{-1}$$

important remark:

important remark:

$$\begin{aligned}
\mathbf{x}_a &= \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{H}\mathbf{x}_a &= \mathbf{H}\mathbf{x}_b + \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{y} - \mathbf{H}\mathbf{x}_a &= \mathbf{y} - \mathbf{H}\mathbf{x}_b - \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{d}_{n+1} &= \mathbf{d}_n - \mathbf{H}\mathbf{K}\mathbf{d}_n
\end{aligned}$$

important remark:

$$
\begin{aligned}
\mathbf{x}_a &= \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{H}\mathbf{x}_a &= \mathbf{H}\mathbf{x}_b + \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{y} - \mathbf{H}\mathbf{x}_a &= \mathbf{y} - \mathbf{H}\mathbf{x}_b - \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{d}_{n+1} &= \mathbf{d}_n - \mathbf{H}\mathbf{K}\mathbf{d}_n
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{d}_{n+1} &= \mathbf{y} - \mathbf{H}\mathbf{x}_a \\
\mathbf{d}_n &= \mathbf{y} - \mathbf{H}\mathbf{x}_b
\end{aligned}
$$

important remark:

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b)$$

$$\mathbf{H}\mathbf{x}_a = \mathbf{H}\mathbf{x}_b + \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b)$$

$$\mathbf{y} - \mathbf{H}\mathbf{x}_a = \mathbf{y} - \mathbf{H}\mathbf{x}_b - \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b)$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mathbf{H}\mathbf{K}\mathbf{d}_n$$

$$\mathbf{d}_{n+1} = \mathbf{y} - \mathbf{H}\mathbf{x}_a$$

$$\mathbf{d}_n = \mathbf{y} - \mathbf{H}\mathbf{x}_b$$

$$\rightarrow \quad \frac{d}{dt}\mathbf{d}(t) = -\frac{1}{\Delta t}\mathbf{H}\mathbf{K}\mathbf{d}(t)$$

important remark:

$$\begin{aligned}\mathbf{x}_a &= \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b)\\ \mathbf{H}\mathbf{x}_a &= \mathbf{H}\mathbf{x}_b + \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b)\\ \mathbf{y} - \mathbf{H}\mathbf{x}_a &= \mathbf{y} - \mathbf{H}\mathbf{x}_b - \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b)\\ \mathbf{d}_{n+1} &= \mathbf{d}_n - \mathbf{H}\mathbf{K}\mathbf{d}_n\end{aligned}$$

$$\begin{aligned}\mathbf{d}_{n+1} &= \mathbf{y} - \mathbf{H}\mathbf{x}_a\\ \mathbf{d}_n &= \mathbf{y} - \mathbf{H}\mathbf{x}_b\end{aligned}$$

$$\rightarrow \quad \frac{d}{dt}\mathbf{d}(t) = -\frac{1}{\Delta t}\mathbf{H}\mathbf{K}\mathbf{d}(t)$$

$\boxed{\mathbf{d}(t) \rightarrow 0 \quad \text{for } t \rightarrow \infty}$ since $\mathbf{HK} = \mathbf{HBH}^t(\mathbf{R} + \mathbf{HBH}^t)^{-1}$ is positive definite

important remark:

$$\begin{aligned}
\mathbf{x}_a &= \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{H}\mathbf{x}_a &= \mathbf{H}\mathbf{x}_b + \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{y} - \mathbf{H}\mathbf{x}_a &= \mathbf{y} - \mathbf{H}\mathbf{x}_b - \mathbf{H}\mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \\
\mathbf{d}_{n+1} &= \mathbf{d}_n - \mathbf{H}\mathbf{K}\mathbf{d}_n
\end{aligned}$$

$$\begin{aligned}
\mathbf{d}_{n+1} &= \mathbf{y} - \mathbf{H}\mathbf{x}_a \\
\mathbf{d}_n &= \mathbf{y} - \mathbf{H}\mathbf{x}_b
\end{aligned}$$

$$\rightarrow \quad \frac{d}{dt}\mathbf{d}(t) = -\frac{1}{\Delta t}\mathbf{H}\mathbf{K}\mathbf{d}(t)$$

$$\boxed{\mathbf{d}(t) \rightarrow 0 \quad \text{for } t \rightarrow \infty}$$

since $\mathbf{HK} = \mathbf{HBH}^t(\mathbf{R} + \mathbf{HBH}^t)^{-1}$ is positive definite

analysis is always closer to observation than the background

motivation

basic methods

BLUE

3DVar

model and implementation

example results

reconstruction of hidden variables

parameter estimation

Kalman filter

prediction and verification

Example: FitzHugh-Nagumo model

$$\dot{x}_1 = x - \frac{1}{3}x_1^3 - y_1 + I_1 + \xi_1(t)$$

$$\tau_1 \dot{y}_1 = (b_{11}x_1 + b_{01} - y_1)$$

$$\dot{x}_2 = x - \frac{1}{3}x_2^3 - y_2 + I_2 + \xi_2(t)$$

$$\tau_2 \dot{y}_2 = (b_{12}x_2 + b_{02} - y_2)$$

Example: FitzHugh-Nagumo model

$$\dot{x}_1 = x - \frac{1}{3}x_1^3 - y_1 + I_1 + \xi_1(t)$$

$$\tau_1 \dot{y}_1 = (b_{11}x_1 + b_{01} - y_1)$$

$$\dot{x}_2 = x - \frac{1}{3}x_2^3 - y_2 + I_2 + \xi_2(t)$$

$$\tau_2 \dot{y}_2 = (b_{12}x_2 + b_{02} - y_2)$$

$$\xrightarrow[\text{discretization}]{\text{time}}$$

$$x_{n+1}^{(1)} = x_n^{(1)} + \Delta t(x_n^{(1)} - \frac{1}{3}(x_{n+1}^{(1)})^3 - y_n^{(1)} + I_1) + \xi_n^{(1)}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + \Delta t(b_{11}x^{(1)} + b_{01} - y_n^{(1)})/\tau_1$$

$$x_{n+1}^{(2)} = x_n^{(2)} + \Delta t(x_n^{(2)} - \frac{1}{3}(x_{n+1}^{(2)})^3 - y_n^{(2)} + I_2) + \xi_n^{(2)}$$

$$y_{n+1}^{(2)} = y_n^{(2)} + \Delta t(b_{12}x^{(2)} + b_{02} - y_n^{(2)})/\tau_2$$

$$\mathbf{x} \in \mathbb{R}^4 \qquad \mathbf{B} \in \mathbb{R}^{4\times 4}$$

Example: FitzHugh-Nagumo model

$$\dot{x}_1 = x - \frac{1}{3}x_1^3 - y_1 + I_1 + \xi_1(t)$$

$$\tau_1 \dot{y}_1 = (b_{11}x_1 + b_{01} - y_1)$$

$$\dot{x}_2 = x - \frac{1}{3}x_2^3 - y_2 + I_2 + \xi_2(t)$$

$$\tau_2 \dot{y}_2 = (b_{12}x_2 + b_{02} - y_2)$$

time

discretization

$$x_{n+1}^{(1)} = x_n^{(1)} + \Delta t(x_n^{(1)} - \frac{1}{3}(x_{n+1}^{(1)})^3 - y_n^{(1)} + I_1) + \xi_n^{(1)}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + \Delta t(b_{11}x^{(1)} + b_{01} - y_n^{(1)})/\tau_1$$

$$x_{n+1}^{(2)} = x_n^{(2)} + \Delta t(x_n^{(2)} - \frac{1}{3}(x_{n+1}^{(2)})^3 - y_n^{(2)} + I_2) + \xi_n^{(2)}$$

$$y_{n+1}^{(2)} = y_n^{(2)} + \Delta t(b_{12}x^{(2)} + b_{02} - y_n^{(2)})/\tau_2$$

$\mathbf{x} \in \mathbb{R}^4$ $\qquad$ $\mathbf{B} \in \mathbb{R}^{4\times 4}$

observations $y_1 = x_1$ and $y_2 = x_2$

$\mathbf{y} \in \mathbb{R}^2$

Example: FitzHugh-Nagumo model

$$\dot{x}_1 = x - \frac{1}{3}x_1^3 - y_1 + I_1 + \xi_1(t)$$

$$\tau_1 \dot{y}_1 = (b_{11}x_1 + b_{01} - y_1)$$

$$\dot{x}_2 = x - \frac{1}{3}x_2^3 - y_2 + I_2 + \xi_2(t)$$

$$\tau_2 \dot{y}_2 = (b_{12}x_2 + b_{02} - y_2)$$

$$\xrightarrow[\text{discretization}]{\text{time}}$$

$$x_{n+1}^{(1)} = x_n^{(1)} + \Delta t(x_n^{(1)} - \frac{1}{3}(x_{n+1}^{(1)})^3 - y_n^{(1)} + I_1) + \xi_n^{(1)}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + \Delta t(b_{11}x^{(1)} + b_{01} - y_n^{(1)})/\tau_1$$

$$x_{n+1}^{(2)} = x_n^{(2)} + \Delta t(x_n^{(2)} - \frac{1}{3}(x_{n+1}^{(2)})^3 - y_n^{(2)} + I_2) + \xi_n^{(2)}$$

$$y_{n+1}^{(2)} = y_n^{(2)} + \Delta t(b_{12}x^{(2)} + b_{02} - y_n^{(2)})/\tau_2$$

$\mathbf{x} \in \mathbb{R}^4$ $\qquad$ $\mathbf{B} \in \mathbb{R}^{4\times 4}$

observations $y_1 = x_1$ and $y_2 = x_2$

$\mathbf{y} \in \mathbb{R}^2$

$$\Longrightarrow \quad \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 4}$$

implementation in Python code (version 3.9):

implementation in Python code (version 3.9):

*Method_Model*_params.py

setting of parameters and model definition

implementation in Python code (version 3.9):

*Method_Model*_params.py

setting of parameters and model definition

*Method_Model*_observations.py

generation of observations and plot of results

implementation in Python code (version 3.9):

*Method_Model*_params.py

setting of parameters and model definition

*Method_Model*_observations.py

generation of observations and plot of results

*Method_Model*_da.py

data assimilation cycle and plot of results

implementation in Python code (version 3.9):

*Method_Model*_params.py

setting of parameters and model definition

*Method_Model*_observations.py

generation of observations and plot of results

*Method_Model*_da.py

data assimilation cycle and plot of results

run first *Method_Model*_observations.py, then run *Method_Model*_da.py

motivation

basic methods

BLUE **3DVar**

model and implementation

example results

reconstruction of hidden variables

parameter estimation

Kalman filter

prediction and verification

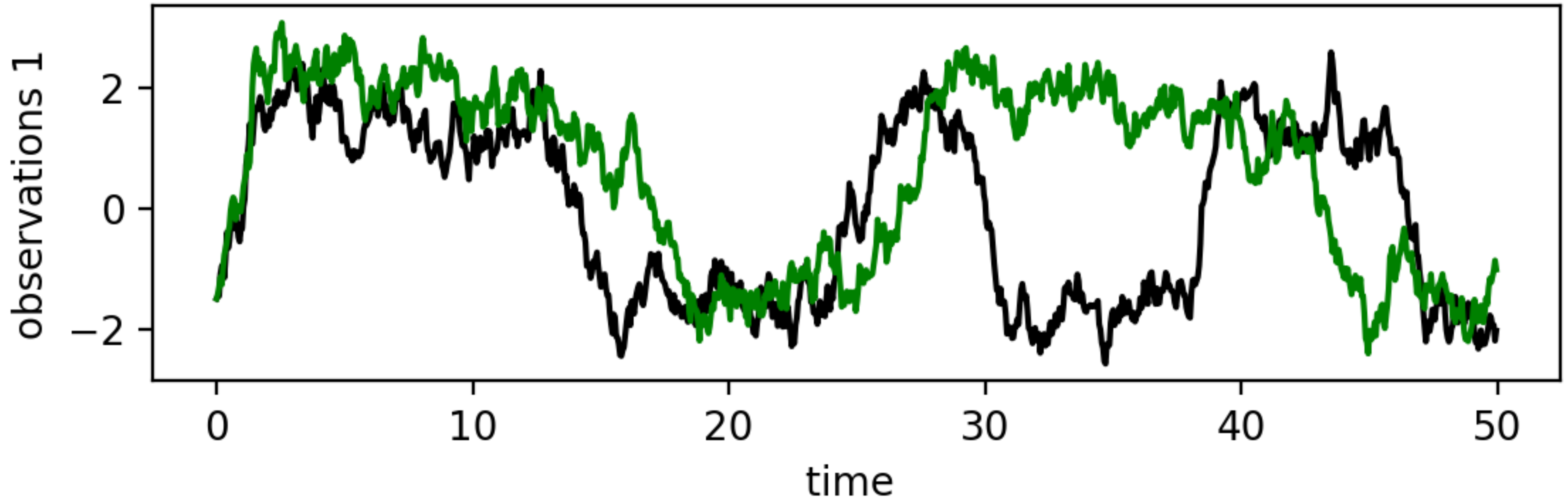
important logic:

the model assumed is not correct, but nevertheless permits to track the system activity important logic:

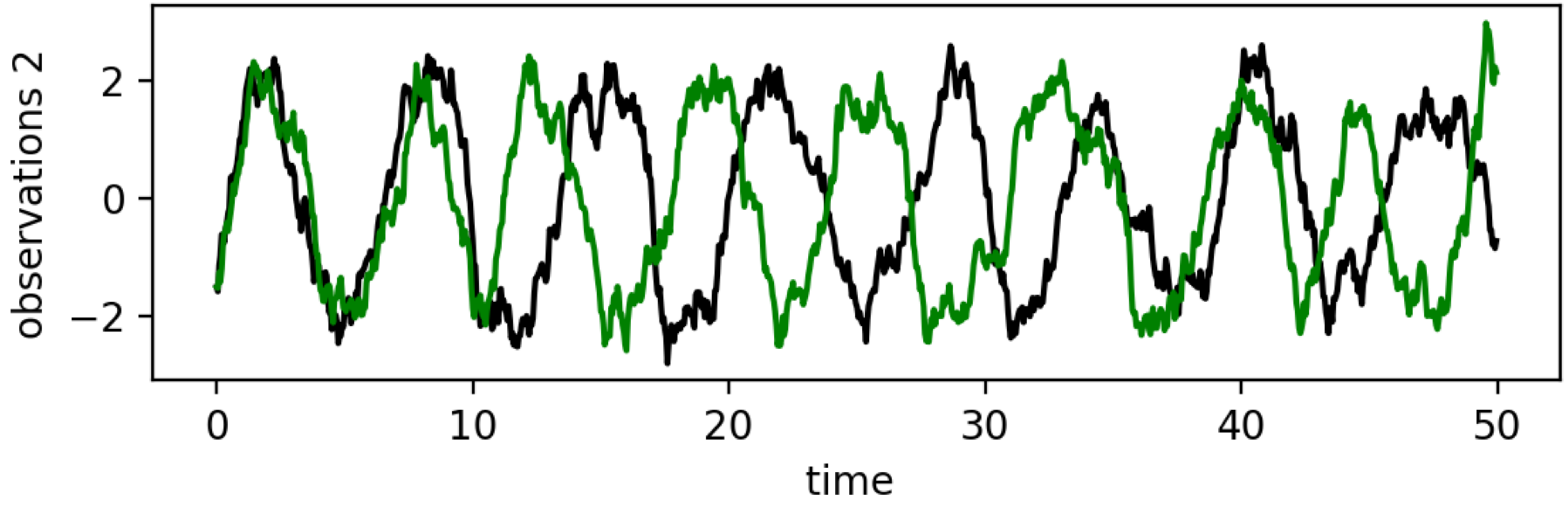
the model assumed is not correct, but nevertheless permits to track the system activity

parameter set 1
(*nature run*)

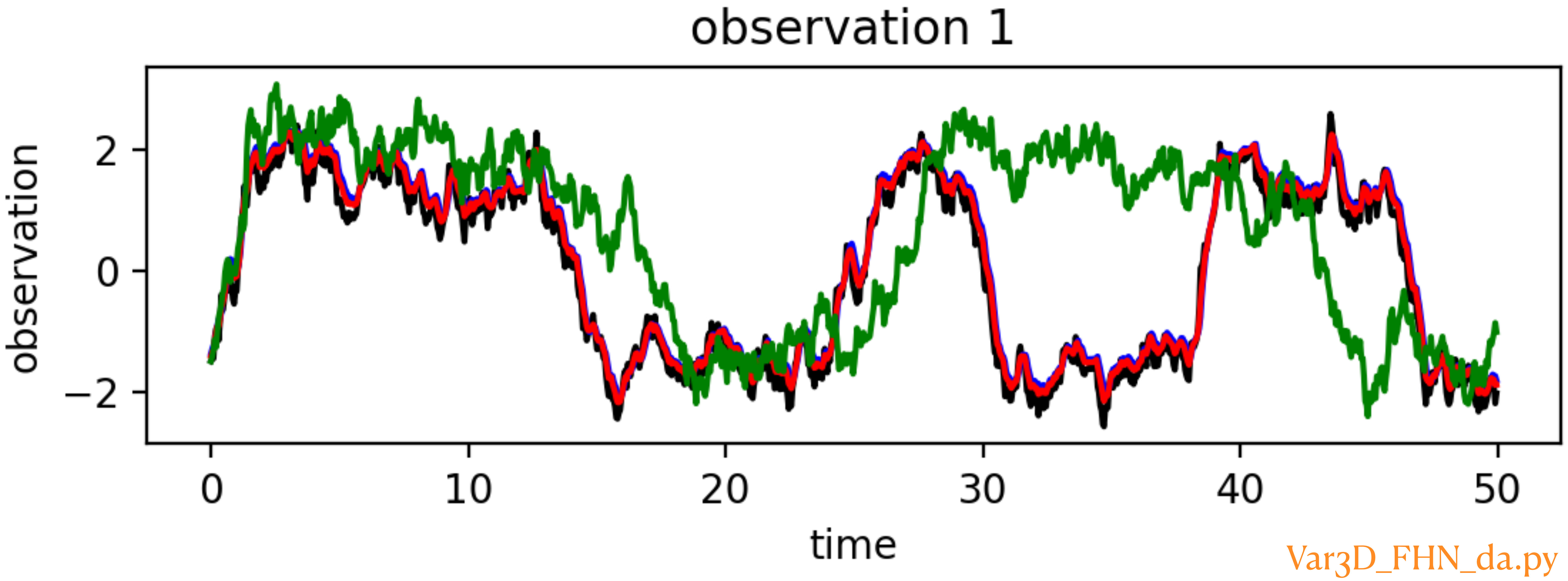
parameter set 2
(false model)



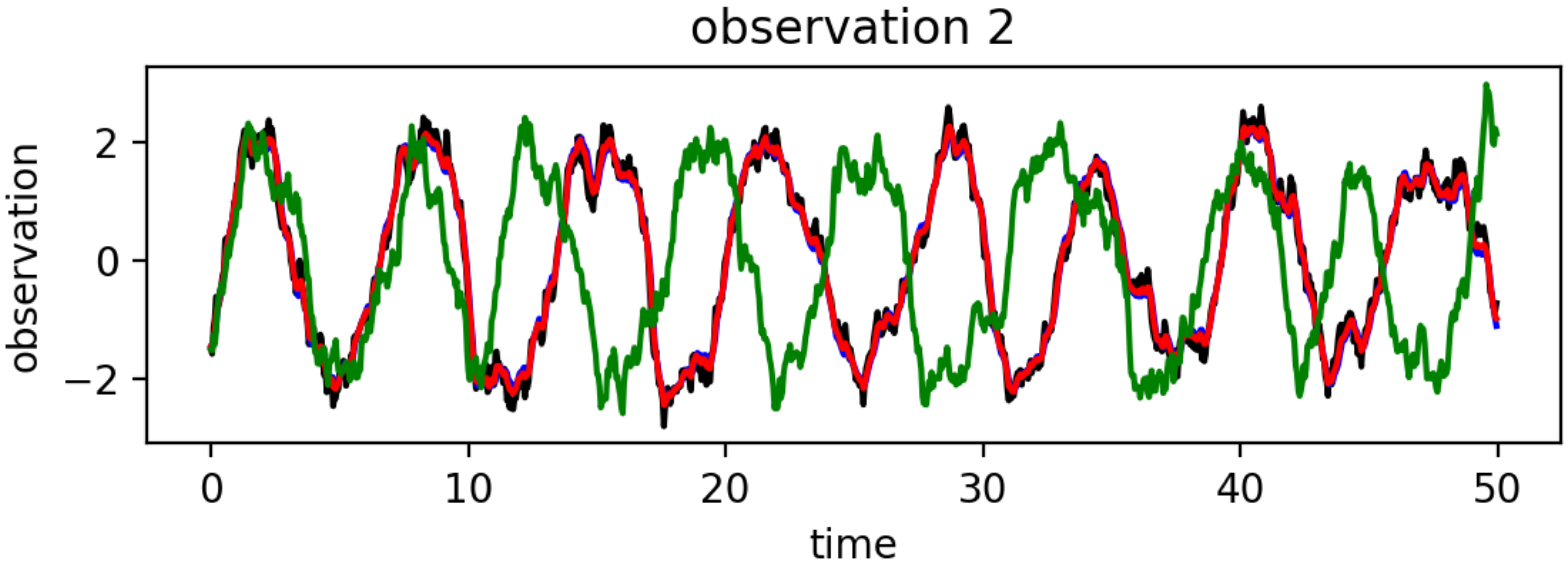
Var3D_FHN_observations.py



first data assimilation results for 3DVar: observation space

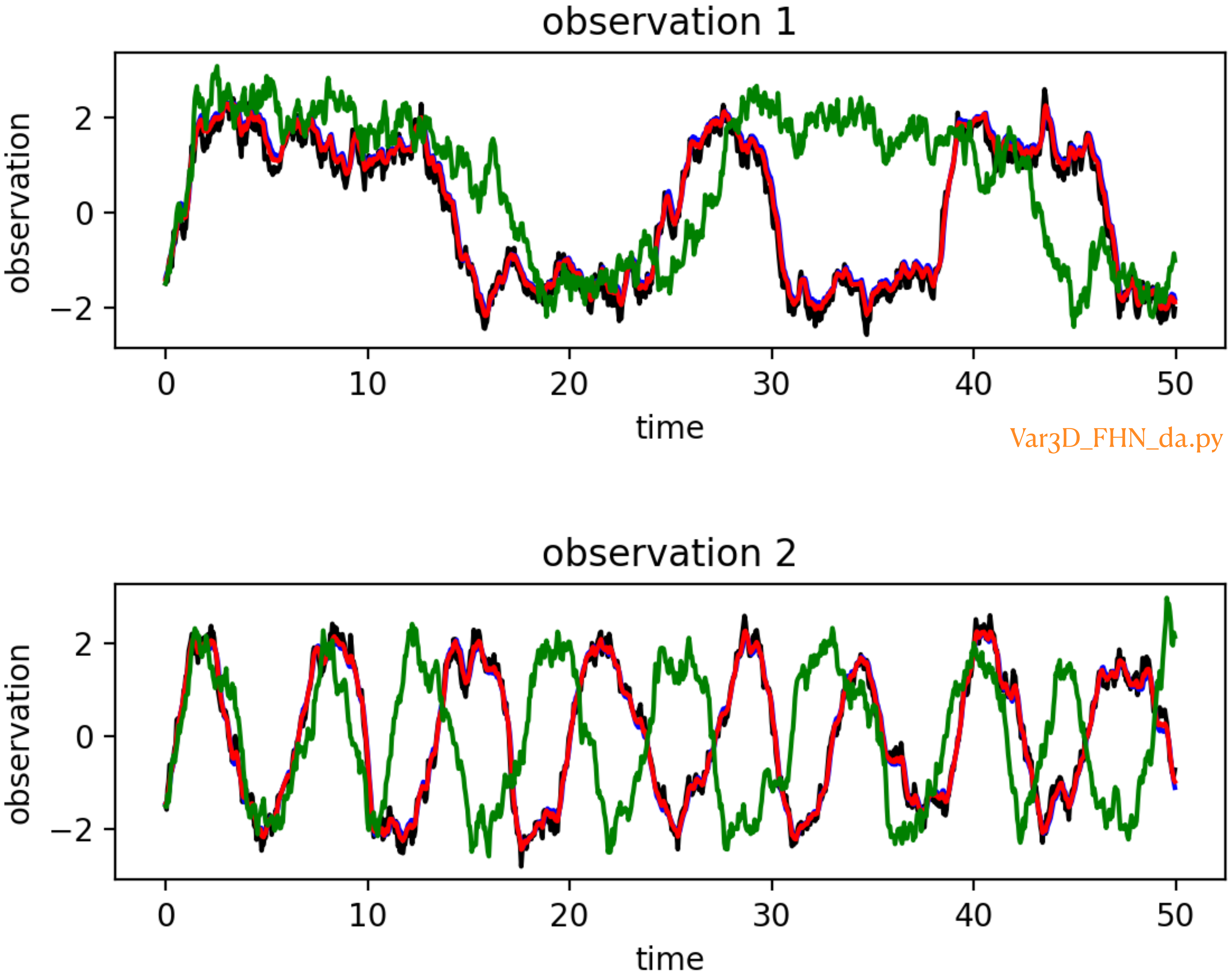


Var3D_FHN_da.py



observations **y**

first data assimilation results for 3DVar: observation space

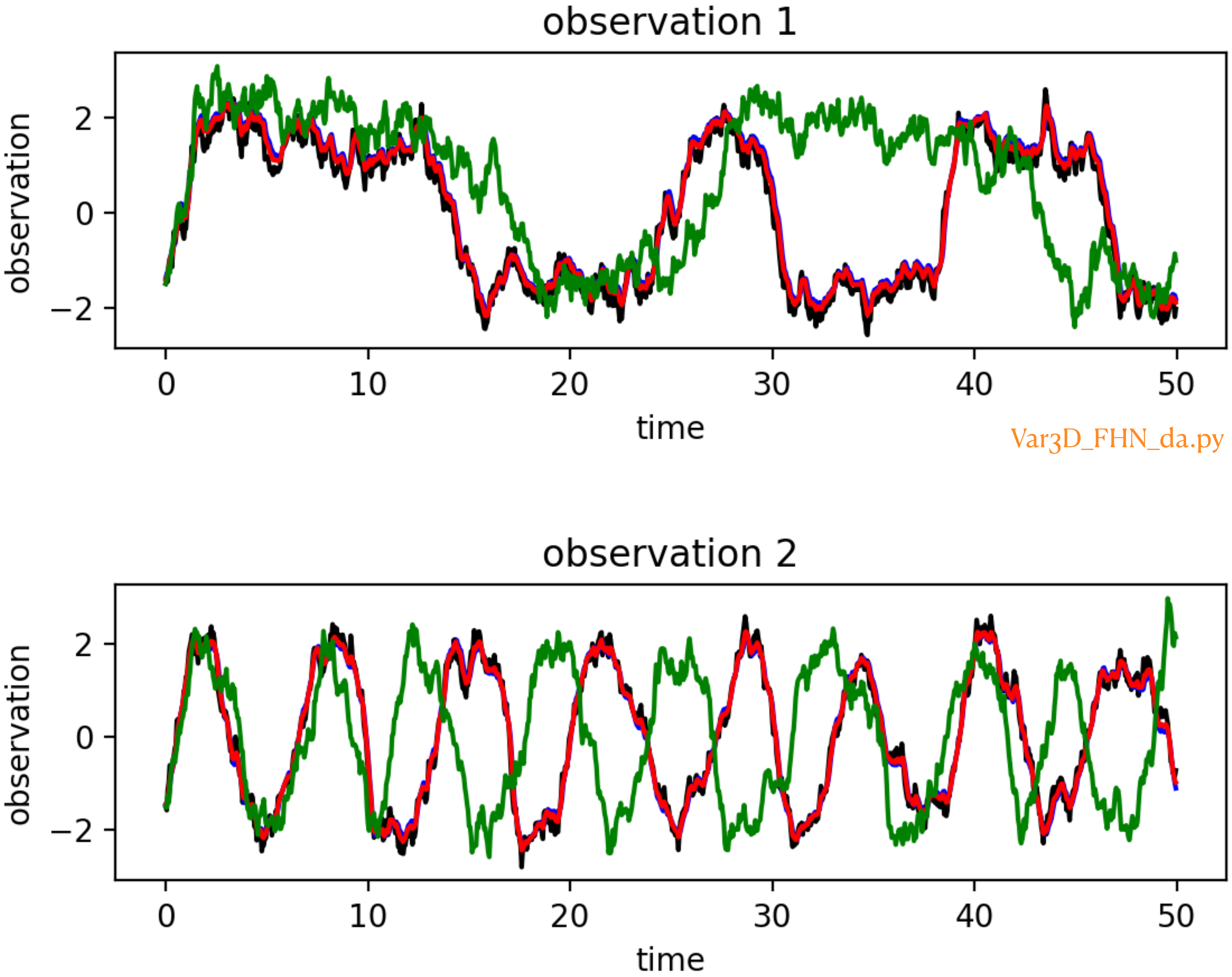


Var3D_FHN_da.py

observations **y**

false observations
from false model

first data assimilation results for 3DVar: observation space

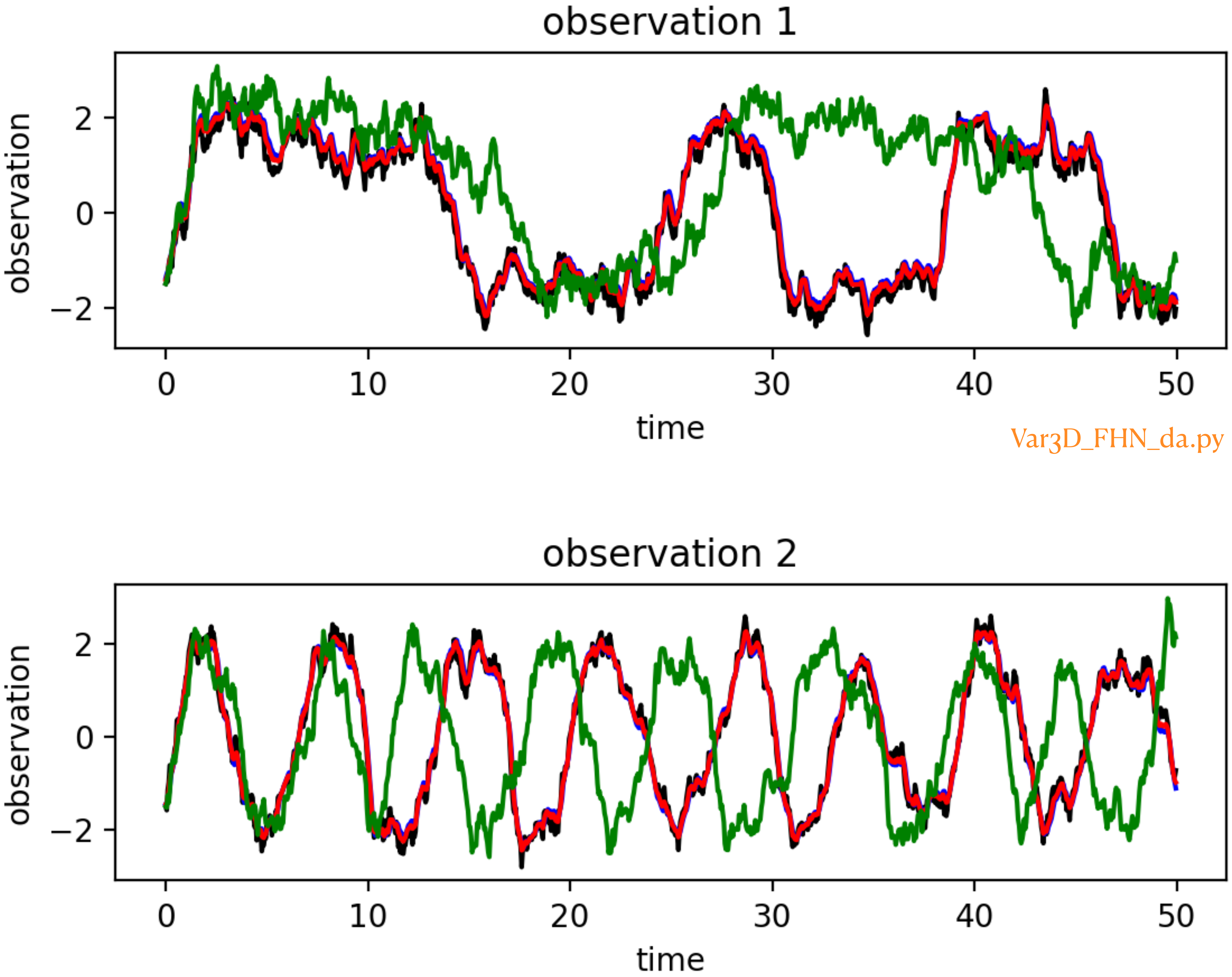


observations $\mathbf{y}$

false observations
from false model

background observations $\mathbf{H}\mathbf{x}_b$

first data assimilation results for 3DVar: observation space



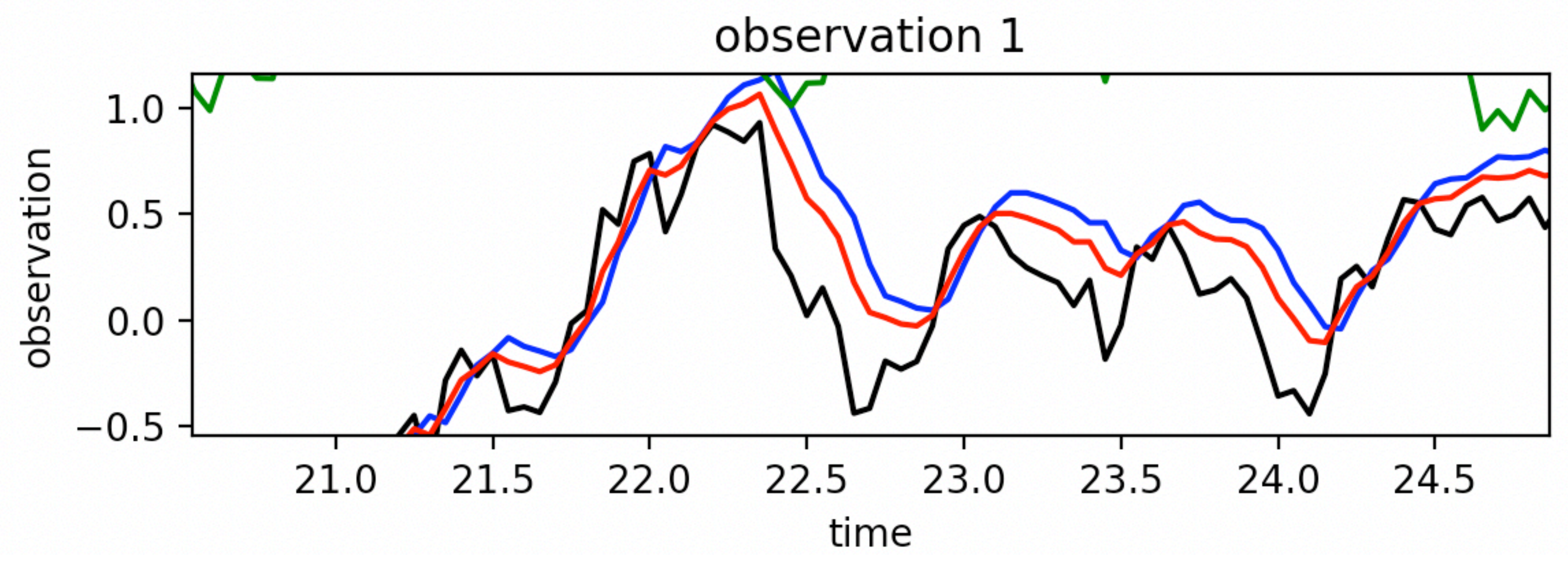
Var3D_FHN_da.py

observations $\mathbf{y}$

false observations from false model

background observations $\mathbf{Hx}_b$

analysis observations $\mathbf{Hx}_a$



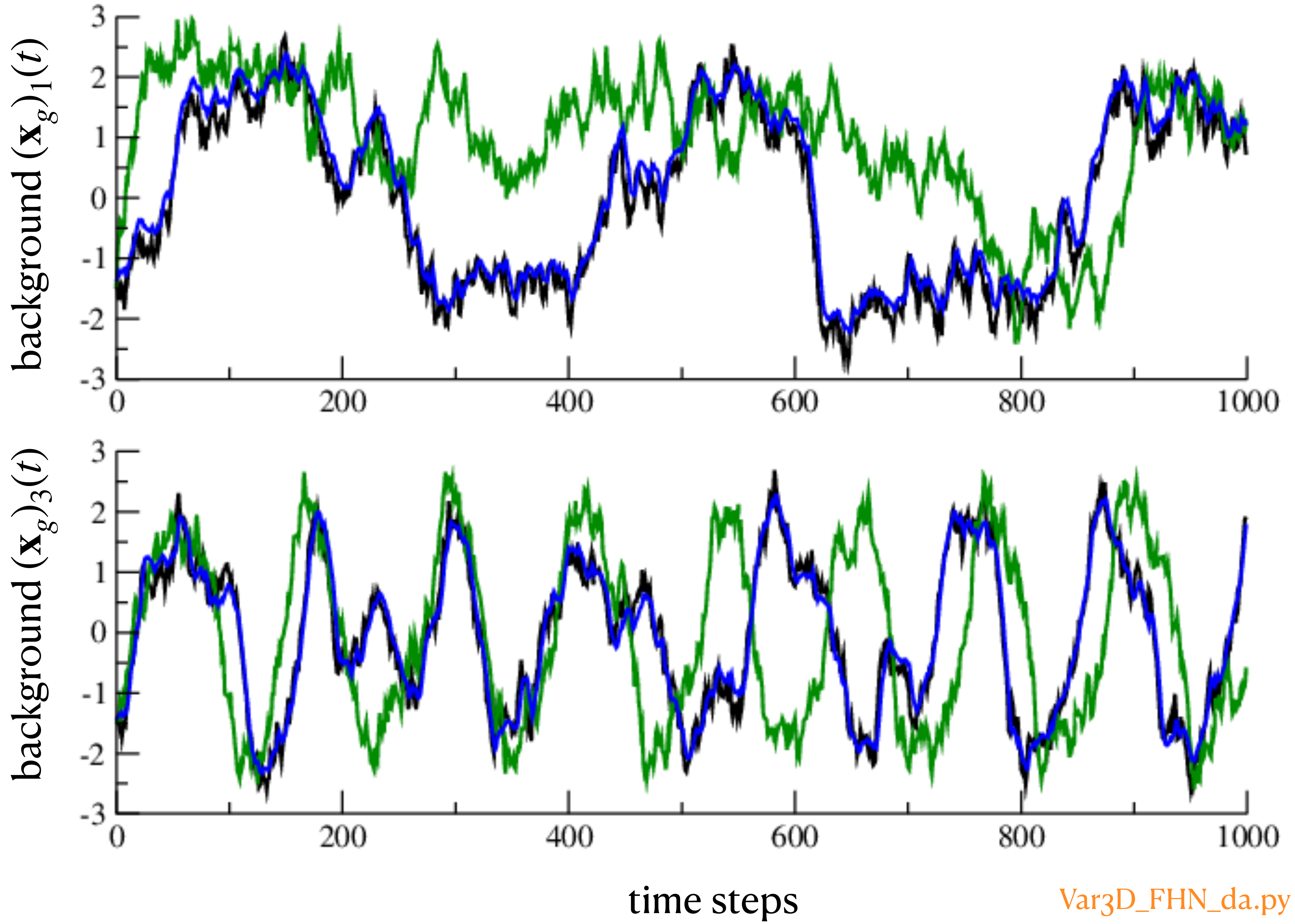
observations $\mathbf{y}$

background observations $\mathbf{Hx}_b$

analysis observations $\mathbf{Hx}_a$

validity check: analysis is always closer to observation than background

first data assimilation results for 3DVar: state space



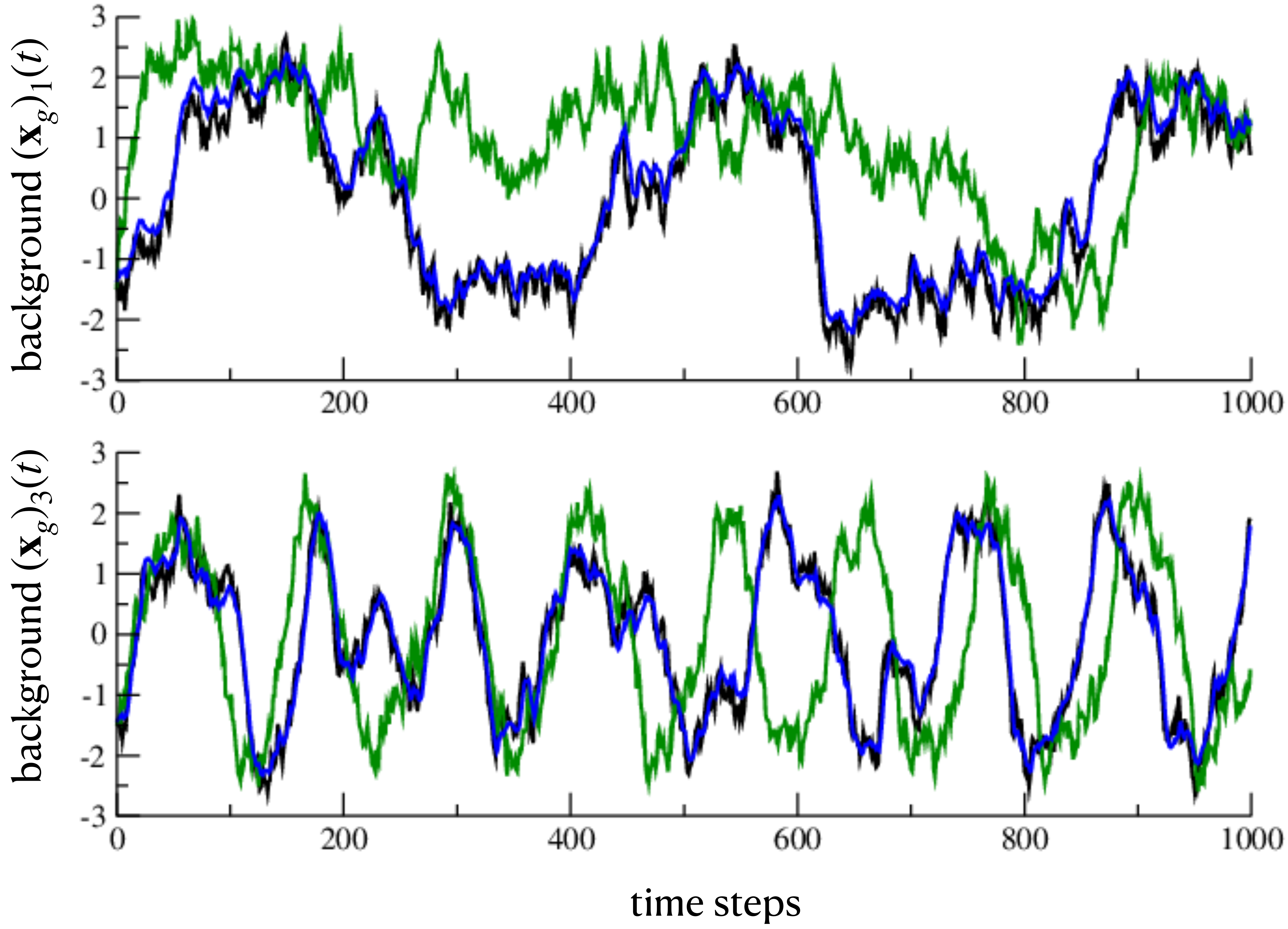
Var3D_FHN_da.py

nature run $\mathbf{x}_b$

false model state

estimated background $\mathbf{x}_b$

first data assimilation results for 3DVar: state space



nature run $\mathbf{x}_b$

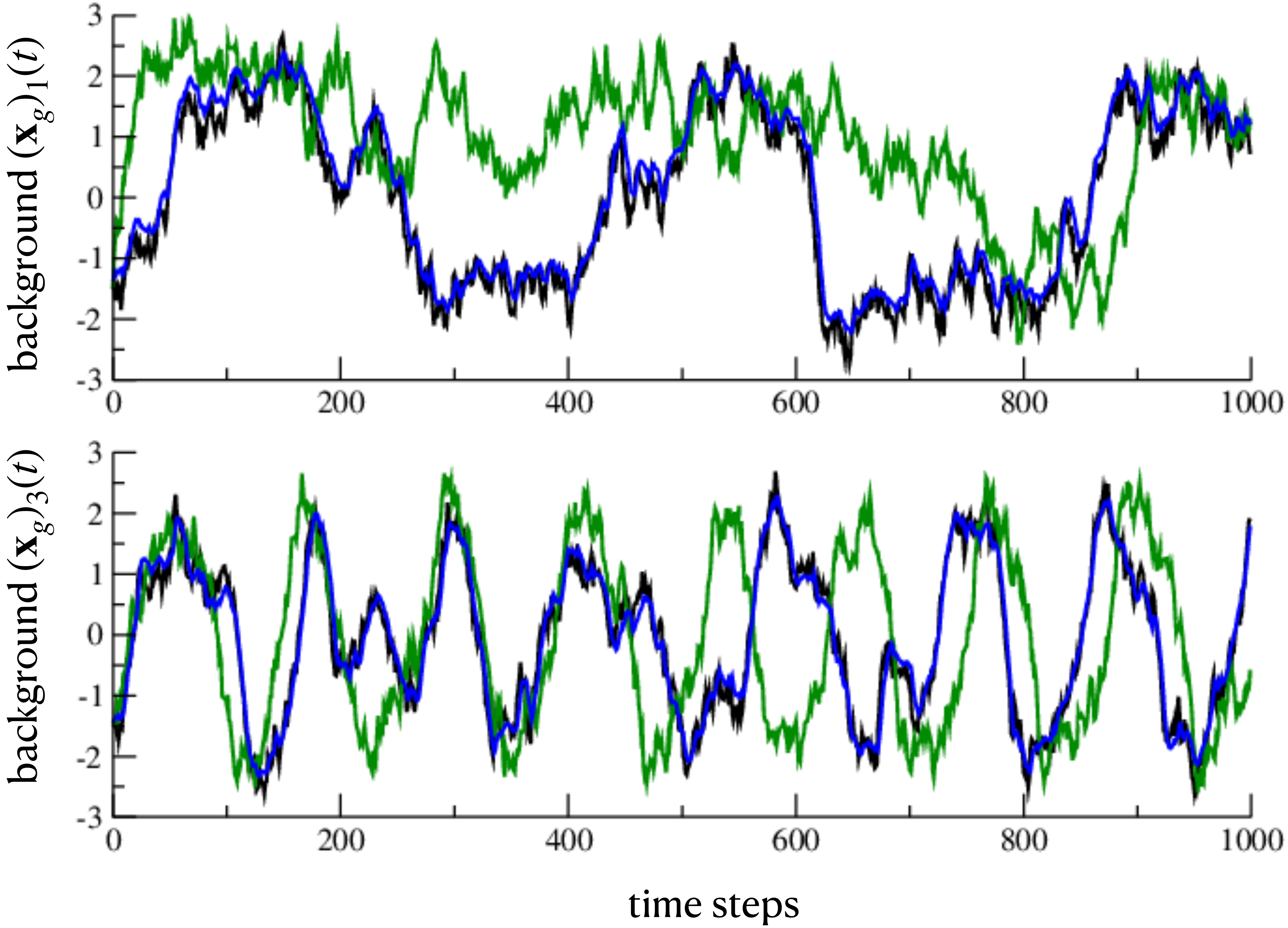
false model state

estimated background $\mathbf{x}_b$

background $\mathbf{x}_b$ represents

short-time prediction of the model

first data assimilation results for 3DVar: state space



nature run $\mathbf{x}_b$

false model state

estimated background $\mathbf{x}_b$

background $\mathbf{x}_b$ represents

short-time prediction of the model

optimal tracking of observed variable

motivation

basic methods

BLUE

3DVar

model and implementation

example results

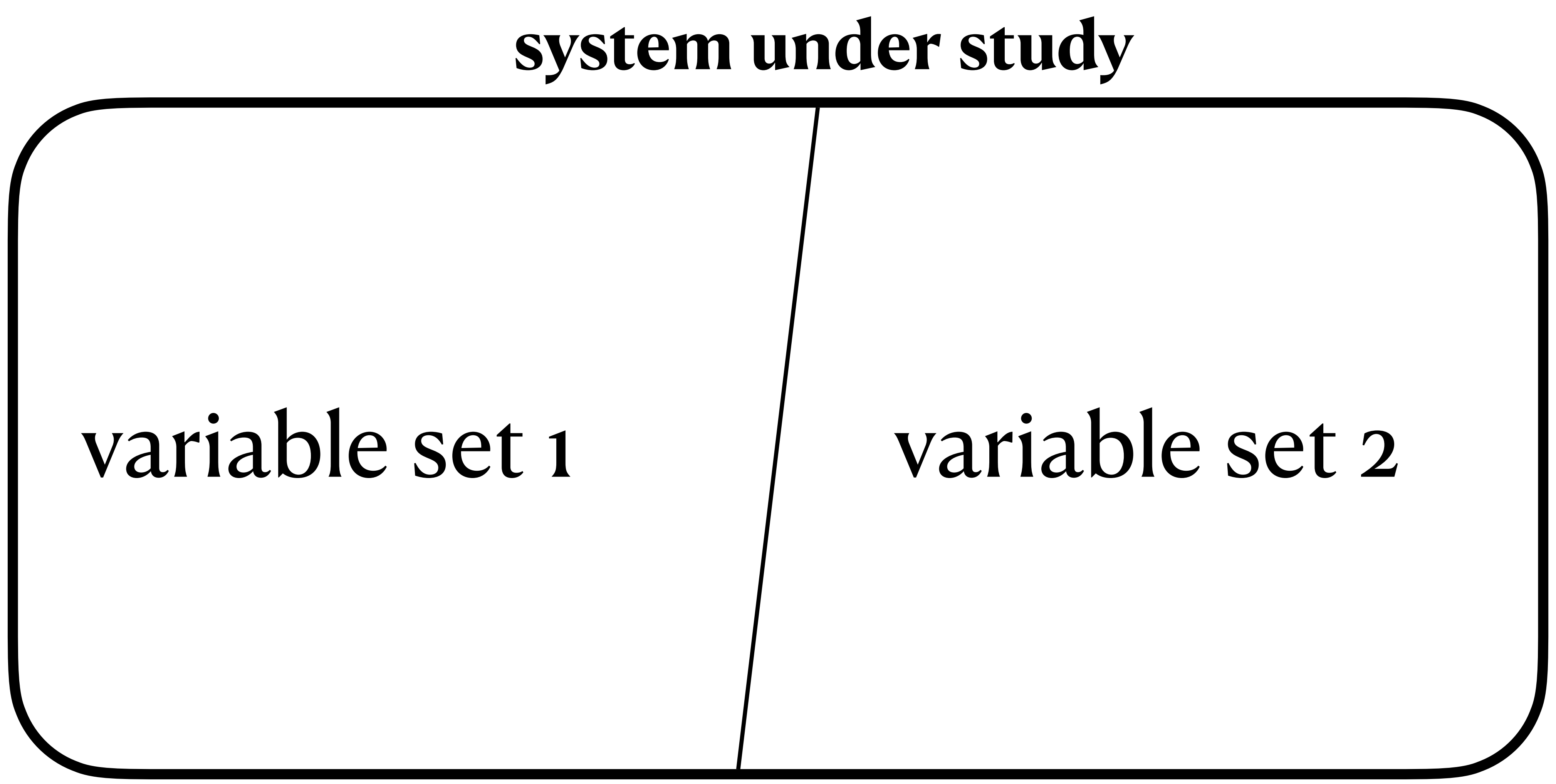
reconstruction of hidden variables

parameter estimation

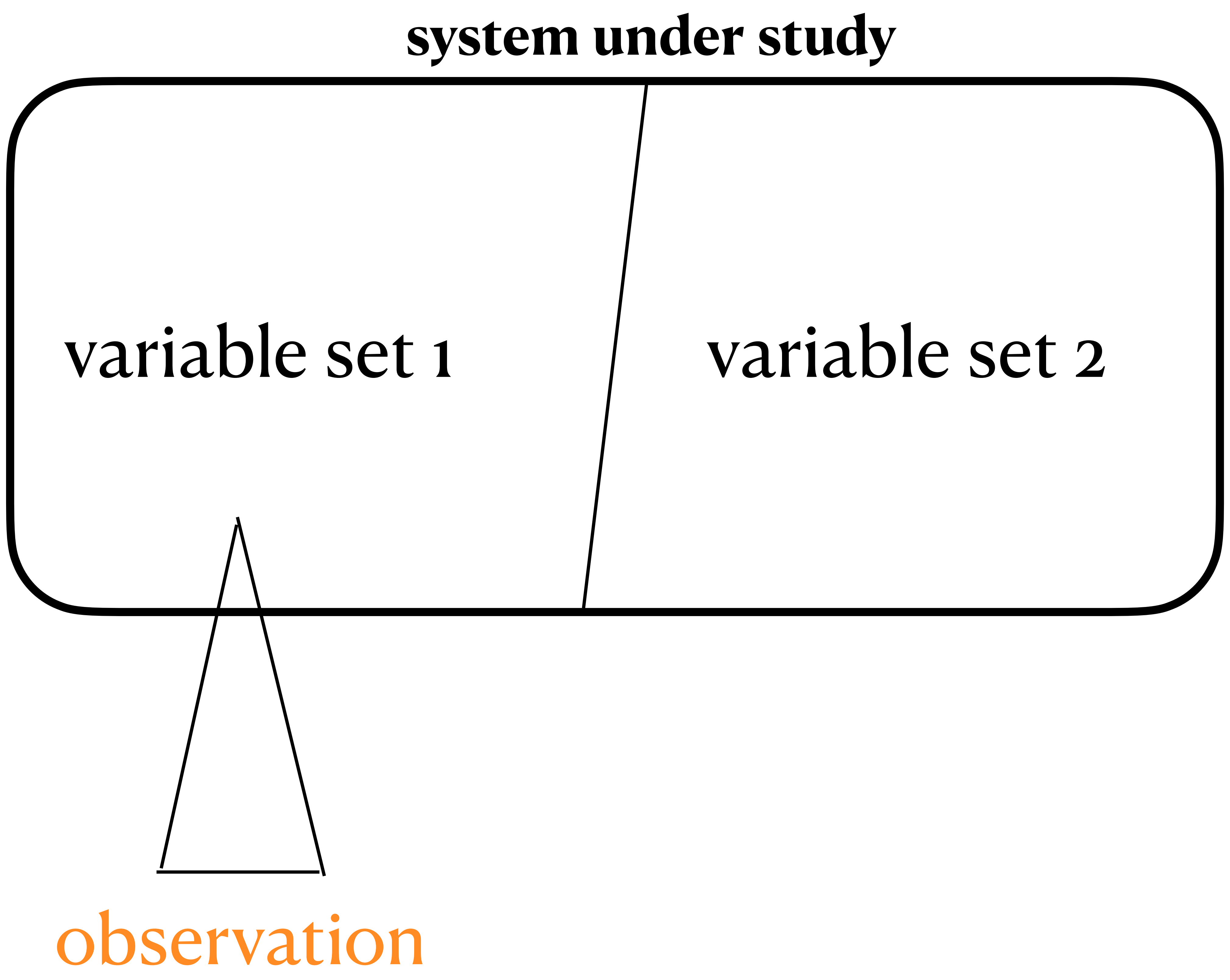
Kalman filter

prediction and verification

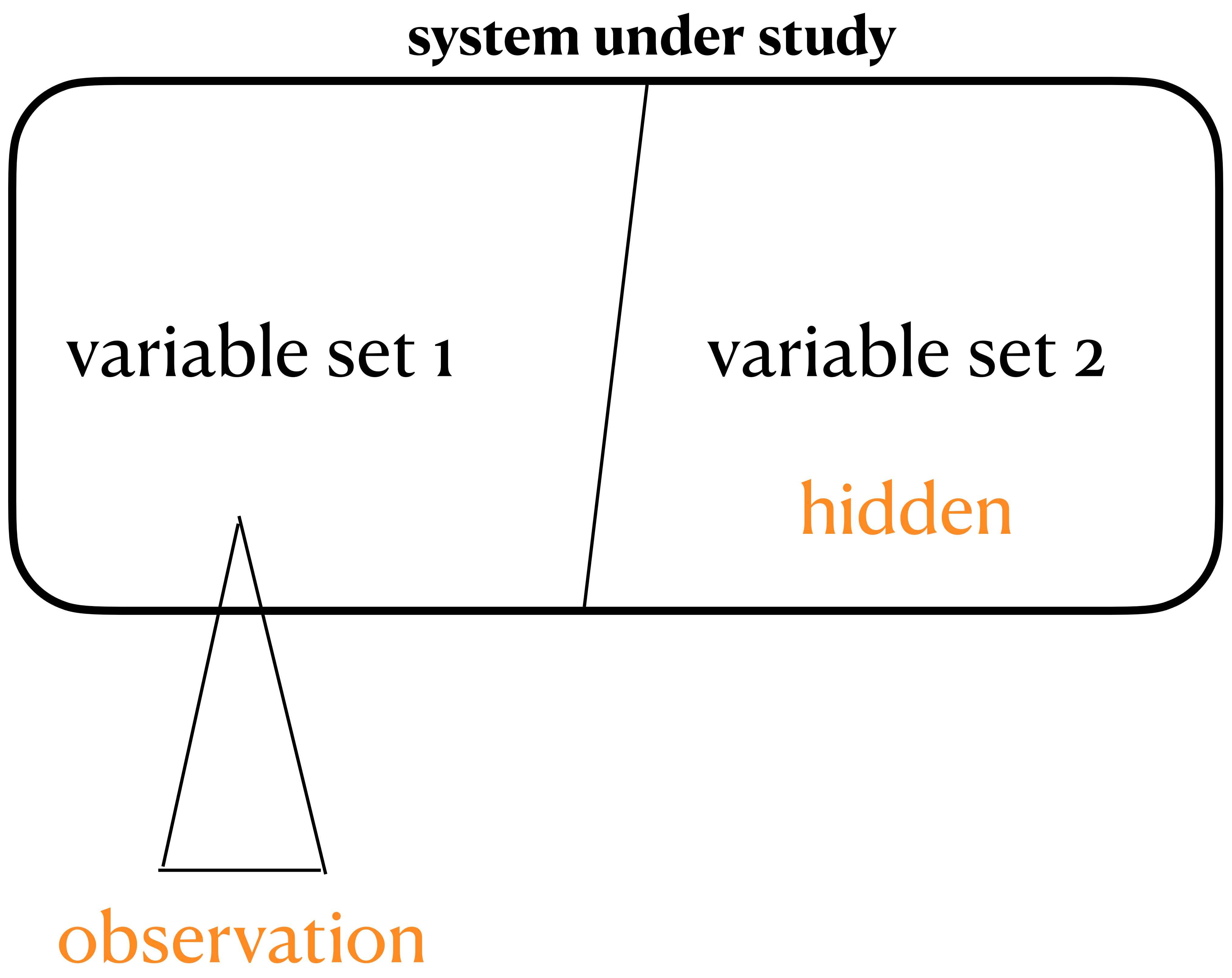
major scientific question



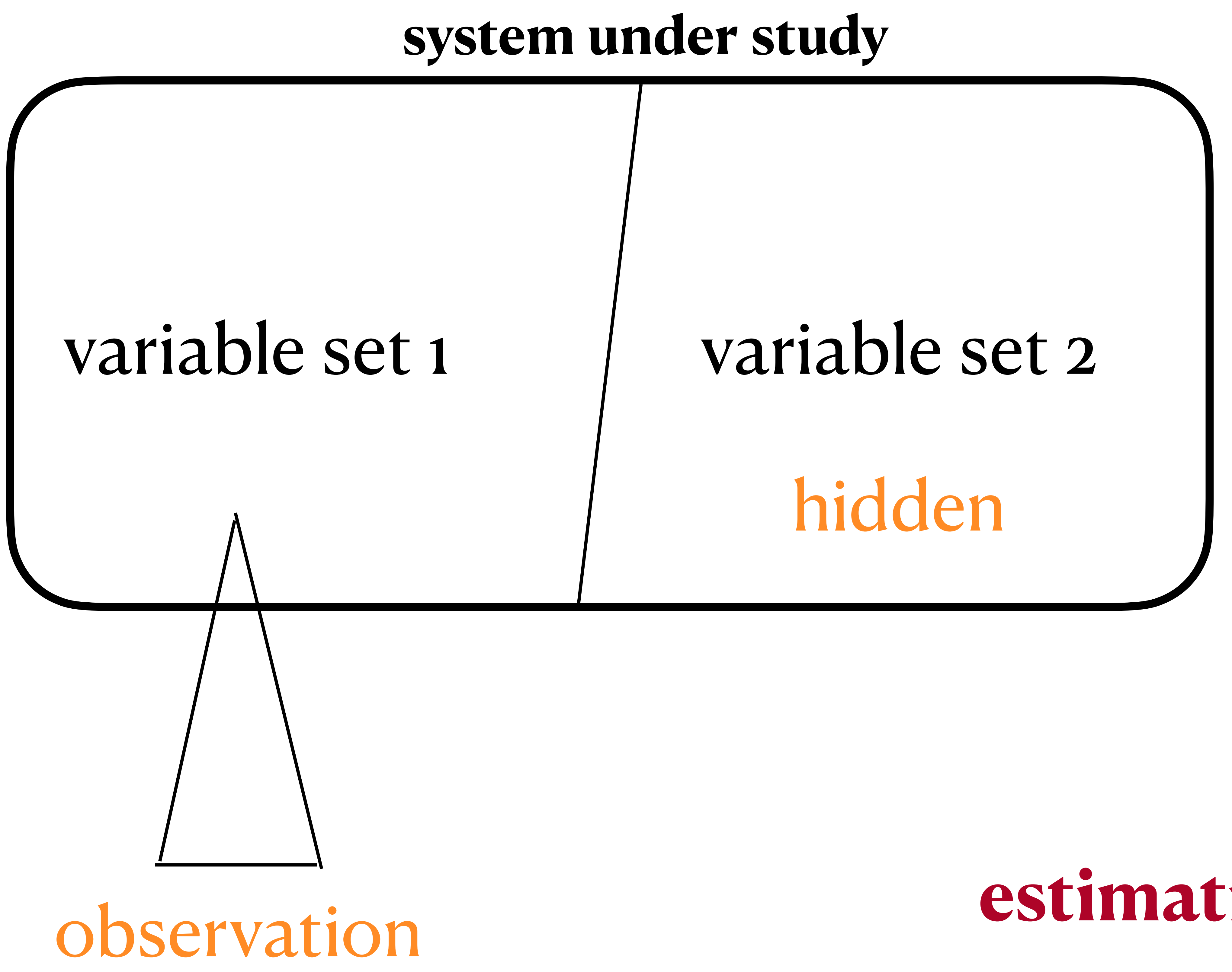
major scientific question



major scientific question

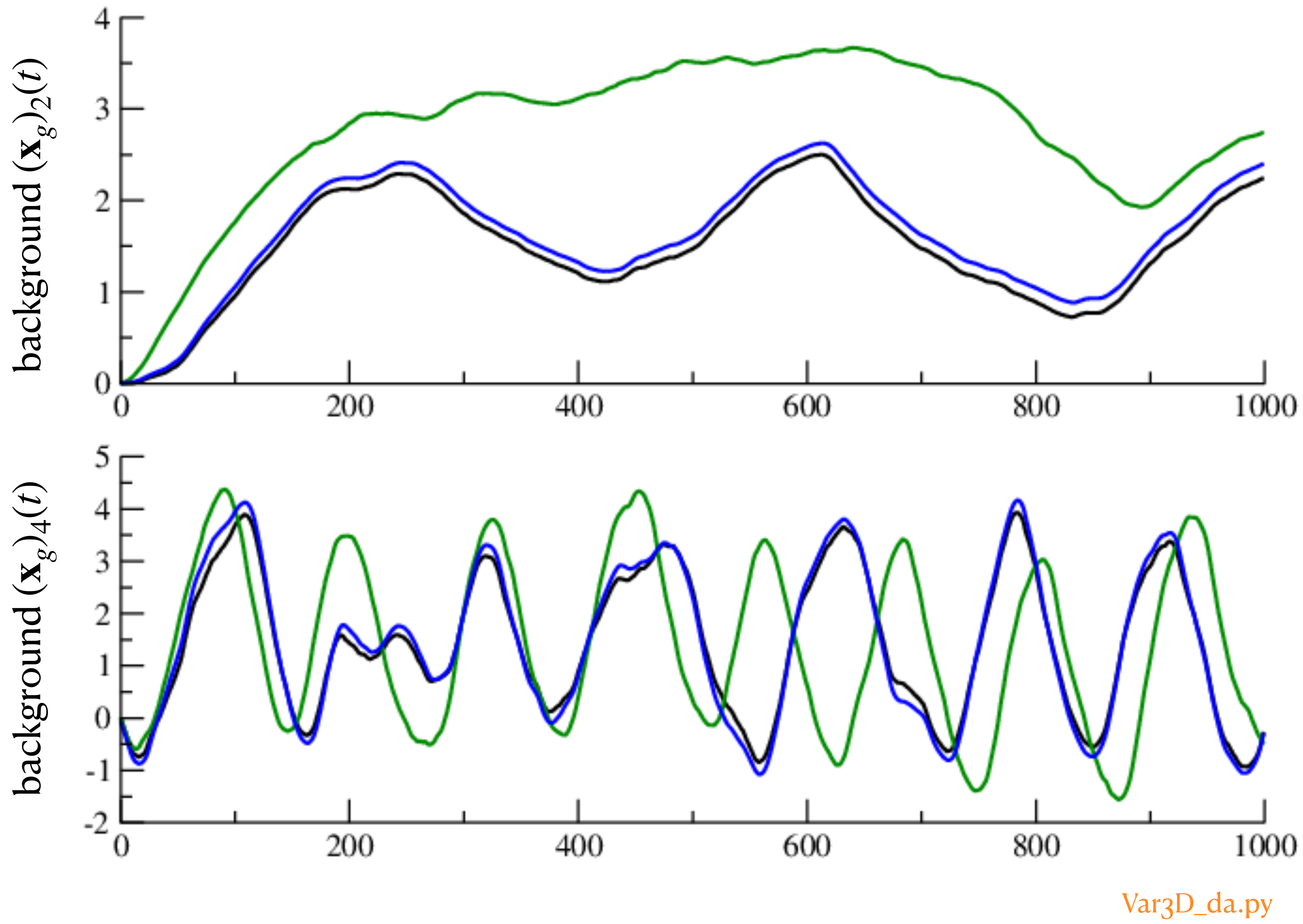


major scientific question



estimation of hidden state variables

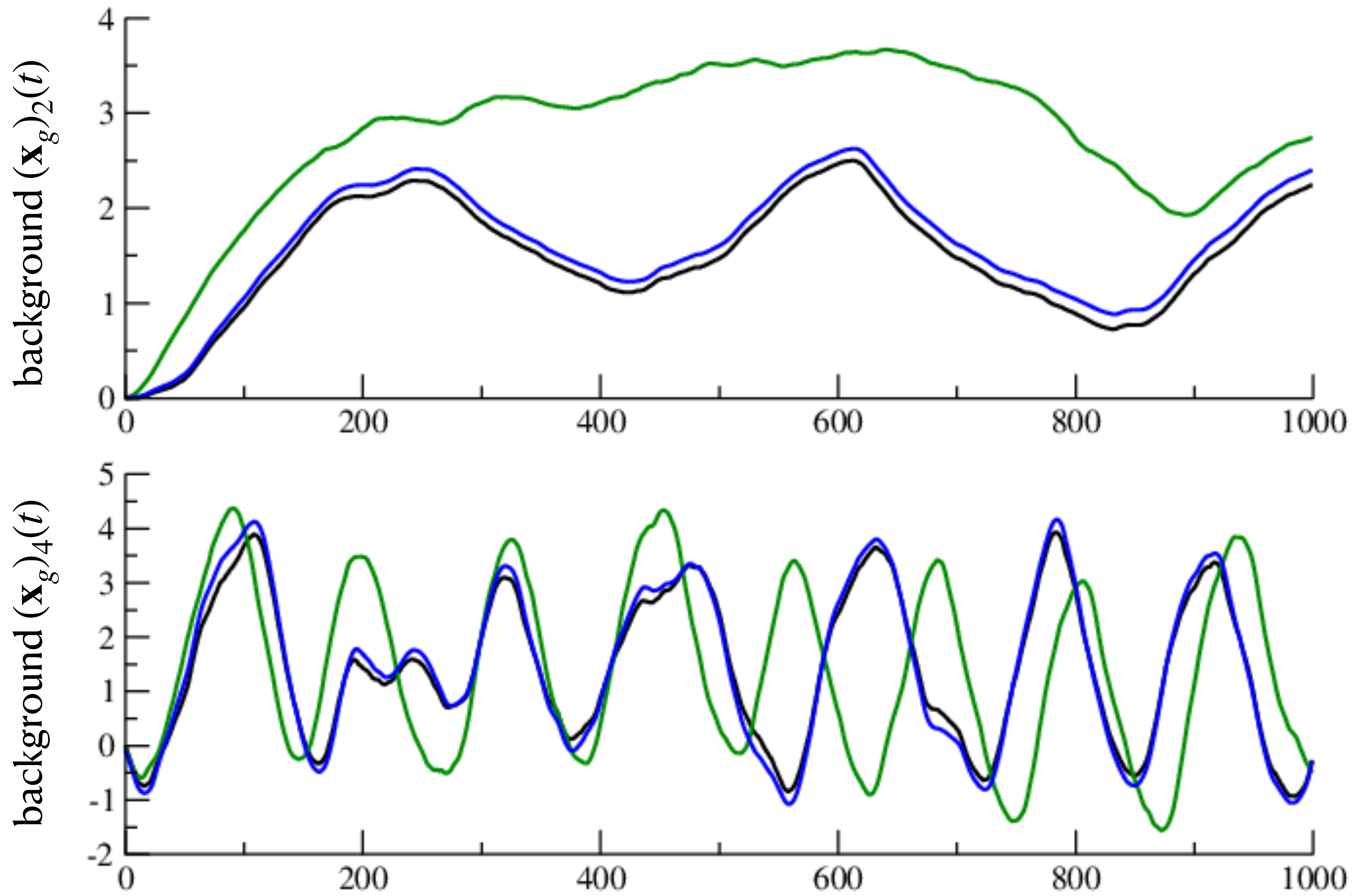
by observation of other state variables ?



nature run $\mathbf{x}_b$

false model state

estimated background $\mathbf{x}_b$



nature run $\mathbf{x}_b$

false model state

estimated background $\mathbf{x}_b$

optimal tracking of hidden variable !!!

motivation

basic methods

BLUE **3DVar**

model and implementation

example results

reconstruction of hidden variables

Kalman filter

parameter estimation

prediction and verification

parameter estimation

now: treating unknown model parameter **a** as hidden variables (*augmented state space*)

$$\mathbf{x}_{n+1}^{\text{model}} = \mathbf{F}(\mathbf{a_n})\mathbf{x}_n^{\text{model}} + \varepsilon$$

$$\mathbf{a}_{n+1} = \mathbf{a}_n$$

parameter estimation

now: treating unknown model parameter **a** as hidden variables (*augmented state space*)

$$\mathbf{x}_{n+1}^{\text{model}} = \mathbf{F}(\mathbf{a_n})\mathbf{x}_n^{\text{model}} + \varepsilon$$
$$\mathbf{a}_{n+1} = \mathbf{a}_n$$

$$\rightarrow \quad \mathbf{x}_{n+1} = \mathcal{M}\mathbf{x}_n$$

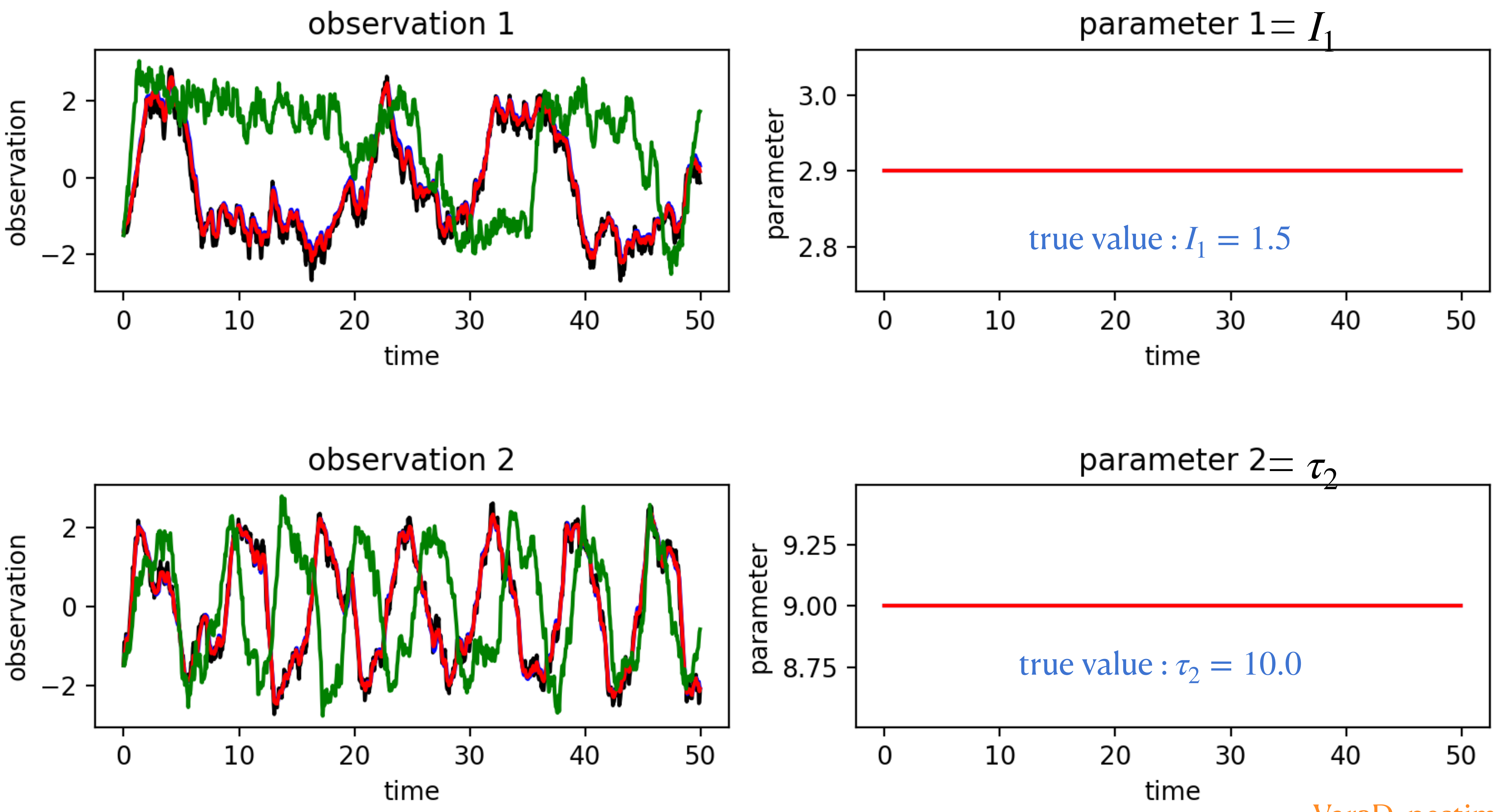
with $\mathbf{x}_n = \begin{pmatrix} \mathbf{x}_n^{\text{model}} \\ \mathbf{a}_n \end{pmatrix} \quad \mathcal{M}\mathbf{x} = \begin{pmatrix} \mathbf{M}(\mathbf{a}_n)\mathbf{x}^{\text{model}} \\ \mathbf{a}_n \end{pmatrix}$

typically, model+parameters is nonlinear

parameter estimation

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$



Var3D_pestim_da.py

no parameter estimation

reason:

if $\mathbf{B}$ has the form $\mathbf{B} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ S & S & S & S & P & P \\ S & S & S & S & P & P \end{pmatrix}$ with $S, P > 0$ then $\mathbf{K} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ S\cdot & S\cdot \\ S\cdot & S\cdot \end{pmatrix}$

reason:

if $\mathbf{B}$ has the form $\mathbf{B} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ S & S & S & S & P & P \\ S & S & S & S & P & P \end{pmatrix}$ with $S, P > 0$ then $\mathbf{K} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ S\cdot & S\cdot \\ S\cdot & S\cdot \end{pmatrix}$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \mathbf{K} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

reason:

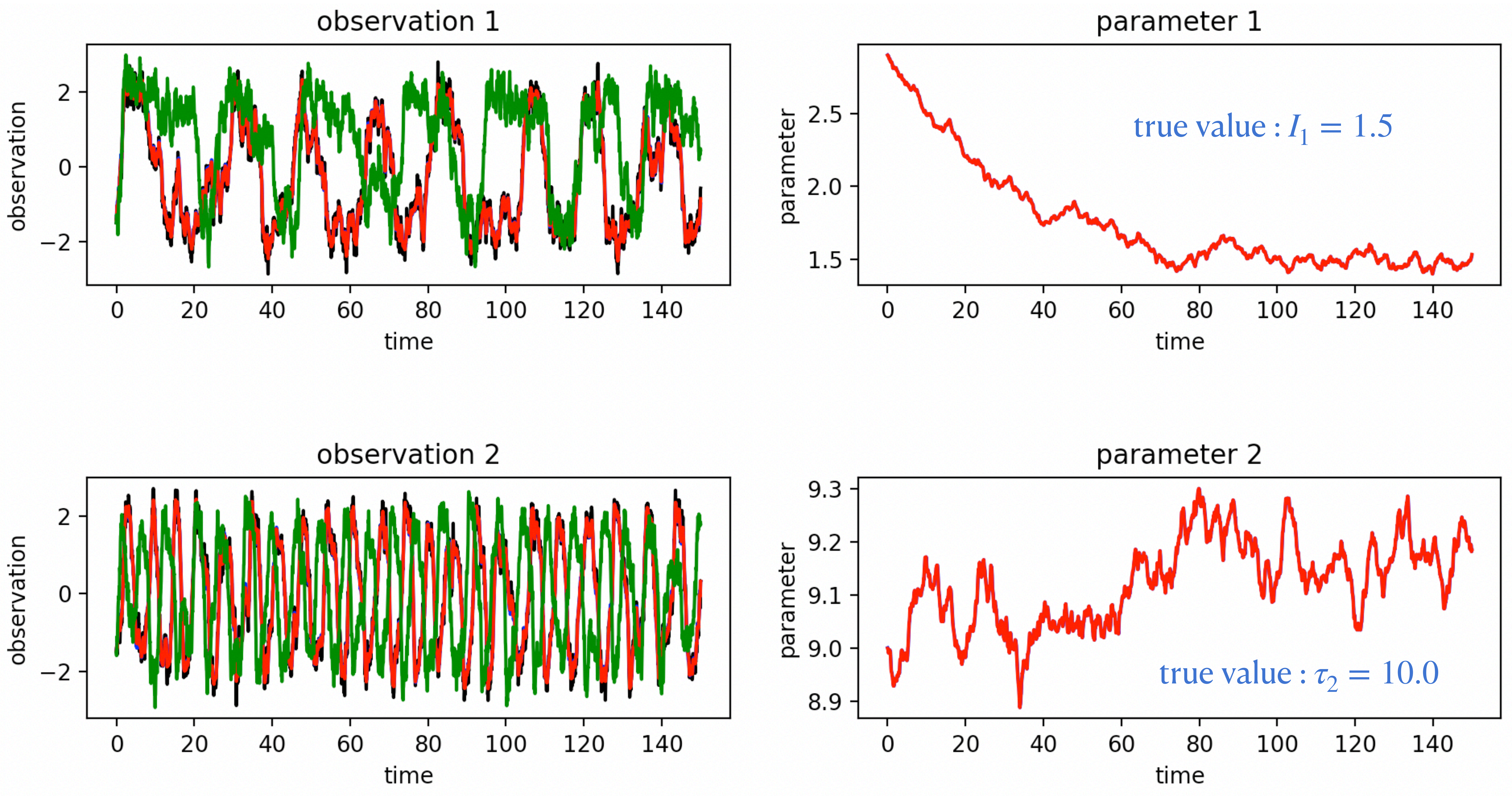
if $\mathbf{B}$ has the form $\mathbf{B} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ \cdot & \cdot & \cdot & \cdot & S & S \\ S & S & S & S & P & P \\ S & S & S & S & P & P \end{pmatrix}$ with $S, P > 0$ then $\mathbf{K} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ S\cdot & S\cdot \\ S\cdot & S\cdot \end{pmatrix}$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \Longrightarrow \quad \mathbf{K} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

parameter estimation possible only

if covariance between state and parameters

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.05 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.05 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.05 & 0 & 0 & 1 \end{pmatrix}$$



Var3D_pestim_da.py

good parameter estimation

linear 3DVar

<u>advantage:</u>

- simple to implement

- numerically fast
  since analysis is given analytically

linear 3DVar

advantage:

- simple to implement
- numerically fast
  since analysis is given analytically

disadvantage:

- assumes linear observation operator
- assumes constant **B**

for nonlinear observation operator $H(\mathbf{x}_b)$:

for nonlinear observation operator $H(\mathbf{x}_b)$:

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{B}$$

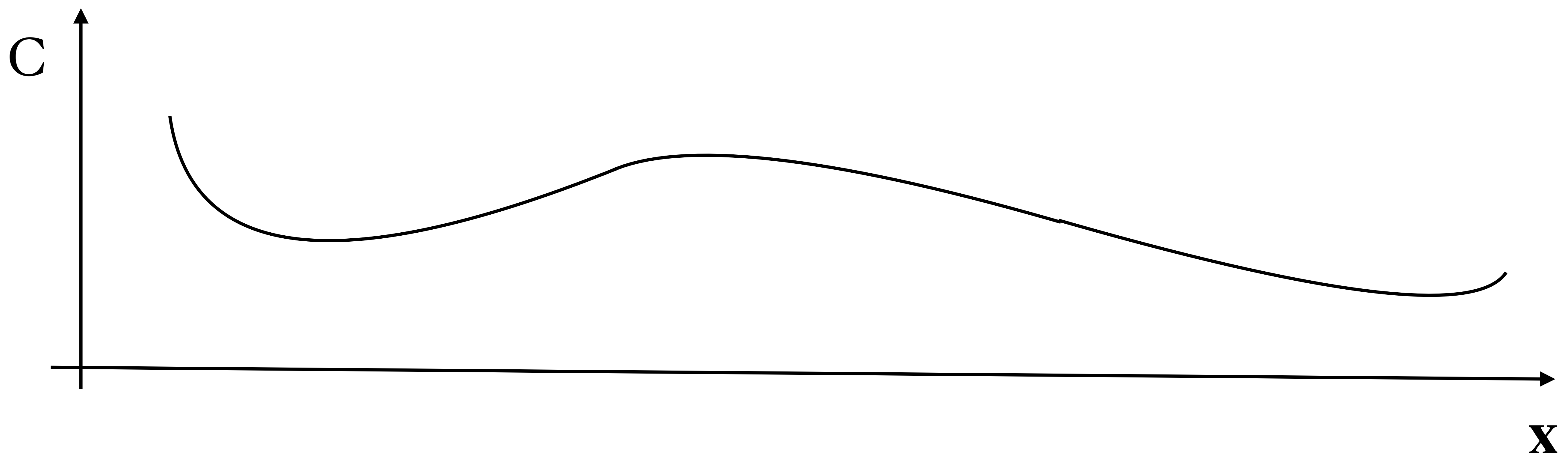
$$\mathbf{y}(t_n) = H(\mathbf{x}_b(t_n)) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R}$$

for nonlinear observation operator $H(\mathbf{x}_b)$:

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{B}$$

$$\mathbf{y}(t_n) = H(\mathbf{x}_b(t_n)) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R}$$

$$C(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^t\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + (\mathbf{y} - H(\mathbf{x}))^t\mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

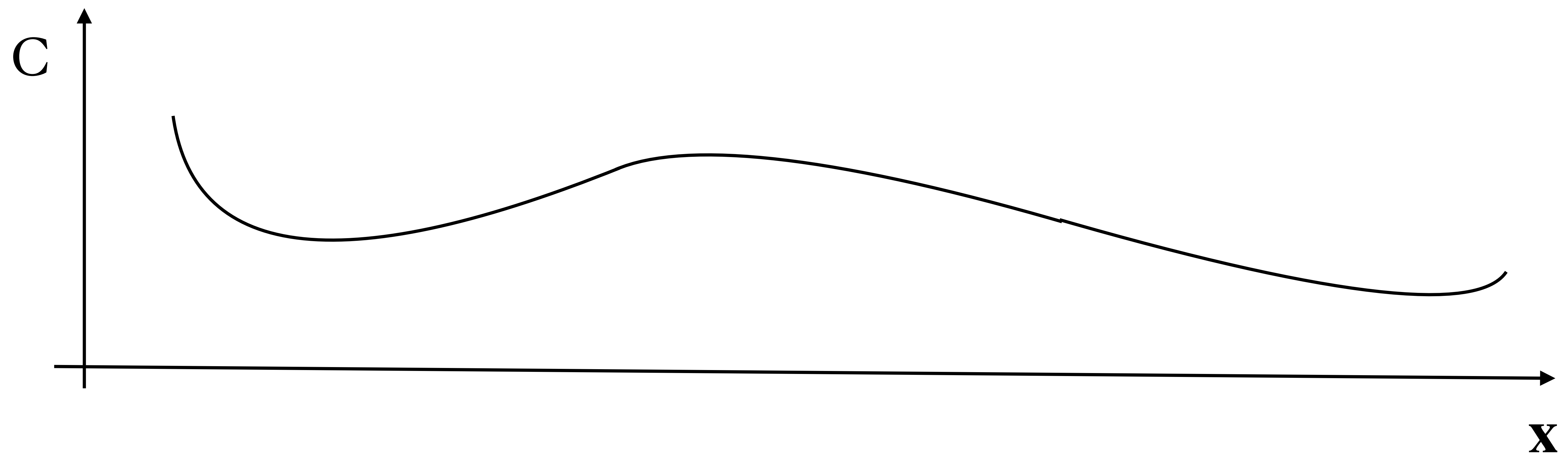


for nonlinear observation operator $H(\mathbf{x}_b)$:

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{B}$$

$$\mathbf{y}(t_n) = H(\mathbf{x}_b(t_n)) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R}$$

$$C(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^t\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + (\mathbf{y} - H(\mathbf{x}))^t\mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$



numerical estimation of **global minimum**, e.g. by gradient descent

nonlinear 3DVar

<u>advantage:</u>

- valid for nonlinear model and observation operator

nonlinear 3DVar

<u>advantage:</u>

- valid for nonlinear model and observation operator

<u>disadvantage:</u>

- numerically slow and analysis search is complex
- assumes constant $\mathbf{B}$

motivation

basic methods

Kalman filter

prediction and verification

motivation

basic methods

Kalman filter

linear EKF UKF ETKF LETKF

prediction and verification

Kalman Filter: The probability picture

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{Q} \qquad E[\varepsilon] = 0$$

$$\mathbf{y}(t_n) = \mathbf{H}\mathbf{x}_b(t_n) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R} \qquad E[\eta] = 0$$

Kalman Filter: The probability picture

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{Q} \qquad\qquad E[\varepsilon] = 0$$

$$\mathbf{y}(t_n) = \mathbf{H}\mathbf{x}_b(t_n) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R} \qquad\qquad E[\eta] = 0$$

In fact, there is a true system state that we want to estimate: $\mathbf{x}_t$

estimate of this true system state: $\mathbf{x}_b$

estimate of true observation: $\mathbf{H}\mathbf{x}_b$

Kalman Filter: The probability picture

$$\mathbf{x}_b(t_n) = \mathcal{M}\mathbf{x}_b(t_{n-1}) + \varepsilon \quad , \quad E[\varepsilon\varepsilon^t] = \mathbf{Q} \qquad E[\varepsilon] = 0$$

$$\mathbf{y}(t_n) = \mathbf{H}\mathbf{x}_b(t_n) + \eta \quad , \quad E[\eta\eta^t] = \mathbf{R} \qquad E[\eta] = 0$$

In fact, there is a true system state that we want to estimate: $\mathbf{x}_t$

estimate of this true system state: $\mathbf{x}_b$

estimate of true observation: $\mathbf{H}\mathbf{x}_b$

error of system state: $\varepsilon_b = \mathbf{x}_b - \mathbf{x}_t$

error of observation innovation : $\varepsilon_o = \mathbf{y} - \mathbf{H}\mathbf{x}_t$

error of estimated analysis : $\varepsilon_a = \mathbf{x}_a - \mathbf{H}\mathbf{x}_t$

we define: $E[\varepsilon_b \varepsilon_b^t] = \mathbf{P}_b$ $E[\varepsilon_o \varepsilon_o^t] = \mathbf{R}$

we define: $E[\varepsilon_b \varepsilon_b^t] = \mathbf{P}_b$ $\quad E[\varepsilon_o \varepsilon_o^t] = \mathbf{R}$

then we find:

$$\begin{aligned} E[\varepsilon_a \varepsilon_a^t] &= (\mathbf{I} - \mathbf{KH})\mathbf{P}_b \\ &= \mathbf{P}_a \end{aligned}$$

covariance of analysis

we define: $E[\varepsilon_b \varepsilon_b^t] = \mathbf{P}_b$ $\quad E[\varepsilon_o \varepsilon_o^t] = \mathbf{R}$

then we find: $$E[\varepsilon_a \varepsilon_a^t] = (\mathbf{I} - \mathbf{KH})\mathbf{P}_b$$
$$= \mathbf{P}_a$$ covariance of analysis

<u>interpretation:</u> $\mathbf{x}_b$ is Gaussian distributed with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$

we define: $E[\varepsilon_b \varepsilon_b^t] = \mathbf{P}_b$ $E[\varepsilon_o \varepsilon_o^t] = \mathbf{R}$

then we find:
$$E[\varepsilon_a \varepsilon_a^t] = (\mathbf{I} - \mathbf{KH})\mathbf{P}_b$$
$$= \mathbf{P}_a$$
covariance of analysis

<u>interpretation:</u> $\mathbf{x}_b$ is Gaussian distributed with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$

$\mathbf{y} - \mathbf{H}\mathbf{x}_b$ is Gaussian distributed with a mean $\mathbf{y} - \bar{\mathbf{H}}\mathbf{x}_b$ and covariance $\mathbf{R}$

we define: $E[\varepsilon_b \varepsilon_b^t] = \mathbf{P}_b$ $E[\varepsilon_o \varepsilon_o^t] = \mathbf{R}$

then we find:
$$\begin{aligned} E[\varepsilon_a \varepsilon_a^t] &= (\mathbf{I} - \mathbf{KH})\mathbf{P}_b \\ &= \mathbf{P}_a \end{aligned}$$
covariance of analysis

<u>interpretation:</u> $\mathbf{x}_b$ is Gaussian distributed with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$

$\mathbf{y} - \mathbf{H}\mathbf{x}_b$ is Gaussian distributed with a mean $\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b$ and covariance $\mathbf{R}$

$\mathbf{x}_a$ is Gaussian distributed with a mean $\bar{\mathbf{x}}_a$ and covariance $\mathbf{P}_a$

we define: $E[\varepsilon_b \varepsilon_b^t] = \mathbf{P}_b$ $\quad E[\varepsilon_o \varepsilon_o^t] = \mathbf{R}$

then we find: $E[\varepsilon_a \varepsilon_a^t] = (\mathbf{I} - \mathbf{KH})\mathbf{P}_b$

$= \mathbf{P}_a$ covariance of analysis

<u>interpretation:</u> $\mathbf{x}_b$ is Gaussian distributed with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$

$\mathbf{y} - \mathbf{H}\mathbf{x}_b$ is Gaussian distributed with a mean $\mathbf{y} - \bar{\mathbf{H}}\mathbf{x}_b$ and covariance $\mathbf{R}$

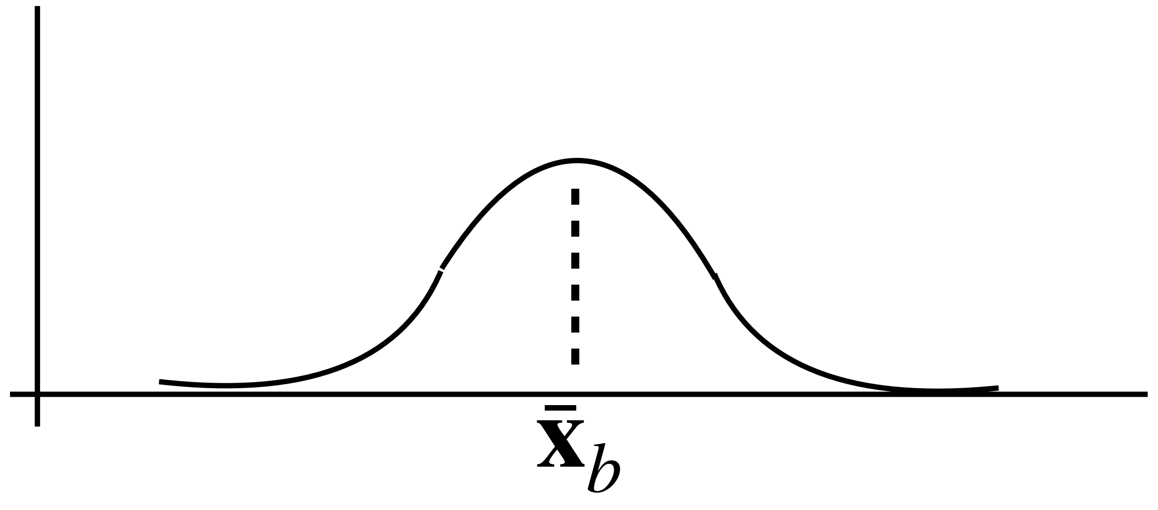
$\mathbf{x}_a$ is Gaussian distributed with a mean $\bar{\mathbf{x}}_a$ and covariance $\mathbf{P}_a$

$$\boxed{\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_\mathbf{b})}$$

$$\boxed{\mathbf{K} = \mathbf{P}_b\mathbf{H}^t(\mathbf{H}\mathbf{P}_b\mathbf{H}^t + \mathbf{R})^{-1}}$$

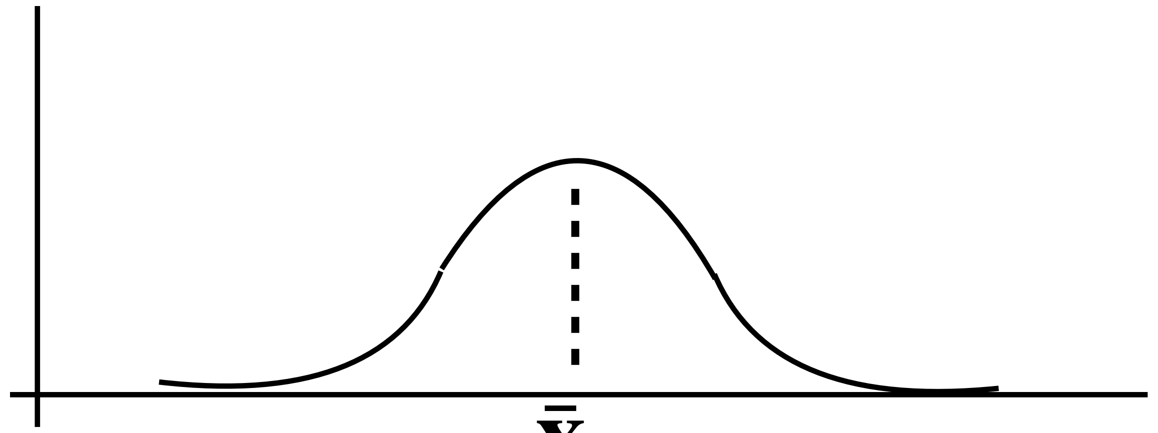
system state normal distribution

with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$



system state normal distribution

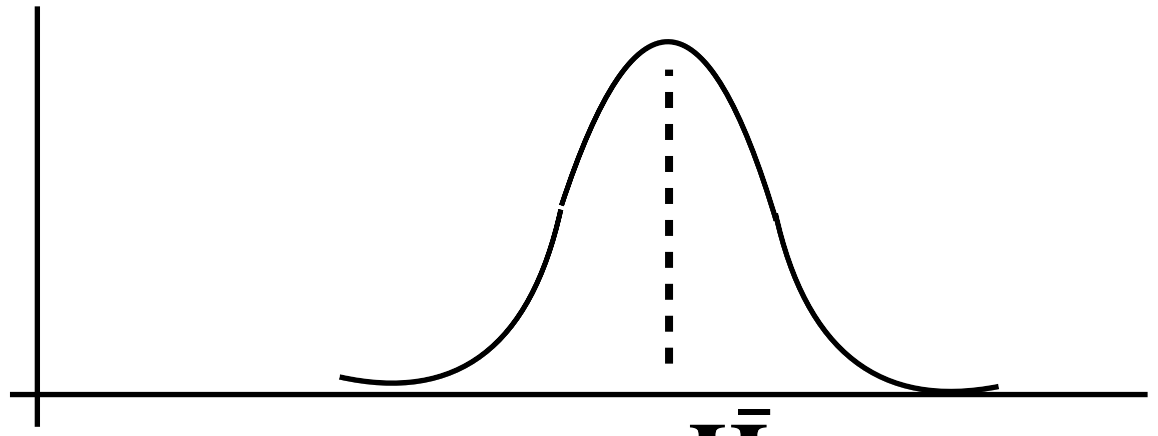
with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$



$\bar{\mathbf{x}}_b$

observation innovation normal distribution

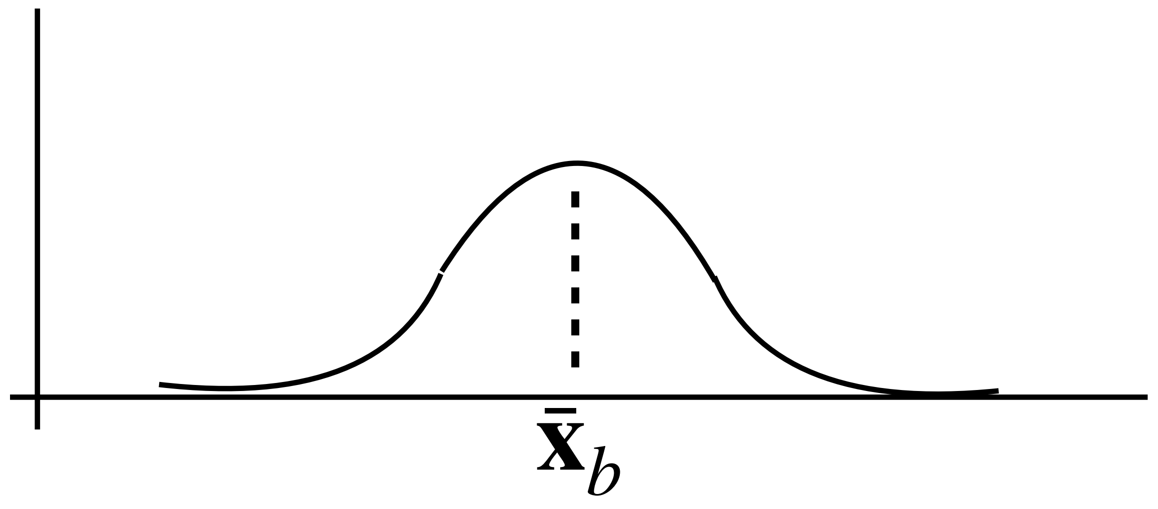
with a mean $\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b$ and covariance $\mathbf{R}$



$\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b$

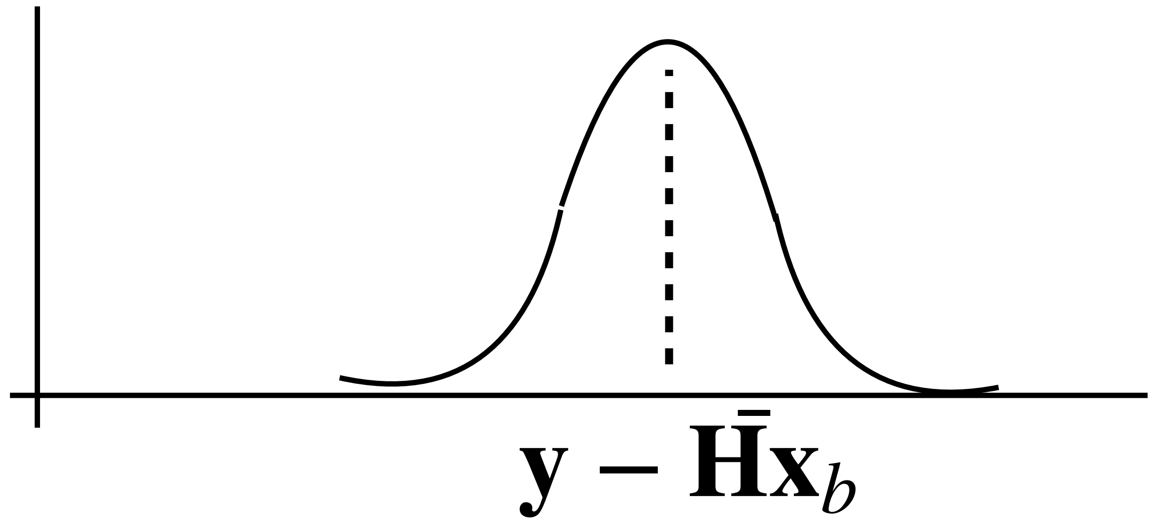
system state normal distribution

with a mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$



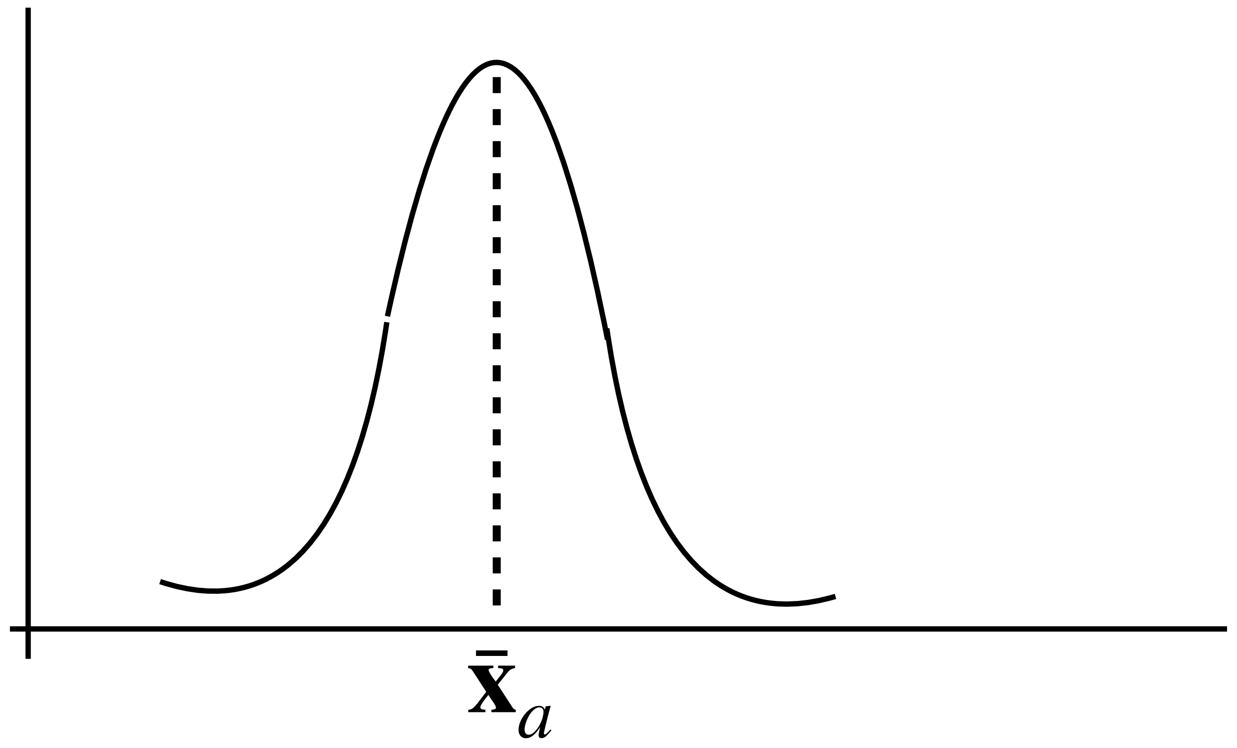
observation innovation normal distribution

with a mean $\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b$ and covariance $\mathbf{R}$



analysis system state normal distribution

with a mean $\bar{\mathbf{x}}_a$ and covariance $\mathbf{P}_a$



Data Assimilation

re-call:

system under study → observations

state evolution:

$$x_n = F(x_{n-1})$$

error

$$y_n = H(x^f_n) + \eta_n$$

observations observation operator

model evolution yields forecast:

$$x^f_n = M(x^a_{n-1}) + \varepsilon_n$$

model error

$$\boxed{x^a_n = x^f_n + \mathbf{K}\,(y_n - H(x^f_n))}$$

weight matrix / Kalman matrix

re-call:

Data Assimilation

system under study

observations

state evolution:

$x_n = F(x_{n-1})$

error

$y_n = H(x^f_n) + \eta_n$

observations   observation operator

model evolution yields forecast:

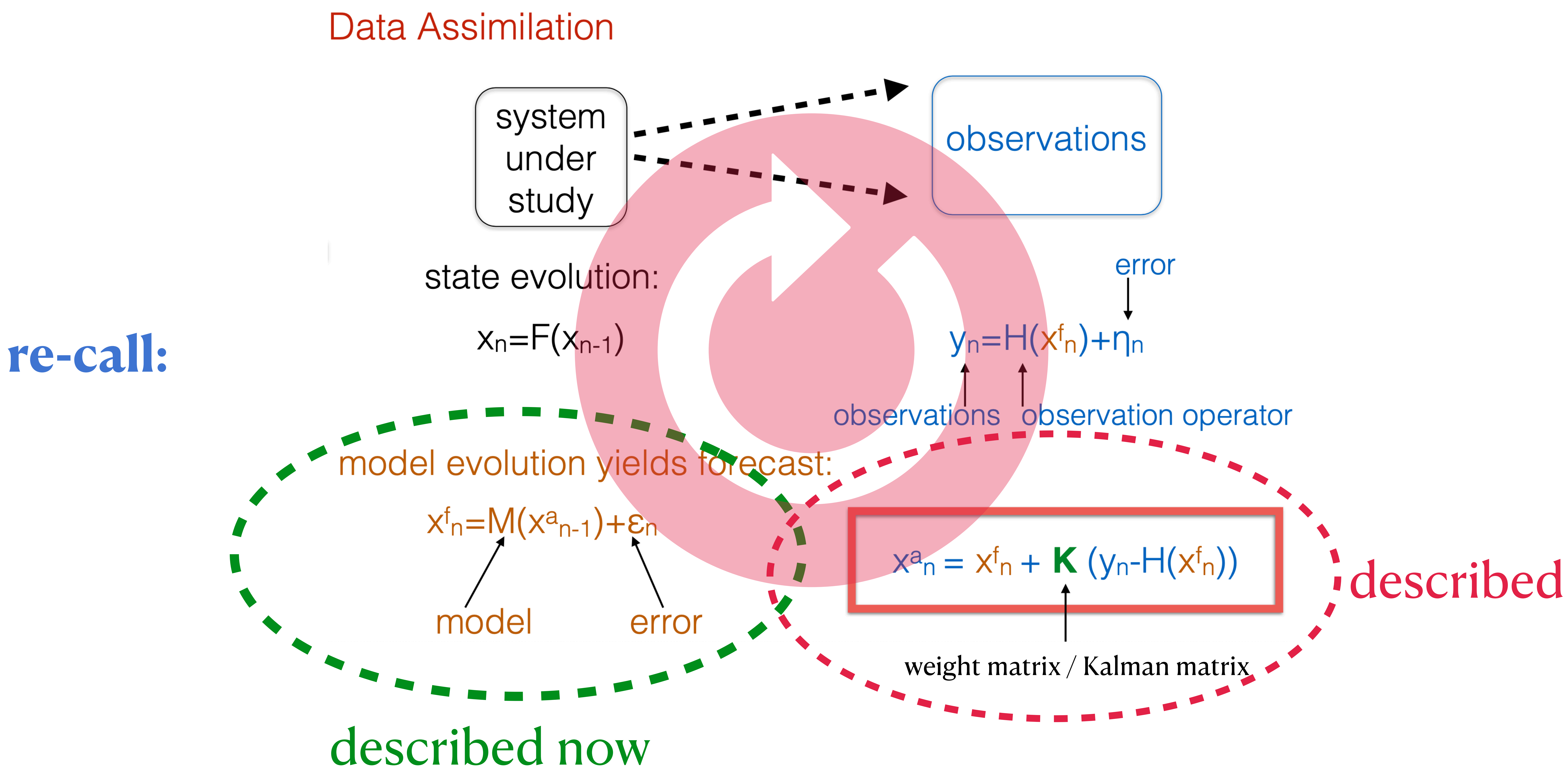
$x^f_n = M(x^a_{n-1}) + \varepsilon_n$

model   error

$$\boxed{x^a_n = x^f_n + \mathbf{K}\,(y_n - H(x^f_n))}$$

weight matrix / Kalman matrix

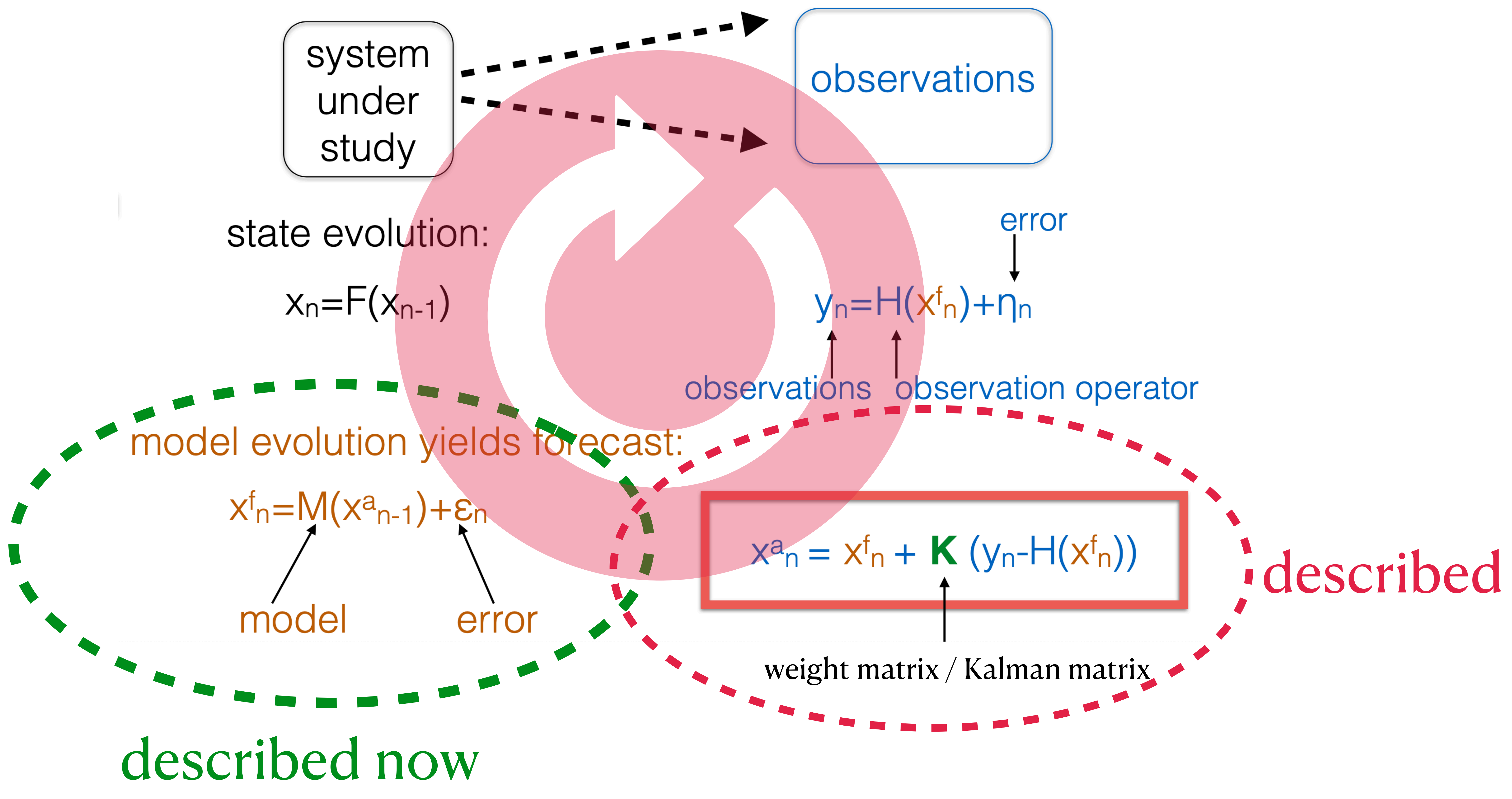
described



Data Assimilation
re-call:
system under study
observations
state evolution:
$x_n=F(x_{n-1})$
error
$y_n=H(x^f_n)+\eta_n$
observations
observation operator
model evolution yields forecast:
$x^f_n=M(x^a_{n-1})+\varepsilon_n$
model
error
$x^a_n = x^f_n + \mathbf{K}\,(y_n-H(x^f_n))$
weight matrix / Kalman matrix
described
described now

Data Assimilation

re-call:



for linear models: $\mathscr{M}\mathbf{x}_b = \mathbf{M}\mathbf{x}_b$

$$\boxed{\begin{aligned} \bar{\mathbf{x}}_b(t_n) &= \mathbf{M}\bar{\mathbf{x}}_a(t_{n-1}) \\ \mathbf{P}_b(t_n) &= \mathbf{M}\mathbf{P}_a(t_{n-1})\mathbf{M}^t + \mathbf{Q} \end{aligned}}$$

background covariance $\mathbf{P}_b$

is time-dependent now !!!

Kalman filter

all together:

Kalman filter

<u>all together:</u>

- system model and observation operator is linear

Kalman filter

<u>all together:</u>

- system model and observation operator is linear
- probability density of background and observation innovation is Gaussian

Kalman filter

<u>all together:</u>

- system model and observation operator is linear
- probability density of background and observation innovation is Gaussian

at time $t_{n-1} \rightarrow t_n$ prediction step

$$\boxed{\begin{aligned} \bar{\mathbf{x}}_b(t_n) &= \mathbf{M}\bar{\mathbf{x}}_a(t_{n-1}) \\ \mathbf{P}_b(t_n) &= \mathbf{M}\mathbf{P}_a(t_{n-1})\mathbf{M}^t + \mathbf{Q} \end{aligned}}$$

Kalman filter

all together:

- system model and observation operator is linear
- probability density of background and observation innovation is Gaussian

at time $t_{n-1} \rightarrow t_n$ prediction step

$$\boxed{\begin{aligned} \bar{\mathbf{x}}_b(t_n) &= \mathbf{M}\bar{\mathbf{x}}_a(t_{n-1}) \\ \mathbf{P}_b(t_n) &= \mathbf{M}\mathbf{P}_a(t_{n-1})\mathbf{M}^t + \mathbf{Q} \end{aligned}}$$

at time t_n analysis step

$$\boxed{\begin{aligned} \bar{\mathbf{x}}_a &= \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b) \\ \mathbf{P}_a &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_b \end{aligned}}$$

$$\mathbf{K} = \mathbf{P}_b\mathbf{H}^t(\mathbf{H}\mathbf{P}_b\mathbf{H}^t + \mathbf{R})^{-1}$$

Kalman filter

<u>all together:</u>

- system model and observation operator is linear
- probability density of background and observation innovation is Gaussian

at time $t_{n-1} \rightarrow t_n$ prediction step

$$\boxed{\begin{aligned} \bar{\mathbf{x}}_b(t_n) &= \mathbf{M}\bar{\mathbf{x}}_a(t_{n-1}) \\ \mathbf{P}_b(t_n) &= \mathbf{M}\mathbf{P}_a(t_{n-1})\mathbf{M}^t + \mathbf{Q} \end{aligned}}$$

cycle step

at time t_n analysis step

$$\boxed{\begin{aligned} \bar{\mathbf{x}}_a &= \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b) \\ \mathbf{P}_a &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_b \end{aligned}}$$

$$\mathbf{K} = \mathbf{P}_b\mathbf{H}^t(\mathbf{H}\mathbf{P}_b\mathbf{H}^t + \mathbf{R})^{-1}$$

linear Kalman Filter (KF)

<u>advantage:</u>

- simple to implement
- numerically fast since analysis is given analytically
- background covariance $\mathbf{P}_b$ is flow-dependent

linear Kalman Filter (KF)

advantage:

- simple to implement
- numerically fast since analysis is given analytically
- background covariance $\mathbf{P}_b$ is flow-dependent

disadvantage:

- linear model and observation operator
- assumes Gaussian distribution

motivation

basic methods

Kalman filter

linear **EKF** UKF ETKF LETKF

prediction and verification

Extended Kalman filter (EKF)

- if system model is non-linear:

prediction step

$$\boxed{\begin{aligned}\mathbf{x}_b(t_n) &= \mathcal{M}\mathbf{x}_b(t_{n-1}) \\ \mathbf{P}_b(t_n) &= \mathbf{M}\mathbf{P}_a(t_{n-1})\mathbf{M}^t + \mathbf{Q}\end{aligned}}$$

$$\boxed{\mathbf{M} = \frac{\partial \mathcal{M}\mathbf{x}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_b(t_{n-1})}}$$

model Jacobian

analysis step

$$\boxed{\begin{aligned}\bar{\mathbf{x}}_a &= \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}_b) \\ \mathbf{P}_a &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_b\end{aligned}}$$

$$\mathbf{K} = \mathbf{P}_b\mathbf{H}^t(\mathbf{H}\mathbf{P}_b\mathbf{H}^t + \mathbf{R})^{-1}$$

Extended Kalman Filter (EKF)

advantage:

- simple to implement
- numerically fast since analysis is given analytically
- background covariance $\mathbf{P}_b$ is flow-dependent
- **non-linear model**

disadvantage:

- linear observation operator
- **depends on model Jacobian**
- assumes Gaussian distribution

motivation

basic methods

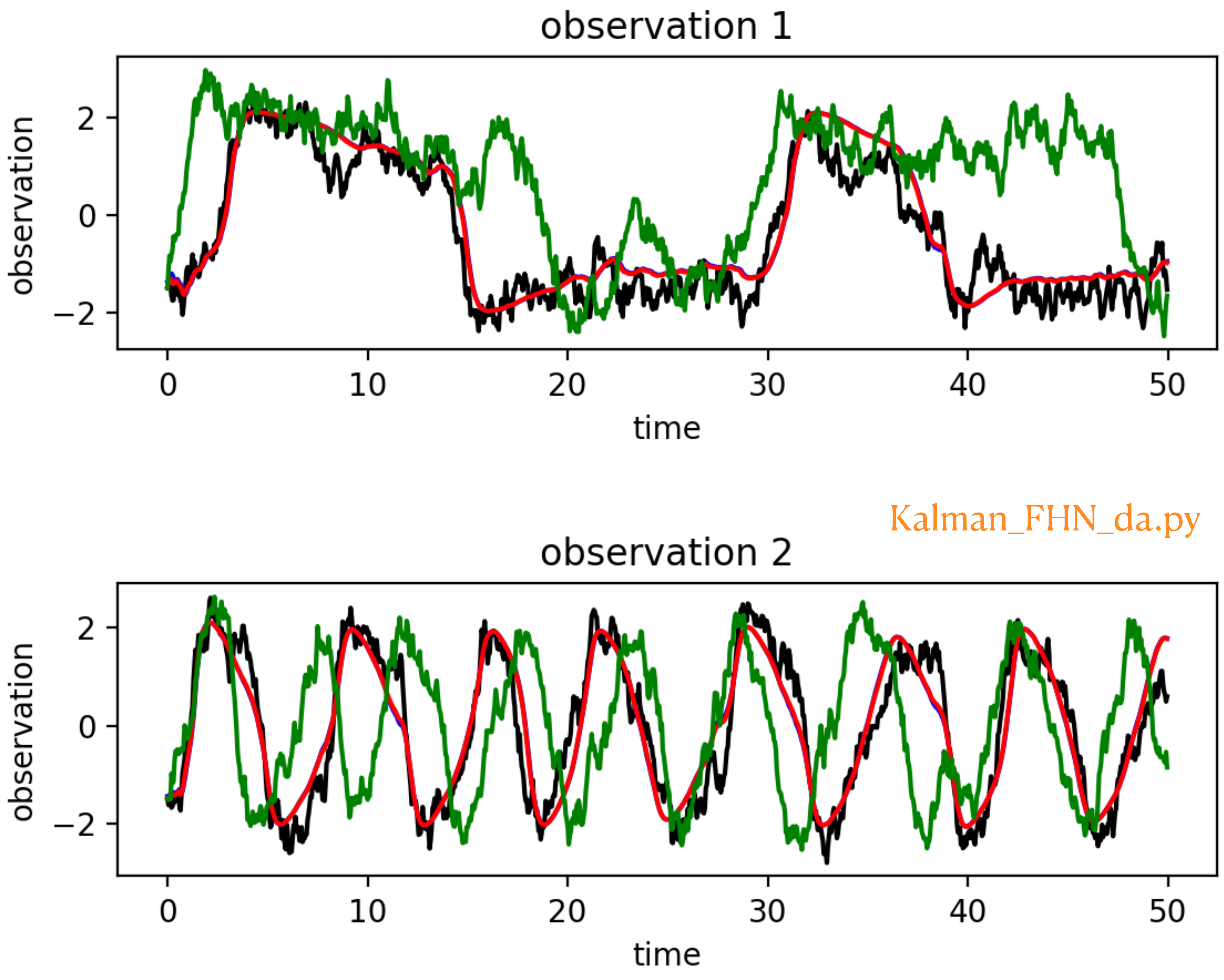
prediction and verification

Kalman filter

linear **EKF** UKF ETKF LETKF

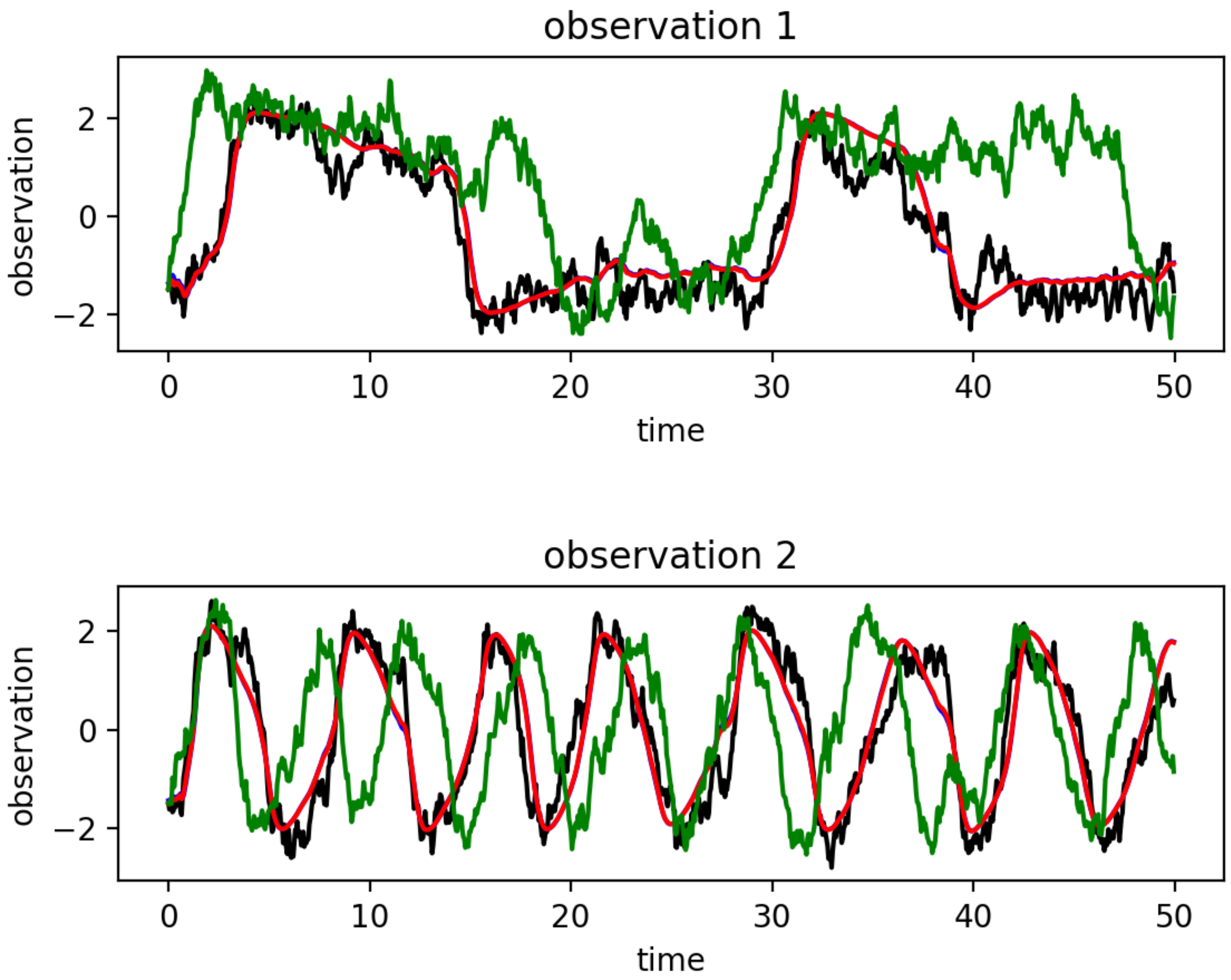
example

parameter estimation

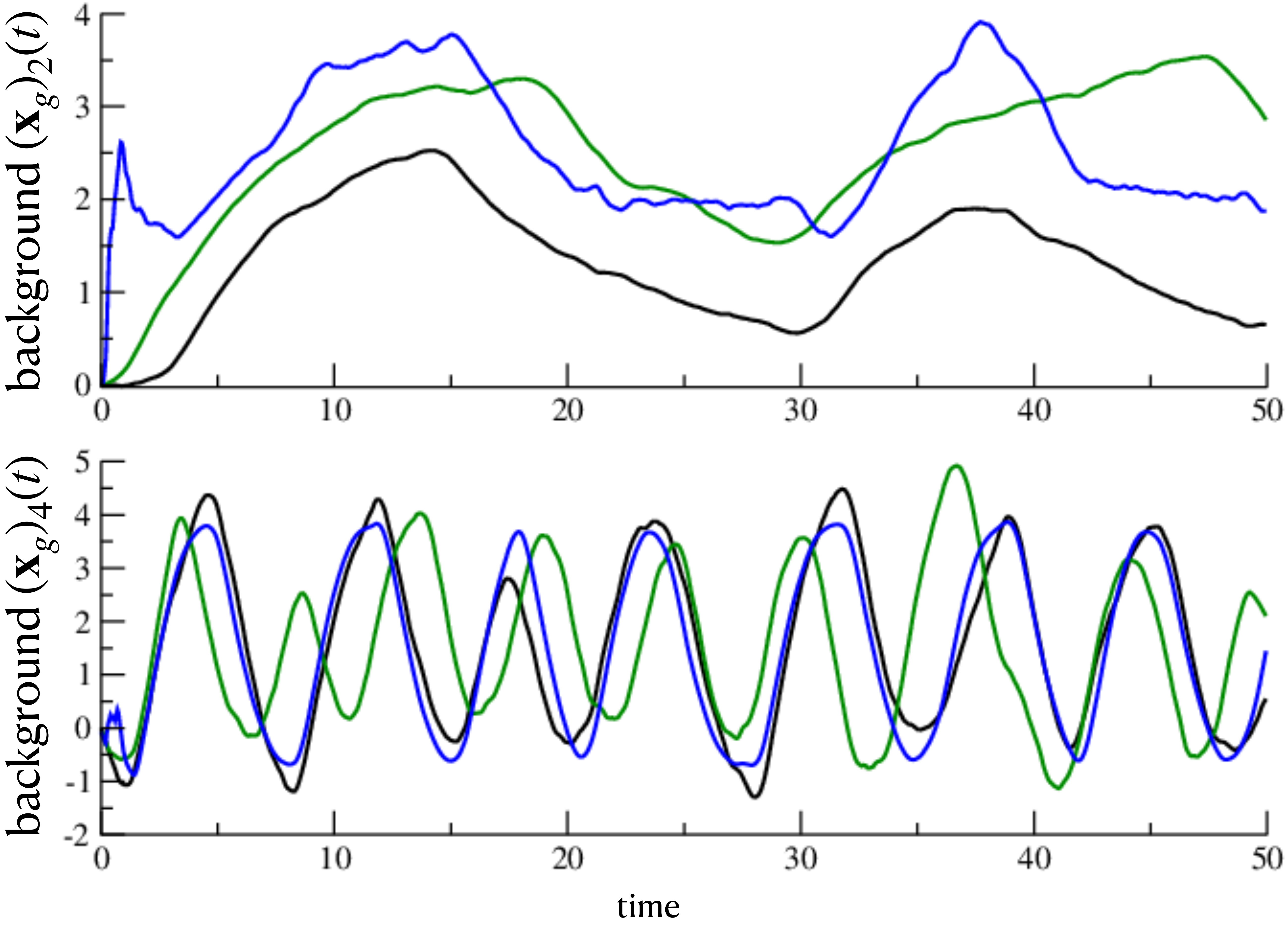


with $\mathbf{Q} = 0.0001\mathbf{I}$ and $\mathbf{R} = 0.1\mathbf{I}$

with $\mathbf{Q} = 0.0001\mathbf{I}$ and $\mathbf{R} = 0.1\mathbf{I}$



hidden variables



good reconstruction of hidden variables

motivation

basic methods

Kalman filter

linear **EKF** UKF ETKF LETKF

example

parameter estimation

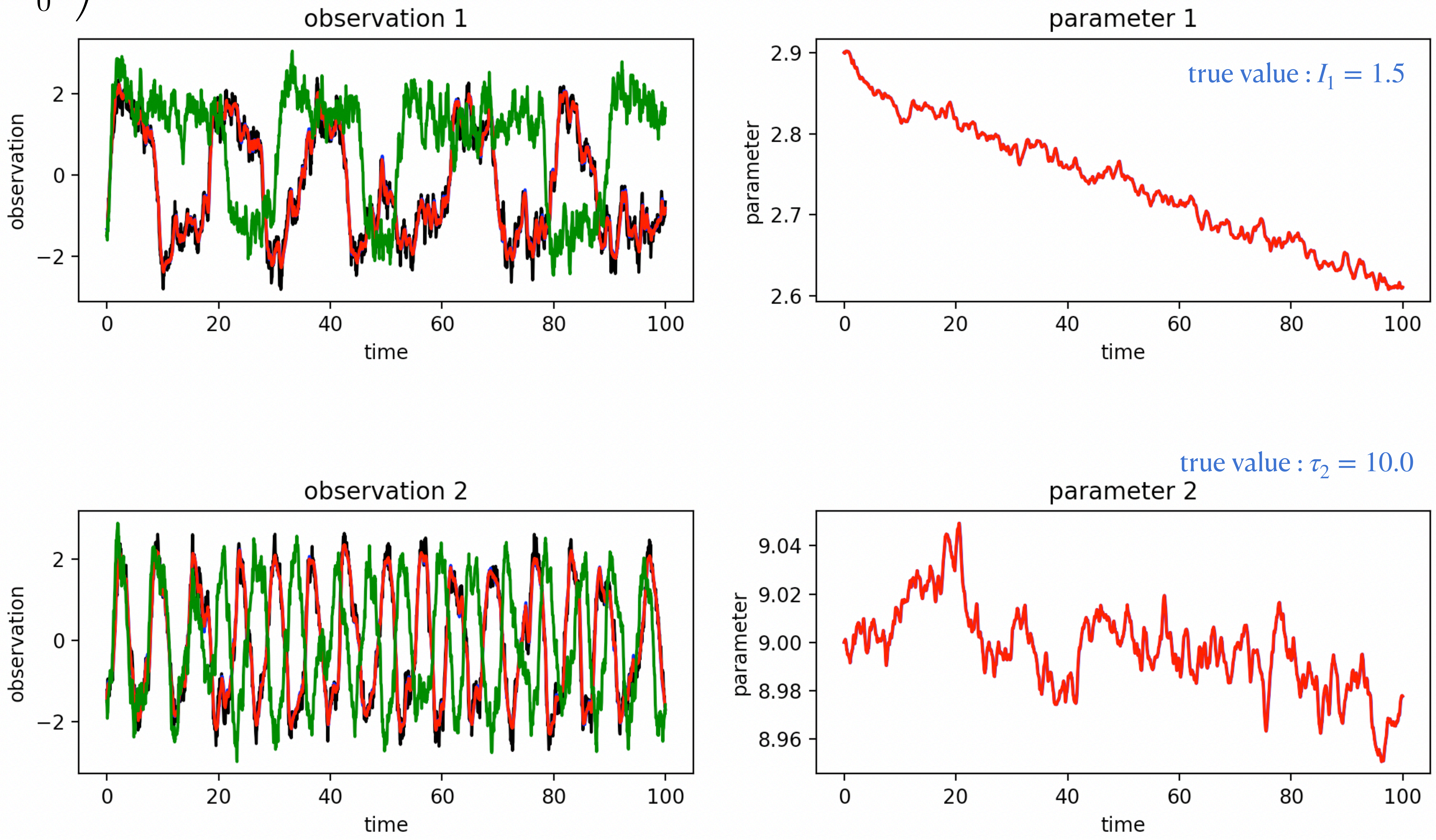
prediction and verification

parameter estimation

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 & 10^{-3} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 10^{-3} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 10^{-3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$$

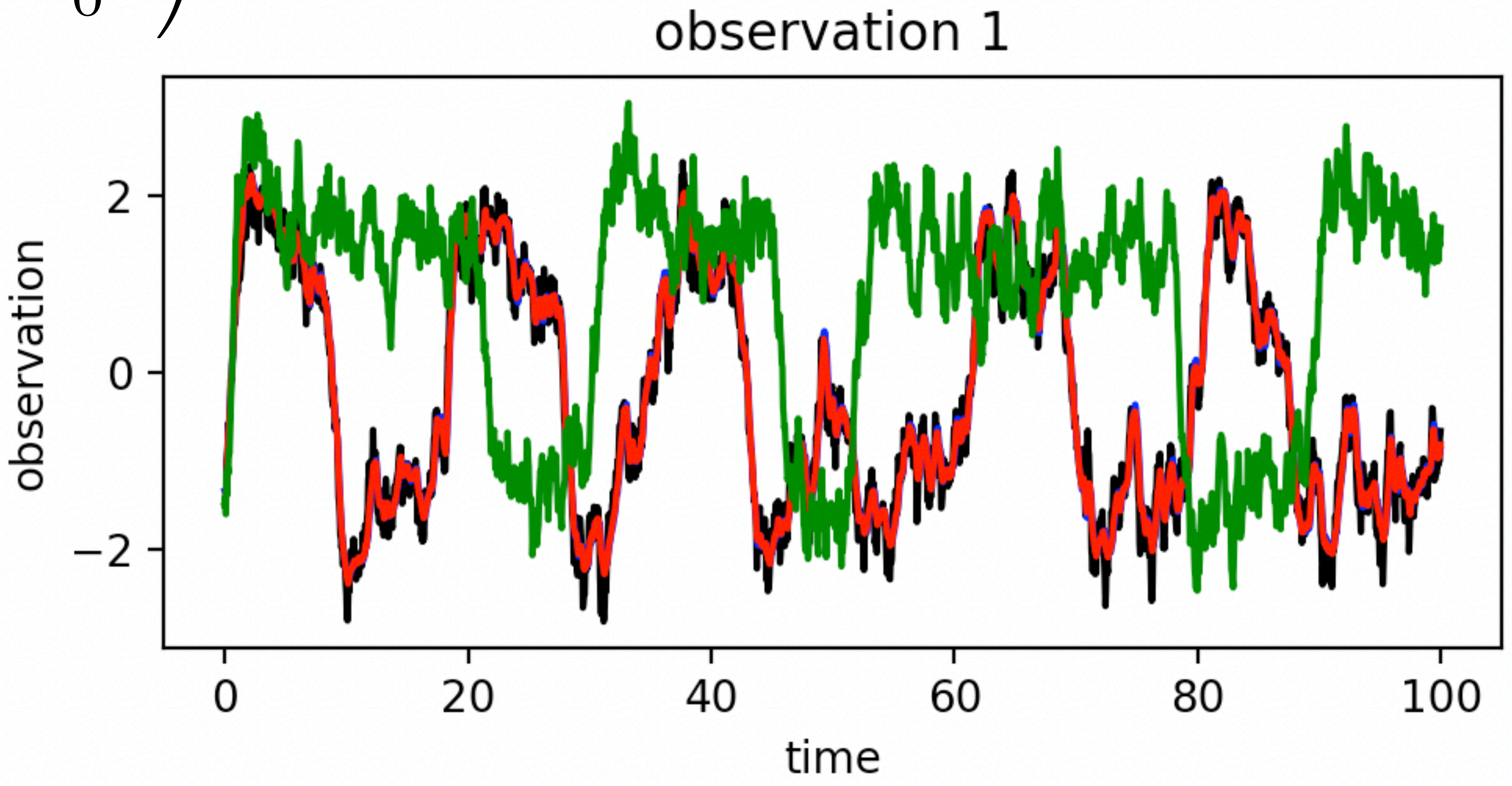
Kalman_FHN_pestim_da.py

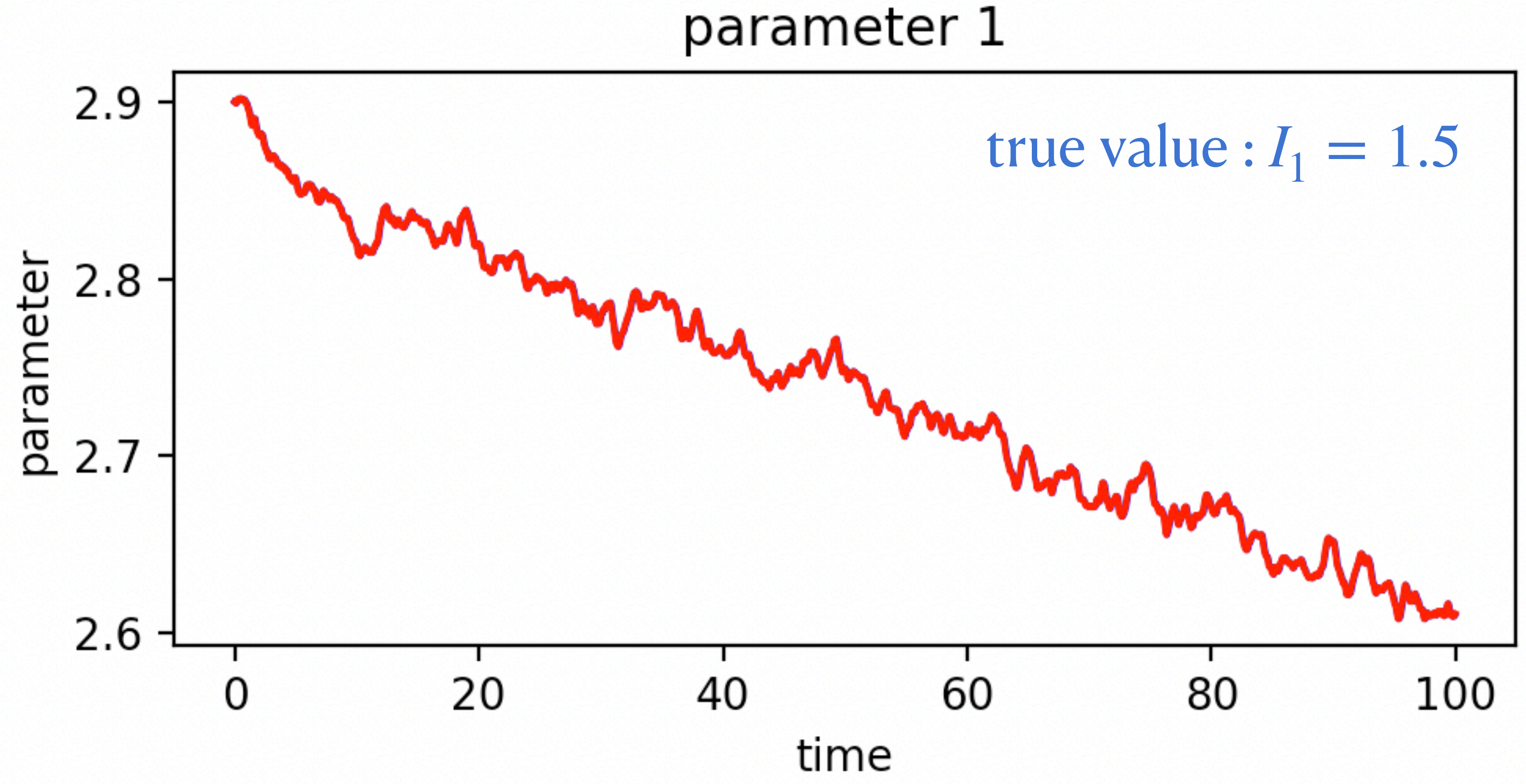


$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 & 10^{-3} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 10^{-3} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 10^{-3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 & 0 & 0 \end{pmatrix}$$

parameter estimation

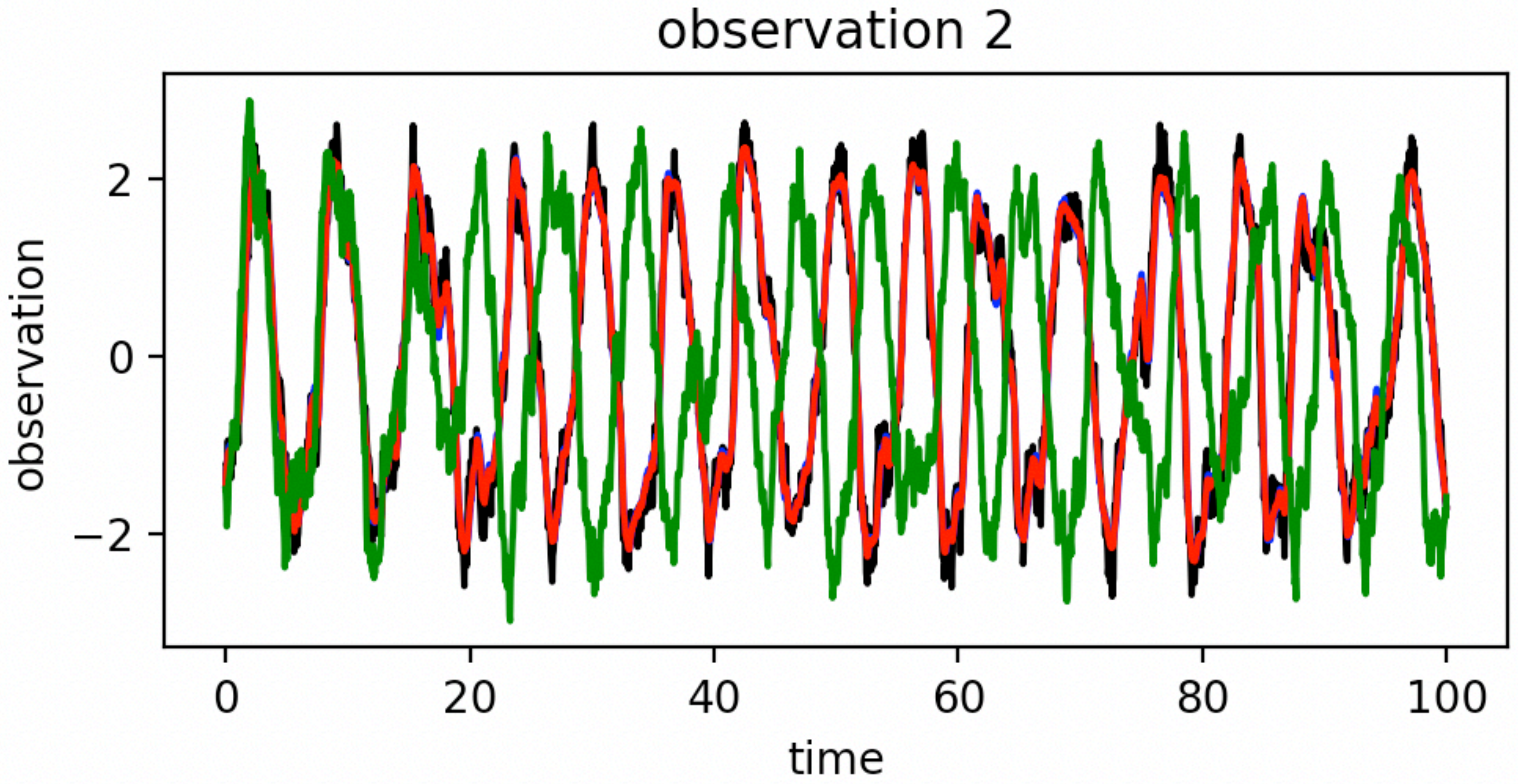
$$\mathbf{R} = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix}$$

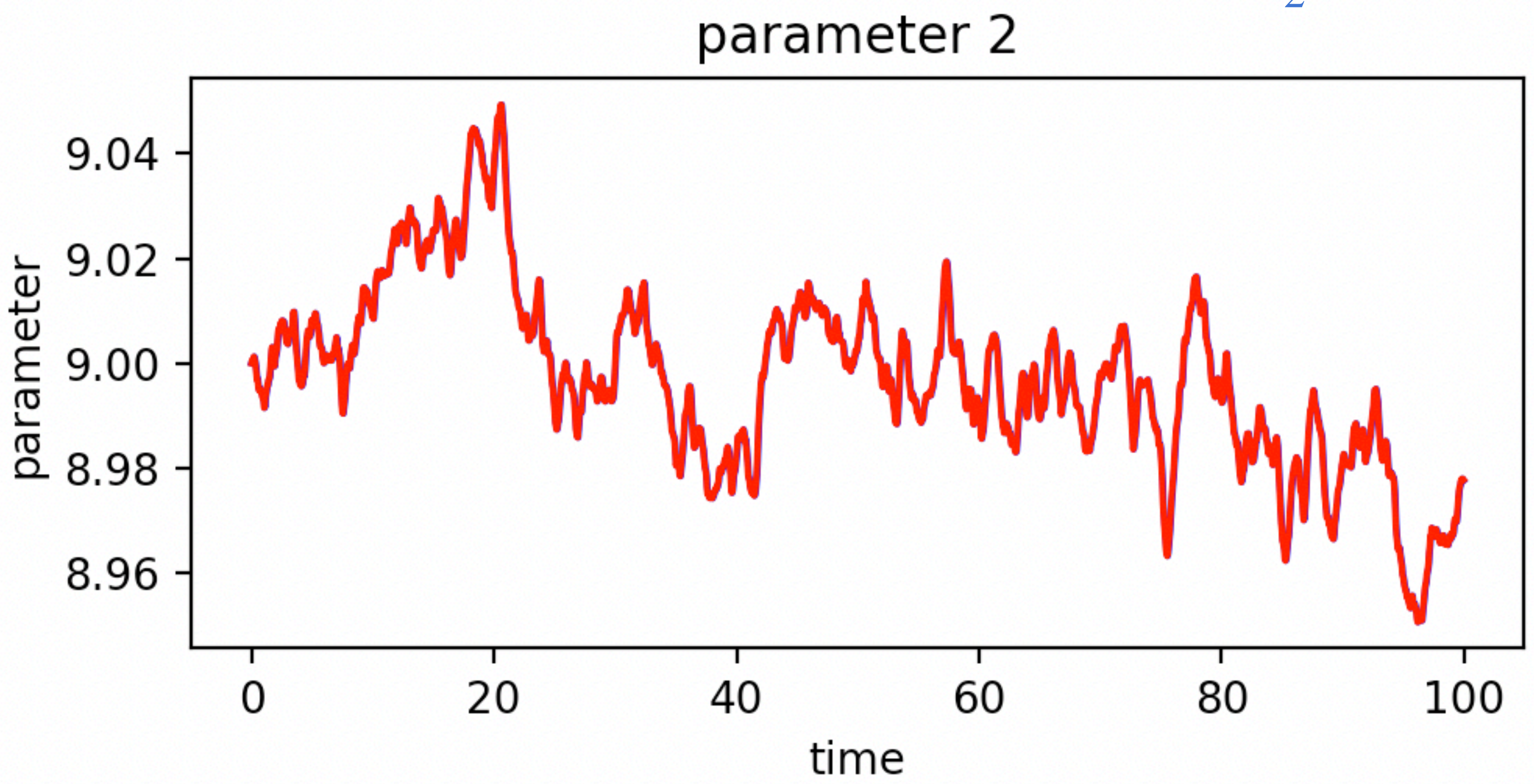




bad parameter estimation

true value : $\tau_2 = 10.0$





motivation

basic methods

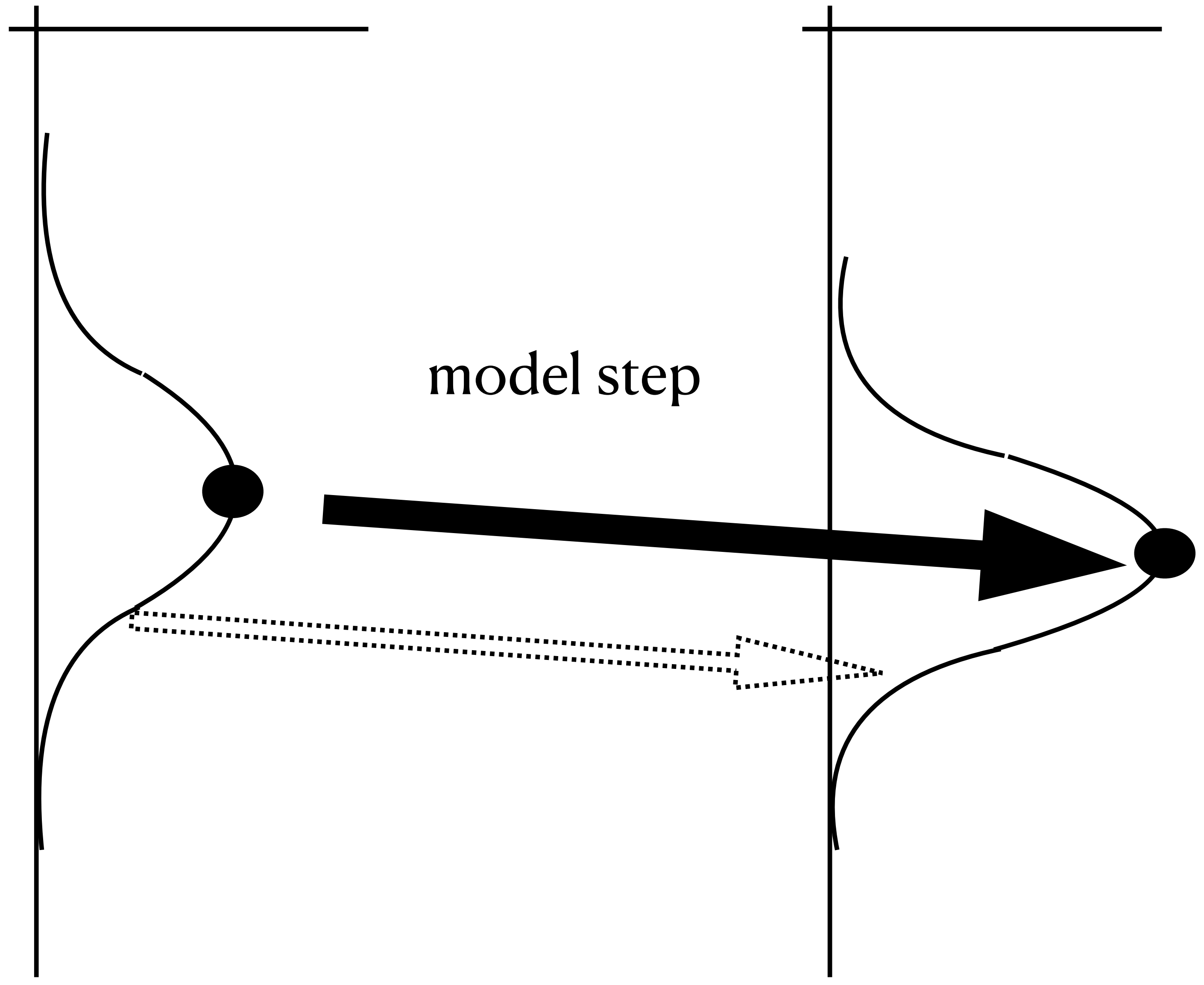
Kalman filter

linear EKF **UKF** ETKF LETKF

prediction and verification

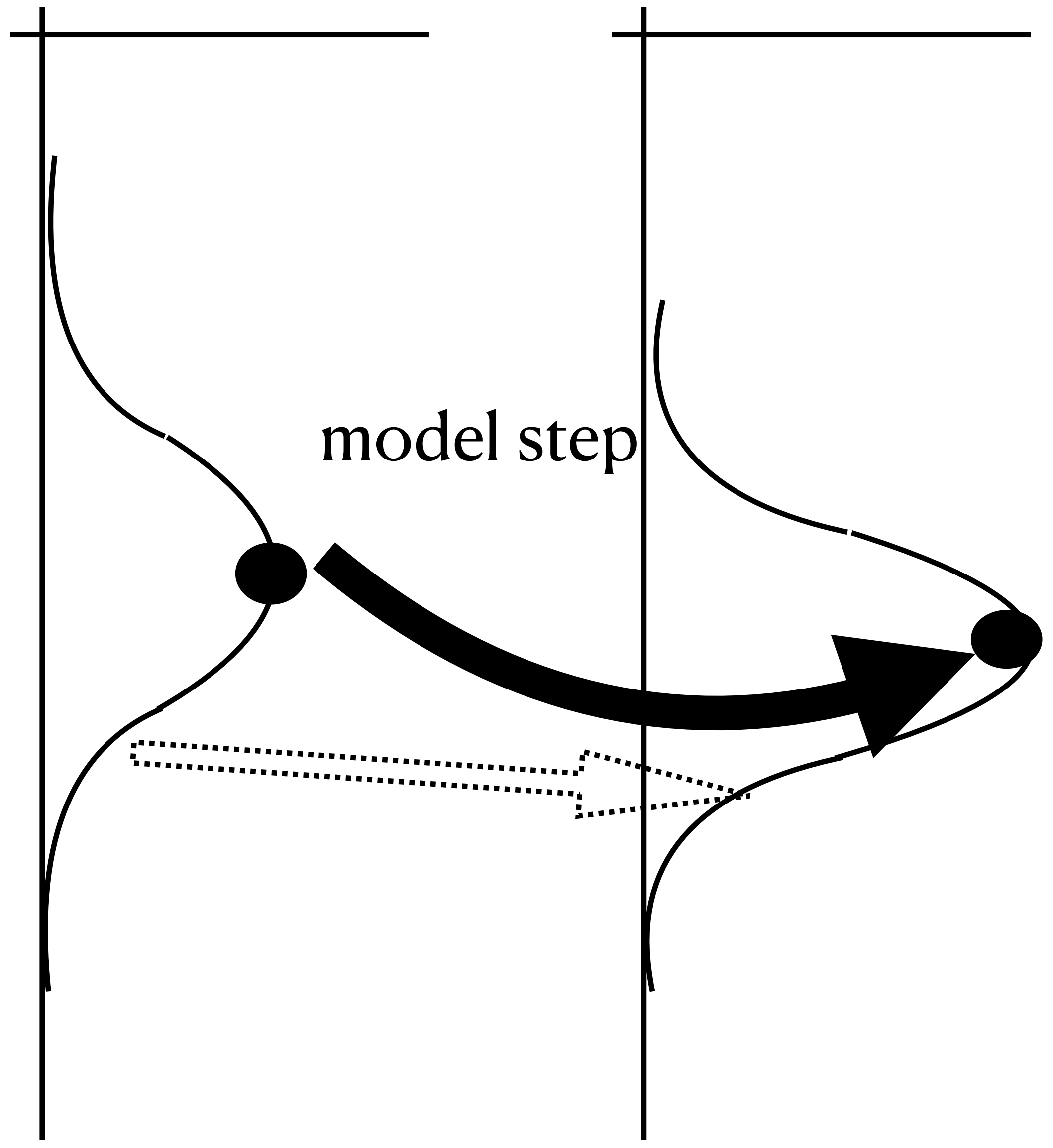
Unscented Kalman filter (UKF)

Kalman filter



linear model

Extended Kalman filter



nonlinear model

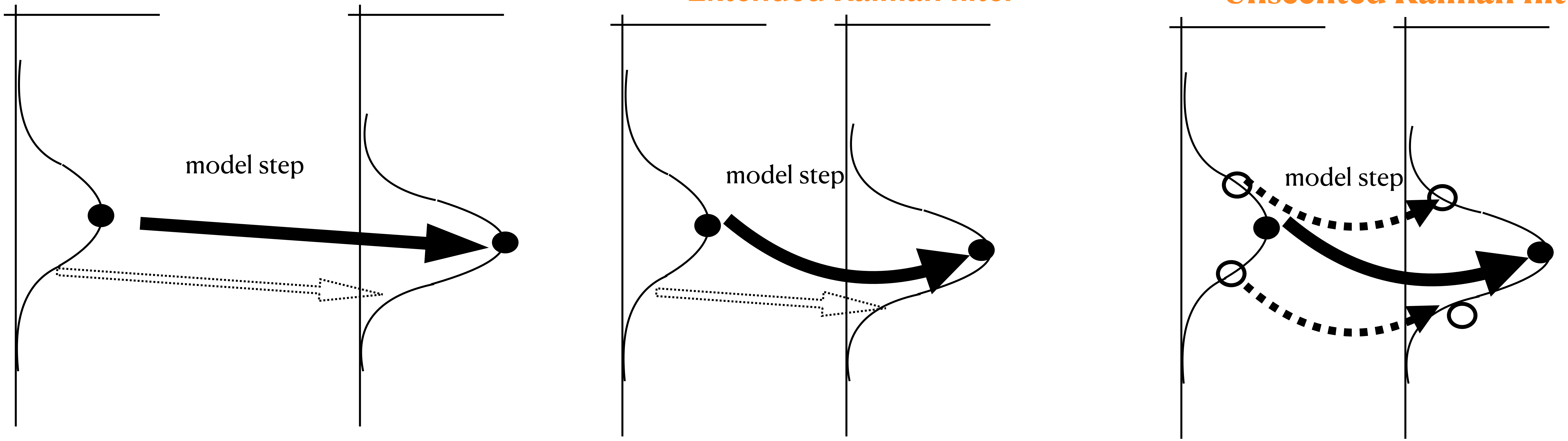
linear map of distribution

distribution is represented by **mean** and **covariance**

mean and variance are transformed

Unscented Kalman filter (UKF)

Kalman filter | Extended Kalman filter | **Unscented Kalman filter**



linear model

nonlinear model
linearised map of distribution

nonlinear model
nonlinear map of distribution

○ sigma points

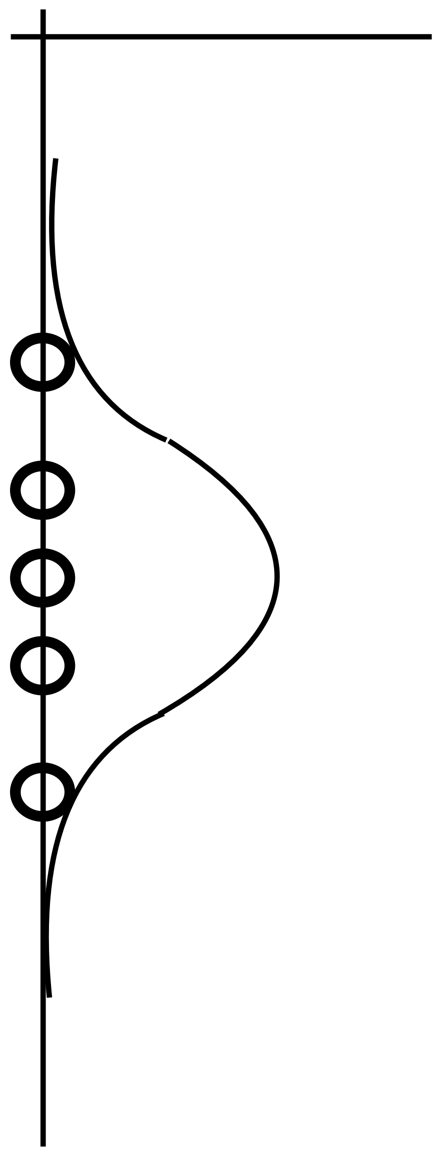
distribution is represented by **mean** and **covariance**

mean and variance are transformed

distribution is represented by **sigma points**

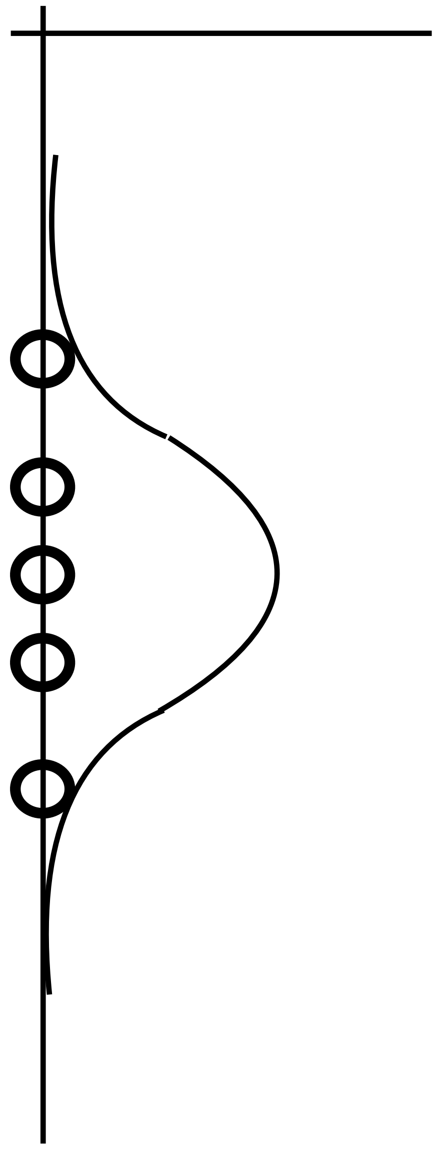
sigma points are transformed

Unscented Transform



- given: mean $\bar{\mathbf{x}}_a \in \mathbb{R}^L$ and $\mathbf{P}_a \in \mathbb{R}^{L \times L}$ at time t_{n-1}

Unscented Transform



- given: mean $\bar{\mathbf{x}}_a \in \mathbb{R}^L$ and $\mathbf{P}_a \in \mathbb{R}^{L\times L}$ at time t_{n-1}
- represent distribution by L sampled points, so-called sigma points $\mathbf{s}_j$

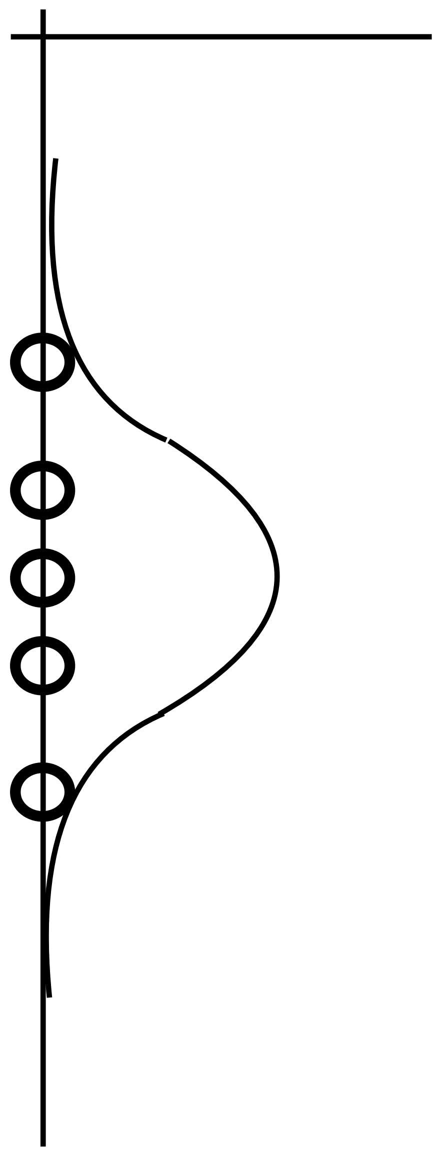
$$\mathbf{s}_0(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1})$$

$$\mathbf{s}_j(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) + \sqrt{\frac{\lambda}{\lambda + L}}\mathbf{A}_j \qquad j = 1, \ldots, L$$

$$\mathbf{s}_{L+j}(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) - \sqrt{\frac{\lambda}{\lambda + L}}\mathbf{A}_j \qquad j = 1, \ldots, L$$

$\mathbf{P}_a = \mathbf{A}\mathbf{A}^t$ $\qquad$ $\mathbf{A}_j$: j-th column vector of $\mathbf{A}$

Unscented Transform



- given: mean $\bar{\mathbf{x}}_a \in \mathbb{R}^L$ and $\mathbf{P}_a \in \mathbb{R}^{L\times L}$ at time t_{n-1}
- represent distribution by L sampled points, so-called sigma points $\mathbf{s}_j$

$$\mathbf{s}_0(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1})$$

$$\mathbf{s}_j(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) + \sqrt{\frac{\lambda}{\lambda+L}}\mathbf{A}_j \qquad j = 1,\ldots,L$$

$$\mathbf{s}_{L+j}(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) - \sqrt{\frac{\lambda}{\lambda+L}}\mathbf{A}_j \qquad j = 1,\ldots,L$$

$\mathbf{P}_a = \mathbf{A}\mathbf{A}^t$ $\qquad$ $\mathbf{A}_j$: j-th column vector of $\mathbf{A}$

weight $\lambda = 3 - L$

<u>model prediction</u> of sigma points: $\mathbf{x}_b^{(j)}(t_n) = \mathcal{M}\mathbf{s}_j(t_{n-1})$

description of
background distribution

$$\bar{\mathbf{x}}_b(t_n) = \sum_{j=0}^{2L} W_j^{(m)} \mathbf{x}^{(j)}(t_n)$$

$$\mathbf{P}_b(t_n) = \sum_{j=0}^{2L} W_j^{(c)} \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right) \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)^t + \mathbf{Q}$$

weights

$$W_0^{(m)} = \frac{\lambda}{L+\lambda}$$

$$W_j^{(m)} = \frac{1}{2(L+\lambda)} \quad , \quad j = 1, \ldots, 2L$$

$$W_0^{(c)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta$$

$$W_j^{(c)} = \frac{1}{2(L+\lambda)} \quad , \quad j = 1, \ldots, 2L$$

description of
background distribution

$$\bar{\mathbf{x}}_b(t_n) = \sum_{j=0}^{2L} W_j^{(m)} \mathbf{x}^{(j)}(t_n)$$

$$\mathbf{P}_b(t_n) = \sum_{j=0}^{2L} W_j^{(c)} \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right) \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)^t + \mathbf{Q}$$

weights

$$W_0^{(m)} = \frac{\lambda}{L+\lambda}$$

$$W_j^{(m)} = \frac{1}{2(L+\lambda)} \quad , \quad j = 1, \ldots, 2L$$

$$W_0^{(c)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta$$

$$W_j^{(c)} = \frac{1}{2(L+\lambda)} \quad , \quad j = 1, \ldots, 2L$$

<u>update by observations:</u>

- generate sigma points
  from background distribution $\mathbf{u}_j(t_n)$

description of
background distribution

$$\bar{\mathbf{x}}_b(t_n) = \sum_{j=0}^{2L} W_j^{(m)} \mathbf{x}^{(j)}(t_n)$$

$$\mathbf{P}_b(t_n) = \sum_{j=0}^{2L} W_j^{(c)} \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right) \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)^t + \mathbf{Q}$$

weights

$$W_0^{(m)} = \frac{\lambda}{L+\lambda}$$

$$W_j^{(m)} = \frac{1}{2(L+\lambda)} \quad , \quad j = 1, \ldots, 2L$$

$$W_0^{(c)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta$$

$$W_j^{(c)} = \frac{1}{2(L+\lambda)} \quad , \quad j = 1, \ldots, 2L$$

<u>update by observations:</u>

- generate sigma points
  from background distribution $\mathbf{u}_j(t_n)$
- unscented transform
  to gain background in observation space

$$\mathbf{y}^{(j)}(t_n) = H(\mathbf{u}_j(t_n))$$

$$\boxed{\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)}$$

analysis mean at time t_n

$$\boxed{\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)}$$

analysis mean at time t_n

$$\bar{\mathbf{y}}_b(t_n) = \sum_{j=0}^{2L} W_j^{(m)} \mathbf{y}^{(j)}(t_n)$$

$$\mathbf{K} = \mathbf{C}_{uy}\mathbf{C}_{yy}^{-1}$$

$HB^t(HBH^t+R)^{-1}$

$$\mathbf{C}_{yy}(t_n) = \sum_{j=0}^{2L} W_j \left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)\left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)^t + \mathbf{R}$$

$$\mathbf{C}_{uy}(t_n) = \sum_{j=0}^{2L} W_j^{(c)} \left(\mathbf{u}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)\left(\mathbf{u}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)^t$$

$$\boxed{\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)}$$

analysis mean at time t_n

$$\bar{\mathbf{y}}_b(t_n) = \sum_{j=0}^{2L} W_j^{(m)} \mathbf{y}^{(j)}(t_n)$$

$$\mathbf{K} = \mathbf{C}_{uy}\mathbf{C}_{yy}^{-1}$$

$$\mathbf{C}_{yy}(t_n) = \sum_{j=0}^{2L} W_j \left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)\left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)^t + \mathbf{R}$$

$$\mathbf{C}_{uy}(t_n) = \sum_{j=0}^{2L} W_j^{(c)} \left(\mathbf{u}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)\left(\mathbf{u}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)^t$$

$$\boxed{\mathbf{P}_a = \mathbf{P}_b - \mathbf{K}\mathbf{C}_{yy}\mathbf{K}^t}$$

analysis covariance at time t_n

Unscented Kalman Filter (UKF)

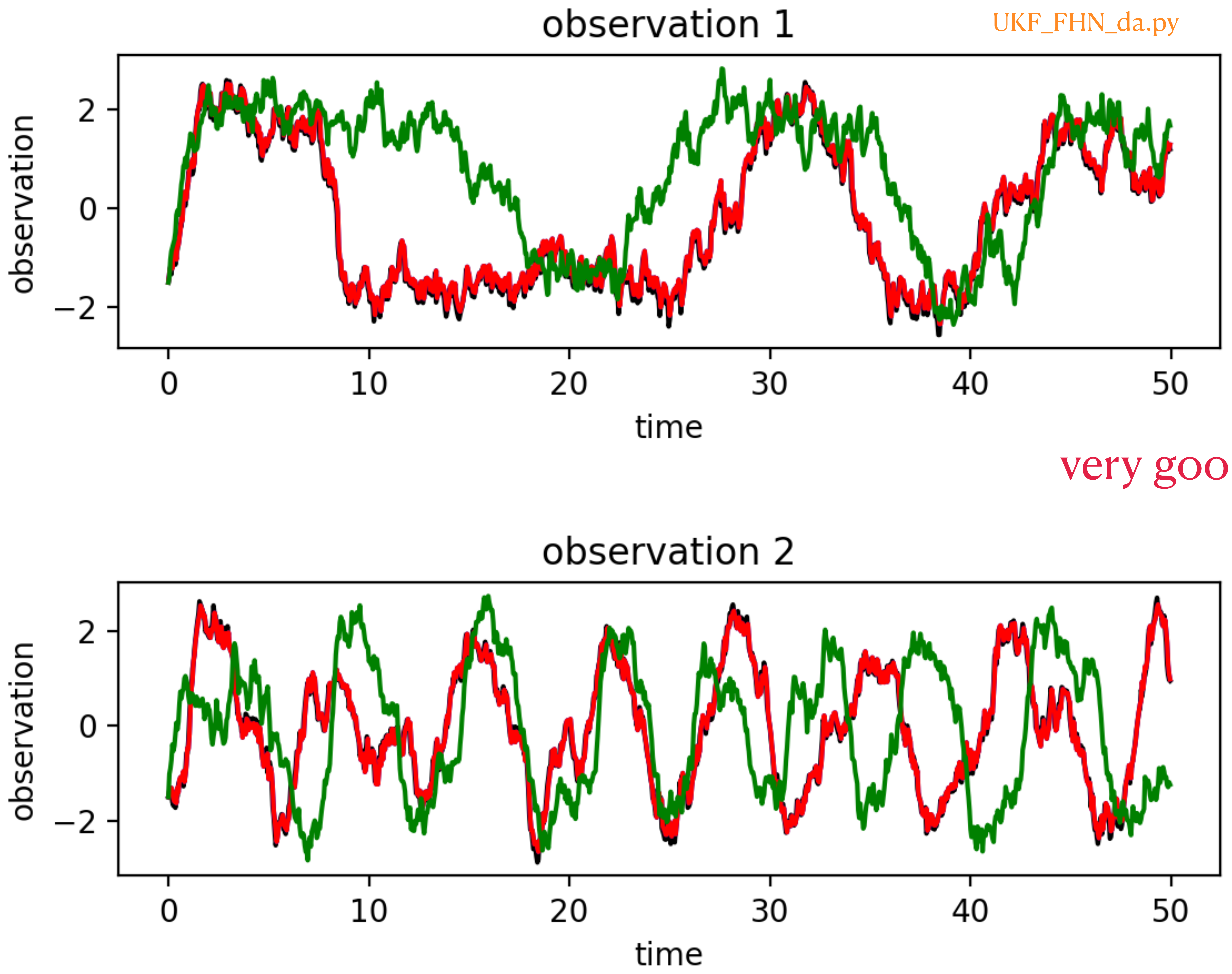
advantage:

- non-linear model and observation operator
- **does not depend on model Jacobian anymore**

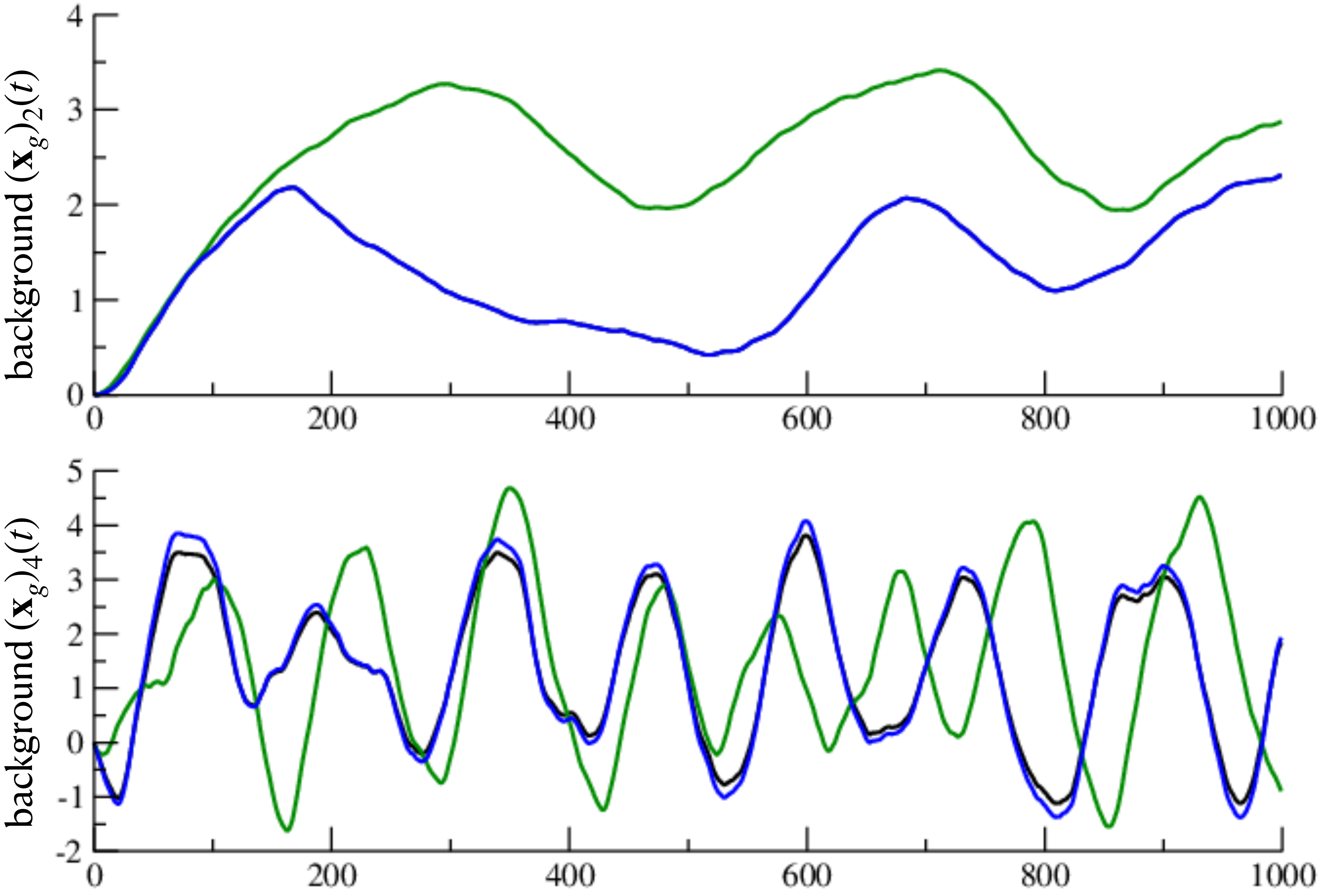
disadvantage:

- more complex to implement numerically
- may be numerically unstable in large dimensional model, see $\mathbf{A}_i = \sqrt{\mathbf{P}_a}_i$

UKF_FHN_da.py



very good tracking of observations



very good reconstruction

of hidden variables

motivation

basic methods

Kalman filter

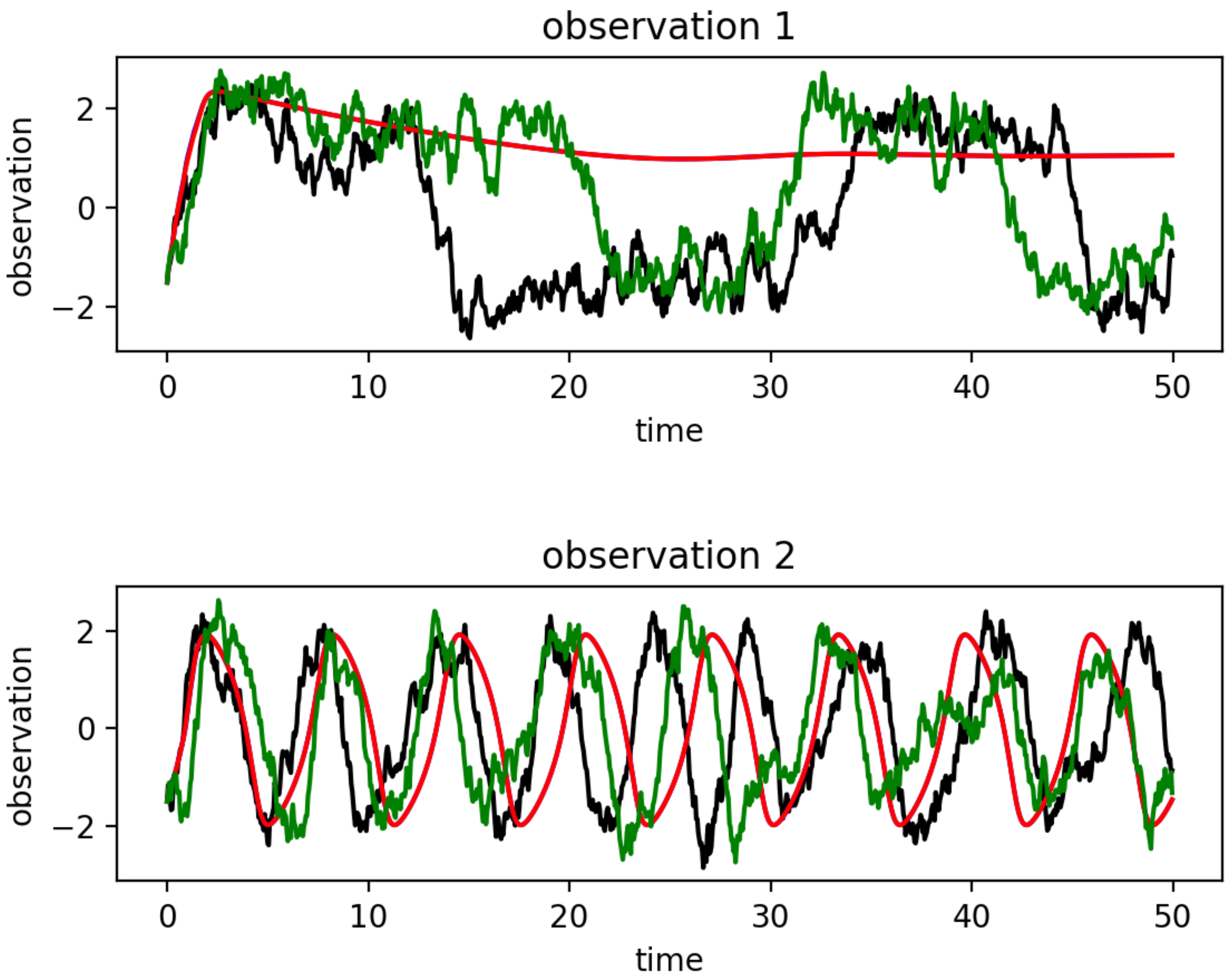
linear EKF **UKF** ETKF LETKF

impact of R and Q

parameter estimation

prediction and verification

impact of observation error $\mathbf{R}$ and model error $\mathbf{Q}$

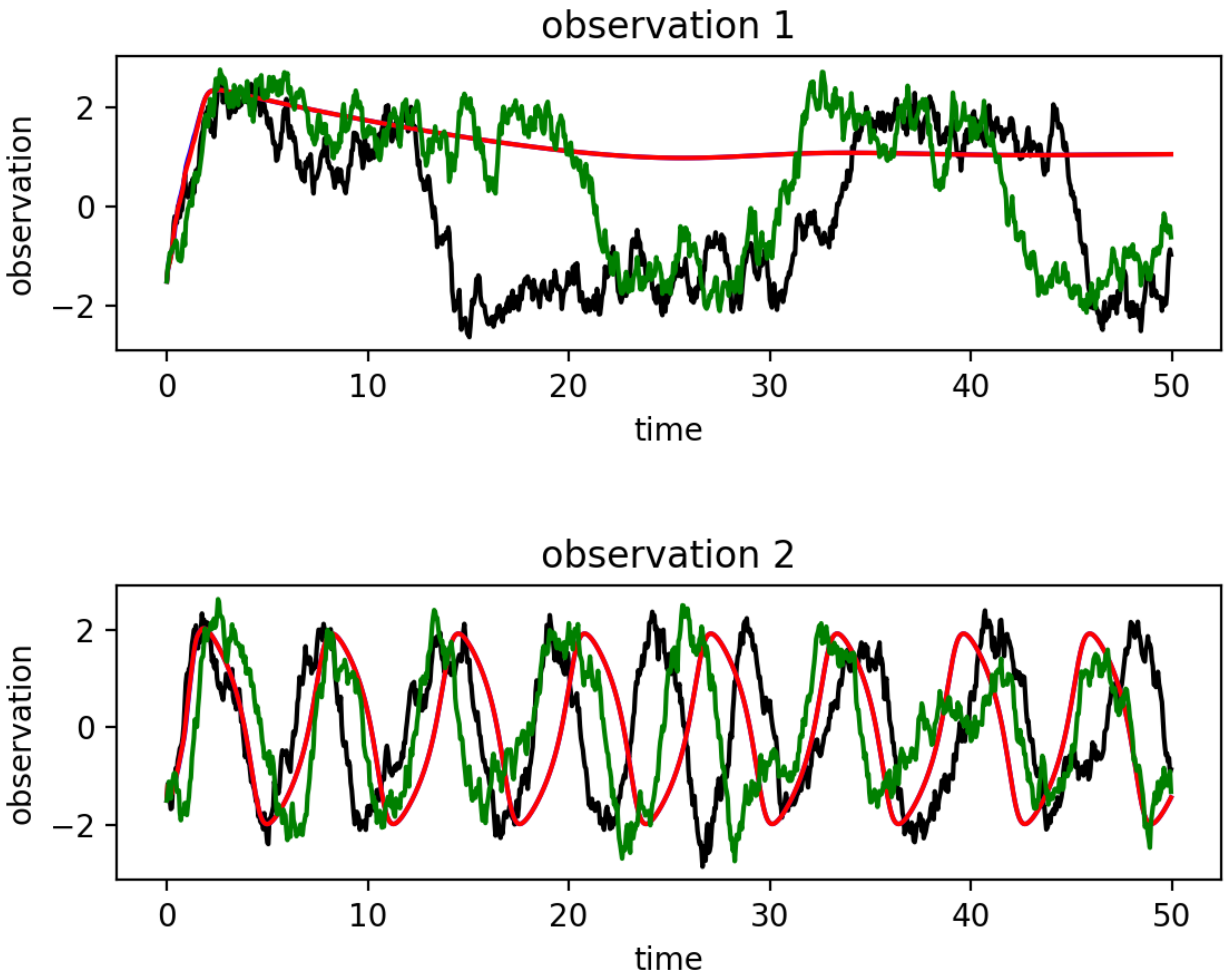


$$\mathbf{Q} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 50 & 0 \\ 0 & 50 \end{pmatrix}$$

background observations

do not fit observations

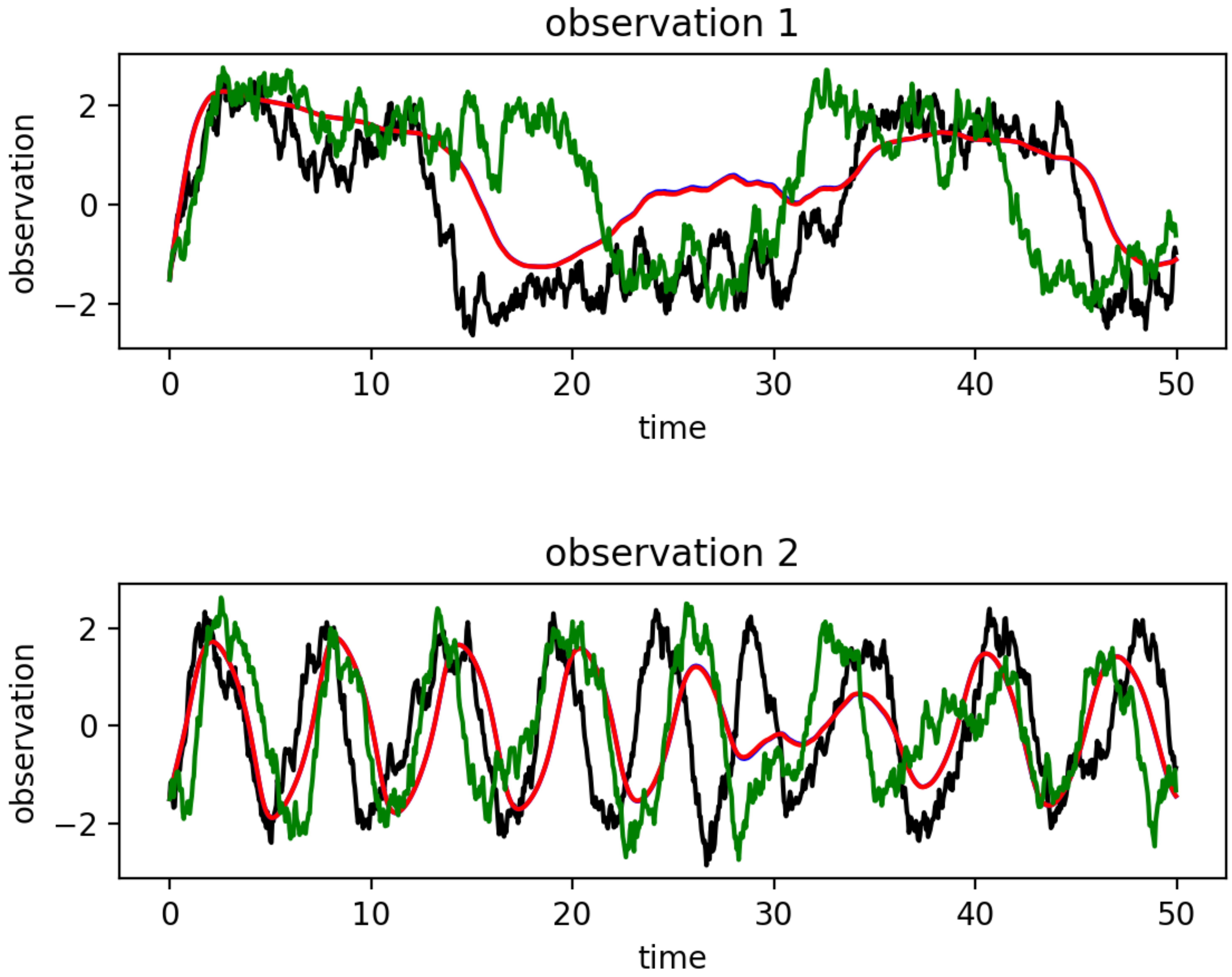


$$\mathbf{Q} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$$

background observations

do still not fit observations

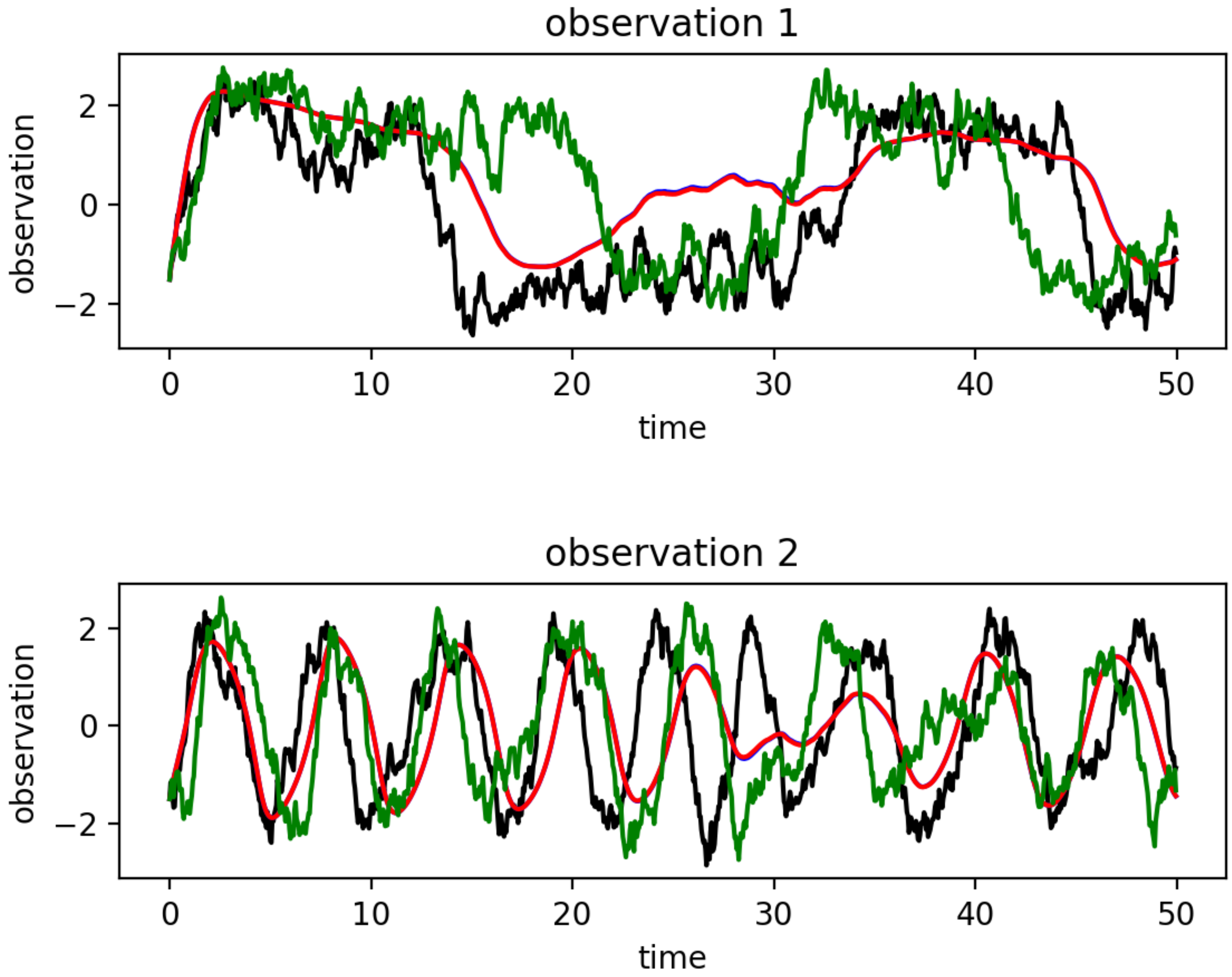


$$\mathbf{Q} = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 50 & 0 \\ 0 & 50 \end{pmatrix}$$

background observations

do fit better observations

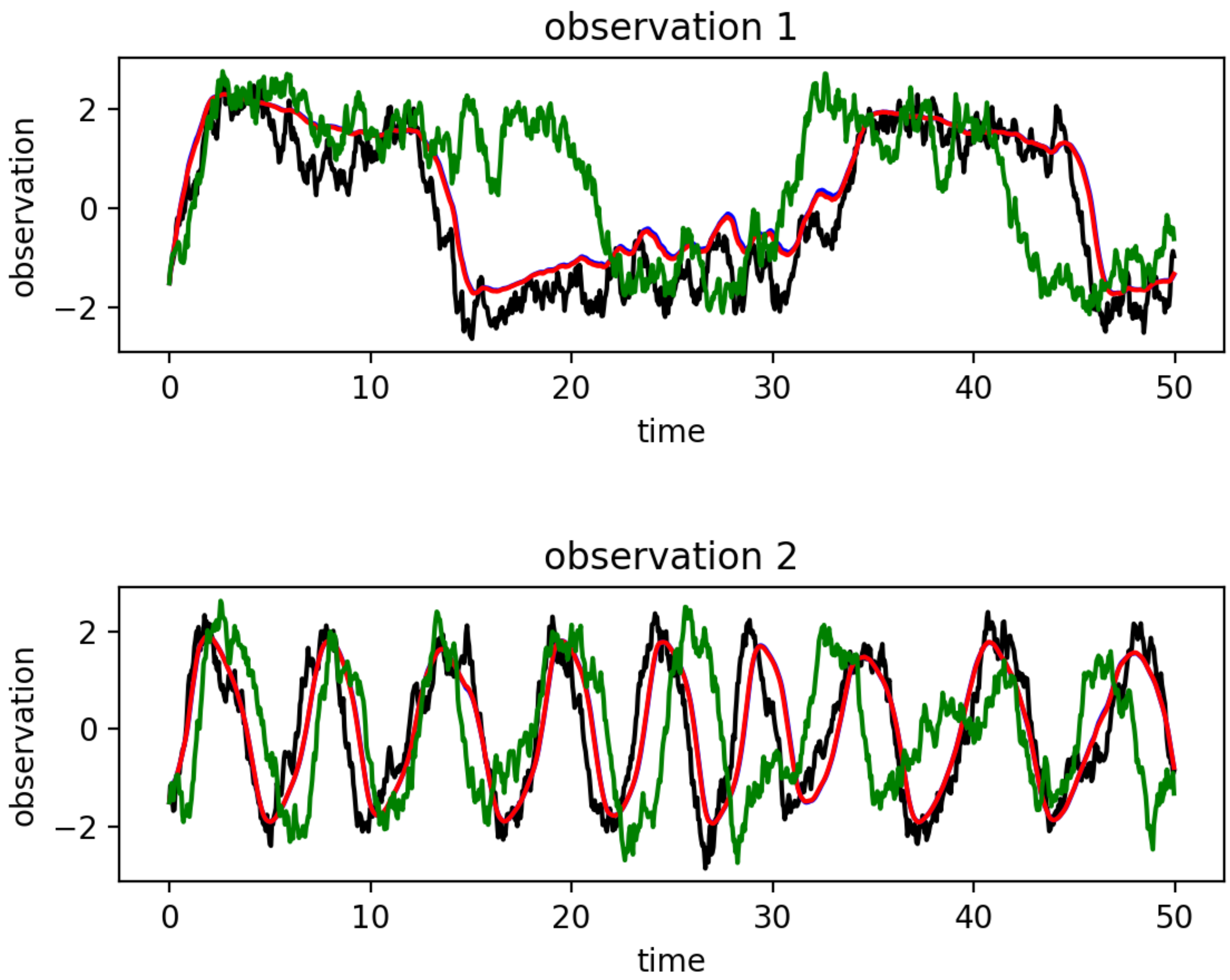


$$\mathbf{Q} = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 50 & 0 \\ 0 & 50 \end{pmatrix}$$

background observations

do fit better observations

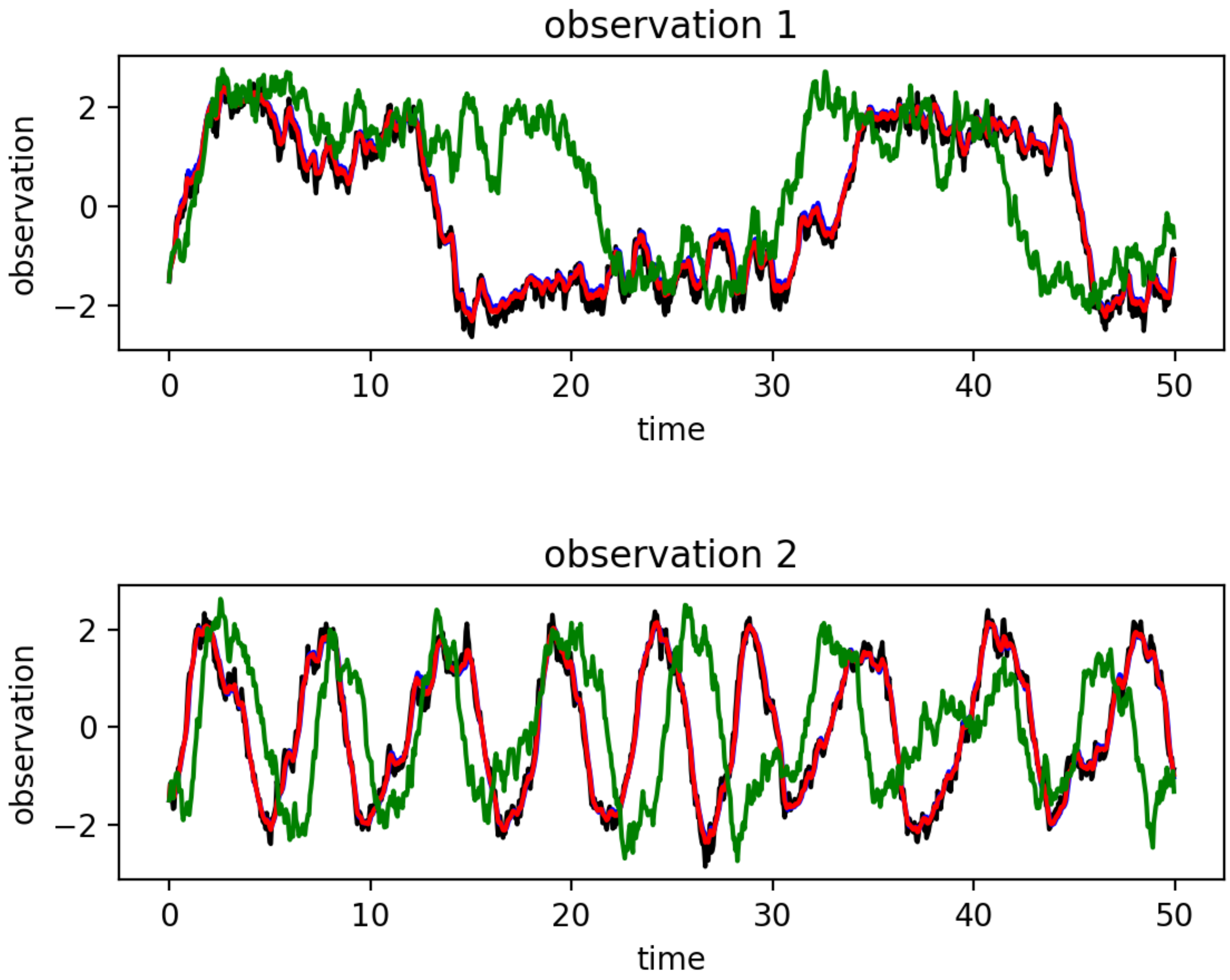


$$\mathbf{Q} = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$$

background observations

do even better fit observations



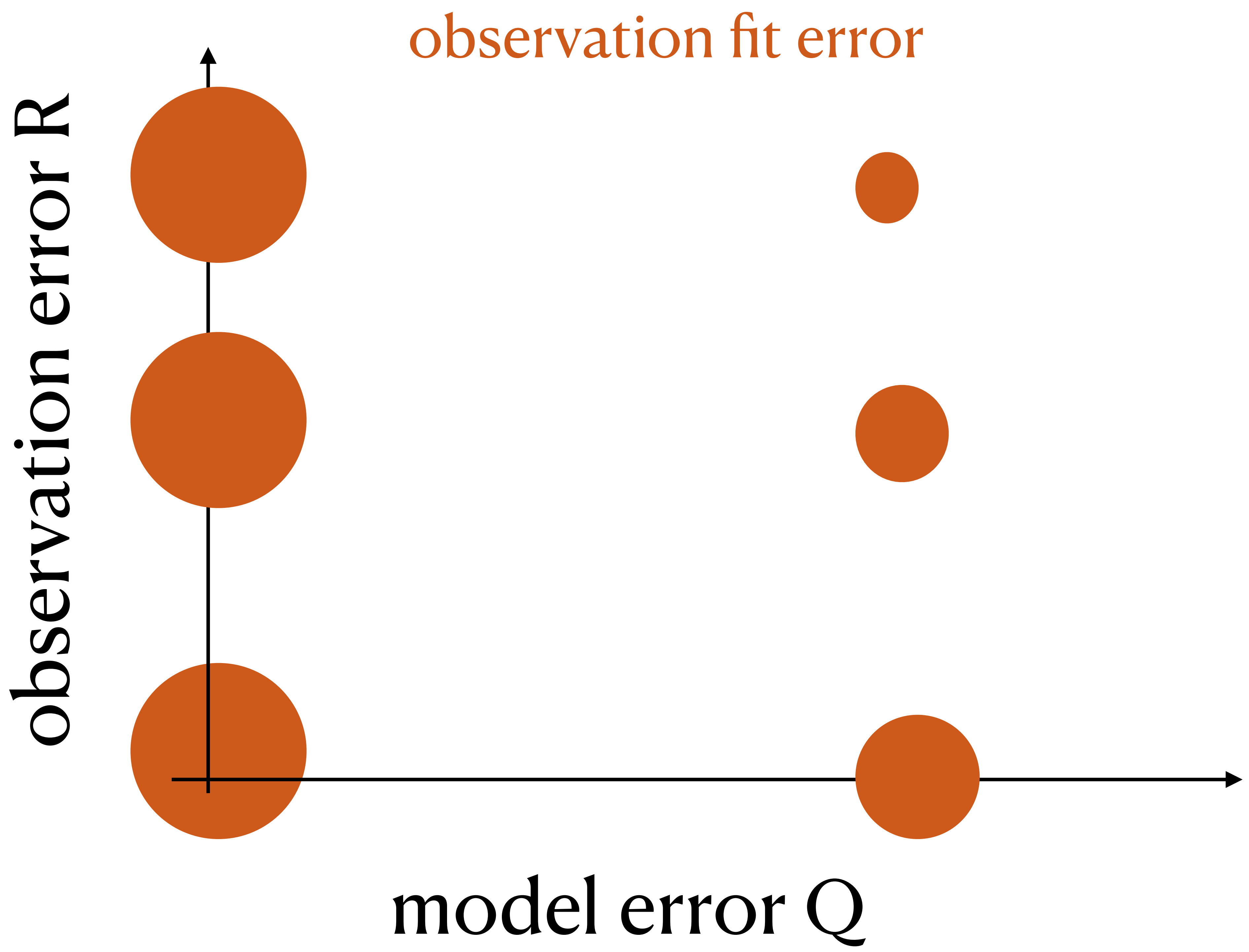
$$\mathbf{Q} = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$$

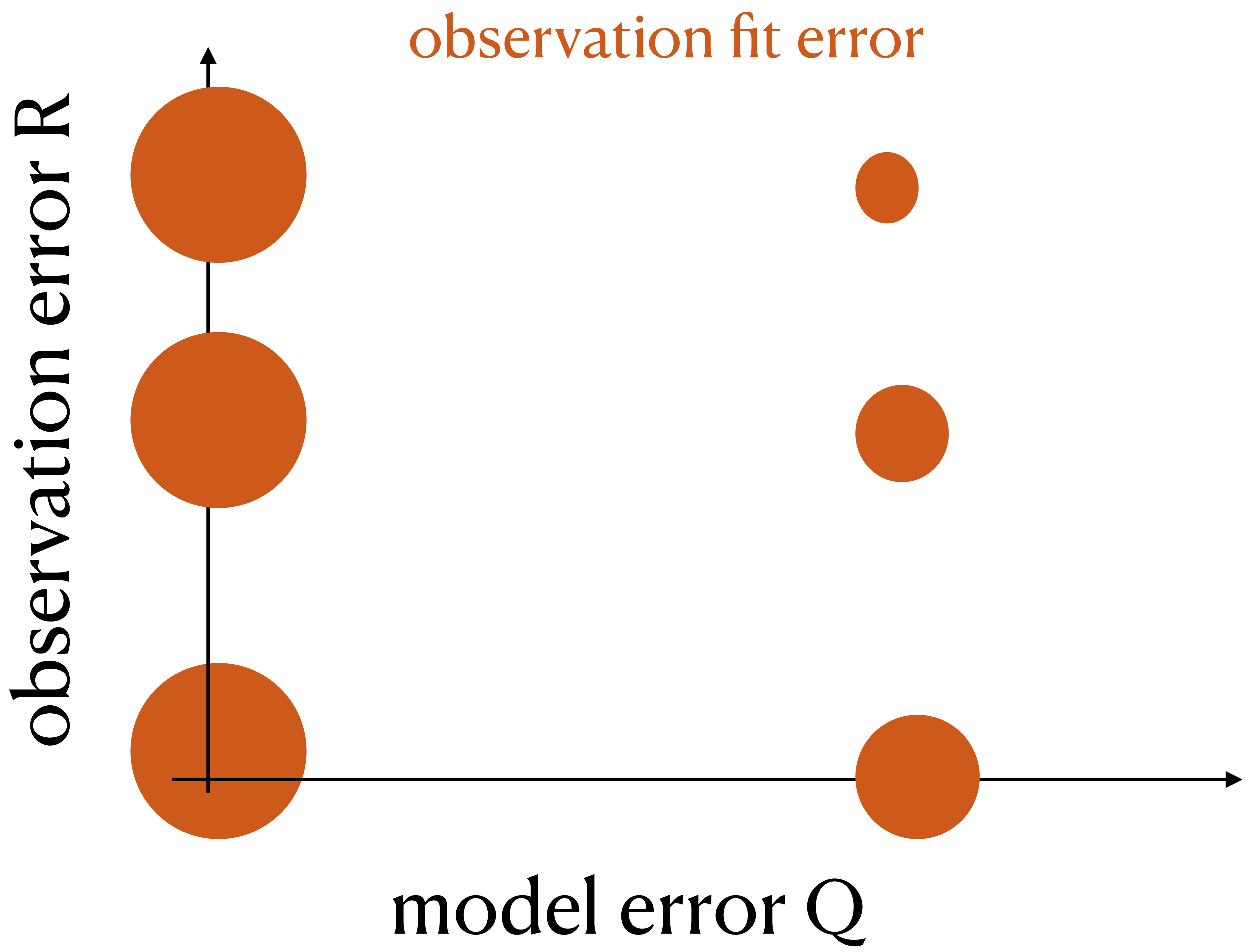
background observations

do fit observations well

short summary on impact of R and Q :



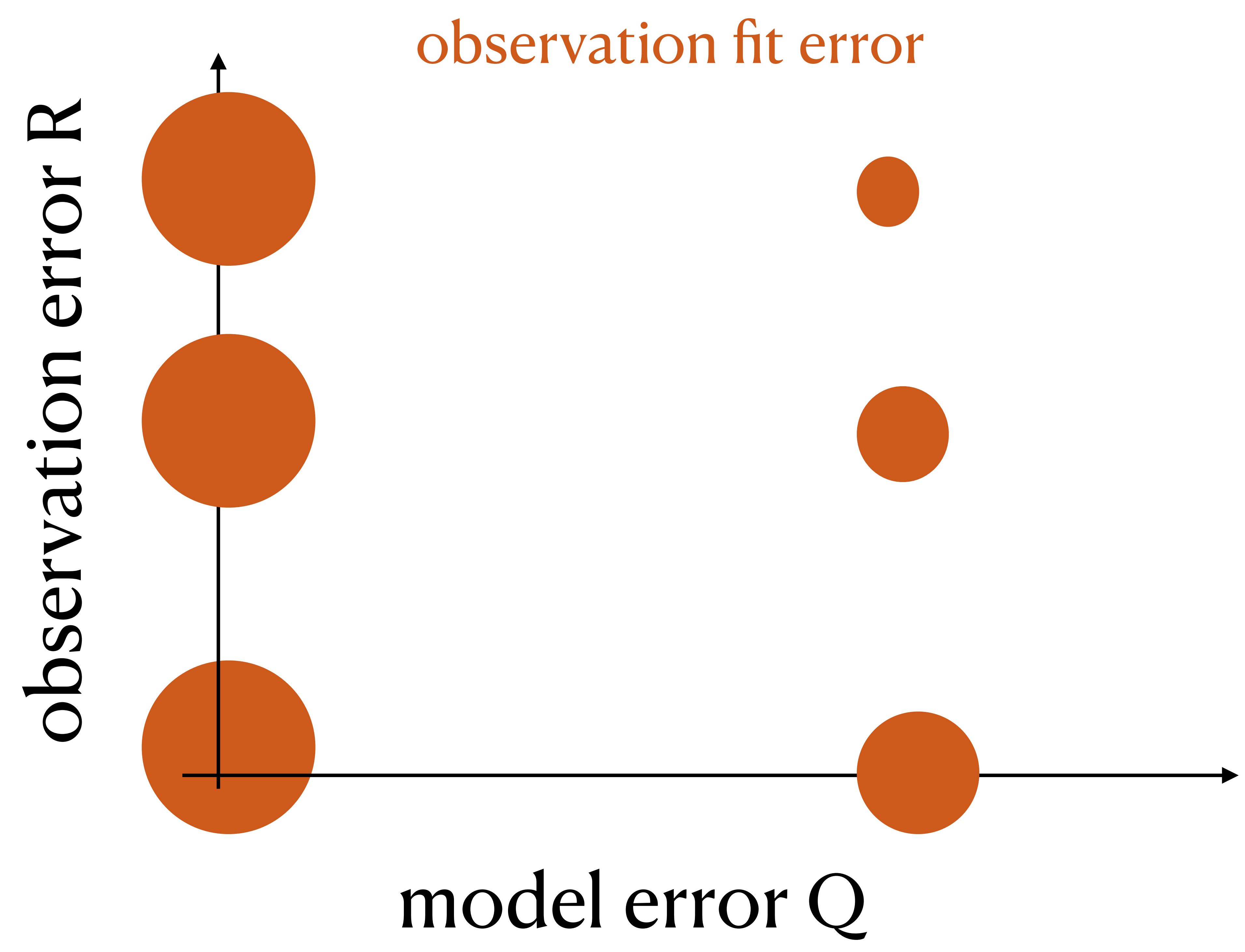
short summary on impact of R and Q :



- if there is some model error:

  the smaller the observation error R,
  the better the observation fit

short summary on impact of R and Q :

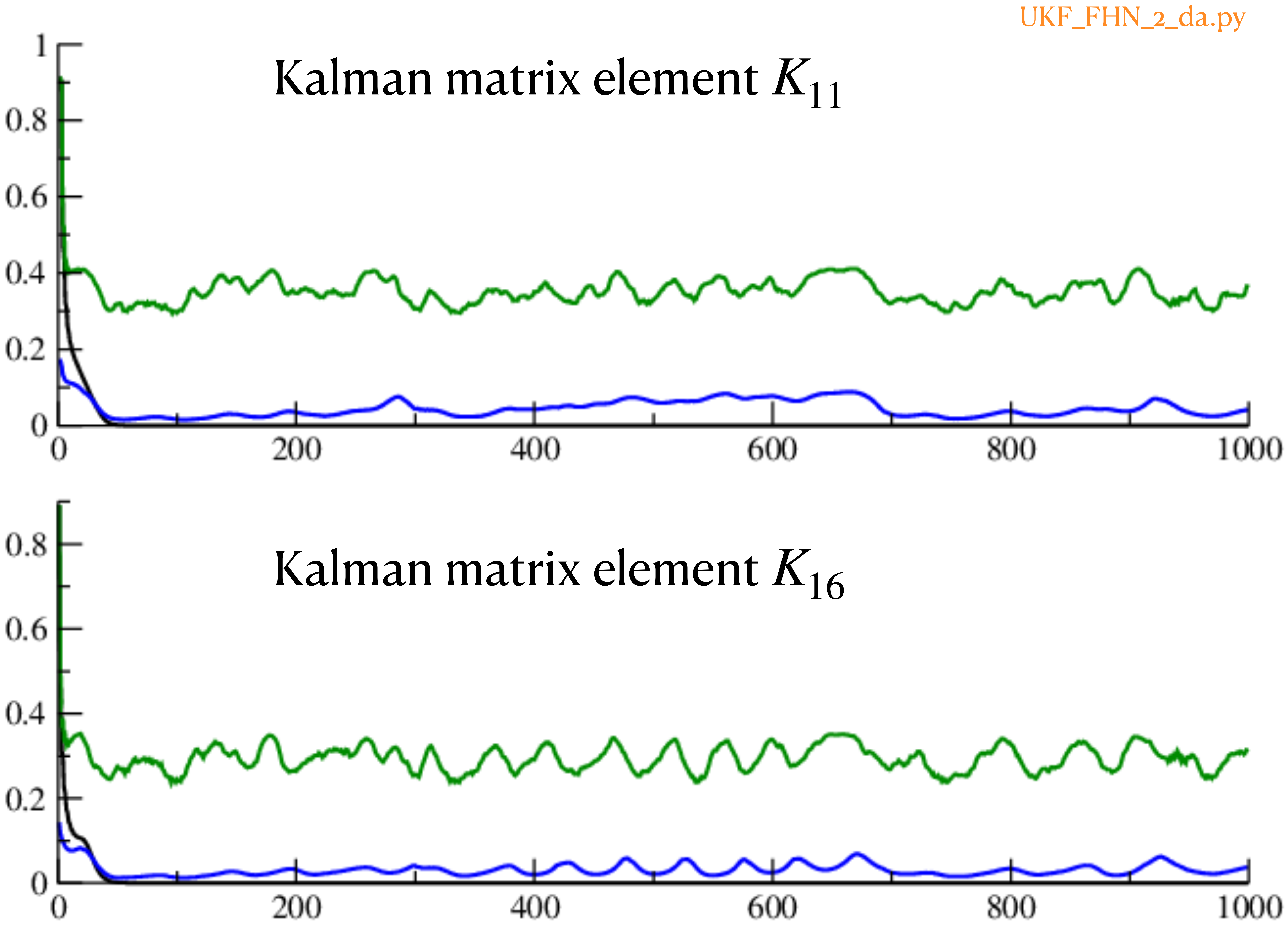


- if there is some model error:

  the smaller the observation error R,
  the better the observation fit

- without model error, observation fit is bad

explanation:



$Q_{ii}=0$, $R_{ii}=5.0$
no fit

$Q_{ii}=0.01$, $R_{ii}=5.0$

$Q_{ii}=0.01$, $R_{ii}=0.1$
best fit

recall:

$$\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)$$

$$\mathbf{K} = \mathbf{C}_{uy}\mathbf{C}_{yy}^{-1}$$

$$\mathbf{C}_{yy}(t_n) = \sum_{j=0}^{2L} W_j \left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)\left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)^t + \mathbf{R}$$

recall:

$$\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)$$

$$\mathbf{K} = \mathbf{C}_{uy}\mathbf{C}_{yy}^{-1}$$

$$\mathbf{C}_{yy}(t_n) = \sum_{j=0}^{2L} W_j \left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)\left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)^t + \mathbf{R}$$

the smaller R_{ii},

recall:

$$\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)$$

$$\mathbf{K} = \mathbf{C}_{uy}\mathbf{C}_{yy}^{-1}$$

$$\mathbf{C}_{yy}(t_n) = \sum_{j=0}^{2L} W_j \left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)\left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)^t + \mathbf{R}$$

the smaller R_{ii},

the larger the Kalman gain K

recall:

$$\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)$$

$$\mathbf{K} = \mathbf{C}_{uy}\mathbf{C}_{yy}^{-1}$$

$$\mathbf{C}_{yy}(t_n) = \sum_{j=0}^{2L} W_j \left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)\left(\mathbf{y}_b^{(j)}(t_n) - \bar{\mathbf{y}}_b(t_n)\right)^t + \mathbf{R}$$

the smaller R_{ii},

the larger the Kalman gain K

and the larger the impact of the observations

recall in addition:

$$\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{K}(\mathbf{y} - \bar{\mathbf{y}}_b)$$

one important property of the Kalman filter is

$\mathbf{x}_a \to \mathbf{x}_b$ and $\mathbf{K} \to 0$ for $t \to \infty$

Filter divergence

to prevent $\mathbf{K} \to 0$: inflation of covariance by $\mathbf{Q} > 0$

$$\mathbf{P}_b(t_n) = \sum_{j=0}^{2L} W_j^{(c)} \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right) \left(\mathbf{x}_b^{(j)}(t_n) - \bar{\mathbf{x}}_b(t_n)\right)^t + \mathbf{Q}$$

motivation

basic methods

prediction and verification

Kalman filter

linear EKF **UKF** ETKF LETKF

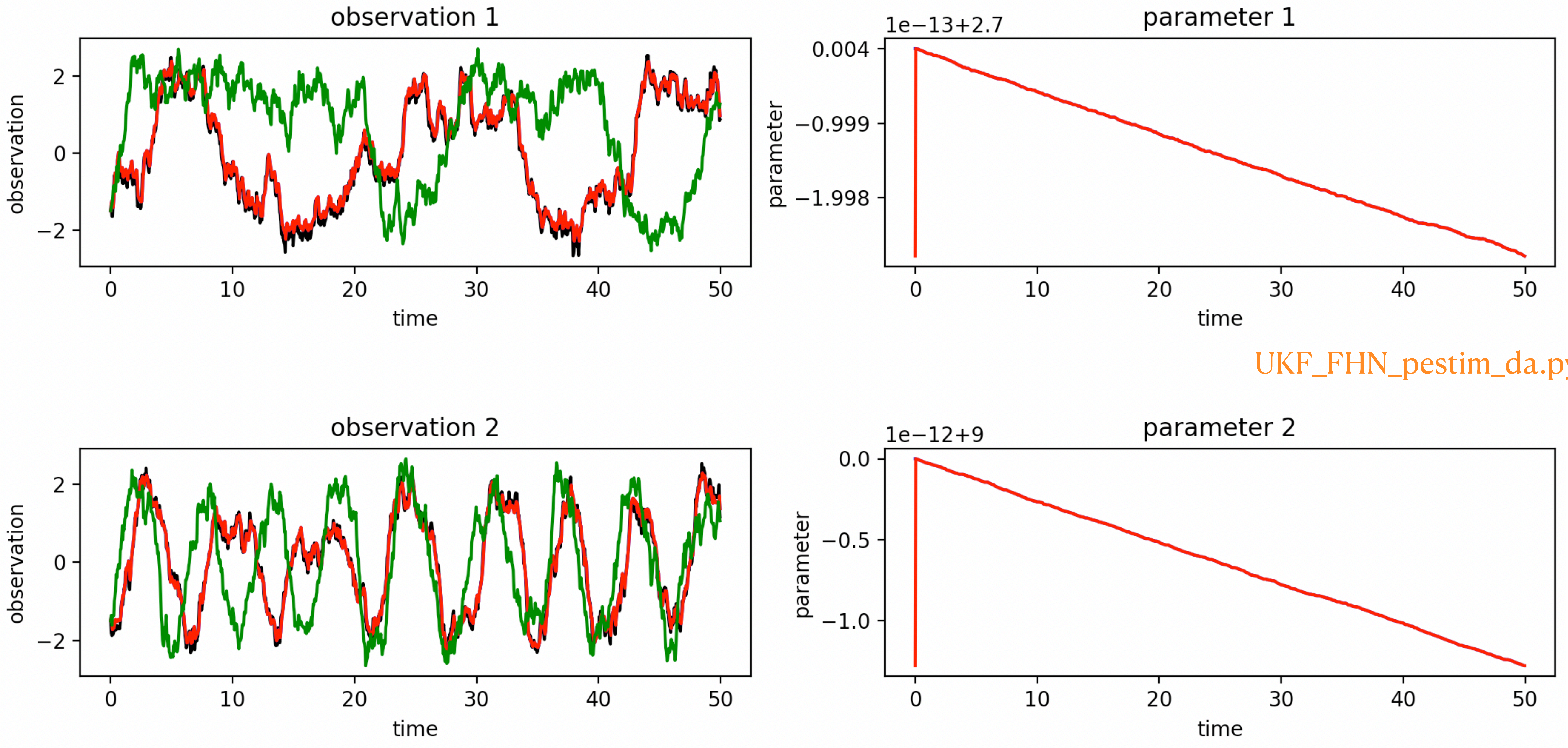
impact of R and Q

parameter estimation

parameter estimation

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0.1 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$$



UKF_FHN_pestim_da.py

poor estimation of parameters

$$x_{n+1}^{(1)} = x_n^{(1)} + \Delta t(x_n^{(1)} - \frac{1}{3}(x_{n+1}^{(1)})^3 - y_n^{(1)} + I_1) + \xi_n^{(1)}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + \Delta t(b_{11}x^{(1)} + b_{01} - y_n^{(1)})/\tau_1$$

$$x_{n+1}^{(2)} = x_n^{(2)} + \Delta t(x_n^{(2)} - \frac{1}{3}(x_{n+1}^{(2)})^3 - y_n^{(2)} + I_2) + \xi_n^{(2)}$$

$$y_{n+1}^{(2)} = y_n^{(2)} + \Delta t(b_{12}x^{(2)} + b_{02} - y_n^{(2)})/\tau_2$$

$$(I_1)_{n+1} = (I_1)_n$$

$$(\tau_2)_{n+1} = (\tau_2)_n$$

parameters affect state

$$x_{n+1}^{(1)} = x_n^{(1)} + \Delta t(x_n^{(1)} - \frac{1}{3}(x_{n+1}^{(1)})^3 - y_n^{(1)} + I_1) + \xi_n^{(1)}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + \Delta t(b_{11}x^{(1)} + b_{01} - y_n^{(1)})/\tau_1$$

$$x_{n+1}^{(2)} = x_n^{(2)} + \Delta t(x_n^{(2)} - \frac{1}{3}(x_{n+1}^{(2)})^3 - y_n^{(2)} + I_2) + \xi_n^{(2)}$$

$$y_{n+1}^{(2)} = y_n^{(2)} + \Delta t(b_{12}x^{(2)} + b_{02} - y_n^{(2)})/\tau_2$$

$$(I_1)_{n+1} = (I_1)_n$$

$$(\tau_2)_{n+1} = (\tau_2)_n$$

but state does not affect parameters

→ no information flow from model to parameters

→ parameters do weakly change

improvement

$$x_{n+1}^{(1)} = x_n^{(1)} + \Delta t(x_n^{(1)} - \frac{1}{3}(x_{n+1}^{(1)})^3 - y_n^{(1)} + I_1) + \xi_n^{(1)}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + \Delta t(b_{11}x^{(1)} + b_{01} - y_n^{(1)})/\tau_1$$

$$x_{n+1}^{(2)} = x_n^{(2)} + \Delta t(x_n^{(2)} - \frac{1}{3}(x_{n+1}^{(2)})^3 - y_n^{(2)} + I_2) + \xi_n^{(2)}$$

$$y_{n+1}^{(2)} = y_n^{(2)} + \Delta t(b_{12}x^{(2)} + b_{02} - y_n^{(2)})/\tau_2$$

$$(I_1)_{n+1} = (I_1)_n + \rho_n^{(1)}$$

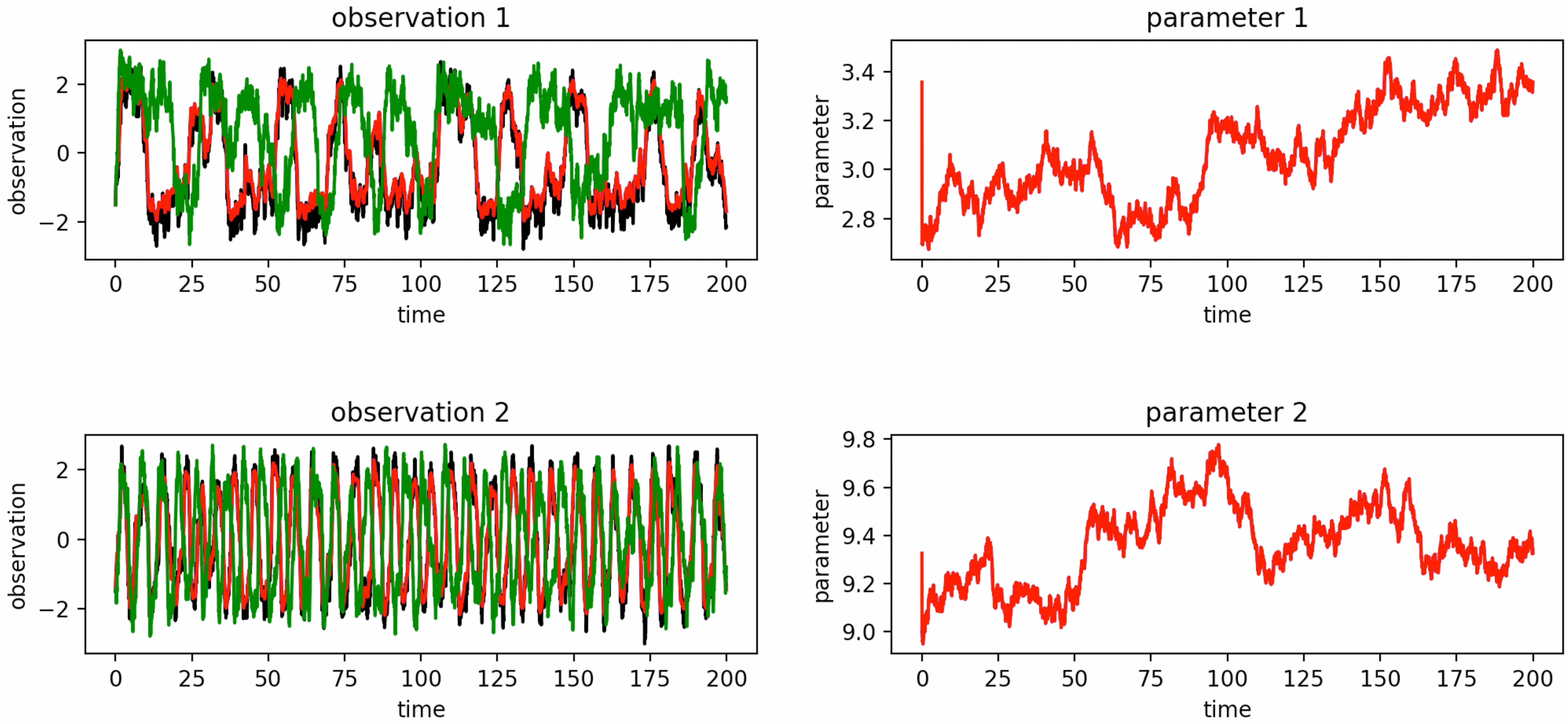
$$(\tau_2)_{n+1} = (\tau_2)_n + \rho_n^{(2)}$$

parameters change stochastically

with noise $\rho_n^{(1,2)}$

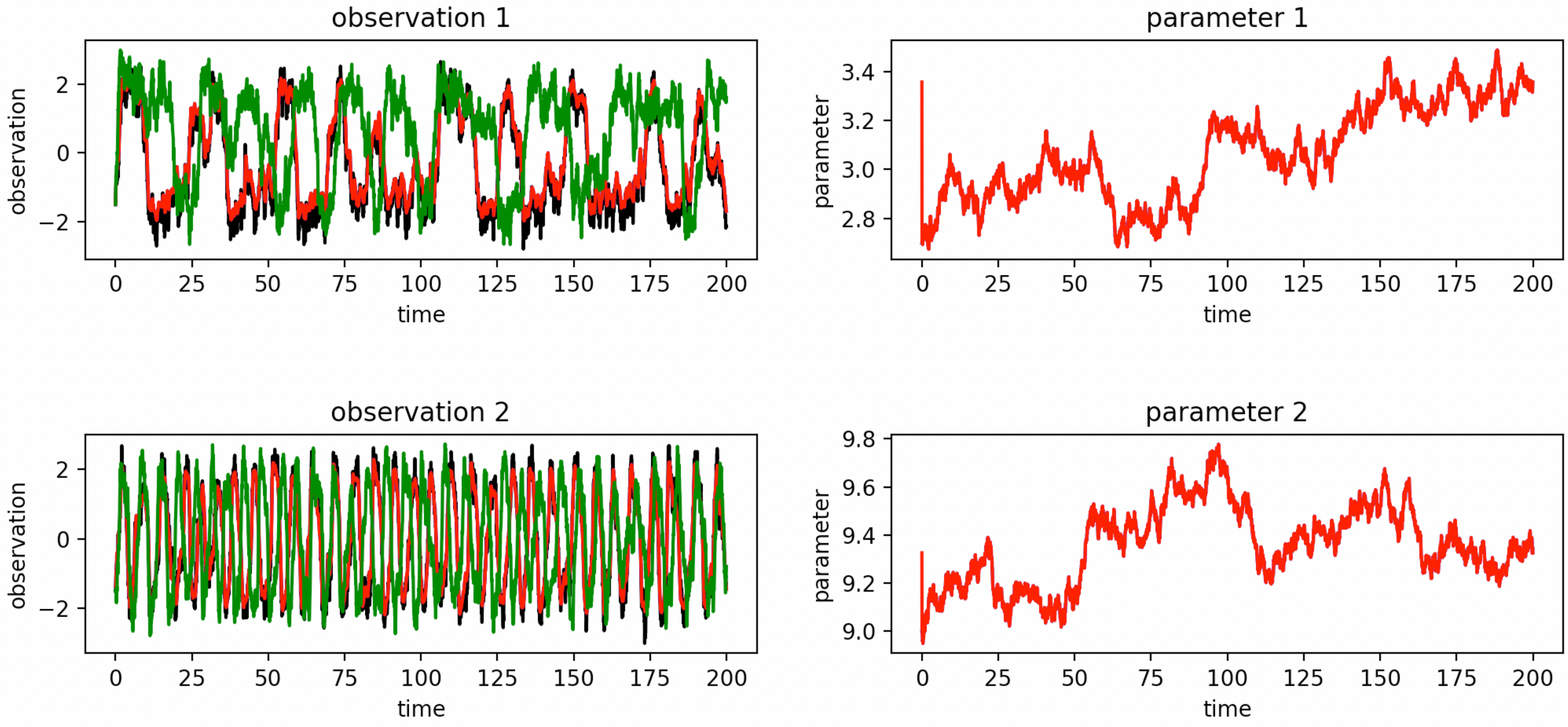
$$E[\rho_n^{(1,2)}] = 0$$

$$Var[\rho_n^{(1,2)}] = Q_2$$



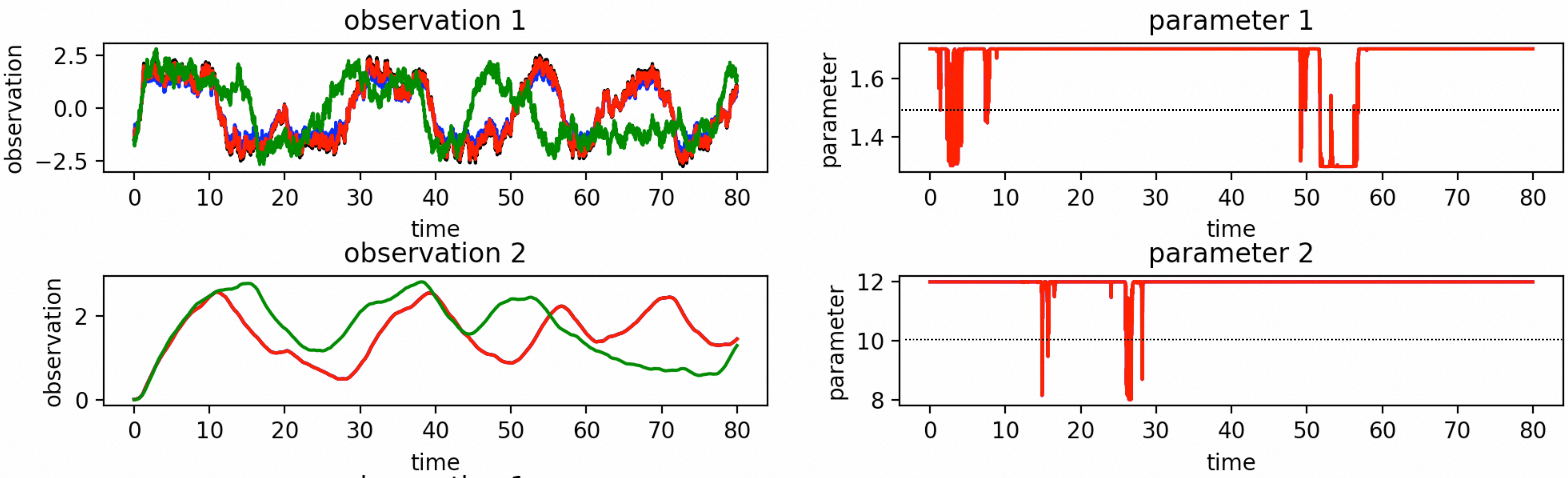
UKF_FHN_pestim_da.py

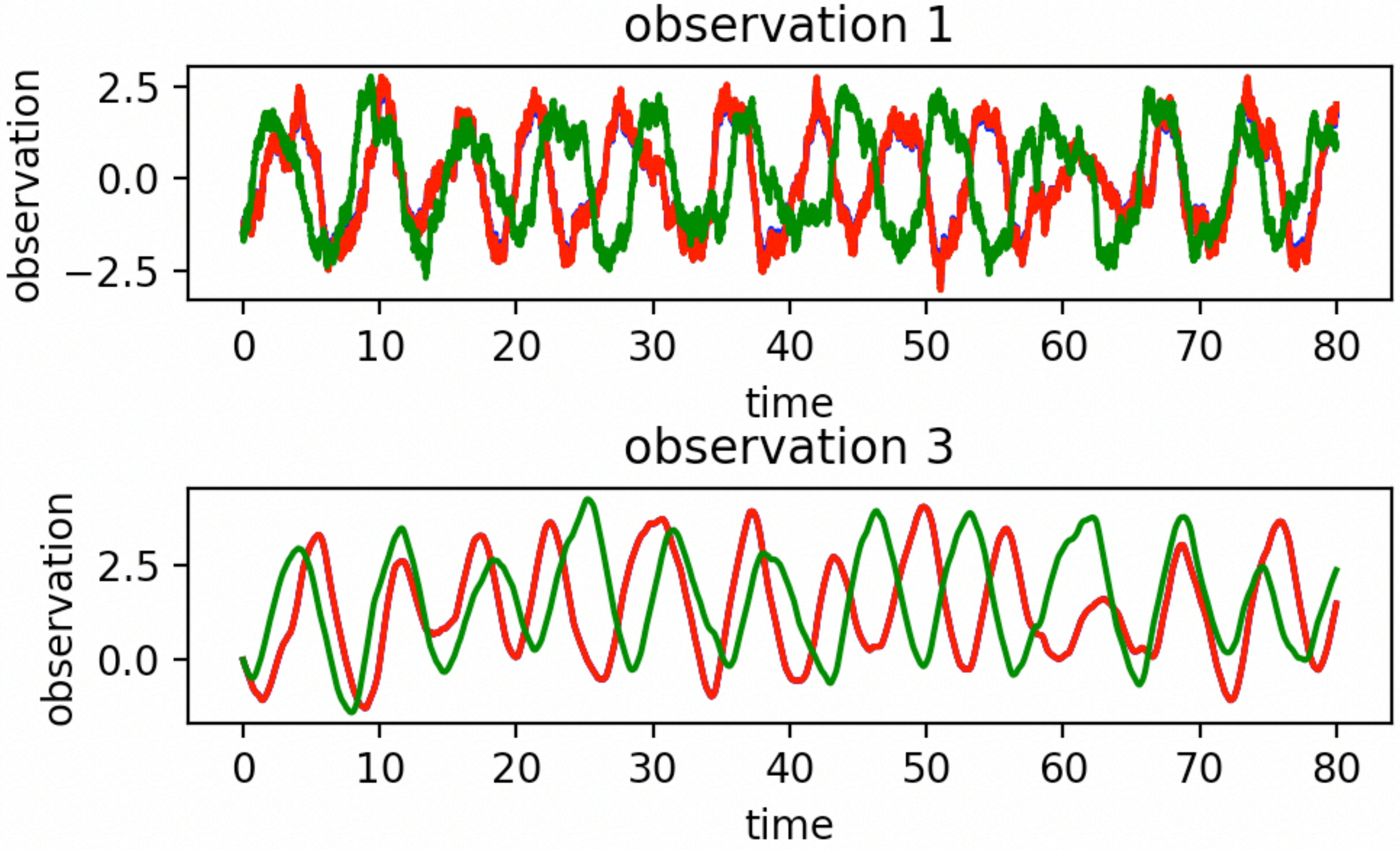
parameters change in time, but still poor estimation



parameters change in time, but still poor estimation

→ different parameter estimation techniques necessary





possible improvements:

- constraint of parameters between borders
- increase number of observations

slightly improved parameter estimation

please take a look at huge literature on parameter estimation, e.g. keywords are

- choice of sigma points, e.g. star sigma points or simplex sigma points

$$\mathbf{s}_j(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) + \boxed{I_j} \mathbf{A}_j$$

$$\mathbf{s}_{j+L}(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) - \boxed{I_{j+L}} \mathbf{A}_j$$

please take a look at huge literature on parameter estimation, e.g. keywords are

- choice of sigma points, e.g. star sigma points or simplex sigma points

$$\mathbf{s}_j(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) + \boxed{I_j}\mathbf{A}_j$$

$$\mathbf{s}_{j+L}(t_{n-1}) = \bar{\mathbf{x}}_a(t_{n-1}) - \boxed{I_{j+L}}\mathbf{A}_j$$

- parameter identifiability analysis

- adaptive UKF

- weight decay method to adapt stochastic parameter noise $\rho_n^{(1,2)}$

motivation

basic methods

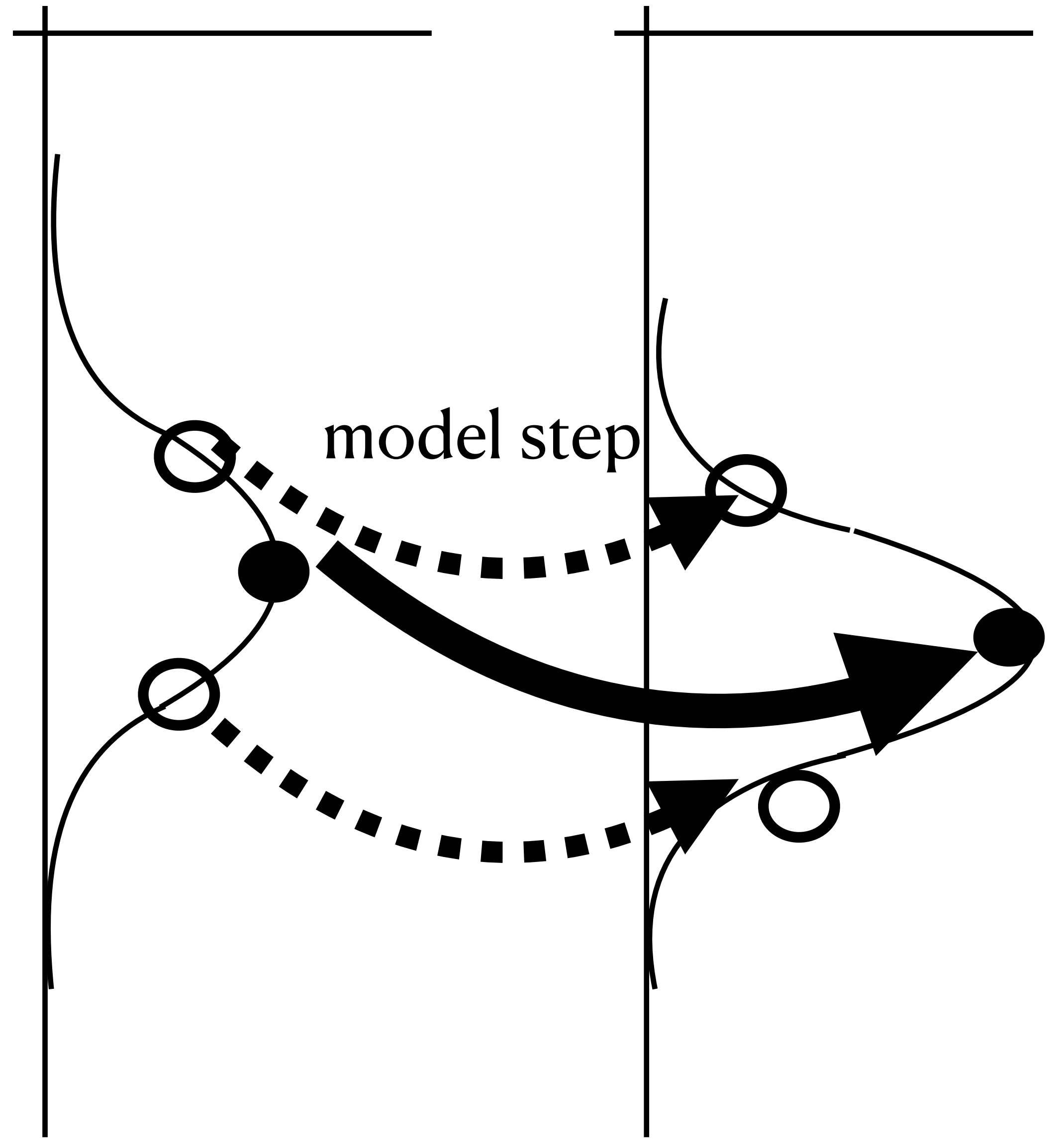
Kalman filter

linear EKF UKF **ETKF** LETKF

prediction and verification

Ensemble Transform Kalman filter (ETKF)

Unscented Kalman filter



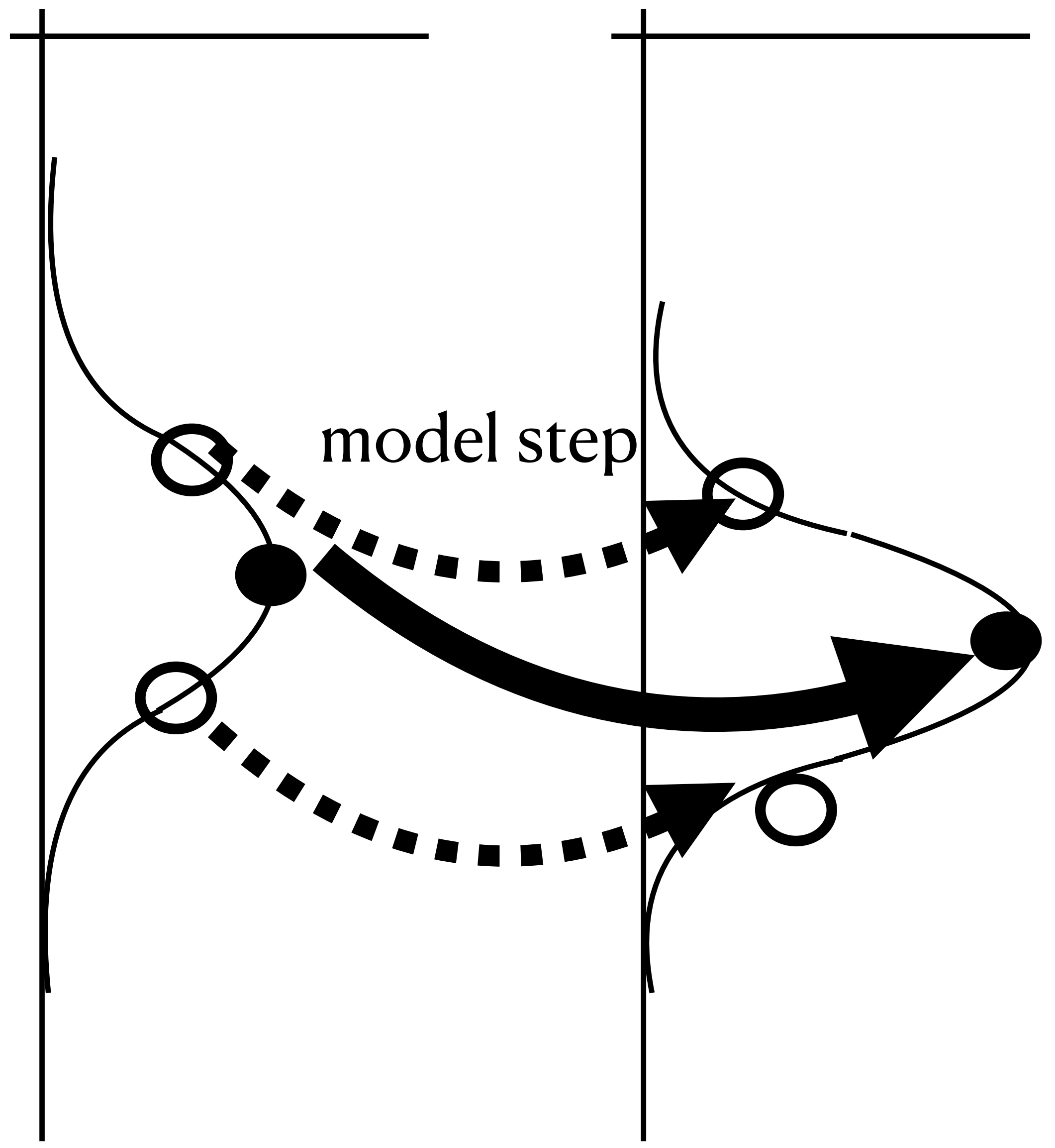
nonlinear model

nonlinear map of distribution

sigma points

Ensemble Transform Kalman filter (ETKF)

Unscented Kalman filter

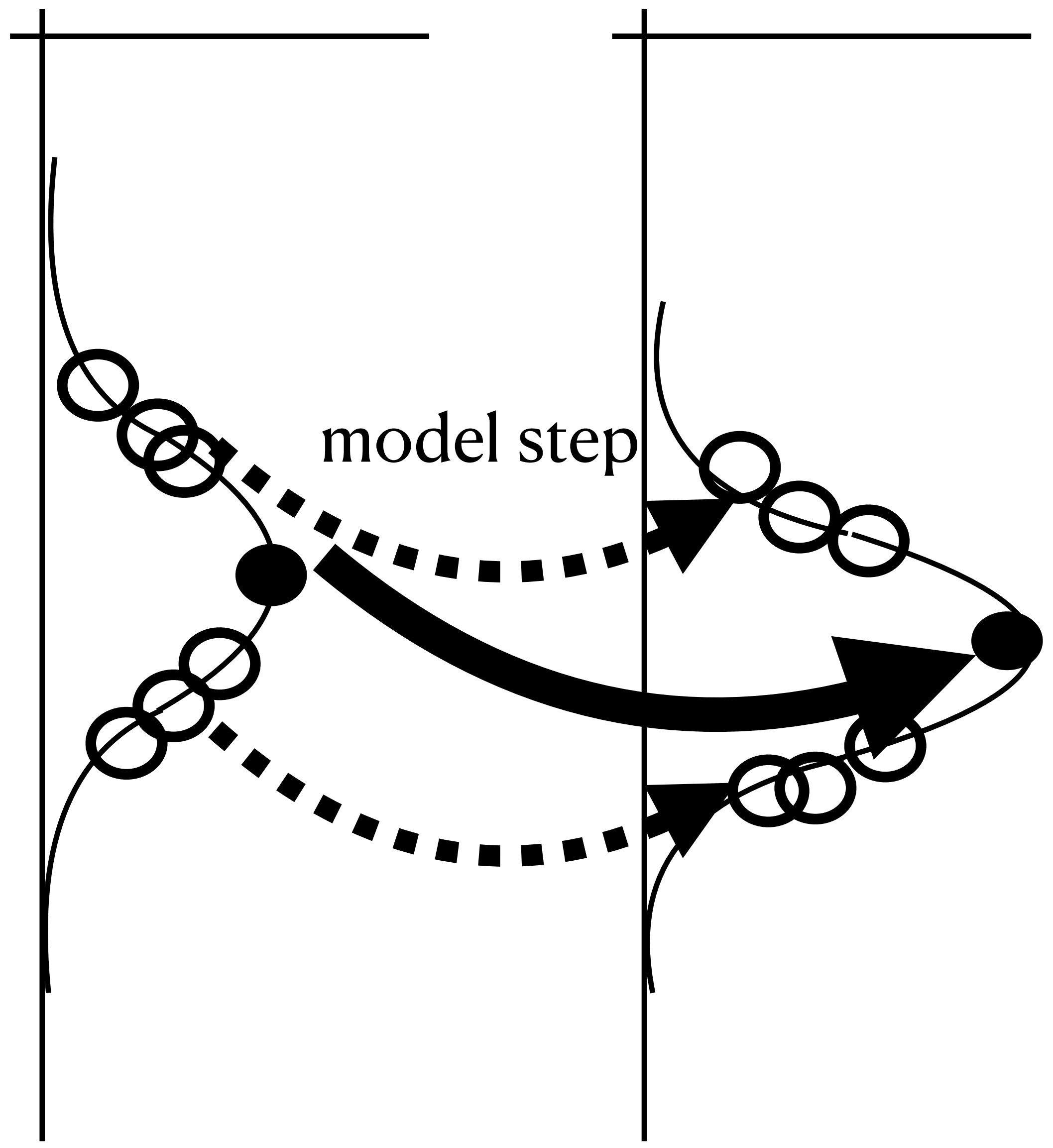


nonlinear model

nonlinear map of distribution

sigma points

Ensemble Kalman filter

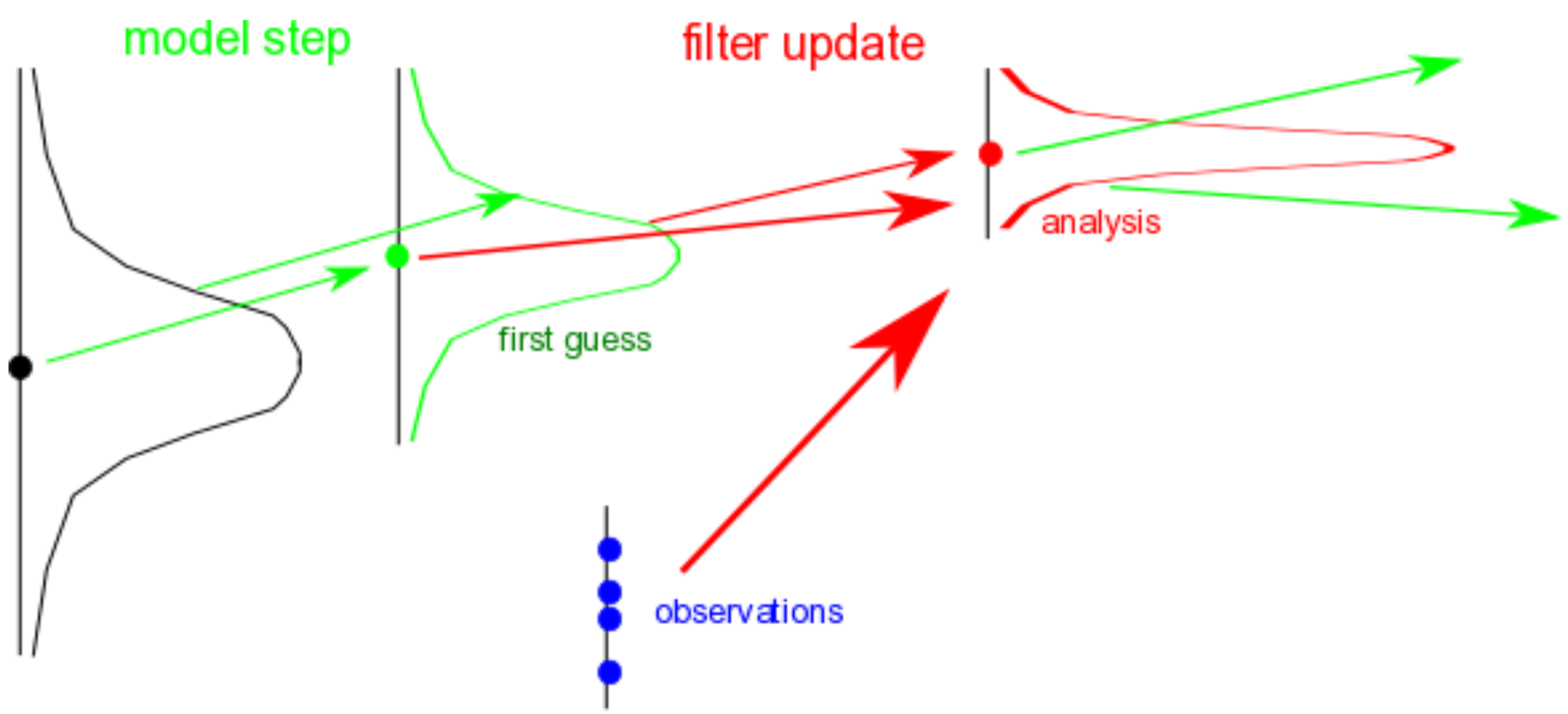


nonlinear model

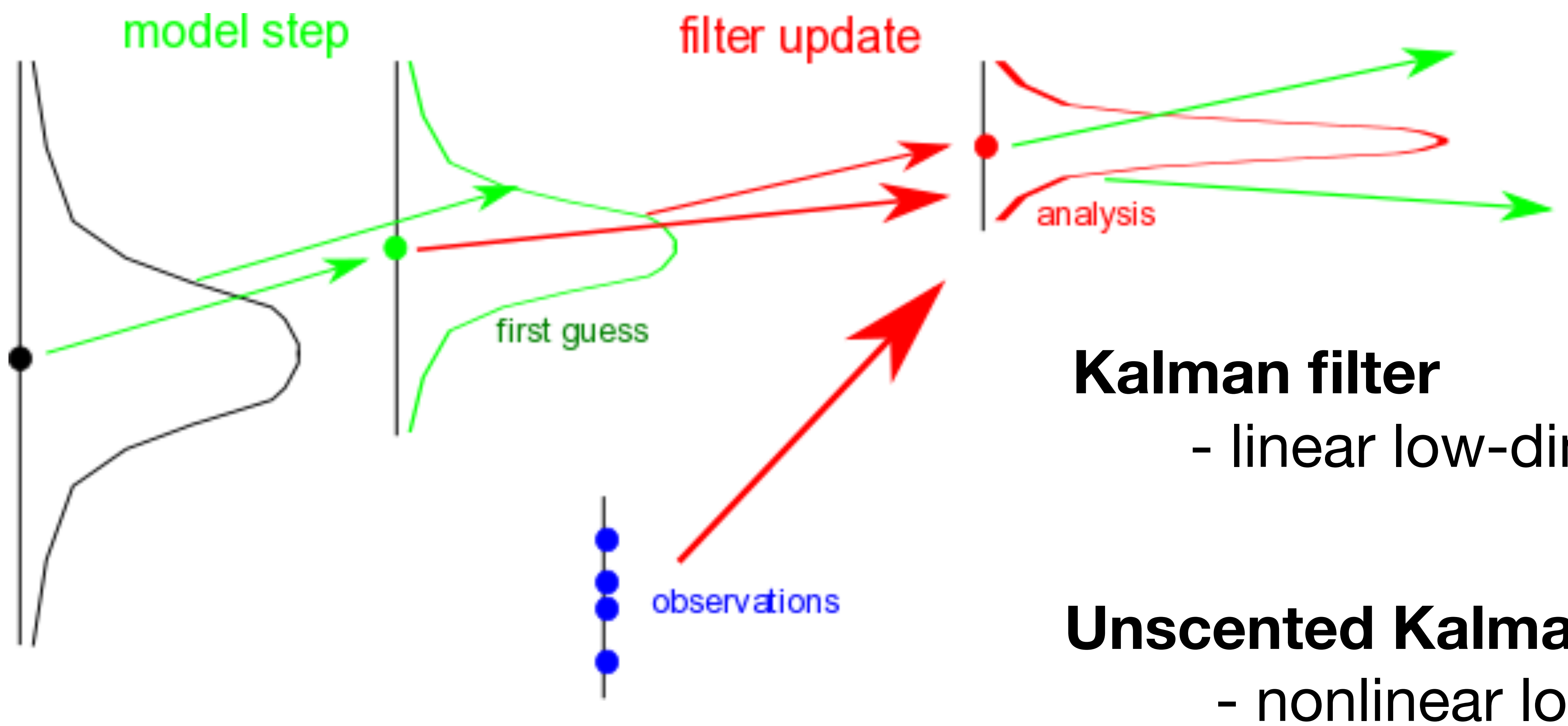
nonlinear map of distribution

ensemble of sample points

Kalman filtering



Kalman filtering



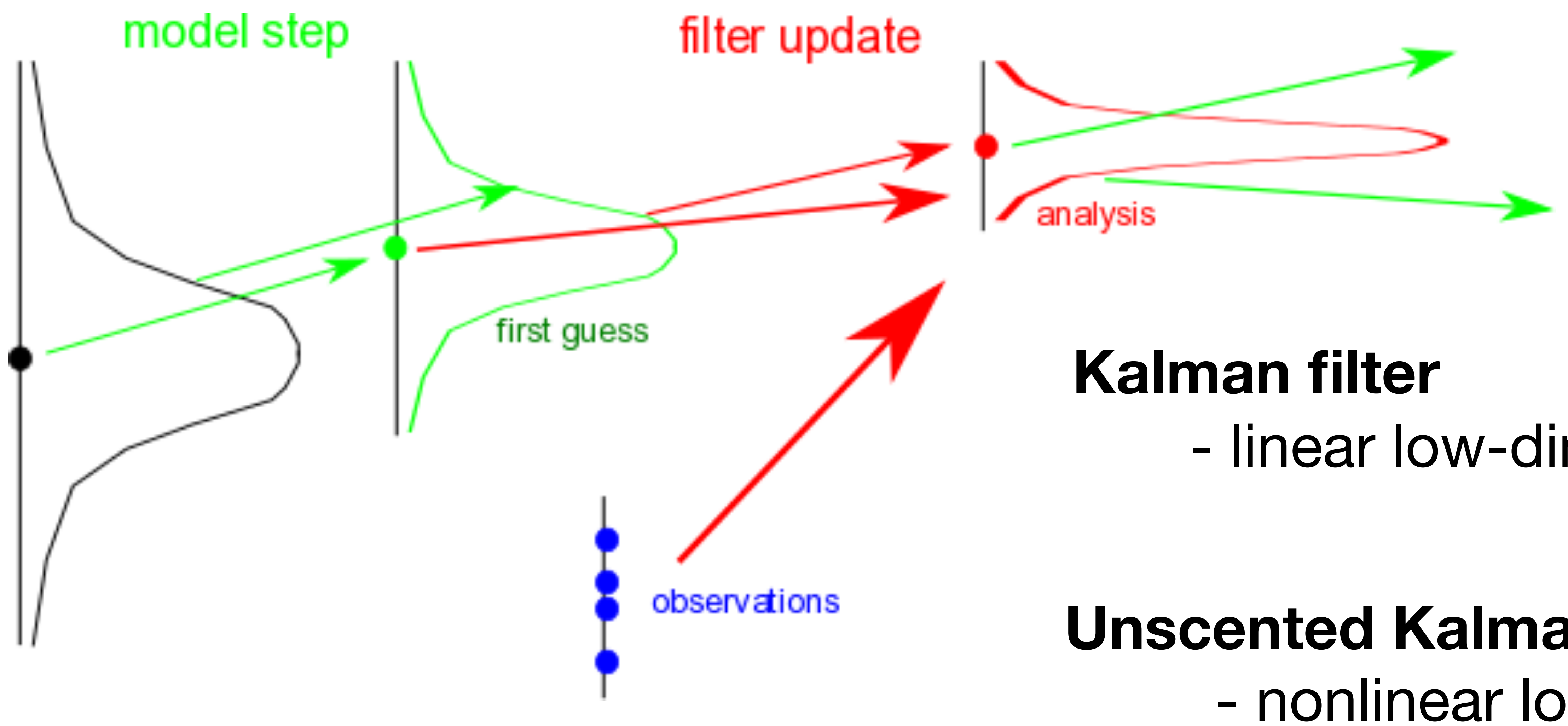
Kalman filter

- linear low-dimensional models

Unscented Kalman filter

- nonlinear low-dimensional systems

Kalman filtering

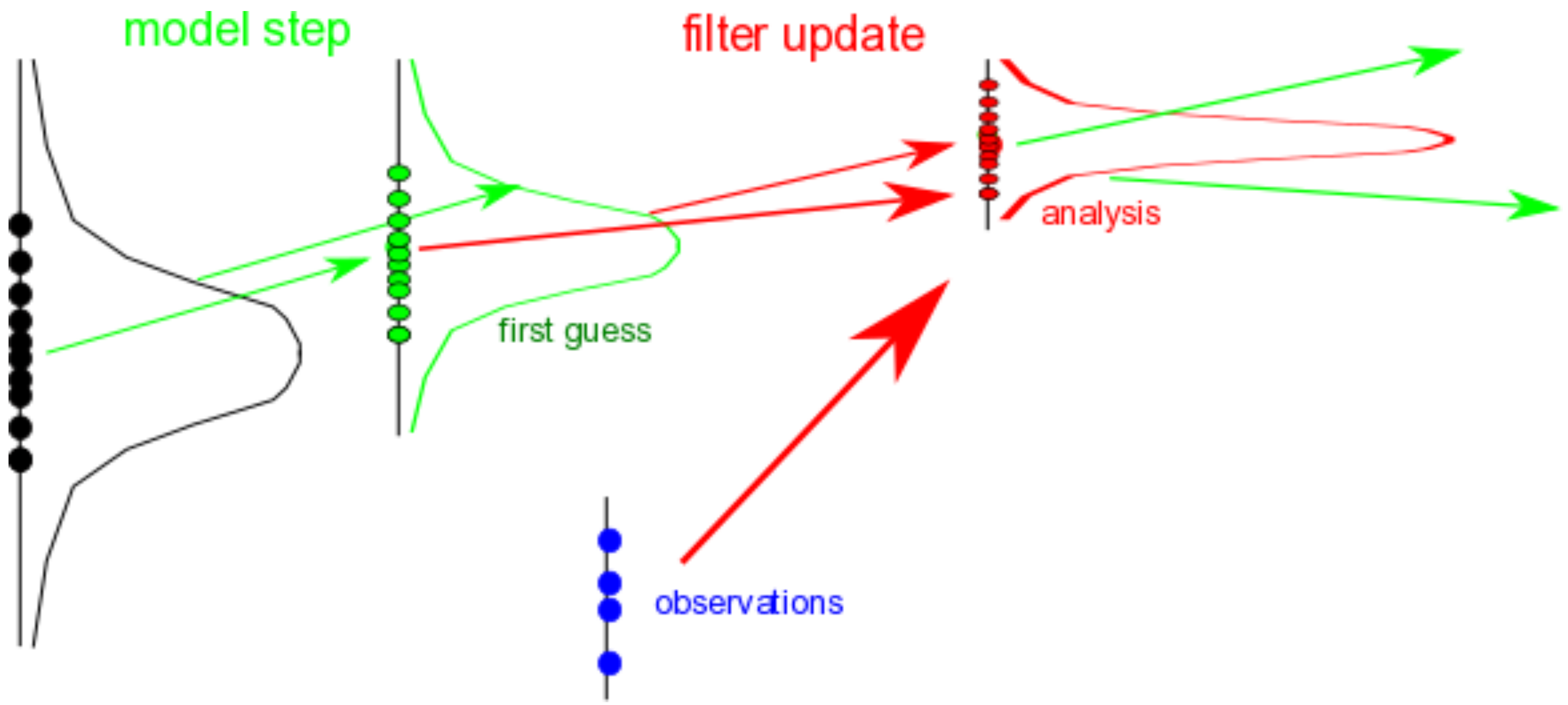


Kalman filter

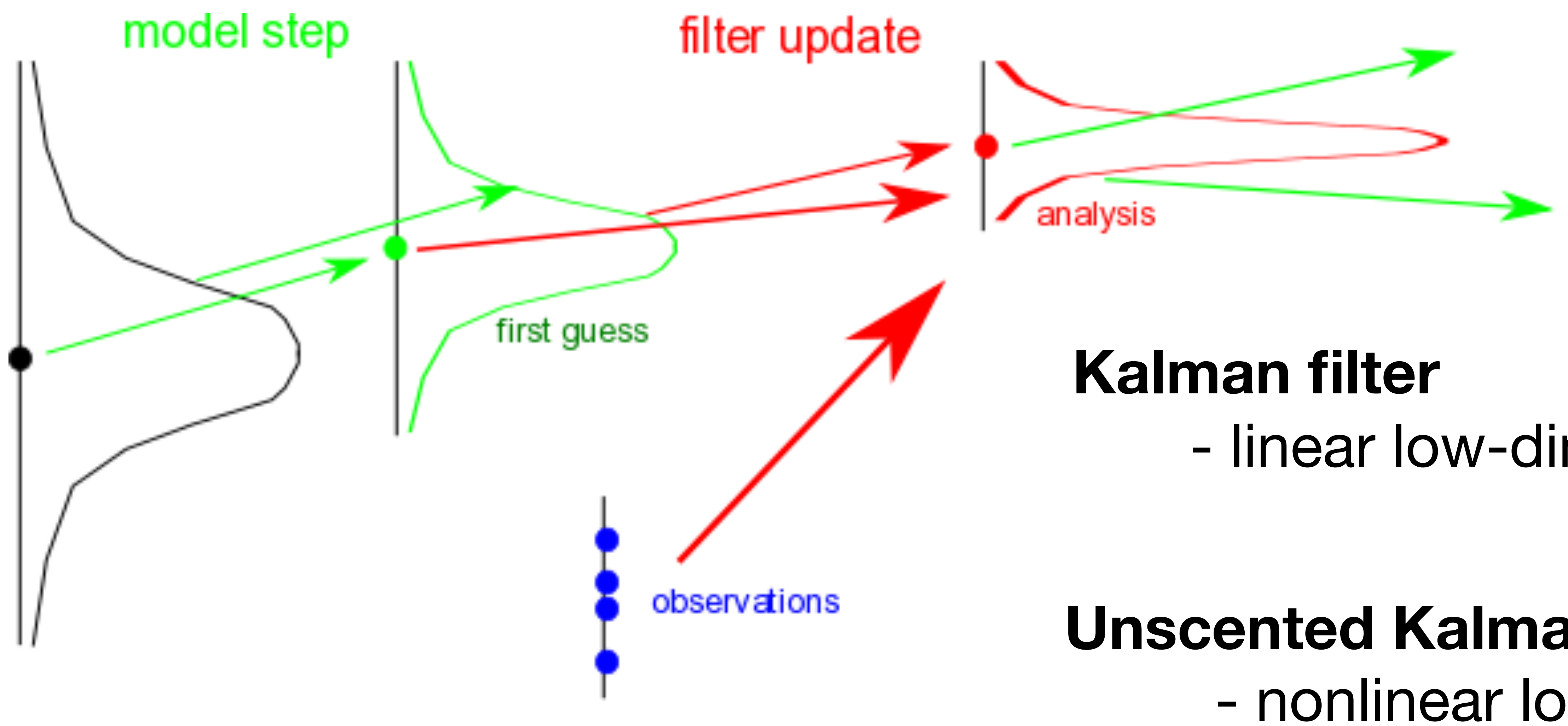
- linear low-dimensional models

Unscented Kalman filter

- nonlinear low-dimensional systems

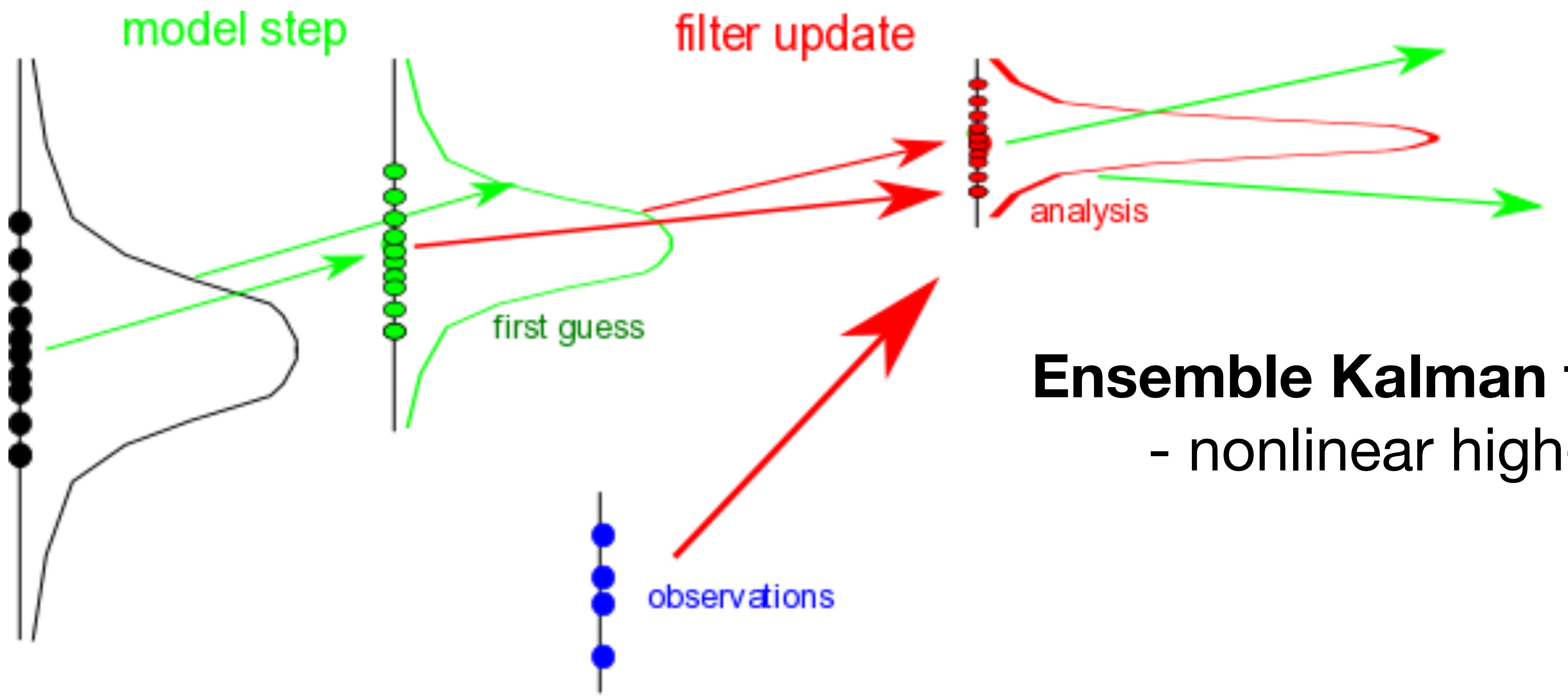


Kalman filtering

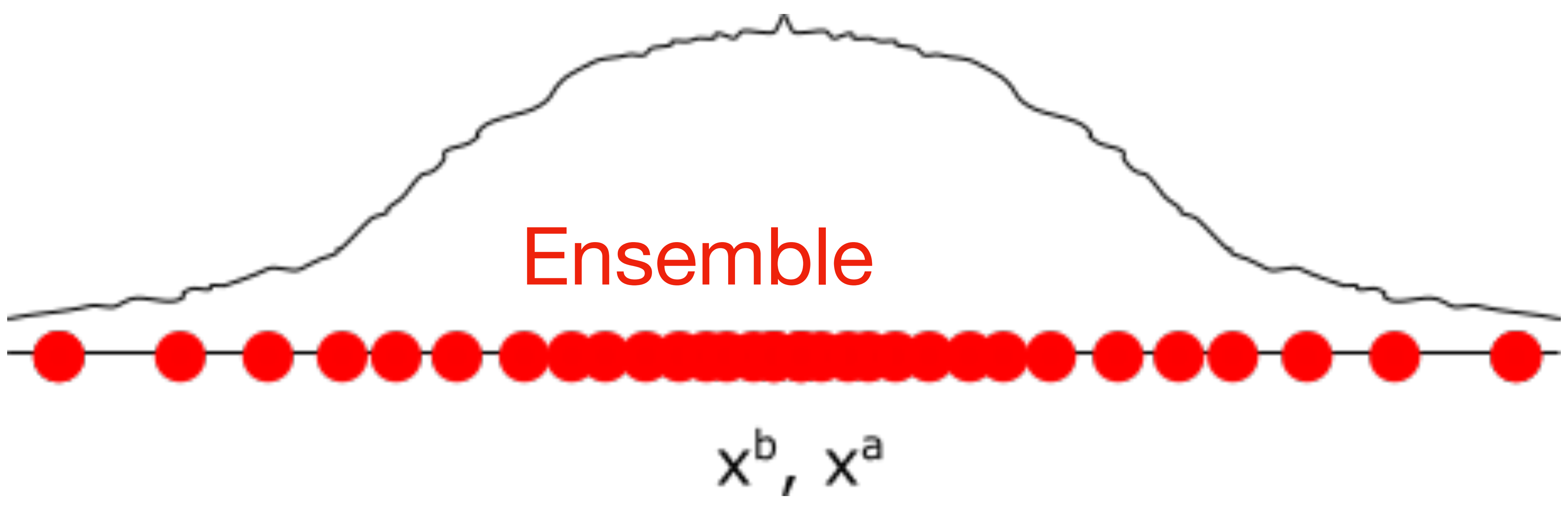


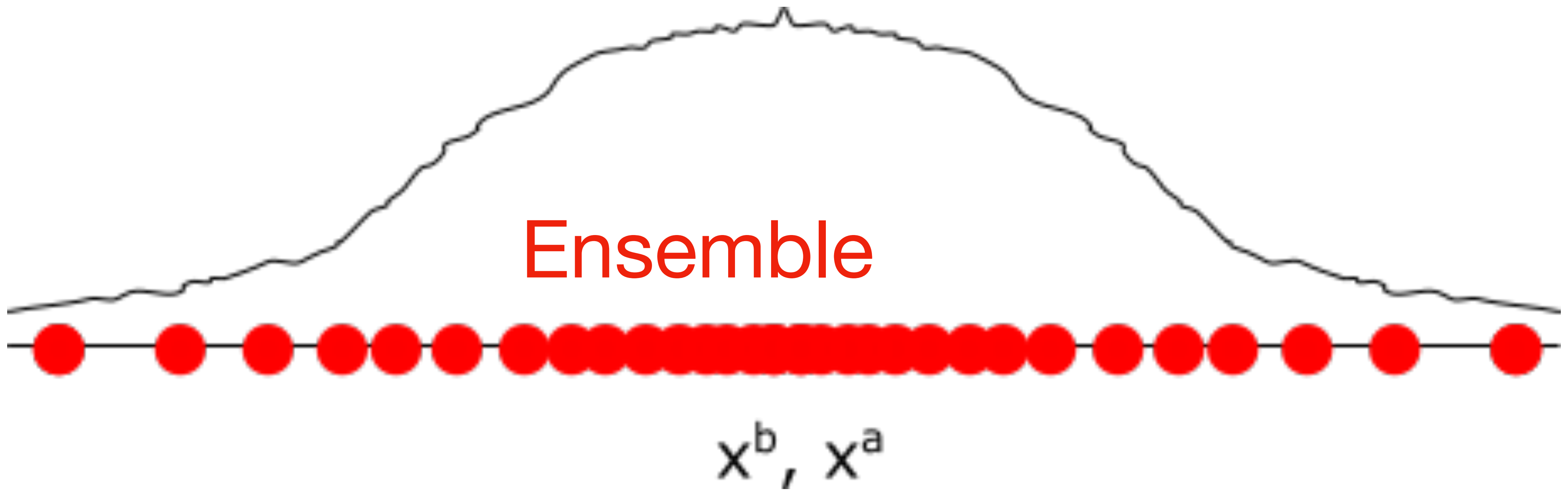
Kalman filter
- linear low-dimensional models

Unscented Kalman filter
- nonlinear low-dimensional systems



Ensemble Kalman filter
- nonlinear high-dimensional models

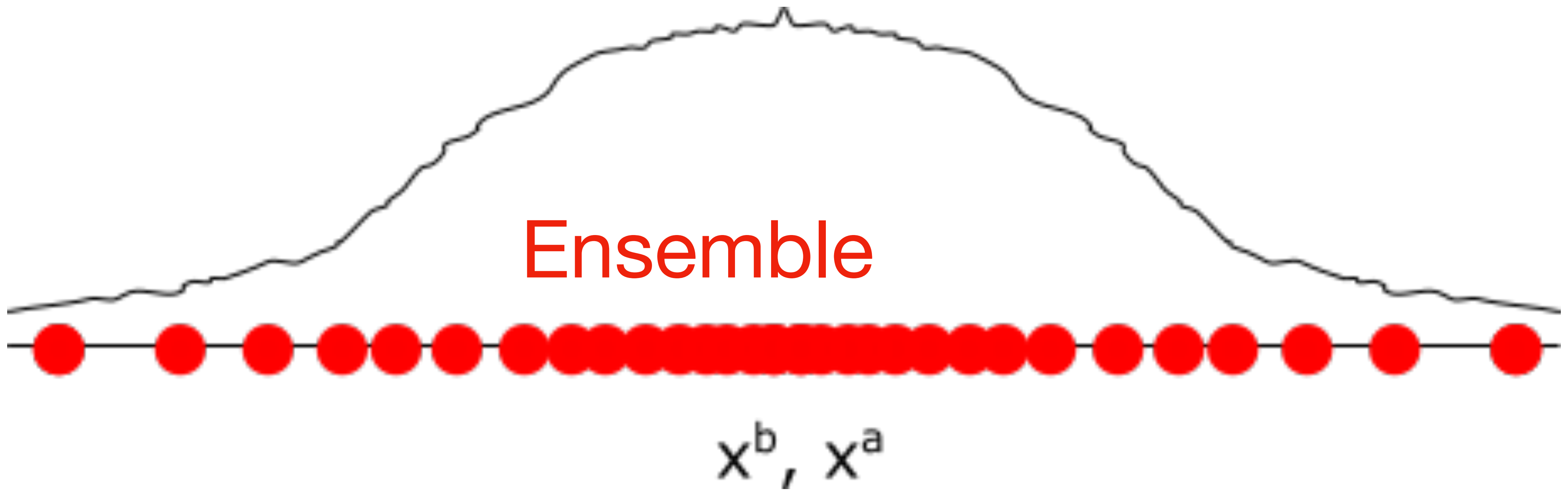

Ensemble
x^b, x^a



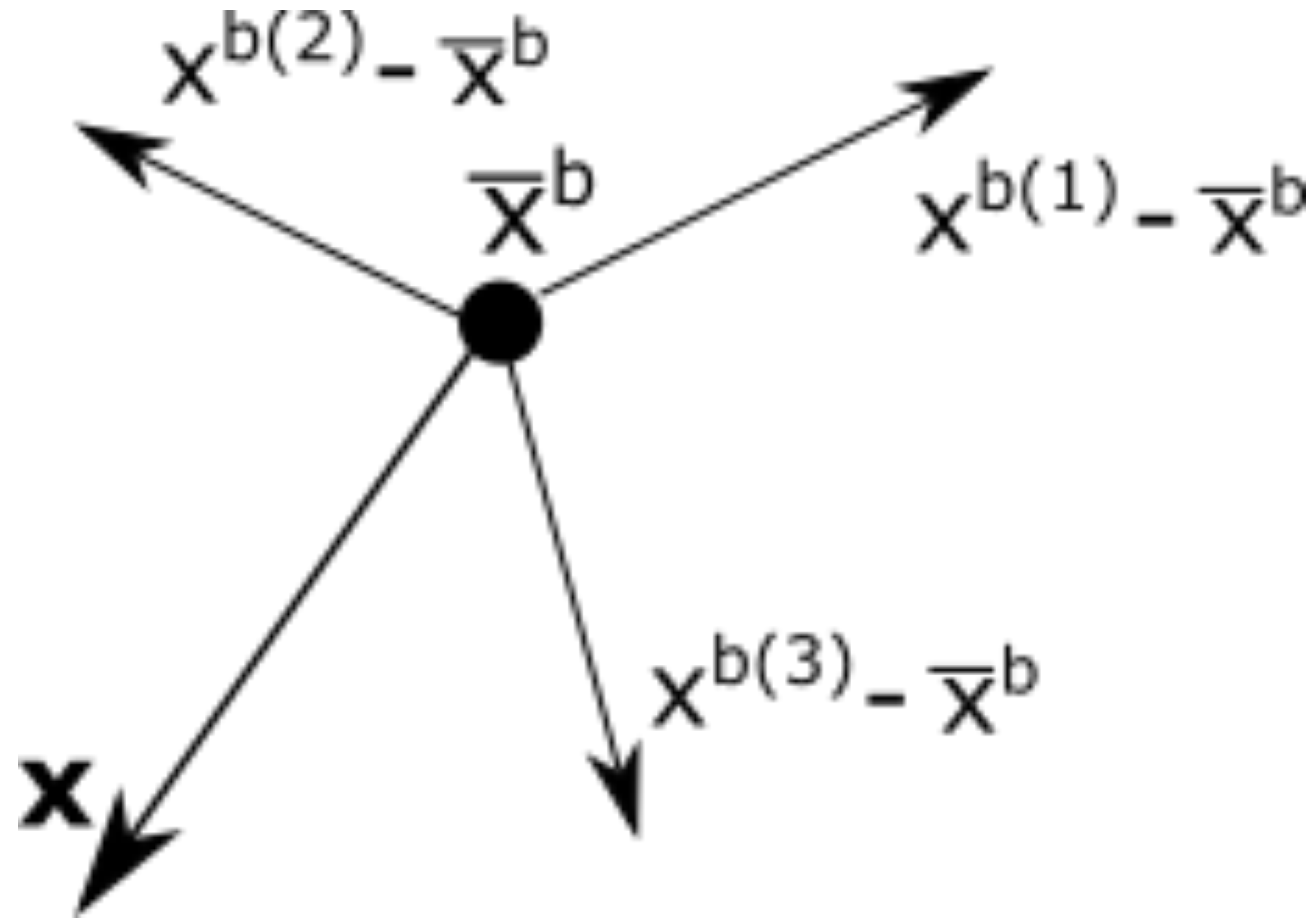
Ensemble mean

$$\bar{x}^b = \frac{1}{L}\sum_{l=1}^{L} x^{b(l)}$$

number of ensemble members: L



description by ensemble



Ensemble mean $$\bar{x}^b = \frac{1}{L}\sum_{l=1}^{L} x^{b(l)}$$

number of ensemble members: L

Ensemble $X = \left(x^{b(1)} - \bar{x}^b \quad , \quad \dots \quad , \quad x^{b(L)} - \bar{x}^b\right)$

Ensemble $$X = \left(x^{b(1)} - \bar{x}^b \quad , \quad \dots \quad , \quad x^{b(L)} - \bar{x}^b \right)$$

Covariance matrix

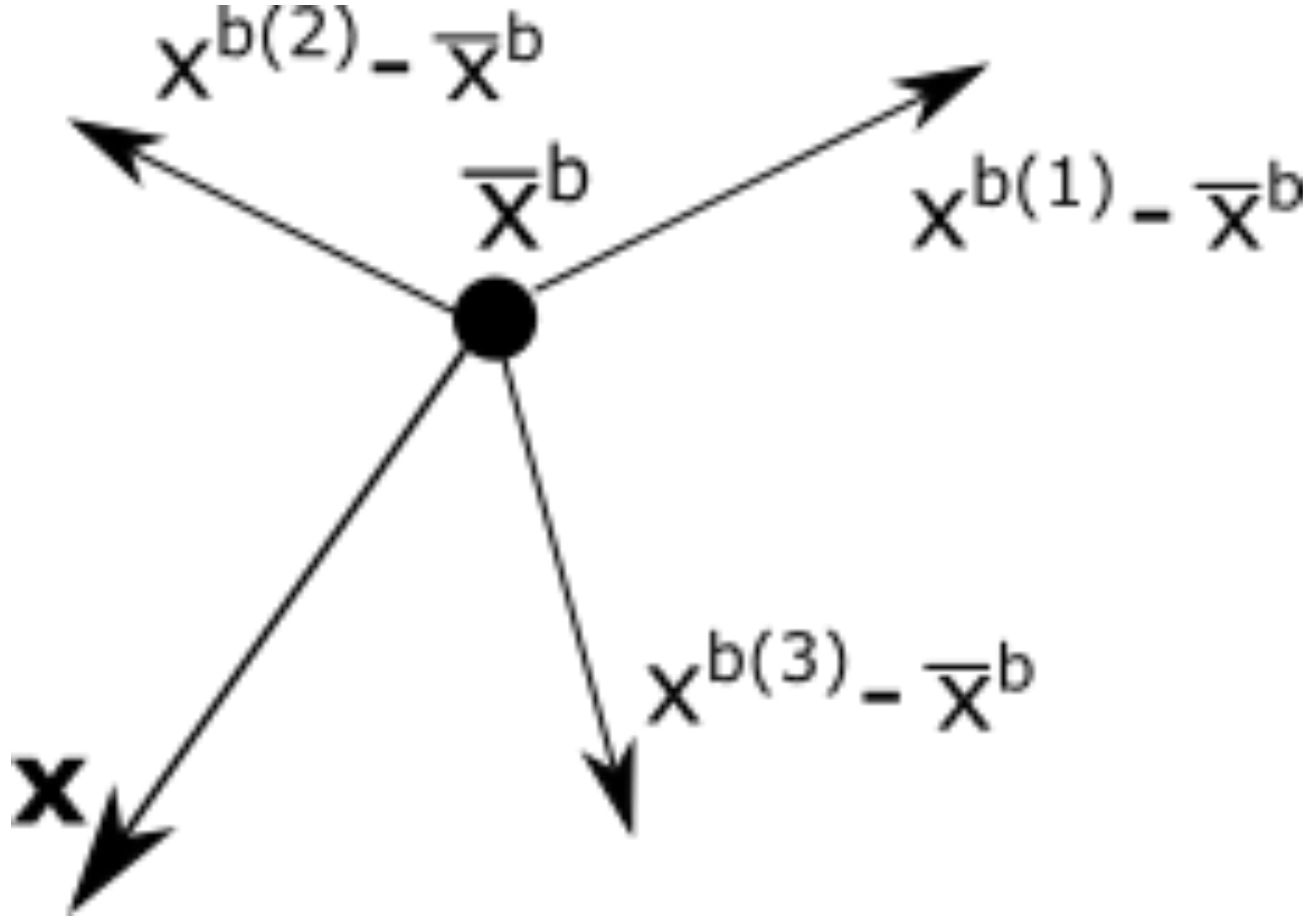
$$\begin{aligned} B &= \frac{1}{L-1} \sum_{l=1}^{L} (x^{b(l)} - \bar{x}^b)(x^{b(l)} - \bar{x}^b)^t \\ &= \frac{1}{L-1} X X^t \end{aligned}$$

Ensemble $X = \left(x^{b(1)} - \bar{x}^b \quad , \quad \ldots \quad , \quad x^{b(L)} - \bar{x}^b\right)$

Covariance matrix

$$B = \frac{1}{L-1}\sum_{l=1}^{L}(x^{b(l)} - \bar{x}^b)(x^{b(l)} - \bar{x}^b)^t$$

$$= \frac{1}{L-1}XX^t$$



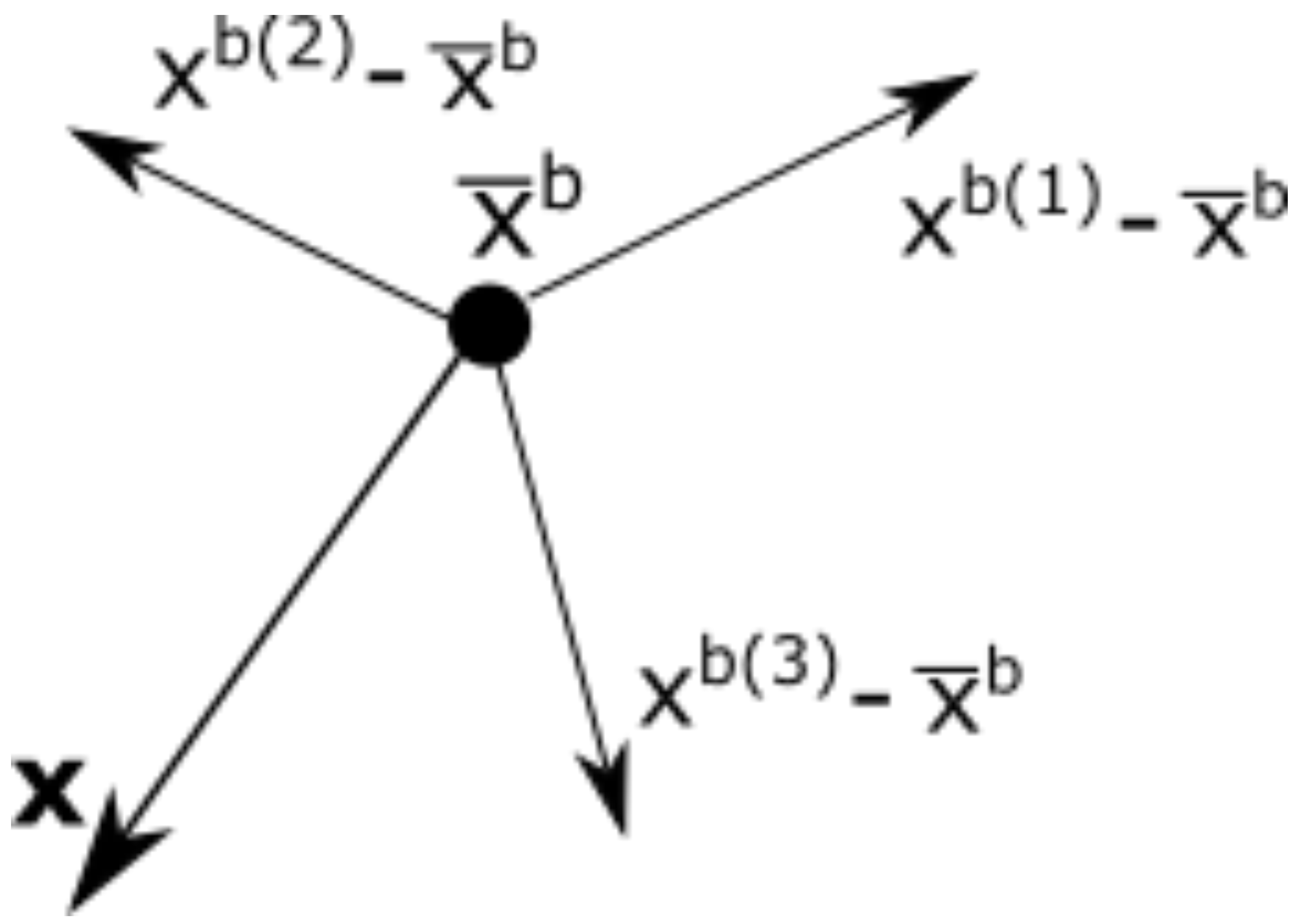
$$x = \bar{x}^b + Xw$$

Ensemble $X = \left(x^{b(1)} - \bar{x}^b \quad , \quad \ldots \quad , \quad x^{b(L)} - \bar{x}^b\right)$

Covariance matrix

$$B = \frac{1}{L-1}\sum_{l=1}^{L}(x^{b(l)} - \bar{x}^b)(x^{b(l)} - \bar{x}^b)^t$$

$$= \frac{1}{L-1} X X^t$$



$$x = \bar{x}^b + Xw$$

Transformation from location space to ensemble space

$$x \in \mathbb{R}^N \qquad w \in \mathbb{R}^L$$

new coordinates in ensemble space

Assumption: probability density of state is Gaussian

$$P(x) \sim e^{-(x-\bar{x})^t \Sigma^{-1} (x-\bar{x})} = e^{-V(x)}$$

Assumption: probability density of state is Gaussian

$$P(x) \sim e^{-(x-\bar{x})^t \Sigma^{-1} (x-\bar{x})} = e^{-V(x)}$$

Given: ensemble of states with ensemble mean $\bar{x}^b$

Given: observations y

Given: observation operator $y^b = \mathcal{H}(x^b)$

Assumption: probability density of state is Gaussian

$$P(x) \sim e^{-(x-\bar{x})^t \Sigma^{-1} (x-\bar{x})} = e^{-V(x)}$$

Given: ensemble of states with ensemble mean $\bar{x}^b$

Given: observations y

Given: observation operator $y^b = \mathcal{H}(x^b)$

error : $$V(x) = \underbrace{(x - \bar{x}^b)^t B^{-1} (x - \bar{x}^b)}_{\text{analysis close to first guess}} + \underbrace{(y - \mathcal{H}(x))^t R^{-1} (y - \mathcal{H}(x))}_{\text{analysis close to observation}}$$

Assumption: probability density of state is Gaussian

$$P(x) \sim e^{-(x-\bar{x})^t \Sigma^{-1} (x-\bar{x})} = e^{-V(x)}$$

Given: ensemble of states with ensemble mean $\bar{x}^b$

Given: observations y

Given: observation operator $y^b = \mathcal{H}(x^b)$

error : $$V(x) = (x - \bar{x}^b)^t B^{-1} (x - \bar{x}^b) + (y - \mathcal{H}(x))^t R^{-1} (y - \mathcal{H}(x))$$

analysis close to first guess analysis close to observation

inserted $x = \bar{x}^b + Xw$ (transformation to ensemble space):

Assumption: probability density of state is Gaussian

$$P(x) \sim e^{-(x-\bar{x})^t \Sigma^{-1} (x-\bar{x})} = e^{-V(x)}$$

Given: ensemble of states with ensemble mean $\bar{x}^b$

Given: observations y

Given: observation operator $y^b = \mathcal{H}(x^b)$

error : $$V(x) = \underbrace{(x - \bar{x}^b)^t B^{-1} (x - \bar{x}^b)}_{\text{analysis close to first guess}} + \underbrace{(y - \mathcal{H}(x))^t R^{-1} (y - \mathcal{H}(x))}_{\text{analysis close to observation}}$$

inserted $x = \bar{x}^b + Xw$ (transformation to ensemble space):

$$V(w) = ww^t (L-1) + (y - \mathcal{H}(\bar{x}^b + Xw))^t R^{-1} (y - \mathcal{H}(\bar{x}^b + Xw))$$

$$\mathcal{H}(\bar{x}^b + Xw) \approx \mathcal{H}\bar{x}^b + Yw$$

observation operator is quasi-linear

$$Y = \left(\mathcal{H}x^{b(1)} - \mathcal{H}\bar{x}^b \quad , \quad \ldots \quad , \quad \mathcal{H}x^{b(L)} - \mathcal{H}\bar{x}^b\right)$$

$$\mathcal{H}(\bar{x}^b + Xw) \approx \mathcal{H}\bar{x}^b + Yw$$

observation operator is quasi-linear

$$Y = \left(\mathcal{H}x^{b(1)} - \mathcal{H}\bar{x}^b \quad , \quad \ldots \quad , \quad \mathcal{H}x^{b(L)} - \mathcal{H}\bar{x}^b\right)$$

$$\Rightarrow \quad \boxed{V(w) = ww^t(L-1) + (y - \mathcal{H}\bar{x}^b - Yw)^t R^{-1}(y - \mathcal{H}\bar{x}^b - Yw)}$$

$$\mathcal{H}(\bar{x}^b + Xw) \approx \mathcal{H}\bar{x}^b + Yw$$

observation operator is quasi-linear

$$Y = \left(\mathcal{H}x^{b(1)} - \mathcal{H}\bar{x}^b \quad , \quad \dots \quad , \quad \mathcal{H}x^{b(L)} - \mathcal{H}\bar{x}^b\right)$$

$$\Rightarrow \quad \boxed{V(w) = ww^t(L-1) + (y - \mathcal{H}\bar{x}^b - Yw)^t R^{-1}(y - \mathcal{H}\bar{x}^b - Yw)}$$

Since $P \sim e^{-V}$: most probable ensemble can be found by $\dfrac{\partial V}{\partial w} = 0$

$$\mathcal{H}(\bar{x}^b + Xw) \approx \mathcal{H}\bar{x}^b + Yw$$

observation operator is quasi-linear

$$Y = \left(\mathcal{H}x^{b(1)} - \mathcal{H}\bar{x}^b \quad , \quad \dots \quad , \quad \mathcal{H}x^{b(L)} - \mathcal{H}\bar{x}^b\right)$$

$\longrightarrow$

$$V(w) = ww^t(L-1) + (y - \mathcal{H}\bar{x}^b - Yw)^t R^{-1}(y - \mathcal{H}\bar{x}^b - Yw)$$

Since $P \sim e^{-V}$: most probable ensemble can be found by $\dfrac{\partial V}{\partial w} = 0$

$$\boxed{\begin{array}{ll} \tilde{w}_a = \tilde{\mathbf{P}}_a \mathbf{Y}^t \mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}\bar{\mathbf{x}}_\mathbf{b}) & \bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{X}\tilde{w}_a \\ \tilde{\mathbf{P}}_a = \left((L-1)\mathbf{I} + \mathbf{Y}^t\mathbf{R}^{-1}\mathbf{Y}\right)^{-1} & x^{a(l)} = \bar{x}^a + Xw^{a(l)} \\ & \text{analysis ensemble member} \end{array}}$$

$$Z = (L-1)\sqrt{\tilde{P}^a} \qquad Z = \left(w^{a(1)} - \bar{w}^a \quad , \quad \dots \quad , \quad w^{a(L)} - \bar{w}^a\right)$$

all together: state description in ensemble space by L ensemble members, i.e. $\{\mathbf{x}_b^{(l)}\}$

prediction:

$$\mathbf{x}_b^{(l)}(t_n) = \mathcal{M}\mathbf{x}_a^{(l)}(t_{n-1})$$

$$\bar{\mathbf{x}}_b = \frac{1}{L}\sum_{l=1}^{L}\mathbf{x}_b^{(l)}$$

$$\mathbf{X} = \begin{pmatrix} (\mathbf{x}_b^{(1)} - \bar{\mathbf{x}}_b) & (\mathbf{x}_b^{(2)} - \bar{\mathbf{x}}_b) & \cdots & (\mathbf{x}_b^{(L)} - \bar{\mathbf{x}}_b) \end{pmatrix}$$

$$\mathbf{P}_b = \frac{1}{L-1}\mathbf{X}\mathbf{X}^t$$

$$\bar{\mathbf{y}}_b = \frac{1}{L}\sum_{l=1}^{L}\mathcal{H}\mathbf{x}_b^{(l)}$$

$$\mathbf{Y} = \begin{pmatrix} (\mathcal{H}\mathbf{x}_b^{(1)} - \bar{\mathbf{y}}_b) & (\mathcal{H}\mathbf{x}_b^{(2)} - \bar{\mathbf{y}}_b) & \cdots & (\mathcal{H}\mathbf{x}_b^{(L)} - \bar{\mathbf{y}}_b) \end{pmatrix}$$

analysis update:

$$\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{X}\tilde{w}_a \qquad \tilde{w}_a = \tilde{\mathbf{P}}_a \mathbf{Y}^t \mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}\bar{\mathbf{x}}_{\mathbf{b}})$$

$$\tilde{\mathbf{P}}_a = \left((L-1)\mathbf{I} + \mathbf{Y}^t\mathbf{R}^{-1}\mathbf{Y}\right)^{-1}$$

$$\mathbf{x}^{a(l)} = \bar{\mathbf{x}}_a + \mathbf{X}\mathbf{w}^{a(l)} \qquad \begin{aligned}\mathbf{Z} &= (L-1)\sqrt{\tilde{\mathbf{P}}_a} \\ &= \left((\mathbf{w}^{a(1)} - \bar{\mathbf{w}}_a) \quad (\mathbf{w}^{a(2)} - \bar{\mathbf{w}}_a) \quad \cdots \quad (\mathbf{w}^{a(L)} - \bar{\mathbf{w}}_a)\right)\end{aligned}$$

important to note: assimilation is performed in ensemble space

Ensemble Transform Kalman Filter (UKF)

advantage:

- non-linear model and observation operator
- low number of ensemble members **permits dimensionality reduction**

disadvantage:

- more complex to implement numerically
- still assumes Gaussian distribution

problem : instability by filter divergence

first guess covariance $$B = \frac{1}{L-1} X \ I \ X^t$$

analysis covariance $$\Sigma^a = \frac{1}{L-1} X[I \ + Y^t R^{-1} Y/(L-1) \ ]^{-1} X^t$$

problem : instability by filter divergence

first guess covariance $$B = \frac{1}{L-1} X \; I \; X^t$$

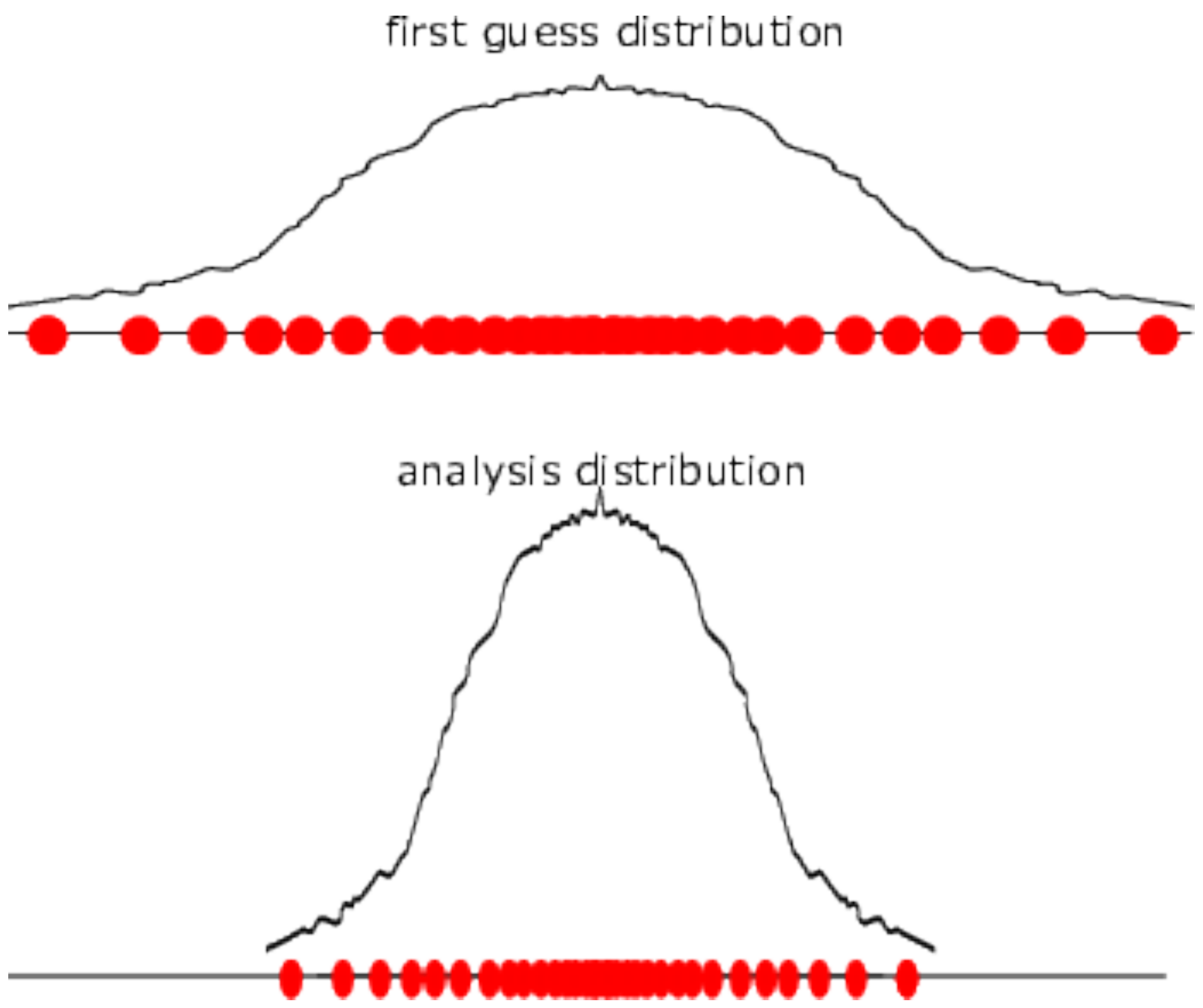
analysis covariance $$\Sigma^a = \frac{1}{L-1} X [I \boxed{+ Y^t R^{-1} Y/(L-1)}]^{-1} X^t$$

problem : instability by filter divergence

first guess covariance $$B = \frac{1}{L-1} X \ I \ X^t$$

analysis covariance $$\Sigma^a = \frac{1}{L-1} X[I \boxed{+Y^t R^{-1} Y/(L-1)}]^{-1} X^t$$

$$|\Sigma^a| < |B|$$



the filter variance shrinks in each update

the filter variance shrinks in each update

—> the ensemble spread decreases, i.e. $\mathbf{X}$ shrinks

the filter variance shrinks in each update

—> the ensemble spread decreases, i.e. $\mathbf{X}$ shrinks

—> the Kalman gain K decreases

$$x^a = x^b + K(y - Hx^b)$$

the filter variance shrinks in each update

—> the ensemble spread decreases, i.e. $\mathbf{X}$ shrinks

—> the Kalman gain *K* decreases $$x^a = x^b + K(y - Hx^b)$$

—> impact of innovation decreases and filter diverges from observations

the filter variance shrinks in each update

—> the ensemble spread decreases, i.e. $\mathbf{X}$ shrinks

—> the Kalman gain *K* decreases $$x^a = x^b + K(y - Hx^b)$$

—> impact of innovation decreases and filter diverges from observations

solution: additional inflation of ensemble typical modification:

$$\mathbf{P}_b = \frac{1}{L-1}\mathbf{X}\mathbf{X}^t \quad \Longrightarrow \quad \mathbf{P}_b = \frac{1}{L-1}\mathbf{X}\mathbf{X}^t + \gamma_a \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1N} \\ \cdot & \cdot & \cdots & \cdot \\ \rho_{N1} & \rho_{N2} & \cdots & \rho_{NN} \end{pmatrix}$$

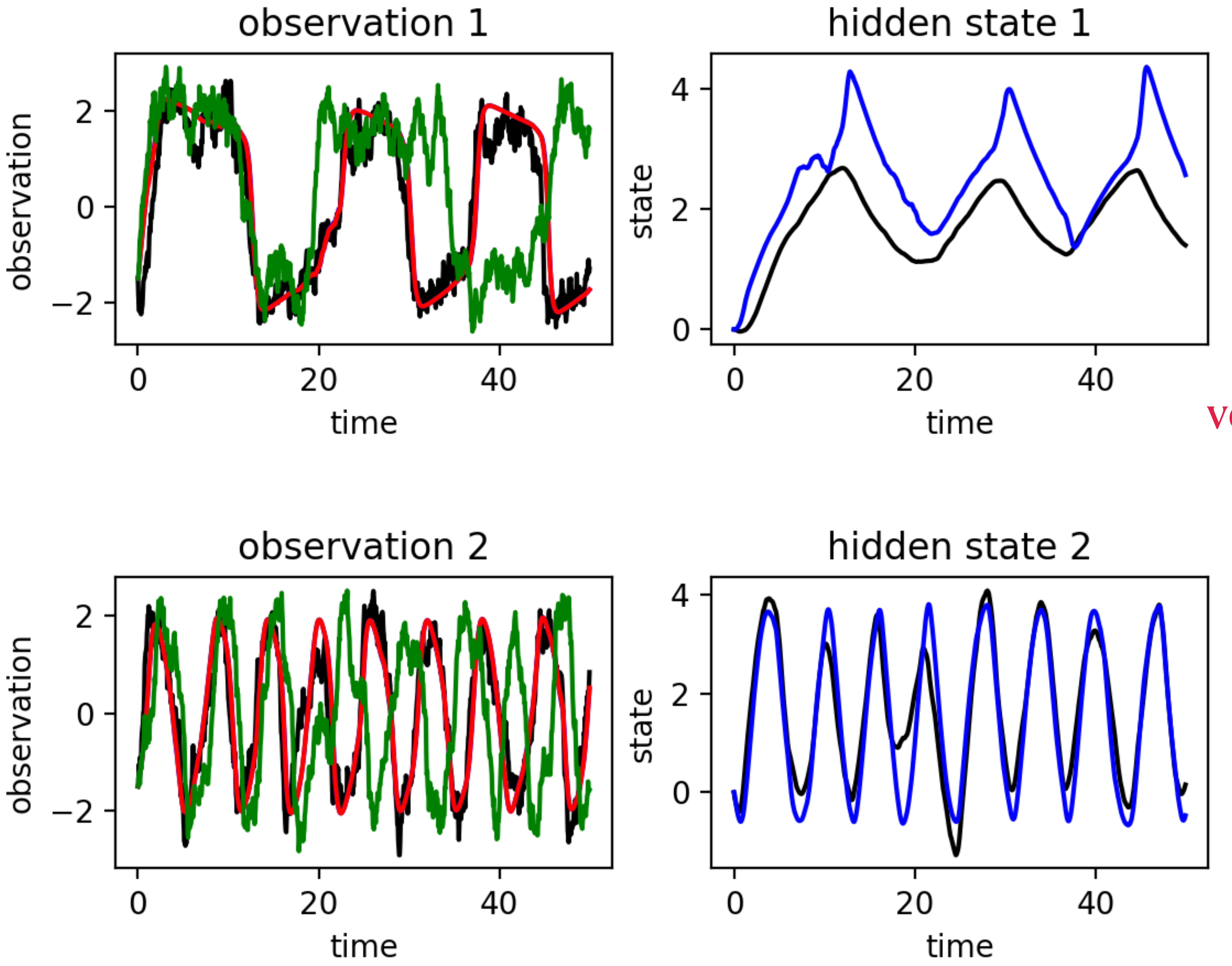
γ_a : additive covariance inflation

$\rho_{kl} \sim \mathcal{N}(0,1)$: Gaussian noise

$$\mathbf{x}^{a(l)} = \bar{\mathbf{x}}_a + \mathbf{X}\mathbf{w}^{a(l)} \quad \Longrightarrow \quad \mathbf{x}^{a(l)} = \bar{\mathbf{x}}_a + \gamma_m \mathbf{X}\mathbf{w}^{a(l)}$$

γ_m : multiplicative ensemble inflation

ETKF_FHN_da.py



very good tracking of observations

good tracking of hidden state

inflation: $\gamma_a = 0$, $\gamma_m = 1.02$

motivation

basic methods

prediction and verification

Kalman filter

linear EKF UKF **ETKF** LETKF

impact of R and inflation

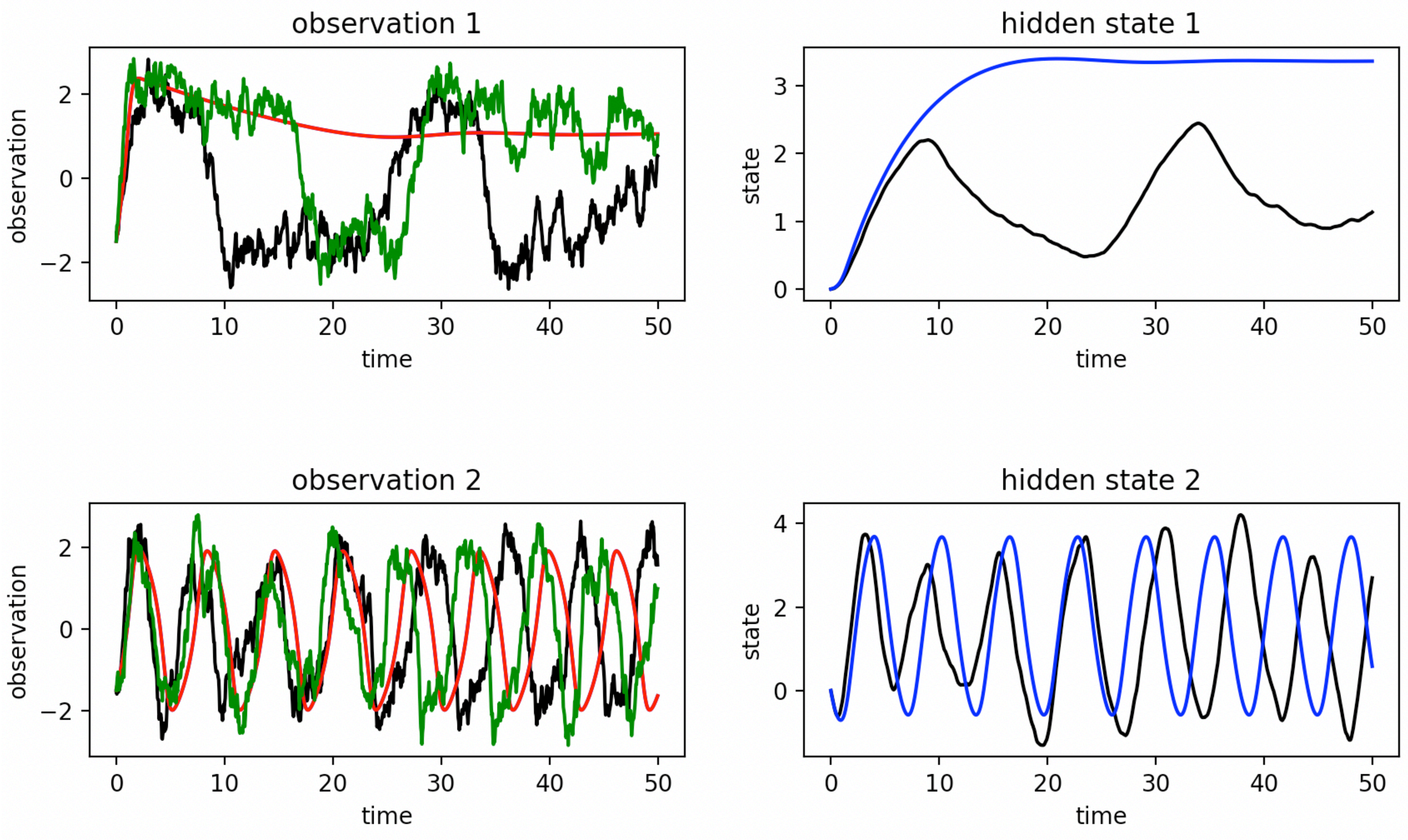
filter of non-stationary dynamics

dimensionality reduction

parameter estimation

impact of observation time

impact of observation error **R** and inflation



no additive inflation $\gamma_a = 0$

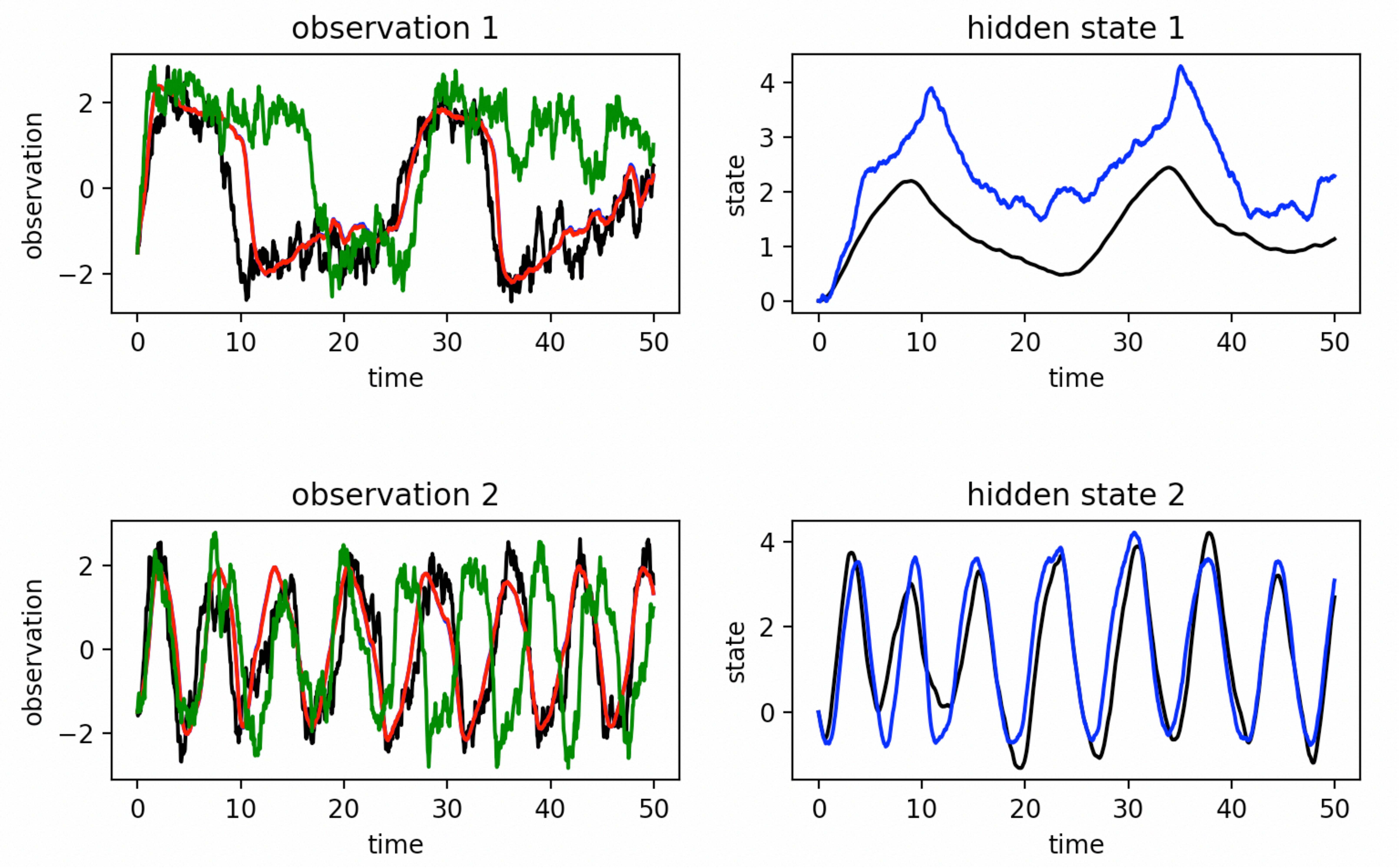
no multiplicative inflation $\gamma_m = 0$

$$\mathbf{R} = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$$

background observations

do not fit observations

ETKF_FHN_2_da.py



additive inflation $\gamma_a = 0.05$

$$\mathbf{R} = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$$

background observations

do fit observations well

motivation

basic methods

Kalman filter

linear EKF UKF **ETKF** LETKF

impact of R and inflation

filter of non-stationary dynamics

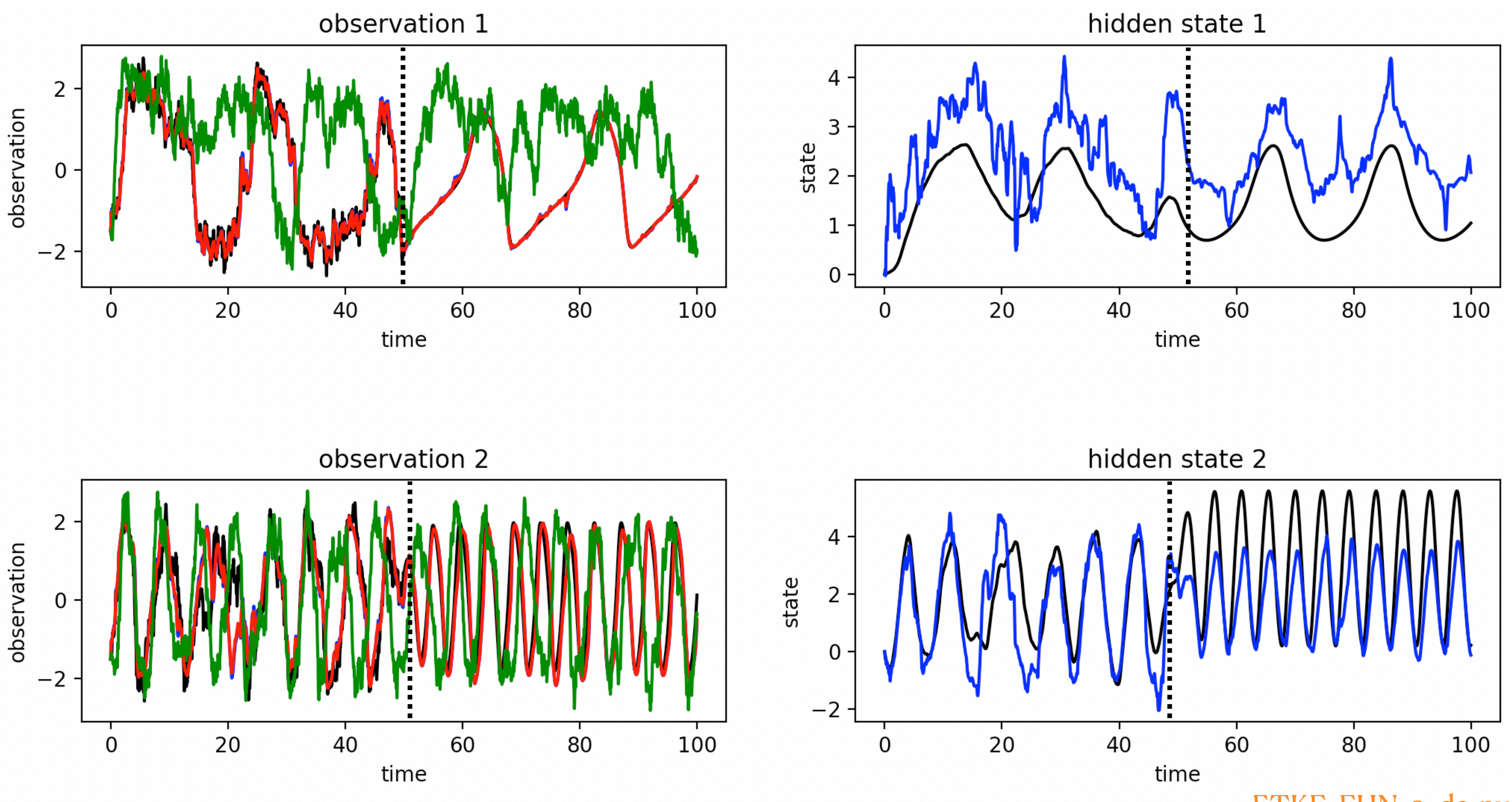
dimensionality reduction

parameter estimation

impact of observation time

prediction and verification

transient model



true dynamics

$I_1 = 1.5 \ , \ \tau_2 = 10$

switch: $\tau_1 = 10 \ \rightarrow \ \tau_1 = 5$

$I_2 = 1.5 \ \rightarrow \ I_2 = 5.0$

$b_{12} = 10 \ \rightarrow \ b_{12} = 15$

false dynamics

$I_1 = 2.7 \ , \ \tau_2 = 9.0$

ETKF_FHN_3_da.py

filter adapts to transient dynamics

motivation

basic methods

Kalman filter

linear EKF UKF **ETKF** LETKF

impact of R and inflation

filter of non-stationary dynamics

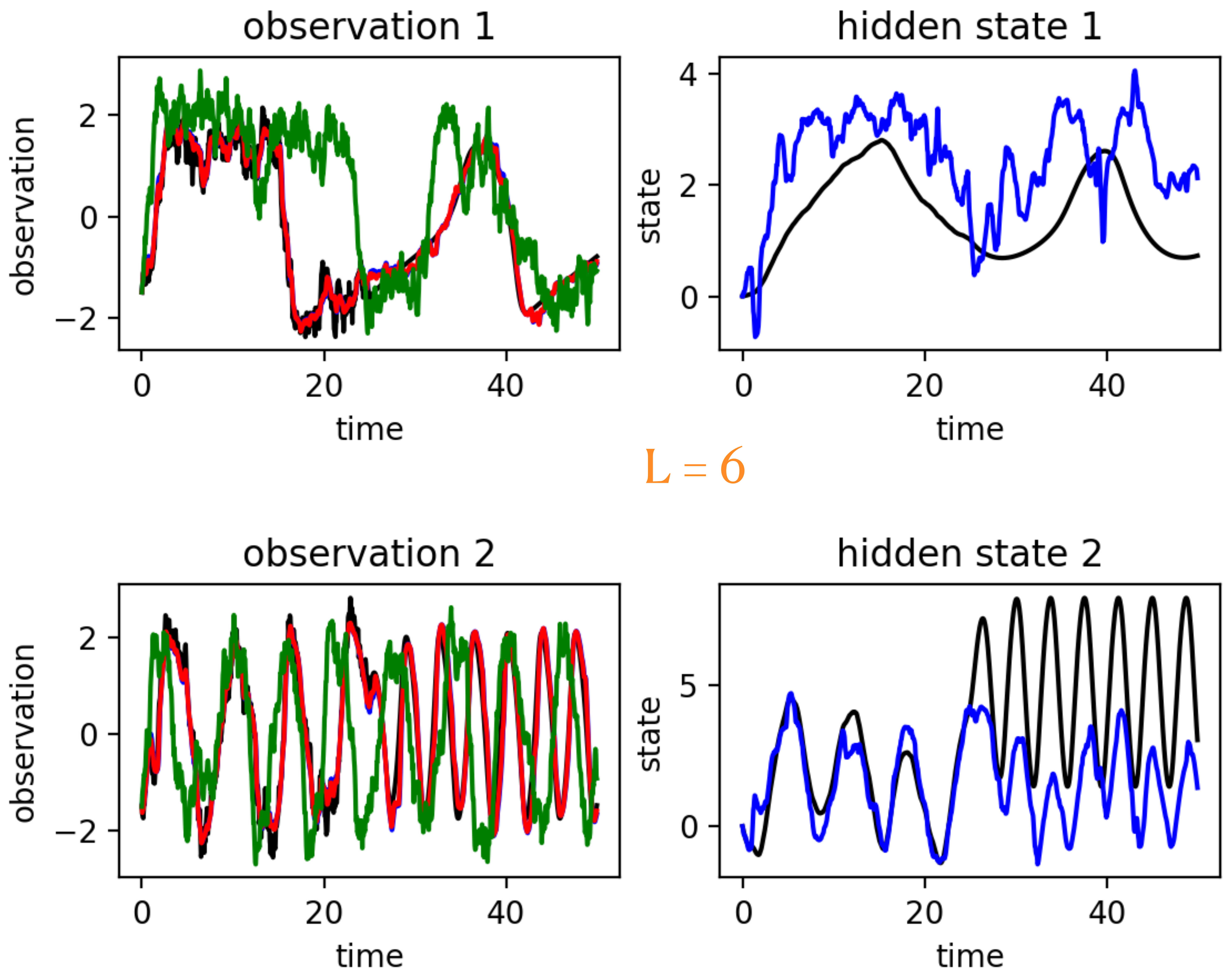
dimensionality reduction

parameter estimation

impact of observation time

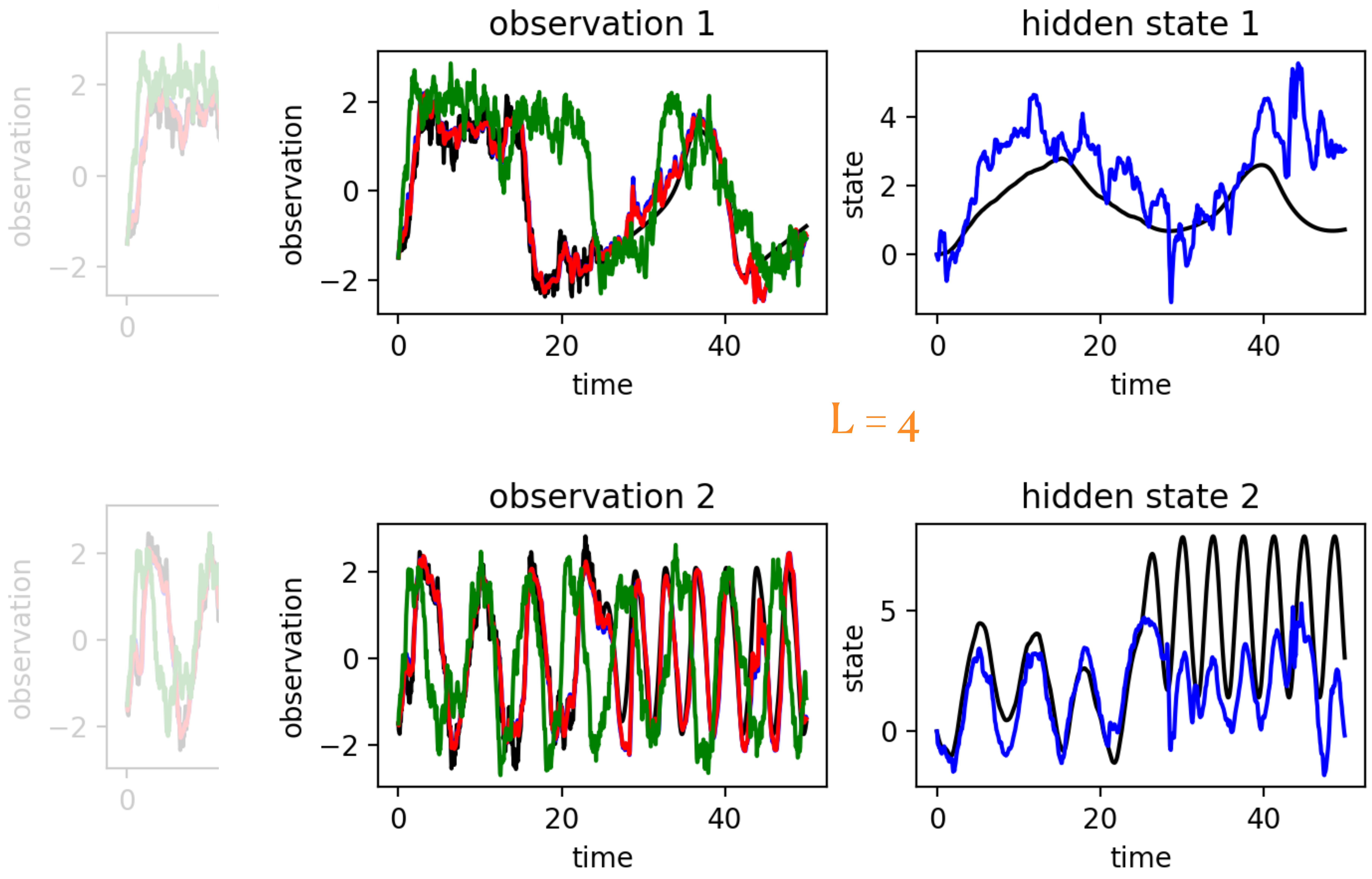
prediction and verification

impact of number of ensemble members

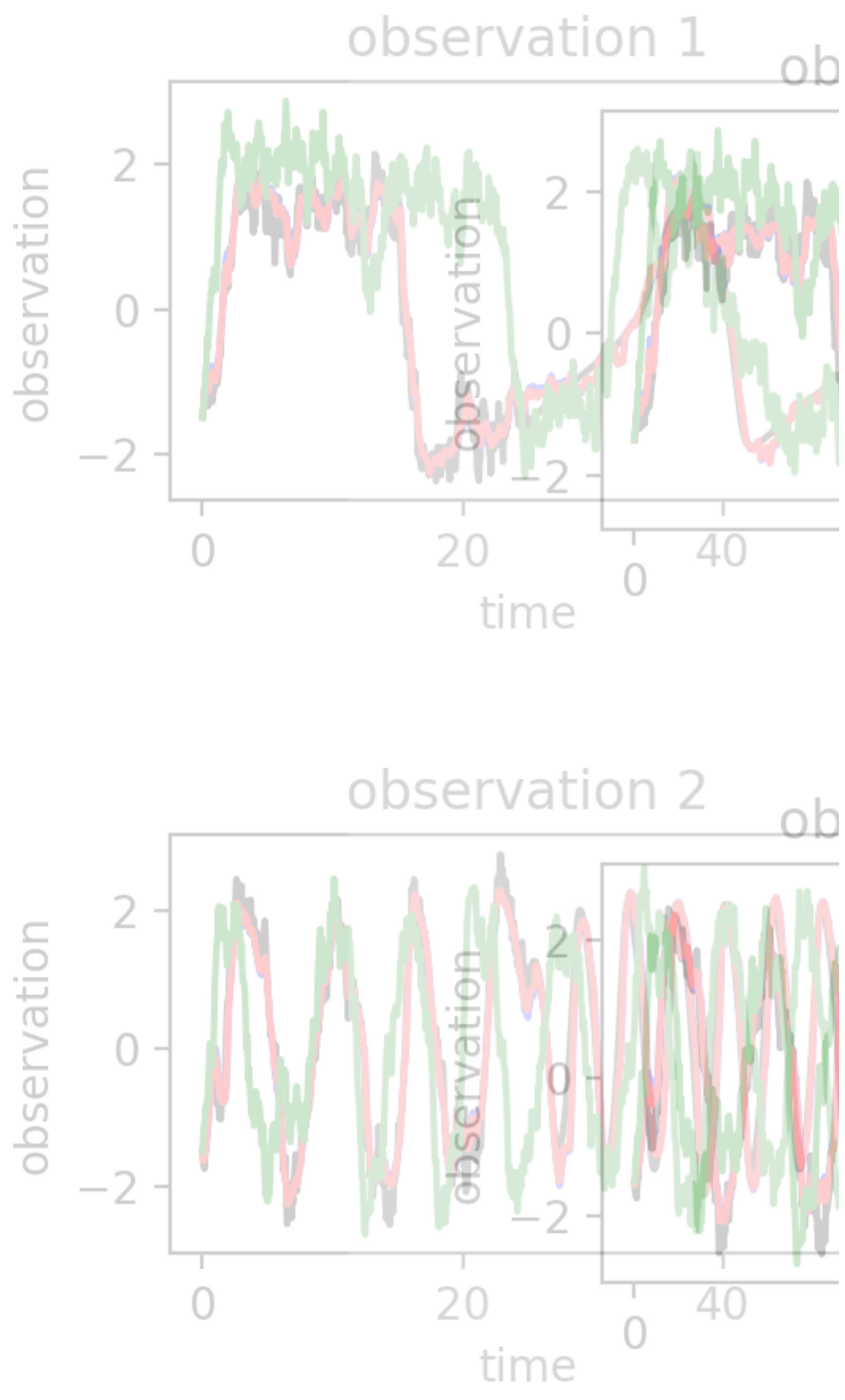


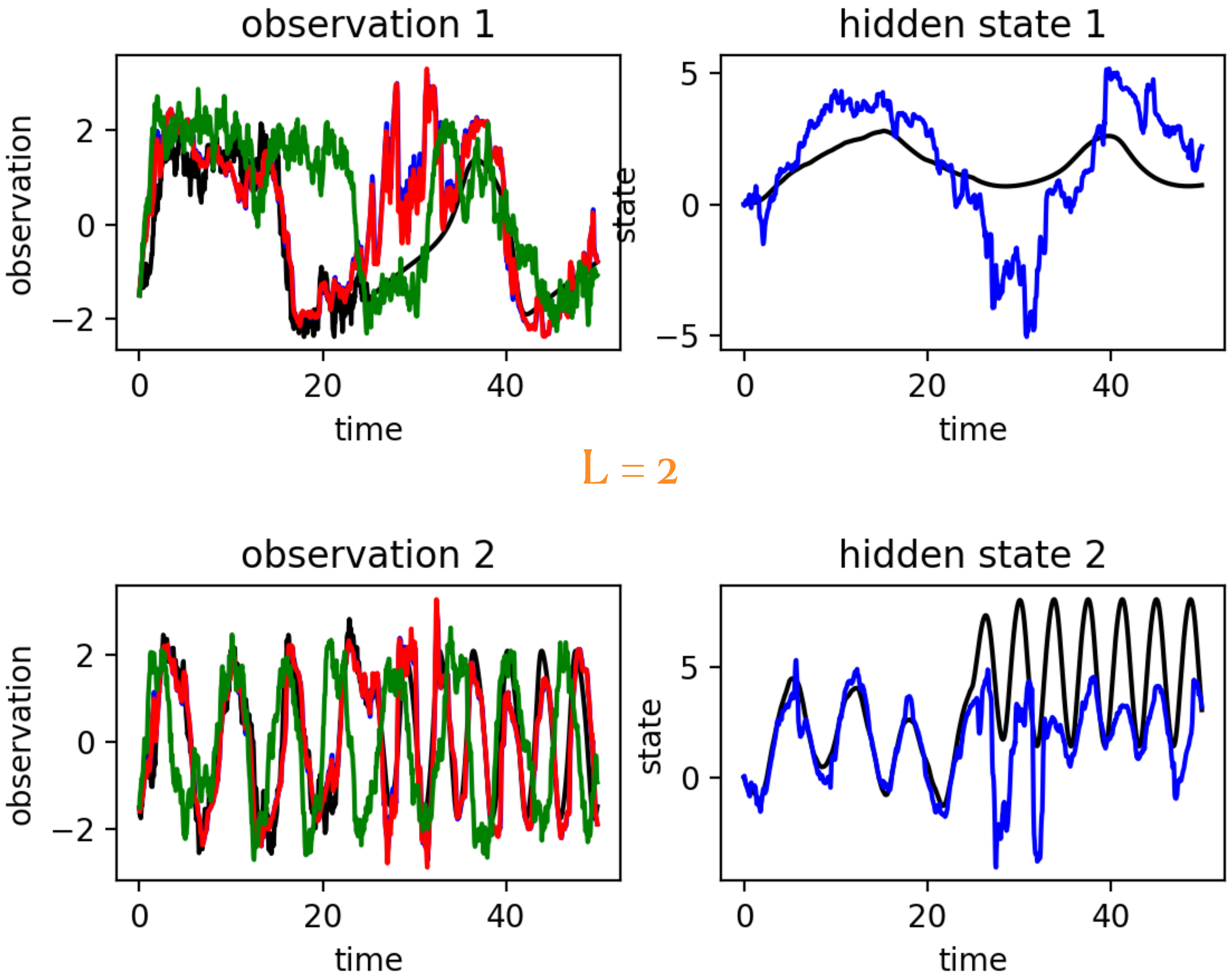
L = 6

impact of number of ensemble members



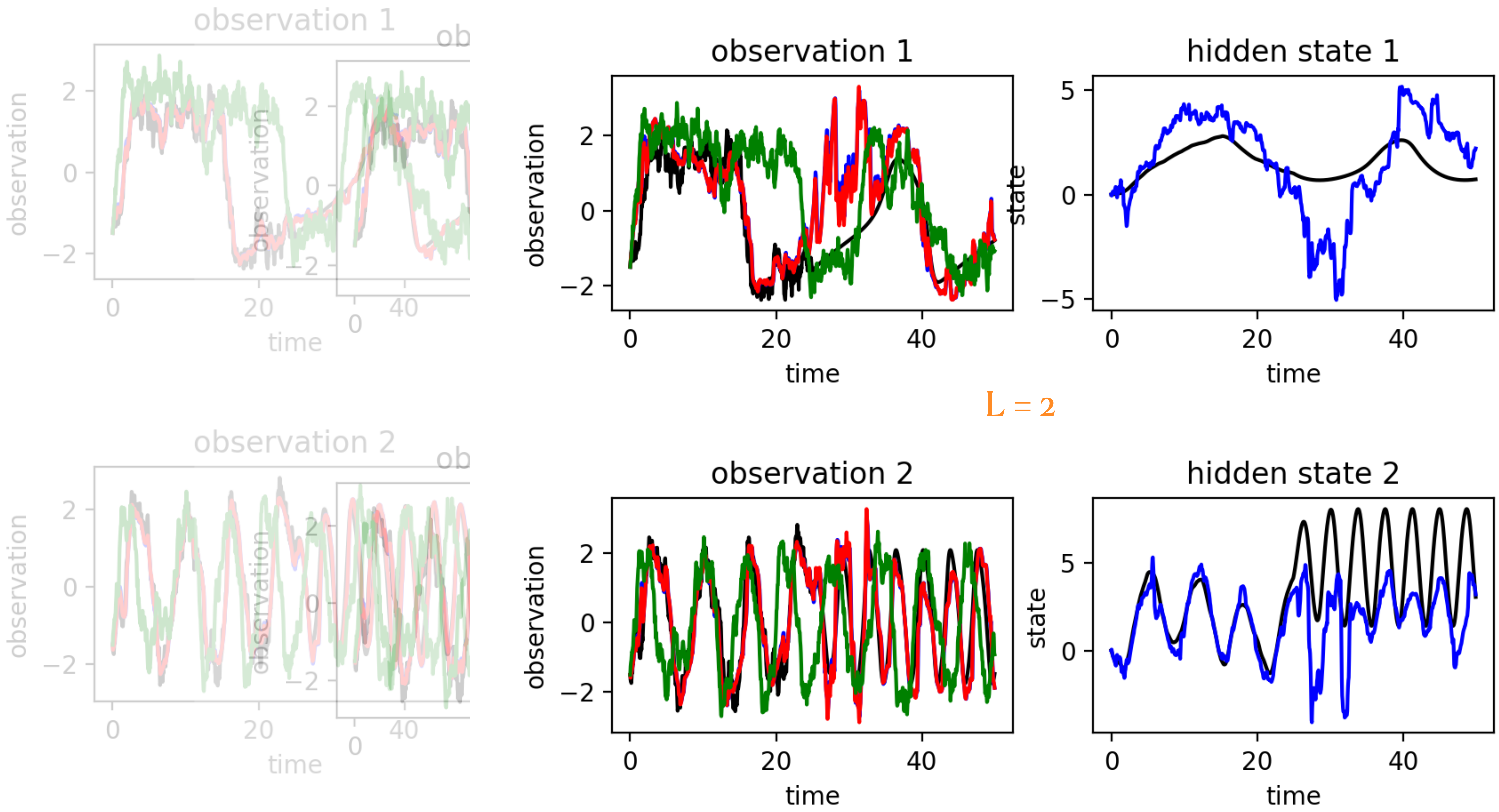
impact of number of ensemble members





L = 2

impact of number of ensemble members



strong dimensionality reduction $L = 6 \;\rightarrow\; L = 2$ possible

motivation

basic methods

prediction and verification

Kalman filter

linear EKF UKF **ETKF** LETKF

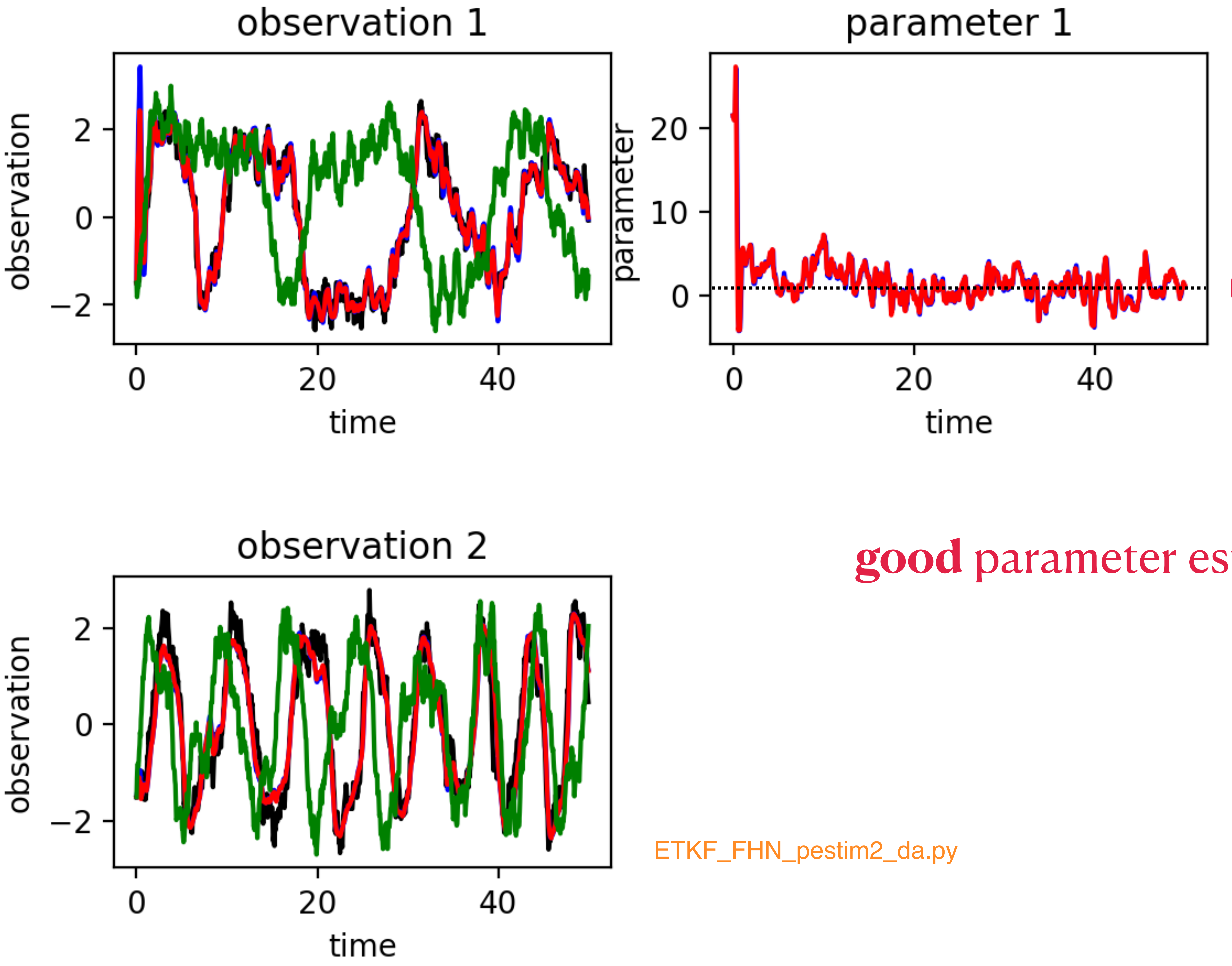
impact of R and inflation

filter of non-stationary dynamics

dimensionality reduction

parameter estimation

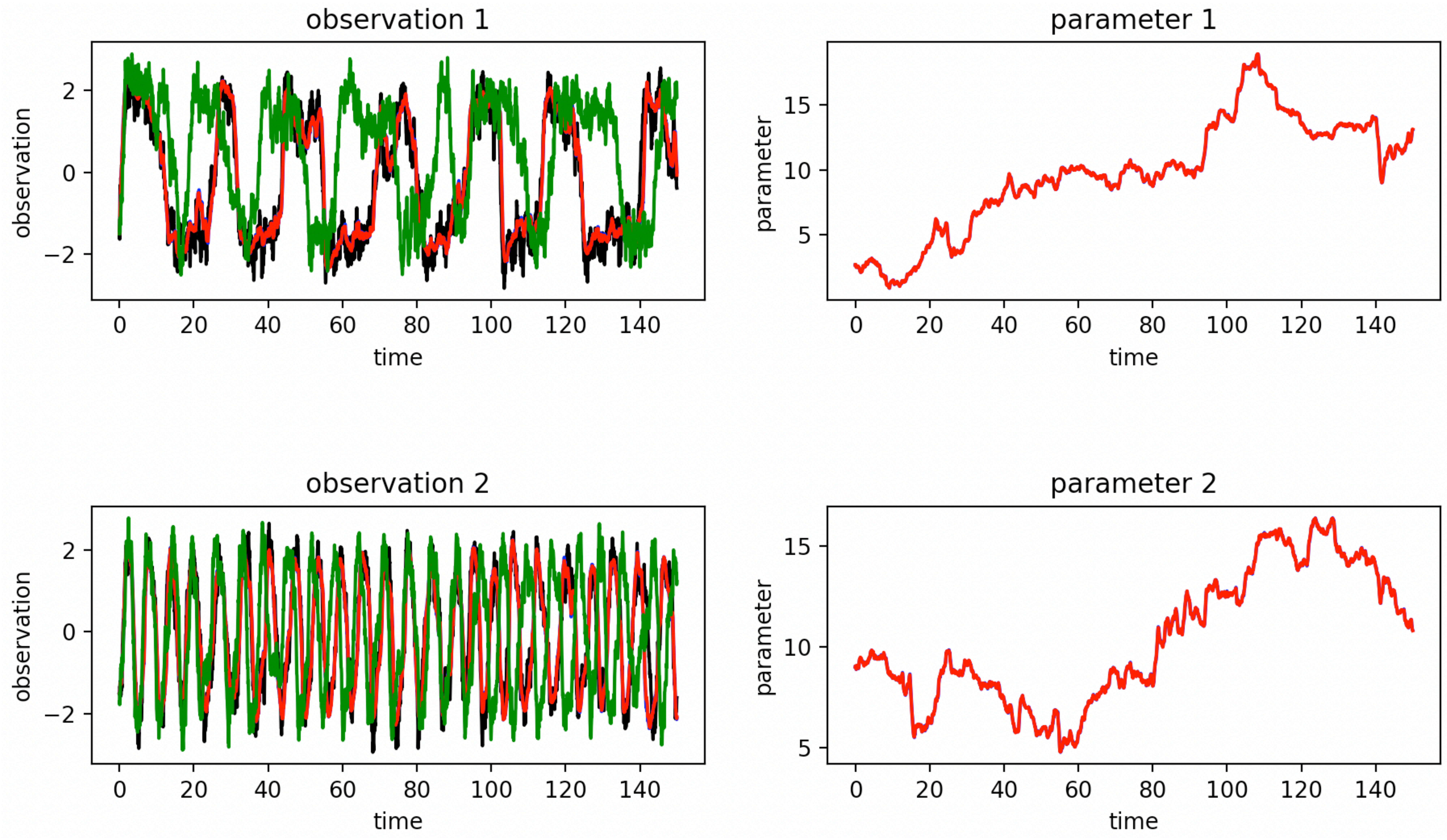
impact of observation time



good parameter estimation of **1 parameter**

ETKF_FHN_pestim2_da.py

On parameter estimation



bad estimation

of **2 parameters**

ETKF_FHN_pestim_da.py

motivation

basic methods

Kalman filter

linear EKF UKF **ETKF** LETKF

impact of R and inflation

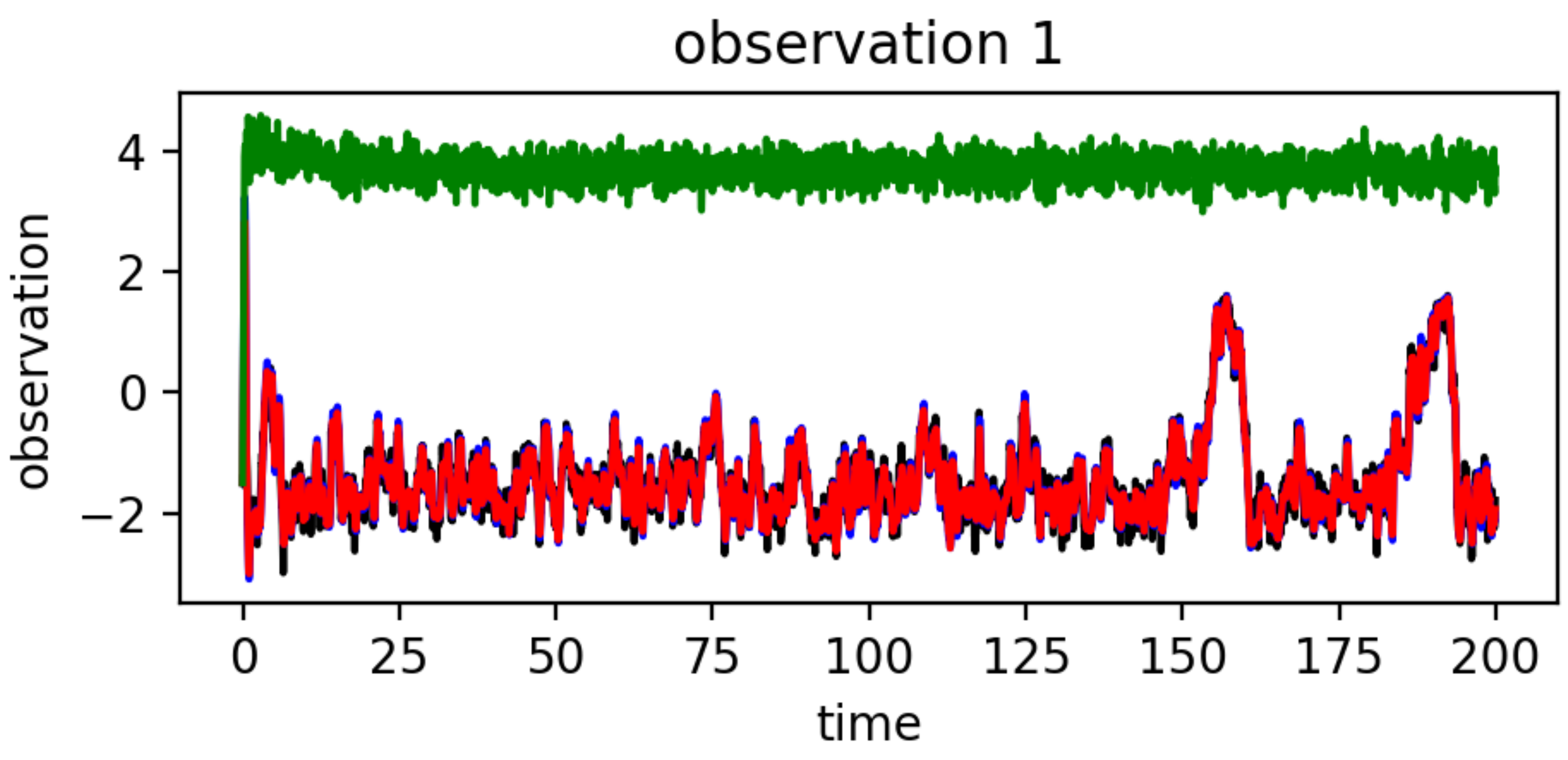
filter of non-stationary dynamics

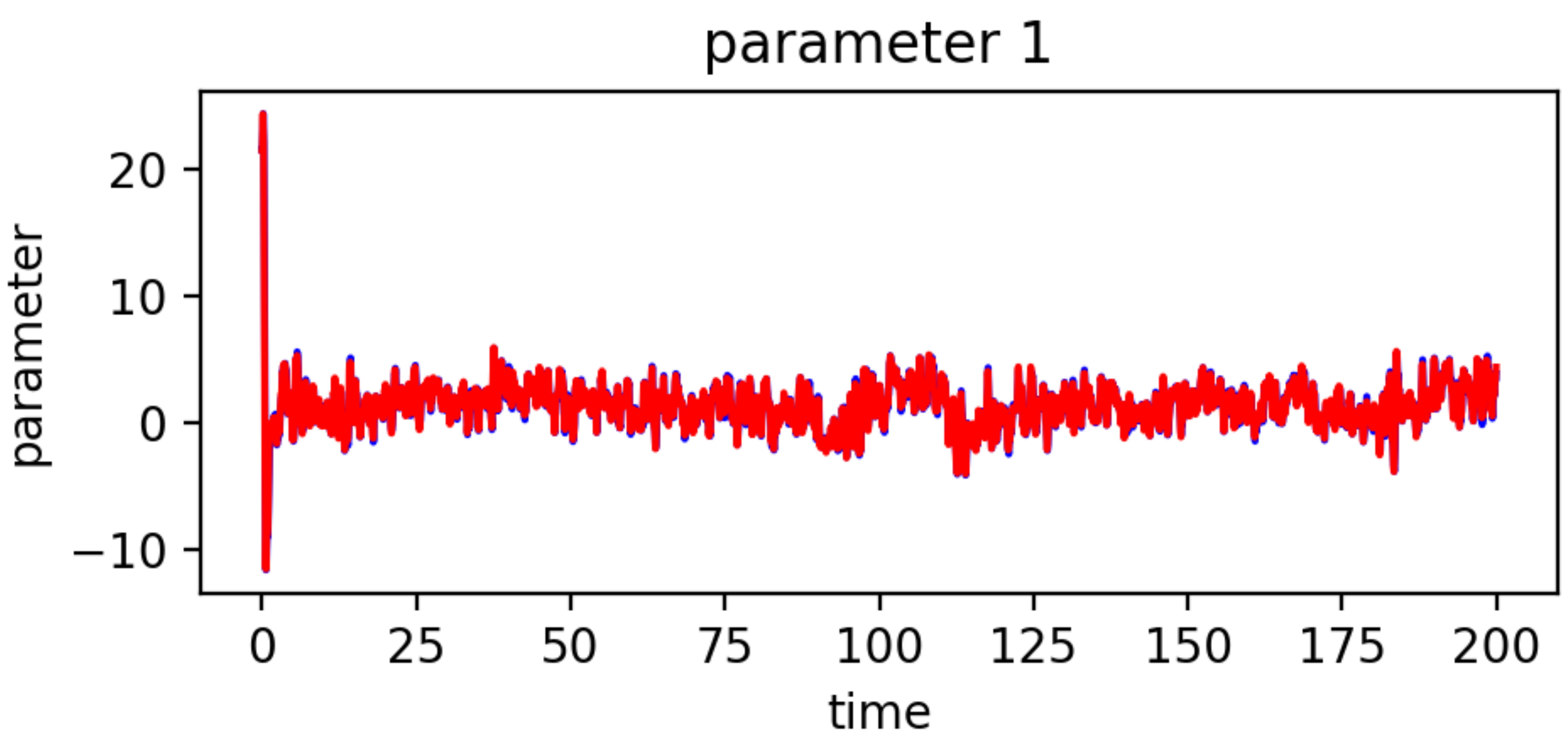
dimensionality reduction

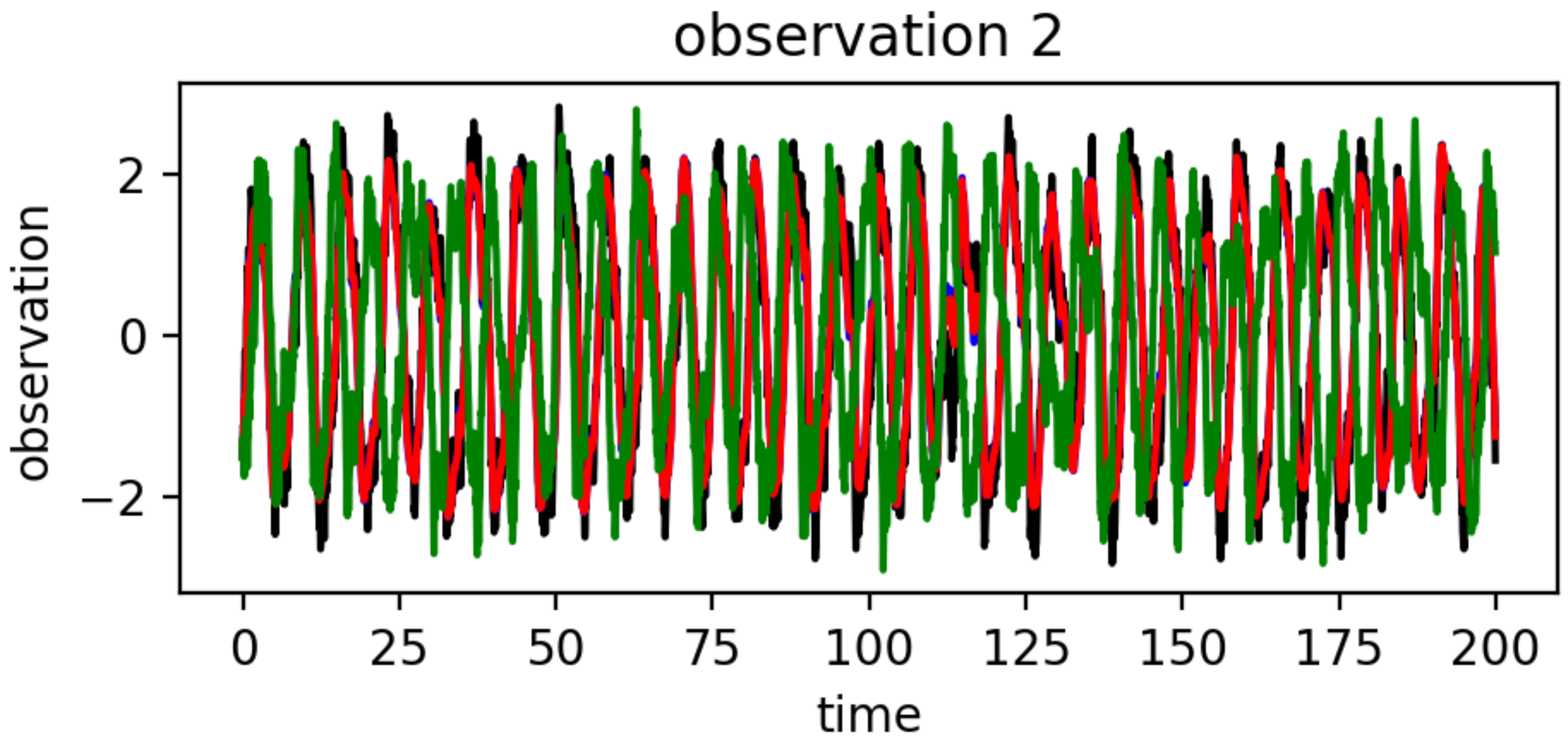
parameter estimation

impact of observation time

Impact of observation sampling time



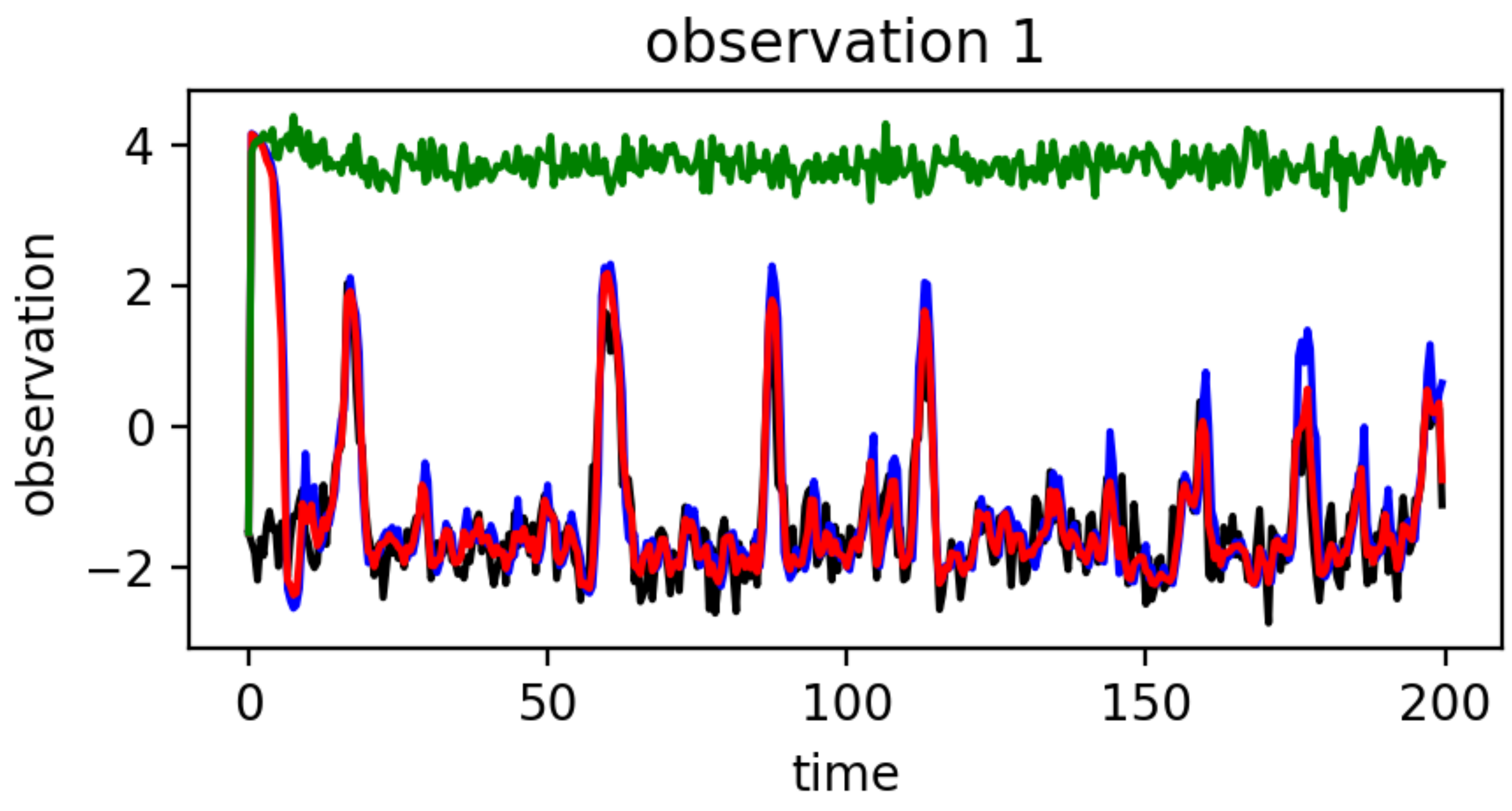


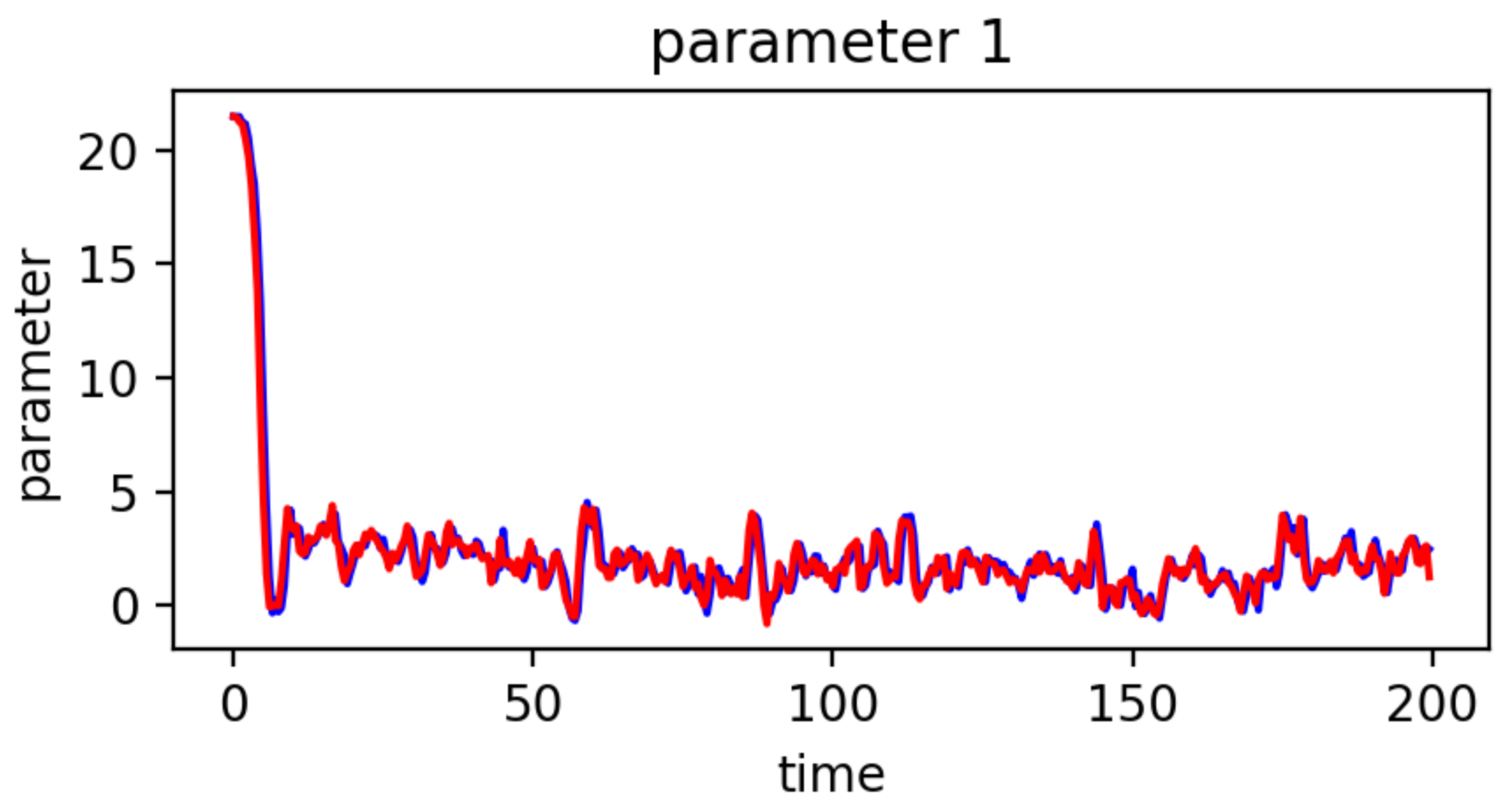


model iteration time: $\Delta t = 0.05$

observation sampling time: Δt

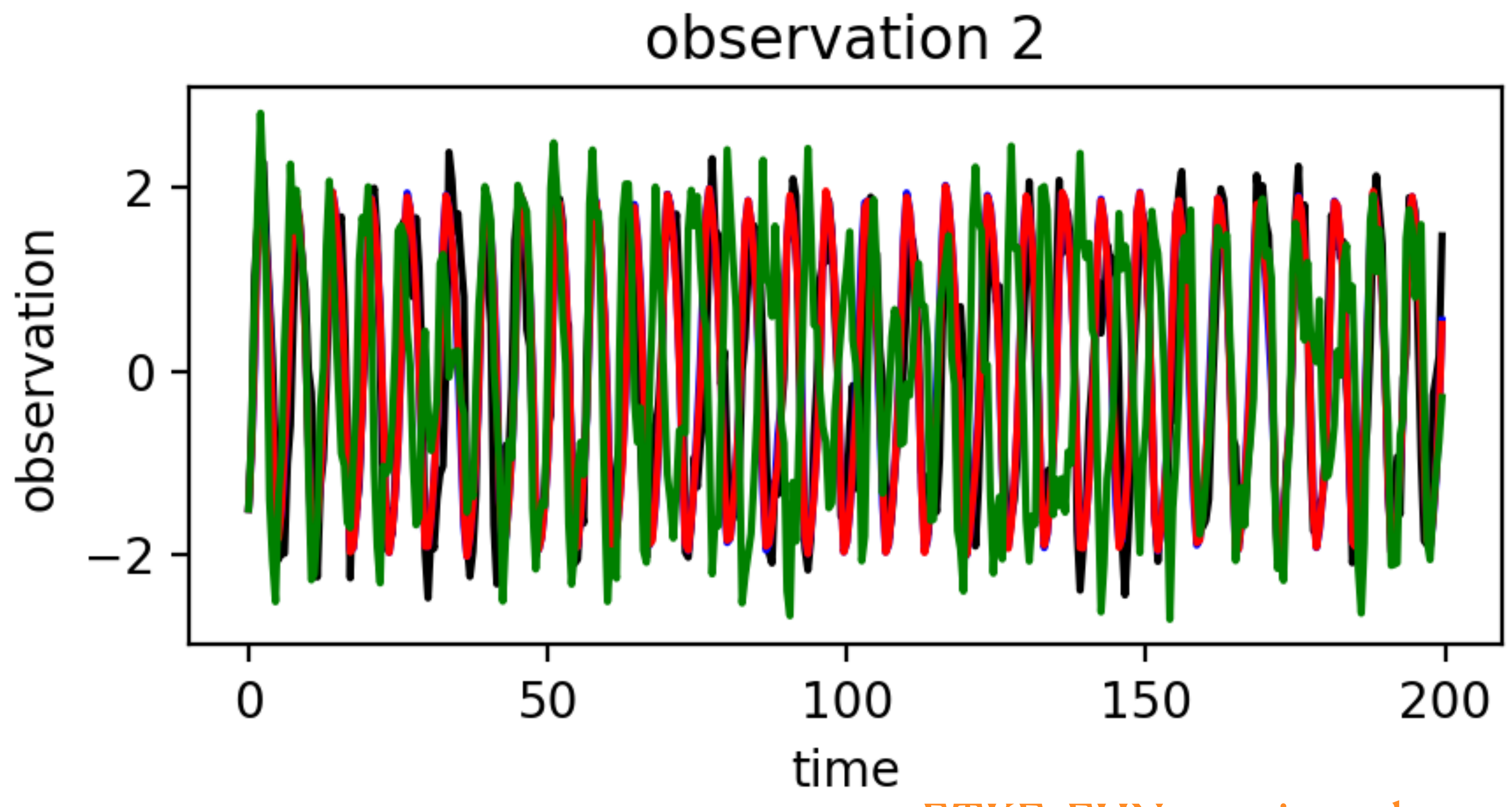
ETKF_FHN_pestim3_da.py



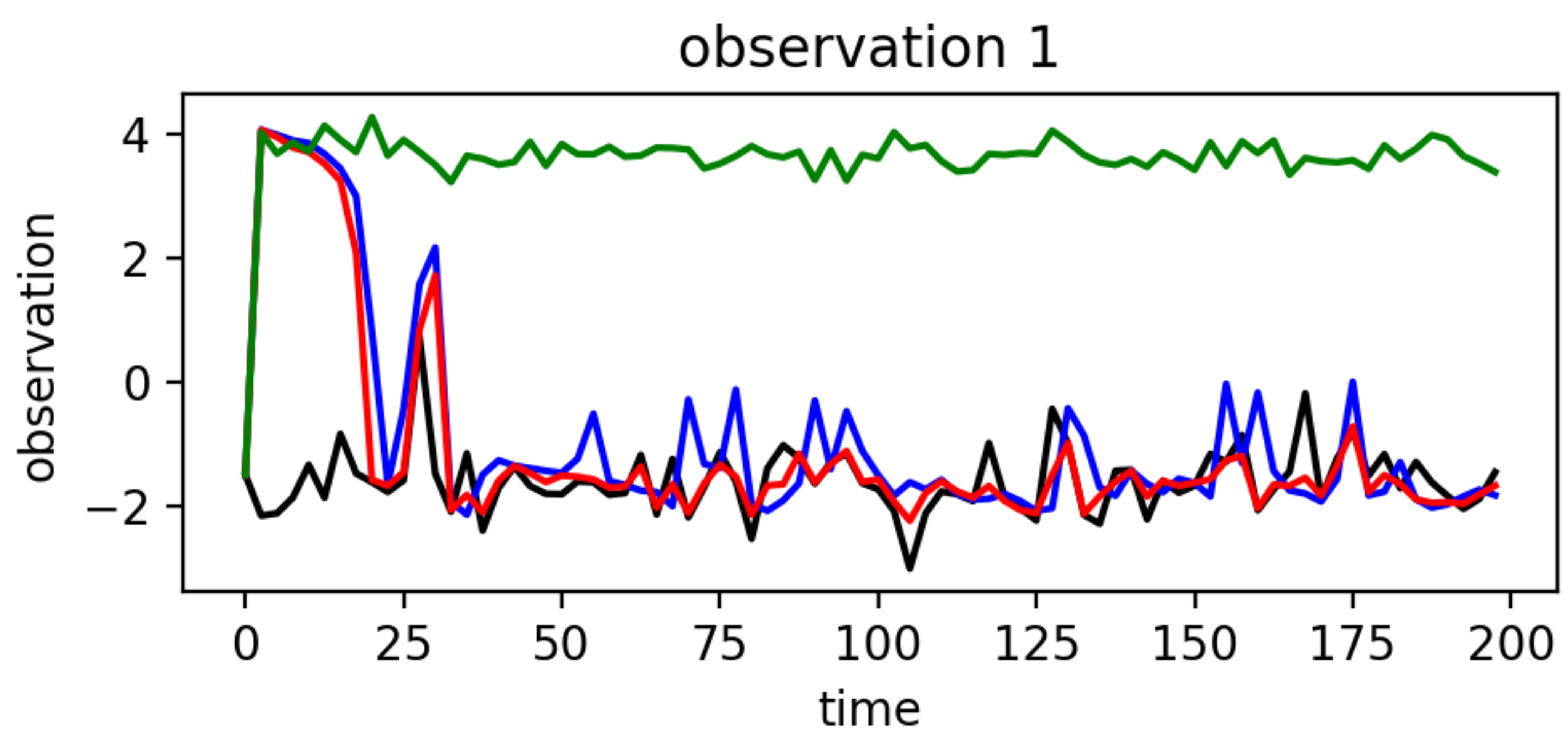


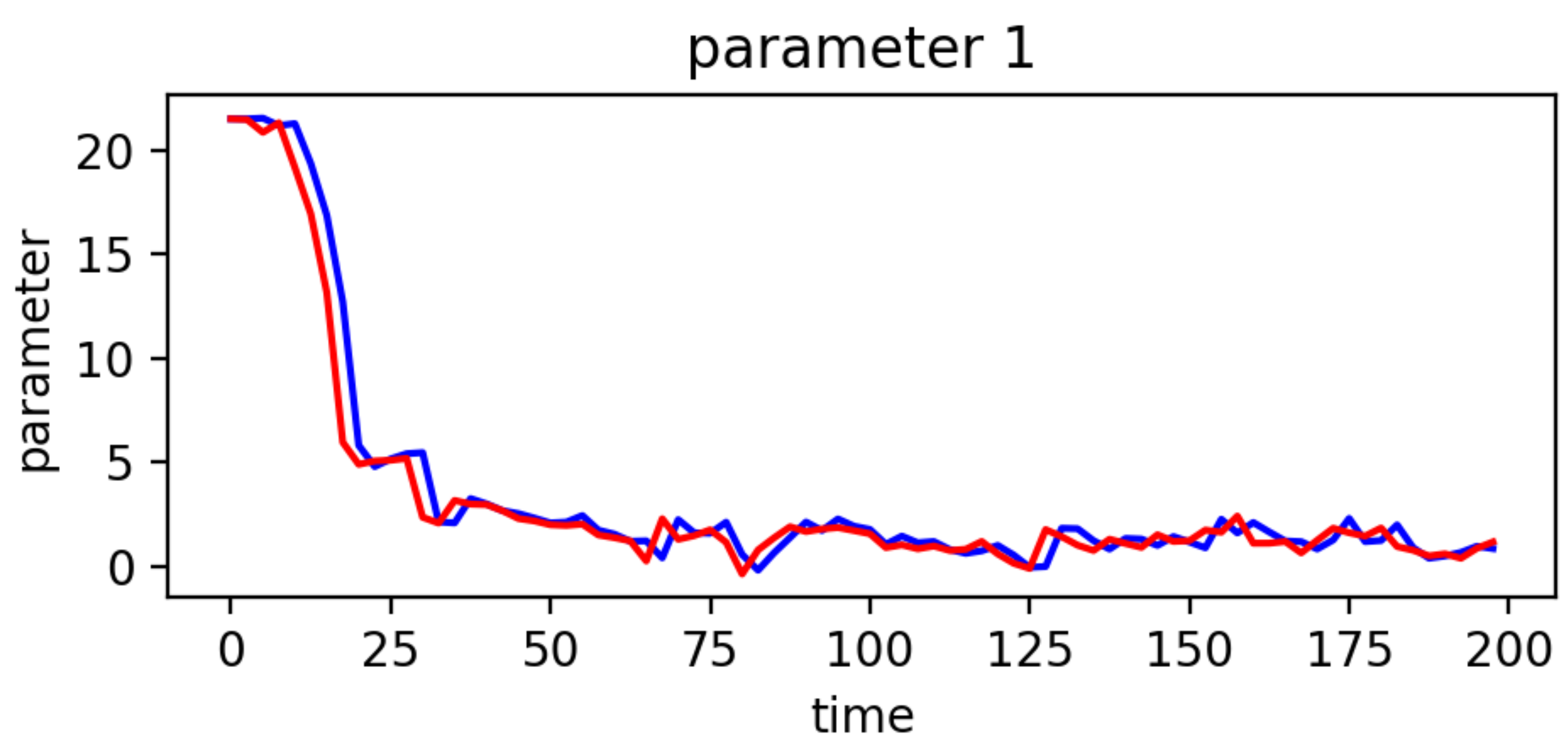
model iteration time: $\Delta t = 0.05$

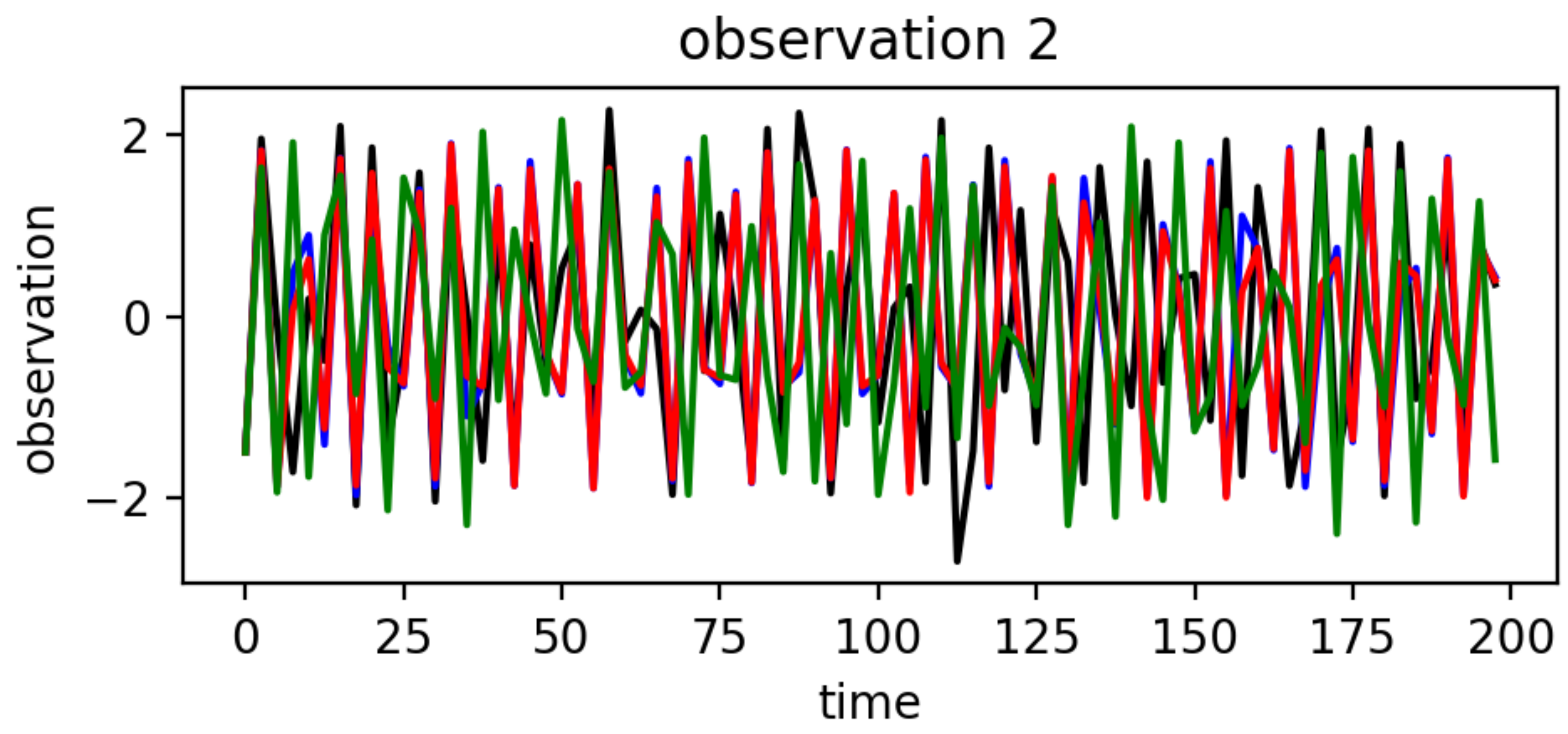
observation sampling time: $10\Delta t$



ETKF_FHN_pestim3_da.py



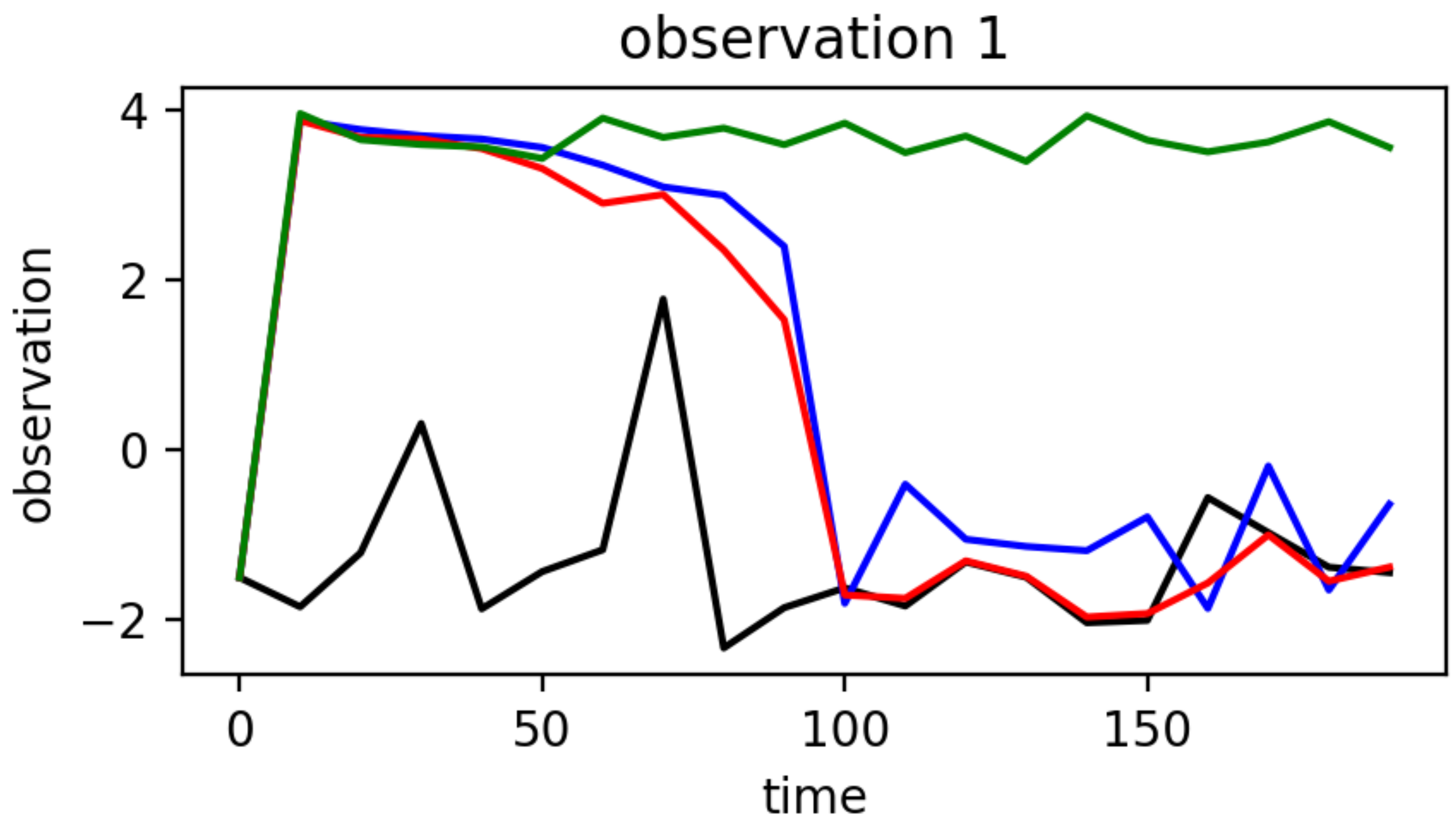


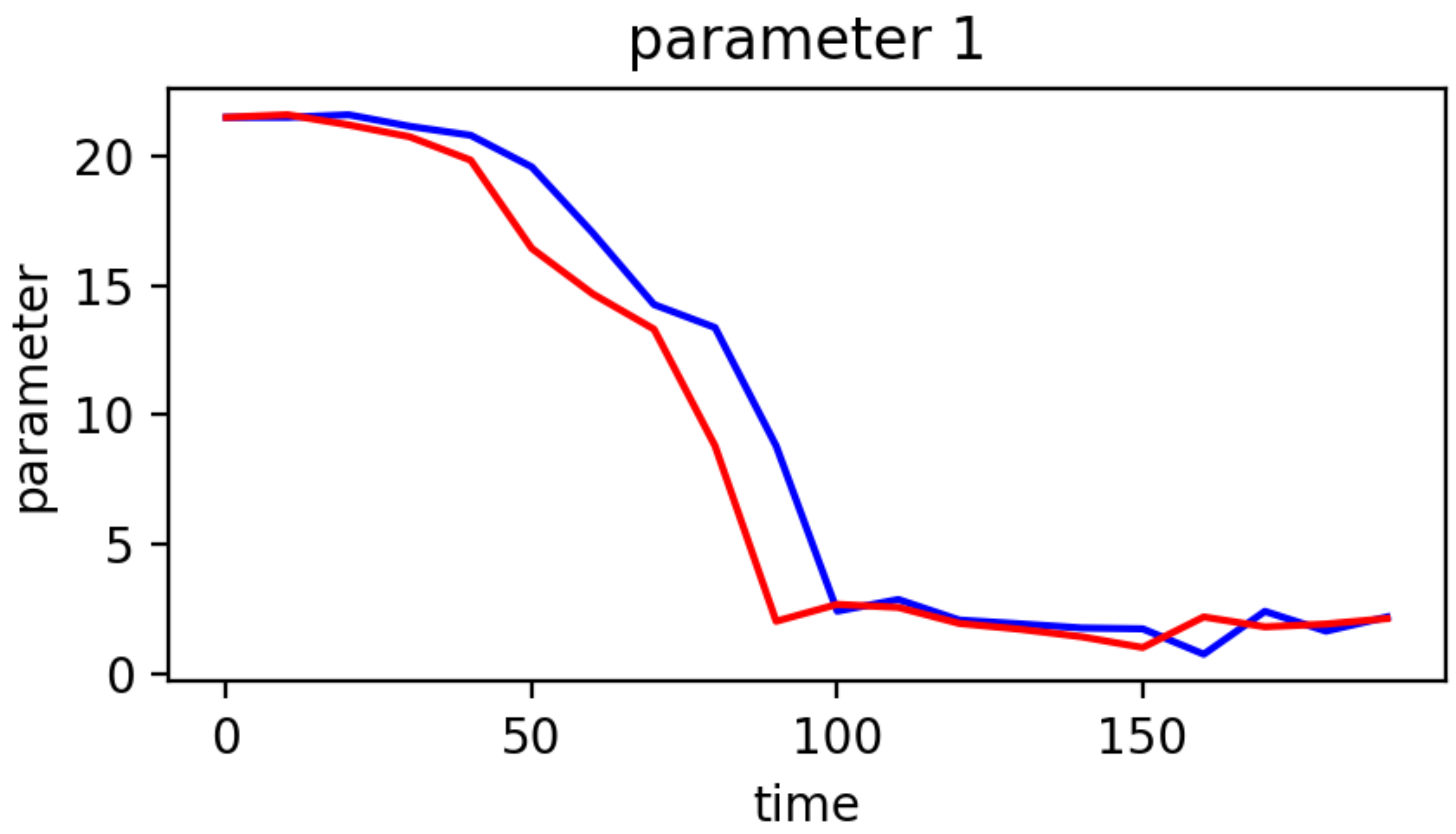


model iteration time: $\Delta t = 0.05$

observation sampling time: $50\Delta t$

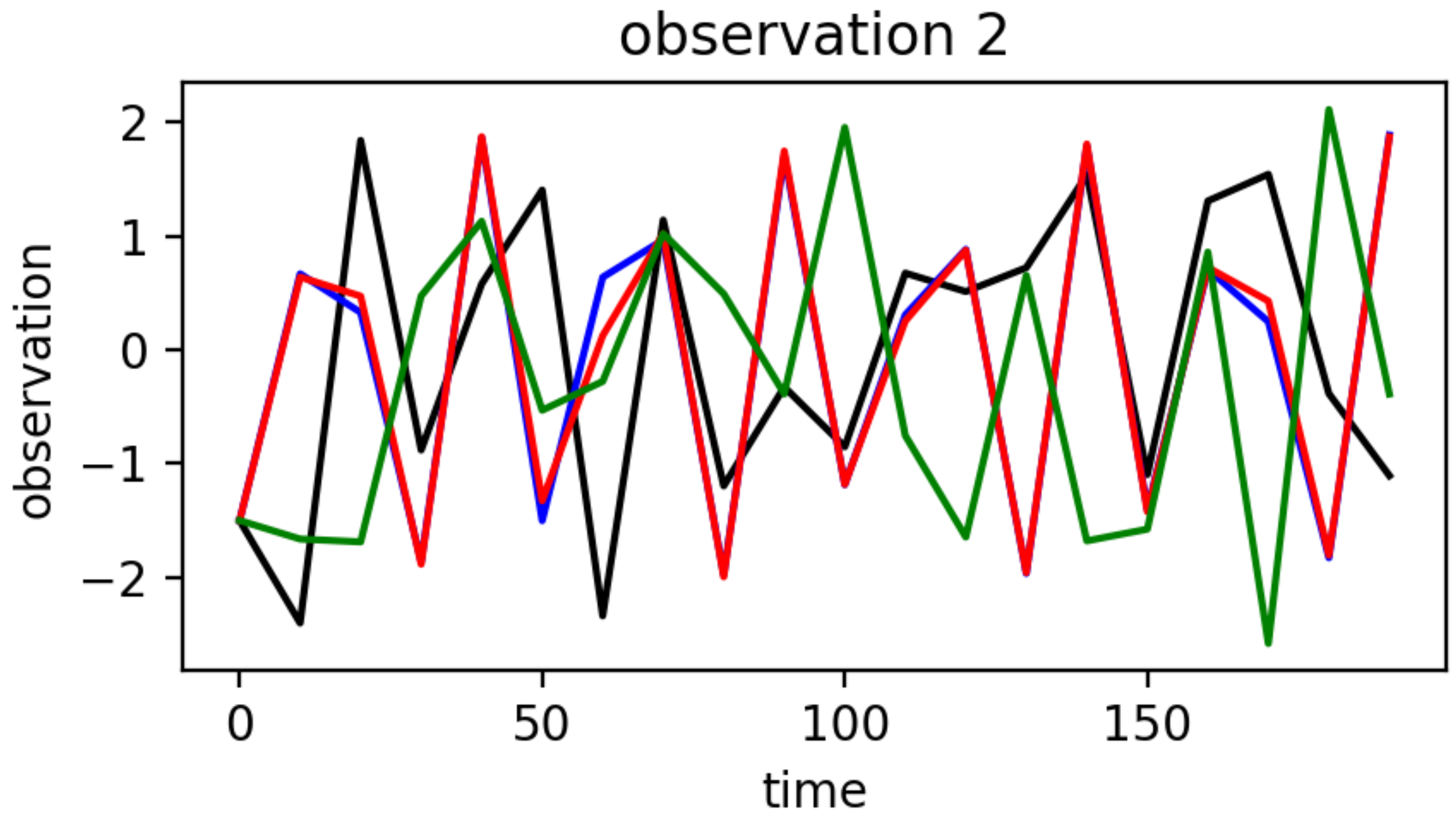
ETKF_FHN_pestim3_da.py





model iteration time: $\Delta t = 0.05$

observation sampling time: $200\Delta t$



ETKF_FHN_pestim3_da.py

motivation

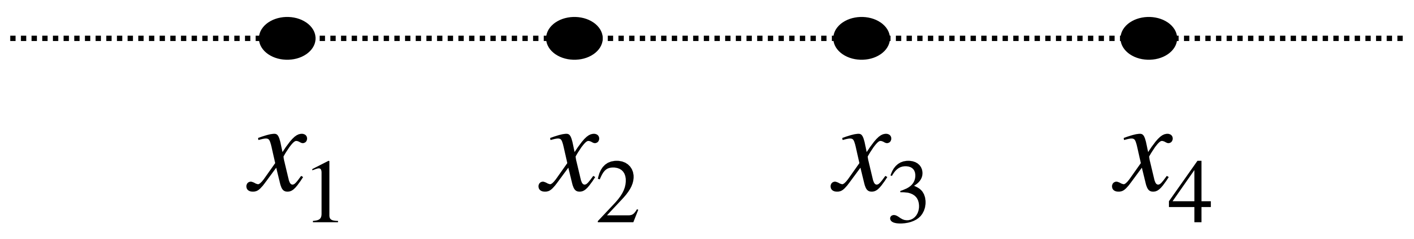
basic methods

Kalman filter

linear EKF UKF **ETKF** **LETKF**

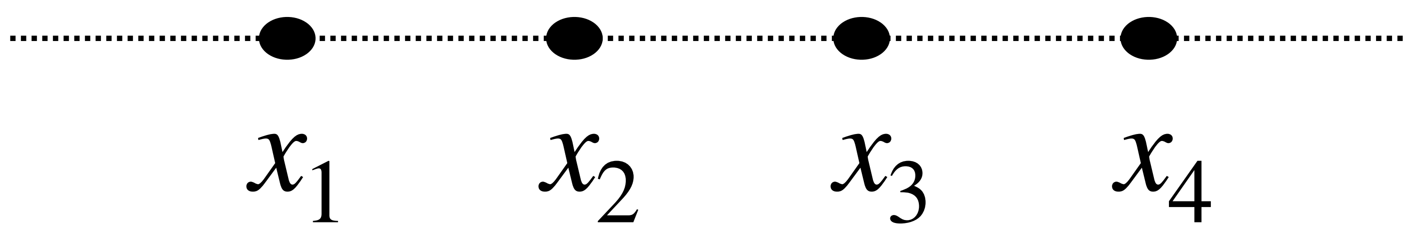
prediction and verification

ETKF:

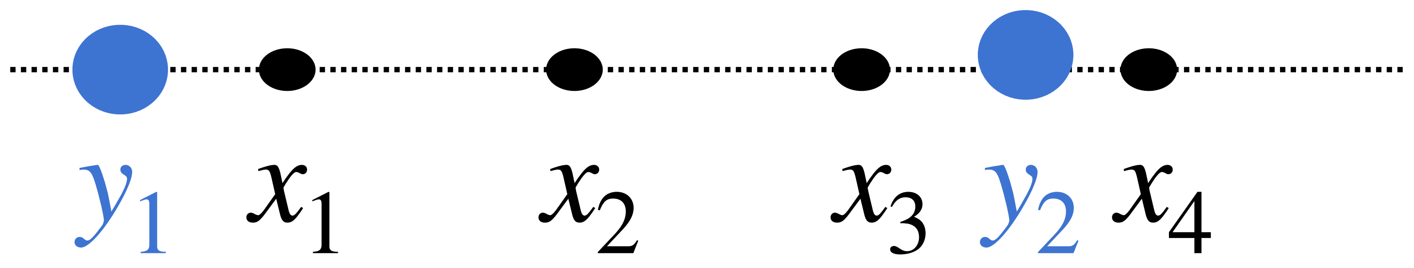


- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$

ETKF:

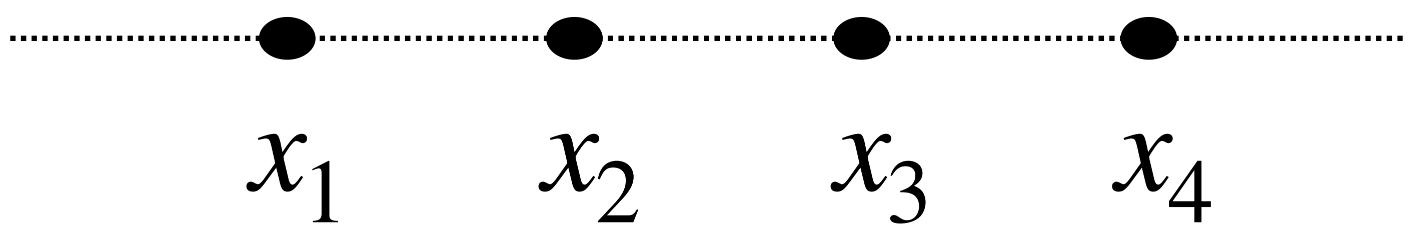


- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$

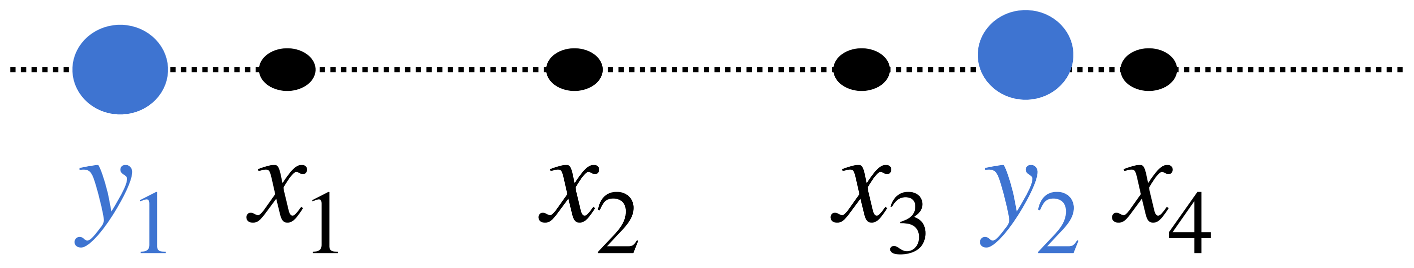


- analysis considers all observations

ETKF:



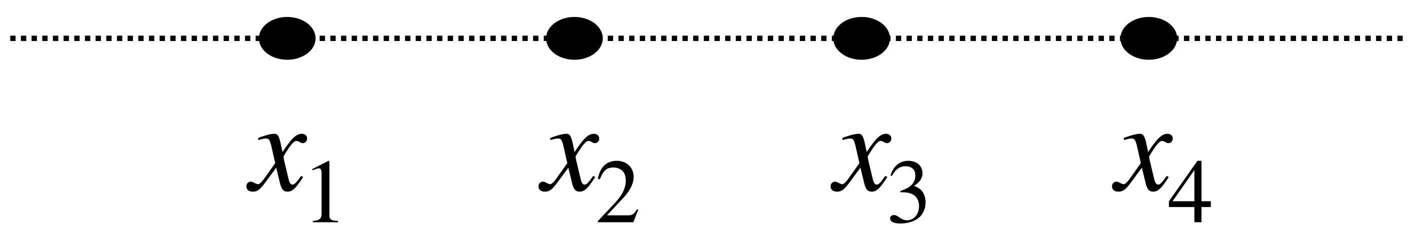
- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$



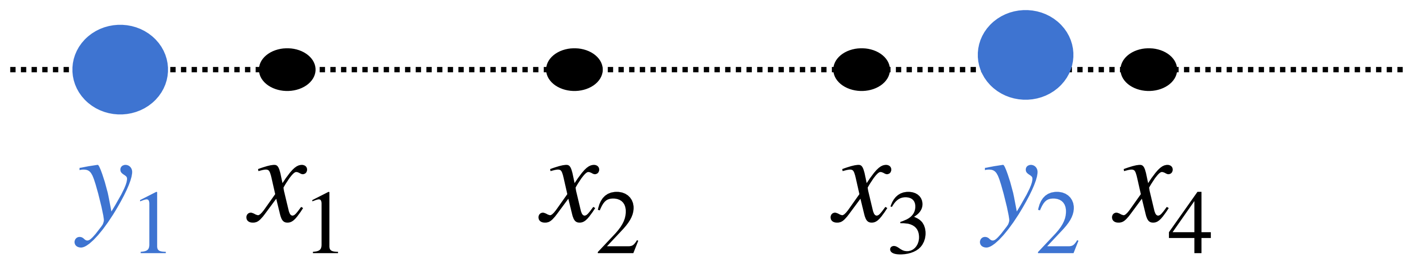
- analysis considers all observations

but: not all observations may affect all state variables

ETKF:



- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$

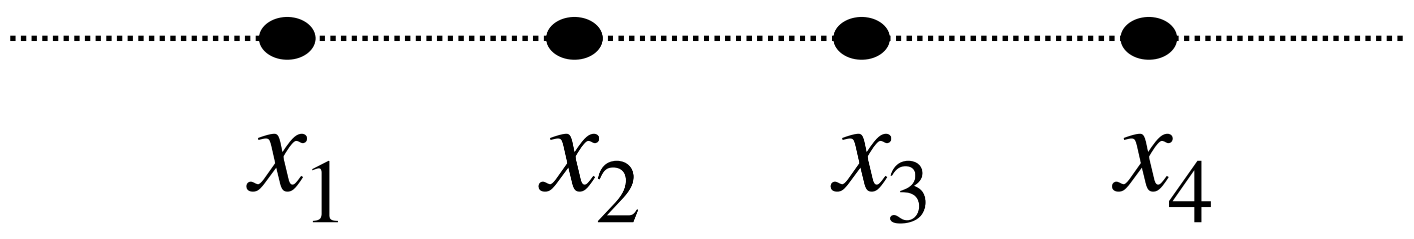


- analysis considers all observations

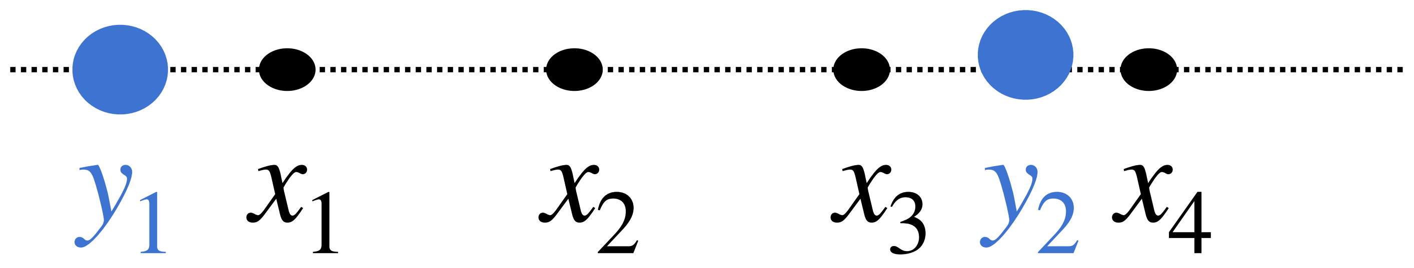
but: not all observations may affect all state variables

idea (Hunt et al. (2007)): **localisation of analysis**

ETKF:



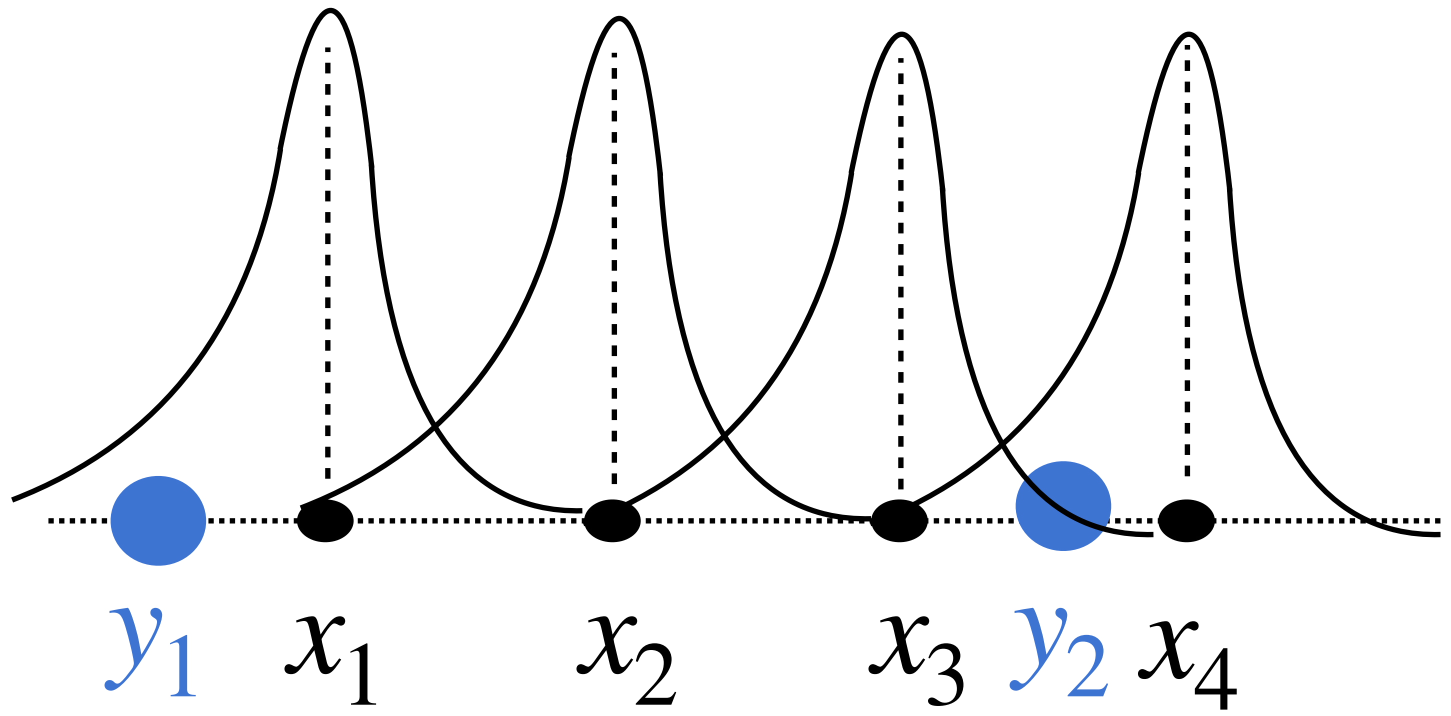
- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$



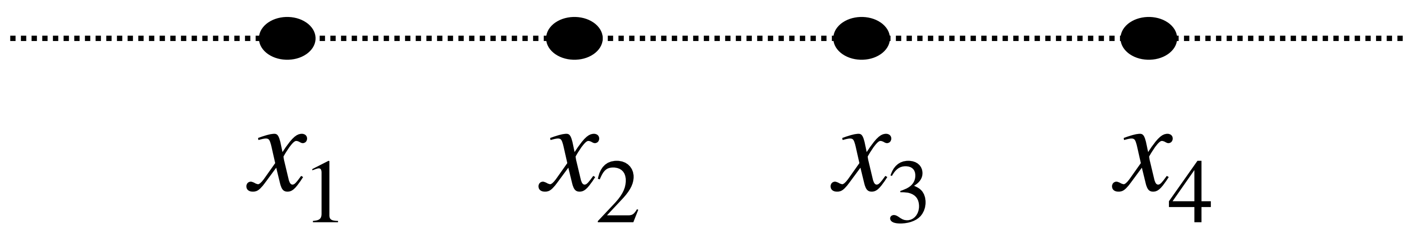
- analysis considers all observations

but: not all observations may affect all state variables

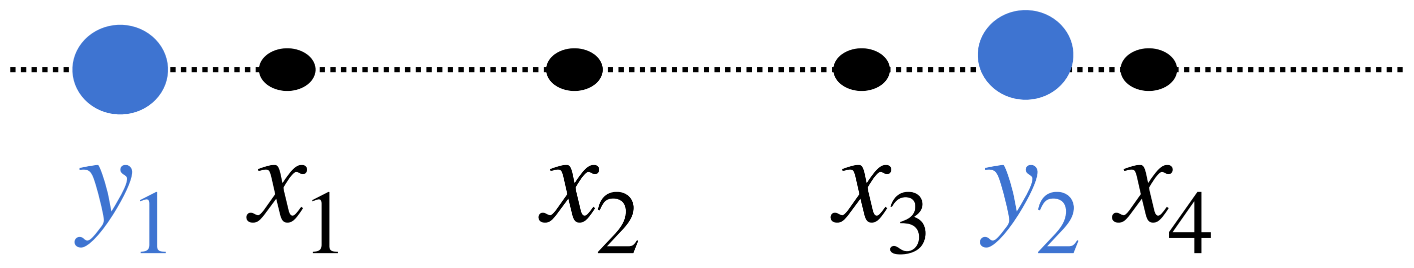
idea (Hunt et al. (2007)): **localisation of analysis**



ETKF:



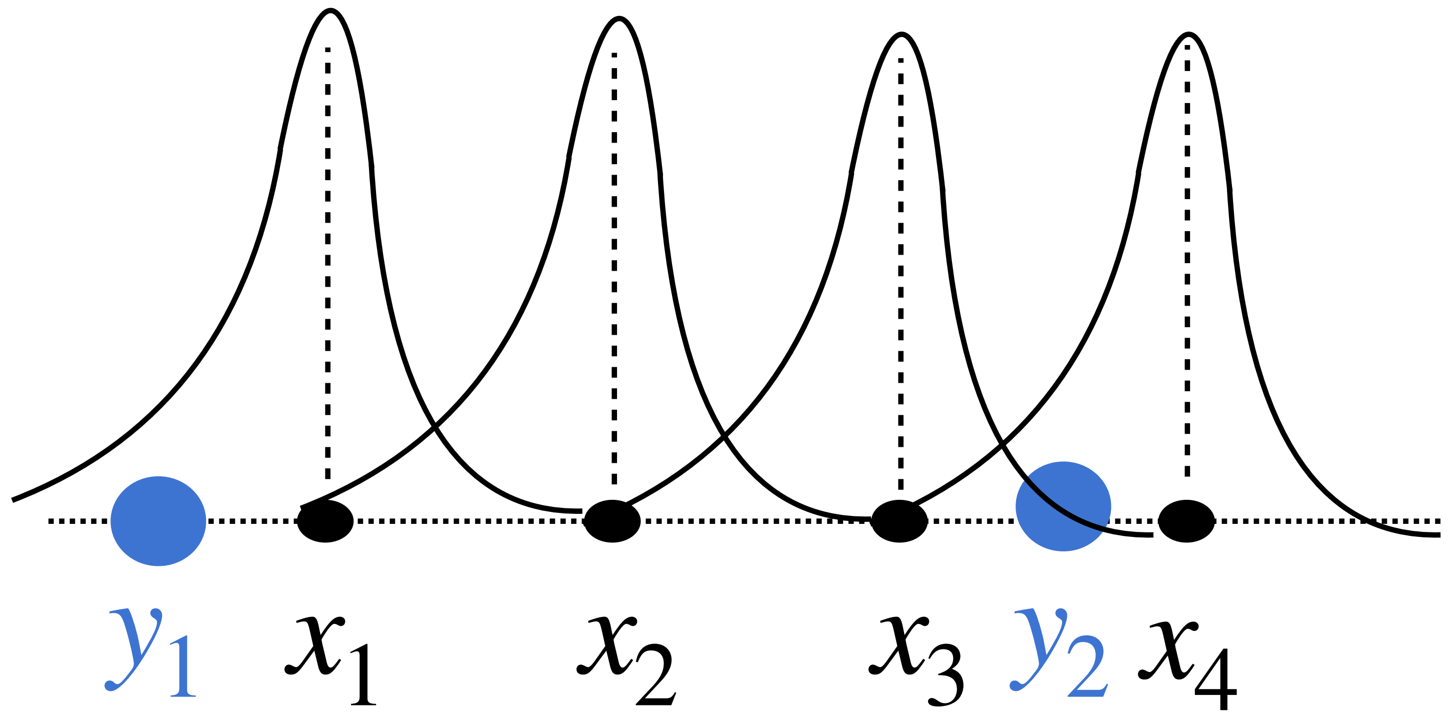
- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$



- analysis considers all observations

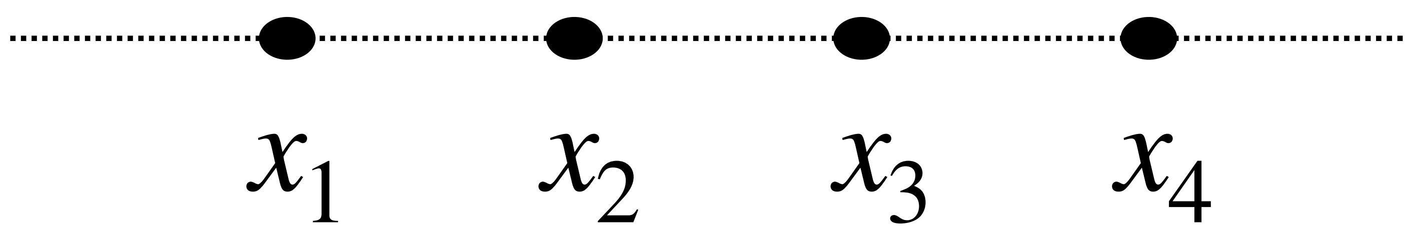
but: not all observations may affect all state variables

idea (Hunt et al. (2007)): **localisation of analysis**

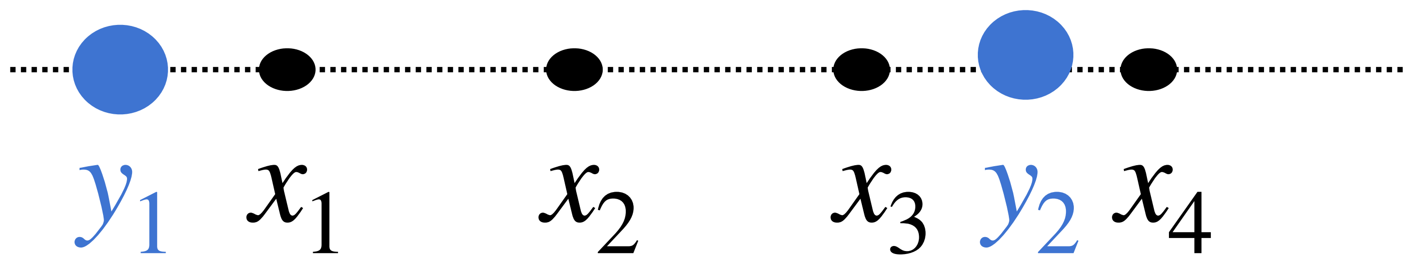


analysis at x_1 considers observation y_1

ETKF:



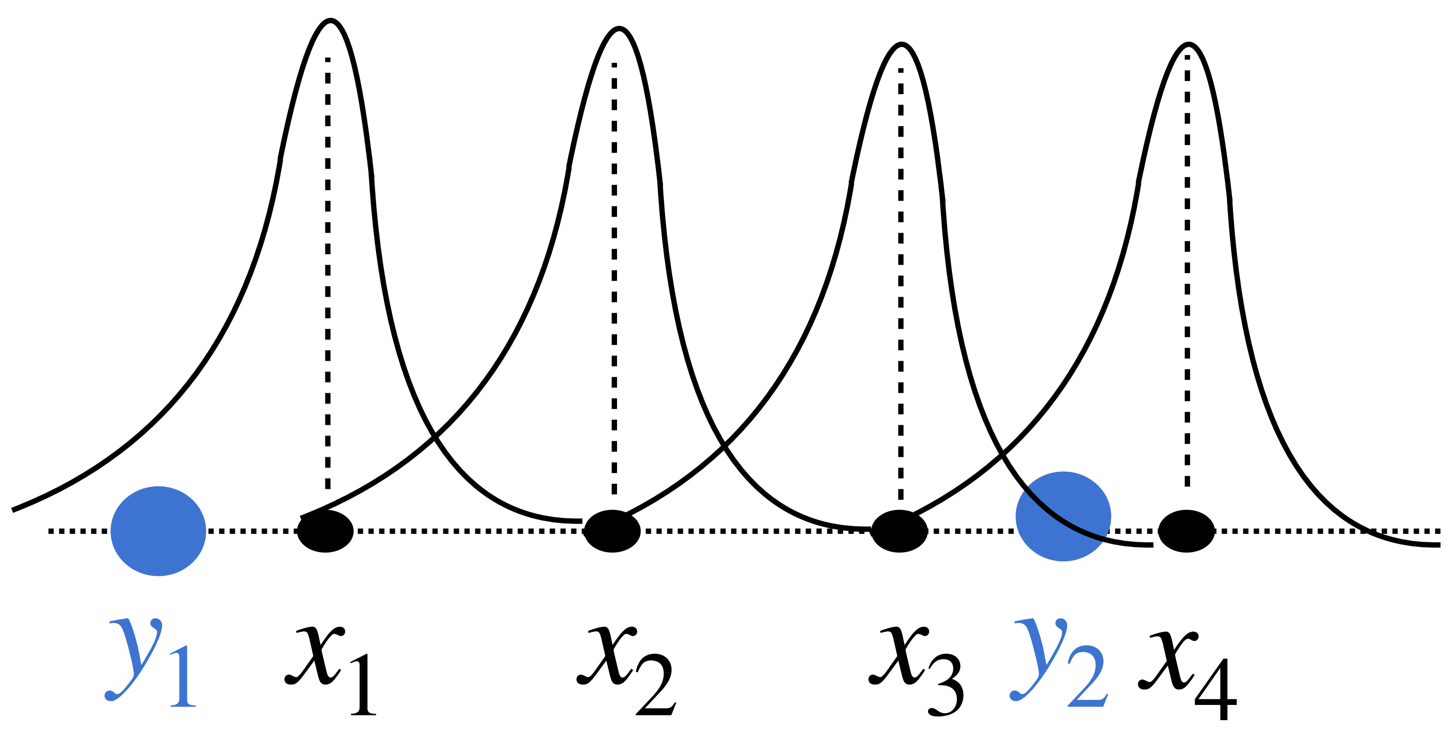
- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$



- analysis considers all observations

but: not all observations may affect all state variables

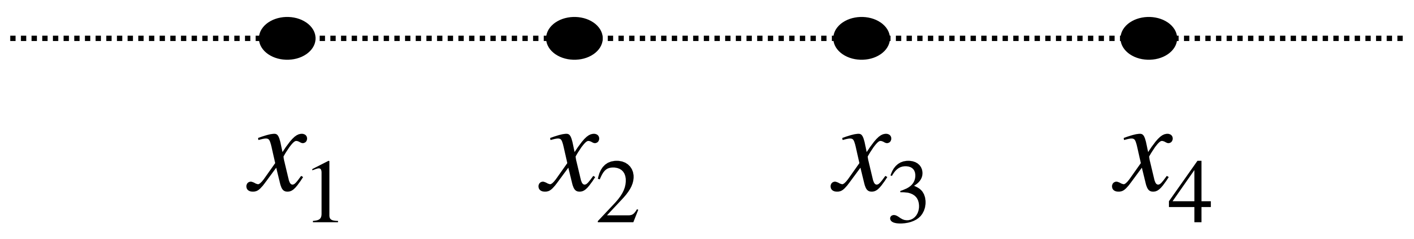
idea (Hunt et al. (2007)): **localisation of analysis**



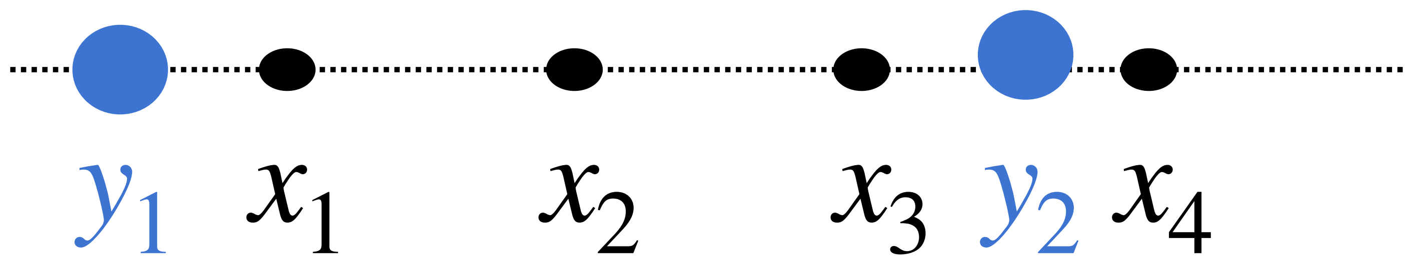
analysis at x_1 considers observation y_1

analysis at x_2 is not computed since observations are too far

ETKF:



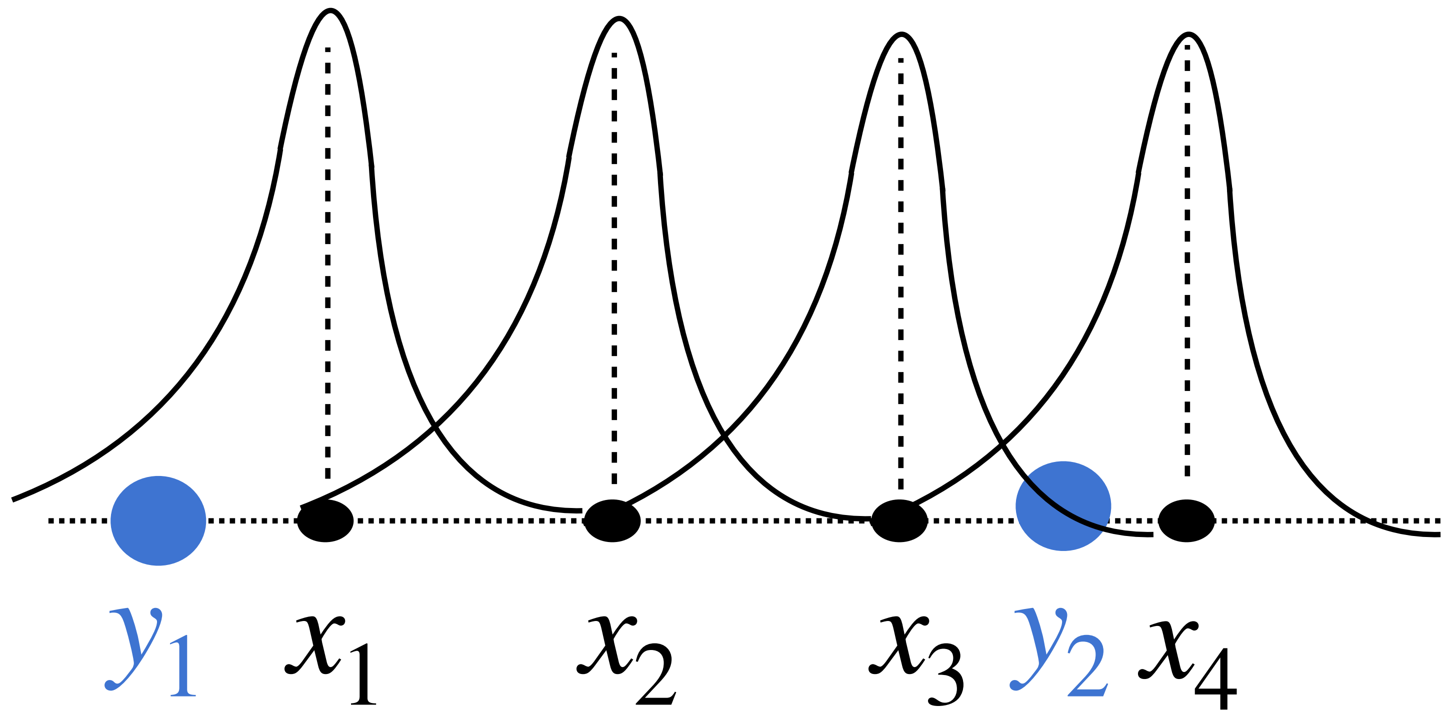
- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$



- analysis considers all observations

but: not all observations may affect all state variables

idea (Hunt et al. (2007)): **localisation of analysis**

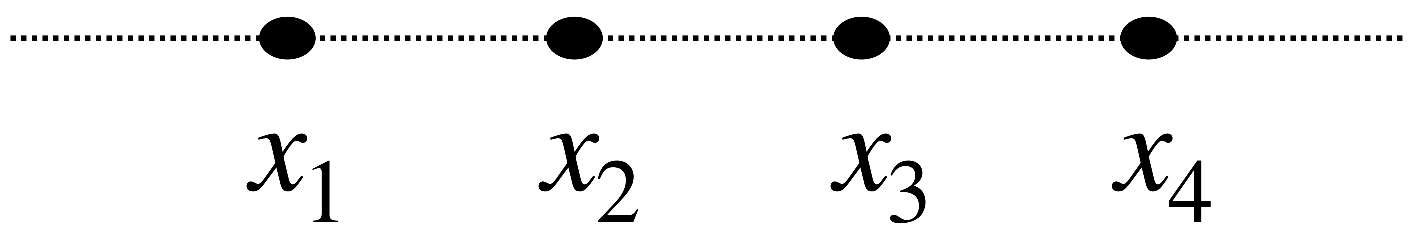


analysis at x_1 considers observation y_1

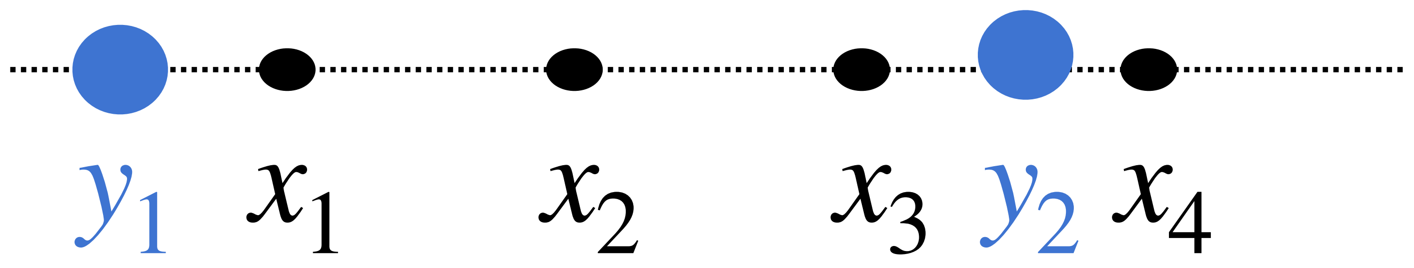
analysis at x_2 is not computed since observations are too far

analysis at x_3 and x_4 considers observation y_2

ETKF:



- description by multivariate probability density function $p(\mathbf{x}_b) = p(x_1^b, x_2^b, \ldots)$

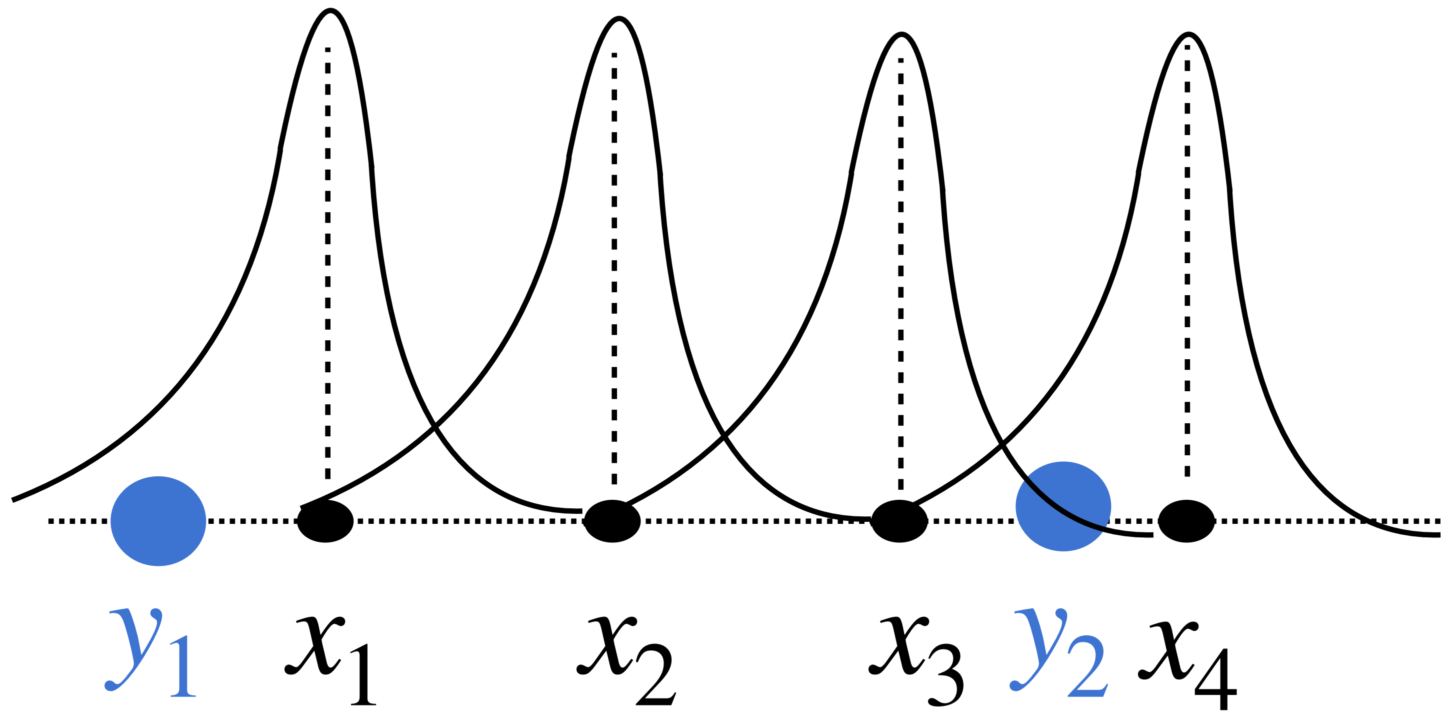


- analysis considers all observations

but: not all observations may affect all state variables

idea (Hunt et al. (2007)): **localisation of analysis**

Localized ETKF (LETKF)



analysis at x_1 considers observation y_1

analysis at x_2 is not computed

analysis at x_3 and x_4 considers observation y_2

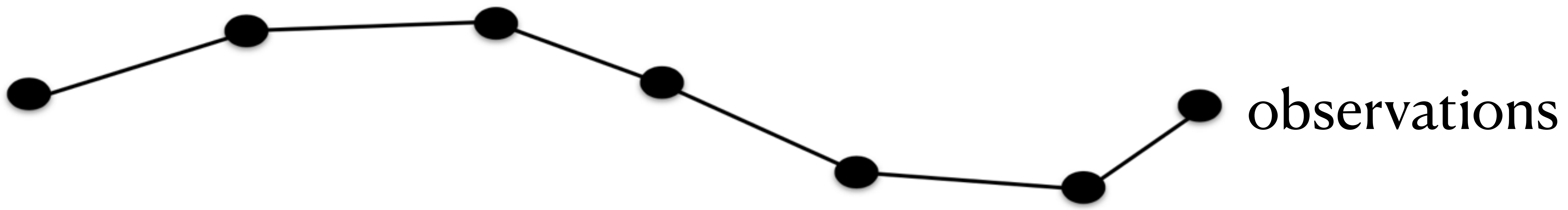
motivation

basic methods

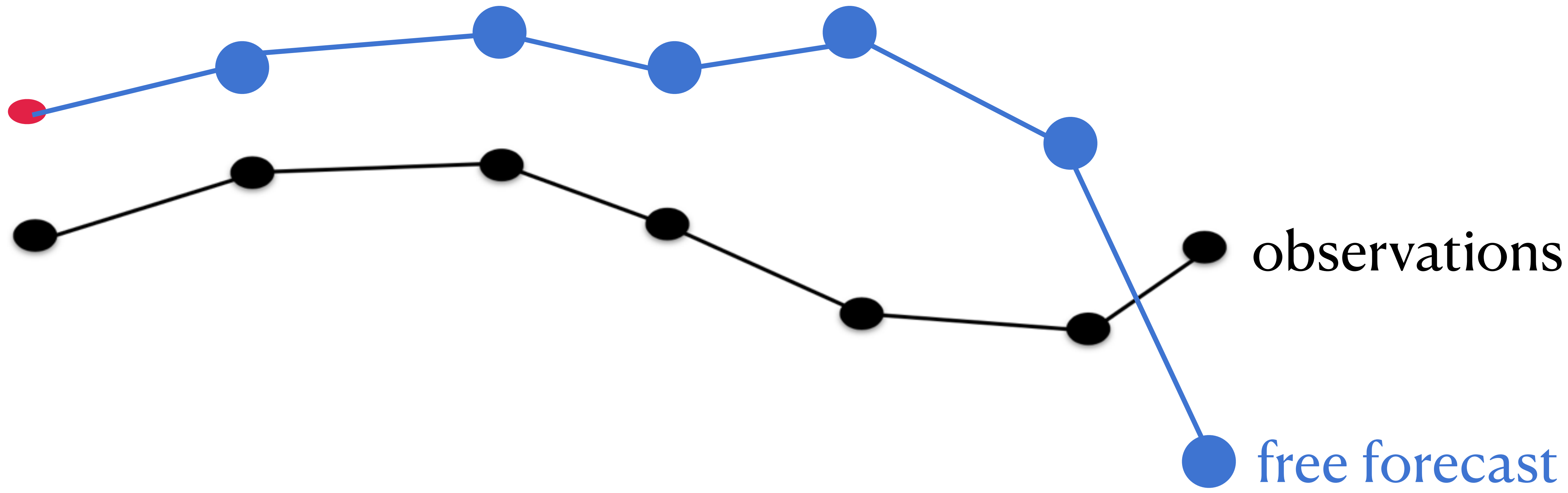
Kalman filter

prediction and verification

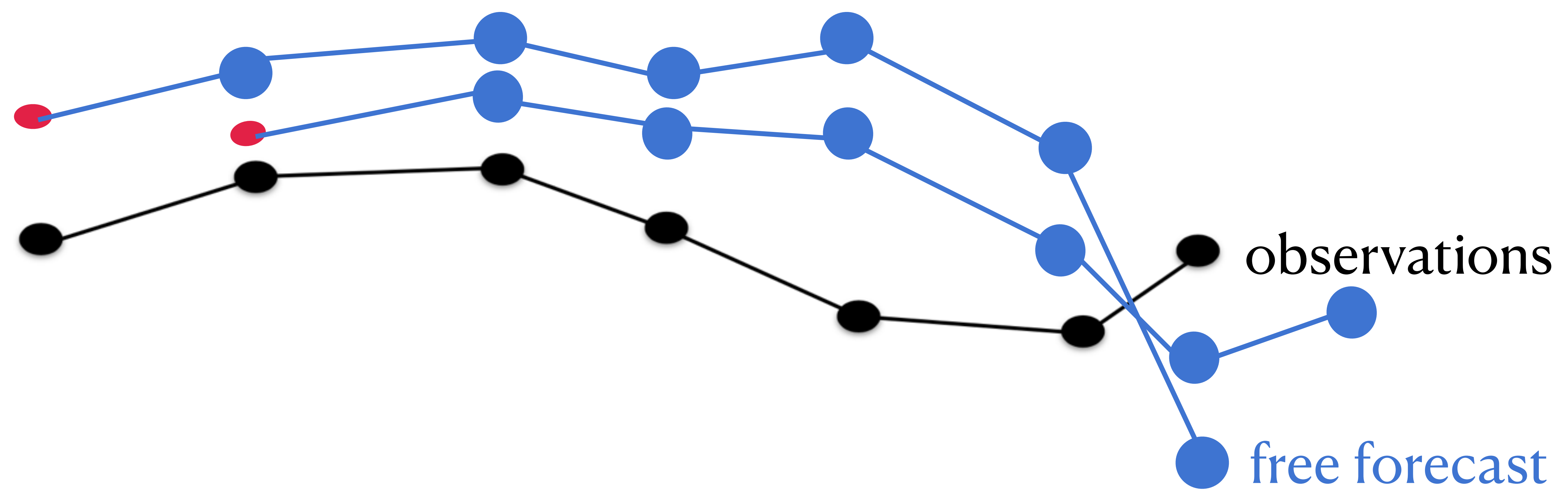
prediction



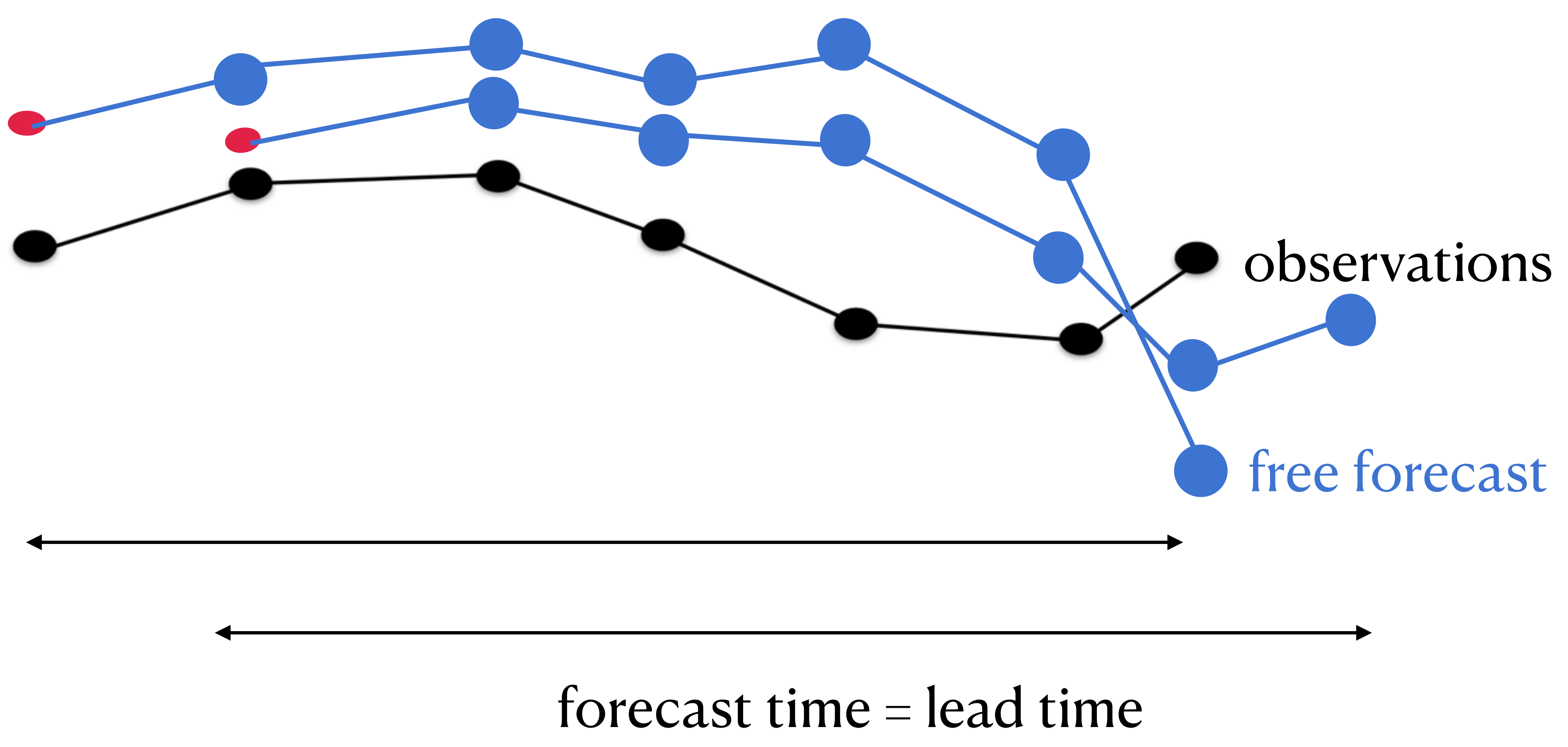
prediction

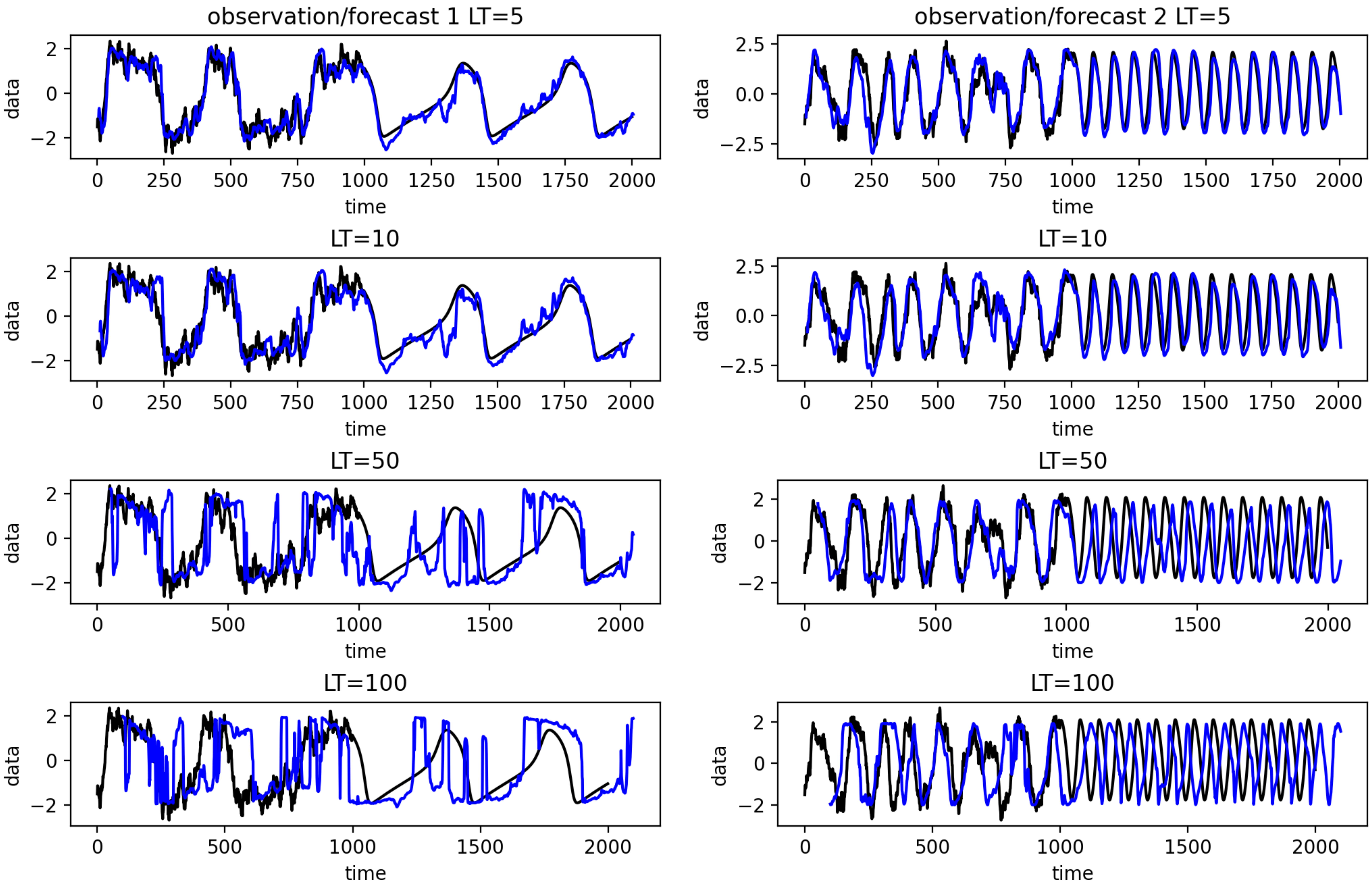


prediction



prediction





- ETKF (L=6)
- **good** forecast for **short** lead time (LT)
- **bad** forecast for **long** lead time (LT)

observation

forecast

ETKF_FHN_5_forecast.py

statistical ensemble verification

ensemble member *l* of **free system state forecast** at time point *n* in observation space: $y_n^{f(l)}$

ensemble mean of **free system state forecast** in observation space: $\bar{y}_n^f = \frac{1}{L}\sum_{l=1}^{L} y_n^{f(l)}$

statistical ensemble verification

ensemble member l of **free system state forecast** at time point n in observation space: $y_n^{f(l)}$

ensemble mean of **free system state forecast** in observation space: $\bar{y}_n^f = \frac{1}{L}\sum_{l=1}^{L} y_n^{f(l)}$

$$\text{bias} = \frac{1}{N}\sum_{n=1}^{N} y_n - \bar{y}_n^f$$

temporal mean error of
predicted forecast compared to observation y_n

statistical ensemble verification

ensemble member *l* of **free system state forecast** at time point *n* in observation space: $y_n^{f(l)}$

ensemble mean of **free system state forecast** in observation space: $\bar{y}_n^f = \frac{1}{L}\sum_{l=1}^{L} y_n^{f(l)}$

$$\text{bias} = \frac{1}{N}\sum_{n=1}^{N} y_n - \bar{y}_n^f$$

temporal mean error of
predicted forecast compared to observation y_n

$$\text{rmse} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (y_n - \bar{y}_n^f)^2}$$

root mean squared error of
predicted forecast compared to observation y_n

statistical ensemble verification

ensemble member *l* of **free system state forecast** at time point *n* in observation space: $y_n^{f(l)}$

ensemble mean of **free system state forecast** in observation space: $\bar{y}_n^f = \frac{1}{L}\sum_{l=1}^{L} y_n^{f(l)}$

$$\text{bias} = \frac{1}{N}\sum_{n=1}^{N} y_n - \bar{y}_n^f$$

temporal mean error of
predicted forecast compared to observation y_n

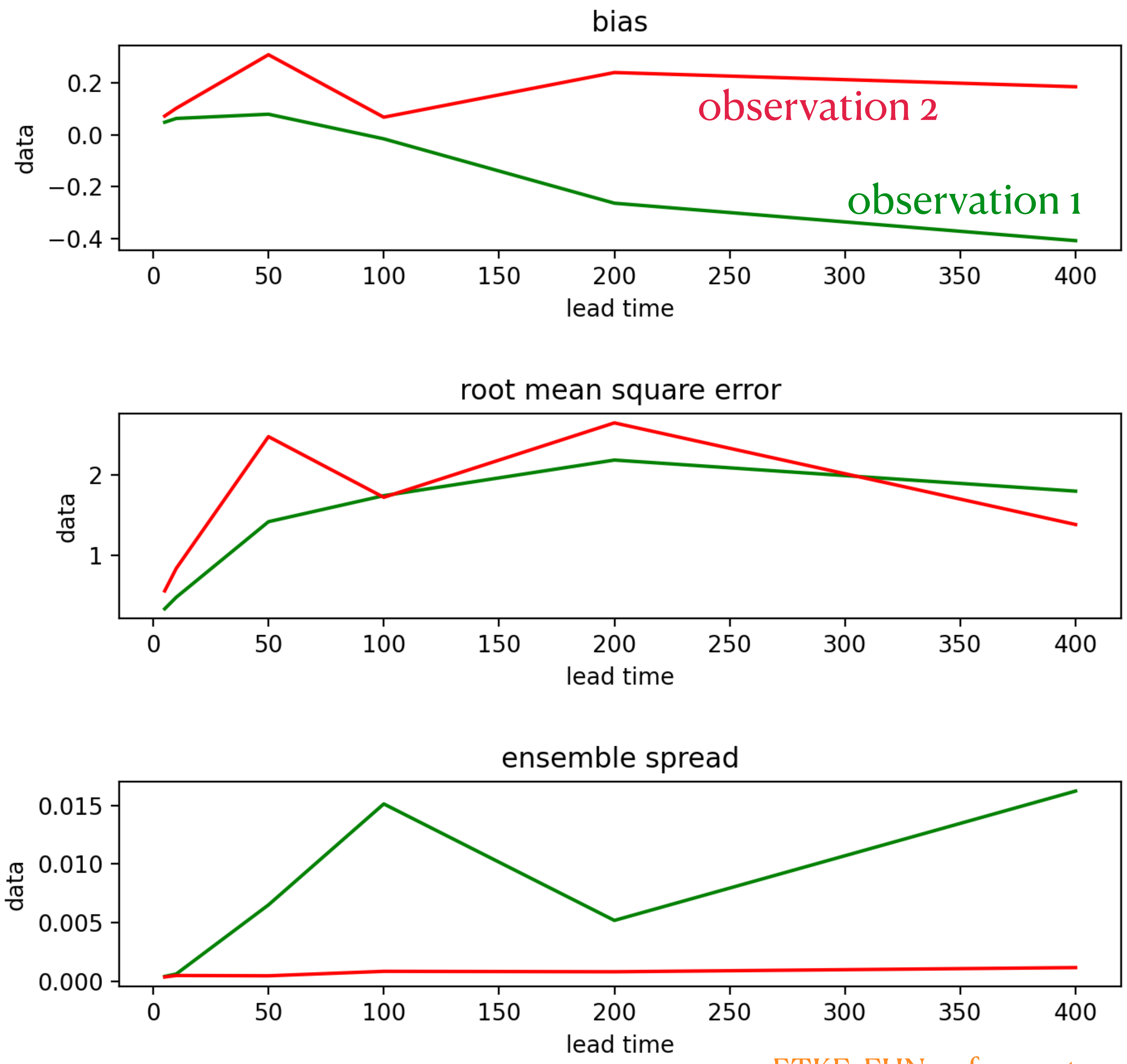
$$\text{rmse} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \bar{y}_n^f)^2}$$

root mean squared error of
predicted forecast compared to observation y_n

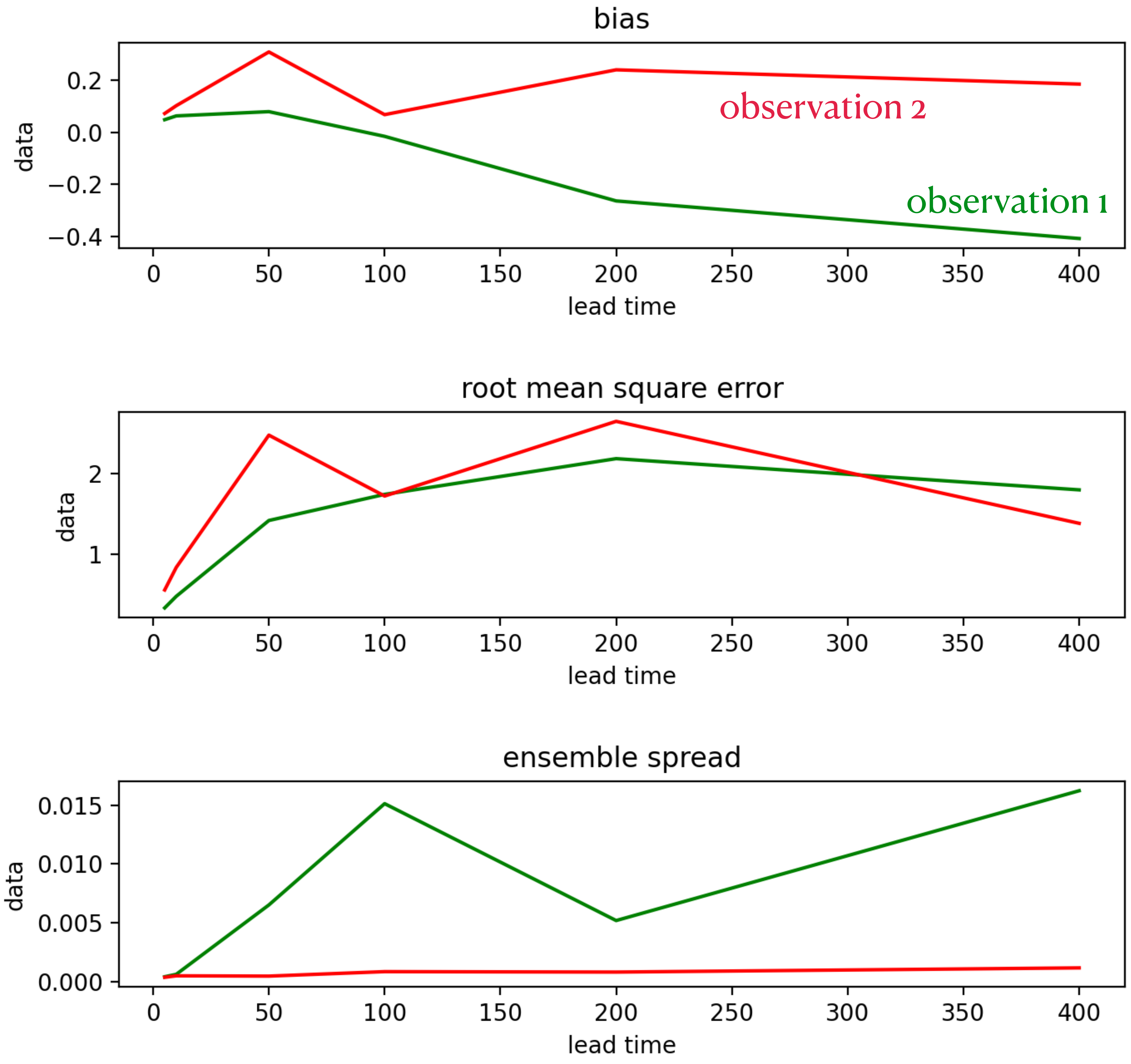
$$\text{spread} = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\frac{1}{L-1}\sum_{l=1}^{L}(y_n^{f(l)} - \bar{y}_n^f)^2}$$

mean ensemble spread of
predicted forecast ensemble compared to observation y_n

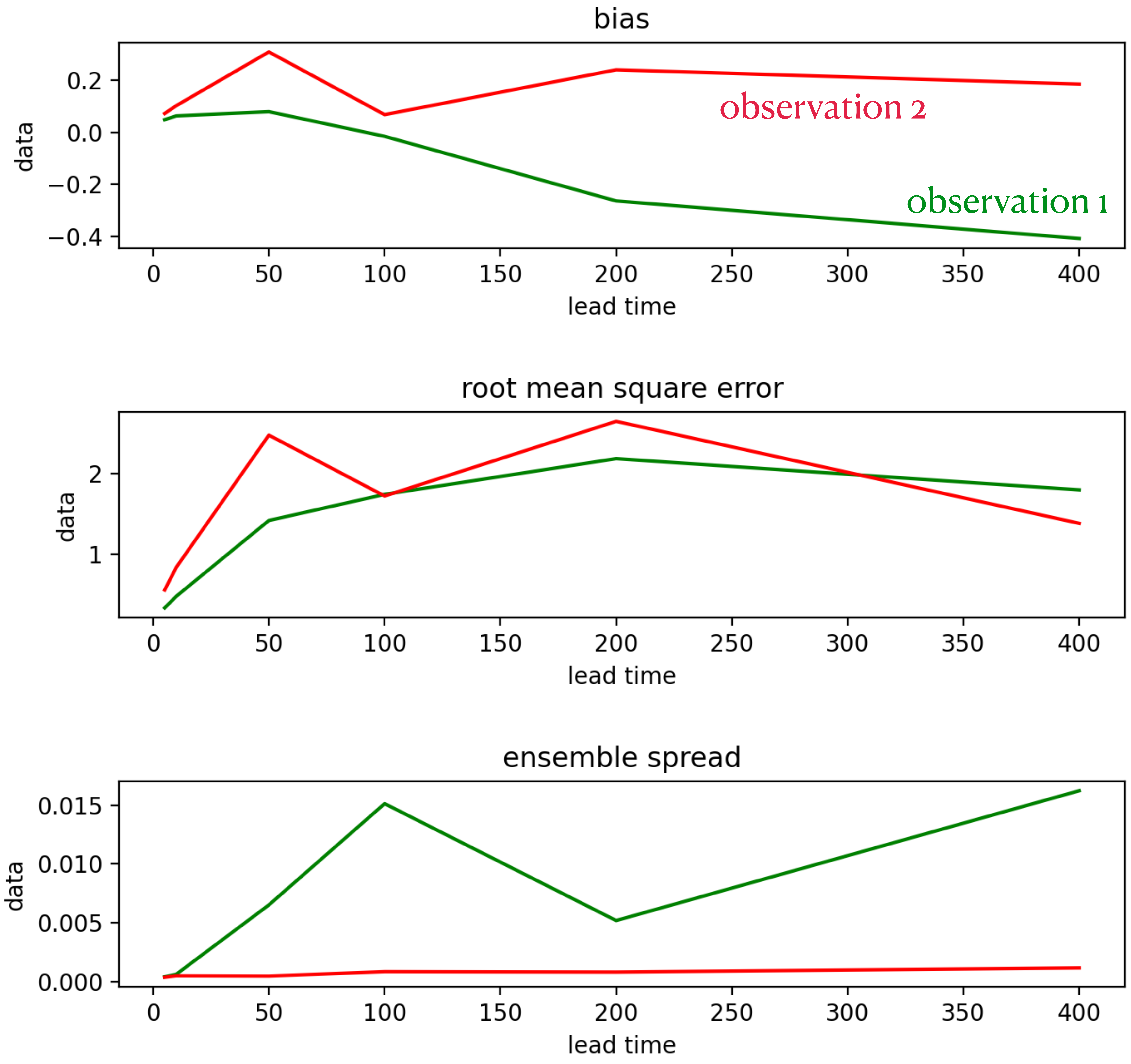
statistical verification



ETKF_FHN_5_forecast.py

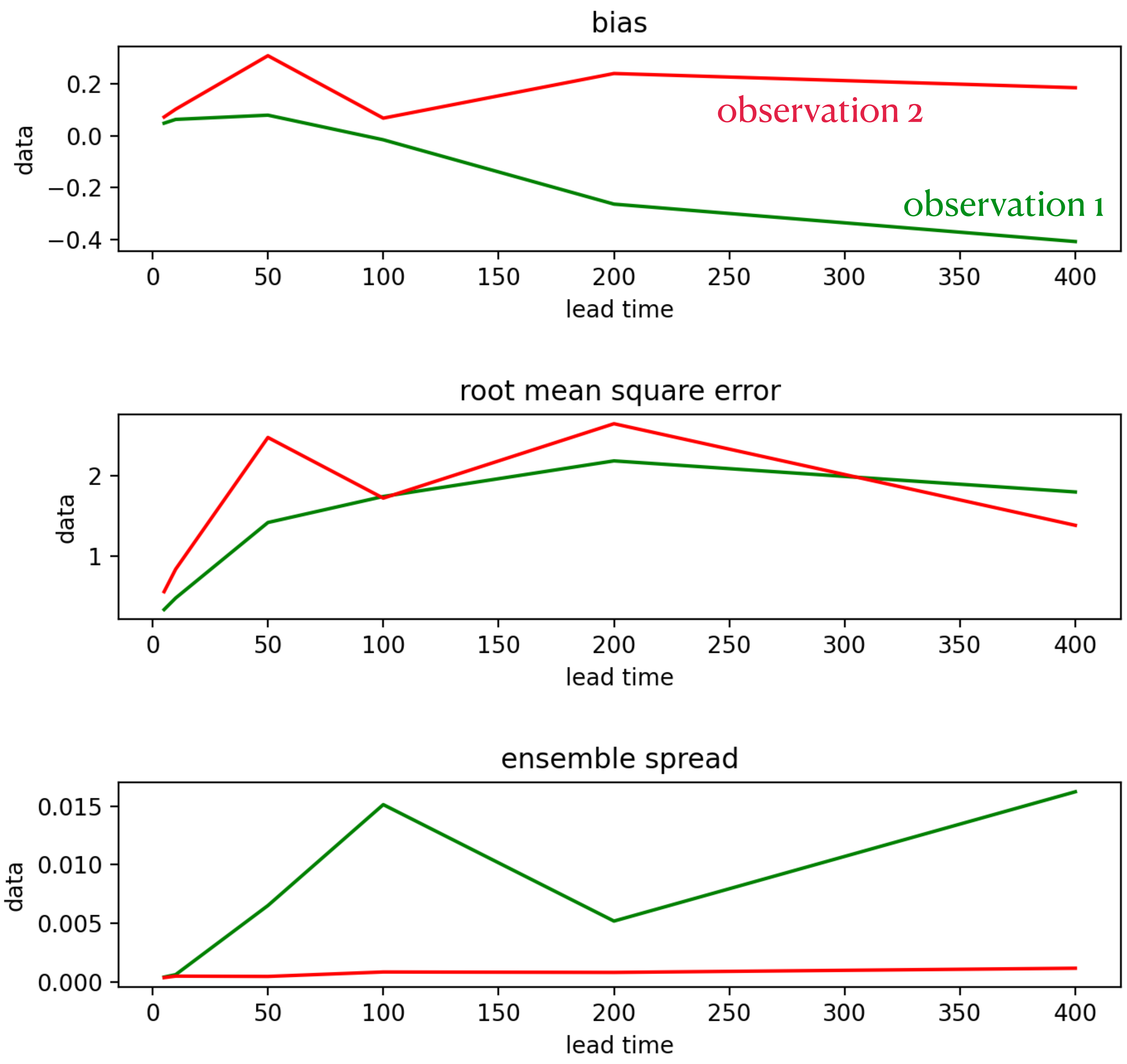


observation 1
has larger bias and ensemble spread



observation 1
has larger bias and ensemble spread

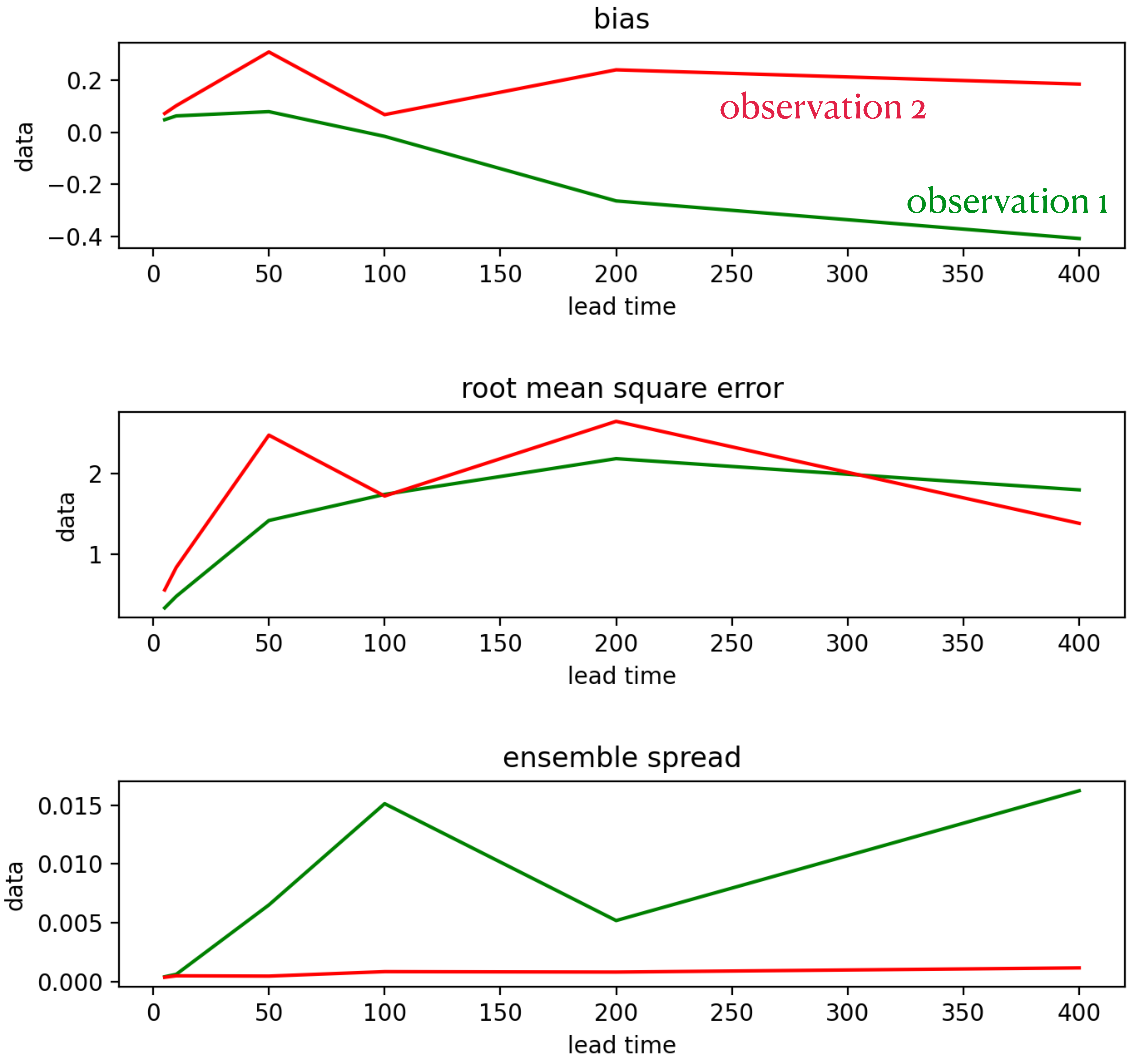
observation 2
has smaller bias and ensemble spread



observation 1
has larger bias and ensemble spread

observation 2
has smaller bias and ensemble spread

observation 1 and 2
have similar rmse



observation 1
has larger bias and ensemble spread

observation 2
has smaller bias and ensemble spread

observation 1 and 2
have similar rmse

→ **forecast of observation 2 is better**


Thank you for your attention

Python libraries for Data assimilation:

ADAO: https://pypi.org/project/adao/

DAPPER: https://github.com/linix100/DAPPER

You have any additional questions to me ? Send me an email to *axel.hutt@inria.fr*