

# 工学応用の観点からのデータ同化 とその特徴

---

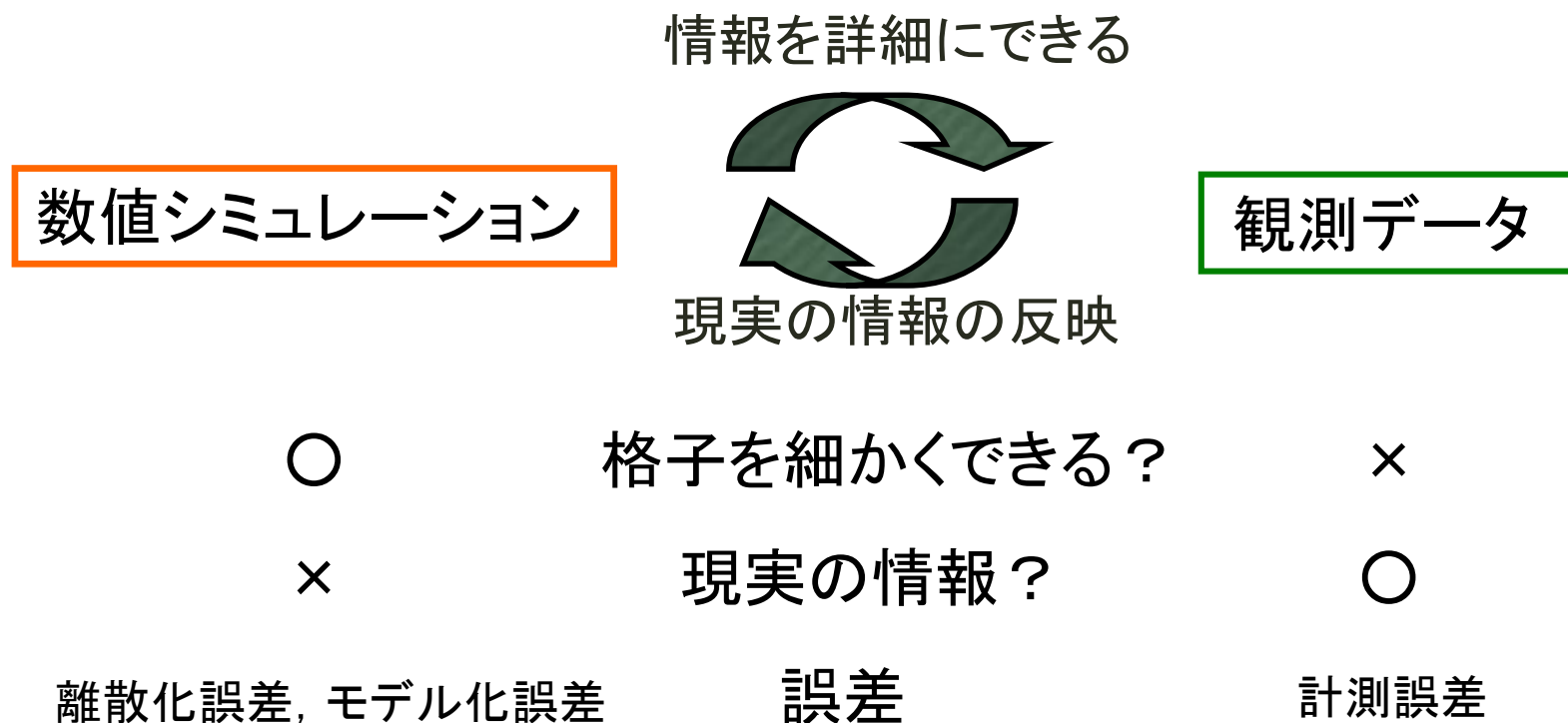
明治大学 中村和幸

# 目次

---

- ✓ データ同化と適用例
  - データ同化とは
  - 適用例
- ✓ データ同化における定式化とアルゴリズム
  - データ同化と状態空間モデル
  - ベイズ更新
  - データ同化アルゴリズム
- ✓ 工学応用に向けたデータ同化の位置づけ
  - 他の類似手法との比較
- ✓ まとめ

# データ同化の目的



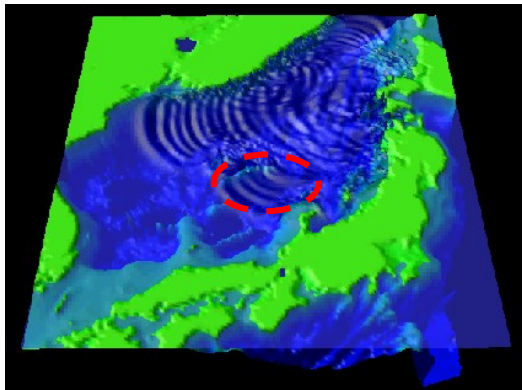
良いところ取りをしたい！

# データ同化でできること

---

- ✓ 予測のための初期条件の構成
  - 予報精度の向上を目指す
  - 現業の天気予報ですでに行われている
- ✓ 観測できない物理変数や状態の推定
  - 3次元, 4次元的な再構成
  - シミュレーションモデルと組み合わせることで, 適切な力学的制約が入る
- ✓ 感度解析
  - 効率のよい計測点・データの設計
- ✓ 経験的パラメータの推定
- ✓ 境界条件の推定

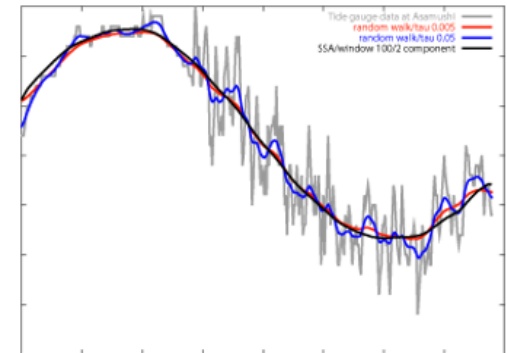
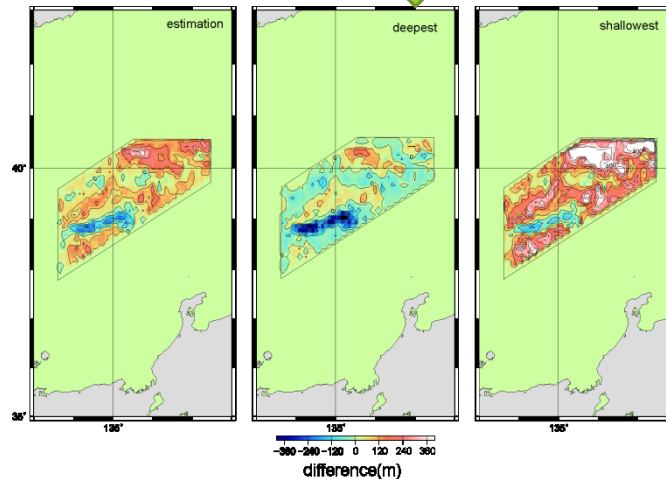
# データ同化例1 津波データ同化



津波モデル



データ同化  
による解析

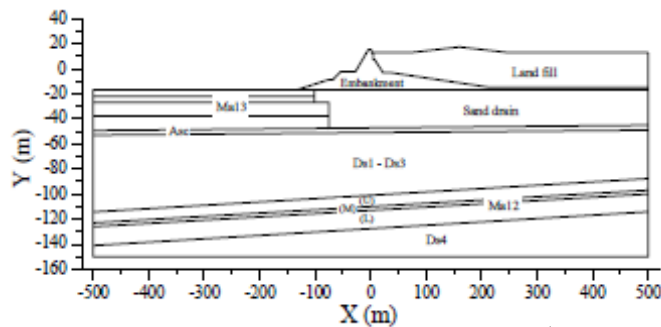


潮位計データ

不確かな海底地形  
の推定

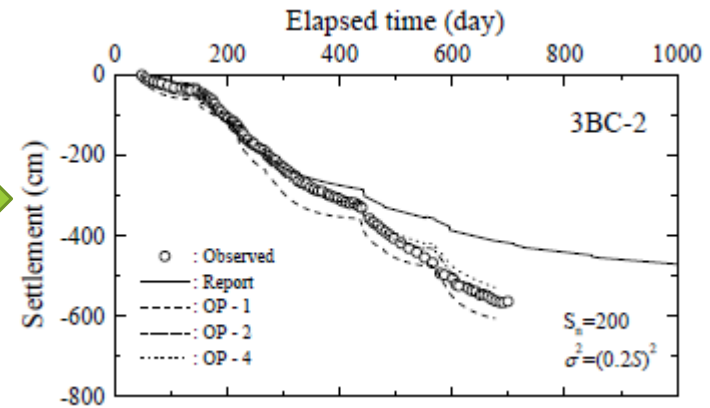
( 樋口(統数研), 広瀬(九大), B.H. Choi(Sung Kyun Kwan 大)各氏との共同研究)

# データ同化例2 神戸空港・地盤沈下



地盤変形モデル

データ同化

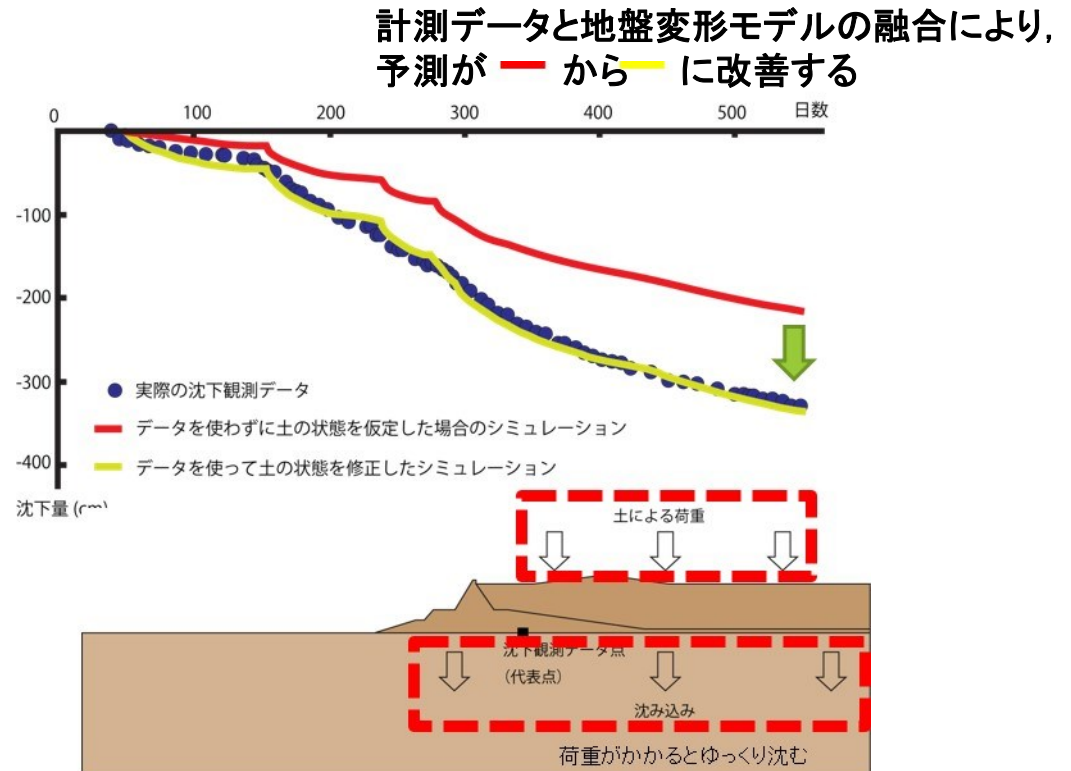
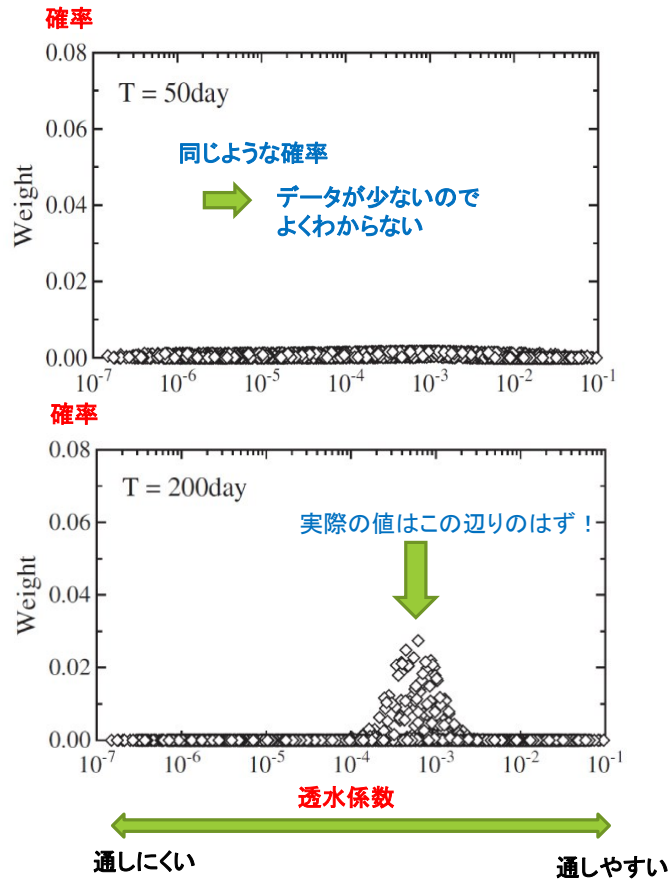


沈下量データ

- ・直接見ることができない地中の土の状態がわかる
- ・予測精度の向上で、中途での工法変更が可能に

(村上・藤澤(京大), 珠玖・西村(岡大)各氏との共同研究)

# データ同化例2 神戸空港・地盤沈下

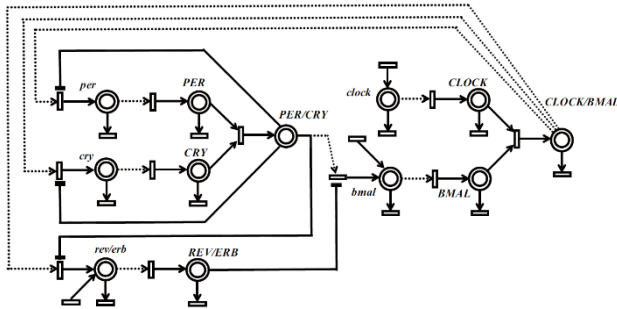


A. Murakami et al.,  
Int. J. Numer. Anal. Methods Geomech. (2012)

# データ同化例3 遺伝子ネットワークモデル

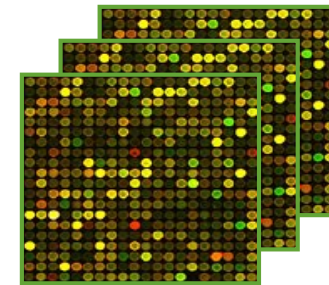
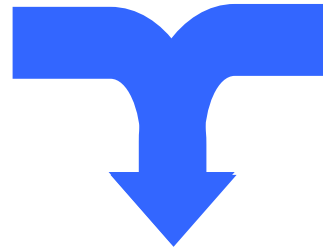
**Simulation model**

**Biological data**



現実の系を表すには不完全  
未知パラメータ

データ同化



ノイズ, 欠測など

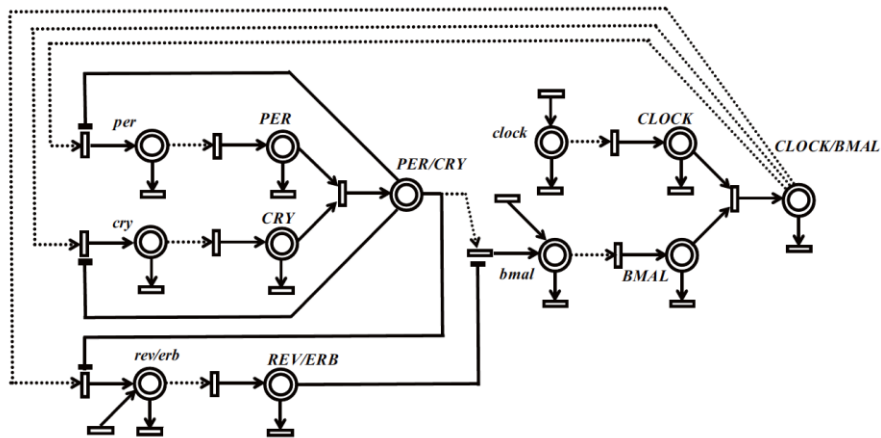
生体プロセスの予測  
生体システムに関する新たな知見

(長崎(東北大), 宮野(東大), 吉田, 樋口(統数研)各氏との共同研究)



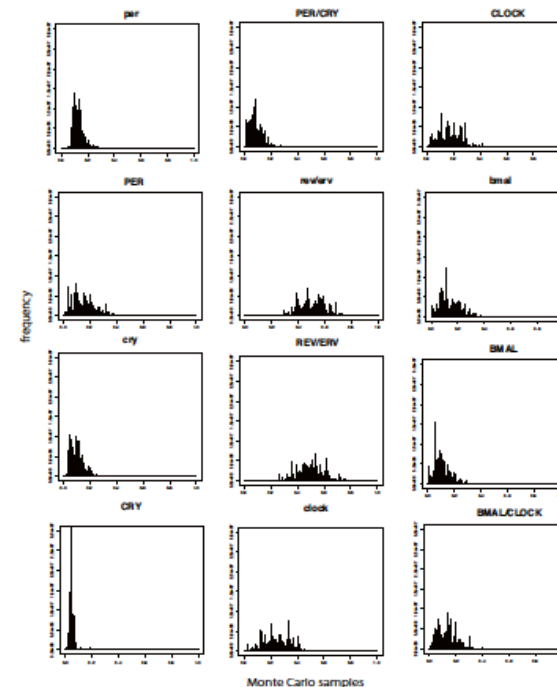
# データ同化例3

## 遺伝子ネットワークモデル



Hybrid Functional Petri Net  
によって表現されたシミュレーションモデル  
(次元は低いが非線形性が強い)

初期状態の推定結果



パラメータの分布を推定できる



予測精度が上がるだけでなく、興味ある  
事象が起こる確率を適切に評価できる

# データ同化と状態 空間モデリング

---

# 数値シミュレーションモデル

- ✓ 基礎となる偏微分方程式の離散化等により構成
- ✓ 基礎ダイナミクスから現実を再現することを目的とする
- ✓ シミュレーションコード(極端な場合, ライブラリ)の形でのみアクセス可能な場合がある

偏微分方程式(物理を反映, 連続時空間)

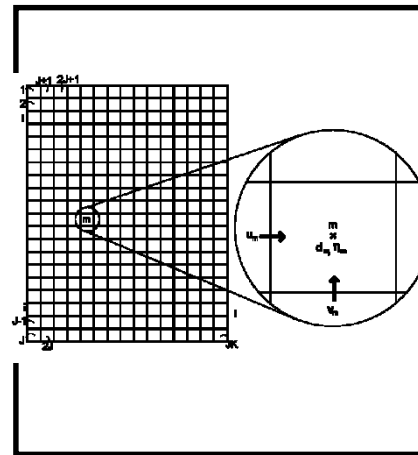
$$\frac{\partial T_o}{\partial t} = -u_{ol} \frac{\partial \bar{T}_o + T_o}{\partial x} - v_{ol} \frac{\partial \bar{T}_o + T_o}{\partial y} - \bar{u}_{ol} \frac{\partial T_o}{\partial x} - \bar{v}_{ol} \frac{\partial T_o}{\partial y} - [M(\bar{w}_{os} + w_{os}) - M(\bar{w}_{os})] \frac{\partial \bar{T}_o}{\partial z} - M(\bar{w}_{os} + w_{os}) \frac{T_o - T_e}{H} - \alpha_s T_o$$

$$\frac{\partial u_o}{\partial t} - \beta_0 y v_o = -g' \frac{\partial h}{\partial x} + \frac{\tau^x}{\rho_o H} - r u_o$$

$$\beta_0 y u_o = -g' \frac{\partial h}{\partial y} + \frac{\tau^y}{\rho_o H} - r v_o$$

$$\frac{\partial h}{\partial t} + H \left( \frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y} \right) = -rh$$

時間・空間  
離散



コーディング



```

...
calp[i*jm+j] = CENS*(2.0*pvar[(i+1)*jm+j+1] + pvar[i*jm+j+1]
+2.0*pvar[(i+1)*jm+j]);
cdlt[i*jm+j] = CENS*(pvar[(i+1)*jm+j+1] + 2.0*pvar[i*jm+j+1]
+pvar[(i+1)*jm+j] + 2.0*pvar[(i+1)*jm+j]);
ceps[i*jm+j] = CENS*(pvar[(i+1)*jm+j+1] + pvar[i*jm+j+1]
-pvar[(i+1)*jm+j] - pvar[(i+1)*jm+j]);
cphi[i*jm+j] = CENS*(CENS*(pvar[(i+1)*jm+j+1] + pvar[i*jm+j+1]
+pvar[i*jm+j] - pvar[(i+1)*jm+j]);
}

for(i=1;i<im;i++){
for(j=0;j<jm-1;j++){
cbet[i*jm+j] = CENS*(pvar[i*jm+j+1] + 2.0*pvar[(i-1)*jm+j+1]
+pvar[(i-1)*jm+j] + 2.0*pvar[i*jm+j]);
cgam[i*jm+j] = CENS*(2.0*pvar[i*jm+j+1] + pvar[(i-1)*jm+j+1]
+2.0*pvar[(i-1)*jm+j] + pvar[i*jm+j]);
}
}

for(i=1;i<im;i++){
for(j=0;j<jm;j++){
ust[i*jm+j] = 0.5*dy*(hnp[i*jm+j]+hnp[(i-1)*jm+j])*up[i*jm+j];
}
}

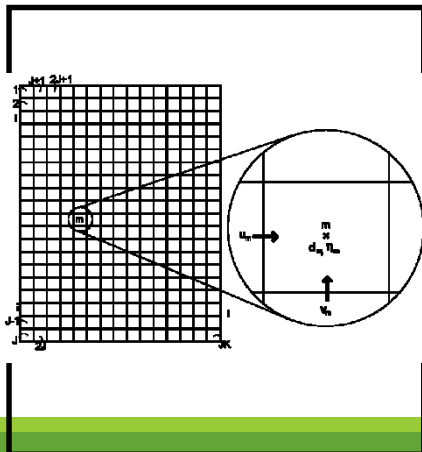
for(i=0;i<im;i++){

```

# シミュレーションモデルとシステムモデル

- ✓ シミュレーションモデルの「誤差」、初期・境界条件などによる状態の誤差が反映されていない
- ✓ このような誤差まで含めたモデルとして、システムモデルを定式化
- ✓  $x_t$  を状態ベクトル,  $v_t$  をシステムノイズと呼ぶ

シミュレーションモデル(離散時間・空間)



コーディング

```

calp[i*jm+j] = CENS*(2.0*pvor[(i+1)*jm+j+1] + pvor[i*jm+j+1]
+ 2.0*pvor[i*jm+j] + pvor[(i+1)*jm+j]);
cdlt[i*jm+j] = CENS*(pvor[(i+1)*jm+j+1] + 2.0*pvor[i*jm+j+1]
+ pvor[i*jm+j] + 2.0*pvor[(i+1)*jm+j]);
ceps[i*jm+j] = CENS*(pvor[(i+1)*jm+j+1] + pvor[i*jm+j+1]
- pvor[i*jm+j] - pvor[(i+1)*jm+j]);
cphi[i*jm+j] = CENS*(-pvor[(i+1)*jm+j+1] + pvor[i*jm+j+1]
+ pvor[i*jm+j] - pvor[(i+1)*jm+j]);

for(i=1;i<=m;i++){
for(j=0;j<=m-1;j++){
cbet[i*jm+j] = CENS*(pvor[i*jm+j+1] + 2.0*pvor[(i-1)*jm+j+1]
+ pvor[(i-1)*jm+j] + 2.0*pvor[i*jm+j]);
cgam[i*jm+j] = CENS*(2.0*pvor[i*jm+j+1] + pvor[(i-1)*jm+j+1]
+ 2.0*pvor[(i-1)*jm+j] + pvor[i*jm+j]);
}
}

for(i=1;i<=m;i++){
for(j=0;j<=m-1;j++){
ust[i*jm+j] = 0.5*dy*(chhp[i*jm+j]+hhp[(i-1)*jm+j])*up[i*jm+j];
}
}

for(i=0;i<=m;i++){

```

形式的にこのように書ける:

$$x_t = f_t(x_{t-1})$$

全シミュレーション変数

「誤差」も含める:

$$x_t = f_t(x_{t-1}, v_t)$$

モデル化誤差など

# 方程式からシステムモデルへ

(日本周辺の簡易化した気象モデルの例を用いて説明)

実システムに対応した偏微分方程式  
(連続時間・空間)

$$\frac{\partial x}{\partial t} = cx^2 + \dots$$

数値シミュレーションモデル  
(離散時間・空間, 有限差分方程式)

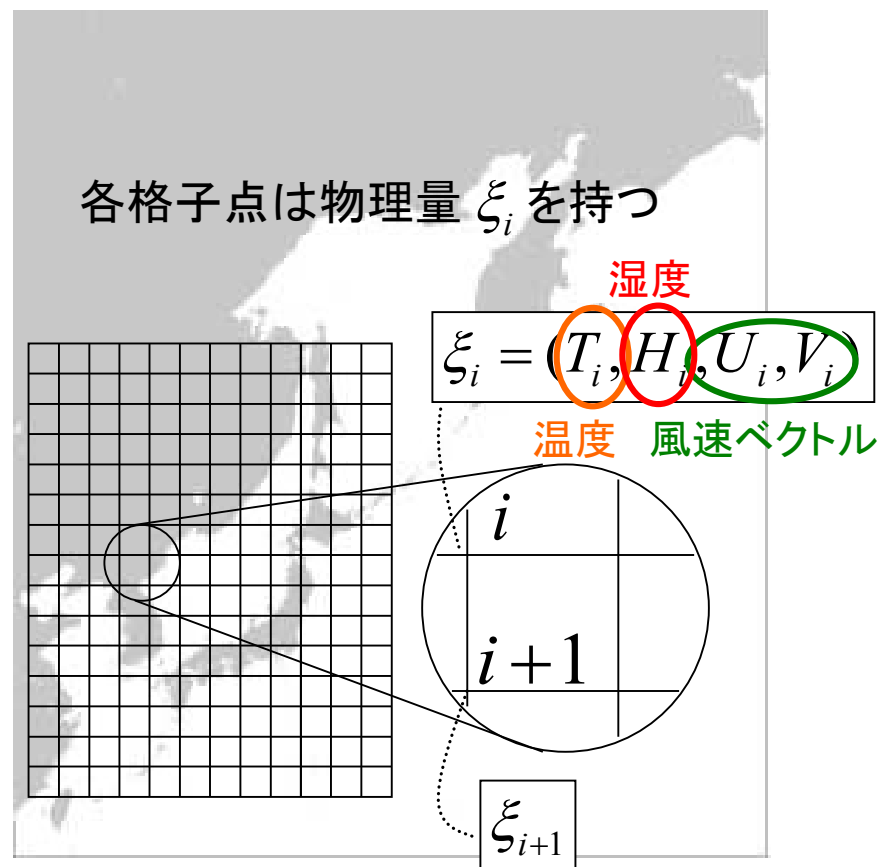
$$x_t = f_t(x_{t-1})$$

境界条件・モデル化誤差  
由来の不確かさ

$v_t$

非線形状態空間表現のシステムモデル  
(離散時間・空間, 確率差分方程式)

$$x_t = f_t(x_{t-1}, v_t)$$



$$x_t = [\xi_1, \xi_2, \dots, \xi_k]^T$$

$k$  は格子点数

# 観測情報と観測モデル

- ✓ ほとんどの場合，観測情報はシミュレーションの情報に比べて圧倒的に不足．ダイナミクスを伴う逆問題．
- ✓ さらに，時点間で独立な「観測ノイズ」もある
- ✓ 観測情報は，「その時点の全物理変数（＝全シミュレーション変数），および「観測ノイズ」が与えられれば，説明できる」という定式化

全観測変数

観測ノイズ

$$y_t = h_t(x_t, w_t)$$


全シミュレーション変数

$$\dim(x_t) \gg \dim(y_t)$$

$$x_t \quad 10^4 \sim 10^6 \quad y_t \quad 10 \sim 10^5$$

# 両者をつなぐ鍵

## ✓ (非線形) 状態空間モデル

- シミュレーションモデルから自然に書き下すことができる
- ほとんど数値シミュレーションモデルは、マルコフ性を満たすか、満たすように変形できる
- 逐次ベイズ更新の式により、のオンライン推定(観測を得る毎の推定)が可能(=逐次データ同化)

全シミュレーション変数

モデル化誤差など

$$\{x_t\} = f_t(\{x_{t-1}\}, \{v_t\})$$

$$\{y_t\} = h_t(\{x_t\}, \{w_t\})$$

全観測変数

観測ノイズ

$$\dim(\{x_t\}) \gg \dim(\{y_t\})$$

# 非線形非ガウス状態空間モデル

非線形非ガウス状態空間モデル:

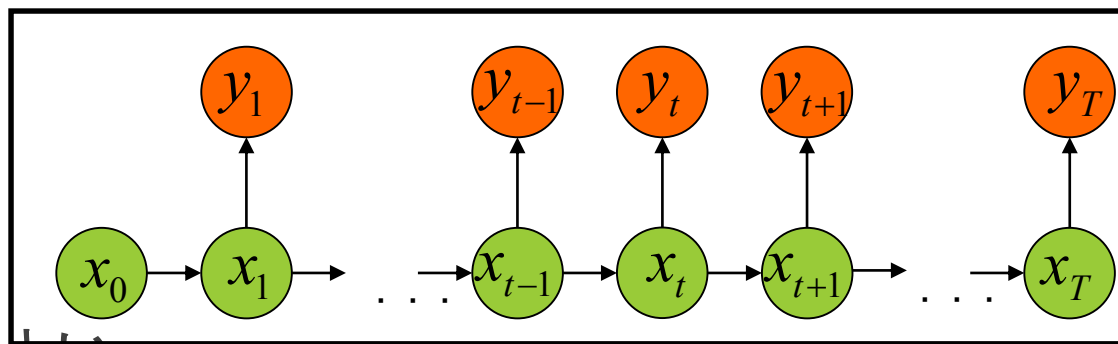
(システムモデル)  $x_t = f_t(x_{t-1}, v_t)$

(観測モデル)  $y_t = h_t(x_t, w_t)$

$$\begin{cases} x_t = f_t(x_{t-1}) + v_t \\ y_t = h_t(x_t) + w_t \end{cases}$$

もこのクラスに含まれる

- $x_t$  状態ベクトル
- $y_t$  観測ベクトル
- $v_t$ : システムノイズ
- $w_t$ : 観測ノイズ
- $v_t, w_t$  は任意の分布でよい

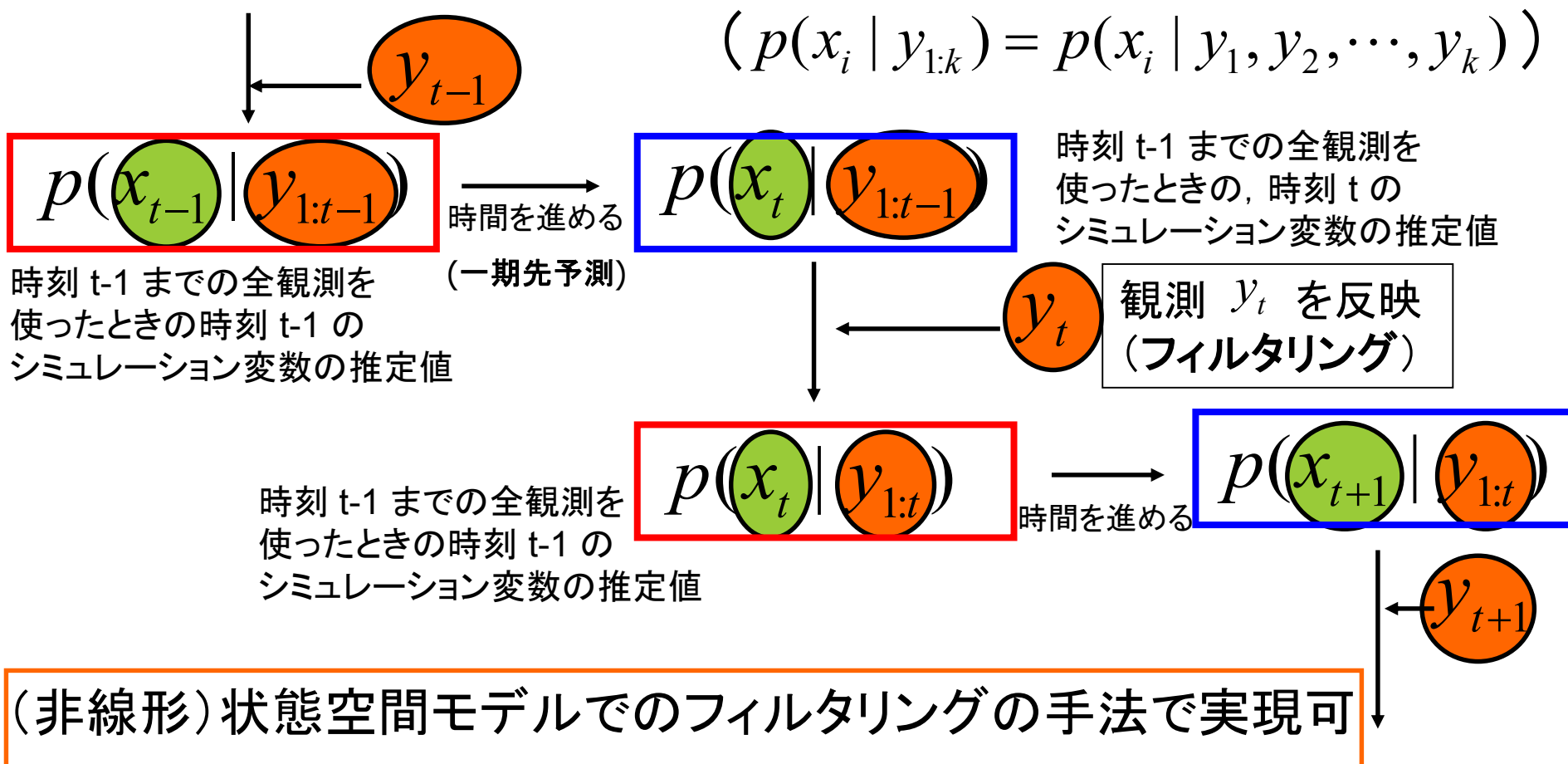


アンサンブルカルマンフィルタ, 粒子フィルタ etc. により,  
フィルタ分布の計算が原理的には可能



# 逐次データ同化

逐次データ同化では一期先予測とフィルタリングを繰り返して、観測 $y_t$ を得る毎にシミュレーション変数  $x_t$  の値(分布)をオンライン推定する



# ベイズ更新

---

# 少しわき道：ベイズの定理の問題

---

$P(A|C)=0.95$ ,  $P(A^c|C^c)=0.95$ ,  $P(C)=0.005$  のとき,  $P(C|A)$ の確率を求めよ.

(例えば,  $A/A^c$ はある病気の検査結果の陽性/陰性,  $C/C^c$ は実際に病気/病気でないを表す)

# 確率はどのくらいでしょうか？

ベイズの定理

$$p(X | Y) = \frac{p(Y | X)p(X)}{p(Y)}$$

$$( \ p(Y) = \sum_{S \in \Omega} p(Y | S)p(S) \ )$$

$$p(C | A) = \frac{p(A | C)P(C)}{\sum_{S \in \Omega} P(A | S)P(S)}$$

$$= \frac{p(A | C)P(C)}{P(A | C)P(C) + P(A | C^c)P(C^c)}$$

# どうして確率が低い？

---

$P(A|C)=0.95$ ,  $P(A^c|C^c)=0.95$ ,  $P(C)=0.005$  のとき,  $P(C|A)$  の確率を求めよ.

もともとの確率が低いから.

仮に  $P(C|A)$  を90パーセント以上にしようとすると,  
検査の精度は99.95パーセント以上にしないといけない

(例えば,  $A/A^c$  はある病気の検査結果の陽性/陰性,  $C/C^c$  は実際に病気/病気でないを表す)

# 一方で...

---

$P(A|C)=0.95$ ,  $P(A^c|C^c)=0.95$ ,  $P(C)=0.005$  のとき,  $P(C|A)$  の確率を求めよ.

もともとの確率は0.5パーセント  
これが, 8.7パーセントになったのだから,  
Aという情報によりCの確率が更新された!

(例えば,  $A/A^c$  はある病気の検査結果の陽性/陰性,  $C/C^c$  は実際に病気/病気でないを表す)

# ベイズ更新

現象Xが発生した条件下で  
データYが得られる確率

現象Xが発生する  
「もともとの」確率

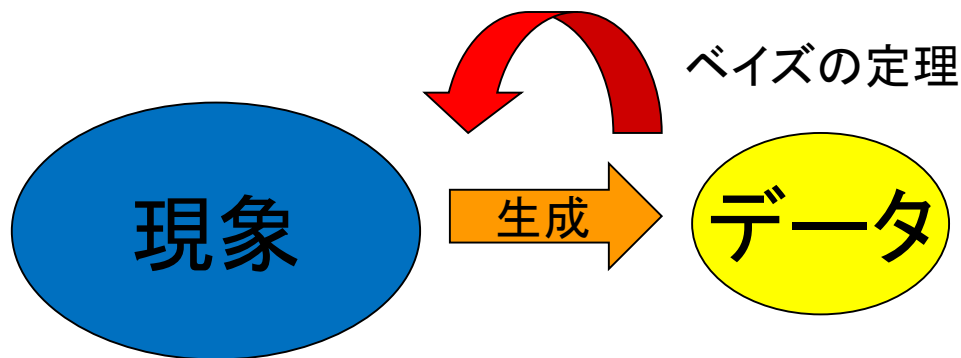
$$p(X | Y) = \frac{p(Y | X) p(X)}{p(Y)}$$

データYが得られた時に  
現象がXである確率

$p(Y)$

データYの生成確率

( $p(Y) = \sum p(Y | S) p(S)$  より,  
必要なのは  $p(Y | X)$  と  $p(X)$  . )



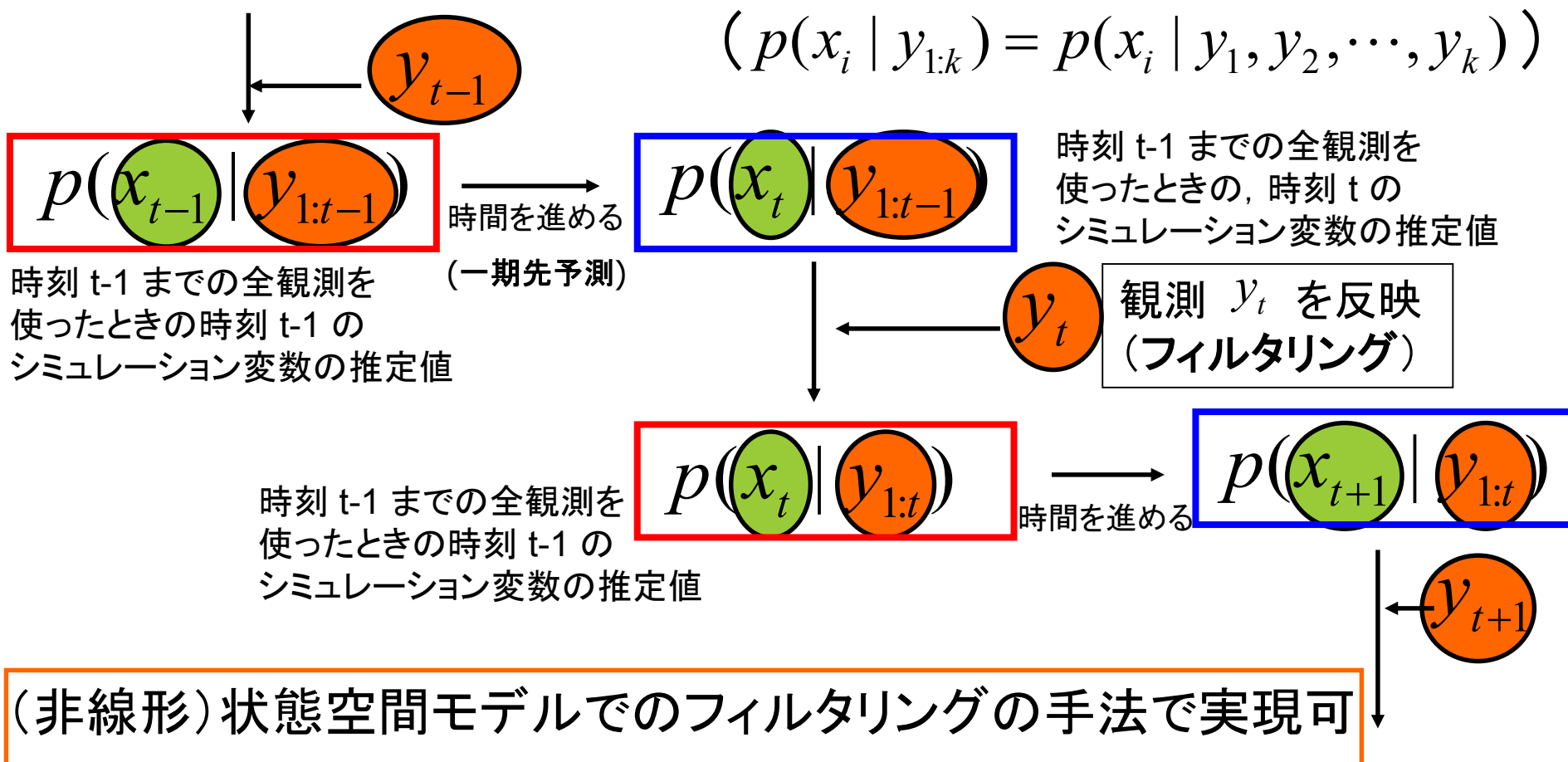
データ生成モデルと現象の発生確率を  
与えれば、データから現象の説明が  
可能！（因果の反転ができる！）

  : 事前知識や数理モデル

  : 観測を表す式

# 逐次データ同化(再掲)

逐次データ同化では一期先予測とフィルタリングを繰り返して、観測 $y_t$ を得る毎にシミュレーション変数  $x_t$  の値(分布)をオンライン推定する





# データ同化 アルゴリズム

---

# データ同化アルゴリズム一覧

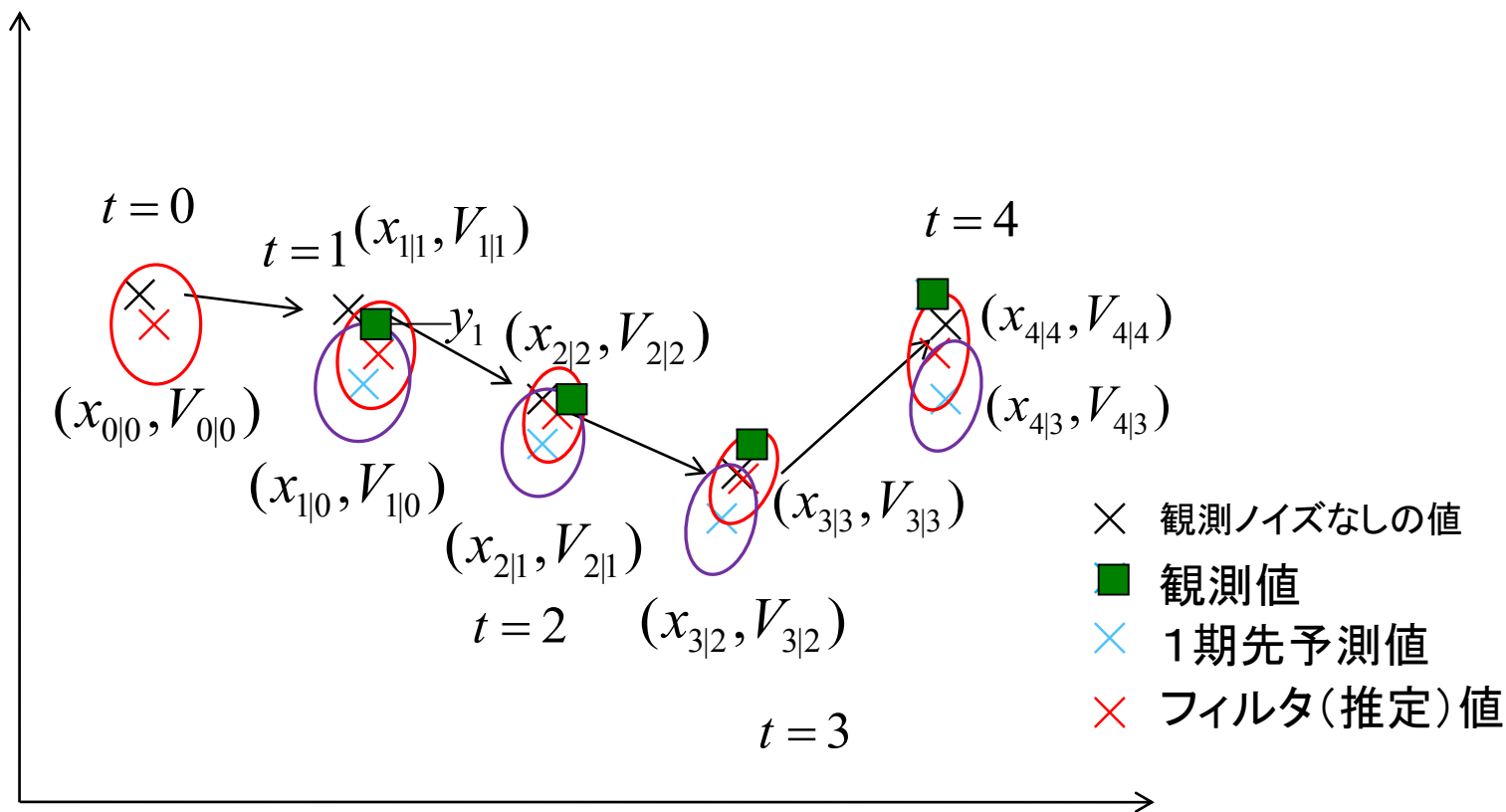
- ✓ Kalman filter
  - ✓ Extended Kalman filter
  - ✓ Ensemble Kalman filter (EnKF)
    - EAKF,ETKF,...
  - ✓ Particle filter (or SIR filter, Monte Carlo filter)
    - SIR でなく SIS filter もある
    - Merging particle filter, Kernel particle filter,...
- } 逐次型
- 
- ✓ 4DVAR      変分(非逐次)型
  - ✓ 3DVAR      1時点の補間と隠れ変数の推定のみ
  - ✓ Nudging, Ol, ...      原始的

# カルマンフィルタ

- 1960年に Kalman によって提案される
- もともとは衛星の位置の同定のために開発された
- 線形の状態空間モデルの状態推定に用いられる

$$\begin{cases} x_t = F_t x_{t-1} + G_t v_t \\ y_t = H_t x_t + w_t \end{cases}$$

# KF・2次元の場合のイメージ図



カルマンフィルタでは、「観測ノイズなし値」に近い「推定値」を得ること  
その分散(=誤差の範囲)の値も得ることが目的

# アンサンブルカルマンフィルタ

- それまでの拡張カルマンフィルタの欠点である線形化モデル構築(=微分計算)の必要性や, 分散共分散行列の推定が不安定である点を克服するために導入
- 気象・海洋の分野(特に研究分野)では, 変種も含めて広く使われている
- 分布を「実現値の集合(=シナリオの集合)」で表現, 計算はカルマンフィルタ

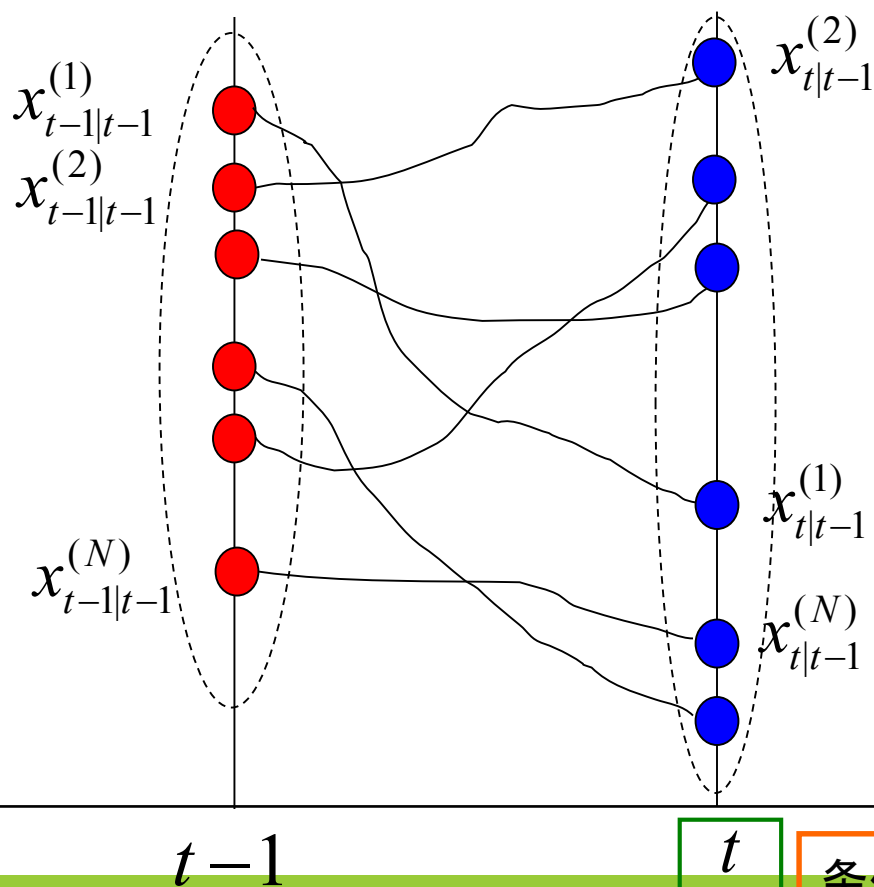
$$\begin{cases} x_t = F_t x_{t-1} + G_t v_t \\ y_t = H_t x_t + w_t \end{cases} \quad \longrightarrow \quad \begin{cases} x_t = f_t(x_{t-1}, v_t) \\ y_t = h_t(x_t) + w_t \end{cases}$$

# 一期先予測 (EnKF, PF(SIR, SIS) 共通)

$$f_t(x_{t-1|t-1}^{(i)}, v_t^{(i)}) \rightarrow x_{t|t-1}^{(i)}$$

$$\{x_{t-1|t-1}^{(i)}\}_{i=1}^N \rightarrow \{x_{t|t-1}^{(i)}\}_{i=1}^N$$

シミュレーション



- 一期先予測からのサンプル
- フィルタ分布からのサンプル

一期先予測

条件の違うシミュレーションを  
複数(Nパターン)繰り返す

# EnKFにおけるフィルタリング

状態  $x_t$

$$\{x_{t|t-1}^{(i)}\}_{i=1}^N$$

● 一期先予測からのサンプル

● フィルタ分布からのサンプル

サンプル分散共分散行列:  $\hat{V}_{t|t-1}$

観測:  $y_t$

修正しました!

$$\hat{K}_t = \hat{V}_{t|t-1} H_t' (H_t \hat{V}_{t|t-1} H_t' + \hat{R}_t)^{-1}$$

$$x_{t|t}^{(i)} = x_{t|t-1}^{(i)} + \hat{K}_t (y_t + w_t^{(i)} - H_t x_{t|t-1}^{(i)})$$

カルマンゲイン

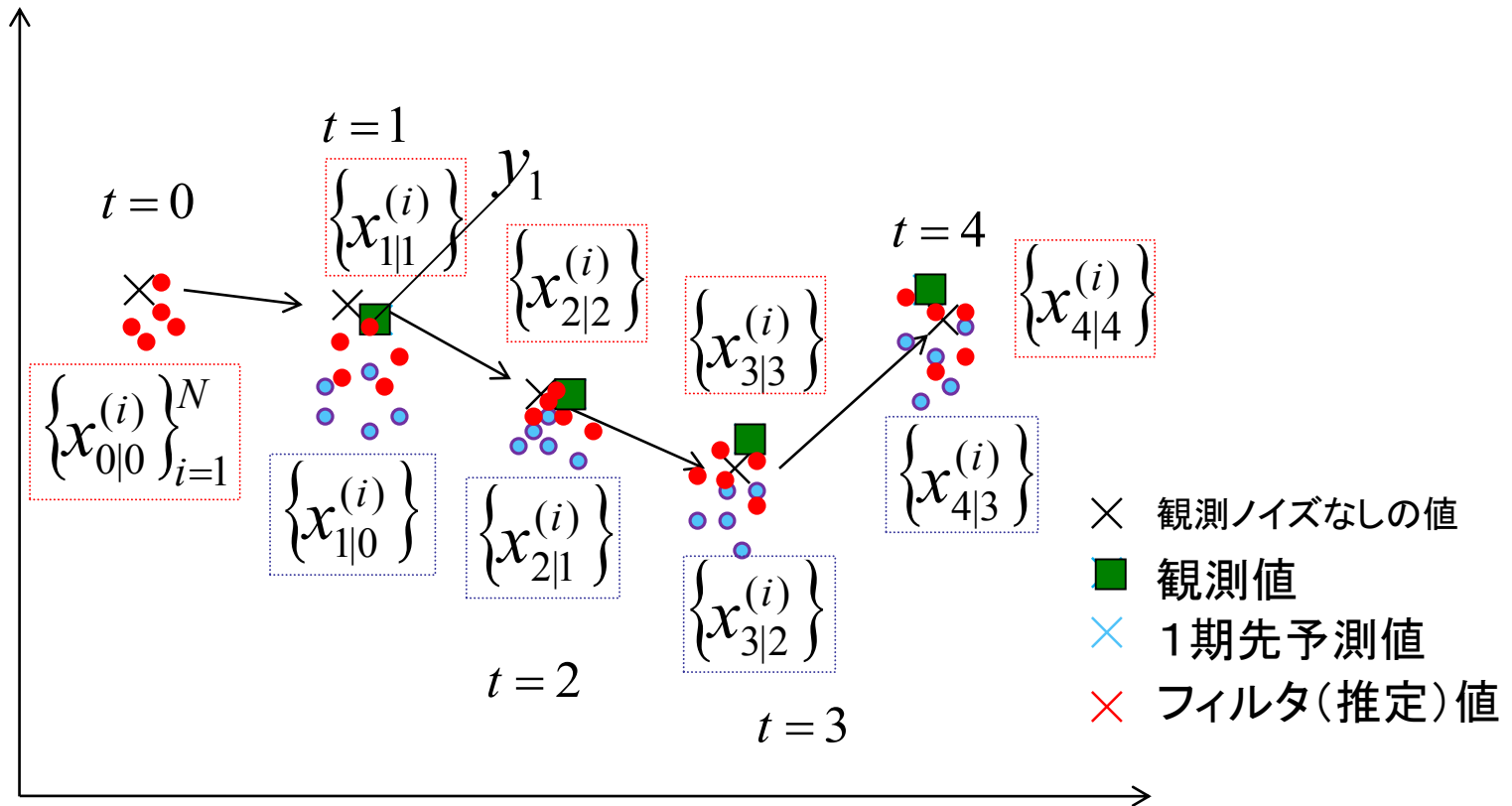
フィルタリング

時刻

$t$



# EnKF・2次元の場合のイメージ図





# 粒子フィルタ

- カメラによる物体追跡に広く使われているアルゴリズム
  - 画像処理の分野ではCondensation としても知られる
- 他に経済時系列, ロボットの状態推定などに使われる
- データ同化では, 系によるが限定的(特に気象・海洋系では)
- 任意のモデルで適用可能

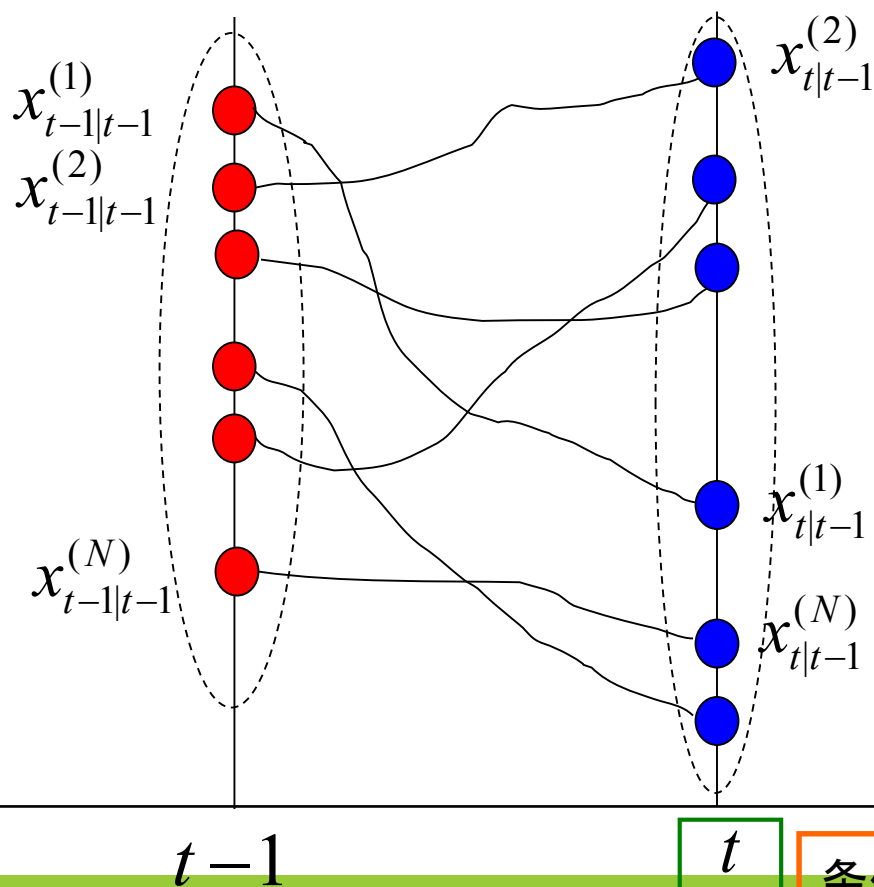
$$\begin{cases} x_t = f_t(x_{t-1}, v_t) \\ y_t = h_t(x_t) + w_t \end{cases} \longrightarrow \begin{cases} x_t \sim Q_t(\cdot | x_{t-1}) \\ y_t \sim R_t(\cdot | x_t) \end{cases}$$

# 一期先予測 (EnKF, PF(SIR, SIS) 共通)

$$f_t(x_{t-1|t-1}^{(i)}, v_t^{(i)}) \rightarrow x_{t|t-1}^{(i)}$$

$$\{x_{t-1|t-1}^{(i)}\}_{i=1}^N \rightarrow \{x_{t|t-1}^{(i)}\}_{i=1}^N$$

シミュレーション

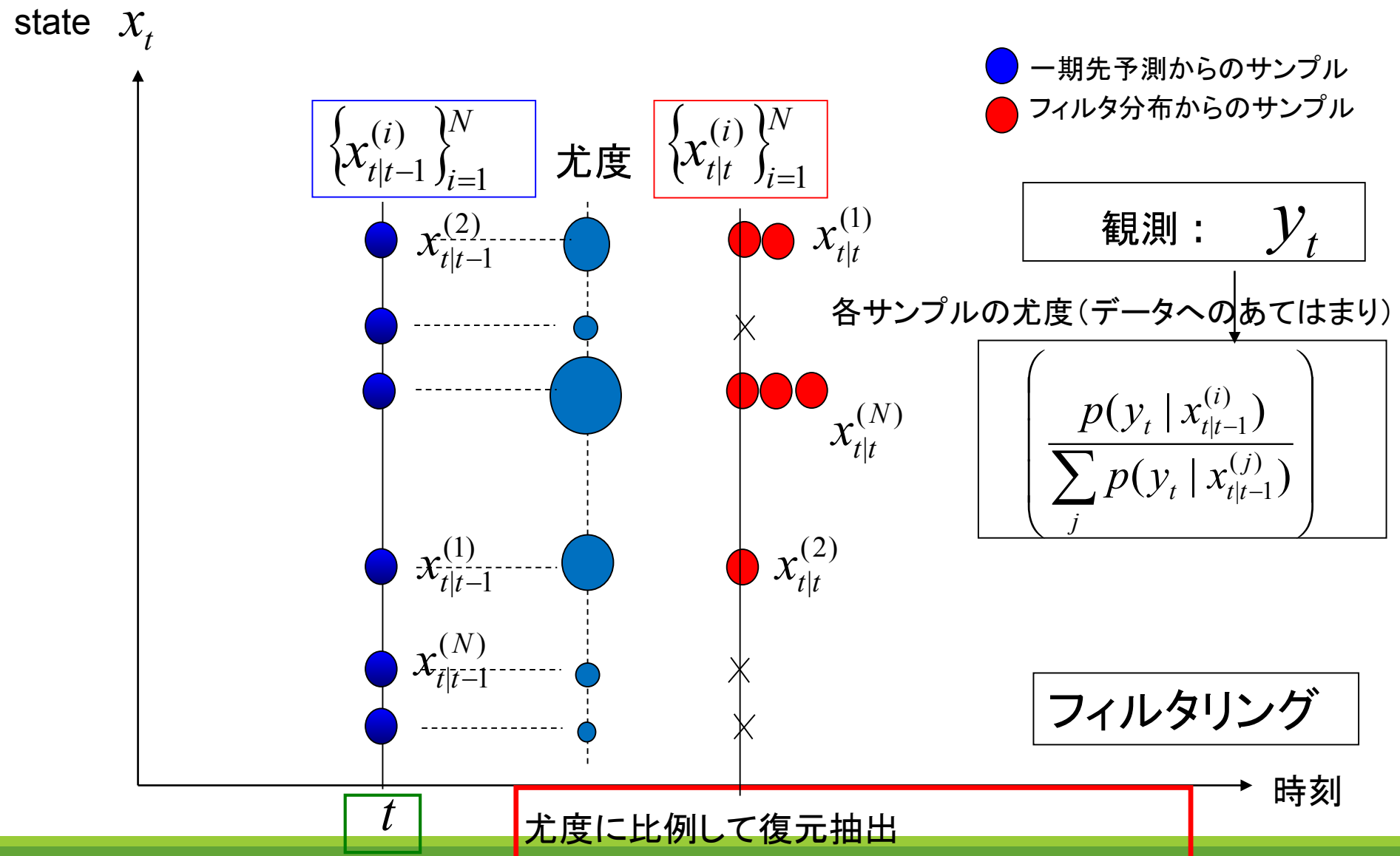


- 一期先予測からのサンプル
- フィルタ分布からのサンプル

一期先予測

条件の違うシミュレーションを  
複数(Nパターン)繰り返す

# フィルタリング(PF(SIR))



# フィルタリング(PF(SIS))

state  $x_t$

- 一期先予測からのサンプル
- フィルタ分布からのサンプル

$$\{x_{t|t-1}^{(i)}\}_{i=1}^N$$

尤度

$$\{x_{t|t}^{(i)}\}_{i=1}^N$$

観測:  $y_t$

各サンプルの尤度(データへのあてはまり)

$$\left( \frac{p(y_t | x_{t|t-1}^{(i)})}{\sum_j p(y_t | x_{t|t-1}^{(j)})} \right)$$

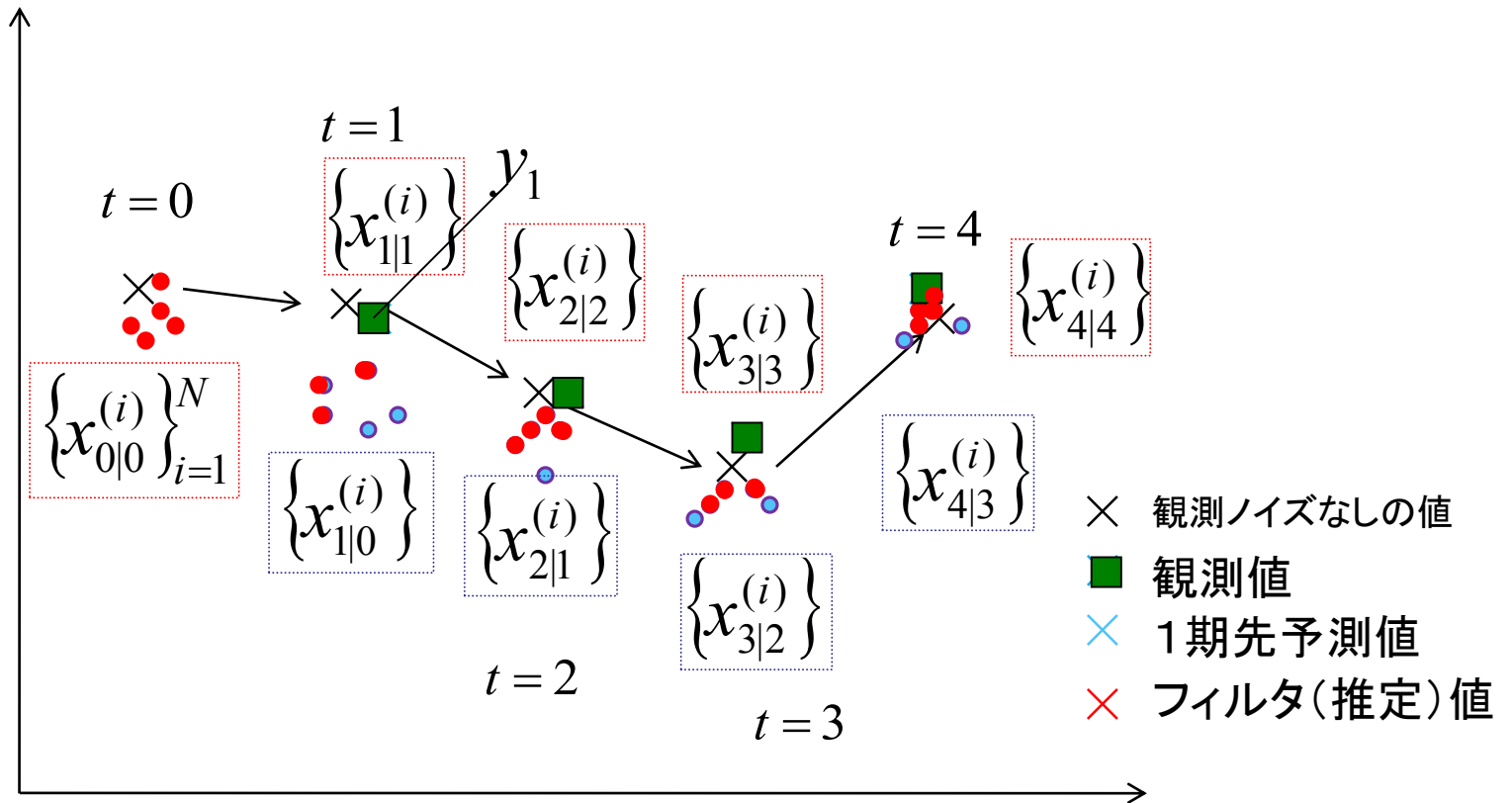
フィルタリング

$t$

各サンプルの重みを積で蓄積していく

時刻

# PF・2次元の場合のイメージ図



# 4次元変分法 (Adjoint 法)

- 1980 年代に開発
- 一定区間について、ダイナミクスを保持したまま、データとモデルから決まるコスト関数を最小化する初期値を探す方法

$$\begin{cases} x_t = f_t(x_{t-1}) \\ y_t = h_t(x_t) + w_t \end{cases}$$

# 手法間の特徴比較

	連続性	非線形性への対応	アンサンブルの効率性
Extended KF	保たれない	弱非線形のみ	N/A
EnKF	モデル次第 / 保たれない	モデル次第	✓
PF(SIR)	モデル次第	✓	状況次第
PF(SIS)	✓	✓	低い
4DVAR	✓	モデル次第	N/A

# 工学応用に向けた データ同化の位置づけ

---



# 類似手法との比較(1): 最適設計

---

## ✓ 同じところ:

- 境界条件推定とすると, 対象となる「不確かさを持つ部分」あるいは「自由度を持つ部分」は同じ

## ✓ 違うところ:

- 隠れている物理状態(特に時変の状態や4次元大浪玖薄)の推定
- 「最適値」か「確率分布」か

# 類似手法との比較(2): システム同定

---

## ✓ 同じところ:

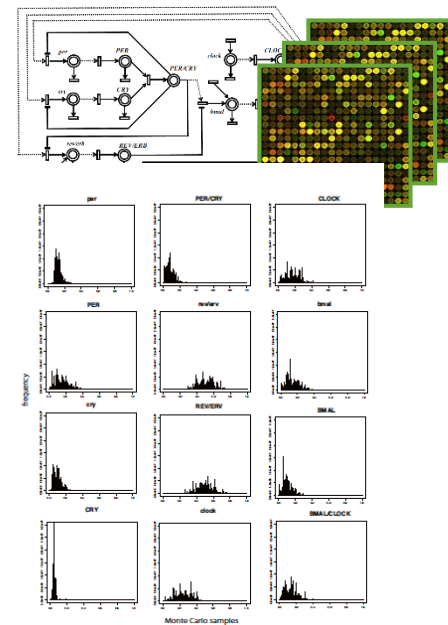
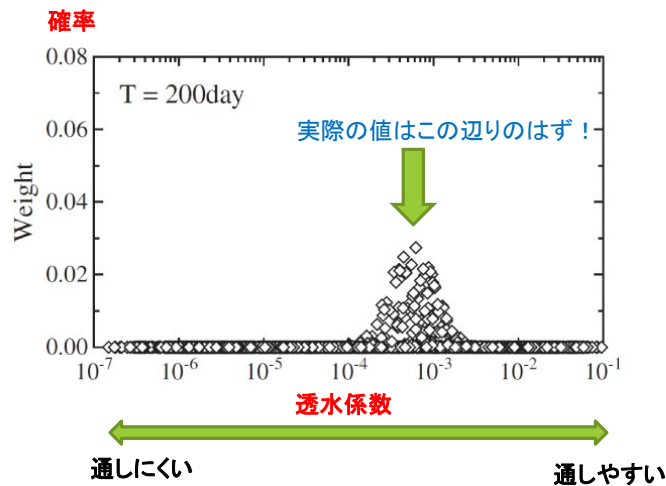
- パラメータ推定の場合には, 決める対象は同一
- 確率的なシステム同定・モデル同定の場合には, 分布で考える点も同一

## ✓ 違うところ:

- モデルや計測の想定規模(対象にもよるが)
- 中心的に想定している不確かさの対象
  - 特にモデル同定の場合にはモデルそのものの不確かさ
  - 通常のデータ同化の場合には, モデルの不確かさは小さく, 状態の不確かさが大きい

# データ同化を工学の道具とした時の「良さ」

- ✓ 推定対象の確率分布を陽に使用する
  - ロバストネスやリスクの評価に使用できる



- ✓ 計測誤差とシステム・シミュレータの誤差を陽に考える
  - 両者を定量的にバランスすることができる

# まとめ

---

# まとめ

---

## ✓ データ同化について説明

- 目的
  - 状態・パラメータ推定
  - 予測精度向上
- アルゴリズム

## ✓ 類似手法との比較

- 「計測」と「システム」の両方にノイズを定量的に想定してバランス

# さらなる発展

---

- ✓ 違うもの（観測ノイズとシステムノイズ）をバランスできているので、他のものも含めることができそう
  - 例えば「コスト」やそのバラツキもバランスできる
- ✓ CFD/EFD 融合・計測融合シミュレーションの各方法との融合
  - 数理的な整理
  - CAE ツールへの融合につながるのでは？

Email : knaka@meiji.ac.jp