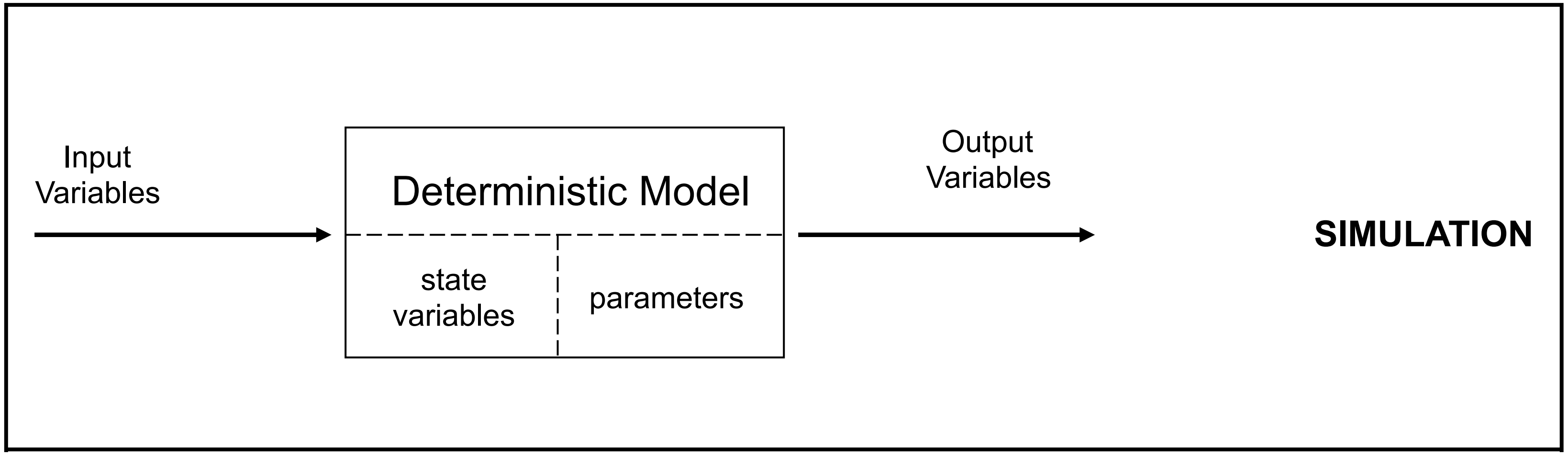
DATA ASSIMILATION

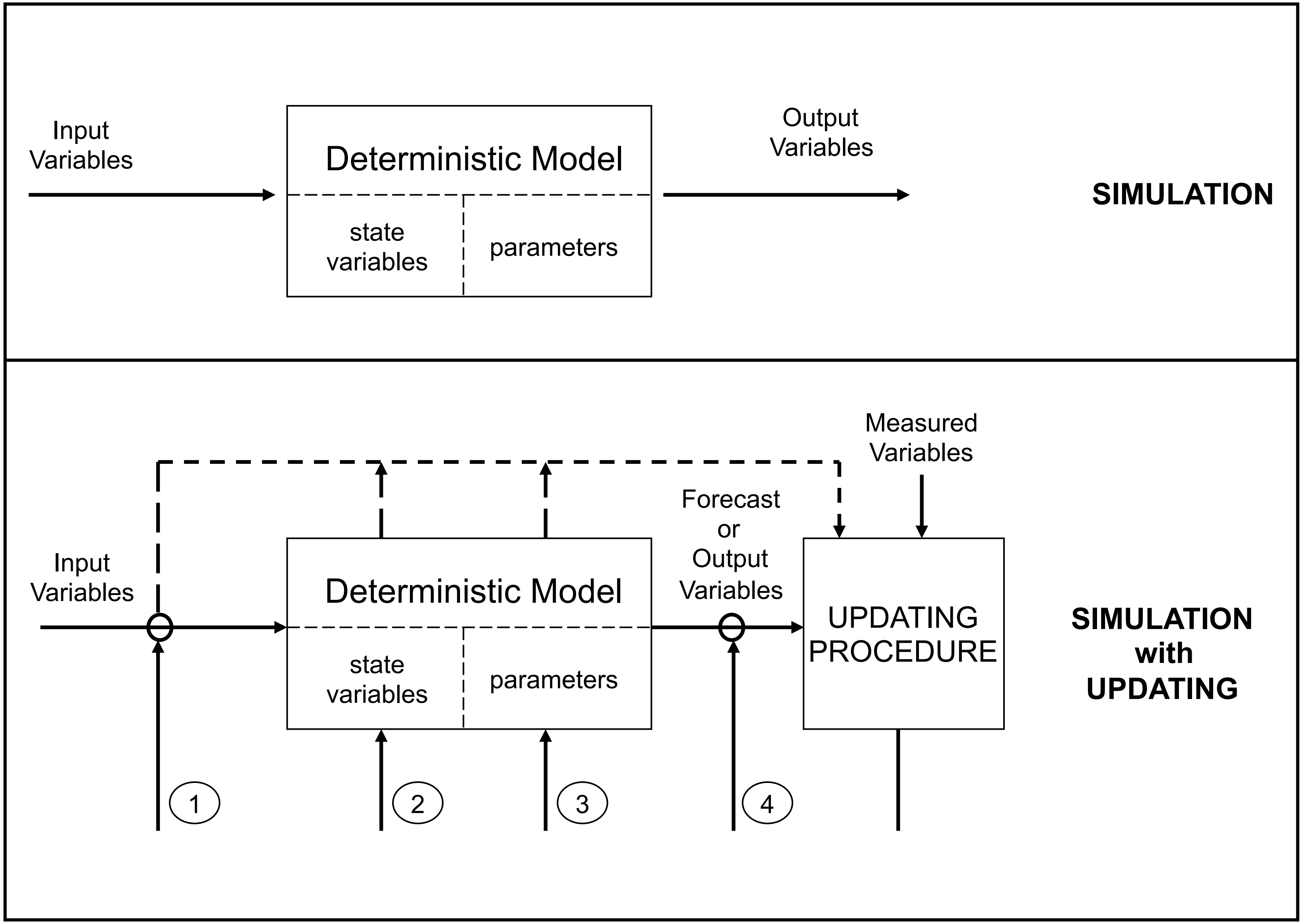
Data-Model Integration

(with focus on Error Correction)

Reminder


Input
Variables
Deterministic Model
state
variables
parameters
Output
Variables
SIMULATION

Reminder


Input
Variables
Deterministic Model
state
variables
parameters
Output
Variables
SIMULATION
Measured
Variables
Forecast
or
Output
Variables
Input
Variables
Deterministic Model
state
variables
parameters
UPDATING
PROCEDURE
SIMULATION
with
UPDATING
1
2
3
4

Outline

- Objectives
- Error Correction
- Time series forecasting
- Local modelling
- Data assimilation and error prediction in entire North Sea computational domain
- Conclusions & recommendation

Error Correction

- This can be done using very simple approaches as well as with more complex methods that can also provide an estimate of uncertainty

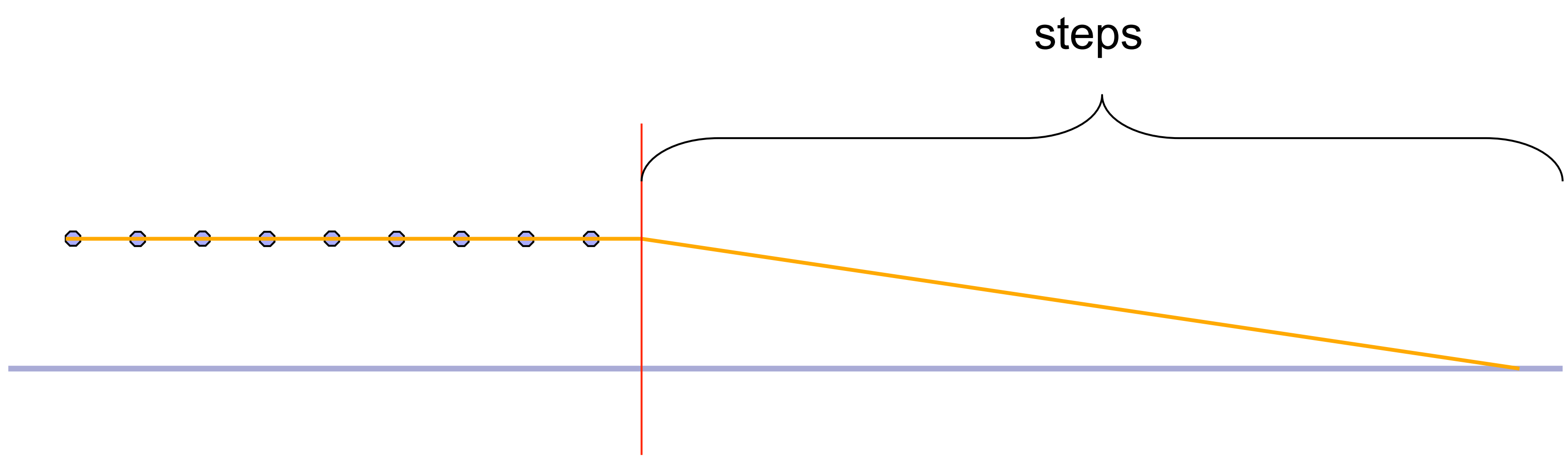
- Simple methods:
- Adjust Output (correction at start forecast)
- AR or ARMA type error correction

- More “complex” methods:
- Local Linear Models
- Neural Networks

The Simplest Idea

- ADJUST Output: Empirical error correction
  (for example ADJUST Q in case of river flow forecast)
- Parameter *steps* determines convergence speed
  - *steps* may be changed interactively during forecast

Example: simple model with constant bias



A More Sophisticated Idea

- Statistical model of error
  - Time series modeling
  - **ARMA**: **A**uto **R**egressive – **M**oving **A**verage

- Concept
  - Error is typically highly correlated in time
  - Establish model of error – predict future error
  - Correct model simulation in forecast period with predicted error

$$Q_{res}(t) = \sum_{k=1}^{K} \alpha_k \cdot Q_{res}(t-k) + \sum_{m=1}^{M} \gamma_m \cdot e(t-m) + \varepsilon$$

Model Order K, M

Model Parameters α_k, γ_m

ARMA: Autoregressive Part

- Autoregressive Moving Average Models used for forecasting <u>stationary</u> timeseries – in this case applied to modelling the time evolution of the model error

- **AR:** This part of the model describes how each observation (**error**) is a function of the previous k observations (**errors**). For example, if k = 1, then each observation is a function of only one previous observation. That is,

$$Q_{res}(t) = c + \alpha_1 \cdot Q_{res}(t-1) + \varepsilon(t)$$

where $Q_{res}(t)$ represents the observed residual (**error**) value at time t, $Q_{res}(t-1)$ represents the previous observed residual (**error**) at time t − 1, e(t) represents some random error and c and a_1 are constants. Other observed values of the series can be included in the right-hand side of the equation if k > 1:

$$Q_{res}(t) = c + \alpha_1 \cdot Q_{res}(t-1) + \alpha_2 \cdot Q_{res}(t-2).....\alpha_k \cdot Q_{res}(t-k) + \varepsilon(t)$$

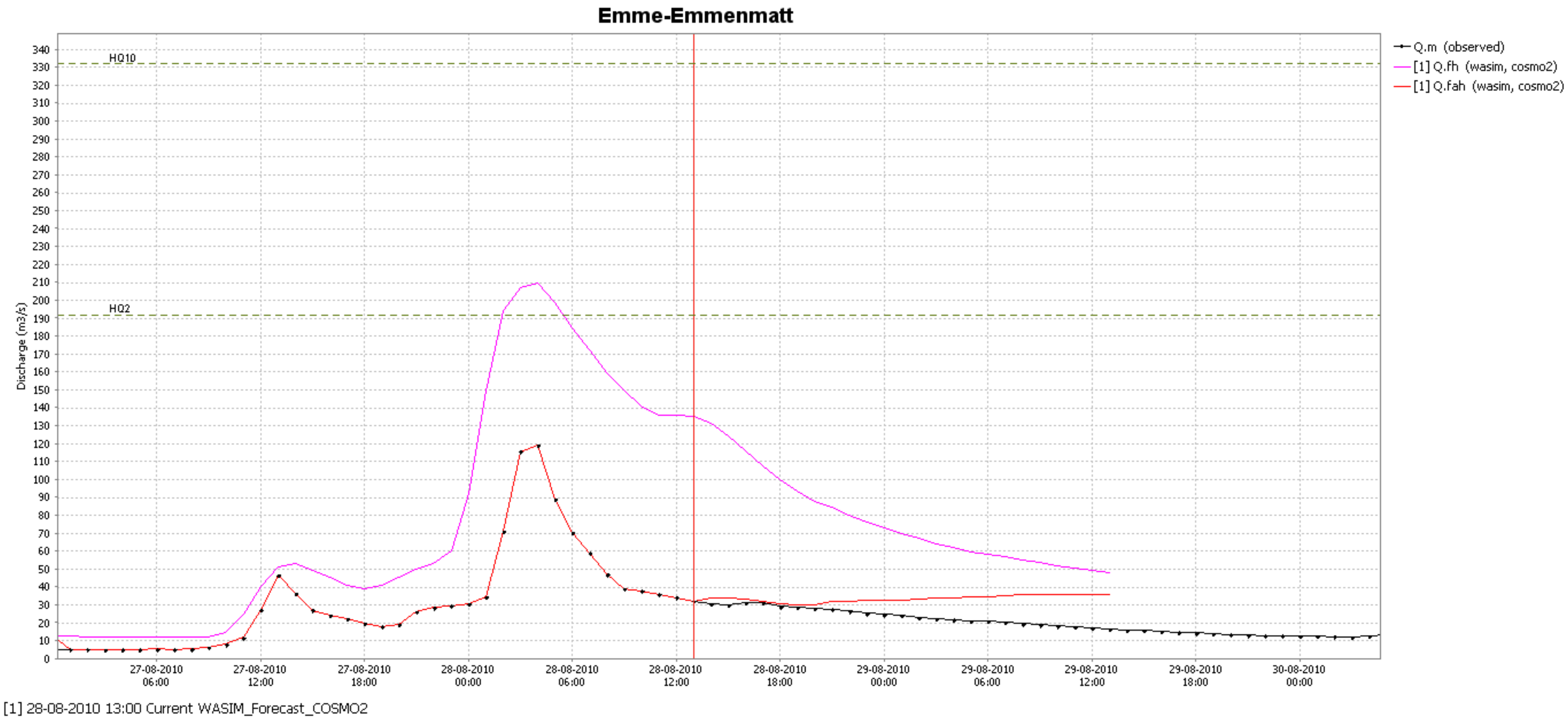
ARMA: Moving Average part - 2

MA: This part of the model describes how each observation is a function of the previous *y* errors. For example, if *y* = 1, then each observation is a function of only one previous error. That is,

$$Q_{res}(t) = c + \gamma_1 \cdot e(t-1) + \varepsilon(t)$$

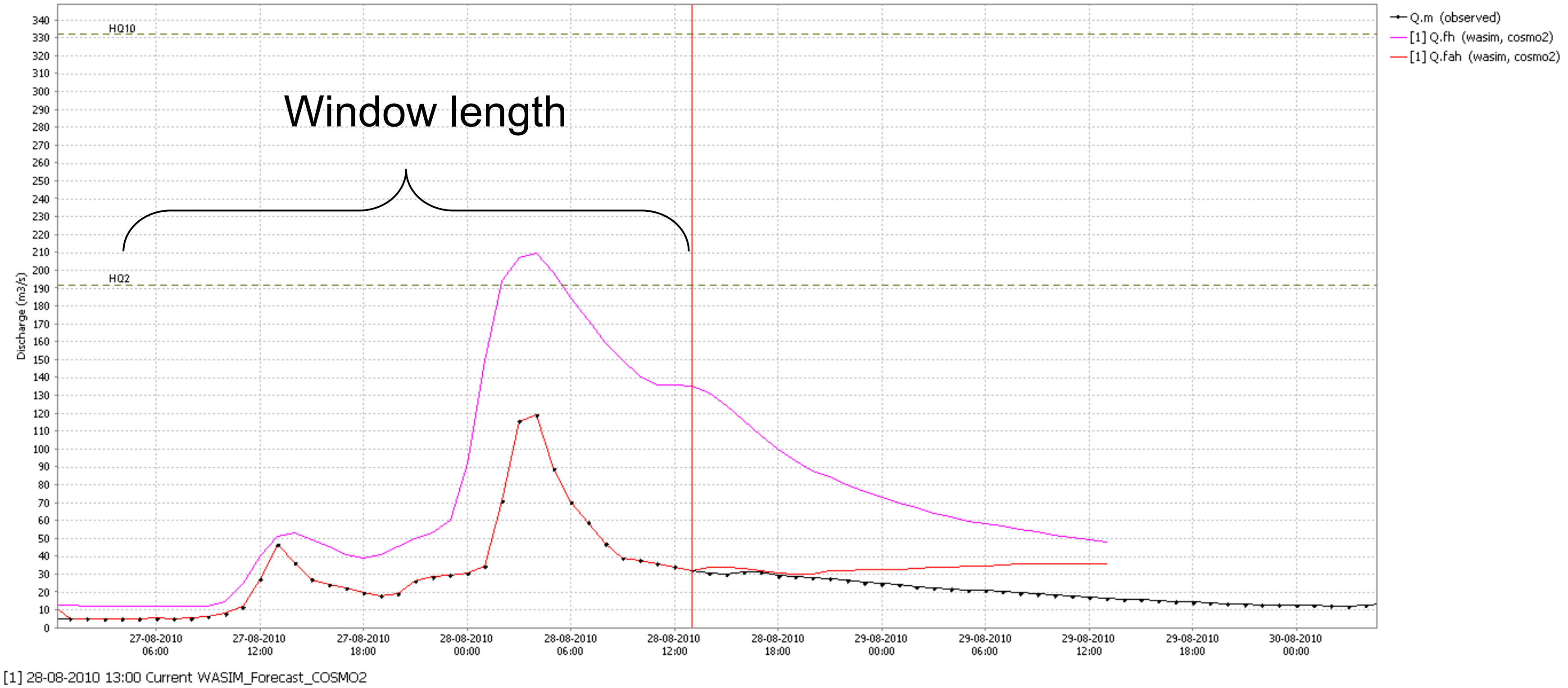
Here e(t) represents the random error at time t and e(t−1) represents the previous random error at time t − 1. Other errors can be included in the right-hand side of the equation if *y* > 1.

ARMA Model



Example of error correction using ARMA. Corrected time series (red) will converge to uncorrected time series (pink) as lead time increases

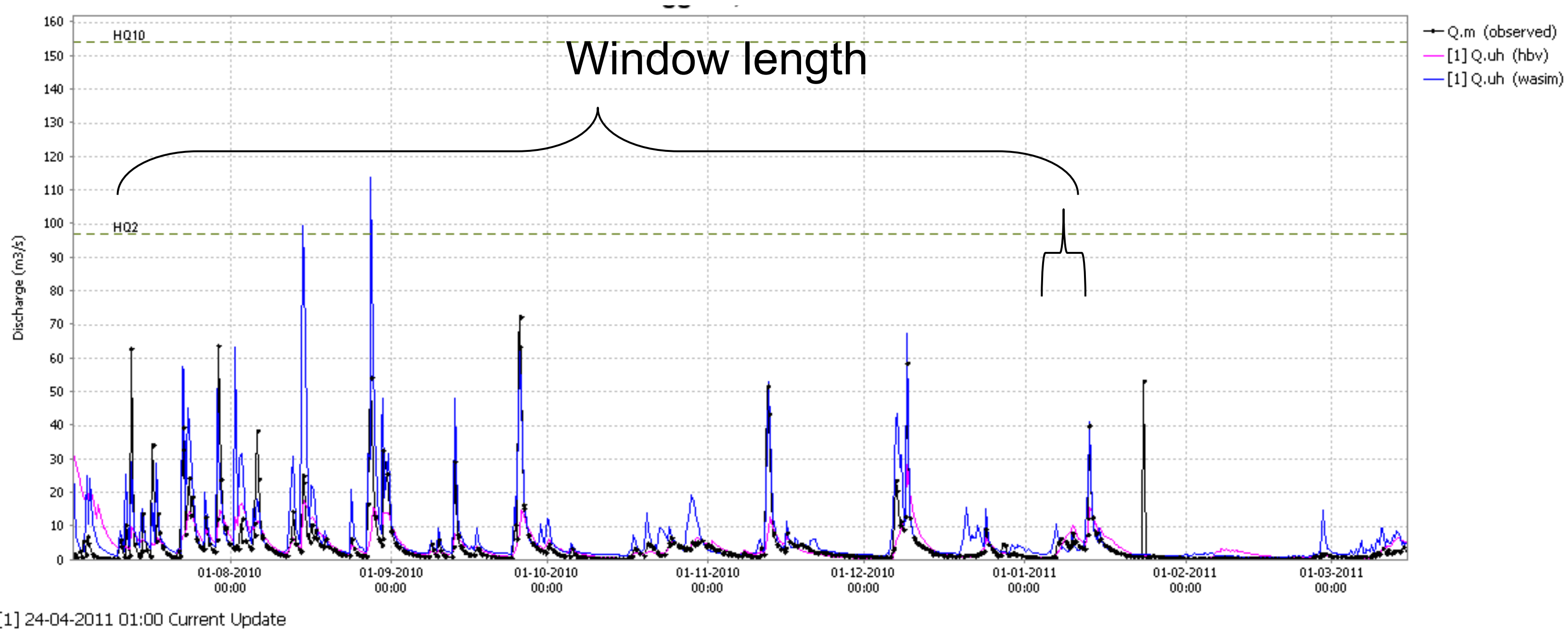
Establishing ARMA model order and parameters



Statistical behavior of error in window of defined length used to identify order and/or parameters of error model.
Rule of thumb: Window should be > 50 x order of AR model

Establishing ARMA model order and parameters

Length of window will influence the estimation of AR parameters.
As window increases autocorrelation of errors will decrease for most hydrological time series



When estimating order of model: Define maximum order
Typical AR orders vary in range 1-3

Error Correction using ARMA model

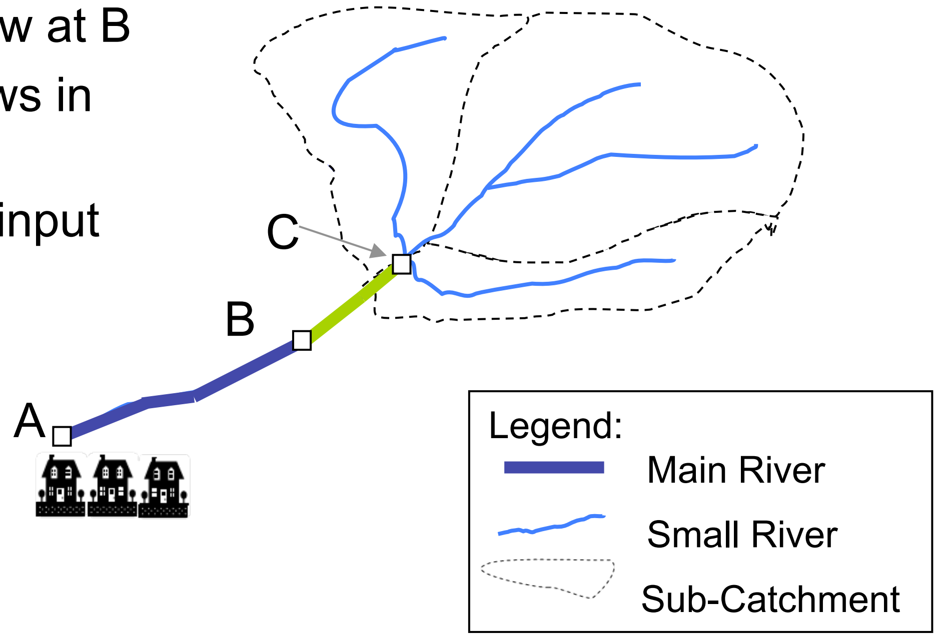
Notes on inputs to Error model

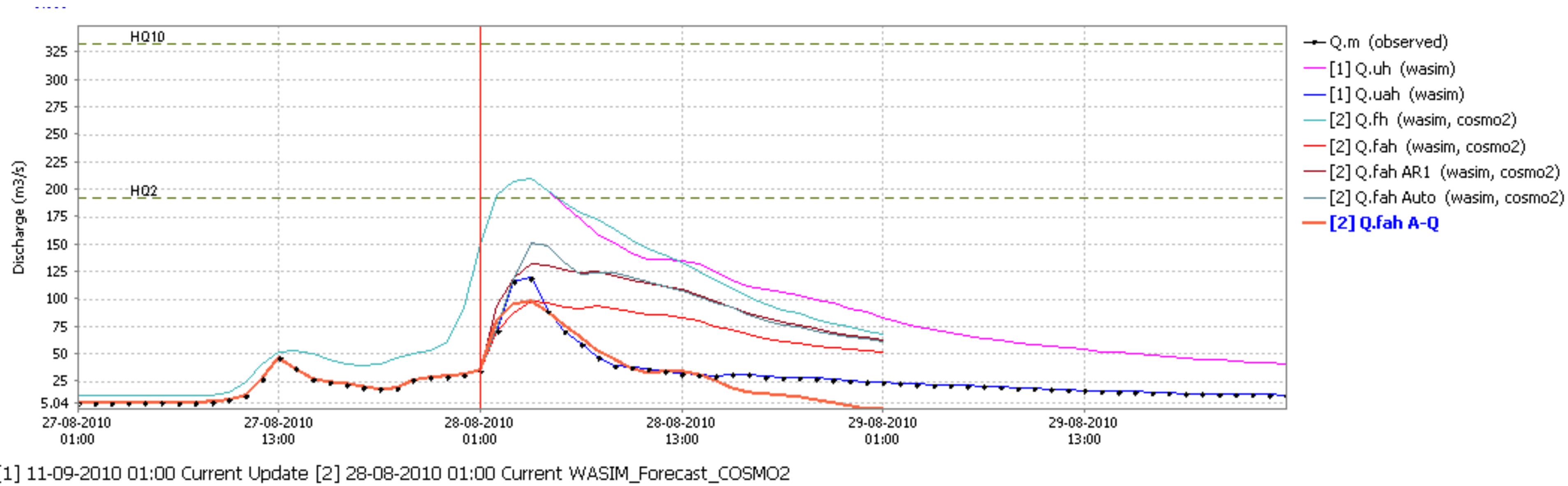
- 2 Traces are required
  - Simulated trace – shoud cover historical & forecast period
  - Observed trace – normally ends at T0
- When there is missing data in simulated time series – failure ☹
- Error correction module allows multiple simulated time series to be allocated
  - Simulated – Forecast
  - Simulated – Historical
  - *Simulated – Backup (use in case problems with cold start!)*

Application

Typical application of error model

- Rainfall-runoff model calculates flow to catchment outlet (C)
  - Error correction applied to flow at C
- Routing-model calculates propagation of flow in steep river
  - Uses error corrected flow as input
  - Error correction applied to flow at B
- HD model calculates levels & flows in reach from B to A
  - Uses error corrected flow as input





Blending steps = 120

[1] 11-09-2010 01:00 Current Update [2] 28-08-2010 13:00 Current WASIM_Forecast_COSMO2

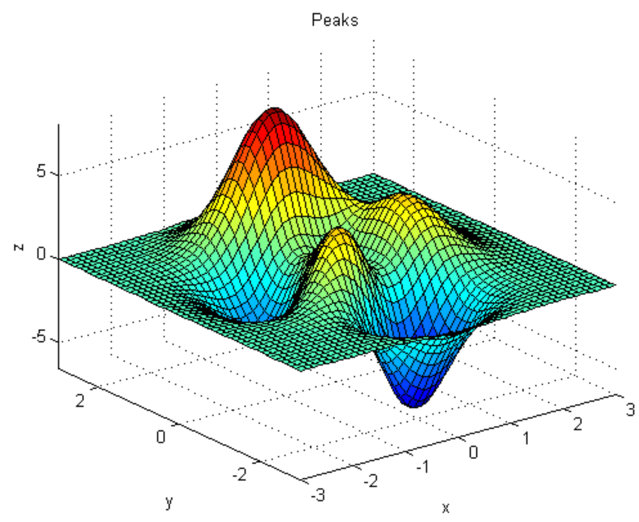
Blending steps = 120

ARMA is Linear (both AR and MA components)

ERRORS ARE NONLINEAR

Local modelling (LM)

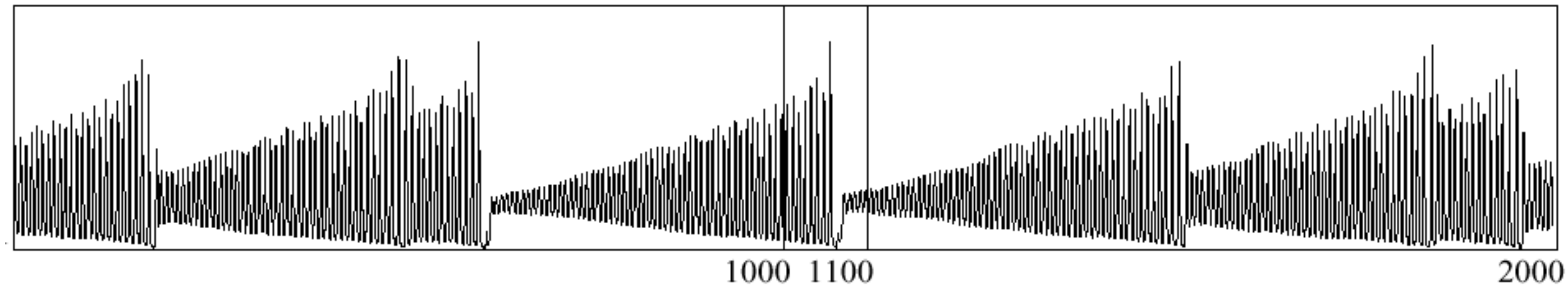
- Data-driven technique
- Use a local approximation of data points nearest in space to the query point (*i.e. a neighbourhood*)
  - averaging or linear
- The overall performance can be highly *nonlinear*
- Example: approximating a nonlinear surface with a series of linear tiles



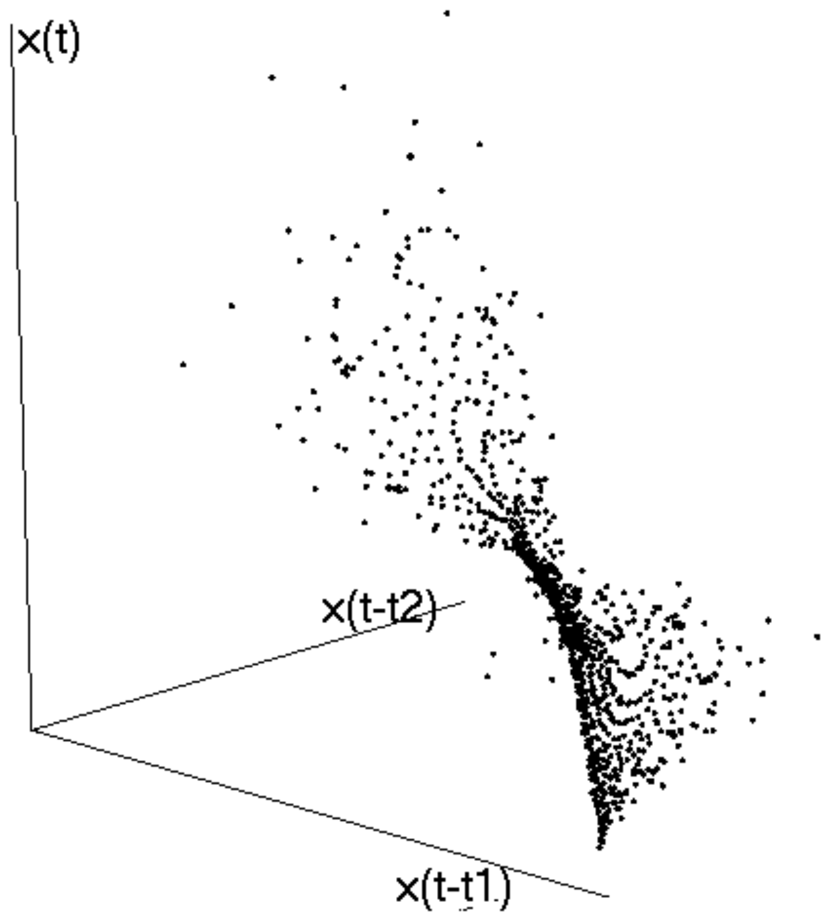
Outline

- Objectives
- Introduction to local modelling
- Time series forecasting
- Data assimilation and error prediction in a hypothetical bay
- Conclusions & recommendation

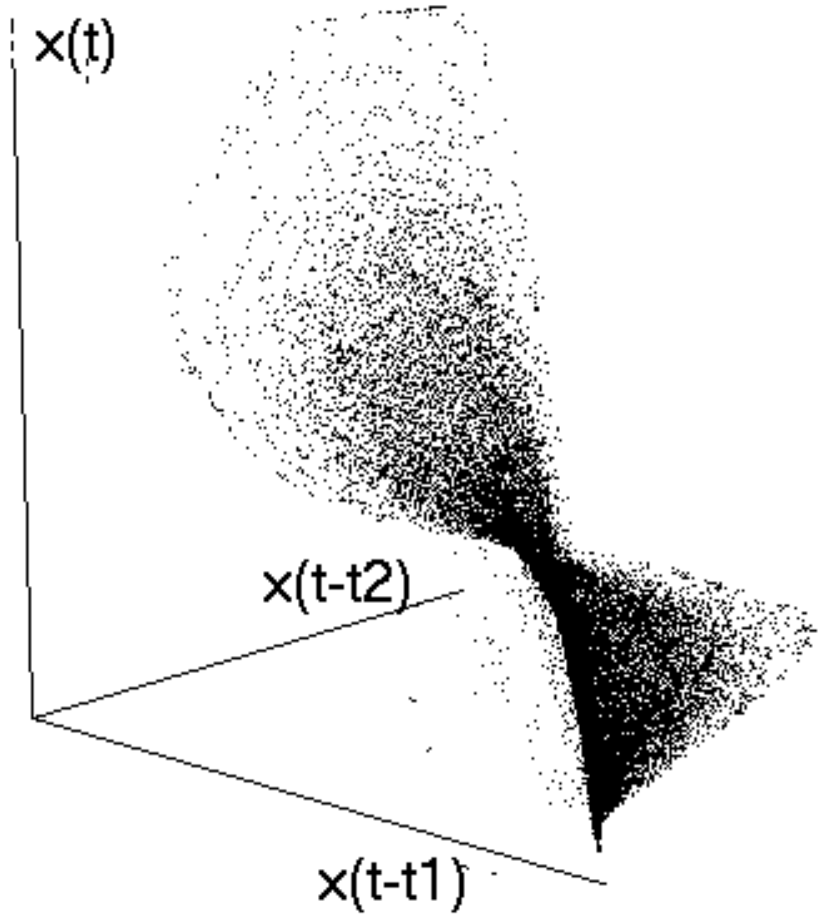
Time Domain



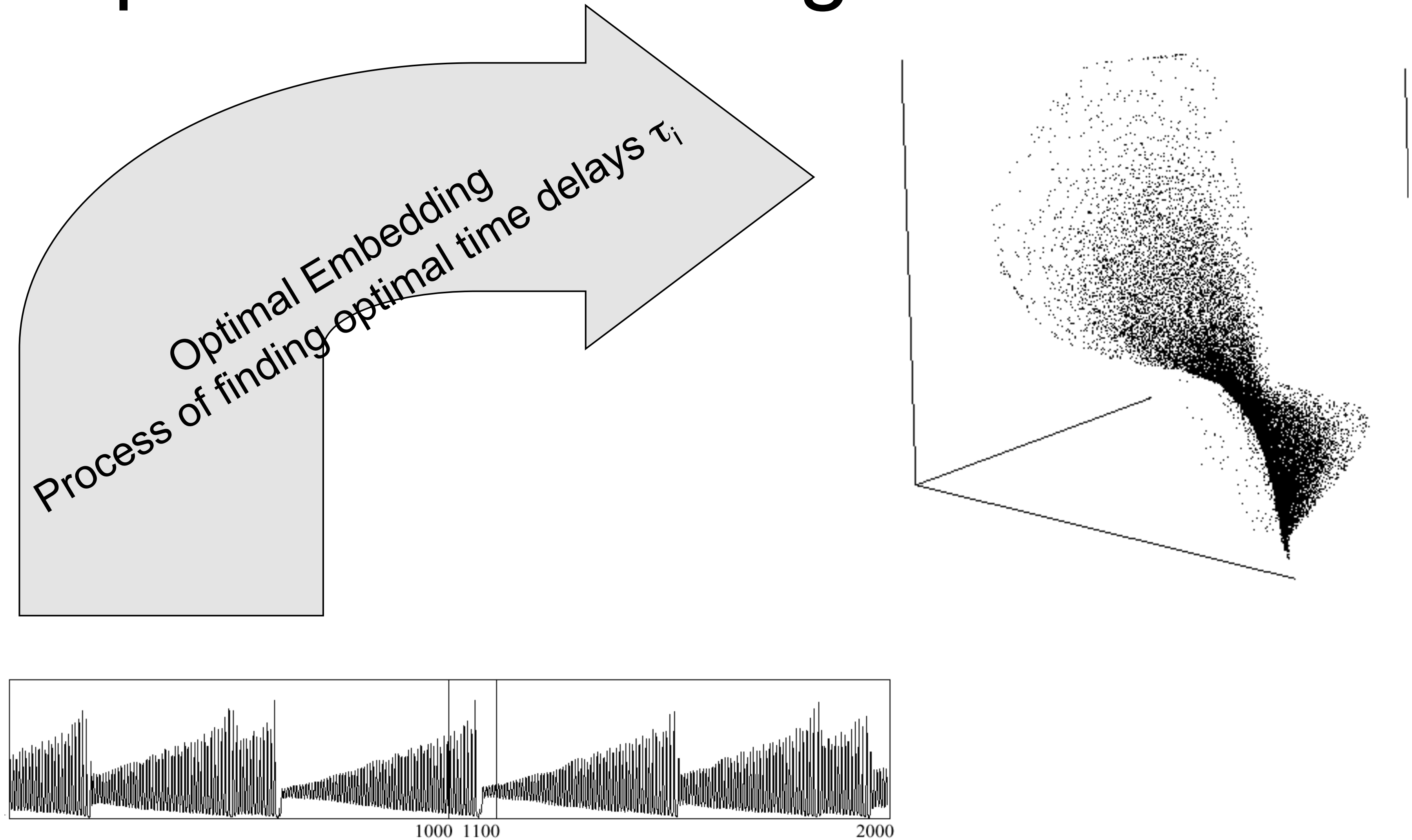
Phase Space Domain



Phase Space Domain

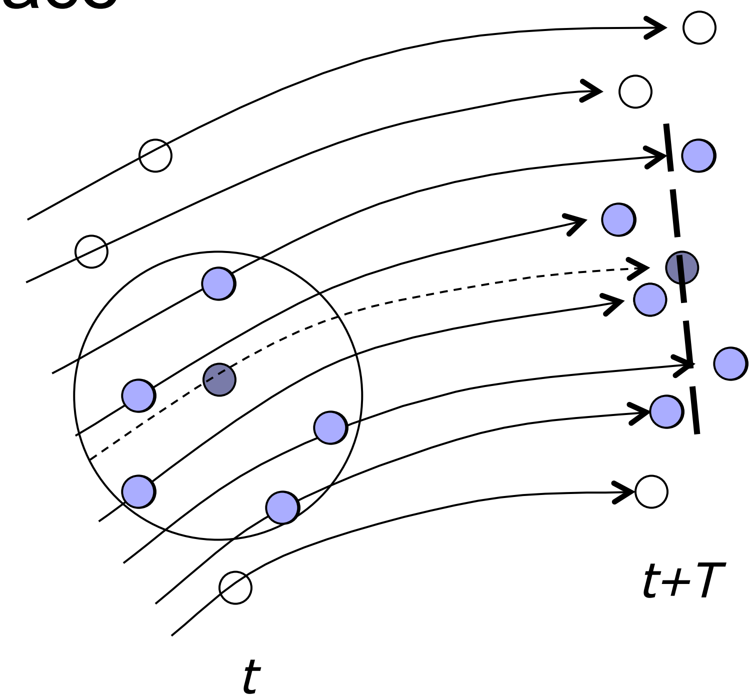


Optimal Embedding



Time series forecasting with LMs (2)

- Time series can be forecasted based on structure in their *phase* space
  - Embed time series into phase space
  - Find local neighbourhood
  - Perform regression within local neighbourhood



Determination of embedding parameters

- Need values for:
  - Time lag, τ
  - Embedding dimension, d_e
- Prescription values:
  - *Average mutual information* (AMI)
  - *False nearest neighbours* (FNNs)

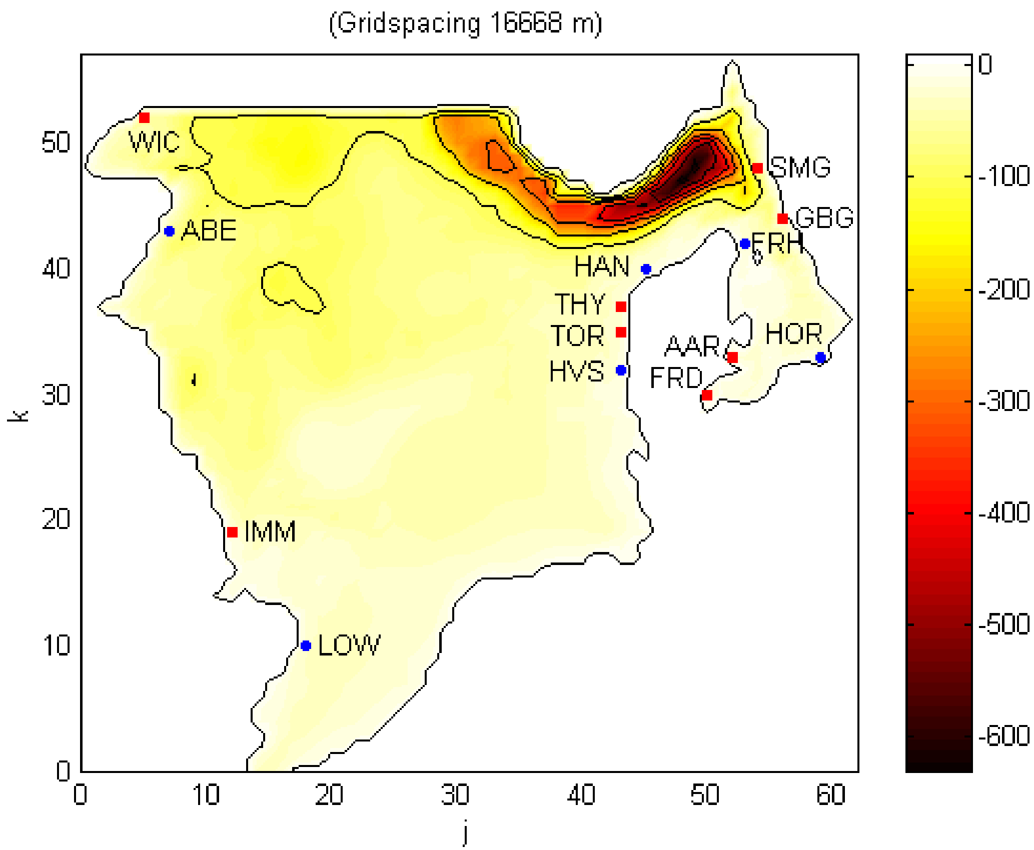
Outline

- Objectives
- Introduction to local modelling
- Time series forecasting
- Data assimilation and error prediction in entire North Sea computational domain
- Conclusions & recommendation

Spatial distribution of errors

- It is obvious that one can utilise evolutionary embedding in order to forecast model errors at observation points
- The question is whether (and how) we can spatially distribute these point error forecasts to the rest of the domain

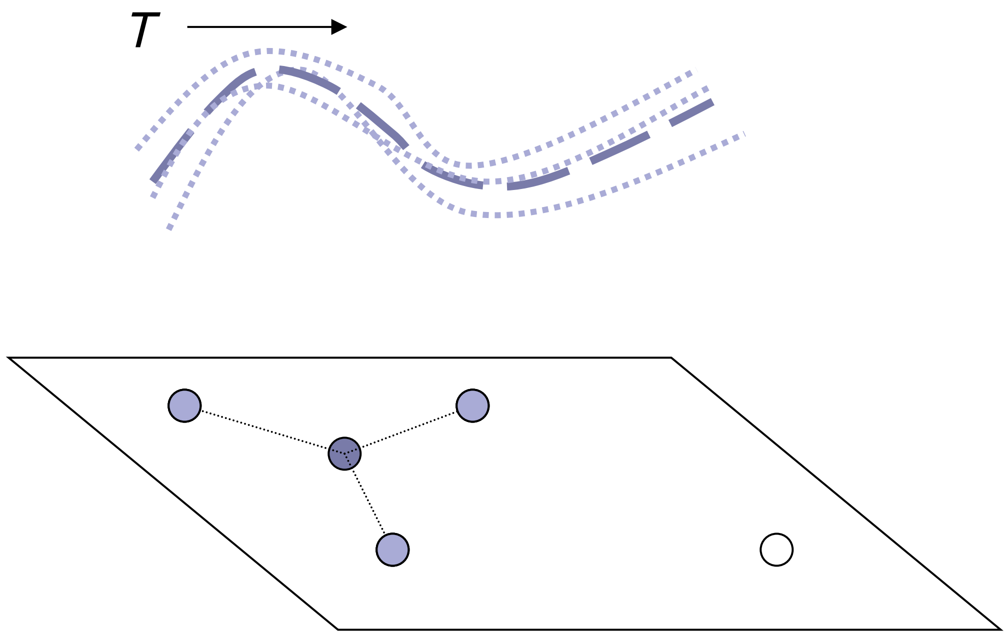
Computational Domain



- Three levels of nested bathymetries (9 nm, 3 nm and 1nm)
- Water levels at open boundaries defined as a sum of astronomical tide and atmospheric pressure correction
- Meteorological forcing (analysed wind + operational HIRLAM: 6 hours – 0.21º)
- Roughness = 32 $m^{1/3/s}$
- Time step = 10 minutes
- Red points are validation pts.

Weighting of local model ensemble

- Select local neighbourhood of measurement points
- Forecast errors using local models from each measurement point
  - Input: model results
  - Output: model errors
- Combine in a weighted ensemble fashion



Model state updating

- Sequential melding scheme

$$x_k^a = x_k^f + G_k e_k$$

x_k^a: model update
x_k^f: model forecast
e_k: error forecast
G_k: Weighting matrix

Model state updating

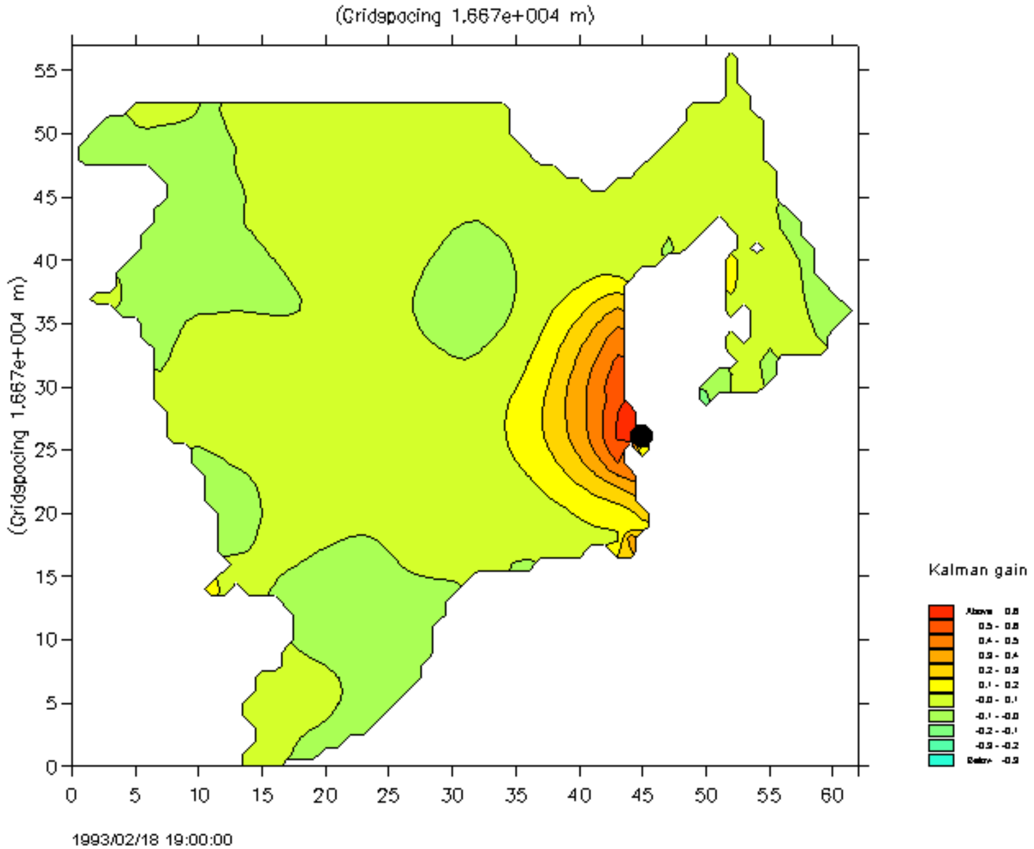
- Assuming perfect error forecast at the observation point, members of weighting matrix G_k can be given as:

$$g_j(i) = \frac{\sum_k (x_k(i) - \overline{x(i)})(e_{j,k} - \overline{e_j})}{\left[\sum_k (x_k(i) - \overline{x(i)})^2 \sum_k (e_{j,k} - \overline{e_j})^2\right]^{1/2}} \quad , \overline{x(i)} = \sum_k x_k(i) \quad , \overline{e_j} = \sum_k e_{j,k} \quad j = 1,.., p$$

- In order to take into account uncertainties, the expression becomes:

$$g_j(i) = \frac{1}{1 + \dfrac{Var\{e_j\}}{Var\{x_j\}}} \frac{\sum_k (x_k(i) - \overline{x(i)})(e_{j,k} - \overline{e_j})}{\left[\sum_k (x_k(i) - \overline{x(i)})^2 \sum_k (e_{j,k} - \overline{e_j})^2\right]^{1/2}} \quad , \quad j = 1,.., p$$

Weighting matrix



Sequential updating algorithm

```
Given: Model forecast x(i), i = 1,n
       Error forecast e_j, j = 1,p
          z_j = x_j + e_j, j = 1, p (x_j is model
          forecast corresponding to
          measurement j)
          For j = 1, p
               Update: x_j^a = x_j
               For i = 1, n
                    x(i) = x(i) + g_j(i)(z_j - x_j^a)
               End
          End
```

Given: Model forecast $x(i)$, $i = 1,n$

Error forecast e_j, $j = 1,p$

$z_j = x_j + e_j$, $j = 1, p$ (x_j is model forecast corresponding to measurement j)

For $j = 1, p$

Update: $x_j^a = x_j$

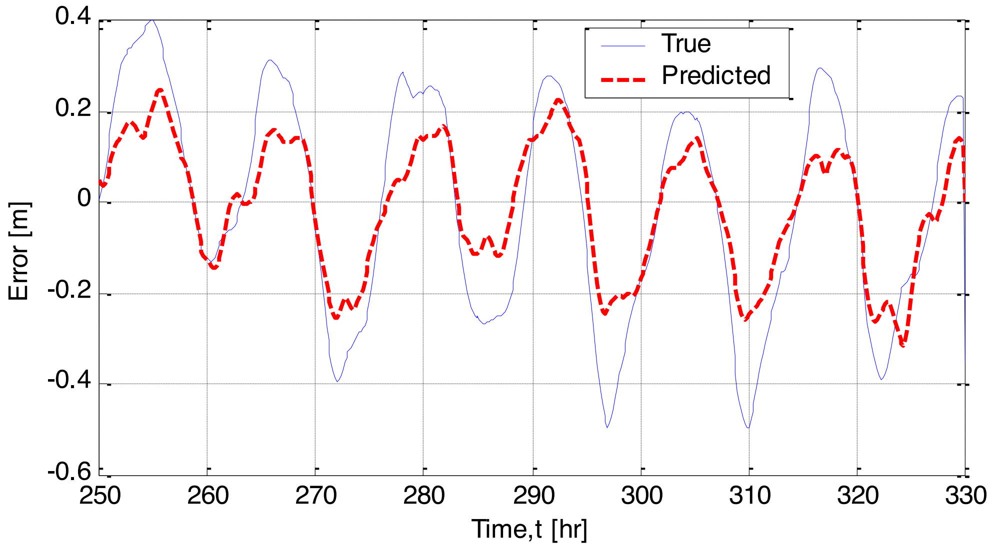
For i = 1, n

$x(i) = x(i) + g_j(i)(z_j - x_j^a)$

End

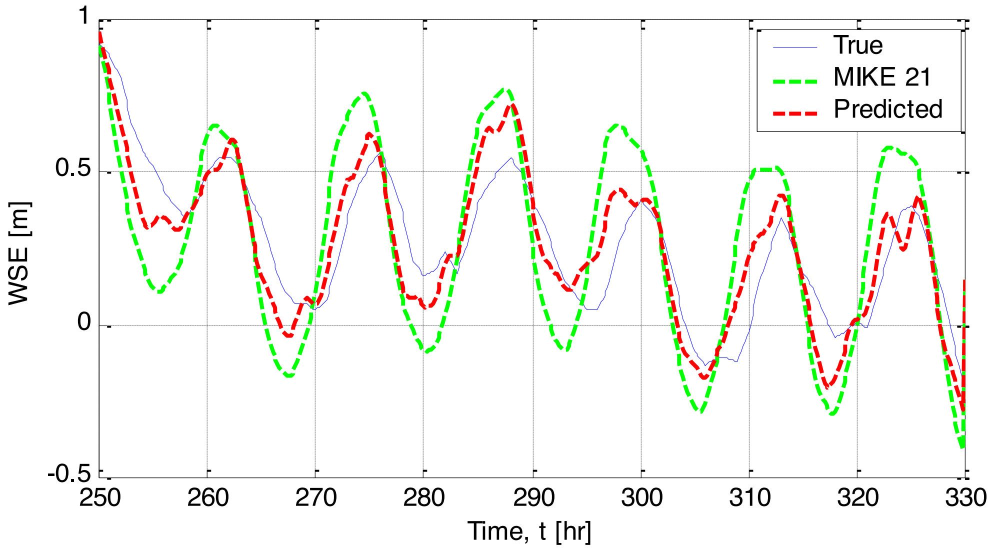
End

Thyboron Havn



RMSE M21= 0.238 m

RMSE $_{corr.}$= 0.118 m

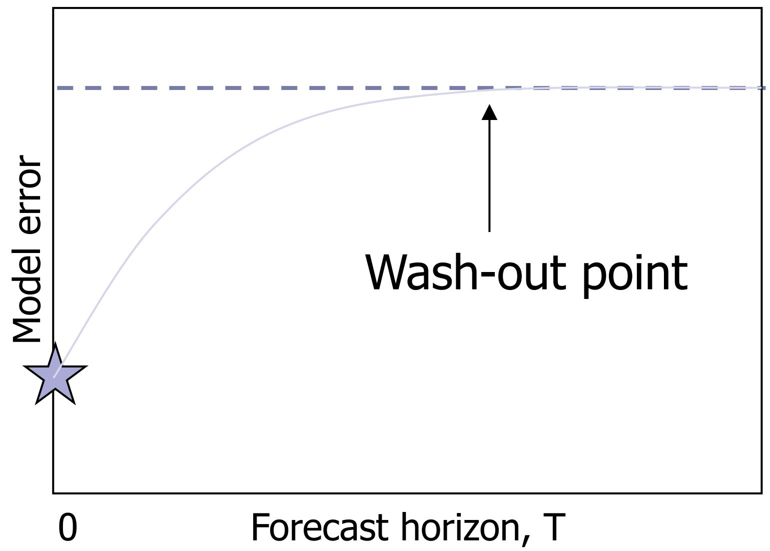


Validation Points

Region	RMSE [m]		
	MIKE 21	EnKF	Gain
British coast	0.6750	0.4894	0.4871
Danish west coast	0.2536	0.2233	0.1620
Inner Danish waters	0.3257	0.1716	0.1704
Swedish coast	0.2064	0.1050	0.0858
Average	0.3652	0.2473	0.2263

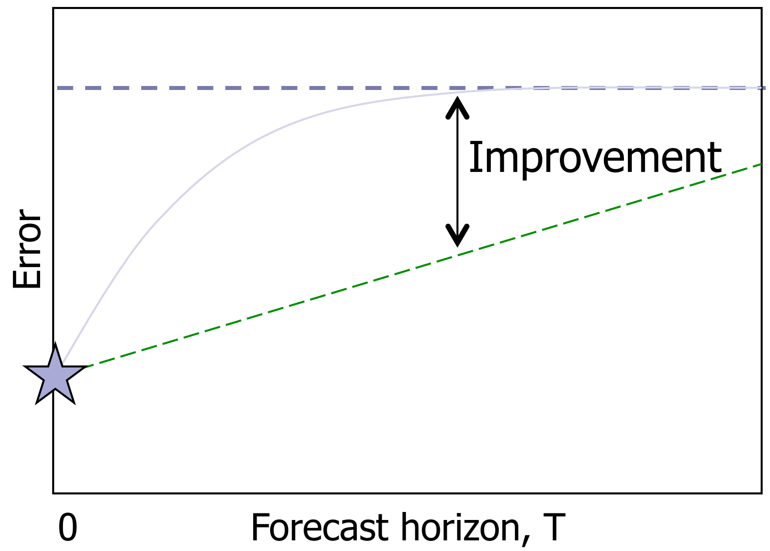
Conventional Data Assimilation

- KF updates the *initial conditions* for a model forecast
- Model is *uncorrected* at future time levels
- Corrected initial conditions are quickly 'washed-out'



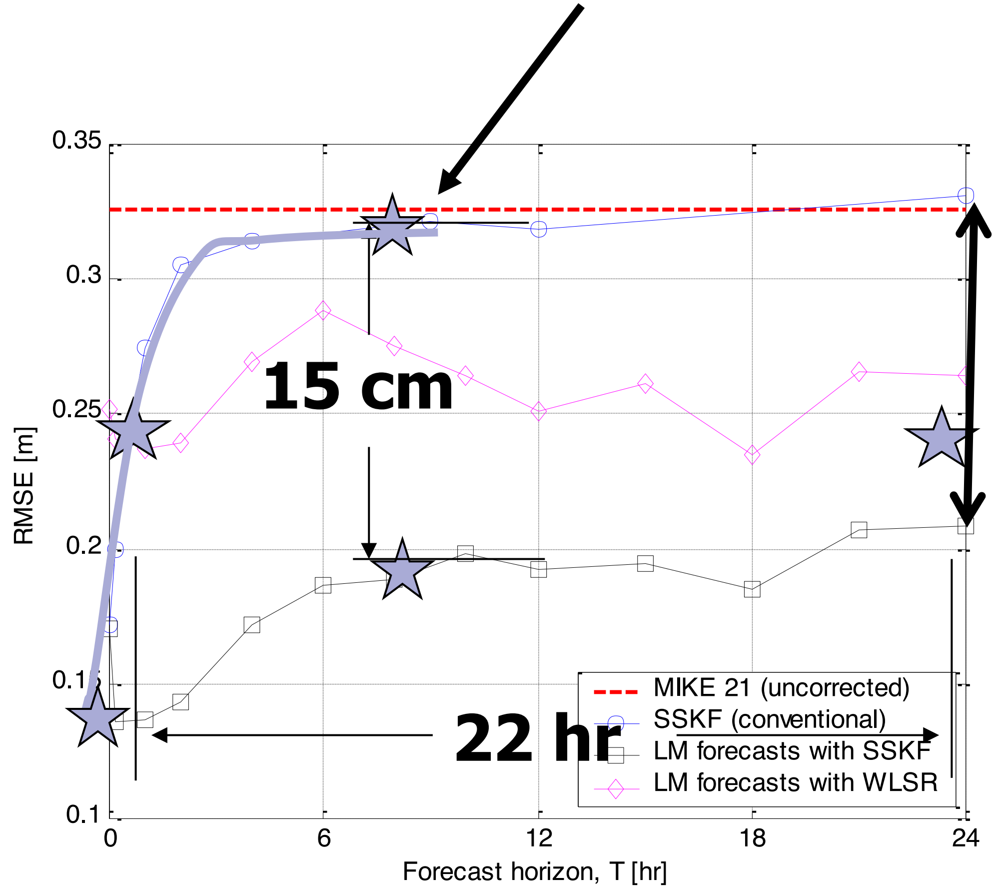
Error forecasting

- Update the initial conditions (as before)
- Forecast errors and correct model at *future* time levels
- Significant improvements for extended forecast horizons



Error forecasting results

- Spatially averaged errors (inner Danish waters)
- Standard approach:
  - Initial correction washed out after 8-12 hr
- Error forecasting:
  - Improved model results even after 24 hr!



Computational expense

- EnKF
  - Typically the equivalent of 20-100 model generations per updating step
- Error Correction based on steady-state gain matrix
  - Slightly more expensive than the model simulation

Conclusions

- Evolutionary embedding gives high quality local modelling parameters and resulting forecast
- Such optimised local models can then be used for error forecasting and data assimilation
- Spatial distribution of errors based on correlation between errors and model dynamics

Conclusions (2)

- Assimilation of error forecasts can be used to significantly improve model results far beyond the time it takes for updated initial conditions to ‘wash-out’
- A ‘hybrid’ scheme is fundamentally sound since it utilises:
  - Gain matrix assimilation
  - Local model forecasts