



# BME I5100: Biomedical Signal Processing

## Hidden Markov Model Kalman Filter



Lucas C. Parra  
Biomedical Engineering Department  
City College of New York





# Schedule

## Week 1: Introduction

Linear, stationary, normal - the stuff biology is **not** made of.

## Week 1-5: Linear systems

Impulse response

Moving Average and Auto Regressive filters

Convolution

Discrete Fourier transform and z-transform

Sampling

## Week 6-7: Analog signal processing

Operational amplifier

Analog filtering

## Week 8-11: Random variables and stochastic processes

Random variables

Moments and Cumulants

Multivariate distributions, Principal Components

Stochastic processes, linear prediction, AR modeling

## Week 12-14: Examples of biomedical signal processing

Harmonic analysis - **estimation** circadian rhythm and speech

Linear discrimination - **detection** of evoked responses in EEG/MEG

Hidden Markov Models and Kalman Filter- **identification** and **filtering**



# Dynamic State Space Model

The basic notion is that there are **states** described by  $\mathbf{x}(t)$  that develop according to some **dynamic** in which the current state depends on the past states:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots)$$

Almost exclusively we are concerned with **discrete time** dynamic, i.e. time  $t$  is measured in integer increments.\*

More generally the dynamic may be stochastic in which case we describe it with the conditional PDF

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots)$$

A **dynamical model** tries to capture the random process with an analytic expression for this conditional PDF.

\* The notation,  $\mathbf{x}_t$ , in these slides will be different from the convention  $\mathbf{x}[n]$  in DSP<sub>3</sub>



# Dynamic State Space Model - Markov

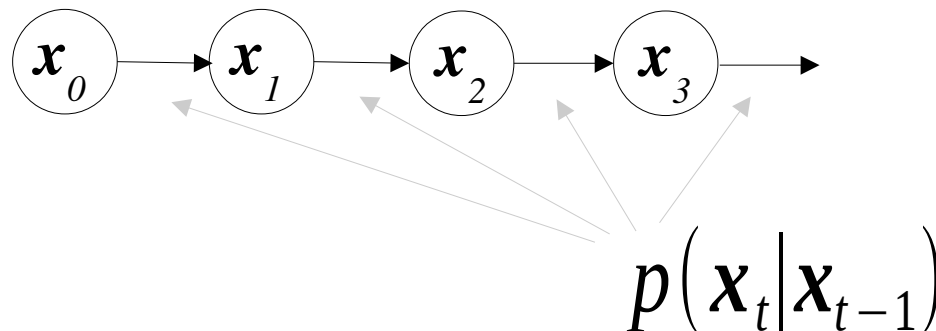
A dynamical model is called **Markov** (of order  $p$ ) if the dependence on the past is limited (to the last  $p$  states):

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-p})$$

For a **first order** Markov model or process the dependence is on the **immediate past**:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots) = p(\mathbf{x}_t | \mathbf{x}_{t-1})$$

This is represented with the following graph

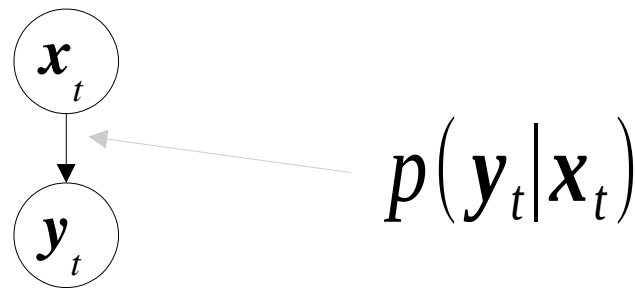




# Dynamic State Space Model - Hidden

In a **hidden** state space model the states can not be directly observed. Instead we observe random variable  $y_t$  that give us only indirect information about  $x_t$ .

The notion is that states  $x_t$  "emit" observation  $y_t$  with a certain probability:



Examples:

Hidden State  $x$

phoneme

sleep state

position of ball

knee angle

pitch frequency

Observation  $y$

acoustic spectra

EEG waveform

blob on video

tilt sensor

acoustic waveform

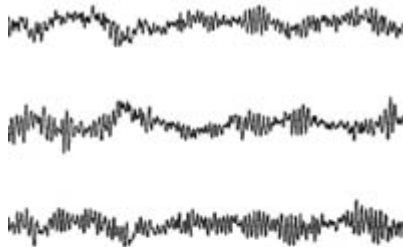


# Dynamic State Space Model - Example

Examples: EEG Sleep States

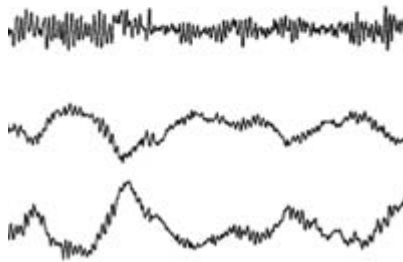
**Awake**

rapid irregular



**Sleep stage 1**

Alpha activity

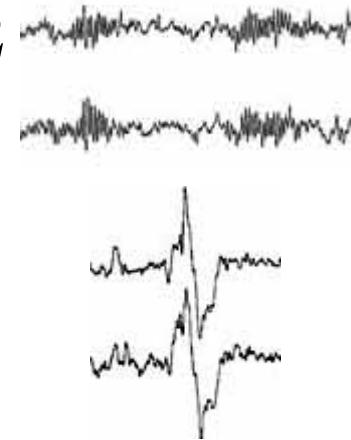


or slow eye  
movements

**Sleep stage 2**

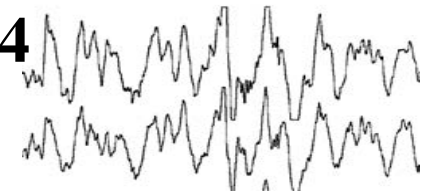
spindles  
and

K-complex



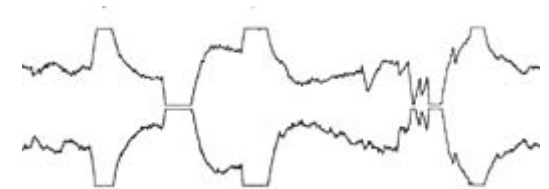
**Sleep stage 3/4**

slow delta  
waves



**REM**

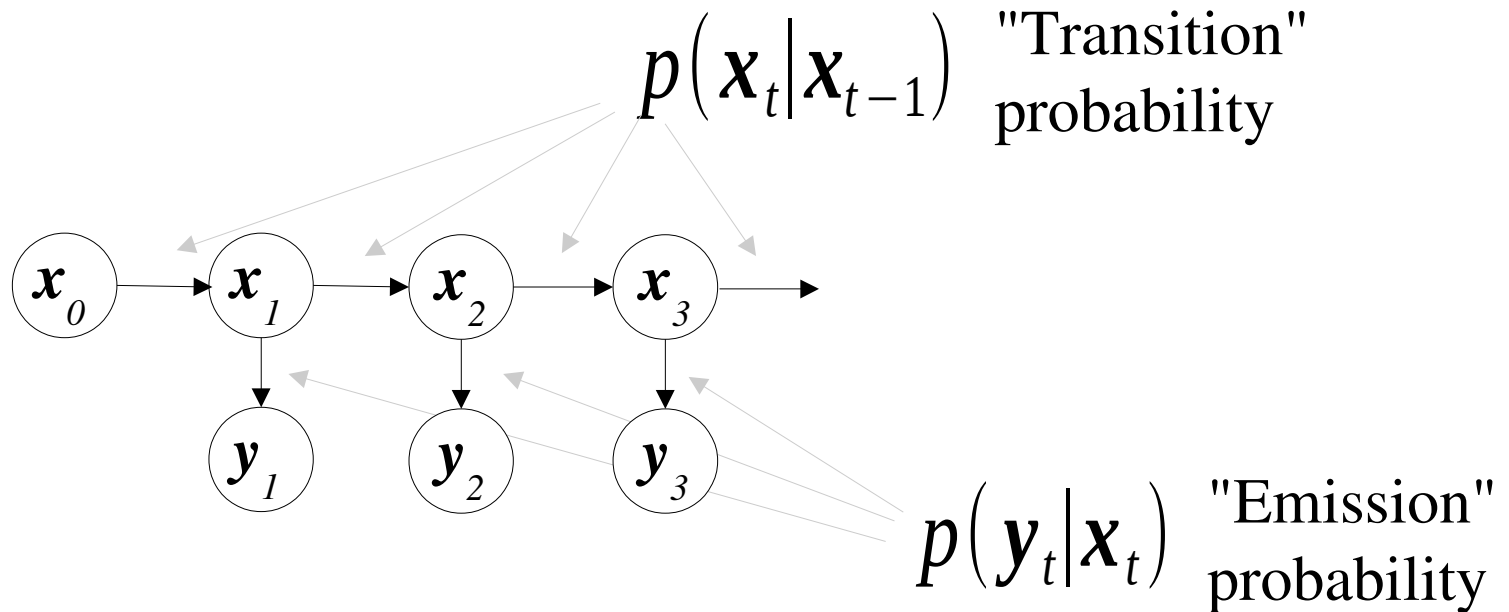
rapid eye  
movements





# DSSM - Hidden Markov and Kalman Model

Together this can be represented as



In a **Hidden Markov Model** the hidden states are conventionally discrete variables and the observation continuous.

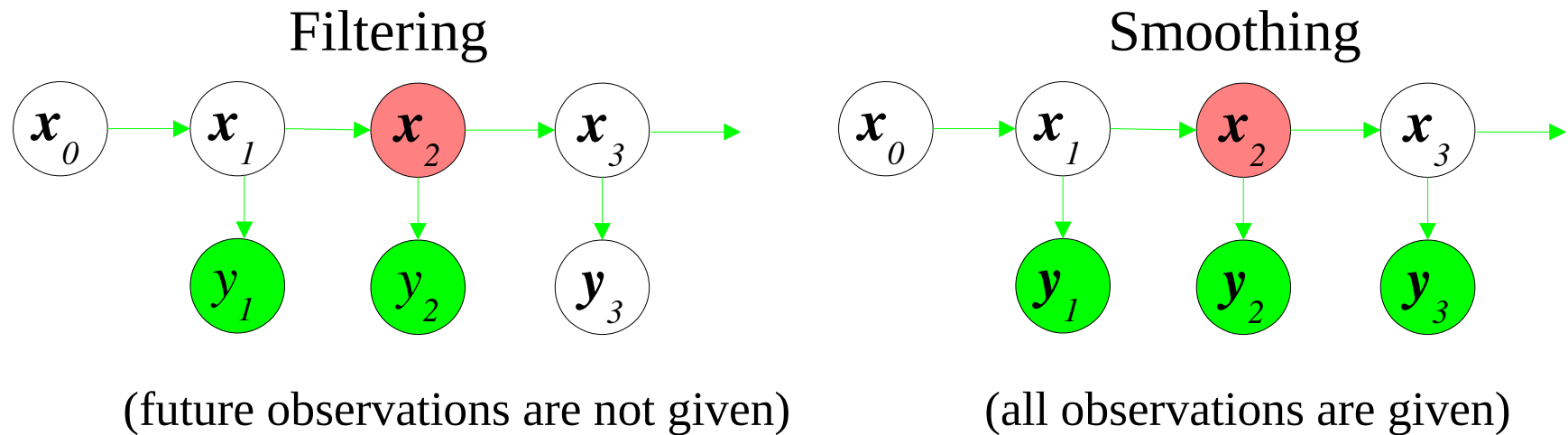
In a **Kalman** State Space Model hidden states and observations are Gaussian and all relations in the model,  $p(\mathbf{y}_t | \mathbf{x}_t)p(\mathbf{x}_t | \mathbf{x}_{t-1})$ , are linear.



# DSSM - Estimation and Identification

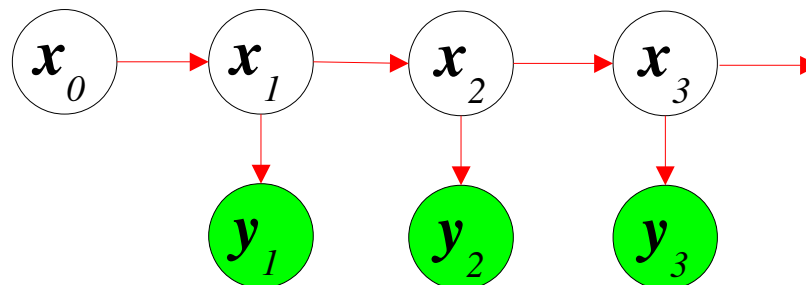
**Estimation** or Inference:

Given model and observations  $y$  what are the hidden states  $x$ ?



**Identification** or Learning:

Given all observations  $y$  what is the model  $p(y_t|x_t)p(x_t|x_{t-1})$ ?







# DSSM - Identification or Learning

**Estimation** and **Identification** can both be based on the joint PDF of the data and hidden states

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{y}_1, \dots, \mathbf{y}_T | \mathbf{x}_0) = \prod_{t=1}^T p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1})$$

In **Identification** we parameterize the distributions with parameters  $\theta$  to get the joint likelihood

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{y}_1, \dots, \mathbf{y}_T | \Theta) = \prod_{t=1}^T p(\mathbf{y}_t | \mathbf{x}_t; \Theta) p(\mathbf{x}_t | \mathbf{x}_{t-1}; \Theta)$$

and find the optimal parameters with maximum likelihood

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} p(\mathbf{y}_1, \dots, \mathbf{y}_T | \Theta)$$

using the EM algorithm that is based on the joint PDF.



# DSSM - Estimation or Inference

**Estimation** and **Identification** can both be based on the joint likelihood of the data and hidden states

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{y}_1, \dots, \mathbf{y}_T | \mathbf{x}_0) = \prod_{t=1}^T p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1})$$

In **Estimation** we use the maximum of the posterior (MAP)

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T | \mathbf{y}_1, \dots, \mathbf{y}_T) = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{y}_1, \dots, \mathbf{y}_T)}{p(\mathbf{y}_1, \dots, \mathbf{y}_T)}$$

For instance in **Filtering**

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}} p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) = \underset{\mathbf{x}_t}{\operatorname{argmax}} p(\mathbf{x}_t, \mathbf{y}_1, \dots, \mathbf{y}_t)$$



# DSSM - Estimation or Inference

**Filtering:** How exactly do we compute the likelihood of the current state from past and current observations for filtering?

$$p(\mathbf{x}_t, \mathbf{y}_T, \dots, \mathbf{y}_1)$$

Starting with  $p(\mathbf{x}_0, \cdot)$  we apply the following (Kalman) recursion:

$$p(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \dots, \mathbf{y}_1) \rightarrow p(\mathbf{x}_t, \mathbf{y}_t, \dots, \mathbf{y}_1)$$

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{y}_t, \dots, \mathbf{y}_1) &= p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t, \mathbf{y}_{t-1}, \dots, \mathbf{y}_1) \\ &= p(\mathbf{y}_t | \mathbf{x}_t) \int d\mathbf{x}_{t-1} p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \dots, \mathbf{y}_1) \end{aligned}$$



# DSSM - Estimation or Inference

**Smoothing:** How exactly do we compute the likelihood of the current state from all observations?

$$p(\mathbf{x}_t, \mathbf{y}_T, \dots, \mathbf{y}_1)$$

With the following we can use the previous recursion up to  $t$

$$p(\mathbf{x}_t, \mathbf{y}_T, \dots, \mathbf{y}_1) = \frac{p(\mathbf{y}_t, \dots, \mathbf{y}_T | \mathbf{x}_t)}{p(\mathbf{y}_t | \mathbf{x}_t)} p(\mathbf{x}_t, \mathbf{y}_t, \dots, \mathbf{y}_1)$$

and need now a backwards recursion starting with  $p(\mathbf{y}_T | \mathbf{x}_T)$

$$p(\mathbf{y}_{t-1}, \dots, \mathbf{y}_T | \mathbf{x}_{t-1}) \leftarrow p(\mathbf{y}_t, \dots, \mathbf{y}_T | \mathbf{x}_t)$$

$$p(\mathbf{y}_{t-1}, \dots, \mathbf{y}_T | \mathbf{x}_{t-1})$$

$$= p(\mathbf{y}_{t-1} | \mathbf{x}_{t-1}) p(\mathbf{y}_t, \dots, \mathbf{y}_T | \mathbf{x}_{t-1})$$

$$= p(\mathbf{y}_{t-1} | \mathbf{x}_{t-1}) \int d\mathbf{x}_t p(\mathbf{y}_t, \dots, \mathbf{y}_T | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1})$$



# DSSM - Kalman Filter

In a **Kalman** State Space Model the hidden states depend on the previous states linearly with additive zero mean white Gaussian *state transition noise*  $\mathbf{w}$ :

$$\mathbf{x}_t = \mathbf{A} \mathbf{x}_{t-1} + \mathbf{w}$$

The observations depend on the current state also linearly with additive zero mean white Gaussian *sensor or observation noise*  $\mathbf{v}$ .

$$\mathbf{y}_t = \mathbf{C} \mathbf{x}_t + \mathbf{v}$$

Hence

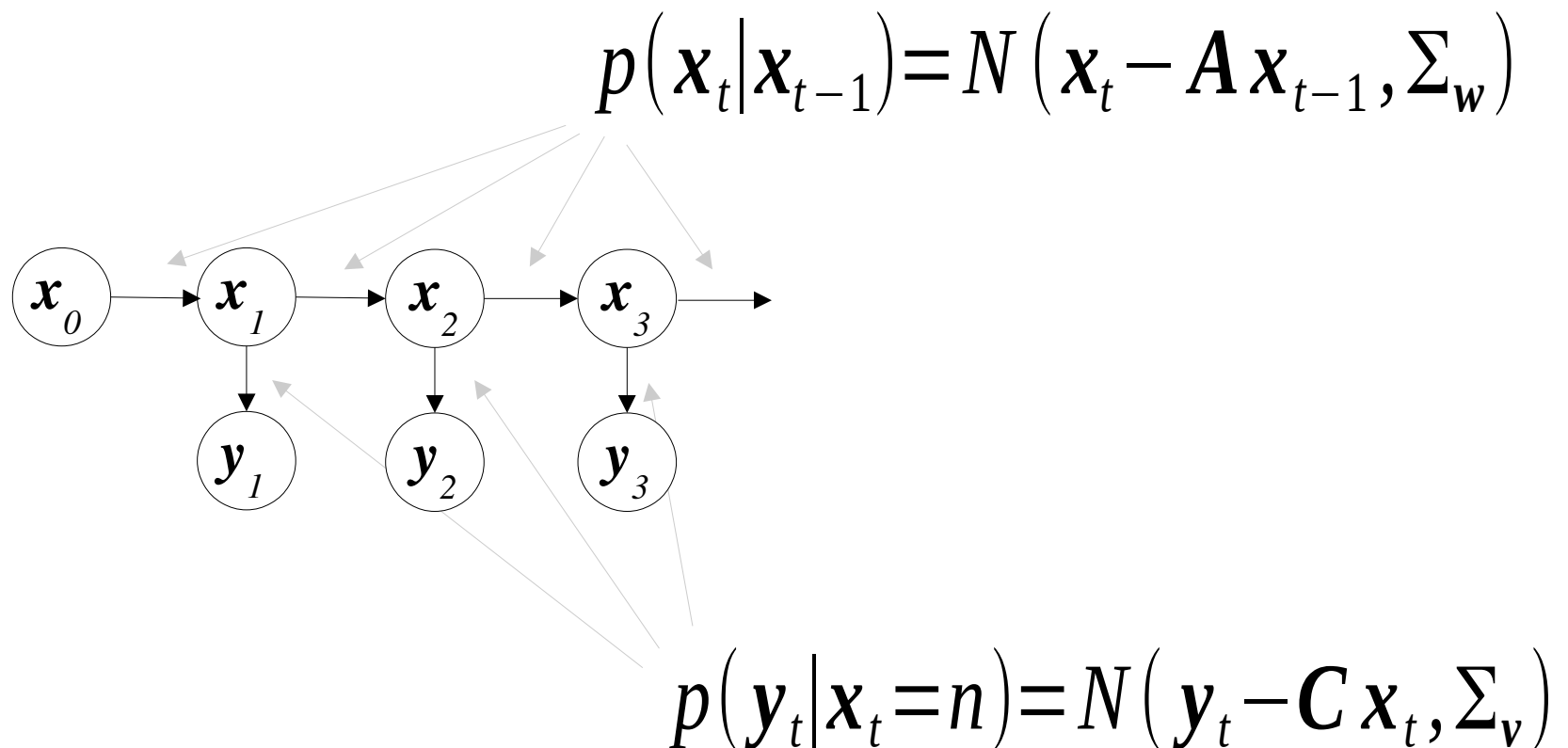
$$p(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t - \mathbf{A} \mathbf{x}_{t-1}, \Sigma_w)$$

$$p(\mathbf{y}_t | \mathbf{x}_t) = N(\mathbf{y}_t - \mathbf{C} \mathbf{x}_t, \Sigma_v)$$



# DSSM - Kalman State Space Model

The model that corresponds to a Kalman filtering has Gaussian transition probabilities and Gaussian emission probabilities:





# DSSM - Kalman Filter

In Kalman filtering all probabilities are Gaussian because the convolution and product of Gaussians is Gaussian:

$$\begin{aligned}
 p(\mathbf{x}_t, \mathbf{y}_t, \dots, \mathbf{y}_1) \\
 &= p(\mathbf{y}_t | \mathbf{x}_t) \int d\mathbf{x}_{t-1} p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \dots, \mathbf{y}_1) \\
 &= N(\mathbf{y}_t - \mathbf{C} \mathbf{x}_t, \Sigma_v) \int d\mathbf{x}_{t-1} N(\mathbf{x}_t - \mathbf{A} \mathbf{x}_{t-1}, \Sigma_w) N(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}, \hat{\Sigma}_{t-1}) \\
 &= N(\mathbf{x}_t - \hat{\mathbf{x}}_t, \hat{\Sigma}_t)
 \end{aligned}$$

This joint likelihood is therefore Gaussian

with mean 
$$\hat{\mathbf{x}}_t = \hat{\Sigma}_t \mathbf{C}^T \Sigma_v^{-1} \mathbf{y} + \hat{\Sigma}_t \bar{\Sigma}_t^{-1} \mathbf{A} \hat{\mathbf{x}}_{t-1}$$

and covariance 
$$\hat{\Sigma}_t^{-1} = \mathbf{C}^T \Sigma_v^{-1} \mathbf{C} + \bar{\Sigma}_t^{-1}$$

where we defined the prediction covariance 
$$\bar{\Sigma}_t \equiv \Sigma_w + \mathbf{A} \hat{\Sigma}_{t-1} \mathbf{A}^T$$

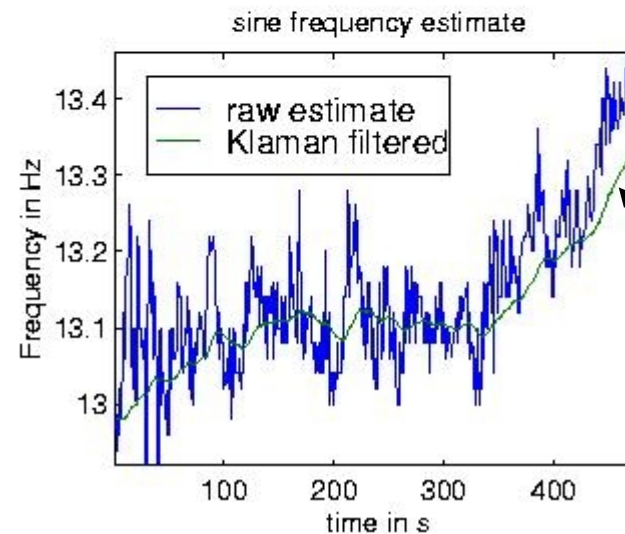
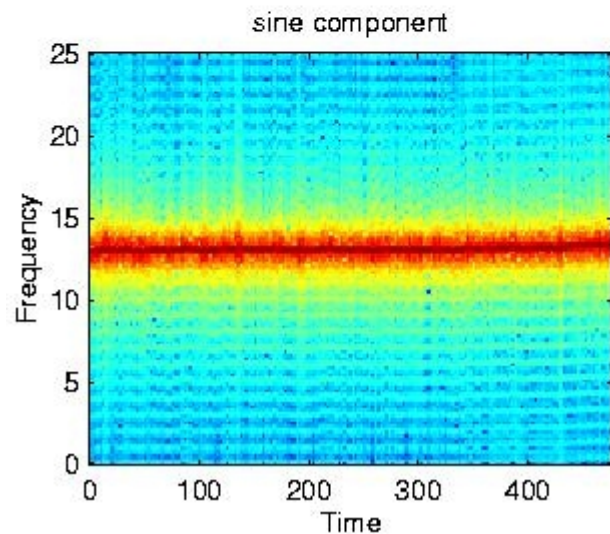
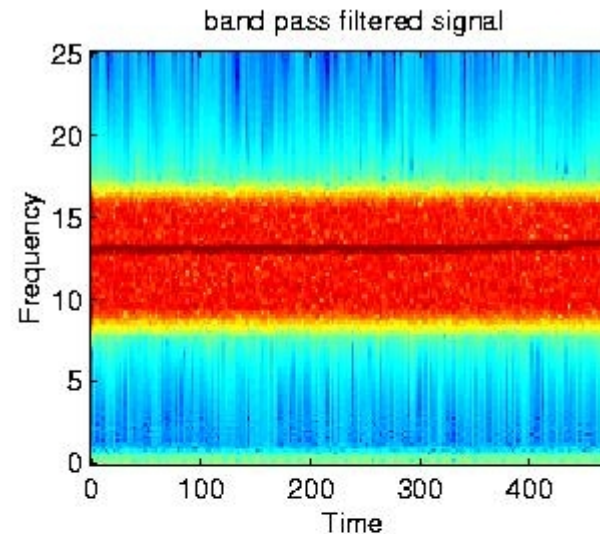
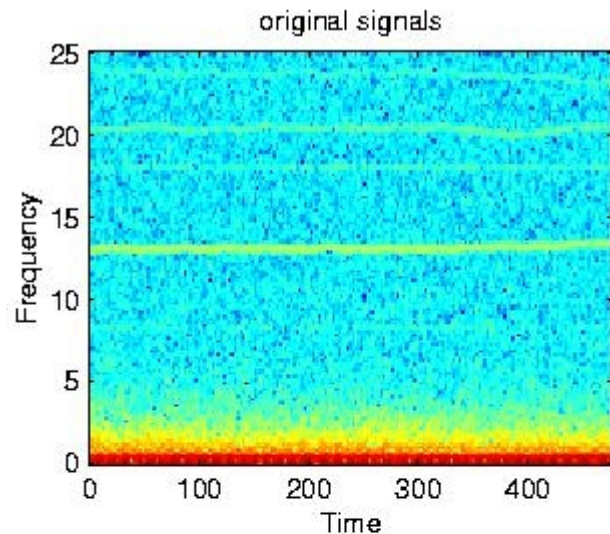
The Kalman filter (MAP) estimate is in fact that mean

$$\hat{\mathbf{x}}_t = \operatorname{argmax}_{\mathbf{x}_t} p(\mathbf{x}_t, \mathbf{y}_1, \dots, \mathbf{y}_t)$$



# DSSM - Kalman Filter

Example: Estimating the changing frequency of a noisy sinusoid.



Given the instantaneous frequency estimate,  $x_t$ , we compute the Kalman filtered estimates using  $A=1$ ,  $C=1$ , and appropriate  $\Sigma_w$  and  $\Sigma_v$ .

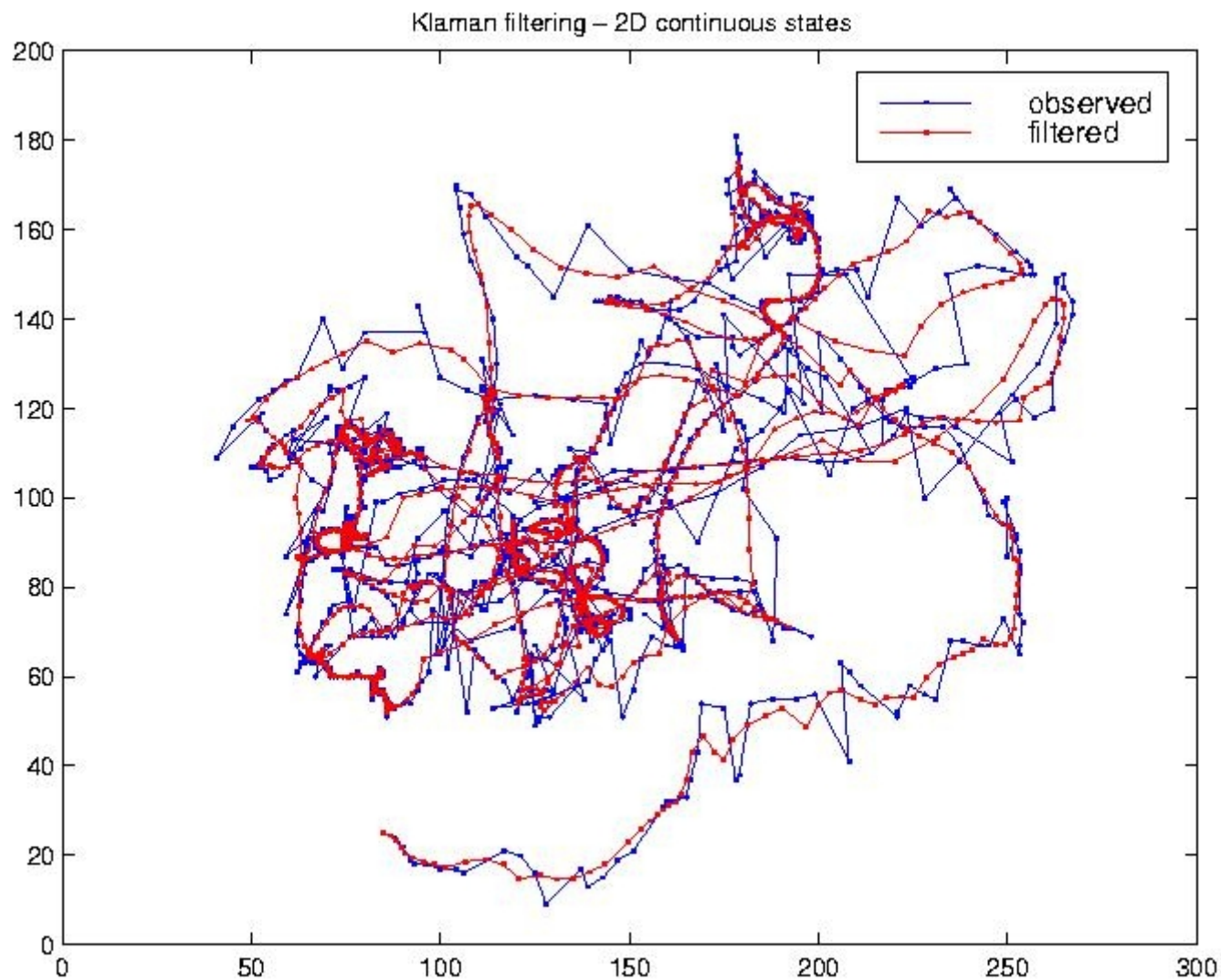
Note that the filter estimate lags behind. Can be avoided by using Kalman smoothing.





# DSSM - Kalman Filter

## Example: 2D Tracking





# Identification or Learning - EM Algorithm

To find good model parameters such as  $\mathbf{A}, \mathbf{C}$ , etc. we maximize the data log likelihood

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} p(\mathbf{y}_1, \dots, \mathbf{y}_T | \Theta) = \underset{\Theta}{\operatorname{argmax}} \log p(\mathbf{Y} | \Theta)$$

It is convenient to consider the following lower bound which is valid for any distribution  $q(\mathbf{X})$ :

$$\begin{aligned} L(\Theta) &= \log p(\mathbf{Y} | \Theta) = \log \int d\mathbf{X} p(\mathbf{Y}, \mathbf{X} | \Theta) \\ &= \log \int d\mathbf{X} q(\mathbf{X}) \frac{p(\mathbf{Y}, \mathbf{X} | \Theta)}{q(\mathbf{X})} \\ &\geq \int d\mathbf{X} q(\mathbf{X}) \log \frac{p(\mathbf{Y}, \mathbf{X} | \Theta)}{q(\mathbf{X})} \equiv F(q, \Theta) \end{aligned}$$

The inequality is known as Jensen's inequality.



# Identification or Learning - EM Algorithm

The EM algorithm maximizes this lower bound

$$\mathbf{E \ step:} \quad q_{k+1} = \underset{q}{\operatorname{argmax}} \ F(q, \Theta_k) = p(\mathbf{X}|\mathbf{Y}, \Theta_k)$$

$$\mathbf{M \ step:} \quad \Theta_{k+1} = \underset{\Theta}{\operatorname{argmax}} \ F(q_{k+1}, \Theta)$$

The **E step** is equivalent with computing the likelihood of the hidden states given the observations and current parameter values. With this likelihood we compute the **Expected** value of the complete data log likelihood:

$$F(\Theta_k, \Theta) = \int d\mathbf{X} \ p(\mathbf{X}|\mathbf{Y}, \Theta_k) \log p(\mathbf{Y}, \mathbf{X}|\Theta) + \text{const.}$$

Hence the name **E step**.

In the **M step** we **Maximize** the lower bound  $F(\Theta_k, \Theta)$  with respect to  $\theta$ . Hence the name **M step**.

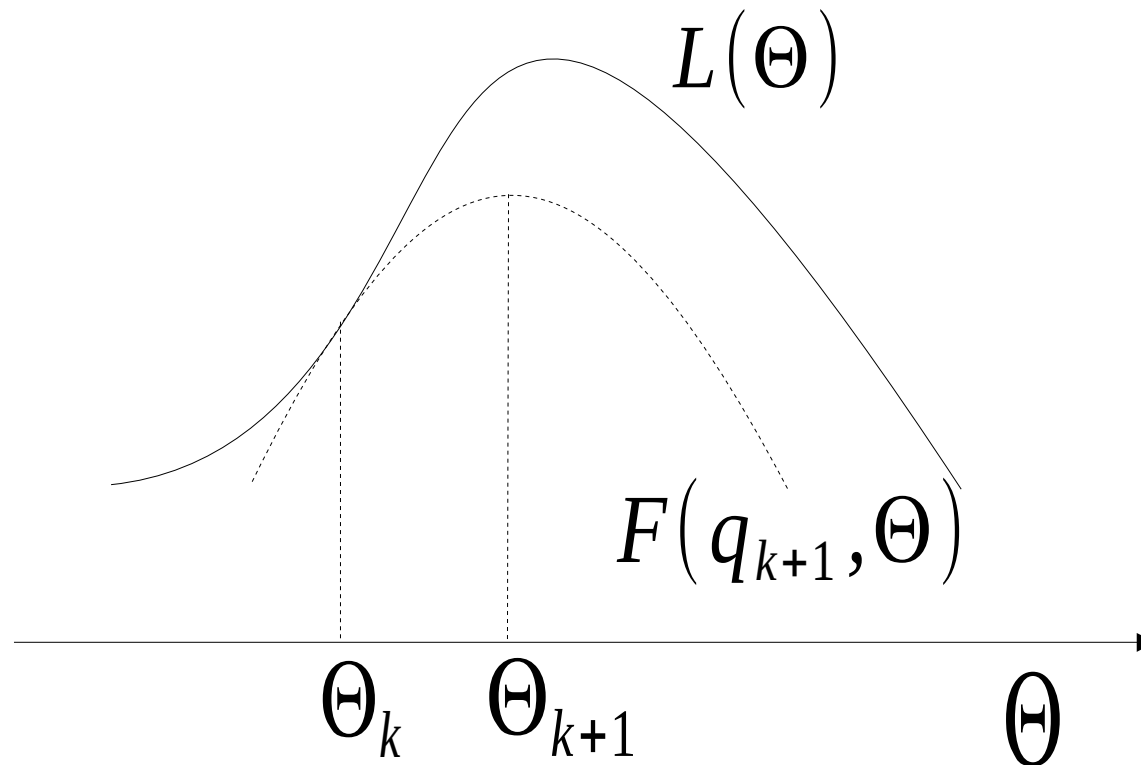


# Identification or Learning - EM Algorithm

**E step:** Find a distribution  $q(\mathbf{X})$  such that  $F(q, \theta_k)$  is maximal.

Turns out that this is,  $q_{k+1}(\mathbf{X}) = p(\mathbf{X}|\mathbf{Y}, \theta_k)$ , and one can show that  $F(\theta_k, \theta_k) = L(\theta_k)$ . Hence  $F(q, \theta)$  is tangential to  $L(\theta)$  at  $\theta_k$ .

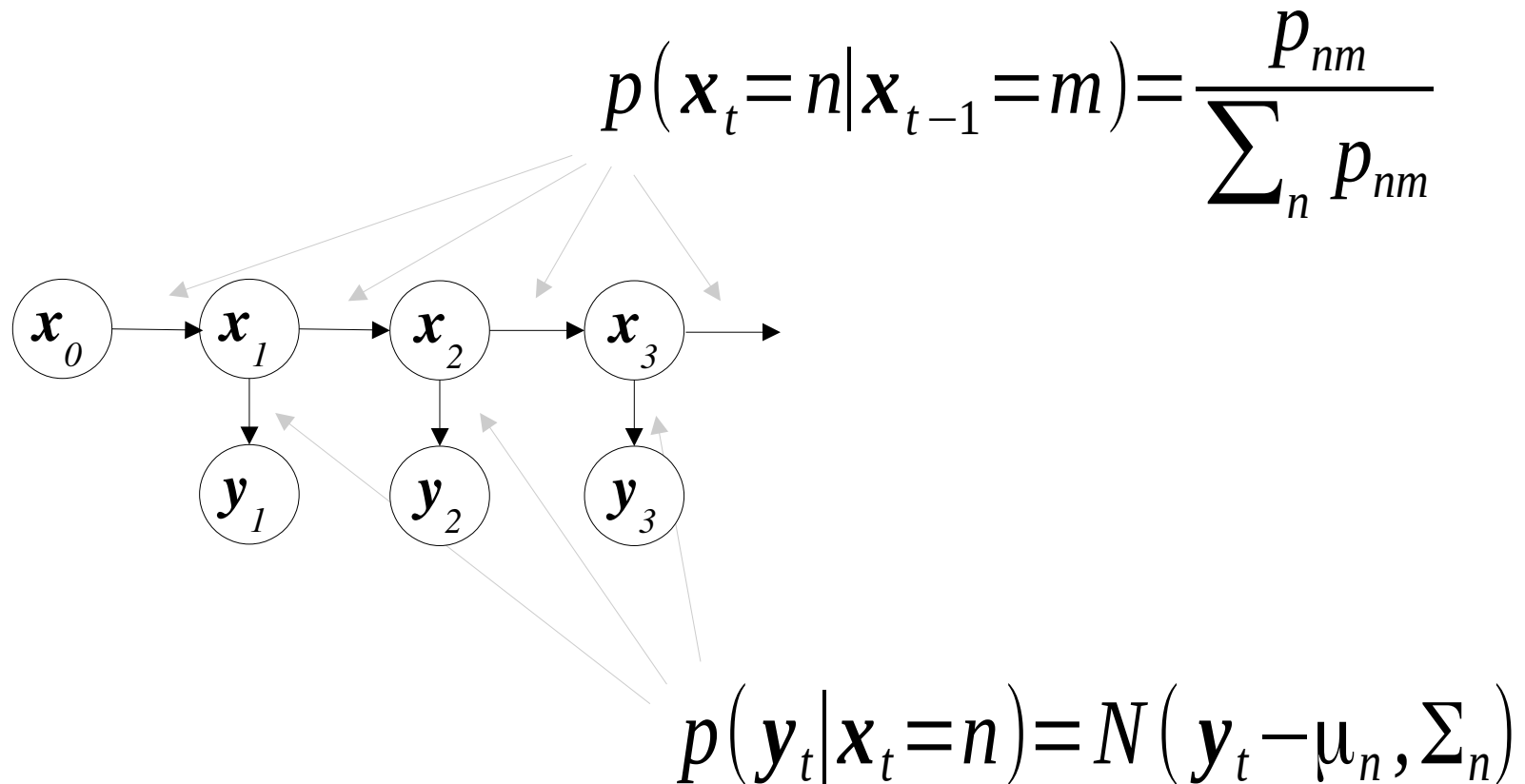
**M step:** Maximize  $F(q_{k+1}, \theta)$  with respect to  $\theta$





# DSSM - Hidden Markov Model

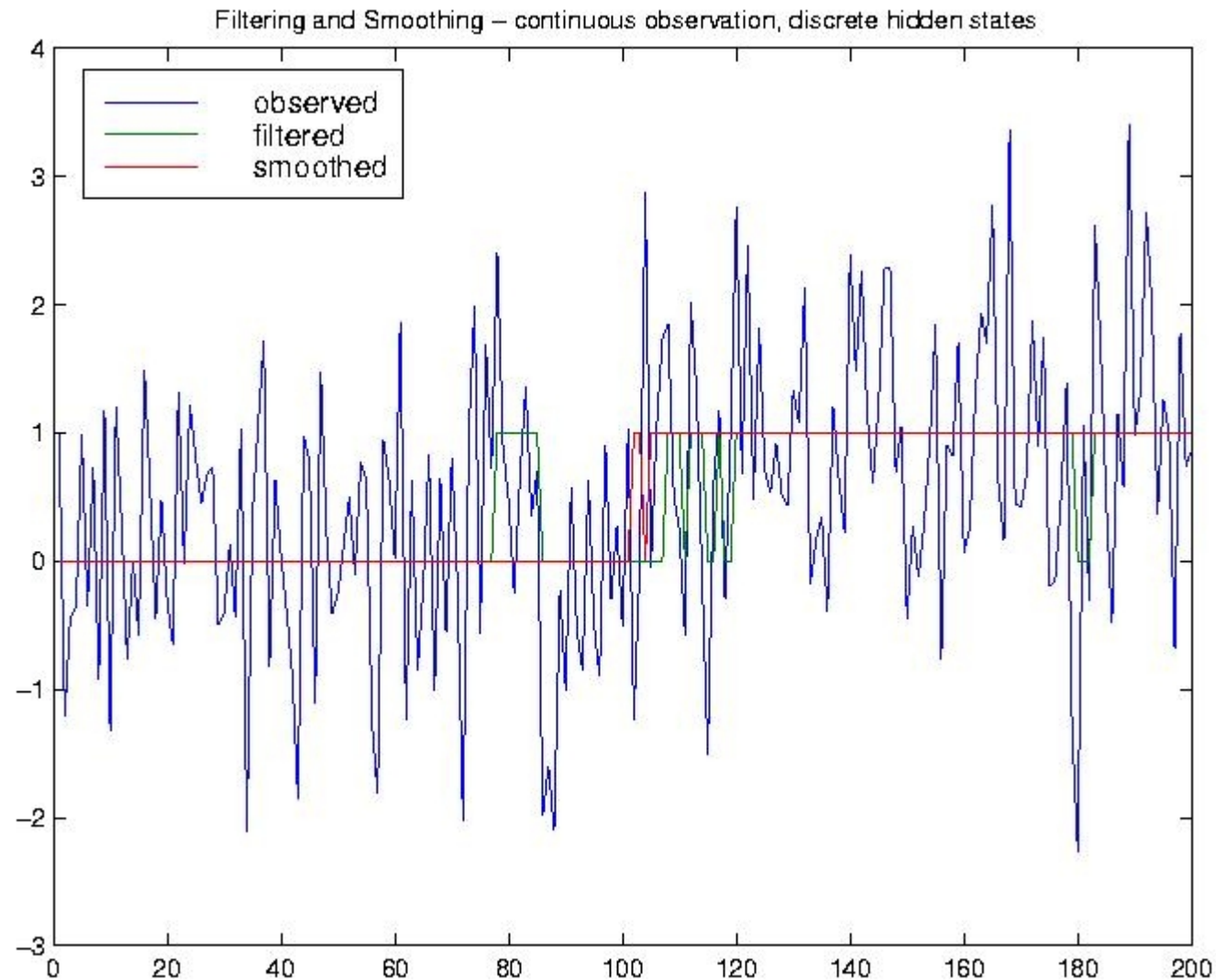
Typical HMMs have discrete states and Gaussian emissions probabilities:





# DSSM - Hidden Markov Model

## Filtering and smoothing example





# Identification or Learning - HMM

**E step:** For an Hidden Markov Model (HMM) an efficient algorithm for computing the likelihood of the hidden state variables is the **Baum-Welsh algorithm**. It uses a *forward* and a *backward* pass that are exactly the *filtering* and *smoothing recursions* discussed above. In the E Step the current parameter values  $\theta_k$  are used.

**M step:** Given these likelihoods the next best parameters for the transition probabilities and emission probabilities are obtained by setting the derivatives of the total log likelihood to zero and solving for new parameters  $\theta_{k+1}$ .



# Identification or Learning - HMM

Consider for example the transition probabilities  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ . Recall that HMM typically have discrete states  $\mathbf{x}_t$ . The transition probability can then be parametrized directly as

$$p(\mathbf{x}_t = n | \mathbf{x}_{t-1} = m) = \frac{p_{nm}}{\sum_n p_{nm}}$$

**M Step:** Setting the derivative of the joint log likelihood with respect to  $p_{nm}$  equal zero gives

$$p_{nm} = \sum_{t=1}^T p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{Y})$$

**E Step:** We compute with the filtering and smoothing algorithm the required probability in the following expression

$$p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{Y}) = p(\mathbf{y}_t, \dots, \mathbf{y}_T | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \dots, \mathbf{y}_1)$$





# For next class

## Assignment 14:

Read Roweis, Ghahramani, "A Unifying Review of Linear Gaussian Models", Neural Computation, Vol. 11, No. 2, 1999.

Select a data set from your own research that you would like to analyze with one of the methods presented in any of the classes thus far. Save the relevant data in a `.mat` file on CD.

We will select one or more data sets from students and analyze is together during class.

**Optional:** Make your best effort and try the method yourself and save a corresponding matlab script on the same CD.