State Estimation-II

Lecture 6 – Thursday November 10, 2016

Objectives

When you have finished this lecture you should be able to:

- Understand **Kalman filter** and its roles in state estimation.
- Understand **Markov process** and **Markov models**.

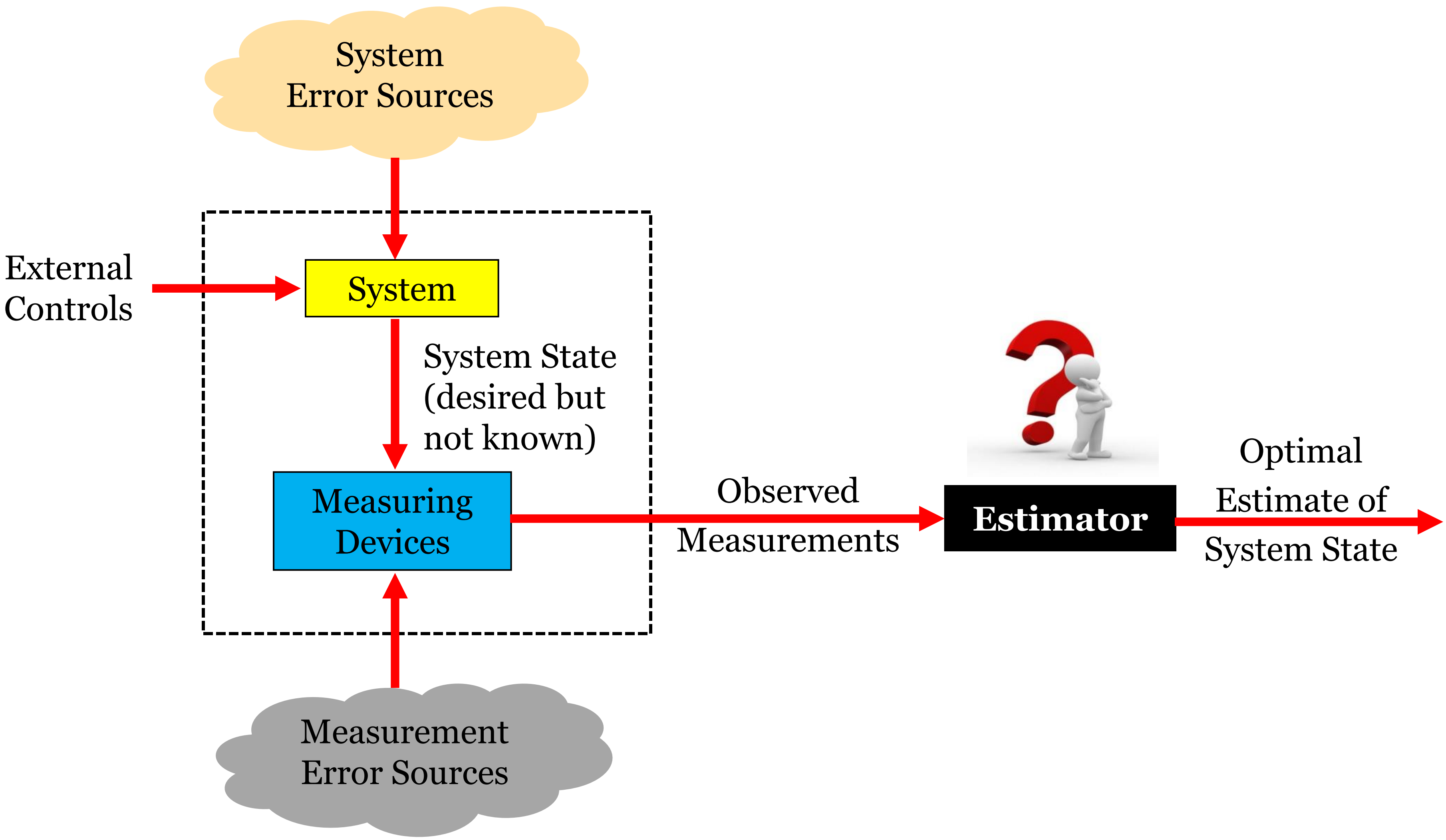
Outline

- Kalman Filters
- Markov Models
- Summary

Outline

- **Kalman Filters**
- Markov Models
- Summary

Kalman Filters



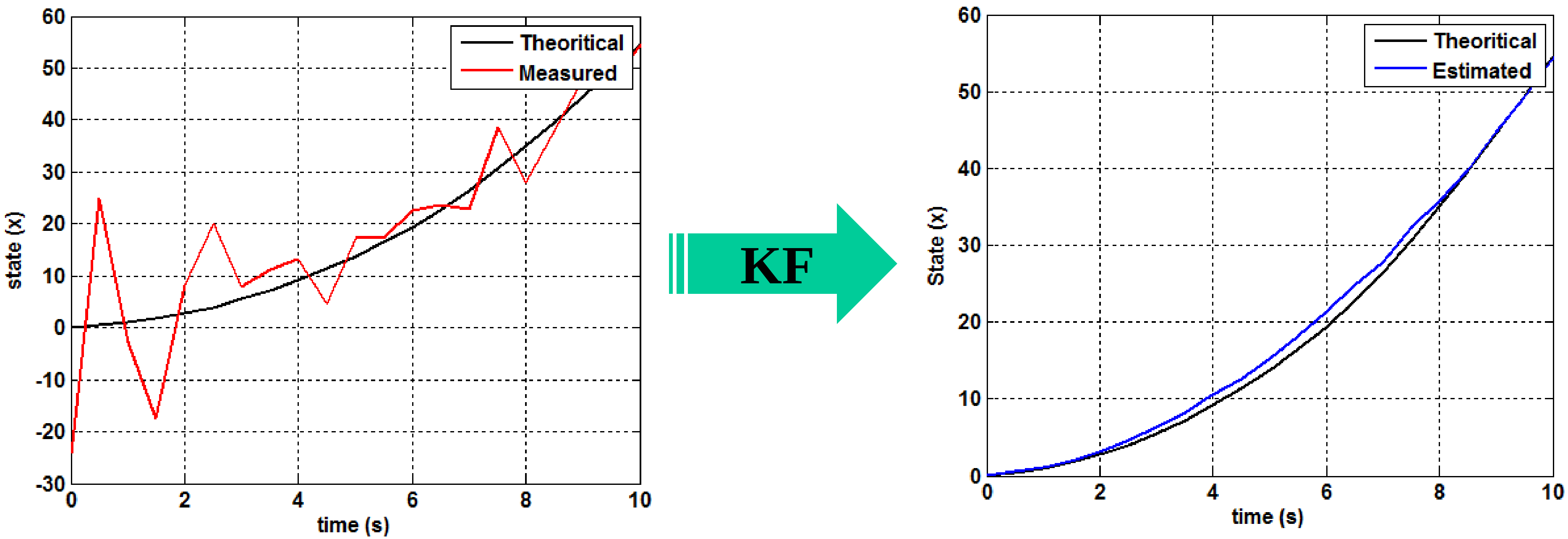
Kalman Filters

- In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete data linear filtering problem.
- This recursive algorithm is known as the Kalman Filter (KF) and it is used to generate optimal estimate of the states of a system from a series of incomplete and noisy measurements. For linear system and white Gaussian noise, Kalman filter is best estimate.
- There are many **applications** for Kalman Filters:
  - ◇ Noise filtration
  - ◇ Tracking objects
  - ◇ Navigation of aircrafts and vehicles
  - ◇ Computer Vision applications, ...

Rudolf Emil Kálmán
(1930-)
Hungarian-American electrical engineer, mathematical system theorist, and college professor

Kalman Filters

Noisy Data → **Kalman Filter** → **Less Noisy Data**



Kalman Filters

- **KF Algorithm**
  - ◇ A linear system can be described using two equations:
    - **State Equation:** $x_t = A_t x_{t-1} + B_t u_{t-1} + w_{t-1}$

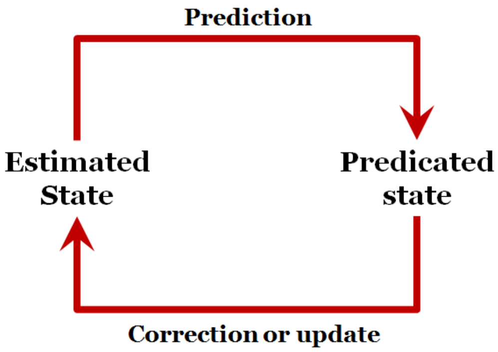
      where: x_t => state
      A_t => state transition matrix
      u_t => control variable(s)
      B_t => control matrix
      w_t => process noise

    - **Measurement Equation:** $z_t = C_t x_t + v_t$

      where: z_t => measurement
      v_t => measurement noise
      C_t => measurement matrix

Kalman Filters

- **KF Algorithm**
  - ◇ The equations of the Kalman Filter are divided into two main groups:
    - **Prediction Equations :**
      - This is called the 'prediction' stage.
      - It projects forward in time the current state to get a priori estimates for the next time step.
    - **Correction Equations :**
      - This is called the 'correction' stage
      - It is responsible for the feedback.
      - It incorporates a new measurement into the a priori estimate to obtained an improved a posteriori estimate.



Kalman Filters

- **KF Algorithm**
  - **Predictor Equations**

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1|t-1} + B_t u_t$$

PREDICTED STATE
Projecting state estimate from time *t-1* to t

Using state estimate from time *t-1*

➢ **Subscripts are as follows:**

- t|t represents the current time period,
- t-1|t-1 previous time period
- t|t-1 are intermediate steps.

Kalman Filters

- **KF Algorithm**
  - ◇ **Predictor Equations**

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1|t-1} + B_t u_t$$

<u>PREDICTED STATE</u>
Projecting state estimate from time *t-1* to t

Using state estimate from time *t-1*

$$P_{t|t-1} = A_t P_{t-1|t-1} A_t^T + Q_t$$

<u>PREDICTED COVARIANCE</u>
Projecting Error Covariance from time *t-1* to *t*

Using Error Covariance at time *t-1*

Process Variance Matrix

Kalman Filters

- **KF Algorithm**
  - ◇ **Corrector Equations**

$$K_t = P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}$$

Kalman Gain (K_t)

Measurement Variance (R_t)

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (z_t - C_t \hat{x}_{t|t-1})$$

UPDATE STATE ESTIMATE ($\hat{x}_{t|t}$)

Measurement (z_t)

Predicted Measurement ($C_t \hat{x}_{t|t-1}$)

Residual/Innovation ($z_t - C_t \hat{x}_{t|t-1}$)

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}$$

UPDATED ERROR COVARIANCE ($P_{t|t}$)

Kalman Filters

- **KF Algorithm**

PREDICTION

Project the state ahead

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1|t-1} + B_t u_t$$

Project the error covariance ahead

$$P_{t|t-1} = A_t P_{t-1|t-1} A_t^T + Q_t$$

CORRECTION

Compute Kalman Gain

$$K_t = P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}$$

Update estimate with measurement z_t

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (z_t - C_t \hat{x}_{t|t-1})$$

Update error covariance

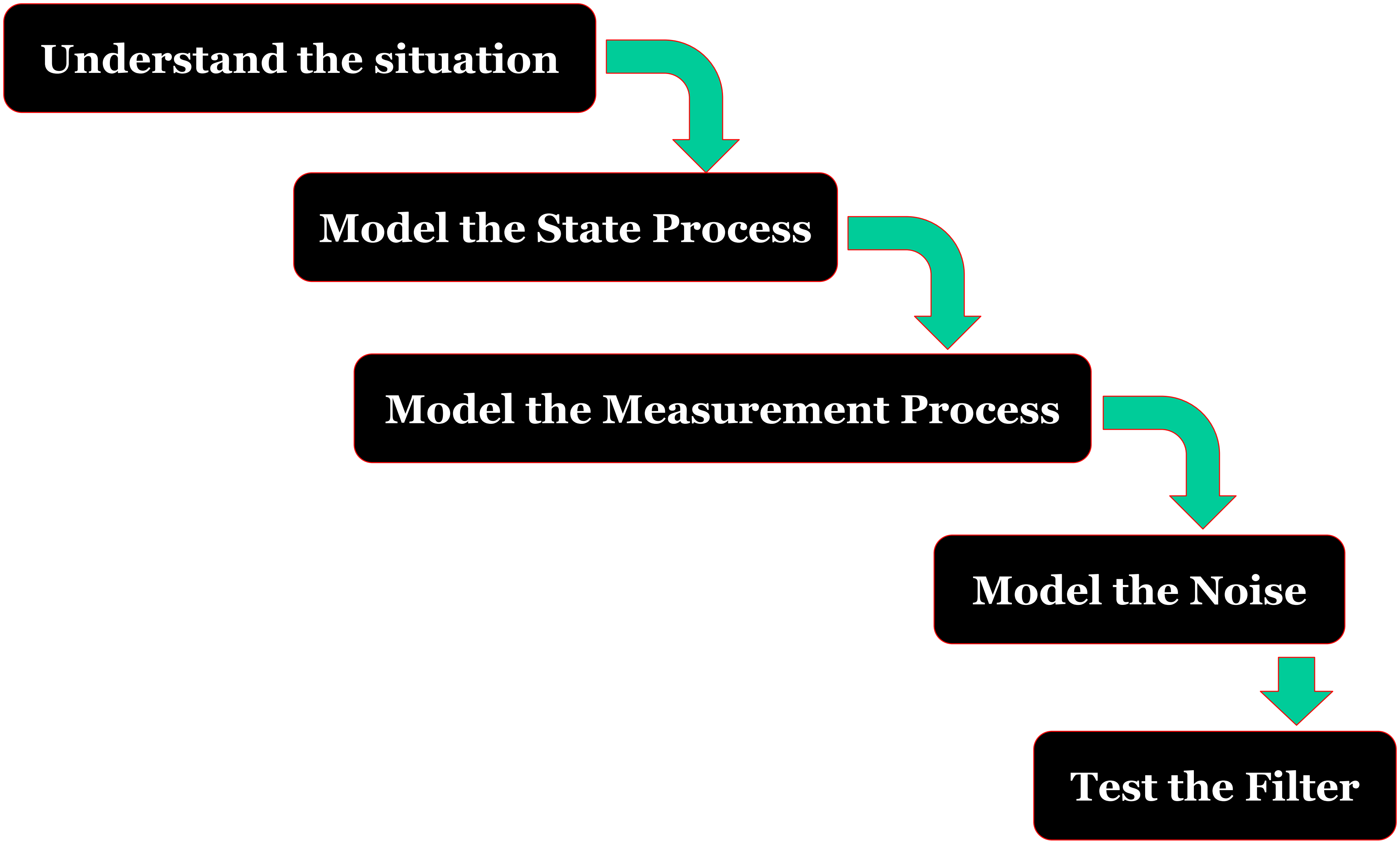
$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}$$

Initial Estimates

$\hat{x}_{t-1}$ and P_{t-1}

Kalman Filters

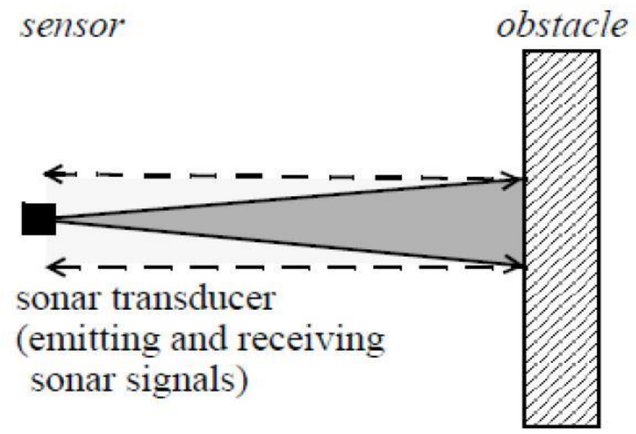
- **KF Algorithm**



Kalman Filters

Example: Mobile robots are equipped with various sensor types for measuring distances to the nearest obstacle around the robot for navigation purposes. These sensors include **Sonar** based on **(Sound Navigation & Ranging)** Sensors, **Laser** Sensors based on **LIDAR (LIght Direction And Ranging)** and *Infrared* Sensors based on **LADAR (RAdio Direction And Ranging)**.

- Assume that a **noisy sonar sensor** is used to estimate the distance from the robot to an obstacle.
- Assume that the distance is static and theoretically **L=100 cm**.



Kalman Filters

- **Model the state process**

The **state variable** $\hat{x}_t$ of the system is the distance to the obstacle.

Since it is a constant model, therefore, A_t is **1** for all time *t*.

The input of the system u_t and matrix B_t are zero.

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1|t-1} + B_t u_t$$

$$\hat{x}_{t|t-1} = \hat{x}_{t-1|t-1}$$

Kalman Filters

- **Model the measurement process**
  - ◇ In this model, the z_t represents the **distance** to the obstacle as measured by the sensors.
  - ◇ It is assumed the measurement is exactly the **same scale** as the estimate $\hat{x}_t$ and so C_t is 1.

$$z_t = C_t x_t$$

$$z_t = x_t$$

Kalman Filters

- **Model the noise**
  - ◇ For this model, we are going to assume that there is a **Gaussian** white noise from the measurement which has a standard deviation of 0.5 cm. Therefore, $R_t = r = 0.25 cm^2$
  - ◇ The **process noise** is assumed to have a standard deviation of 0.01 cm, therefore, $Q_t = q = 0.0001 cm^2$

Kalman Filters

- **Predict equations:**

$$\hat{x}_{t|t-1} = \hat{x}_{t-1|t-1} \qquad P_{t|t-1} = P_{t-1|t-1} + 0.0001$$

- **Update equations:**

$$K_t = P_{t|t-1}(P_{t|t-1} + 0.25)^{-1} \qquad \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - \hat{x}_{t|t-1})$$

$$P_{t|t} = P_{t|t-1} - K_t P_{t|t-1}$$

- **Initialization:**

$$\hat{x}_0 = 0 \qquad \hat{P}_0 = 1000$$

- **Measurements:**

$$z_1 = 99.17cm \qquad z_2 = 100.60cm$$

$$z_3 = 100.12cm \qquad z_4 = 99.61cm$$

Kalman Filters

- **1st Iteration**

$$\hat{x}_{1|0} = \hat{x}_{0|0} = 0$$

$$P_{1|0} = P_{0|0} + 0.0001 = 1000 + 0.0001 = 1000.0001$$

$$K_1 = P_{1|0}(P_{1|0} + 0.25)^{-1} = 1000.0001(1000.0001 + 0.25)^{-1} = 0.9998$$

$$\hat{x}_{1|1} = \hat{x}_{1|0} + K_1(z_1 - \hat{x}_{1|0})$$
$$= 0 + 0.9998(99.17 - 0) = 99.15$$

$$P_{1|1} = P_{1|0} - K_1 P_{1|0} = 1000.0001 - 0.9998(1000.0001) = 0.2$$

Kalman Filters

- **2nd Iteration**

$$\hat{x}_{2|1} = \hat{x}_{1|1} = 99.15$$

$$P_{2|1} = P_{1|1} + 0.0001 = 0.2 + 0.0001 = 0.2001$$

$$K_2 = P_{2|1}(P_{2|1} + 0.25)^{-1} = 0.2001(0.2001 + 0.25)^{-1} = 0.4446$$

$$\hat{x}_{2|2} = \hat{x}_{2|1} + K_2(z_2 - \hat{x}_{2|1})$$

$$= 99.15 + 0.4446(100.60 - 99.15) = 99.79$$

$$P_{2|2} = P_{2|1} - K_2 P_{2|1} = 0.2001 - 0.4446(0.2001) = 0.1111$$

Kalman Filters

- **3rd Iteration**

$$\hat{x}_{3|2} = \hat{x}_{2|2} = 99.79$$

$$P_{3|2} = P_{2|2} + 0.0001 = 0.1111 + 0.0001 = 0.1112$$

$$K_3 = P_{3|2}(P_{3|2} + 0.25)^{-1} = 0.1112(0.1112 + 0.25)^{-1} = 0.3079$$

$$\hat{x}_{3|3} = \hat{x}_{3|2} + K_3(z_3 - \hat{x}_{3|2})$$
$$= 99.79 + 0.3079(100.12 - 99.79) = 99.89$$

$$P_{3|3} = P_{3|2} - K_3 P_{3|2} = 0.1112 - 0.3079(0.1112) = 0.0770$$

Kalman Filters

- **4th Iteration**

$$\hat{x}_{4|3} = \hat{x}_{3|3} = 99.89$$

$$P_{4|3} = P_{3|3} + 0.0001 = 0.077 + 0.0001 = 0.0771$$

$$K_4 = P_{4|3}(P_{4|3} + 0.25)^{-1} = 0.0771(0.0771 + 0.25)^{-1} = 0.2357$$

$$\hat{x}_{4|4} = \hat{x}_{4|3} + K_4(z_4 - \hat{x}_{4|3})$$

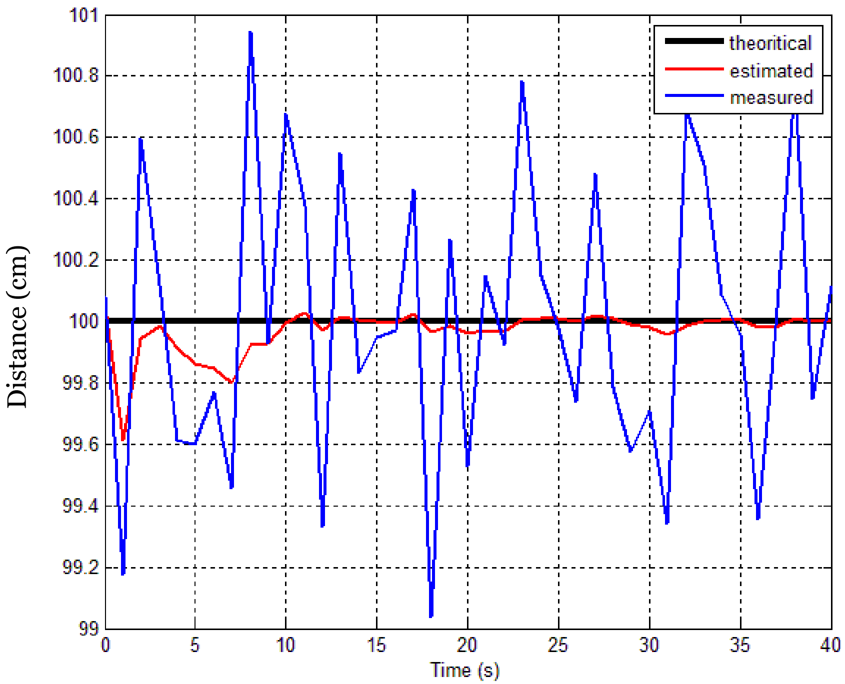
$$= 99.89 + 0.2357(99.61 - 99.89) = 99.82$$

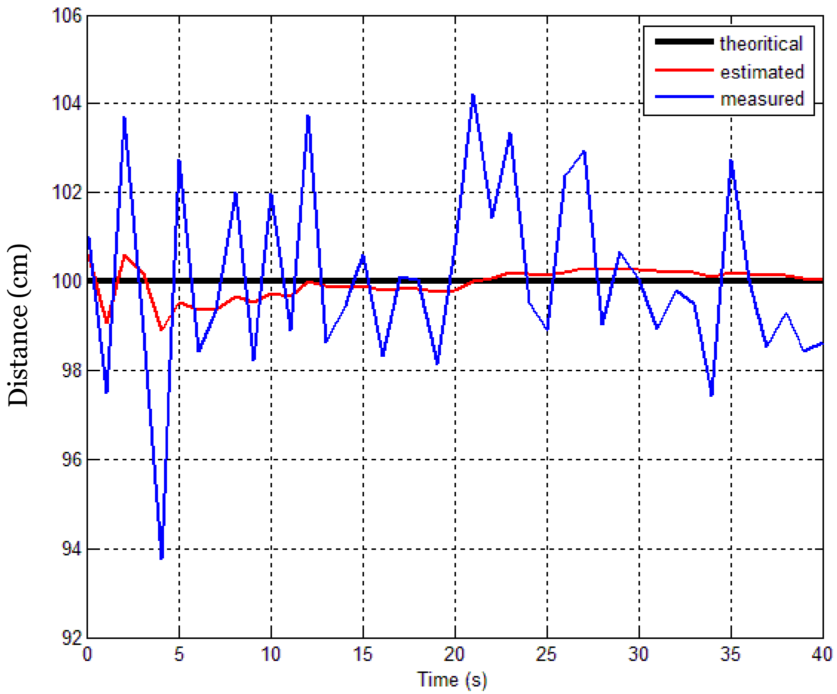
$$P_{4|4} = P_{4|3} - K_4 P_{4|3} = 0.0771 - 0.2357(0.0771) = 0.0589$$

Kalman Filters

Iteration	Measurements	State	Covariance	Gain
0	-	0	1000	0
1	99.17	99.15	0.2	0.9998
2	100.60	99.79	0.1111	0.4446
3	100.12	99.80	0.0770	0.3070
4	99.61	99.82	0.0589	0.2357

Kalman Filters

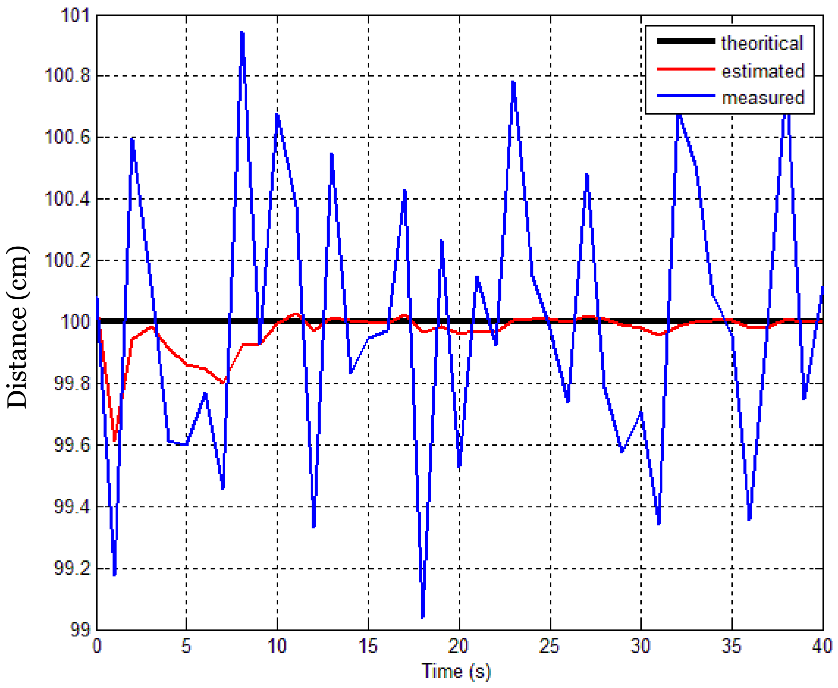




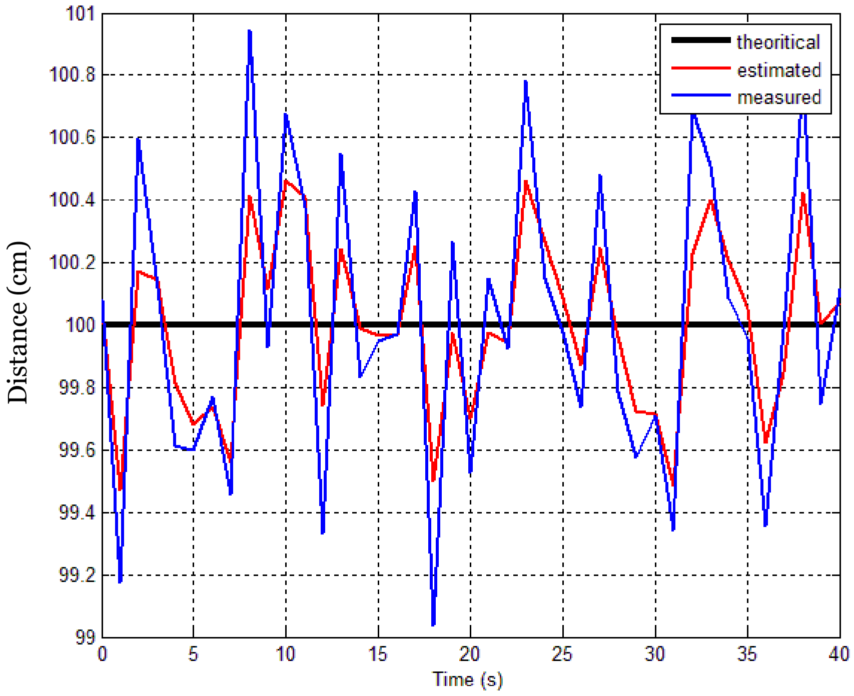
Measurement noise = 0.5 cm

Measurement noise = 2 cm

Kalman Filters



Process noise = 0.01 cm



Process noise = 0.5 cm

Kalman Filters

- What if our system is a **non-linear** system?!
  - ◇ Such as having a constant distance to the obstacle but the obstacle is not steady, it is "vibrating".
  - ◇ The vibration can be modeled as a sine wave with equation

$$L = c\sin(2\pi r\Delta t) + l$$

- In this case, we will have to use a non-linear estimator such as the **Extended Kalman Filter** (EKF).

For more information:

[1] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Janson, R. Karlsson and P. Nordlund, "Particle Filters for Positioning, Navigation and Tracking," in IEEE Transactions on Signal Processing.

[2] F. Germain and T. Skordas, "A Computer Vision Method for Motion Detection using Cooperative Kalman Filters".

See papers on the course website.

Outline

- Kalman Filters
- **Markov Models**
- Summary

Markov Models

- **Markov Property**

  If the random process is characterized as **memoryless**: the next state depends only on the current state and not on the sequence of events that preceded it. This specific kind of "**memorylessness**" is called the **Markov property**.

Russian mathematician Andrey Markov (1856-1922)

Future is independent of the past given the present

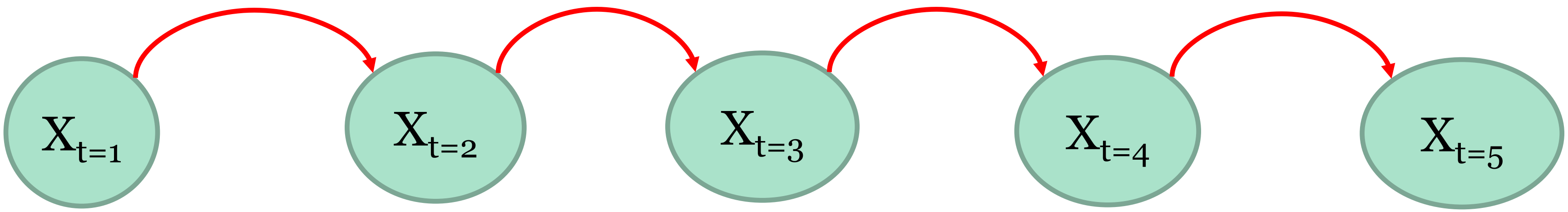
However, past can be used for learning or prediction

Markov Models

- **Markov Property**

Markov Property: The state of the system at time ***t+1*** depends only on the state of the system at time ***t***.

$$P[X_{t+1} = x_{t+1} \mid X_t = x_t , X_{t-1} = x_{t-1} , \ldots , X_1 = x_1 , X_0 = x_0] = P[X_{t+1} = x_{t+1} \mid X_t = x_t]$$



First order dependencies

Markov Models

- **Markov Property**
  - ◇ Higher order models remember more "**history**"
  - ◇ Additional history can have **predictive value**
  - ◇ ***Example:***
    - Predict the next word in this sentence fragment

1st order

"... the___" (duck, end, grain, tide, wall, ...?)

- Now predict it given more history

2nd order

"... against the___" (duck, end, grain, tide, wall, ...?)

- Now predict it given more history

3rd order

"swim against the___" (duck, end, grain, tide, wall, ...?)

Markov Models

- **Markov Properties-I: Limited horizon**
  - ◇ The probability that we're in state $\boldsymbol{s_i}$ at time $\boldsymbol{t+1}$ only depends on where we were at time $\boldsymbol{t}$:

$$P(X_{t+1} = s_i \mid X_1 \ldots X_t) = P(X_{t+1} = s_i \mid X_t)$$

  - ◇ Given this assumption, the probability of any sequence is just:

$$P(X_1, \ldots, X_T) = \prod_{i=1}^{T} P(X_i \mid X_{i-1})$$

Markov Models

- **Markov Property-II: Stationary Assumption**

Probabilities are **independent of *t*** when process is "**stationary**" so,

for all t, $P[X_{t+1} = X_j \mid X_t = X_i] = p_{ij}$

This means that if system is in **state *i***, the probability that the system will next move to **state *j*** is p_{ij} , no matter what the value of ***t*** is.

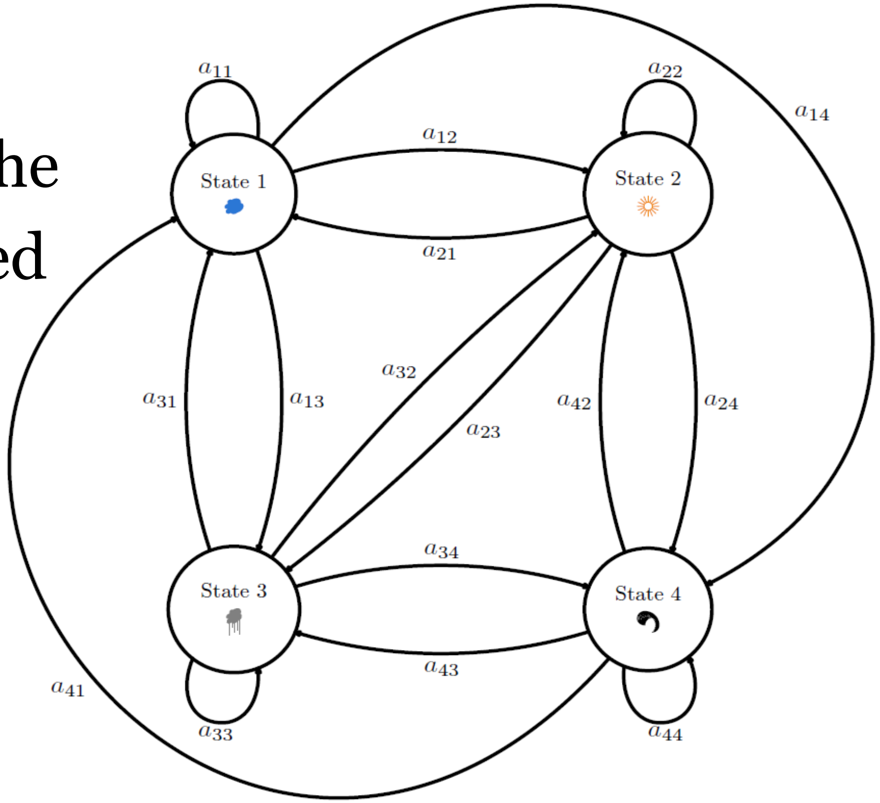
The **probability** of being in state $\boldsymbol{s_i}$ given the previous state **does not change over time**.

Markov Models

- **Weather Predictor Example**

  Assume that once a day (e.g. in the morning), the weather is observed as being one of the following:

  - **State 1:** cloudy
  - **State 2:** sunny
  - **State 3:** rainy
  - **State 4:** windy



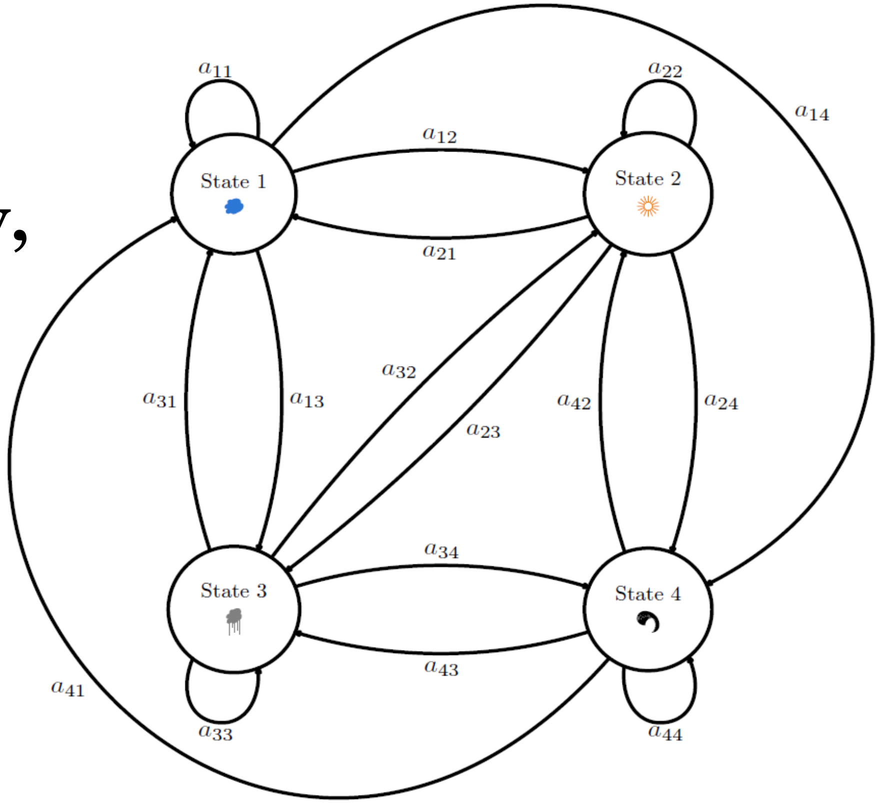
Given the model, it is now possible to answer several interesting questions about the weather patterns over time.

[3]

Markov Models

- **Weather Predictor Example**

  What is the probability to get the sequence "**sunny, rainy, sunny, windy, cloudy, cloudy**" in six consecutive days?



O={sunny, rainy, sunny, windy, cloudy, cloudy)={2,3,2,4,1,1}

Markov model

$$(O|A,\pi) = P(2,3,2,4,1,1|A,\pi)$$
$$= P(2)P(3|2)P(2|3)P(4|2)P(1|4)P(1|1)$$
$$= \pi_{x2}.\, a_{23}.\, a_{32}.\, a_{24}.\, a_{41}.\, a_{11}$$

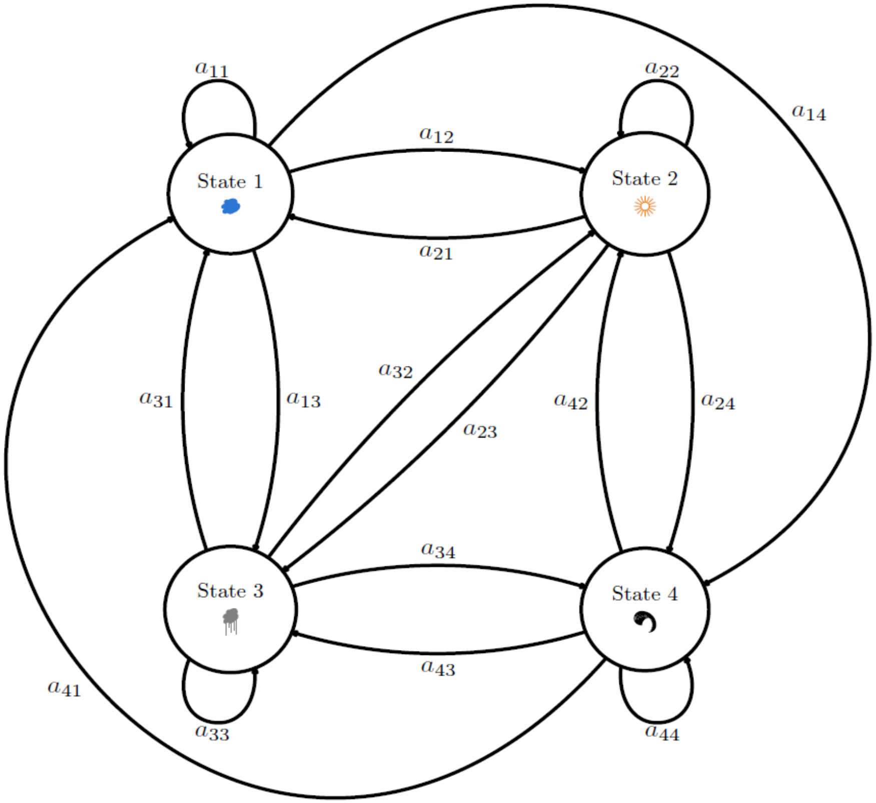
where A and π are transition matrix and initial state respectively. [3]

Markov Models

- **Weather Predictor Example**

  In a **general case**, this calculation of the probability for a state sequence **$\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_T)$** will be:

$$P(X|A,\pi) = \pi_{x_1}.a_{x_1x_2}.a_{x_2x_3}\ldots a_{x_{T-1}x_T}$$



[3]

Markov Models

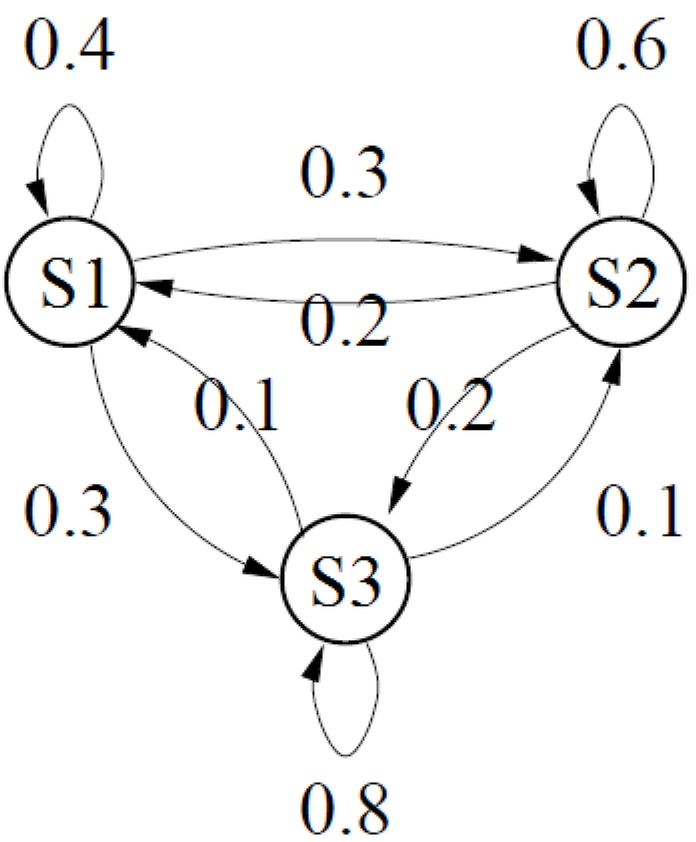
- **Weather Predictor Example**

Given that the weather on **day 1 is sunny**, what is the probability (according to the model) that the weather for the next 6 days will be

"**sunny-sunny-rainy-cloudy-cloudy-sunny**"

Given:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

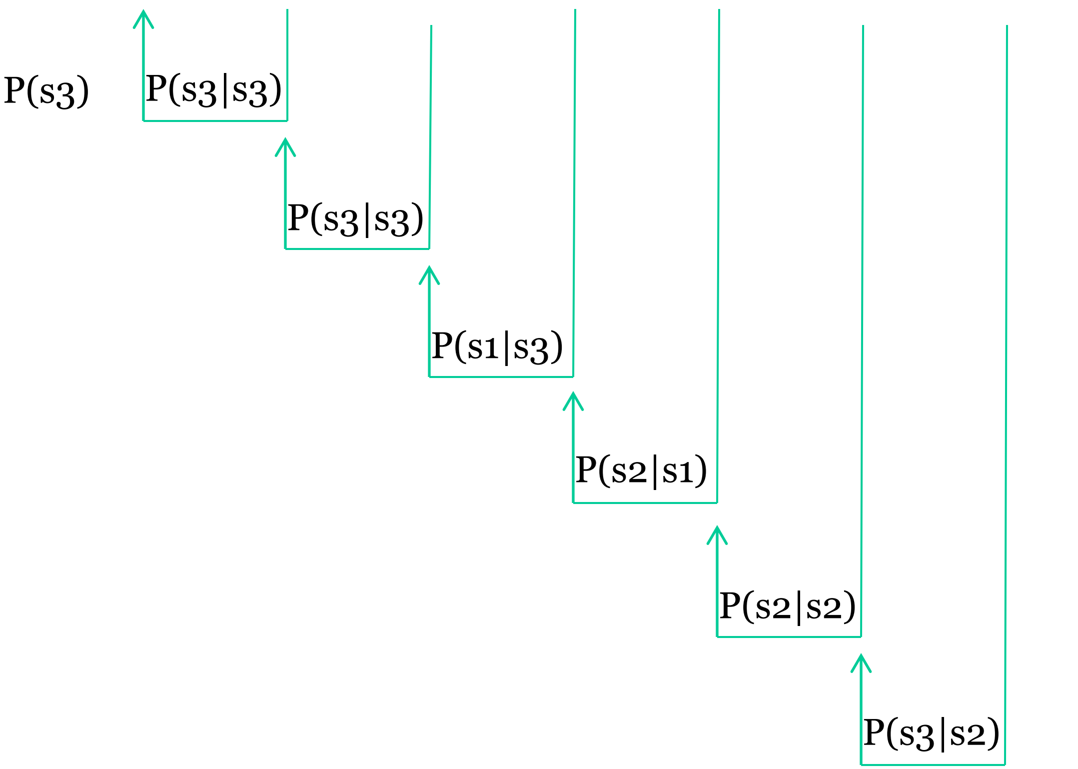


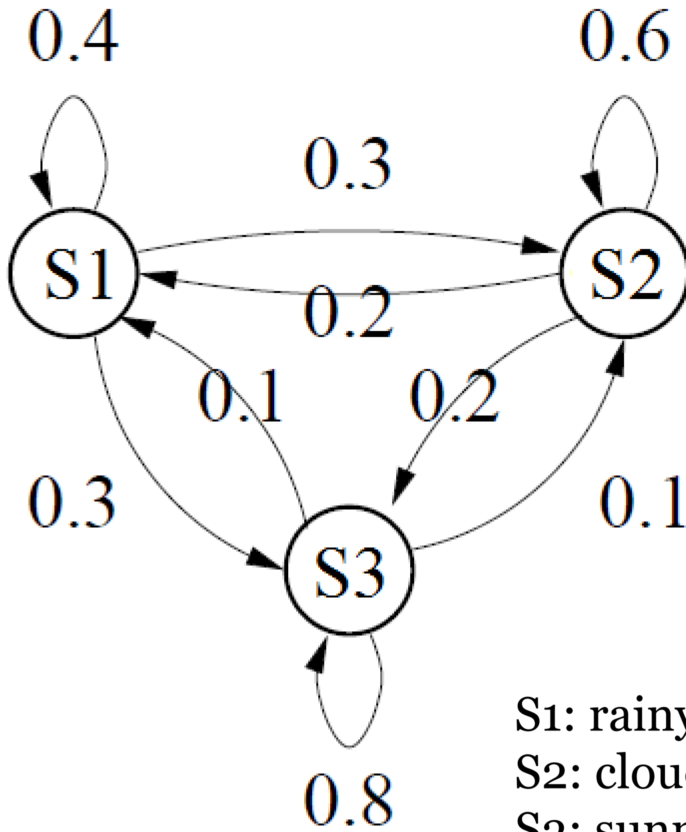
S1: rainy
S2: cloudy
S3: sunny

Markov Models

- **Weather Predictor Example**

S3→S3 →S3 →S1 →S2 →S2 →S3





S1: rainy
S2: cloudy
S3: sunny

$$P(X|M)=P(s_3)\,P(s_3|s_3)^2P(s_1|s_3)\,P(s_2|s_1)\,P(s_2|s_2)\,P(s_3|s_2)$$

$$=1\times 0.8^2\times 0.1\times 0.3\times 0.6\times 0.2$$

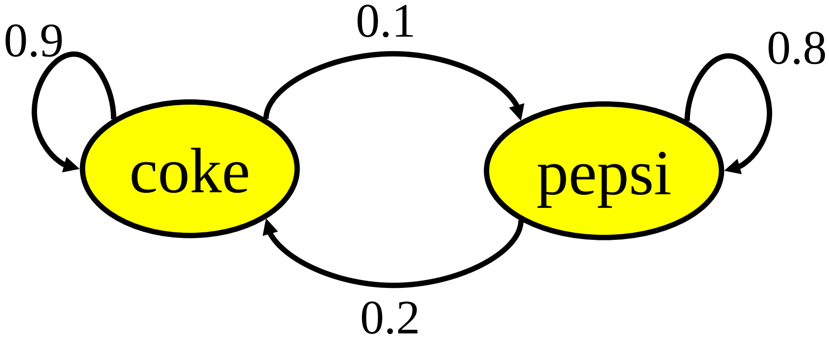
Initial state probability for state i is $\pi_i = P(x_1 = i)$

Markov Models

- **Coke vs. Pepsi Example**

Given that a person's last cola purchase was Coke, there is a 90% chance that his/her next cola purchase will also be Coke.

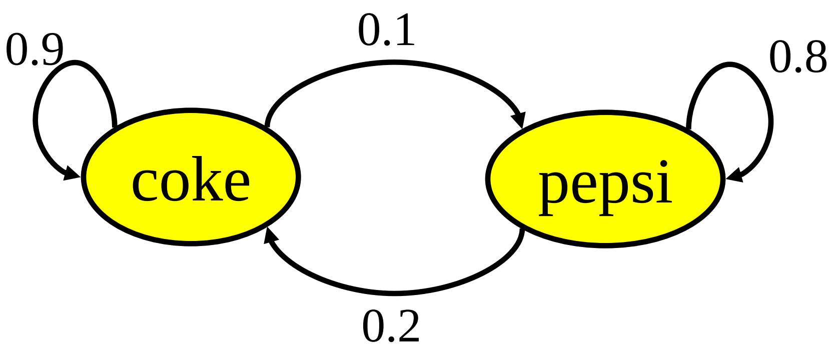
If that person's last cola purchase was Pepsi, there is an 80% chance that his/her next cola purchase will also be Pepsi.



Markov Models

- **Coke vs. Pepsi Example**

  Given that a person is currently a **Pepsi** purchaser, what is the probability that she will purchase **Coke two purchases** from now?

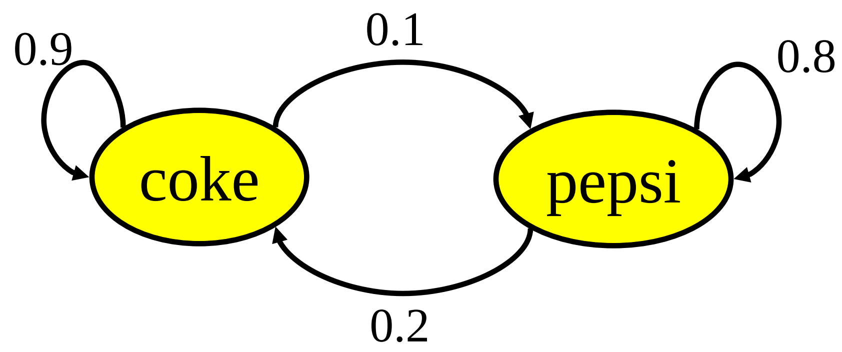


$$p(X|M) = p(X = \{P,C,C\}|M)$$
$$= p(P)\, p(C|P) p(C|C)$$
$$=1\times 0.2\times 0.9=0.18$$

Markov Models

- **Coke vs. Pepsi Example**

  Given that a person is currently a **Coke** drinker, what is the probability that she will purchase **Pepsi three purchases** from now?



$$p(X|M) = p(X = \{C,P,P,P\}|M)$$
$$= p(C)\, p(P|C) p(P|P)\, p(P|P)$$
$$=1\times 0.1\times 0.8\times 0.8=0.064$$

Markov Models

- A Markov model is a **probabilistic model** of symbol sequences in which the probability of the current event is conditioned only by the previous event.

- **Markov Models**

System*	**System state is fully observable**	**System state is partially observable**
System is autonomous	Markov Chain (MC)	Hidden Markov Model (HMM)
System is controlled	Markov Decision Process (MDP)	Partially Observable Markov Decision Process (POMDP)

*whether the system is to be adjusted on the basis of observations made.

Markov Models

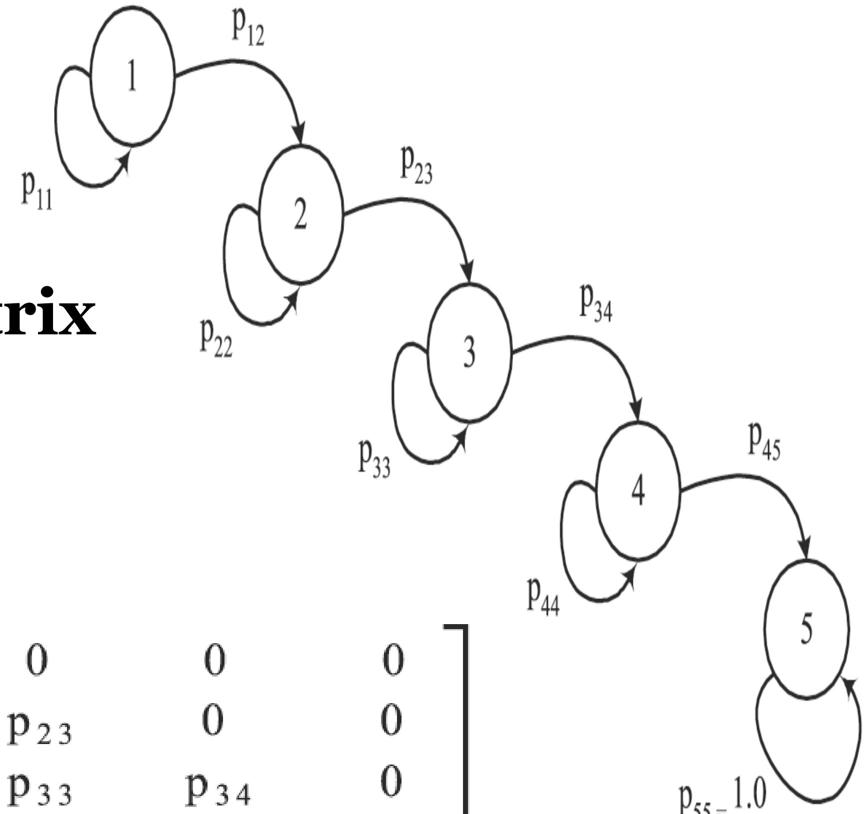
- **Markov Chain**

  Markov chain is a “**memoryless** random process”

  Transitions depend only on

  ◇ **current state** and

  ◇ **transition probabilities matrix**



$$P = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 & 0 \\ 0 & p_{22} & p_{23} & 0 & 0 \\ 0 & 0 & p_{33} & p_{34} & 0 \\ 0 & 0 & 0 & p_{44} & p_{45} \\ 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}$$

Markov Models

- **Markov Chain: Formal Definition**
  - ◇ A Markov Model is a triple **(S, π, A)** where:
    - – **S** is the set of **states**
    - – **π** are the **probabilities** of being initially in some state
    - – **A** are the **transition probabilities**.

Markov Models

- **Markov Chain: Economy Example**

The terms **bull market** and **bear market** describe upward and downward market trends, respectively. The states represent whether the economy is in a **bull market**, a **bear market**, or a **recession**, during a given week.

S={1=bull market,

2=bear market,

3=recession}

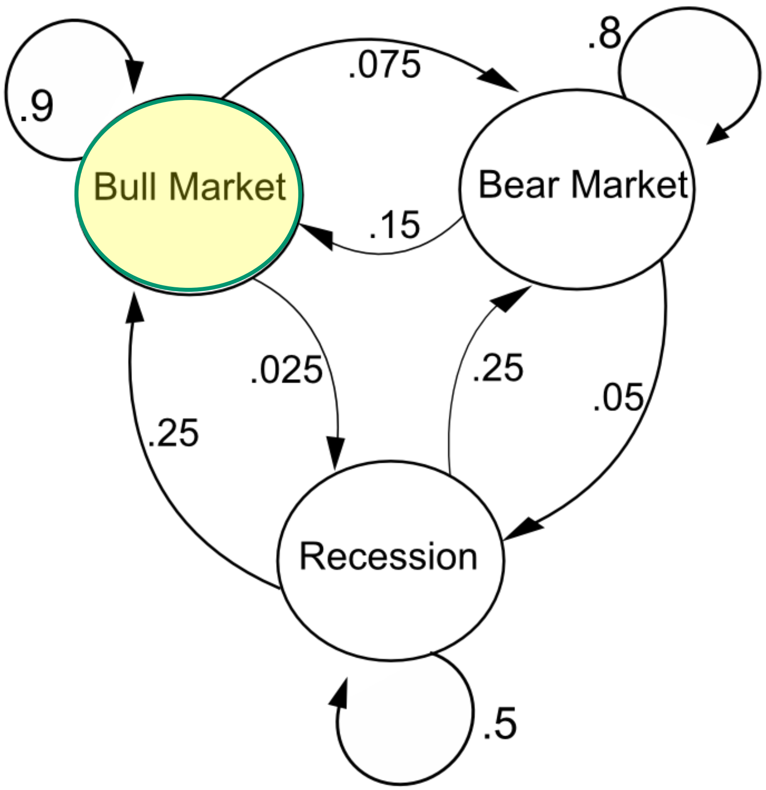
Statues of the two symbolic beasts of finance, the bear and the bull, in front of the Frankfurt Stock Exchange.

[4]

Markov Models

- **Markov Chain: Economy Example**

According to the figure, a **bull week** is followed by another bull week 90% of the time, a **bear market** 7.5% of the time, and a **recession** the other 2.5%.



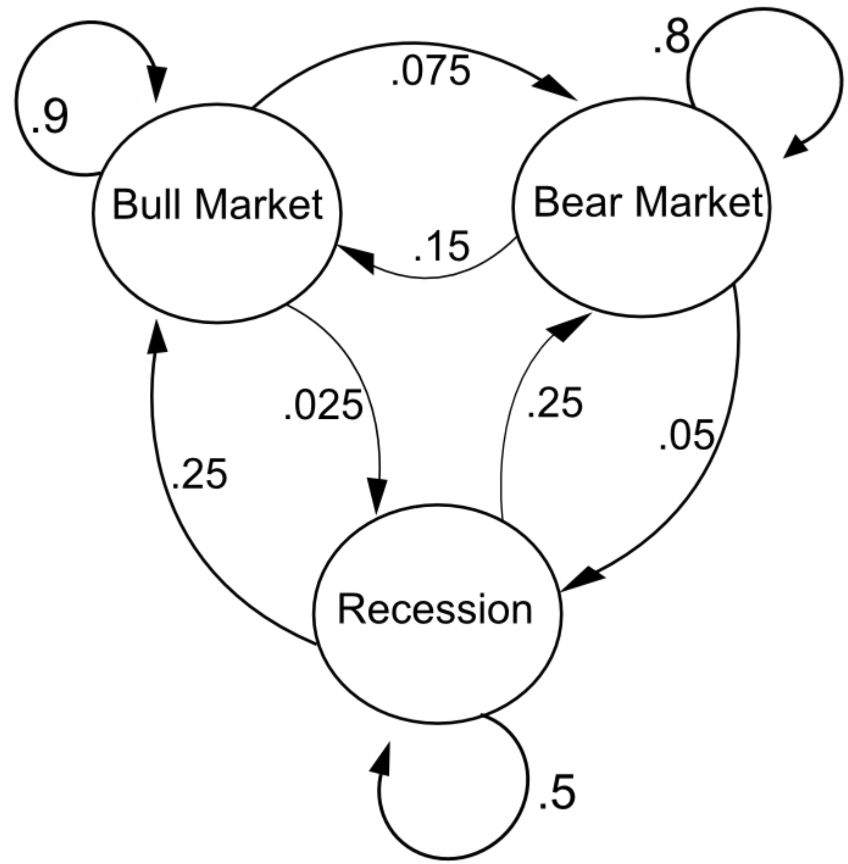
[4]

Markov Models

- **Markov Chain: Economy Example**

The **transition matrix** is

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

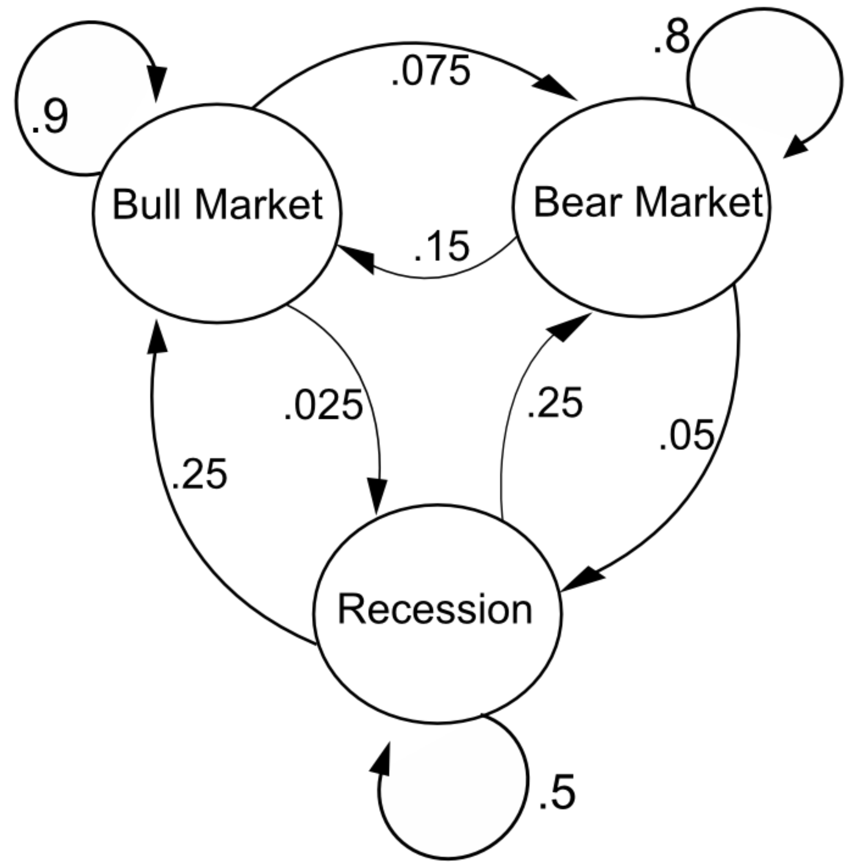


[4]

Markov Models

- **Markov Chain: Economy Example**

  From this figure it is possible to calculate, for example, the **long-term fraction of time** during which the economy is in a **recession**, or on average **how long** it will take to go from a **recession** to a **bull market**.



[4]

Markov Models

- **Markov Chain: Economy Example**

Using the transition probabilities,

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

We will regard bull market as 1, bear market as 2 and recession as 3.

$$0.9P_1 + 0.075P_2 + 0.025P_3 = P_1$$

$$0.15P_1 + 0.8P_2 + 0.05P_3 = P_2$$

$$0.25P_1 + 0.25P_2 + 0.5P_3 = P_3$$

The steady-state probabilities indicate that **62.5%** of weeks will be in a **bull** market, **31.25%** of weeks will be in a **bear** market and **6.25%** of weeks will be in a **recession**.

[4]

Markov Models

- **Markov Chain: Economy Example**

The distribution over states can be written as a stochastic row vector x with the relation $\boldsymbol{x^{(n+1)} = x^{(n)}P}$.

So if at time $\boldsymbol{n}$ the system is in state **2=bear** then **3 time** periods later at time $\boldsymbol{n+3}$ the distribution is

$$x^{(n+3)} = x^{(n+2)}P = \left(x^{(n+1)}P\right)P = \left(x^{(n)}P^2\right)P = x^{(n)}P^3$$

$$= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^3$$

$$= \begin{bmatrix} 0.3575 & 0.56825 & 0.07425 \end{bmatrix}$$

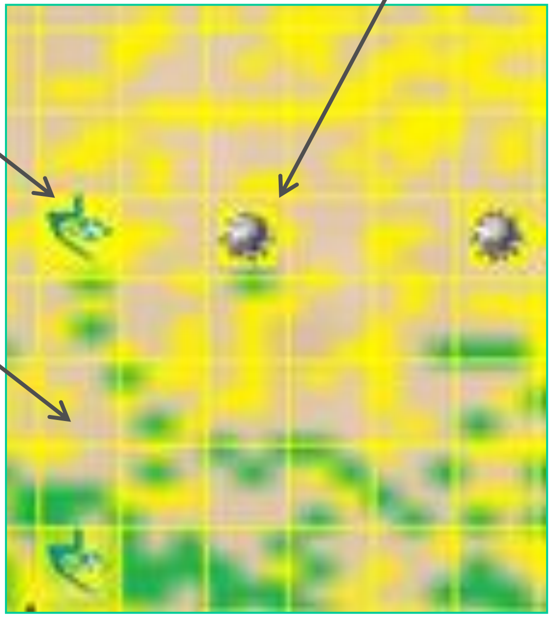
[4]

Markov Models

- **Markov Chain: Landmine Detection**

Suppose a landmine detection robot wants to predict the status of a cell in a minefield. The possible predictions are:

- Mine_free
- Surface_mine
- Buried_mine



Minesweepers: Towards a Landmine-free World: http://www.landminefree.org/ [6]

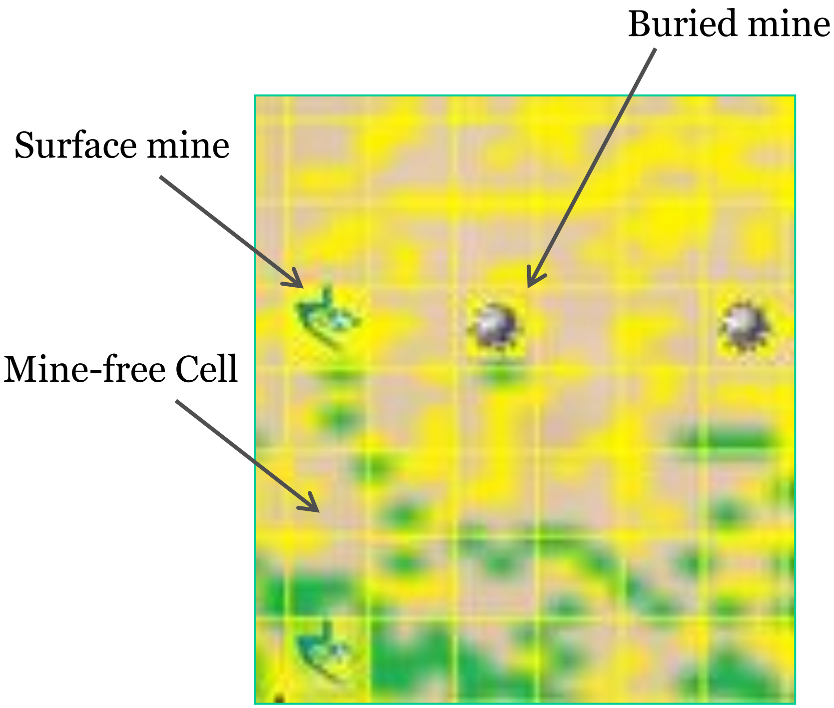
[5]

Markov Models

- **Markov Chain: Landmine Detection**

The robot predicts the next cell status based on the status of the previous cell.

- If the previous cells were **mine-free**, the next cell is **likelier to have surface or buried mine**.
- How far back do we want to go to predict next cell's status?



Markov Models

- **Markov Chain: Landmine Detection**
  - Statistical Landmine Model
    - ◇ Notation:
      - – ***S***: the **state space**, a set of possible values for the cell: {mine_free, surface, buried}
      - – ***X***: a sequence of random variables, each taking a value from S
      - – ***k*** is an integer standing for cells, $k \in [1,N]$
    - ◇ $(X_1, X_2, X_3, \ldots X_N)$ models the value of a series of random variables
      - – each takes a value from ***S*** with a certain probability $P(X=s_i)$
      - – the entire sequence tells us the status of cell over N cells.

Markov Models

- **Markov Chain: Landmine Detection**
  - Statistical Landmine Model
    - ◇ If we want to predict the **status of the cell** *k+1*, our model might look like this:

$$P(X_{k+1} = s_k \mid X_1 \ldots X_k)$$

    - ◇ e.g. **P(cell status = Buried_mine)**, conditional on the status of the **past *k* cells**.
    - ◇ **Problem:** the larger *k* gets, the more calculations we have to make.

Markov Models

- **Markov Chain: Landmine Detection**
  - Concrete instantiation

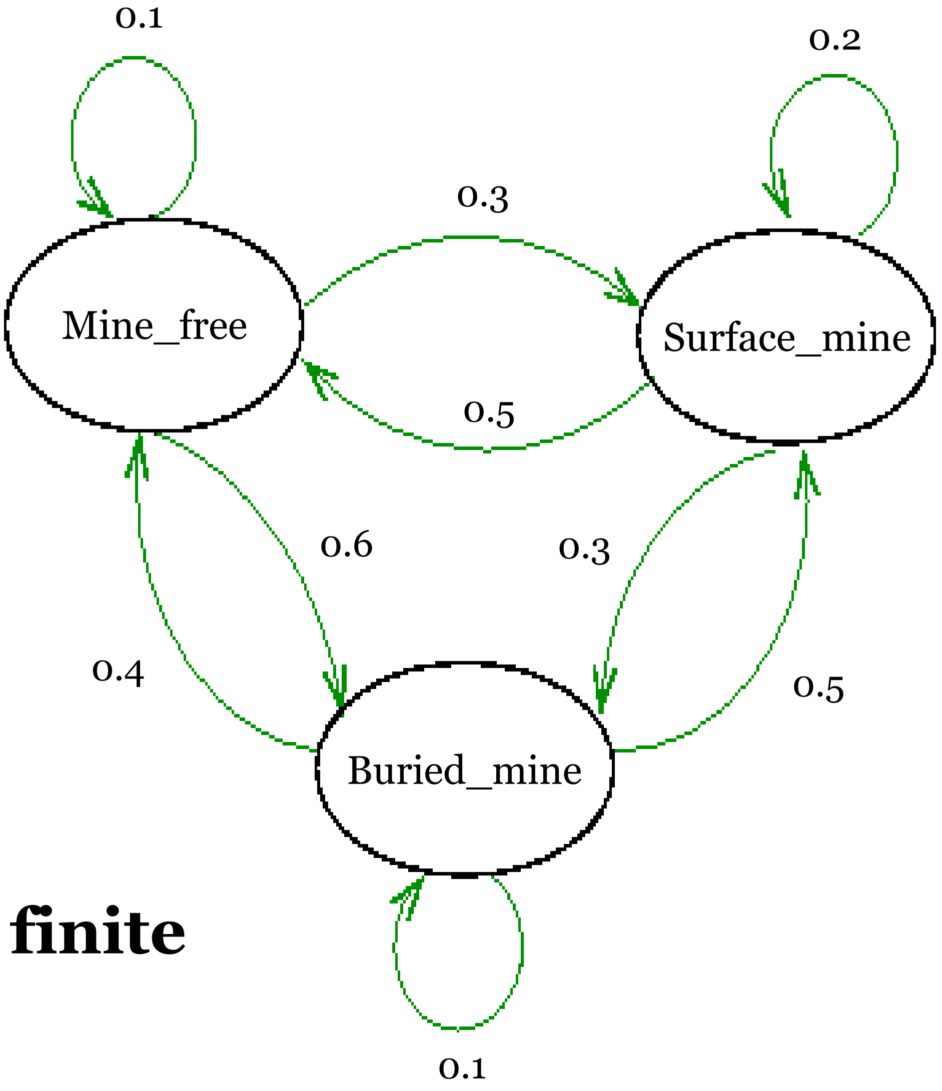
Cell k	Cell k+1		
	Mine_free	Surface_mine	Buried_mine
Mine_free	0.1	0.3	0.6
Surface_mine	0.5	0.2	0.3
Buried_mine	0.4	0.5	0.1

◇ This is essentially a **transition matrix**, which gives us probabilities of going from one state to the other.

◇ We can denote state transition probabilities as a_{ij} (prob. of going from state i to state j).

Markov Models

- **Markov Chain: Landmine Detection**
  - Graphical View
    - ◇ Components of the model:
      1. states (s)
      2. transitions
      3. transition probabilities
      4. initial probability distribution for states

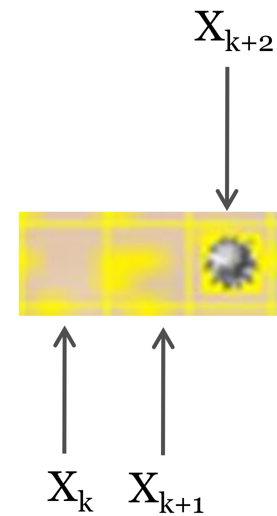


This is a **non-deterministic finite state automaton**.

Markov Models

- **Markov Chain: Landmine Detection**
  - If the **cell (X_k) is Mine-free**, what's the probability that the **next cell (X_{k+1}) is Mine-free** and the **next cell after (X_{k+2}) is Buried-mine**?

$$P(X_{k+1} = \text{Mine\_free}, X_{k+2} = \text{Buried\_mine} \mid X_k = \text{Mine\_free})$$



Markov assumption

$$= P(X_{k+1} = \text{Mine\_free} \mid X_k = \text{Mine\_free}) \times P(X_{k+2} = \text{Buried\_mine} \mid X_{k+1} = \text{Mine\_free}, \cancel{X_k = \text{Mine\_free}})$$

$$= P(X_{k+1} = \text{Mine\_free} \mid X_k = \text{Mine\_free}) \times P(X_{k+2} = \text{Buried\_mine} \mid X_{k+1} = \text{Mine\_free})$$

$$= 0.1 \times 0.6 = 0.06$$

Markov Models

- In all previous examples, we assume that the **state is fully observable**.
- Often we face scenarios where states cannot be directly observed.
- To handle partially observable state, we have to use an extension called **Hidden Markov Model (HMM).**

Outline

- Kalman Filters
- Markov Models
- **Summary**

Summary

- **Kalman filter** is an optimal estimator – i.e infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive. (cf batch processing where all data must be present). If all noise is Gaussian, the Kalman filter minimizes the mean square error of the estimated parameters. KF can be used for Robot Localization and Map building from range sensors/ beacons, determination of planet orbit parameters from limited earth observations and Tracking targets - eg aircraft, missiles using RADAR.

- A Markov model is a **probabilistic model** of symbol sequences in which the probability of the current event is conditioned only by the previous event.

References

1. Christophe Couvreur. *Hidden Markov Models and Their Mixtures*. Diploma of Advanced Studies, Universitee Catholique de Louvain, 1996.
2. Wikipedia: http://en.wikipedia.org/wiki/Hidden_Markov_model
3. Mark Gales and Steve Young, "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends in Signal Processing, Vol. 1, No. 3 (2007) 195–304.
4. Steve Young, Dan Kershaw, Julian Odell, Dave Ollason, Valtcho Valtchev and Phil Woodland. *The HTK Book*. Version 3.1, Microsoft Corporation, 1999.
5. Victor Lavrenko and Nigel Goddard. Introductory Applied Machine Learning. University of Edinburgh, UK, 2011.