CHEN4011
Advanced Modelling and COntrol

Curtin University

Dr. Ranjeet Utikar (RU)

Curtin University Malaysia

Dr. Jobrun Nandong (JN)

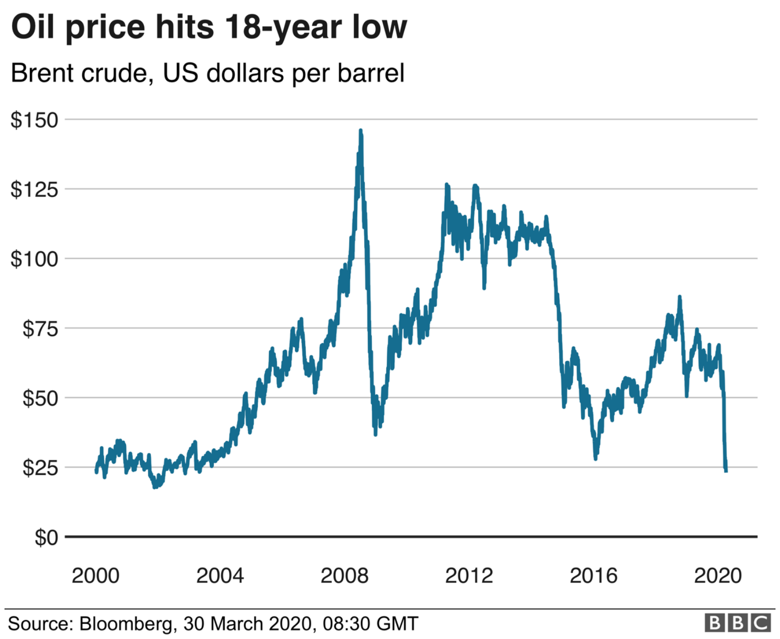
Lecture Note 8
Time Series Modelling and Analysis

Outline

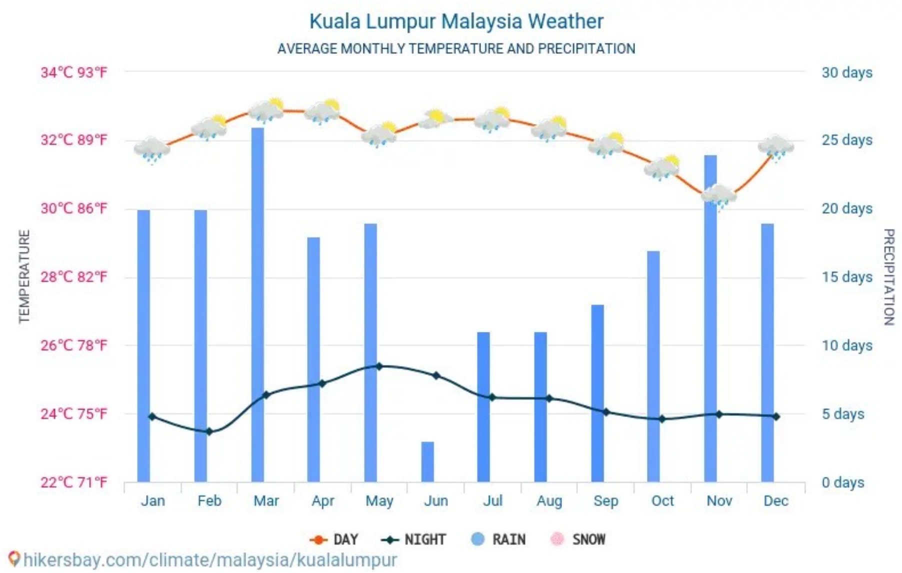
- Introduction to time series
- Stationary vs unstationary behaviours in time series data
- Autoregressive (AR) model
  - AR in MATLAB
- Autoregressive- Exogeneous (ARX) model
  - ARX in MATLAB
- Autoregressive Moving Average (ARMA) model
  - ARIMA in MATLAB

Introduction

- Time-series data consists of a number of observations ordered in time
- Observations (measurements) are often equally spaced, e.g., by day, week, month, etc.
- Examples of time series data
  - Gross domestic product (GDP)
  - Unemployment rate
  - Oil price
  - Building temperature, etc.
- One-way ordering of time – a future value can be expressed in terms of historical values.



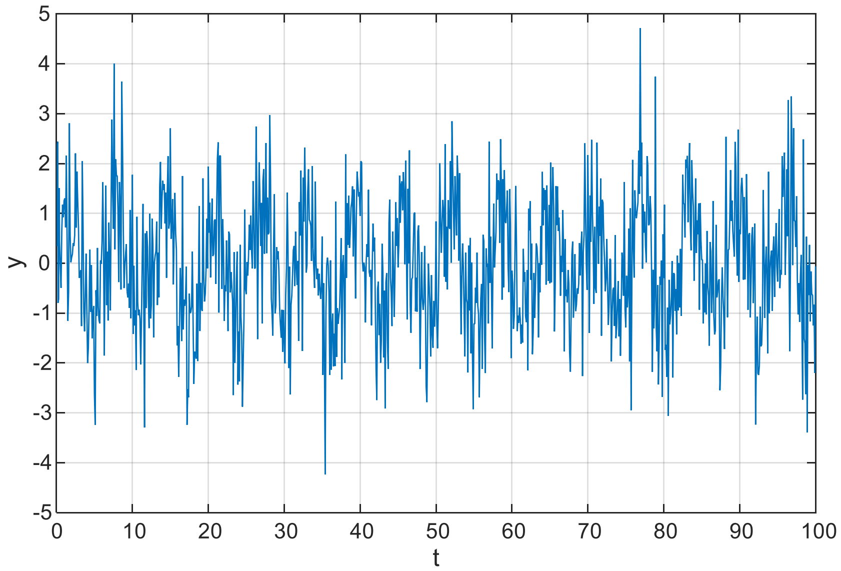
Oil price hits 18-year low
Brent crude, US dollars per barrel
$150
$125
$100
$75
$50
$25
$0
2000
2004
2008
2012
2016
2020
Source: Bloomberg, 30 March 2020, 08:30 GMT



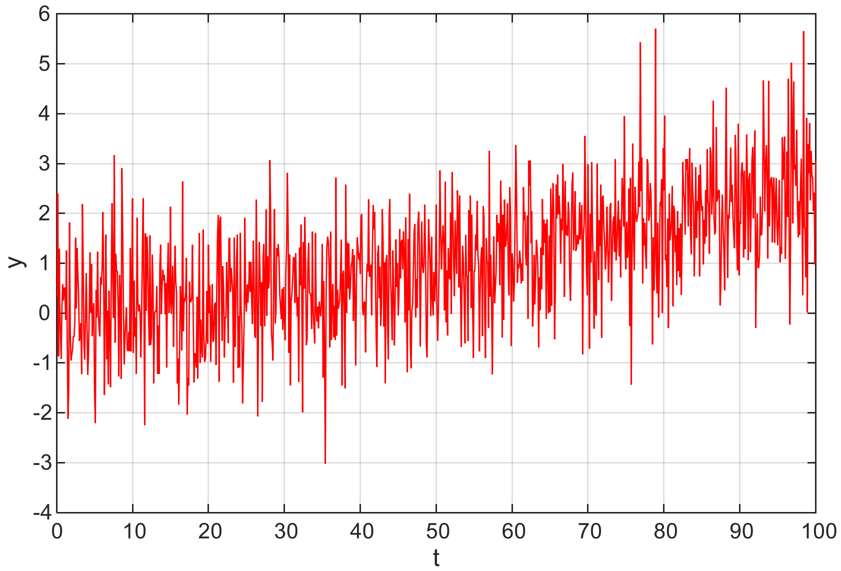
Kuala Lumpur Malaysia Weather
AVERAGE MONTHLY TEMPERATURE AND PRECIPITATION
34°C 93°F
32°C 89°F
30°C 86°F
28°C 82°F
26°C 78°F
24°C 75°F
22°C 71°F
TEMPERATURE
30 days
25 days
20 days
15 days
10 days
5 days
0 days
PRECIPITATION
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
DAY
NIGHT
RAIN
SNOW
hikersbay.com/climate/malaysia/kualalumpur

Stationary vs Non-stationary

- Stationary behaviour
- Mean is at zero



- Non-stationary behaviour
- Mean is varying with time

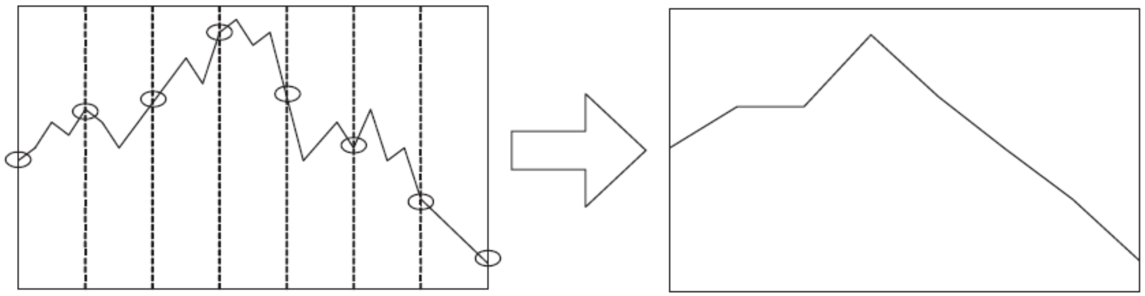


Time series representation

- The nature of time series data includes: large in data size, high dimensionality and update continuously.
- Time series data is characterized by its numerical and continuous nature, is always considered as a whole instead of individual numerical field.
- Unlike traditional databases where similarity search is **exact match** based, ***similarity*** search in time series data is typically carried out in an ***approximate manner***.
- The fundamental problem is **how to represent** the time series data
- Based on the time series representation, different mining tasks can be done:
  i. Pattern discovery and clustering
  ii. Classification
  iii. Rule discovery
  iv. Summarization.

Time series representation and indexing

- One of the reasons of time series representation is to reduce the dimension (i.e., number of data points)



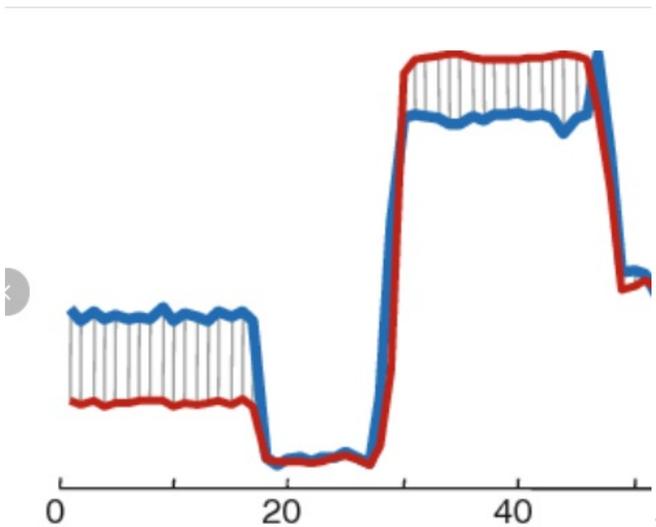
Resampling of the time series data

- Data reduction by resampling can cause distortion of the resampled data

Similarity measure

- Similarity measure is important for a variety of time series analysis and data mining tasks
- To measure the similarity/dissimilarity between two time series, the most popular approach is to evaluate the **Euclidean distance** on the transformed representation

Euclidian distance between the two time-series is the square-root of the sum of square length of the hatch lines.



Time series decomposition

- Goal in analysis is to **decompose a series** into a set of non-observable (latent) components which can be associated to different types of temporal variations
- Note: 17th century astronomers used time series decomposition to calculate the planetary orbits
- 4 types of fluctuations
  i. Long-term tendency
  ii. Cyclical movements
  iii. Seasonal movements
  iv. Residual variations due to, e.g., war and pandemic

Mining in time series

- Mining is to discover ***hidden information*** or knowledge from either the original or the transformed time series data.
- Pattern discovery is the most common mining task
- The clustering method is the most commonly used in the pattern discovery
- The discovery of interesting patterns is an important data mining task that is applicable in many domains
- The discovered rules and patterns can be used to build ***forecasting models*** that are able to predict future developments

What is Model?

- A model called ***structural*** if its parameters has natural or **structural interpretation**
  - The model can provide ***explanation*** and ***control*** of the process generating the data
- When no models are available for a data set from theory or experience, it is still possible to fit models which suffice for:
  - **Simulation** (from what has been observed, generate more data similar to that observed),
  - **Prediction** (from what has been observed, forecast the data that will be observed), and
  - **Pattern recognition** (from what has been observed, infersignificant characteristics of the process generating the data such as significant time lags, significant, frequencies, extractable signals, and noise)
- When a model is ***not structural*** it is called ***synthetic***, and its parameters are called *synthetic parameters*

Autoregression (AR) Model

- Assume the present output value depends on the past output values in discreet time
- AR model is expressed as follows

$$\boldsymbol{y_t = \mathrm{c} + \sum_{i=1}^{n} \alpha_i y_{t-i} + \varepsilon_t}$$

Where c is a constant, α_i is a model parameter, n is the model order, and ε_t is the white noise (or error)

- Eg., for $p = 2$, the corresponding AR model is

$$y_t = c + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \varepsilon_t$$

- Value of output at t is given by the two historical values which 1 and 2 steps before the present value

AR model with back shift operator z^{-k}

$$\boldsymbol{y_t = c + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \varepsilon_t}$$

- This model can also be written as follows

$$y_t = c + (\alpha_1 z^{-1} + \alpha_2 z^{-2}) y_t + \varepsilon_t$$

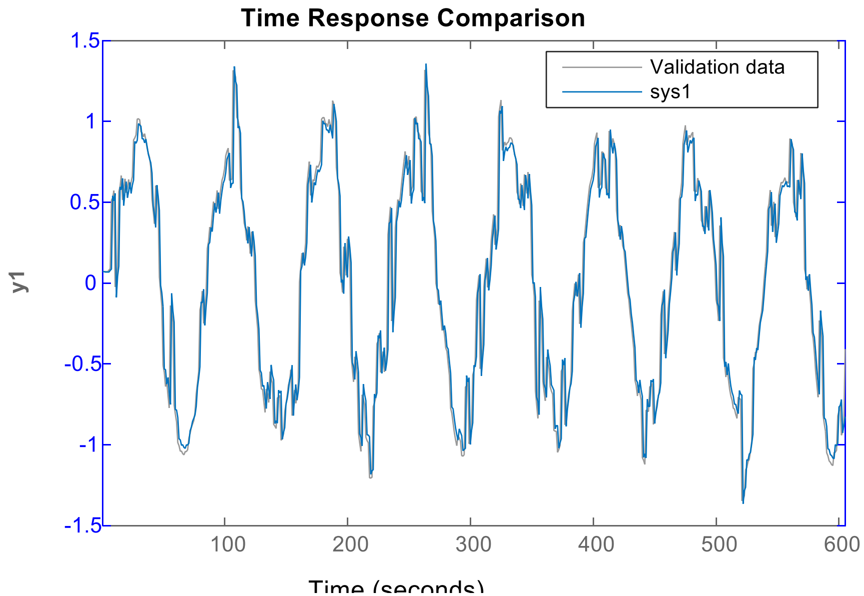
$$\Longrightarrow\ (1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}) y_t = c + \varepsilon_t$$

$$\therefore\ \boldsymbol{y_t = \frac{c + \varepsilon_t}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}} = \frac{c + \varepsilon_t}{A(z)}}$$

Where $A(z) = 1 +\ \alpha_1 z^{-1} + \alpha_2 z^{-2}$

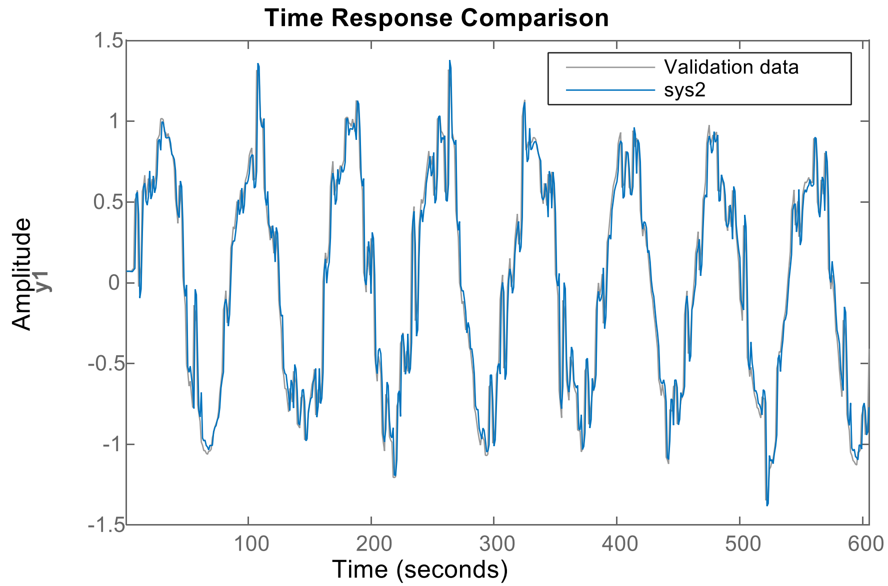
- An all-pole infinite impulse response (IIR) filter driven by the white noise as input
  - Finite impulse response (FIR) system - the impulse response does become exactly zero at times t > T for some finite T

Example 1 – AR model order



- $n = 2$
- Accuracy 74%
- Model is given by

$$A(z) = 1 - 1.073z^{-1} + 0.111z^{-2}$$



- $n = 4$
- Accuracy 75%
- Model is given by

$$A(z) = 1 - 1.093z^{-1} + 0.0061z^{-2} + 0.443z^{-3} - 0.328z^{-4}$$

Example 2 – Malaysia COVID-19 Infection

Infection Data 2020

29-Feb	25
1-Mar	29
2-Mar	29
3-Mar	36
4-Mar	50
5-Mar	55
6-Mar	83
7-Mar	93
8-Mar	99
9-Mar	117
10-Mar	129
11-Mar	149
12-Mar	158

1) Open Matlab
2) COVID-19 cumulative infection from **26/01/2020 to 30/04/2020** used to build an AR model
3) Check the projection using the AR model with data from **01/05/2020**

- On Matlab Command Window, copy the data from Excel and paste into the [ ]. Type as follows:

>> X = [ ]; % paste the data into the [ ], then press enter.

- Invoke the '*ar*' built-in function in Matlab

>> Sys1 = **ar**(X,2); % n = 2

- Type and enter as follows

>> Sys1

Sys1 =

Discrete-time AR model: A(z)y(t) = e(t)

A(z) = 1 - 1.932 z^-1 + 0.9322 z^-2

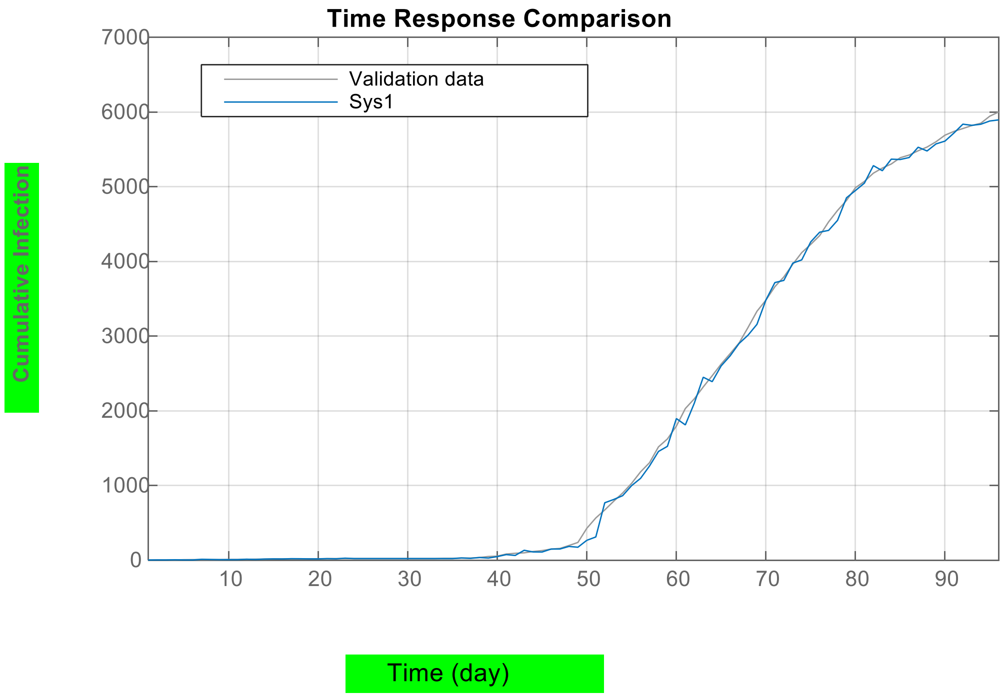
- Model is

$A(z) = 1 - 1.932z^{-1} + 0.9322z^{-2}$

- To compare the model and data, use the built-in '**compare**' function

>> **compare**(X,Sys1,2); % M = 2 is the prediction horizon, where data up to t – M is used to predict the output of Sys1

Example 2 – Malaysia COVID-19 Infection



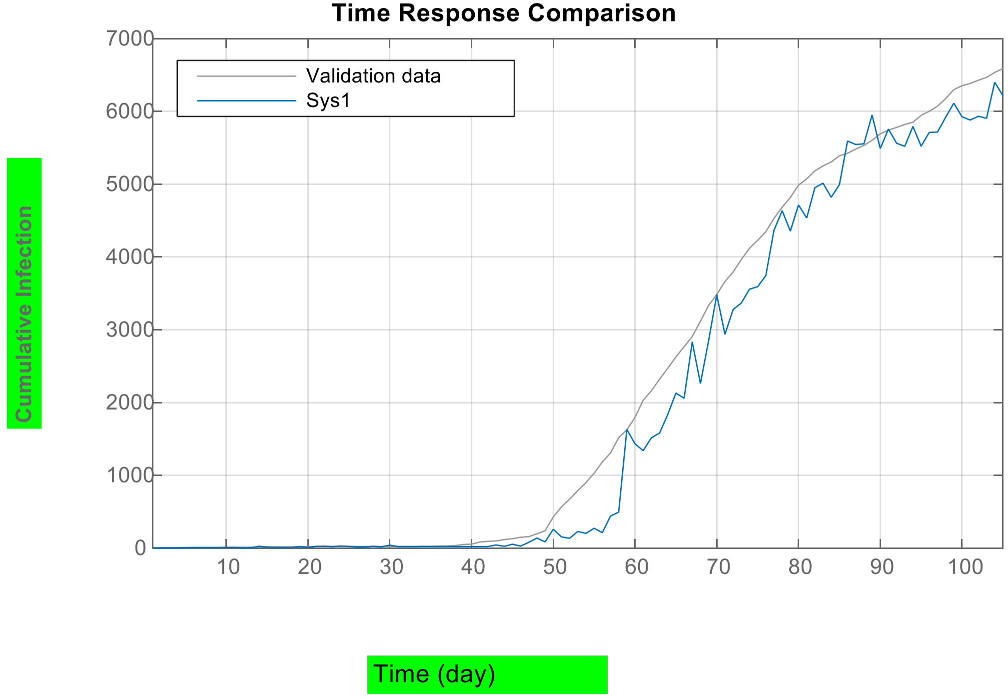
- The 2nd order AR model fit the infection data well (97% fitness)
- The same data used to build the model is used for the prediction
- How accurate the model prediction will be if it is used to forecast the data beyond 30/04/2020
- Let include the data up to 09/05/2020 in X dataset.
- 9 data points added.

Example 2 – Malaysia COVID-19 Infection

- Copy and past the entire dataset (including 9 extra points) onto Matlab Command Window
- Type as follows

>> **compare**(X,Sys1,9);

- Use m = 9, because we want to predict the 9 data points added using the AR model
- Fitness drop to 85%
- Longer prediction, poorer model fitness.
- For m = 2, 3, 4, 5 and 6 the fitness values are 97%, 96%, 94%, 93% and 91% respectively.



ARX model

- Is a linear equation for the present output value as a function of the past output and input values in discrete time
- Single input and single output ARX structure without input delay:

$$\boldsymbol{y_t + \sum_{i=1}^{p} \alpha_i y_{t-i} = \sum_{i=0}^{q} \beta_i u_{t-i} + \varepsilon}$$

- Can be expressed using the back shift operator:

$$\boldsymbol{y(t)A(z) = B(z)u(t) + \varepsilon}$$

Where $A(z) = 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + a_p z^{-p}$ and $B(z) = \beta_1 + \beta_2 z^{-1} + \cdots + \beta_q z^{-q+1}$

- For system with input delay with magnitude n_k:

$$y_t + \sum_{i=1}^{p} \alpha_i y_{t-i} = \sum_{I=1}^{q} \beta_i u_{t-n_k-i} + \varepsilon$$

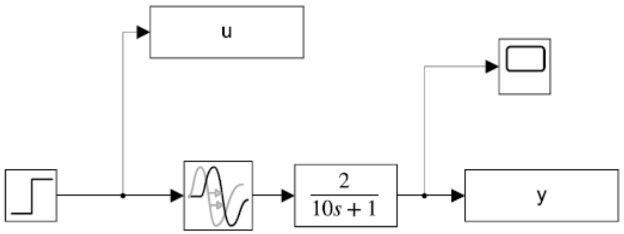
Example 3

u	y
0	0.0000
0	0.0000
0	0.0000
0	0.0000
0	0.0000
1	0.0000
1	0.0000
1	0.1903
1	0.3625
1	0.5184
1	0.6594
1	0.7869
1	0.9024

- Consider a transfer function given as follows:

$$G_p(s) = \frac{2\exp(-s)}{10s+1}$$

- Find an ARX model for the above system, with 1 unit step change in input and sampling time Ts = 1 unit



Convert into IDDATA format in Matlab
Syntax: dat = iddata(y,u,Ts)

```
>> datX = iddata(y,u,1)

>> sys=arx(datX,[3, 2, 1]);
```

Example 3 cont..

From Matlab:

A(z) = 1 - 0.9048 z^-1 - 6.647e-10 z^-2 - 8.099e-16 z^-3

B(z) = 9.789e-07 z^-1 + 0.1903 z^-2

Fit to estimation data: 100% (prediction focus)

FPE: 1.855e-31, MSE: 1.249e-31

ARMA

- Autoregressive-moving average (ARMA) model for "stationary" time series
- Combination of autoregression (AR) and moving average (MA)
- ARMA model can be used to understand and predict future values in time series
- ARMA model:

$$y(t) = c + \varepsilon_t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}$$

where α_i and β_i are the model parameters, p and q are the model orders, c is the constant and ε_t, ε_{t-i} are white noise errors.

- y at time t = constant + weighted sum of the last p values of y + weighted sum of the last q forecast errors

Nonseasonal ARIMA model

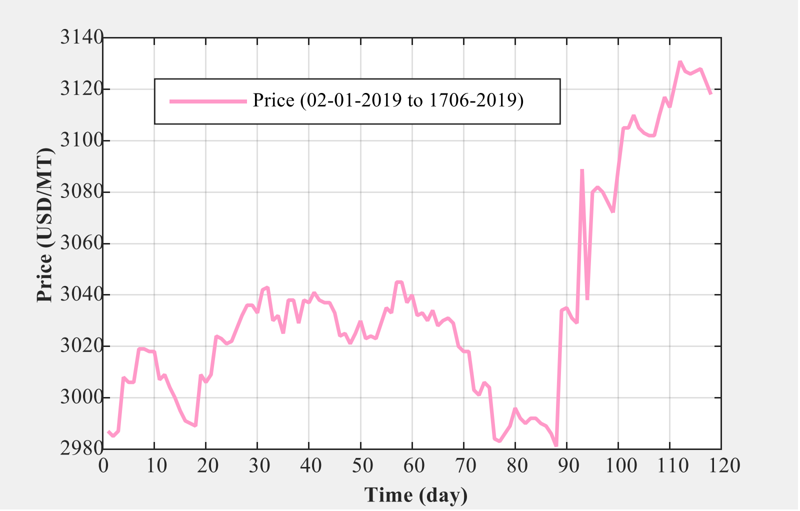
- Non-seasonal time series consists of a trend component and an irregular component.
- Decomposition of the time series into these components and estimation of the trend component and irregular component.
- ARIMA = autoregressive integrated moving average, consists of AR, I and MA where I means the integration.
- Matlab has a built-in ‘arima’ function to build an ARIMA model - the syntax:
- Mdl = arima(p,d,q)
  - p is the number of autoregressive terms,
  - d is the number of nonseasonal differences needed for stationarity, and
  - q is the number of lagged forecast errors in the prediction equation.

Matlab function - arima(p,d,q)

- Significance of d
  - If d = 0, then $\Delta Y_t = y_t$ where ΔY_t denotes the 0th difference of y
  - If d = 1, then $\Delta Y_t = y_t - y_{t-1}$
  - If d = 2, then $\Delta Y_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$
  - Note: d= 2 means the first-difference of the first-difference
- Some examples of typical model specifications:
  - ARIMA(0,1,0) = random walk model
  - ARIMA(2,0,0) = 2nd-order autoregressive model
  - ARIMA(0,1,1) = simple exponential smoothing model
  - ARIMA(1,1,2) = linear exponential smoothing with damped trend

Example 4 – Daily Prices of Black Pepper

- Black pepper prices from 02-01-2019 to 17-06-2019
- Use the following Matlab functions
  1. **arima**(p,d,q) => to build ARIMA model
  2. **estimate**(Mdl,X) => to estimate the ARMA model parameters
  3. **simulate**(EstMdl,t) => to simulate the ARMA model
  4. **plot**(tx,X,tx,y) = > to compare the data and model estimation
- Copy and paste the X data (black pepper price) on Matlab Command window



Sarawak Black Pepper Daily Price (USD/MT)

Example 4 – continue …

- There are 118 data points (over 118 days);
- Type on Matlab Command window:

>> X = [ ]; % copy and paste excel data into '[ ]'

>> tx = [1:1:118]';

- Build ARIMA model, e.g., try 2 specifications

>>Mdl1 = arima(1,0,3);

>>Mdl2 = arima(2,0,0);

- Estimate model parameters

>> EstMdl1 = estimate(Mdl1,X);

>>EstMdl2 = estimate(Mdl2,X);

- ARIMA 1 is given by

$$y_t = 229.38 + 0.925y_{t-1} + 0.03\varepsilon_{t-1} + 0.072\varepsilon_{t-2} + 0.546\varepsilon_{t-3}$$

- ARIMA 2 is given by

$$y_t = 24.13 + 0.635y_{t-1} + 0.357y_{t-2}$$

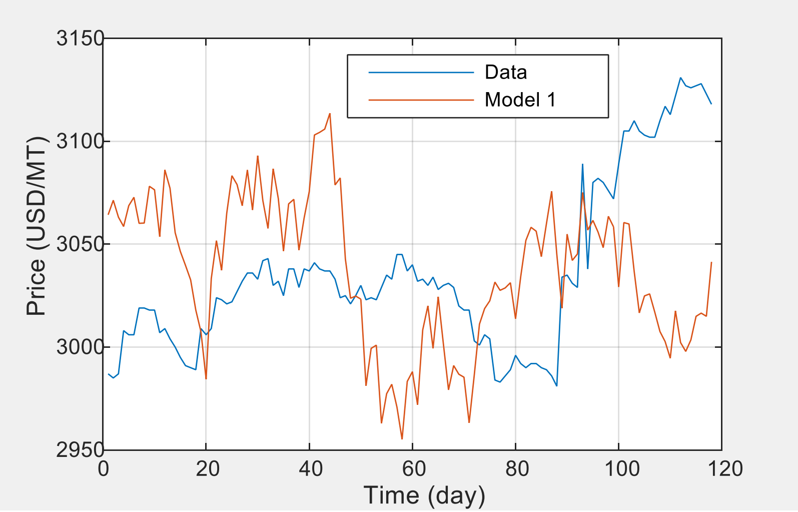
- Simulate the models:

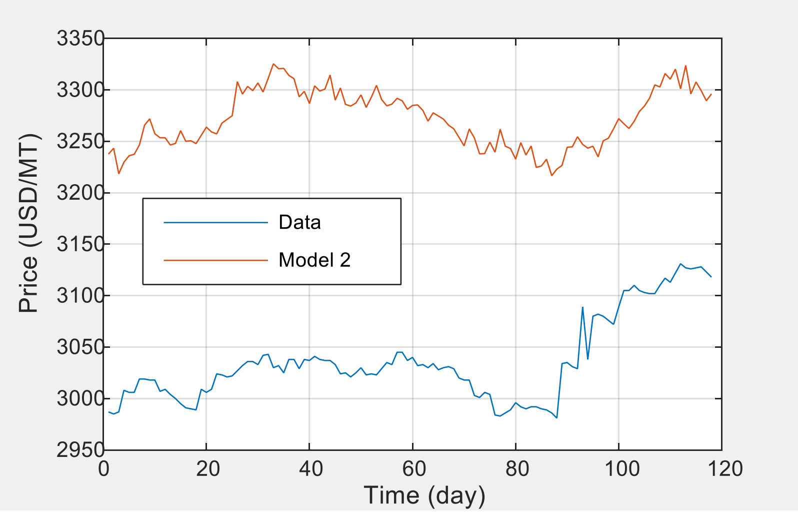
>> y1 = simulate(EstMdl1,n); % n = length(tx)

>> y2 = simulate(EstMdl2,n);

- Plot the data and model estimation

Example 4 – continue…



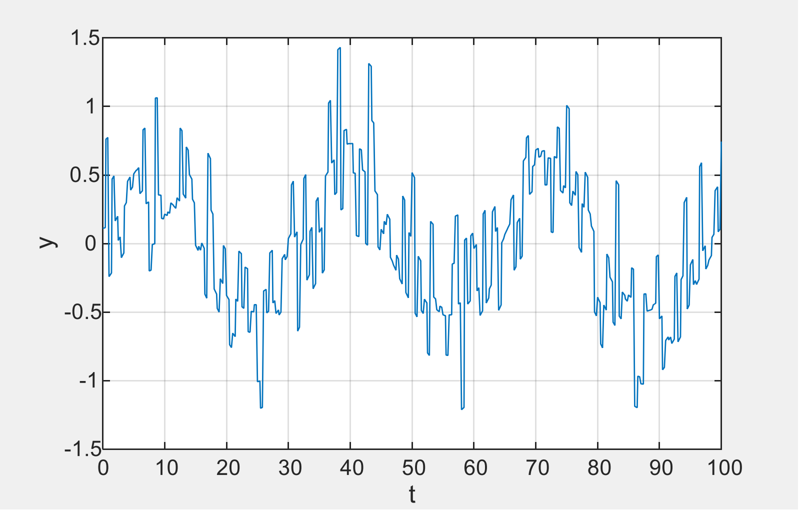


Result not accurate using both models. Why? The presence of significant “**unstationary**” behaviour.

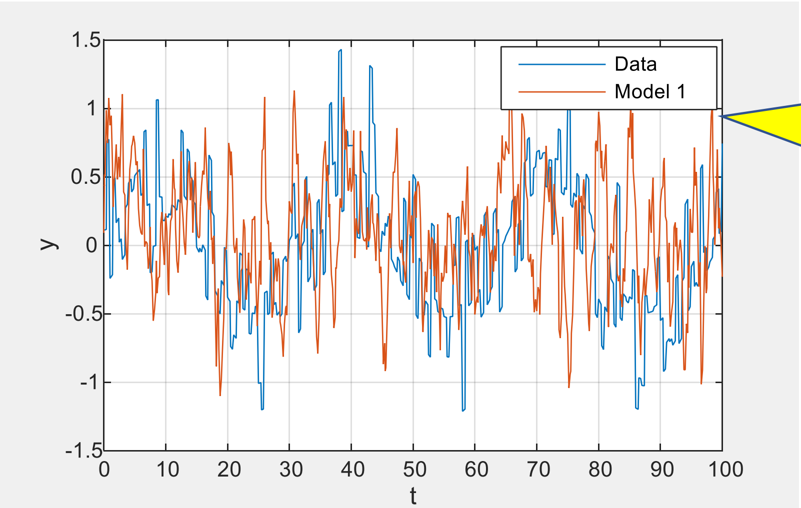
Example 5

- Data trend does not show significant drift.
- Try a few models
  i. Mdl1 = arima(2,0,0)
  ii. Mdl2 = arima(2,1,1);
  iii. Mdl3 = arima(2,0,2);

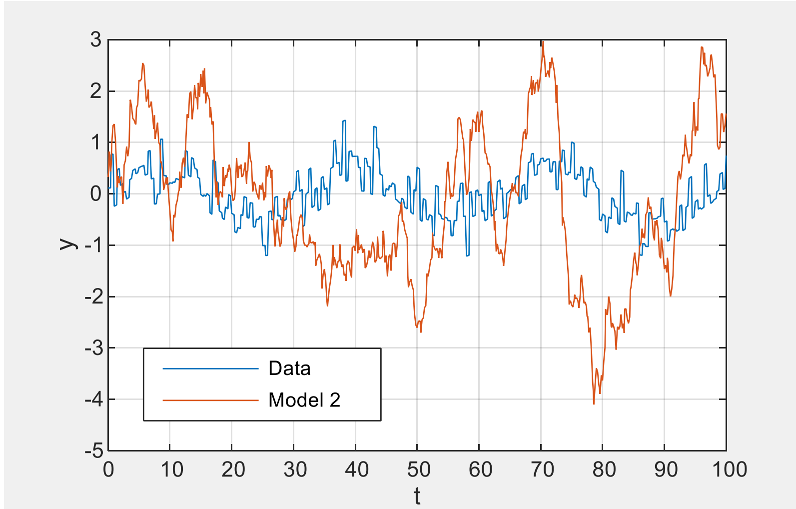
- ARIMA 1
$$y_t = 0.00118 + 0.925y_{t-1} - 0.0769y_{t-2}$$
- ARIMA 2
$$y_t = 0.0016 - 0.5798y_{t-1} + 0.1542y_{t-2} + 0.6022\varepsilon_{t-1}$$
- ARIMA 3
$$y_t = -0.5799y_{t-1} + 0.1542y_{t-2} - 0.3978\varepsilon_{t-1} - 0.6022\varepsilon_{t-2}$$

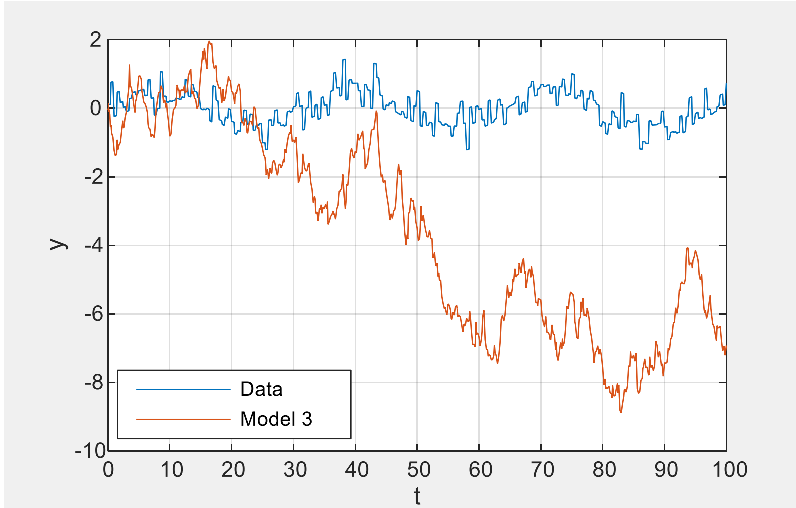


Example 5 – continue …



This is a more suitable model for the data.
The other 2 models are less suitable





Summary

- Time series data analysis is common in process industry and in many other fields
- 3 common models are AR, ARX and ARMA (or ARIMA in Matlab)
- Selection of a suitable model structure requires some knowledge about the data characteristics, e.g., stationary or non-stationary, random or deterministic
- ARX can be used to represent a transfer function model