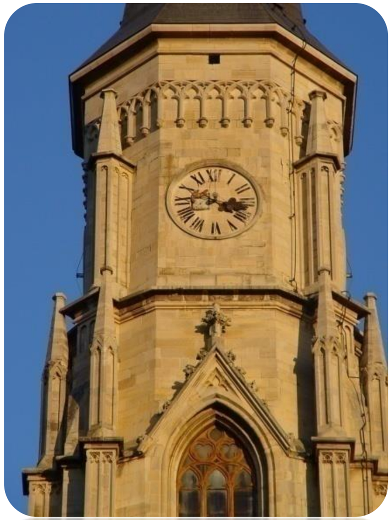




Tehnici de prelucrare a rezultatelor cercetărilor experimentale

S.l. dr. ing. Ciprian LAPUSAN

SUPORT CURS

Content

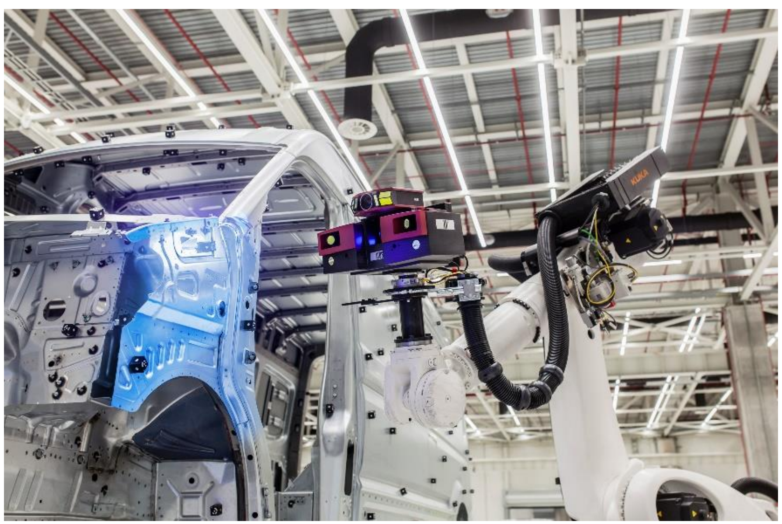
❑ Data Acquisition Systems

- Data Acquisition Overview
- Transducers
- Signals
- Signal Conditioning
- Data Acquisition Device
- Dedicated Software
- Examples

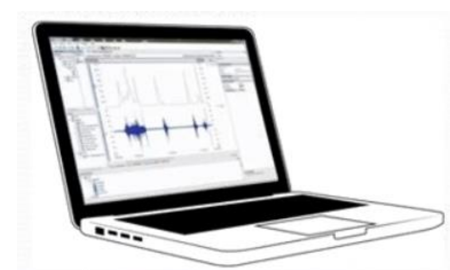
Data Acquisition Overview

What is data Acquisition (DAQ)?

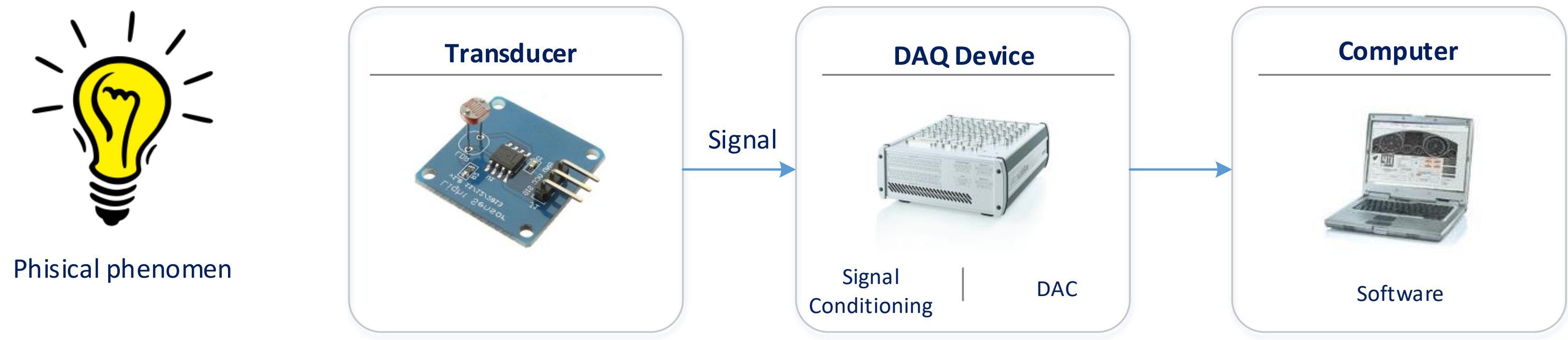
- **Data acquisition** (DAQ) is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.



Sursa imagine:https://zrobotyzowany.pl/

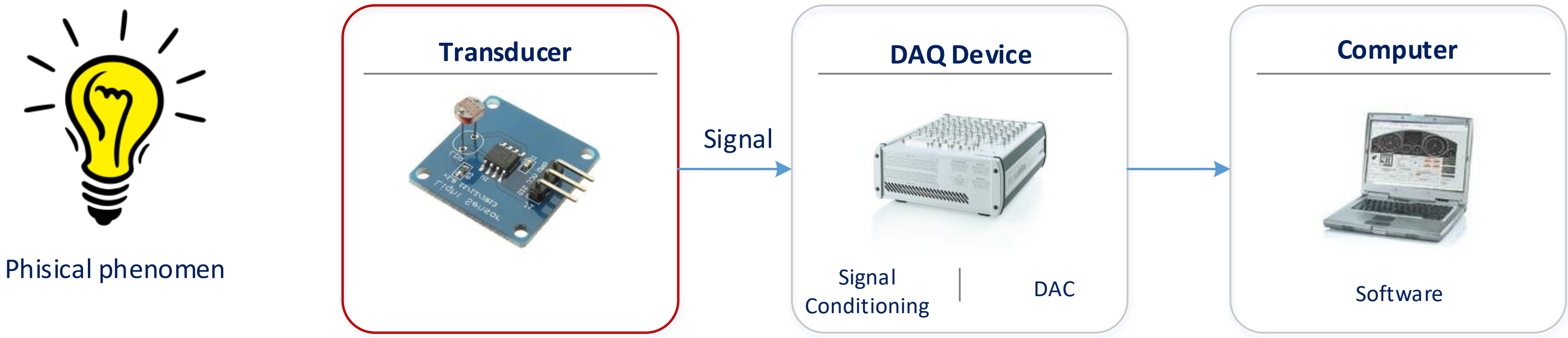


Data Acquisition Overview



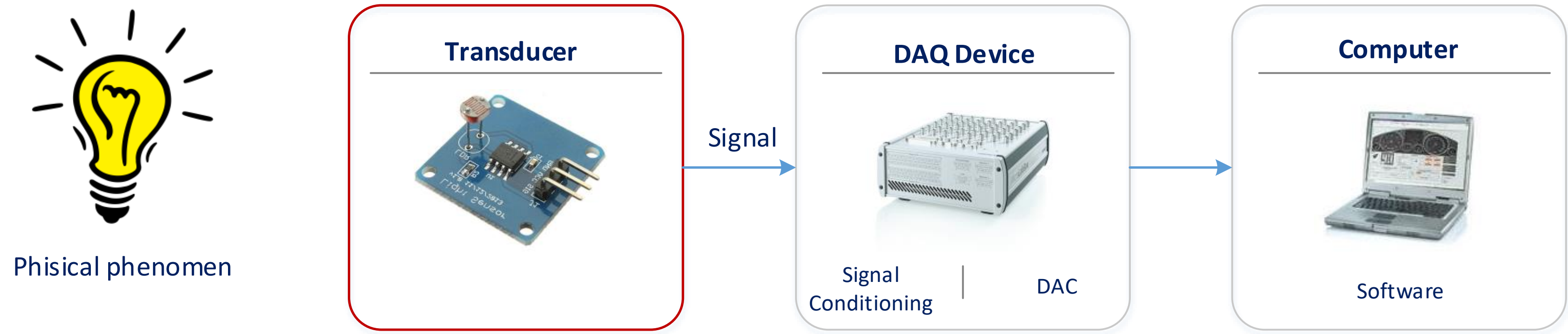
- Main elements are:
  - Transducer
  - Signal
  - Signal Conditioning
  - Data Acquisition (DAQ) device
  - Application level software

Transducers Overview



- What is a Transducer?

Transducers Overview



- What is a Transducer?

- A device that converts a signal from one physical form to a corresponding signal having a different physical form (Physical form: mechanical, thermal, magnetic, electric, optical, chemical etc.)

Transducers Overview

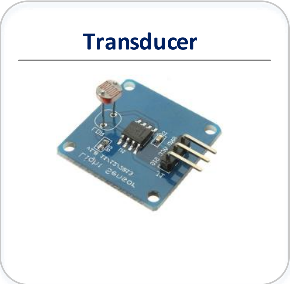
- Types of Transducers?

- Transducer can be further divided into Sensors, which monitors a system and Actuators, which impose an action on the system

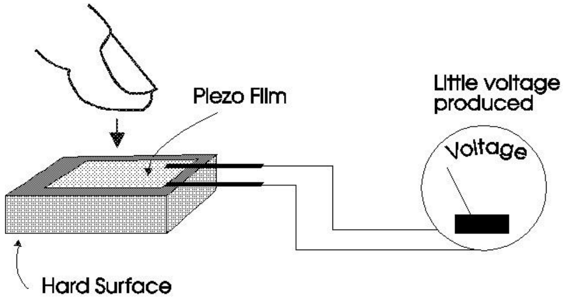
Transducer

- **Sensors**
  - a sensor is a device that measure a physical quantity and converts it into a signal which can be read by an observer or by an instrument

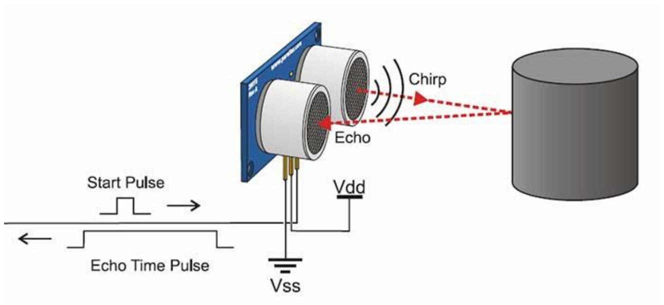
Transducers Overview



- **Sensors**
  - *Passive*
    - A passive sensor does not need any additional energy source and directly generates an electric signal in response to an external stimulus
    - Example: thermocouple, a photodiode, piezoelectric sensor etc.
  - *Active*
    - The active sensors require external power for their operation, which is called an *excitation signal;* that signal is modified by the sensor to produce the output signal



Piezoelectric Sensor



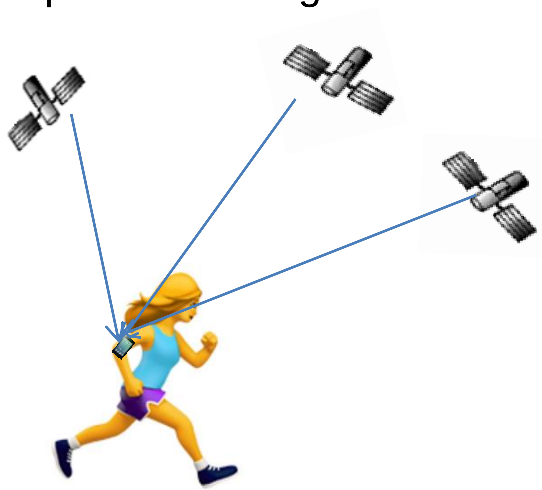
Ultrasonic Sensor

Transducers Overview

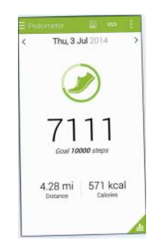
Transducer

▪ Sensors

- *Absolute*
  - An *absolute* sensor detects a stimulus in reference to an absolute physical scale that is independent on the measurement conditions
- *Relative*
  - a *relative* sensor produces a signal that relates to some special case



- GPS location



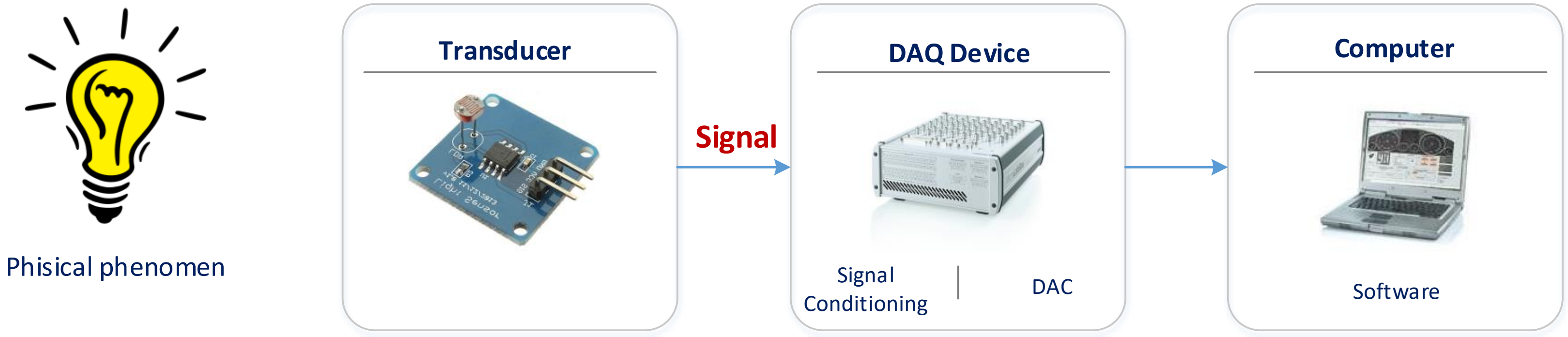
- Distance based on pedometer

Transducers Overview

Transducer

- **Sensors**
  - Based on physical phenomena
    - Mechanical: strain gage, displacement (LVDT), velocity (laser vibrometer), accelerometer, tilt meter, viscometer, pressure, etc.
    - Thermal: thermal couple
    - Optical: camera, infrared sensor
    - Chemical
    - Optical and radiation sensor
    - Electromagnetic
    - Biological

Signal Overview



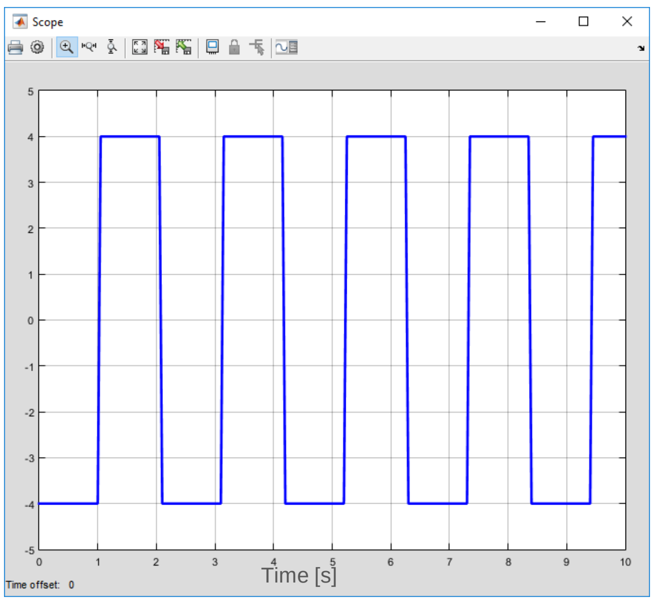
- Type of signals
- Information in a Signal
  - State, Rate, Level, Shape and Frequency

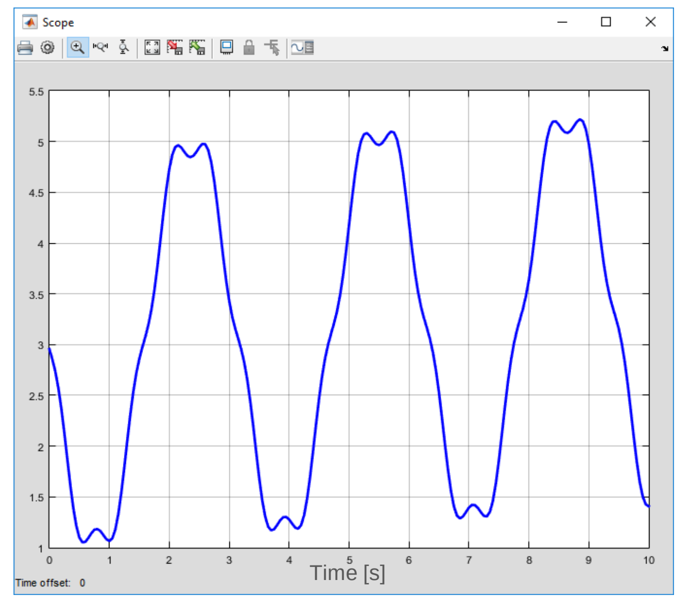
Signal Overview

Your Signal

Digital

Analog

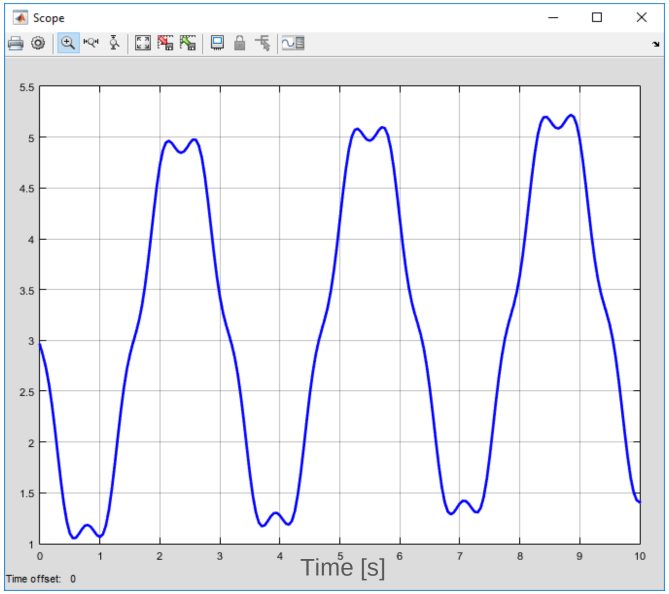




Analog Signal

Def - Continuously varying representation of a condition, physical phenomenon, or quantity such as flow, pressure, or temperature, transmitted as electrical, mechanical, or pneumatic energy

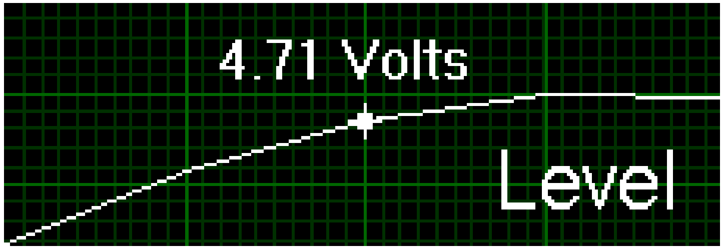
- ❑ Continuous signal
  - Can be at any value with respect to time
- ❑ Three types of information:
  - Level
  - Shape
  - Frequency (Analysis required)



Analog Signal

- ❑ Characteristics

- **Level**
  - gives the corresponding voltage for the measured physical phenomena at a moment in time

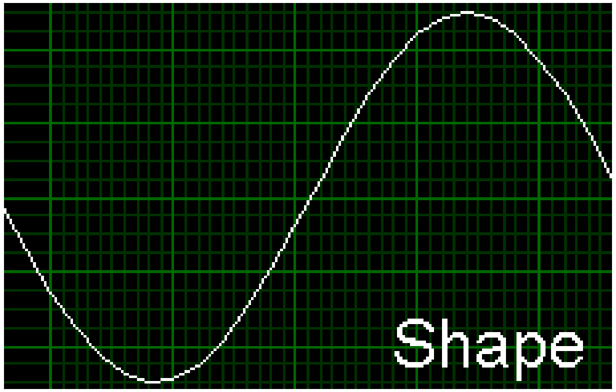


  - When measuring a signal level the accuracy of measurement is very important, a data acquisition system that yields maximum accuracy should be chosen

Analog Signal

❑ Characteristics

- **Shape**
  - Because analog signals can be at any state with respect to time, the shape of the signal is often important.

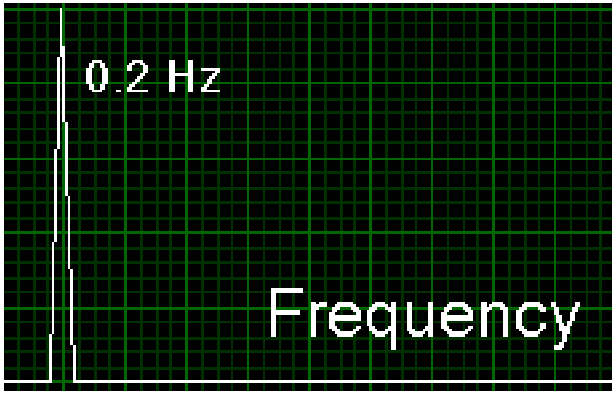


  - Measuring the shape allows further analysis of the signal including peak value, dc values and slope
  - If signals shape is of interest then faster sampling of the analog signal is required

Analog Signal

- ❑ Characteristics

- **Frequency**
  - frequency (f) – number of periods in one second – unit: hertz [Hz] = [1/s]

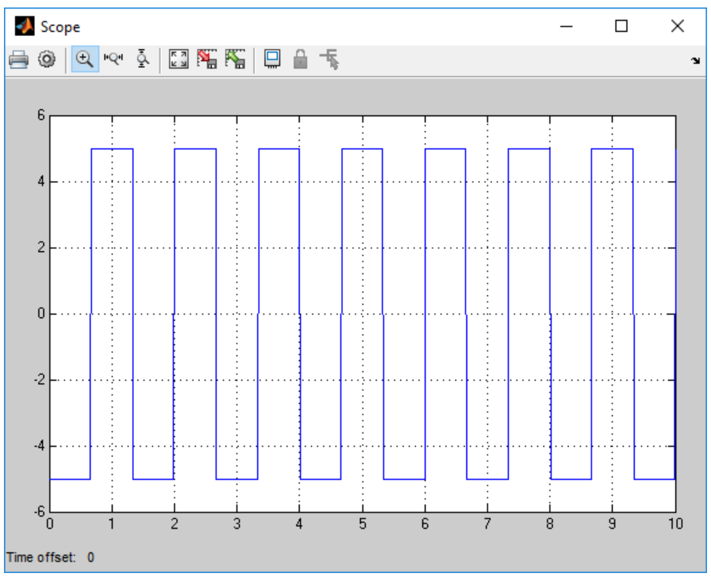


  - Frequency can not be directly measured, the signal is analyzed using specialized software
  - When frequency is of interest both accuracy and acquisition speed is important

Digital Signal

Your Signal

Digital



- Two possible levels:
  - High/On (2 - 5 Volts)
  - Low/Off (0 - 0.8 Volts)
- Two types of information:
  - State
  - Rate

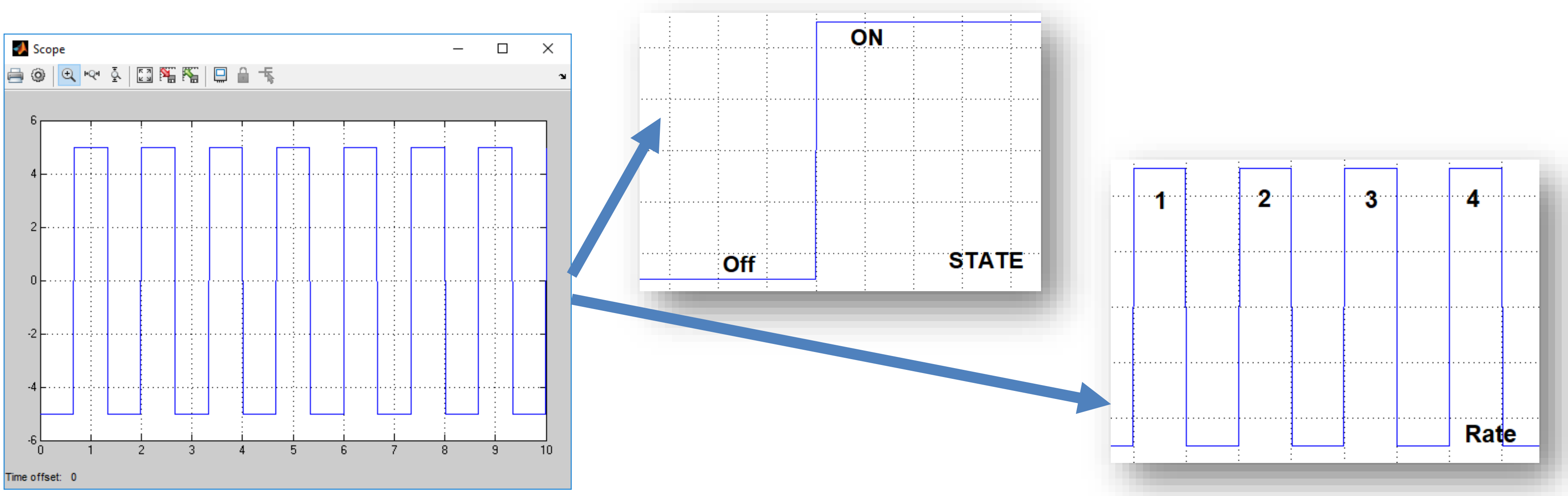
Digital Signal

State

A digital signal only has two possible states: ON or OFF. Thus one of the quantities of a digital signal we can measure is whether the state is ON or OFF.

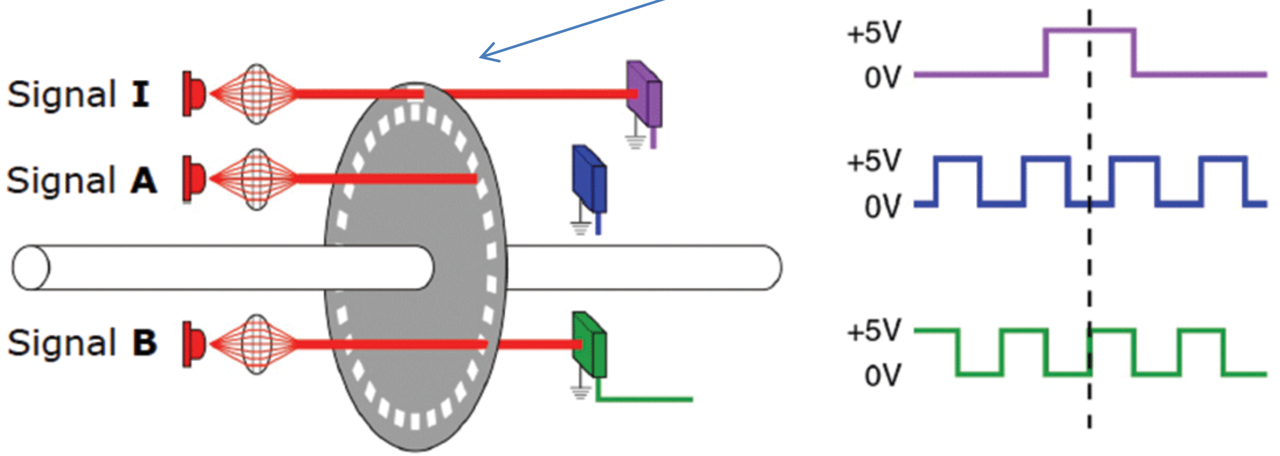
Rate

A digital signal also changes state with respect to time. Therefore, the other quantity of a digital signal we can measure is the rate, or in other words how the digital signal changes states with respect to time.



Digital Signal

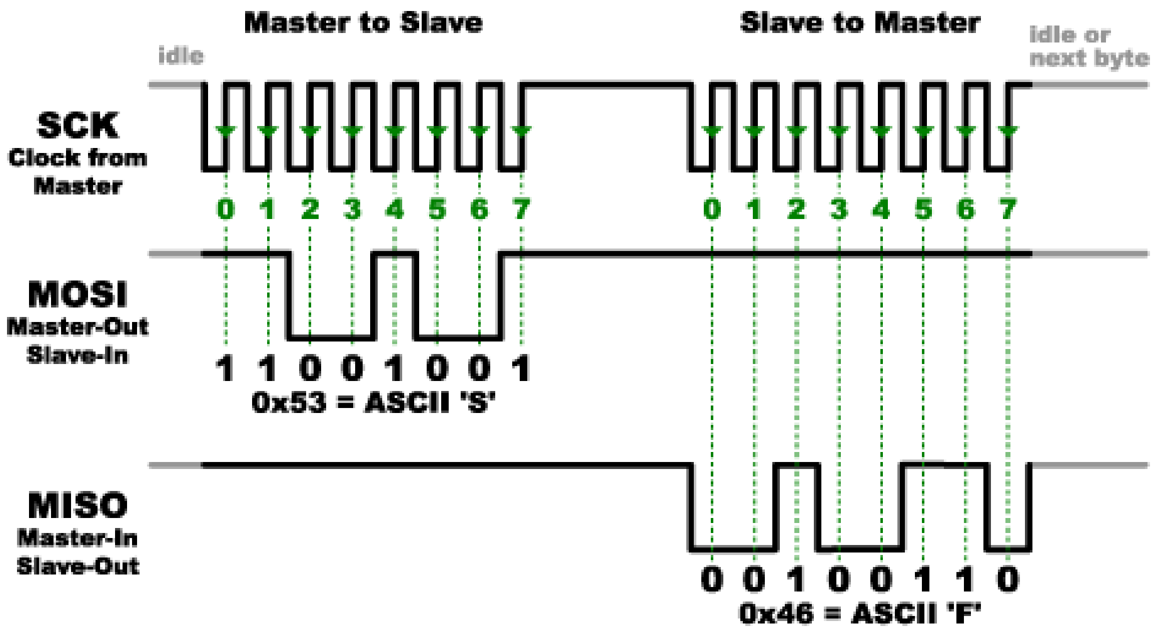
- optical encoder



Standard A and B quadrature signals plus an index signal

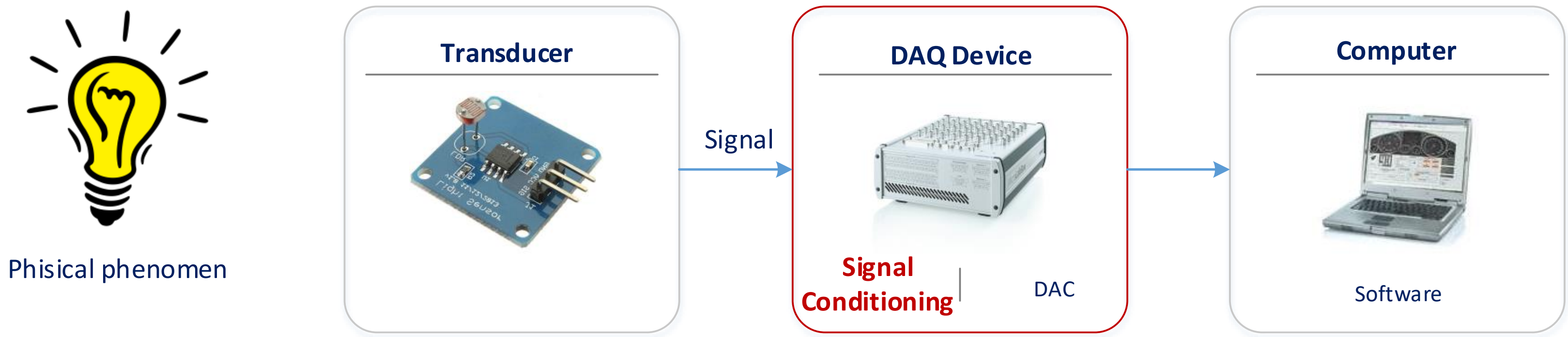
Digital Signal

- Interfaces like serial, I^2C, and SPI transmit data via a coded sequence of square waves.



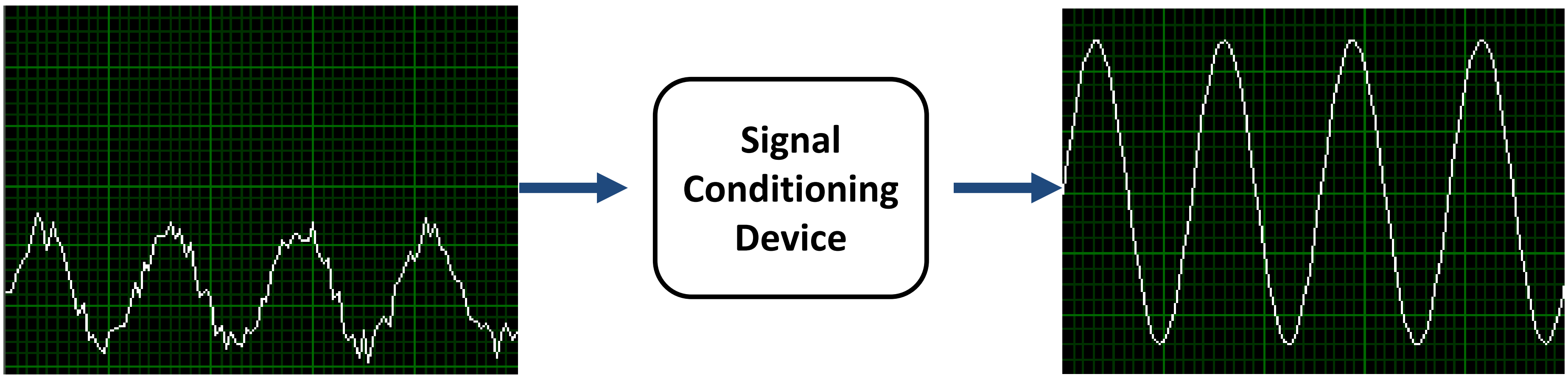
Serial peripheral interface (SPI) uses many digital signals to transmit data between devices

Signal Conditioning



- Purpose of Signal Conditioning
- Types of Signal Conditioning

Signal Conditioning



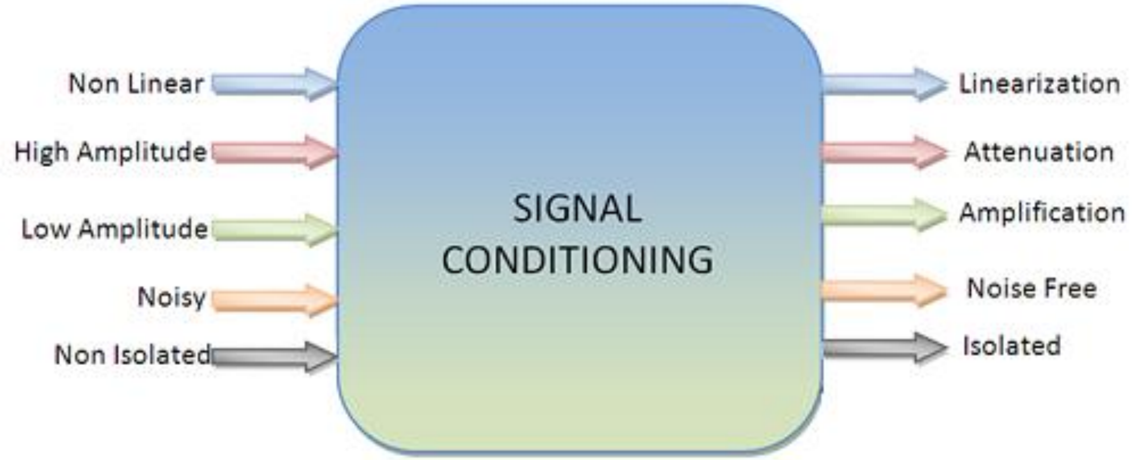
Noisy, Low-Level Signal

Filtered, Amplified Signal

- Signal Conditioning takes a signal that is difficult for your DAQ device to measure and makes it easier to measure
- Signal Conditioning is not always required
    - Depends on the signal being measured

Signal Conditioning

- The following features may be available:
  - Amplification
  - Attenuation
  - Filtering
  - Isolation
  - Linearization



Signal Conditioning

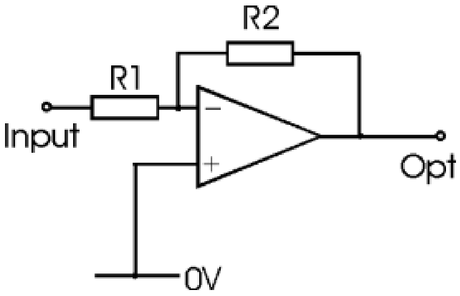
- Amplification
  - Used on low-level signals (i.e. thermocouples)
  - Maximizes use of Analog-to-Digital Converter (ADC) range and increases accuracy
  - Increases Signal to Noise Ratio (SNR)

Signal Conditioning

❑ Amplification

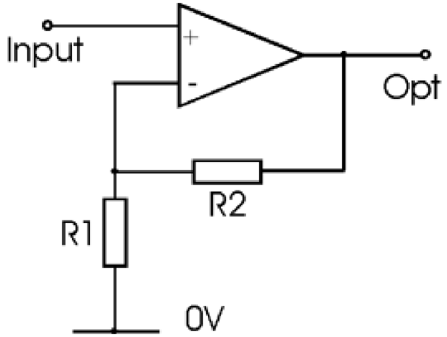
- Inverting op-amp amplifier

$$Vo = -R2/R1 * Vi \qquad (1)$$



- Non-inverting op-amp amplifier

$$Vo = (1+R2/R1) * Vi \qquad (2)$$



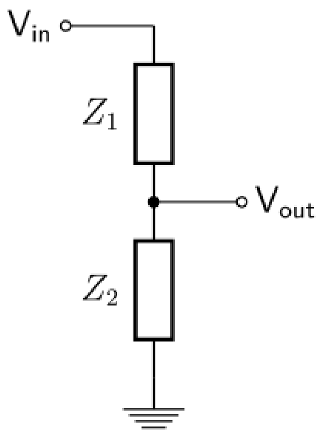
- Non-inverting op-amp amplifier useful when a **high impedance** input is needed

- Inverting op-amp amplifier useful when a **low impedance** input is needed

Signal Conditioning

- **Attenuation**
  - A circuit that produces an output voltage (V_{out}) that is a fraction of its input voltage (V_{in})
  - Can be needed to get a high-level signal down to the acceptable DAQ-card range
  - Voltage divider

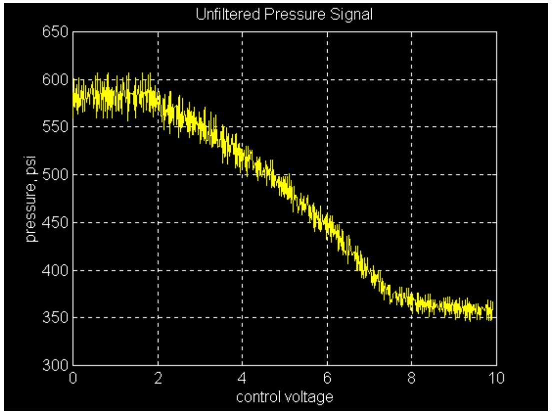
$$V_{out} = \frac{Z_2}{Z_1 + Z_2} \cdot V_{in}$$

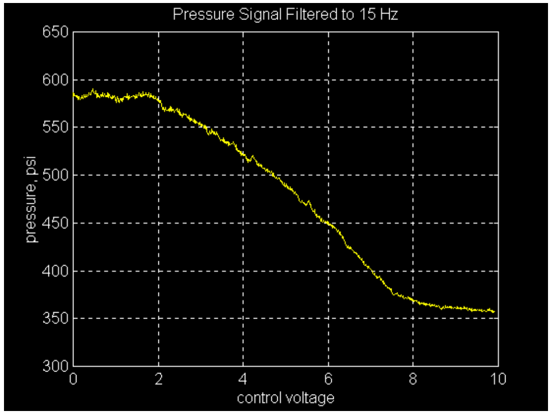


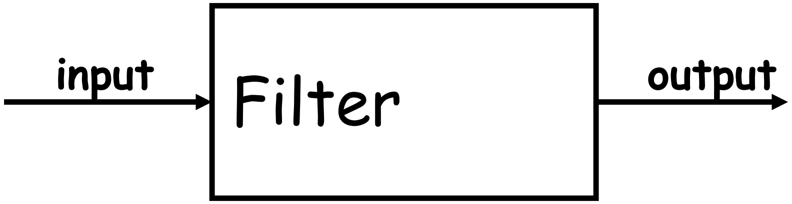
Signal Conditioning

❑ Filtering

- Certain desirable features are **retained**
- Other undesirable features are **suppressed**

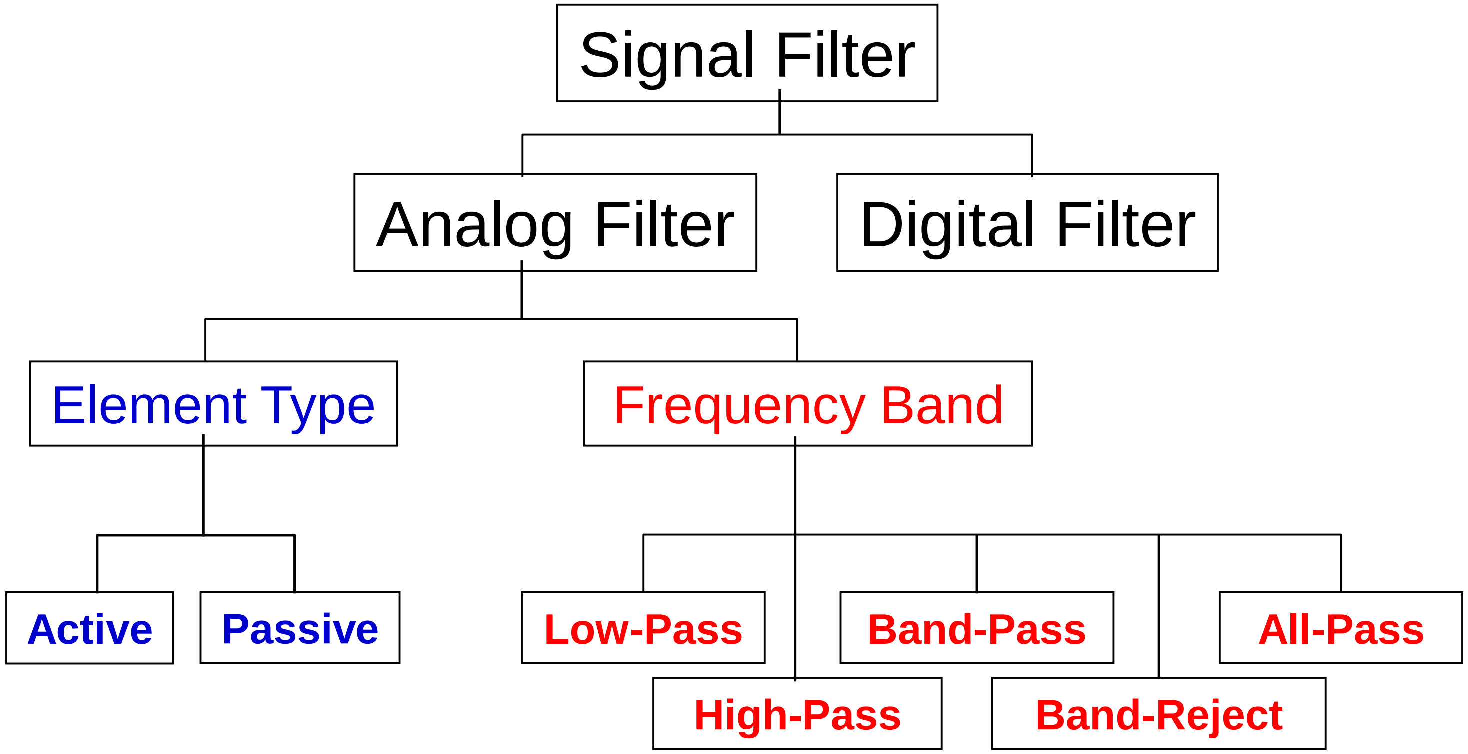






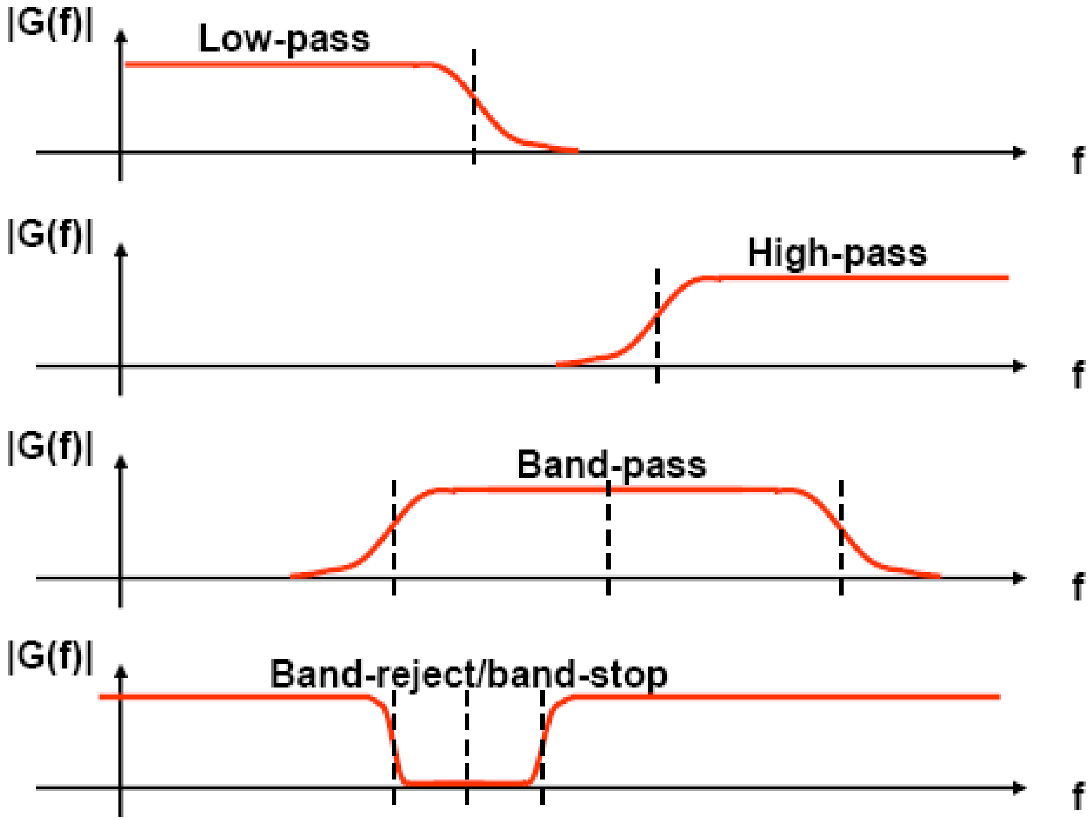
Signal Conditioning

- Filtering
  - Classification of Filters



Signal Conditioning

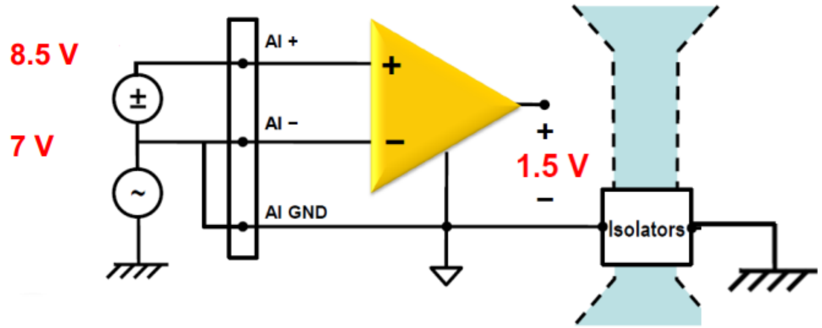
- Filtering



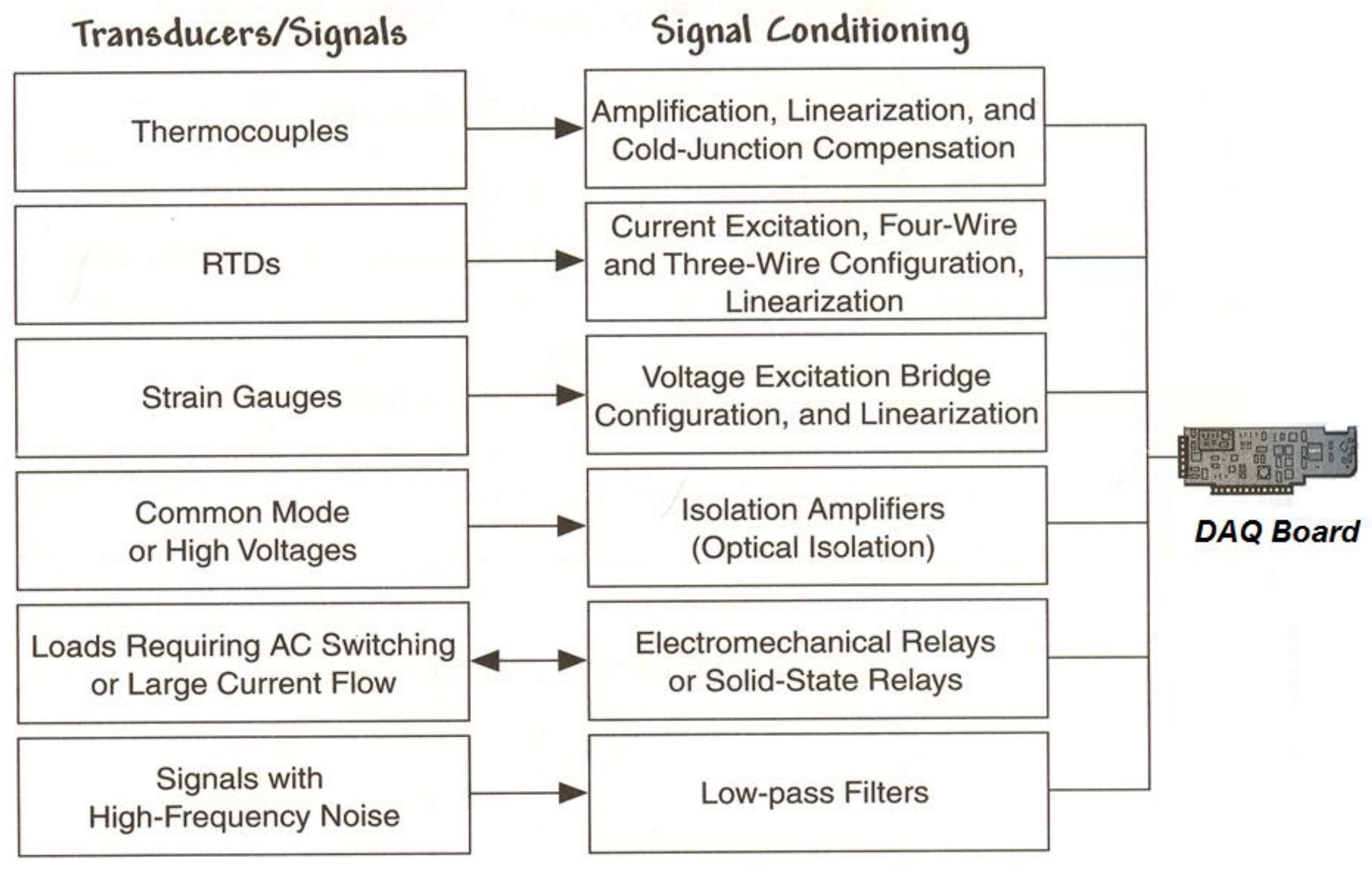
Signal Conditioning

❑ Isolation

- Isolation - electrically separates two parts of a measurement device
- Protects from high voltages
- Prevents ground loops
- Separate ground planes of data acquisition device and sensor
- Isolation techniques: Optical, Capacitive, Inductive Coupling

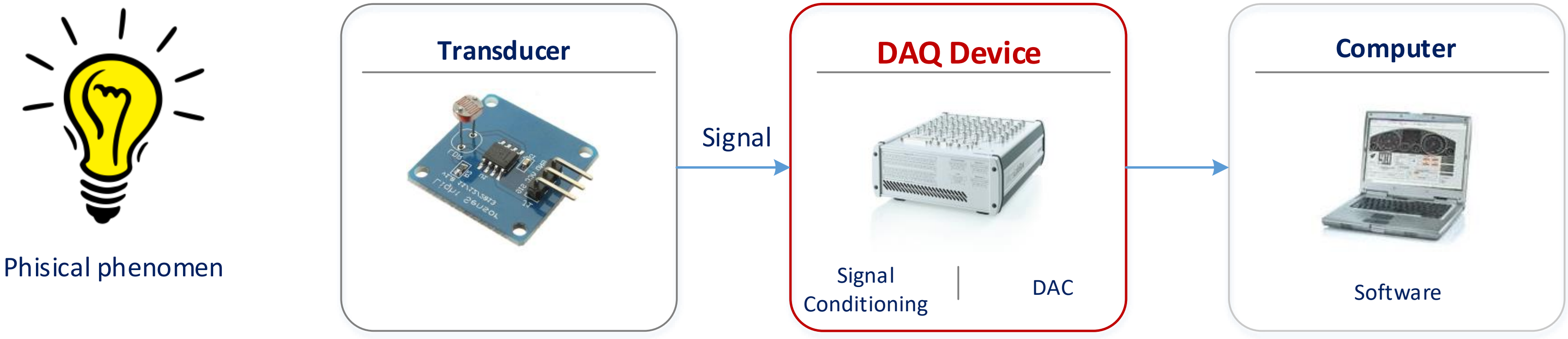


Signal Conditioning



Common types of transducers/signals and the required signal conditioning

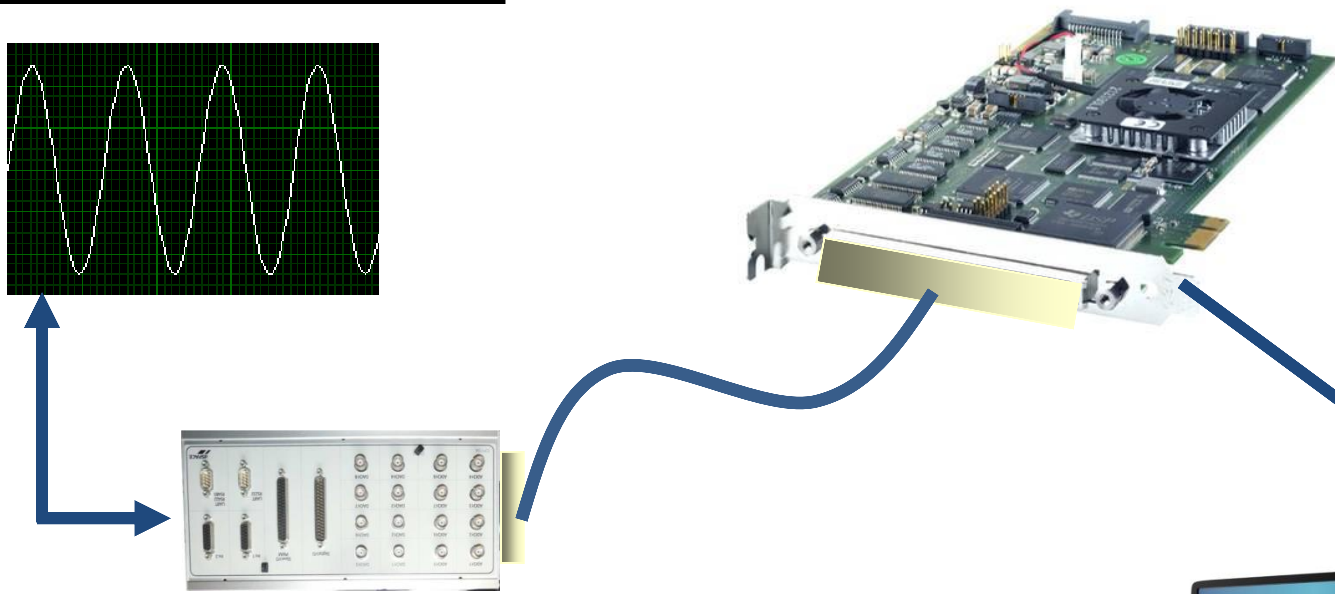
DAQ Hardware Overview



Topics

- Types of DAQ Hardware
- Components of a DAQ device
- Configuration Considerations

Types of DAQ Hardware



- DAQ Hardware turns your PC into a measurement and automation system

DAQ Device

DAQ Device

- Most DAQ devices have:
  - Analog Input
  - Analog Output
  - Digital I/O
  - Counters
- Specialty devices exist for specific applications
  - High speed digital I/O
  - High speed waveform generation
- Connect to the bus of the computer
- Compatible with a variety of bus protocols
  - PCI, PXI/CompactPCI, ISA/AT, PCMCIA, USB, 1394/Firewire

Computer

Configuration Considerations

- Analog Input
  - Resolution
  - Range
  - Gain
  - Input Mode (RSE or NRSE)

Resolution

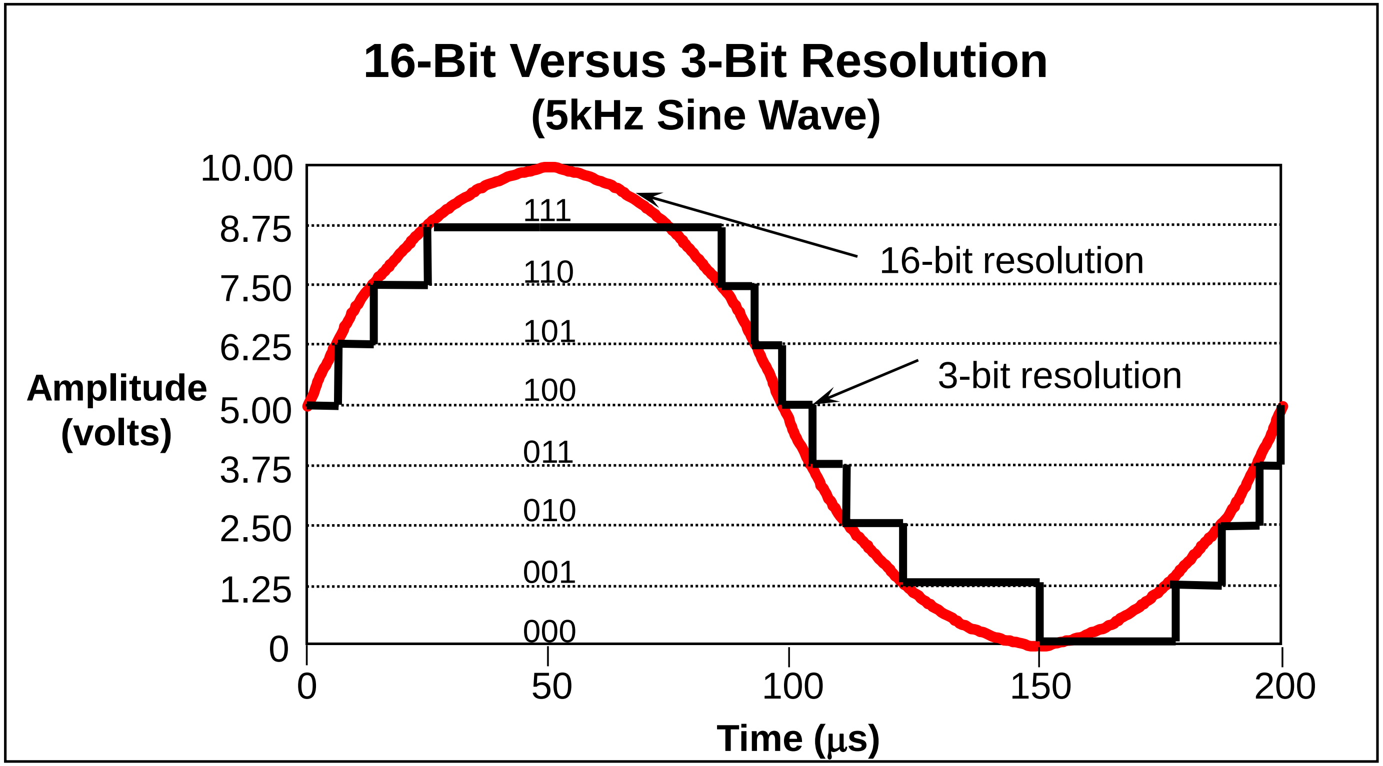
- Number of bits the ADC uses to represent a signal
- Resolution determines how many different voltage changes can be measured
  - Example: 12-bit resolution

$$\text{\# of levels} = 2^{\text{resolution}} = 2^{12} = 4{,}096 \text{ levels}$$

- Larger resolution = more precise representation of your signal

Resolution Example

- 3-bit resolution can represent 8 voltage levels
- 16-bit resolution can represent 65,536 voltage levels

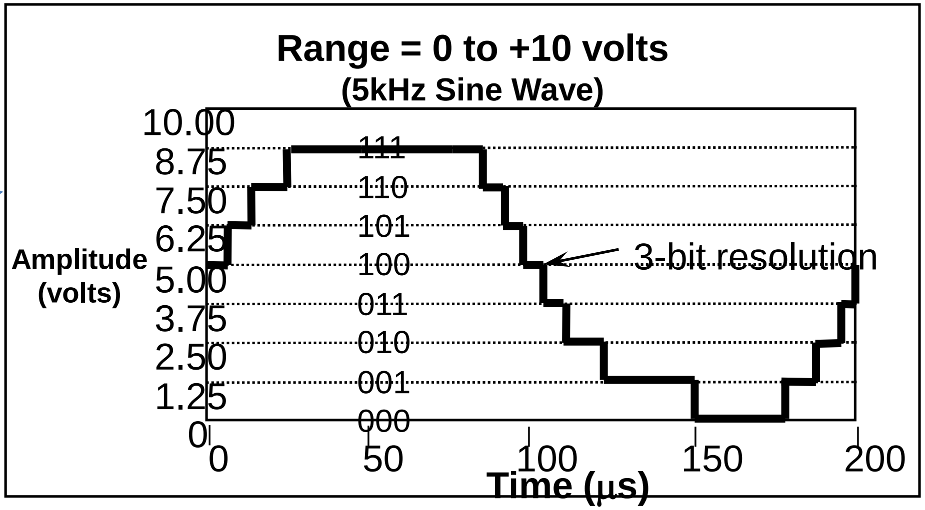


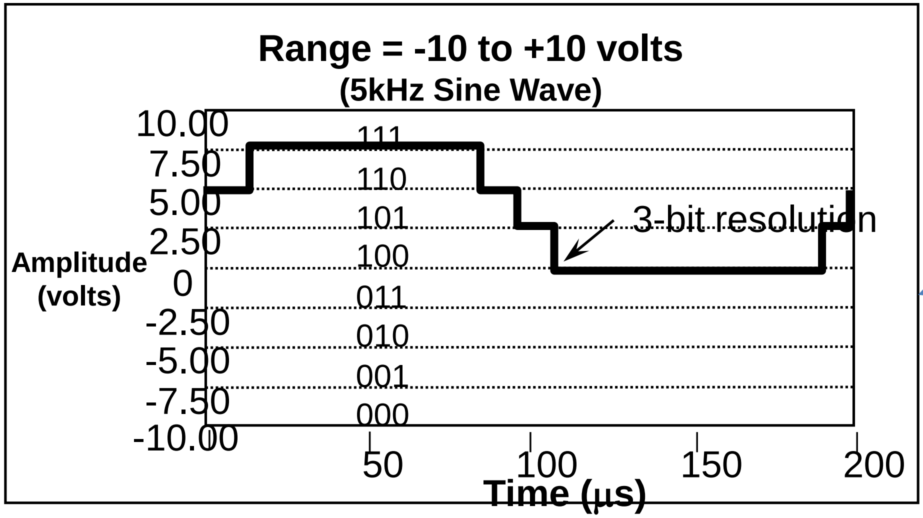
Range

- Minimum and maximum voltages the ADC can digitize
- DAQ devices often have different available ranges
  - 0 to +10 volts
  - -10 to +10 volts
- Pick a range that your signal fits in
- Smaller range = more precise representation of your signal
  - Allows you to use all of your available resolution

Range

- Proper Range
  - Using all 8 levels to represent your signal

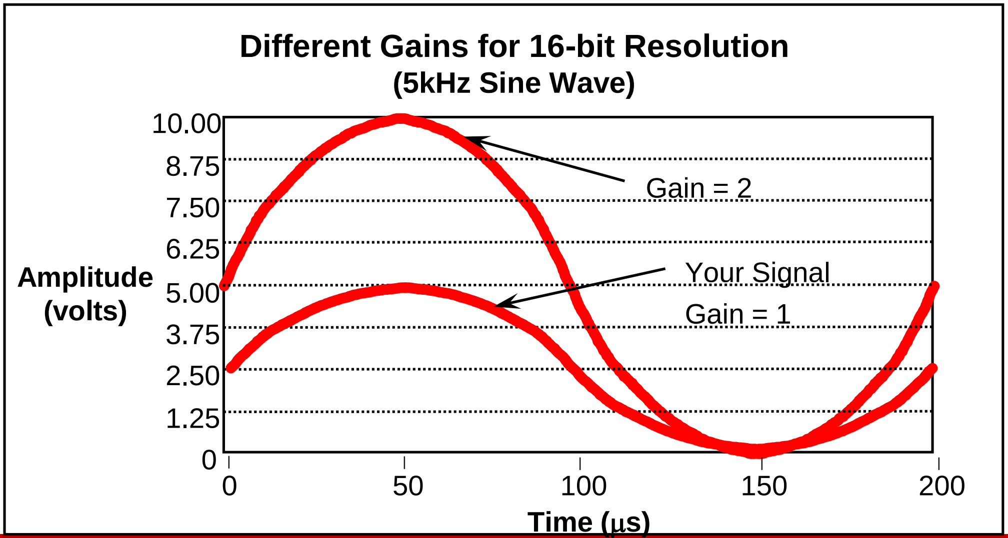




- Improper Range
  - Only using 4 levels to represent your signal

Gain

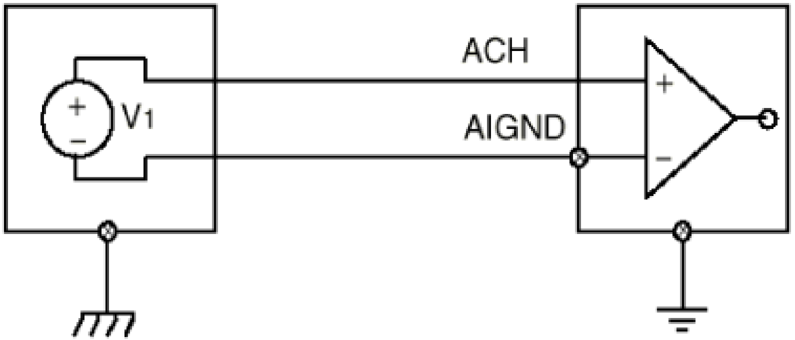
- Gain setting amplifies the signal for best fit in ADC range
- Proper gain = more precise representation of your signal
  - Allows you to use all of your available resolution
- Example
  - Input limits of the signal = 0 to 5 Volts
  - Range Setting for the ADC = 0 to 10 Volts
  - Gain Setting applied by Instrumentation Amplifier = 2



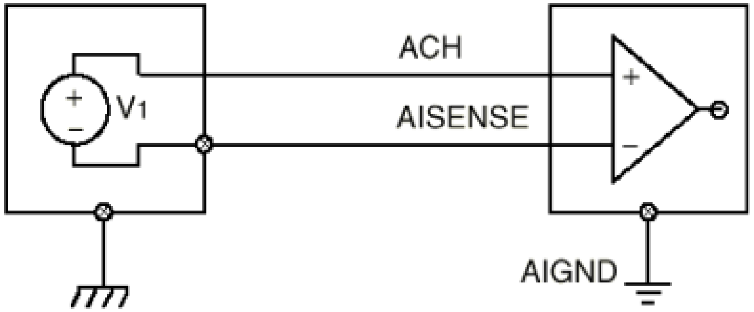
Input Mode

- The **RSE** (Referenced Single-Ended) configuration is used for floating signal sources. In this case, the DAQ hardware device itself provides the reference ground for the input signal.
- The **NRSE** (Non-Referenced Single-Ended) input configuration is used for grounded signal sources. In this case, the input signal provides its own reference ground and the hardware device should not supply one.

RSE

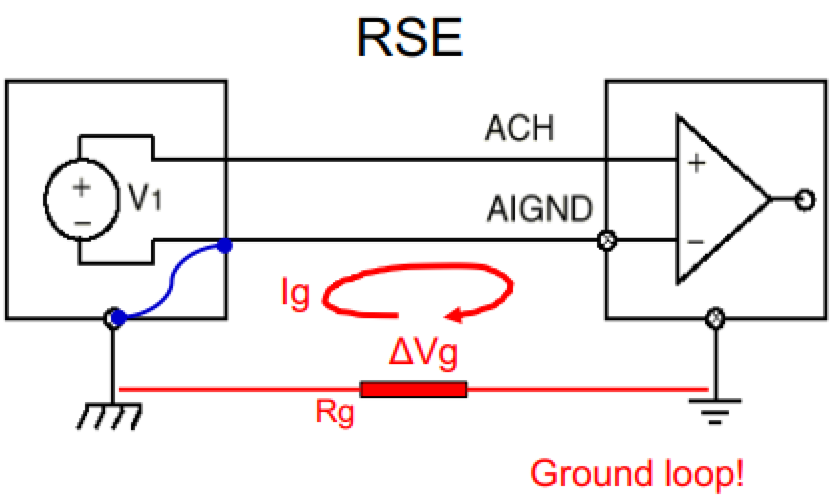


NRSE



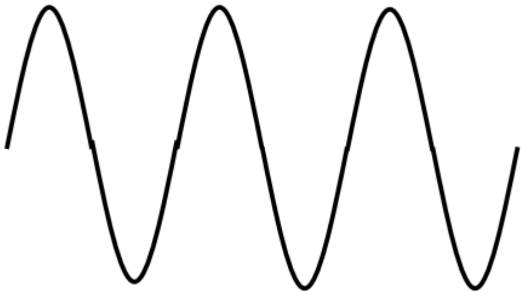
Grounding Issues

- To get correct measurements you must properly ground your system
- How the signal is grounded will affect how we ground the instrumentation amplifier on the DAQ device

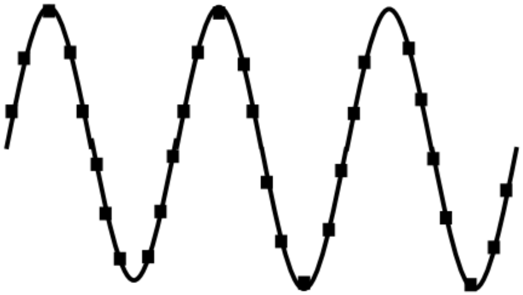


Sampling Considerations

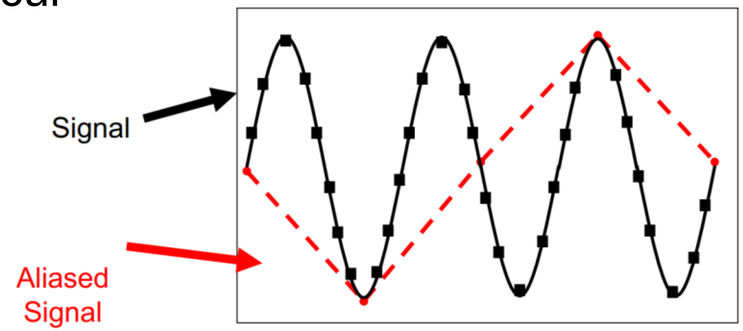
- An analog signal is continuous
- A sampled signal is a series of discrete samples acquired at a specified sampling rate
- The faster we sample the more our sampled signal will look like our actual signal
- If not sampled fast enough a problem known as **aliasing** will occur



Actual Signal



Sampled Signal



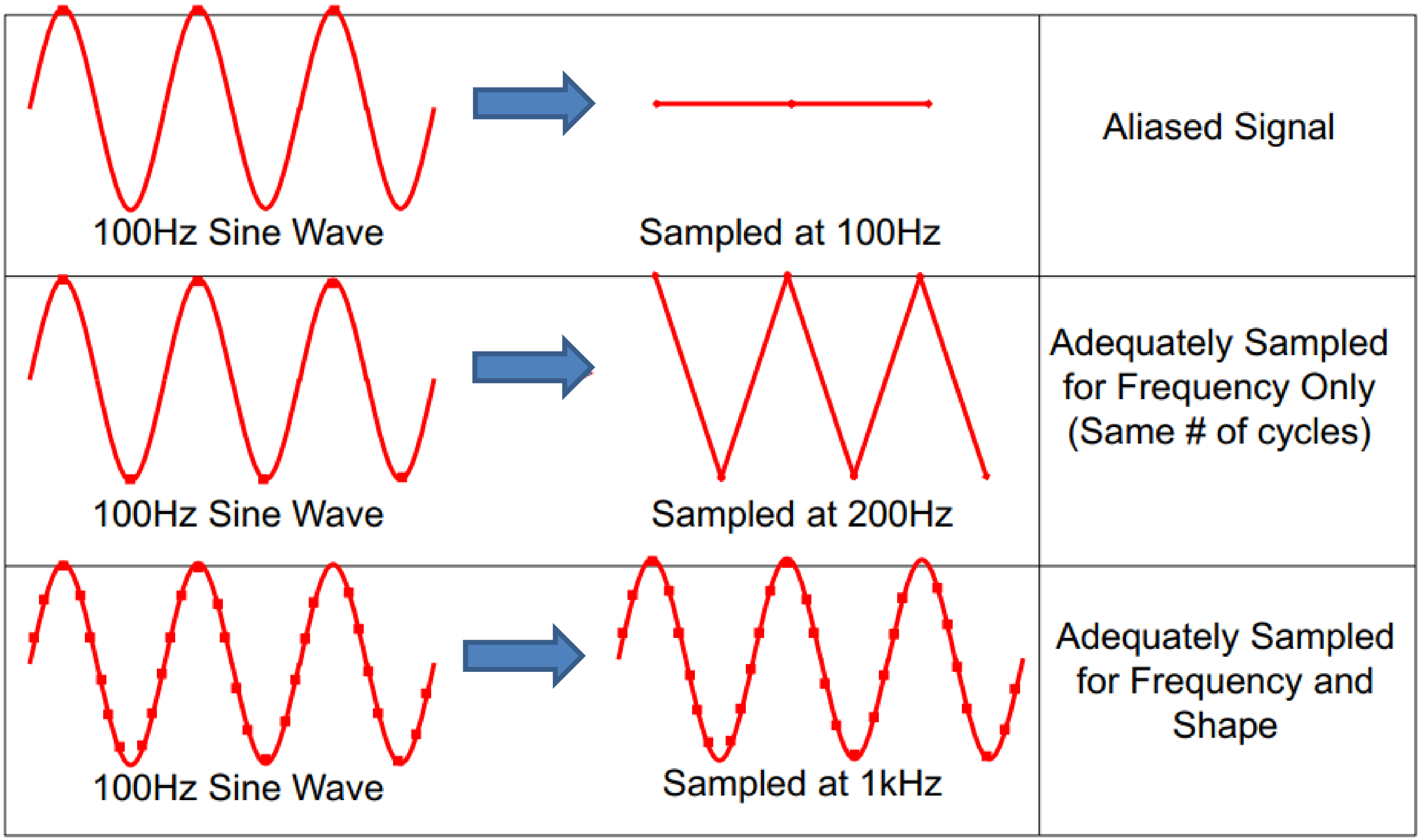
Sampling Considerations

Nyquist's Theorem

– You must sample at greater than 2 times the maximum frequency component of your signal to accurately represent the frequency of your signal

Obs: You must sample between 5 - 10 times greater than the maximum frequency component of your signal to accurately represent the shape of your signal

Sampling Considerations



Data acquisition platforms - hardware and software

dSpace Platform

- The master PPC on the DS1104 controls an ADC unit featuring two different types of A/D converters:

- One A/D converter (ADC1) multiplexed to four channels (signals ADCH1 … ADCH4).The input signals of the converter are selected by a 4:1 input multiplexer. The A/D converters have the following characteristics:
  - 16-bit resolution
  - ±10 V input voltage range
  - ± 5 mV offset error
  - ± 0.25% gain error
  - >80 dB (at 10 kHz) signal-to-noise ratio (SNR)

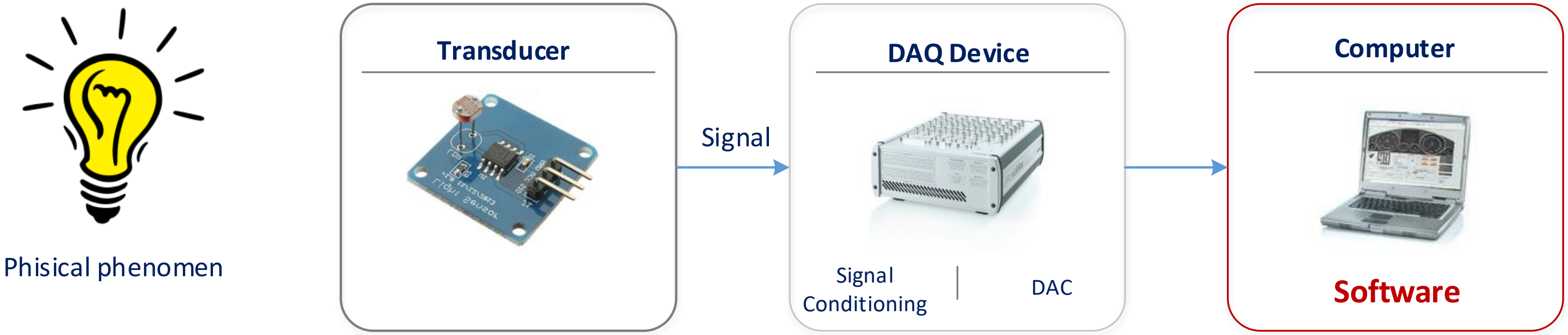
dSpace Platform

- Four parallel A/D converters with one channel each (signals ADCH5 … ADCH8). The A/D converters have the following characteristics:
  - 12-bit resolution
  - ±10 V input voltage range
  - ± 5 mV offset error
  - ± 0.5% gain error
  - > 70 dB signal-to-noise ratio (SNR)

dSpace Platform

- The master PPC on the DS1104 controls a D/A converter. It has the following characteristics:
  - 8 parallel DAC channels (signals DACH1 … DACH8)
  - 16-bit resolution
  - ±10 V output voltage range
  - ± 1 mV offset error, 10 V/K offset drift
  - ± 0.1% gain error, 25 ppm/K gain drift
  - >80 dB (at 10 kHz) signal-to-noise ratio (SNR)
  - Transparent and latched mode

Software



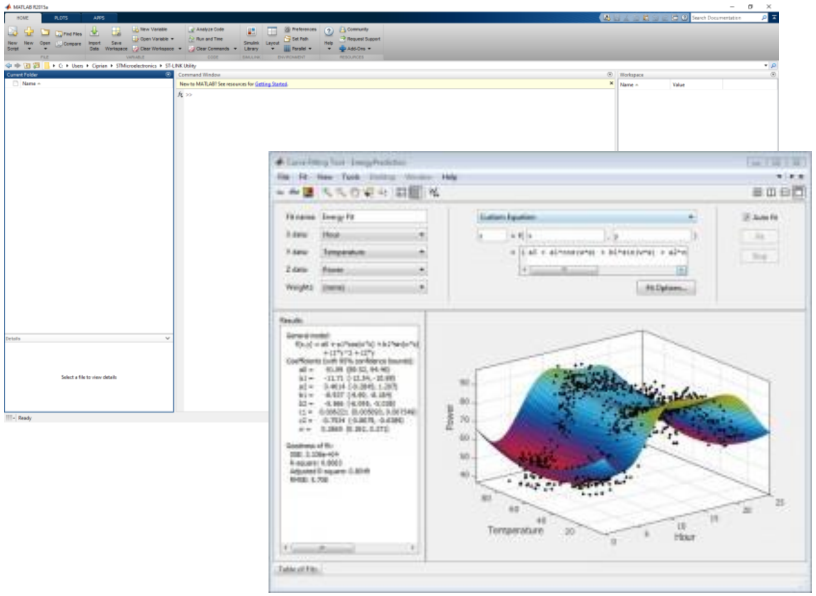
Topics:

- Matlab
- ControlDesk

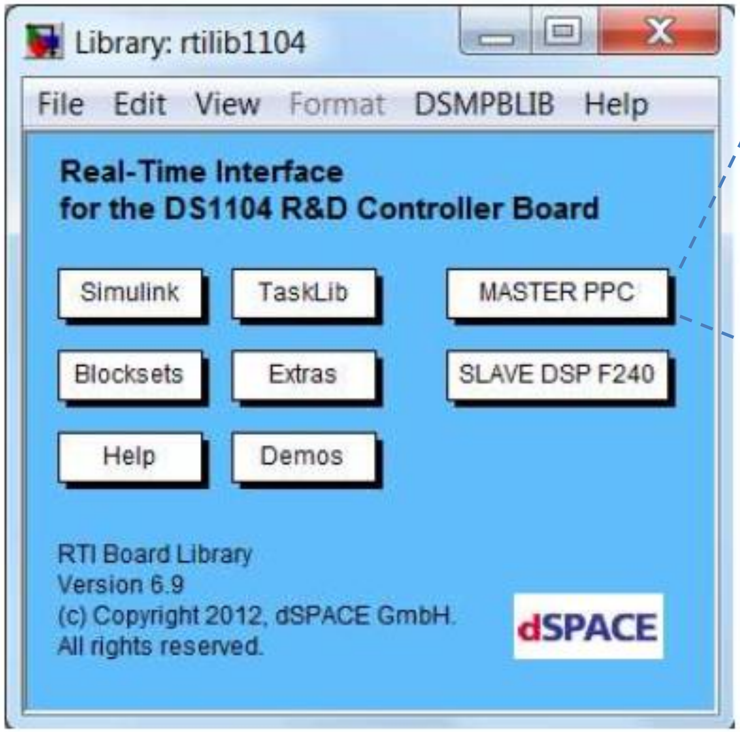
Matlab

MATLAB software is a tool for technical computing, computation and visualization in an integrated environment, e.g.,

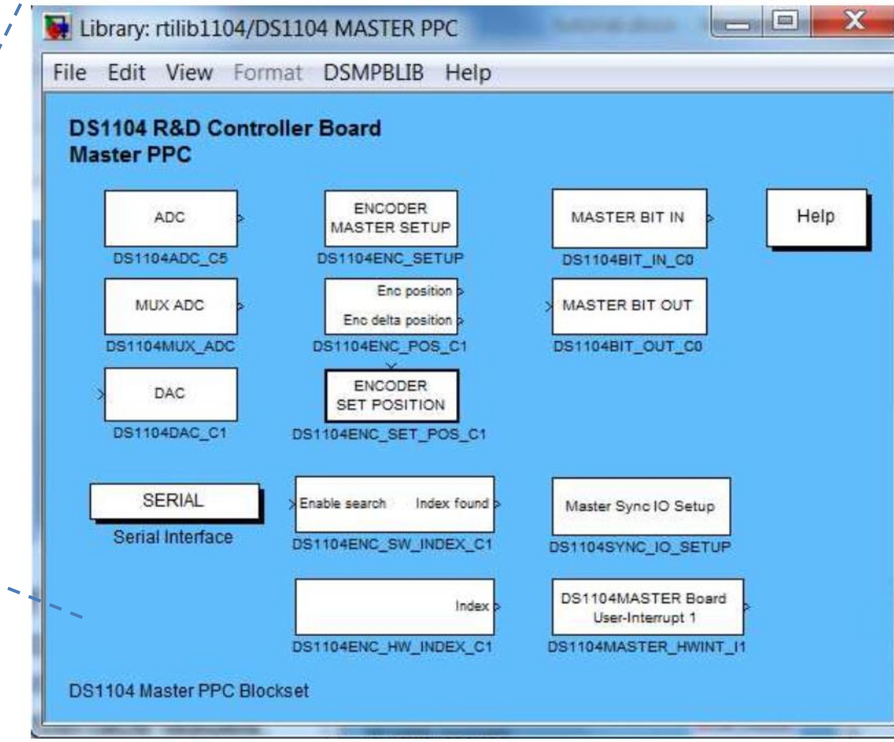
- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building



Matlab - RTI Library



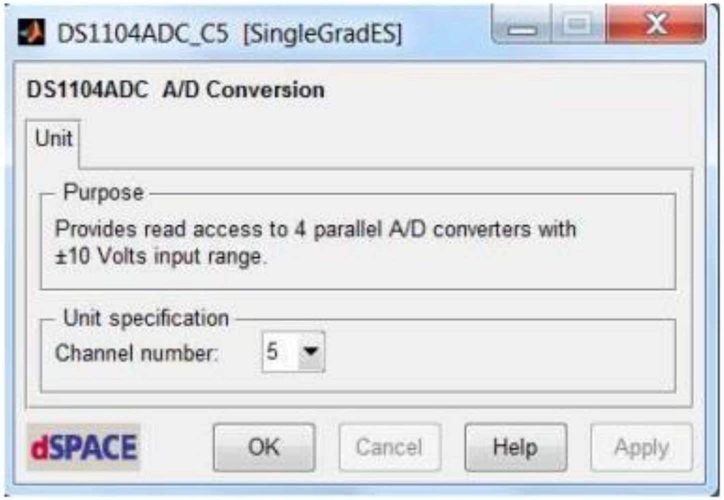
Main RTI Library



Master PPC Library

Matlab - Data acquisition toolbox

- Analog to Digital Conversion (ADC)



DS1104ADC_Cx

Purpose

To read from a single channel of one of 4 parallel A/D converter channels.

I/O mapping

For information on the I/O mapping, refer to ADC Unit.

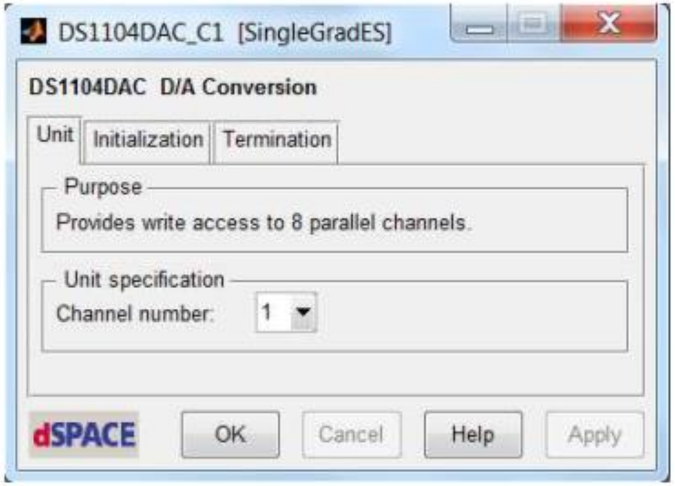
I/O characteristics

Scaling between the analog input voltage and the output of the block:

Input Voltage Range	Simulink Output
−10 V … +10 V	−1 … +1 (double)

Matlab and dSpace Platform

❑ Digital to Analog Conversion (DAC)



Purpose

To write to one of the 8 parallel D/A converter channels.

I/O mapping

For information on the I/O mapping, refer to DAC Unit.

I/O characteristics

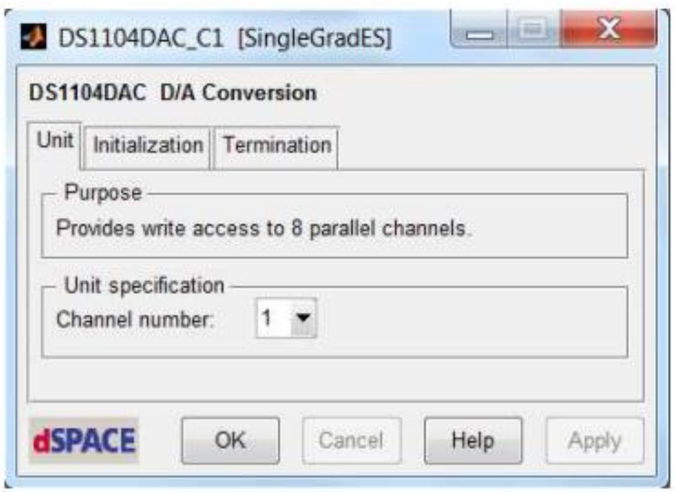
- Scaling between the analog output voltage and the input of the block:

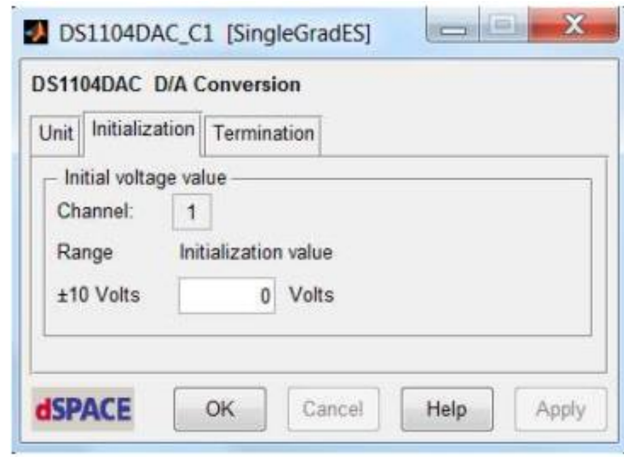
Simulink Input	Output Voltage Range
–1 ... +1 (double)	–10 V ... +10 V

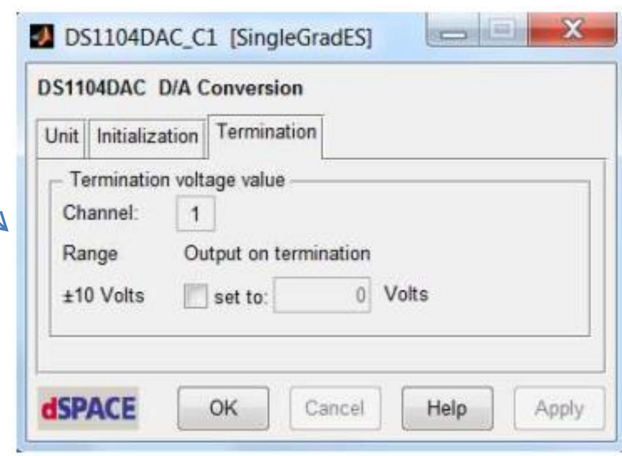
- The block provides its outputs in:
  - Transparent mode, that is the channel is converted and output immediately.
  - Latched mode, that is the channel is converted after synchronous triggering.

Matlab and dSpace Platform

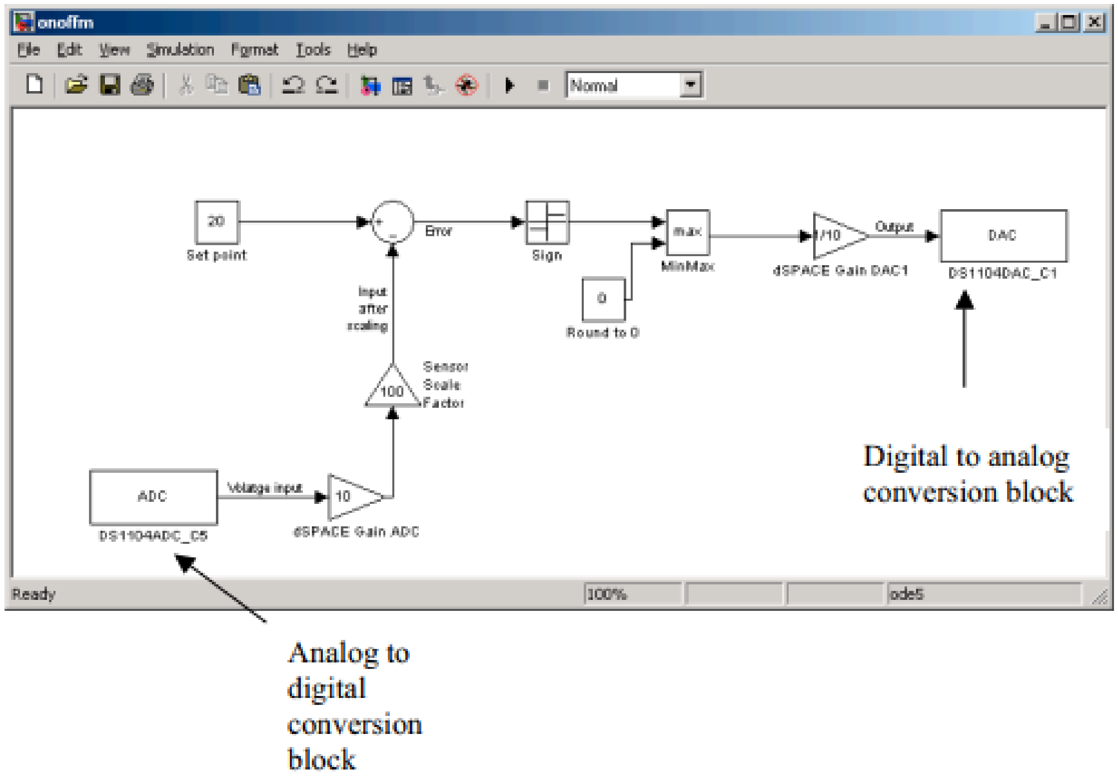
- Digital to Analog Conversion (DAC)





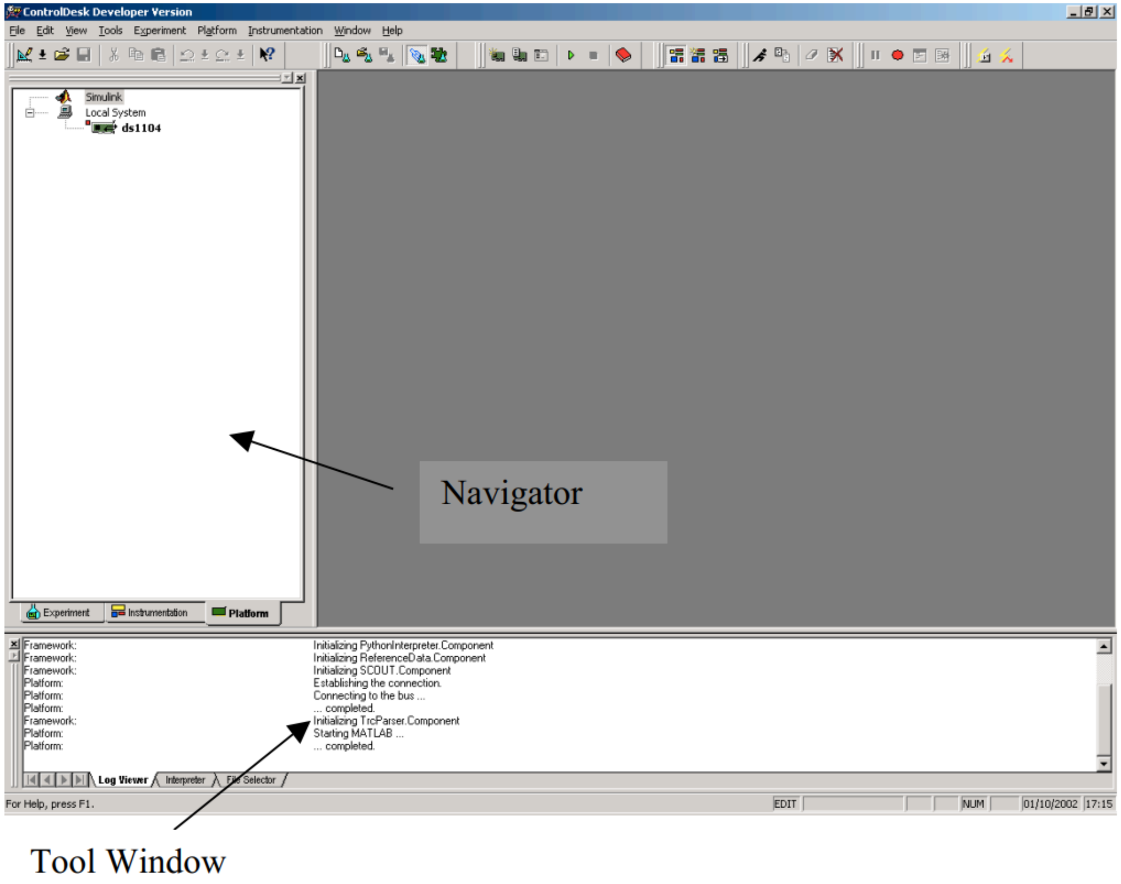


Example


onoffm
File Edit View Simulation Format Tools Help
Normal
20
Set point
Error
Sign
max
MinMax
0
Round to 0
1/10
Output
dSPACE Gain DAC1
DAC
DS1104DAC_C1
Input after scaling
100
Sensor Scale Factor
ADC
Voltage input
10
DS1104ADC_C5
dSPACE Gain ADC
Digital to analog conversion block
Ready
100%
ode5
Analog to digital conversion block

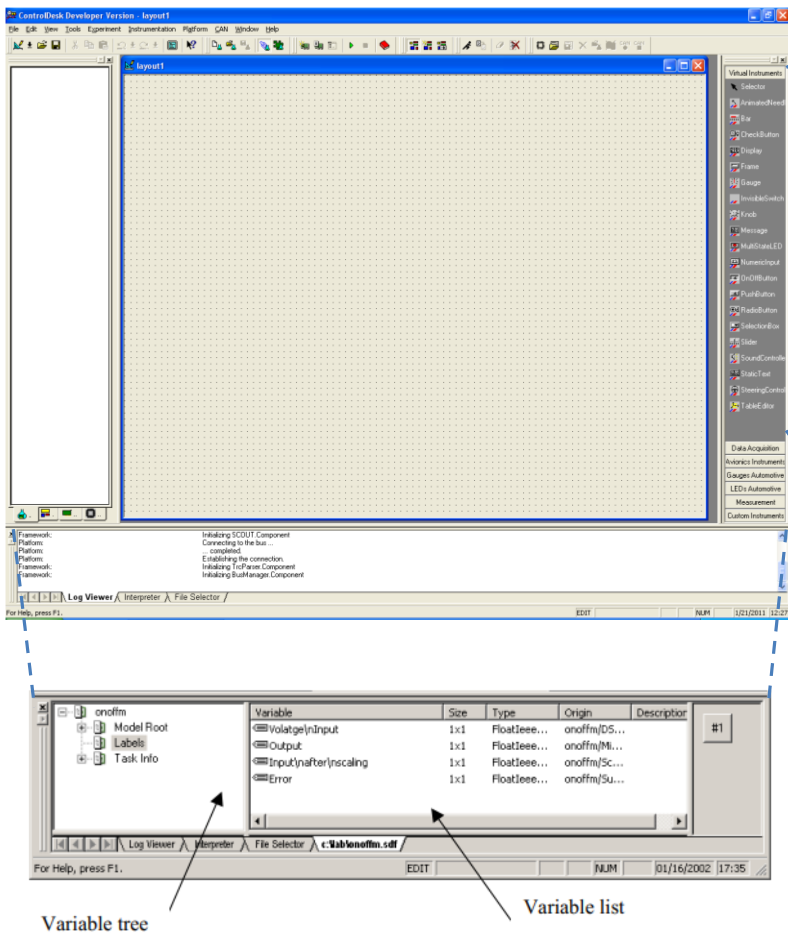
ControlDesk

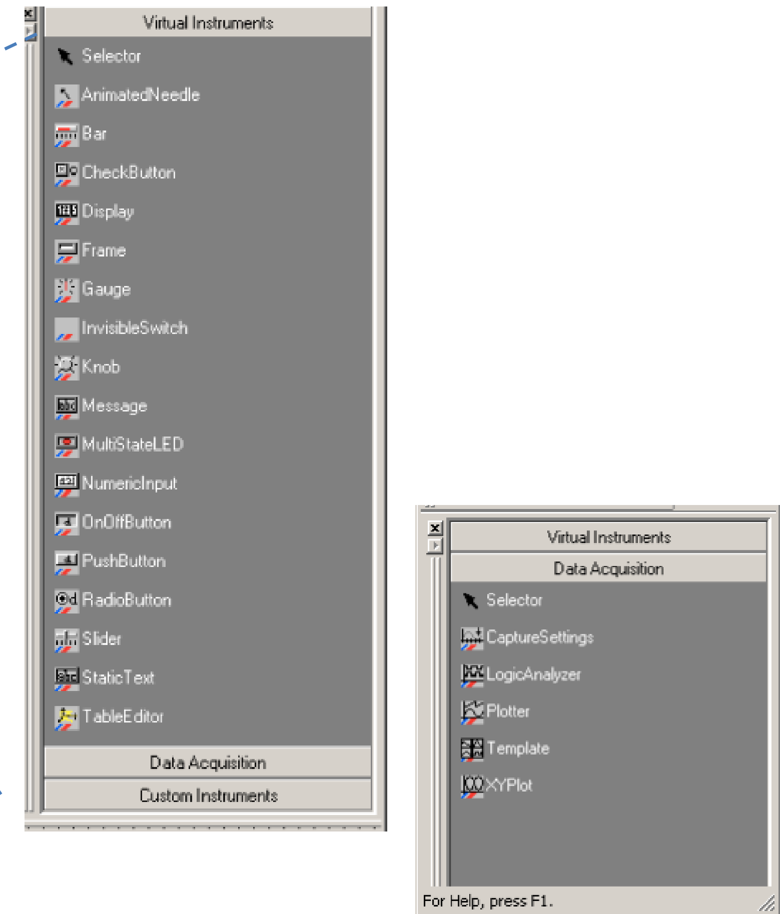
- ❑ Development of Graphical User Interface



ControlDesk

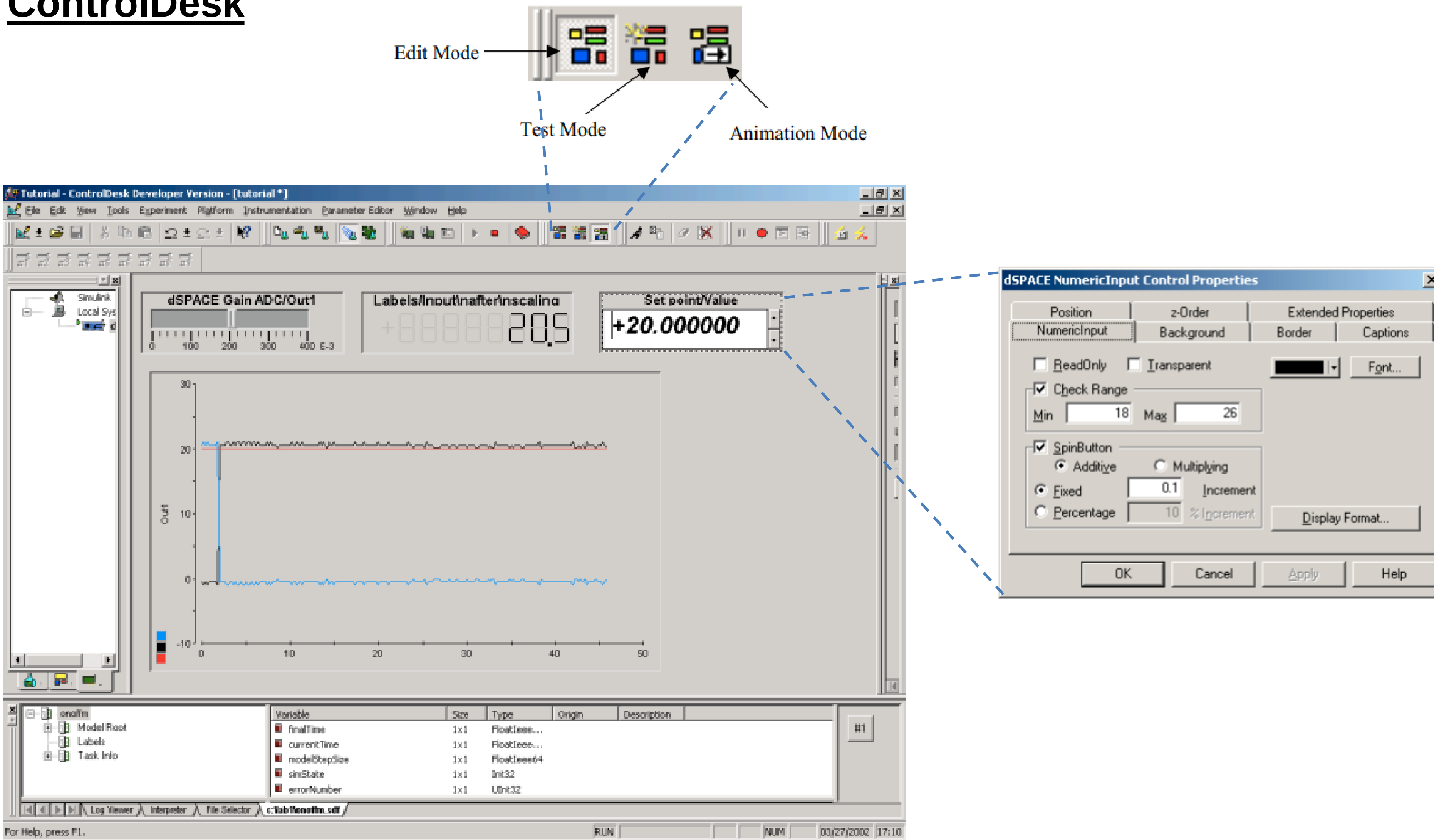
- Development of Graphical User Interfaces





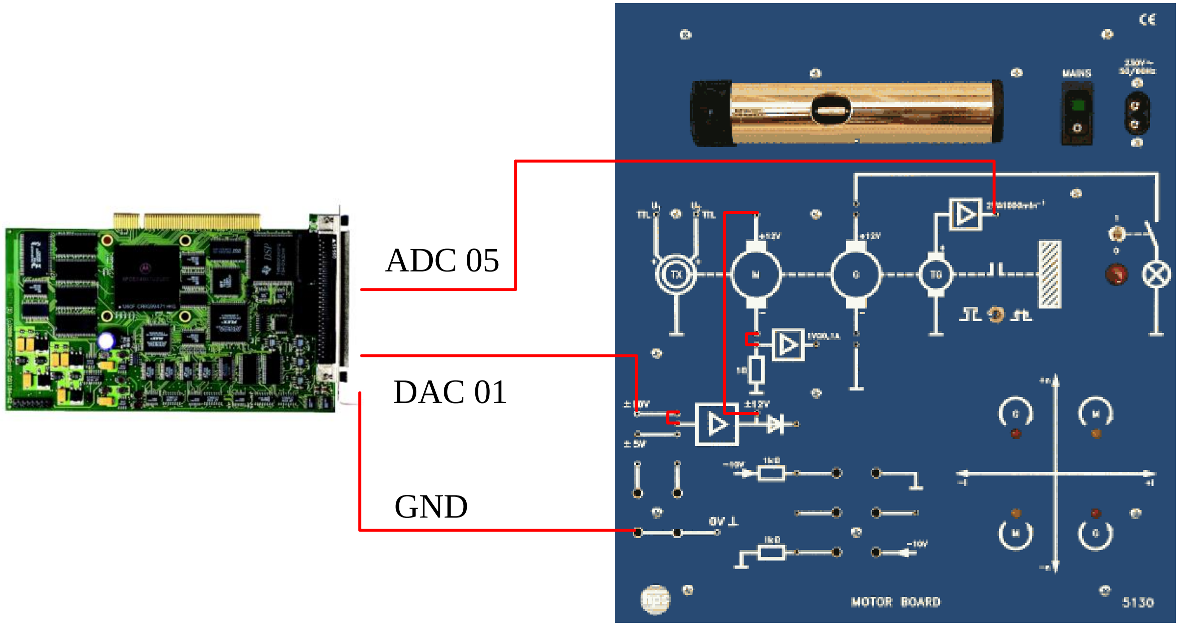
ControlDesk libraries

ControlDesk



Matlab and dSpace Platform

- Example

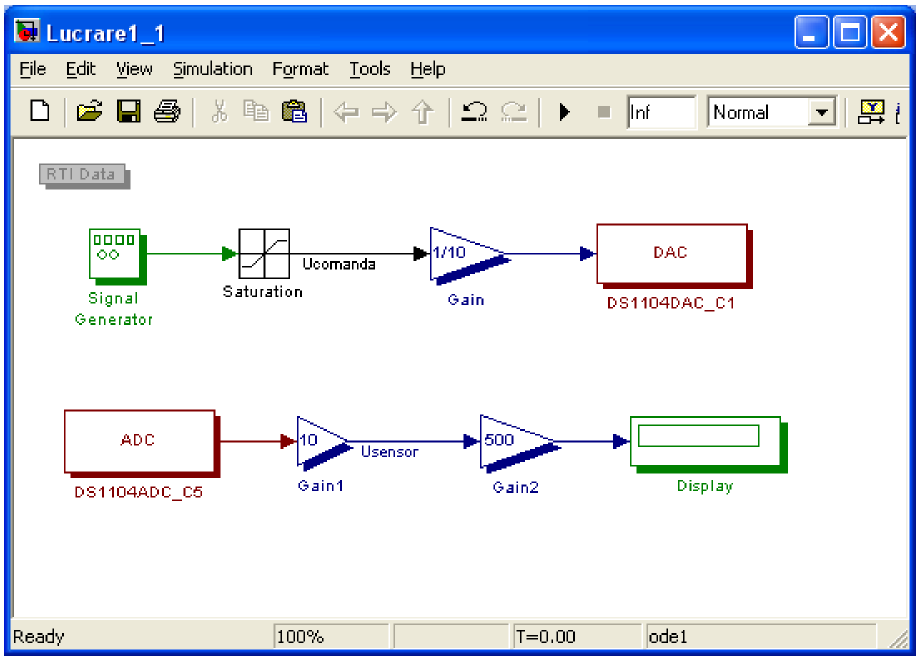


DS1104 **Motor Board**

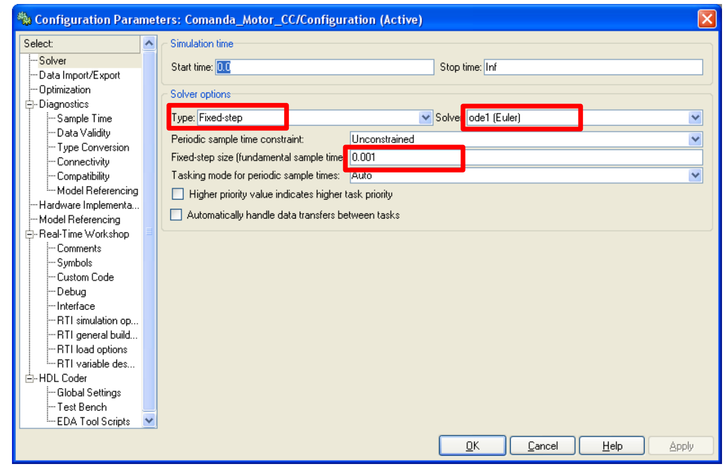
Hardware configuration

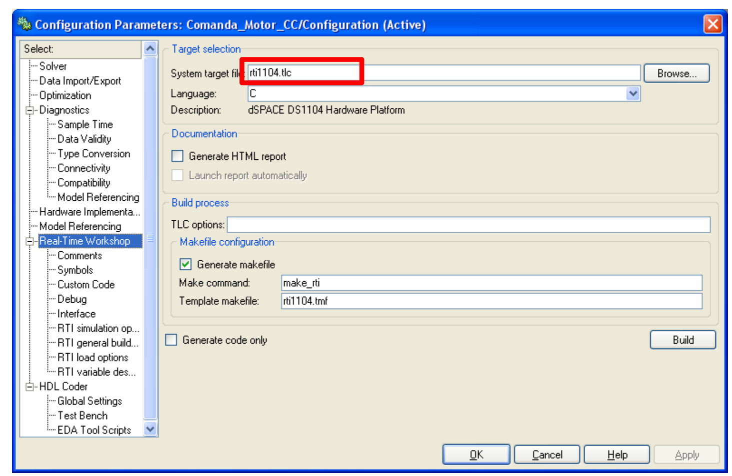
Matlab and dSpace Platform

- Example



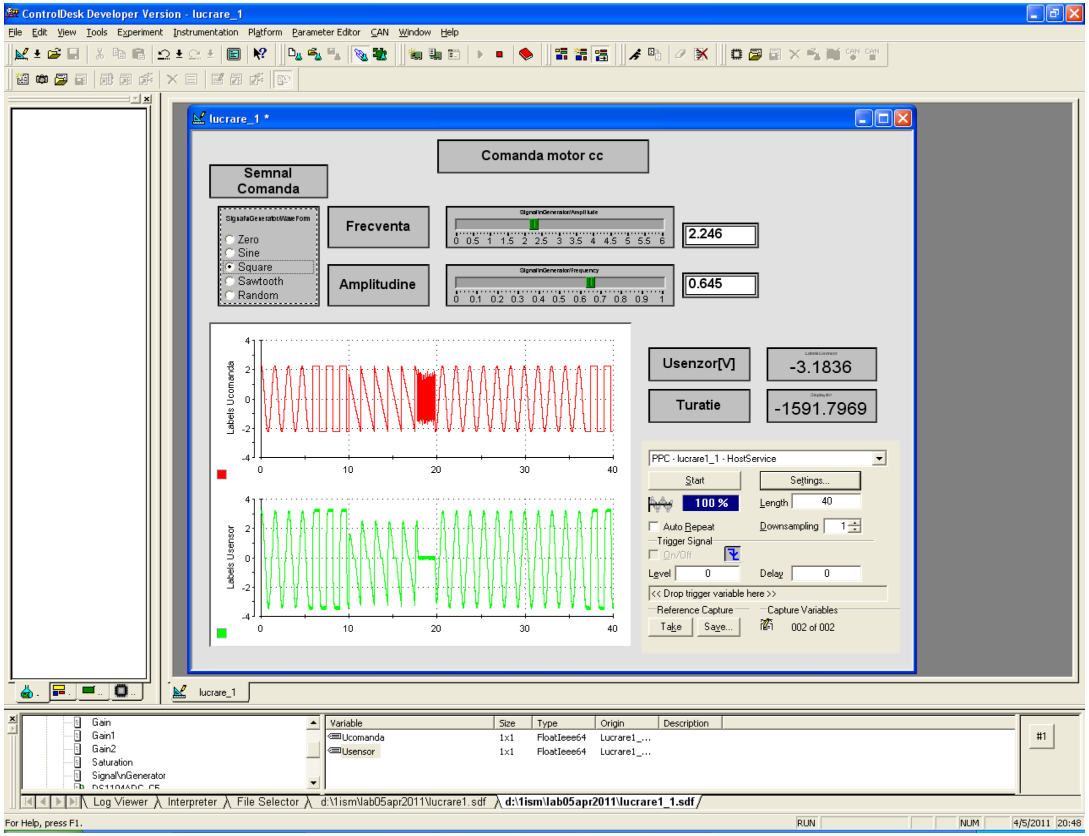
Simulink Model





Matlab and dSpace Platform

- Example



GUI developed in ControlDesk

Topics

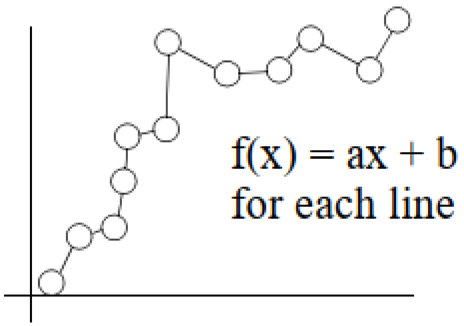
❑ Curve fitting methods

- Linear regression
- Higher order polynomial regression
- Overfit / Underfit

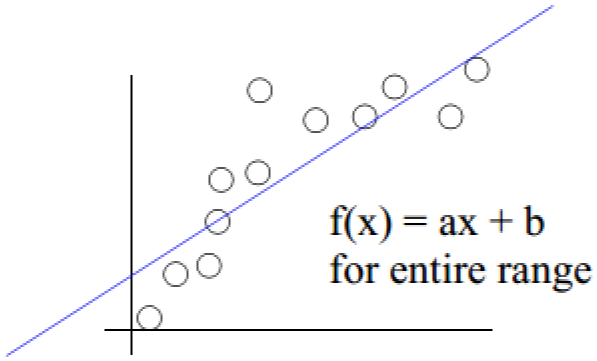
Curve fitting techniques

- Linear regression
- Higher order polynomial form
- Exponential form

Obj.: capturing the trend in the data by assigning a single function across the entire range



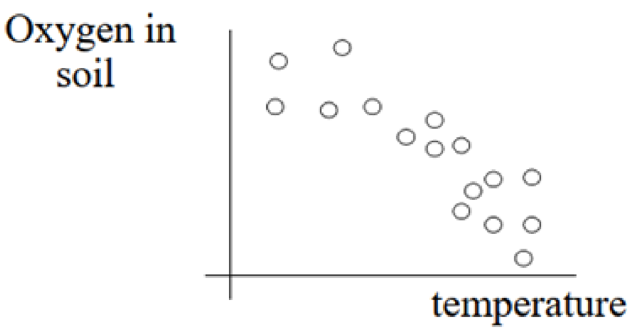
Interpolation

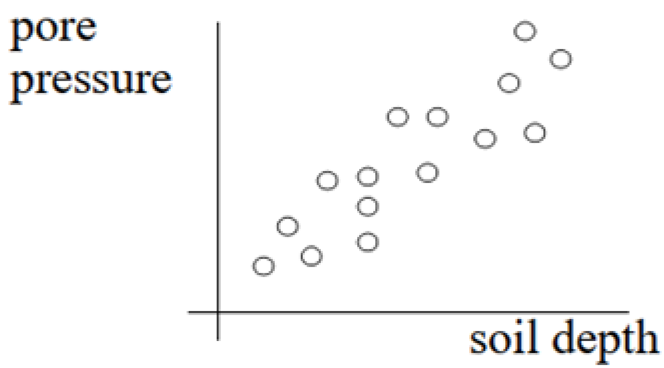


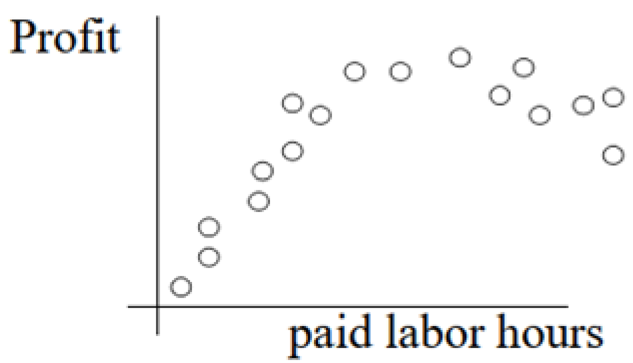
Curve Fitting

Curve fitting techniques

- Example of data sets that we can fit a function



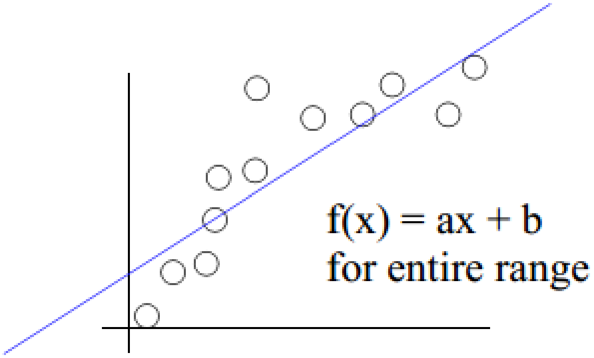




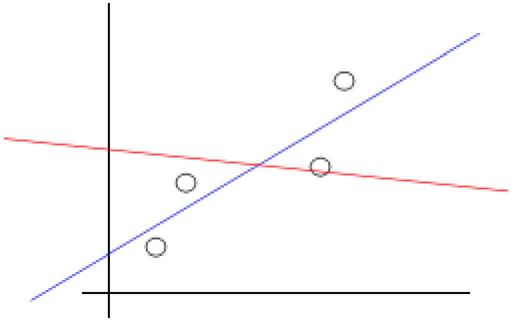
Linear Regression

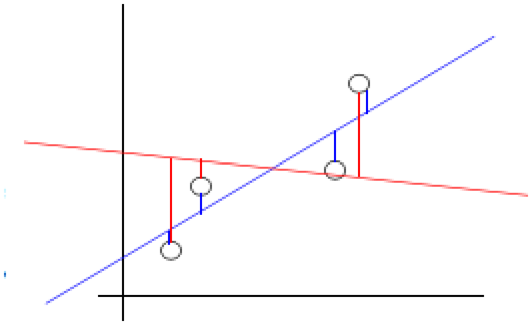
- A straight line is described generically by:

$$f(x) = ax + b$$



- **The goal** is to identify the coefficients 'a' and 'b' such that f(x) 'fits' the data optimum

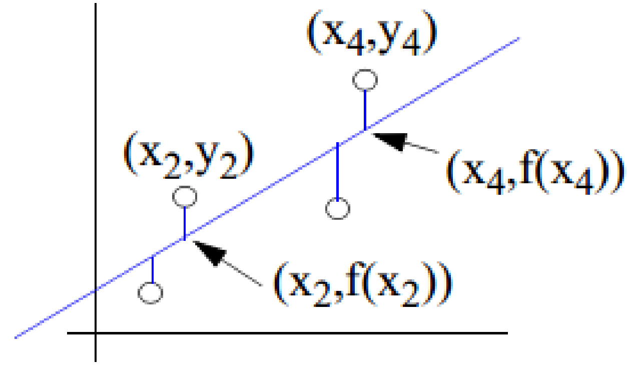




Linear Regression

Quantifying error in a curve fit

- Assumptions:
  - positive or negative error have the same value (data point is above or below the line)
  - Weight greater errors more heavily



- the error at the four data points is:

$$err = \sum (d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - f(x_4))^2$$

- it can be generalized and *f(x)* replaced by *ax+b*

$$err = \sum_{i=1}^{\#\text{ data points}} (y_i - f(x_i))^2 = \sum_{i=1}^{\#\text{ data points}} (y_i - (ax_i + b))^2$$

Linear Regression

- The optimum is obtained when the **error** between line and data points is **minimum (least squares approach)**

$$\text{minimize} \quad err = \sum_{i=1}^{\#\text{ data points} = n} (y_i - (ax_i + b))^2$$

- finding the minimum of the function (first derivative is zero)

$$\frac{\partial err}{\partial a} = -2 \sum_{i=1}^{n} x_i (y_i - ax_i - b) = 0$$

$$\frac{\partial err}{\partial b} = -2 \sum_{i=1}^{n} (y_i - ax_i - b) = 0$$

Linear Regression

- Solve for *a* and *b* so that the two equations are booth =0

$$\frac{\partial err}{\partial a} = -2\sum_{i=1}^{n} x_i(y_i - ax_i - b) = 0$$

$$\frac{\partial err}{\partial b} = -2\sum_{i=1}^{n} (y_i - ax_i - b) = 0$$

$$\Rightarrow$$

$$a\sum x_i^2 + b\sum x_i = \sum(x_i y_i)$$

$$a\sum x_i + b*n = \sum y_i$$

- We put the equations in matrix form

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum(x_i y_i) \end{bmatrix}$$

Linear Regression

- It is known:
  - No. of data points *n*
  - data points (x_i, y_i) {i=1..n}

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

- Using Gaussian elimination we obtain

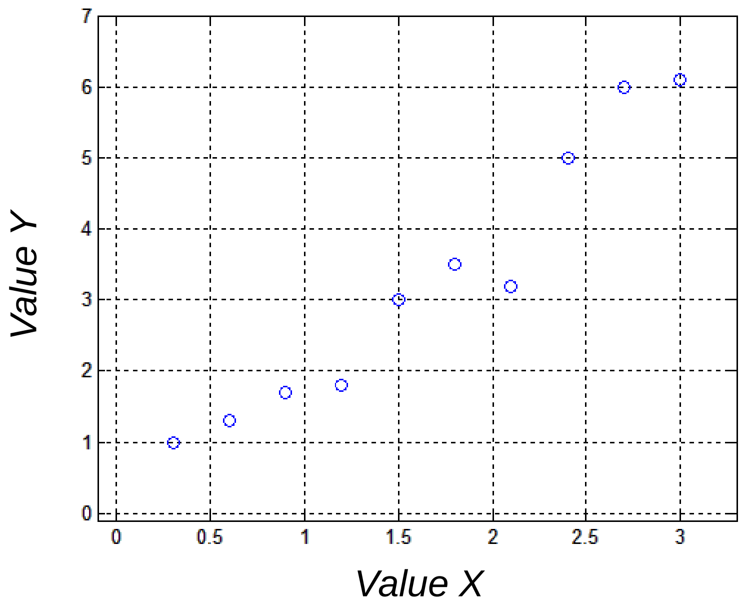
$$A = \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}, \quad X = \begin{bmatrix} b \\ a \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

$$AX = B \Rightarrow X = A^{-1} * B$$

Example I

- Fit a first order function to the following data

Value X	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3
Value Y	1	1.3	1.7	1.8	3	3.5	3.2	5	6	6.1



Example I

- First the summation terms are calculated

$$n = 10$$

$$\sum_{i=1}^{n} x_i = 16.5$$

$$\sum_{i=1}^{n} x_i^2 = 34.65$$

$$\sum_{i=1}^{n} y_i = 32.6$$

$$\sum_{i=1}^{n} x_i y_i = 68.79$$

$$\Rightarrow$$

$$A = \begin{bmatrix} 10.0 & 16.5 \\ 16.5 & 34.5 \end{bmatrix}$$

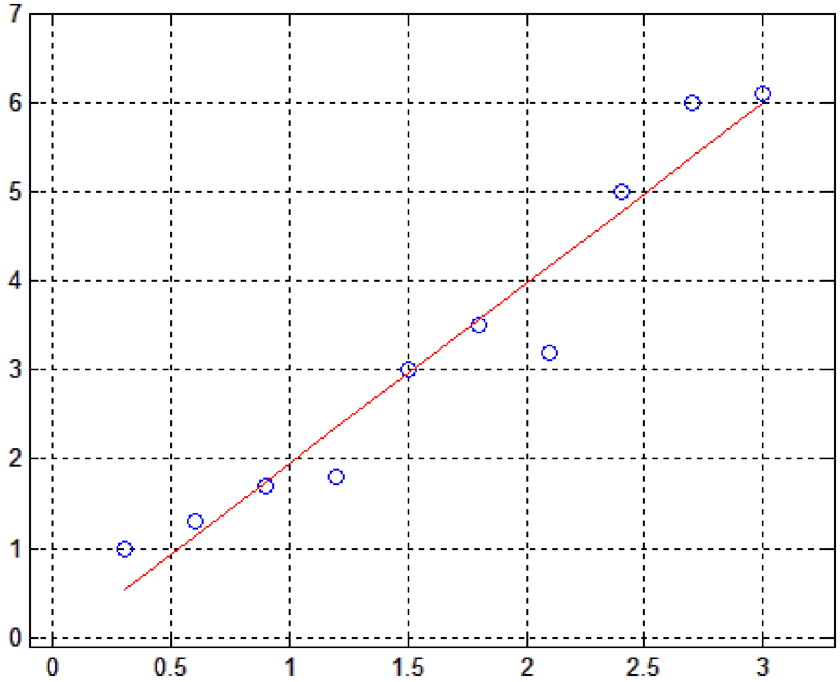
$$B = \begin{bmatrix} 32.6 \\ 68.79 \end{bmatrix}$$

Example I

- Finding the solutions

$$\begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 10.0 & 16.5 \\ 16.5 & 34.5 \end{bmatrix} \begin{bmatrix} 32.6 \\ 68.79 \end{bmatrix} \Rightarrow \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.0733 \\ 2.0202 \end{bmatrix}$$

$$f(x) = 2.0202\,x - 0.0733$$



Example I

```
clc;
clear all;
%% date intrare
x=[0.3 0.6 0.9 1.2 1.5 1.8 2.1 2.4 2.7 3];
y=[1 1.3 1.7 1.8 3 3.5 3.2 5 6 6.1];

plot(x,y,'ob')
grid on
axis([-0.1 3.3 -0.1 7])

%% determinare valori parametri matrici
n=numel(x)
sx=sum(x)
sx2=sum(x.^2)
sy=sum(y)
sxy=sum(x.*y)
```

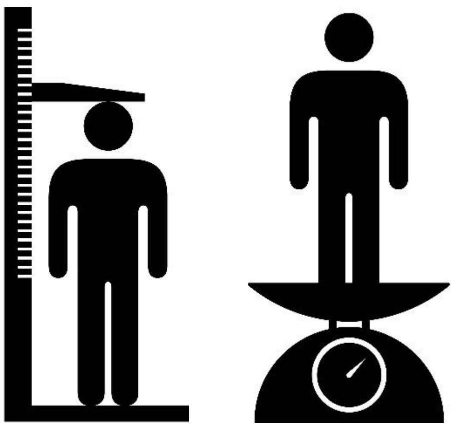
Example I

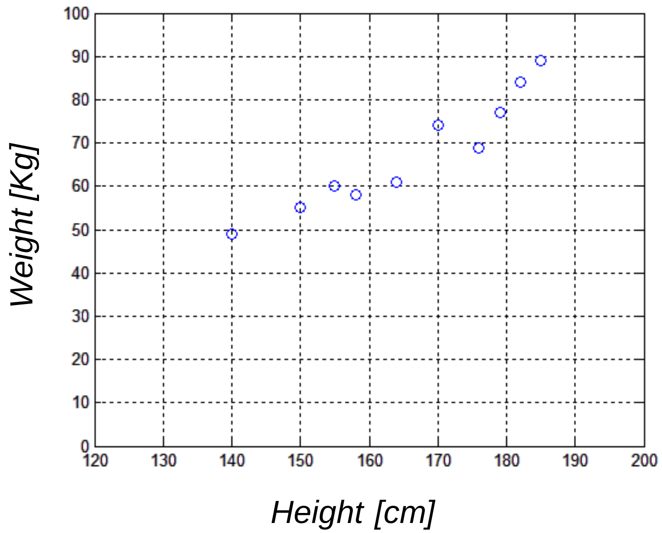
```
%% definire matrici
A=[ n sx;
    sx sx2]

B=[sy;
   sxy]

%calcul parametri

X=inv(A)*B

estimare=X(2,1)*x+X(1,1)
hold on
plot(x,estimare,'-r')
```

Example 2

Determine a mathematical function that estimate the relationship between the people’s height an their weight.

Height [cm]	140	150	155	158	164	170	176	179	182	185
Weight [kg]	49	55	60	58	61	74	69	77	84	89



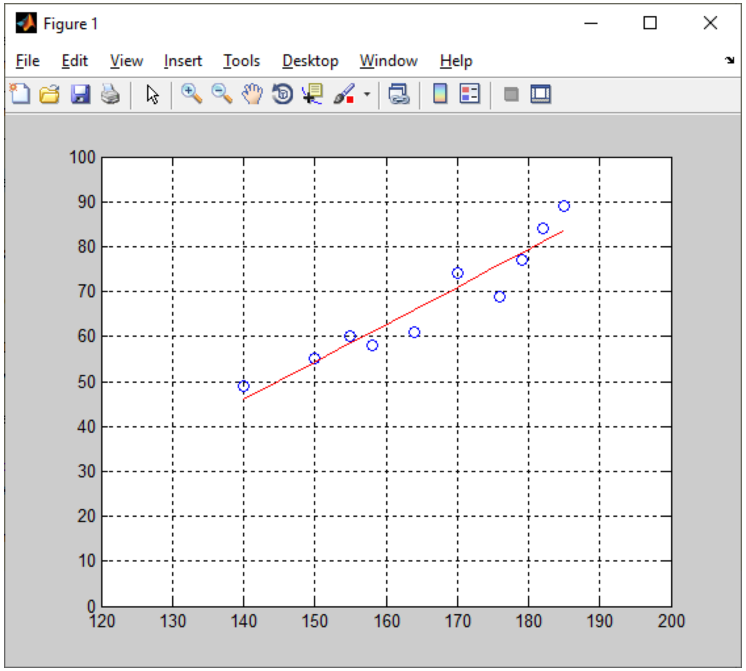


Example 2

Results:

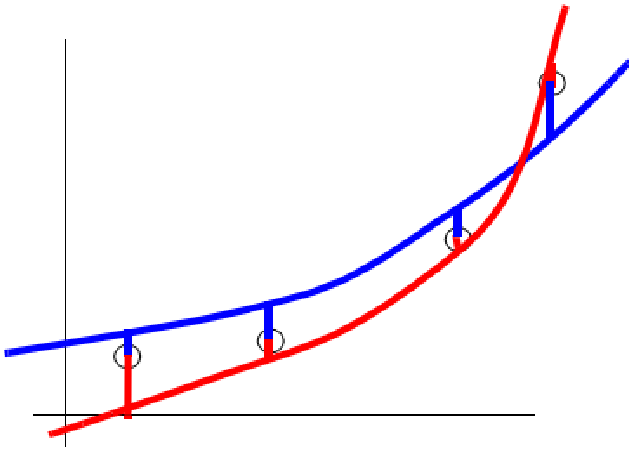
$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -70.9657 \\ 0.8352 \end{bmatrix}$$

$$f(x) = 0.8352\, x - 70.9657$$



Higher order polynomial

- Objective
  - Find the curve that gives minimum error between data *y* and the estimation function *f(x)*
  - Use a polynomial function



- Consider the general form for a polynomial of order *j*

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k$$

Higher order polynomial

- The general expression for any error using the least squares approach

$$err = \sum (d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - f(x_4))^2$$

- Substituting the polynomial equation general form in the above eq.

$$err = \sum_{i=1}^{n} \left(y_i - \left(a_0 + a_1 x_i + a_2 x_i^{2} + a_3 x_i^{3} + \dots + a_j x_i^{j} \right) \right)^2 \qquad (*)$$

where: *n* – no. of measured data points
i – the current data point being summed
j – the polynomial order

Higher order polynomial

- To minimize eq. (*), take the derivative with respect to each coefficient set each to zero

$$\frac{\partial err}{\partial a_0} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) = 0$$

$$\frac{\partial err}{\partial a_1} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x = 0$$

$$\frac{\partial err}{\partial a_2} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x^2 = 0$$

$$\vdots$$

$$\vdots$$

$$\frac{\partial err}{\partial a_j} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x^j = 0$$

Higher order polynomial

- We put the equations in matrix form

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \cdots & \sum x_i^{j+j} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum \left(x_i^2 y_i\right) \\ \vdots \\ \sum \left(x_i^j y_i\right) \end{bmatrix}$$

Higher order polynomial

- Using Gaussian elimination we obtain

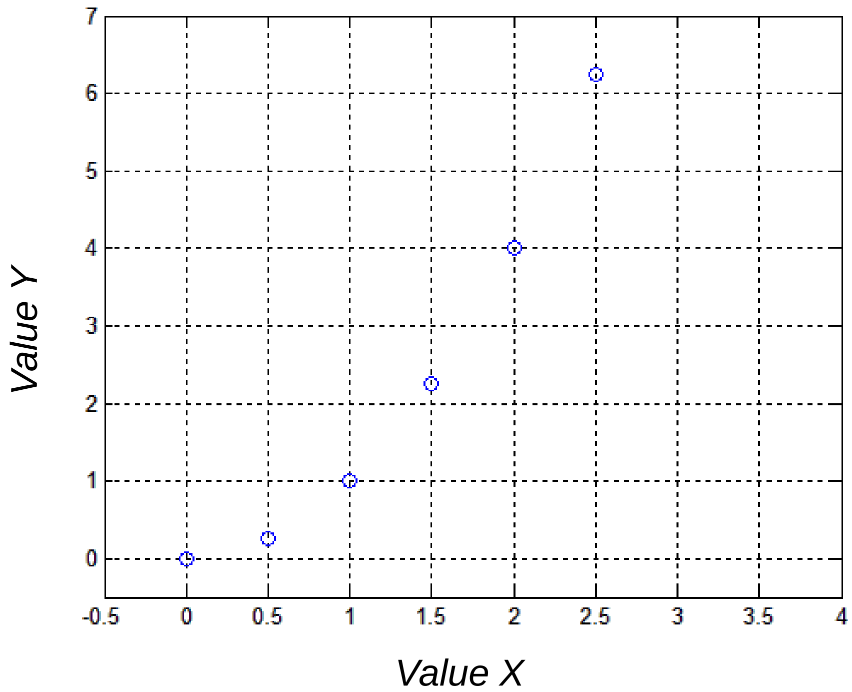
$$A = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \dots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \dots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \dots & \sum x_i^{j+j} \end{bmatrix}, \quad X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum \left(x_i^2 y_i\right) \\ \vdots \\ \sum \left(x_i^j y_i\right) \end{bmatrix}$$

$$AX = B \Longrightarrow X = A^{-1} * B$$

Example III

- Fit a second order polynomial to the following data

Value X	0	0.5	1	1.5	2.0	2.5
Value Y	0	0.25	1	2.25	4	6.25



Example III

- First the summation terms are calculated

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

$$n = 6$$

$$\sum_{i=1}^{n} x_i = 7.50$$

$$\sum_{i=1}^{n} x_i^2 = 13.75$$

$$\sum_{i=1}^{n} x_i^3 = 28.12$$

$$\sum_{i=1}^{n} x_i^4 = 61.18$$

$$A = \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.12 \\ 13.75 & 28.12 & 61.18 \end{bmatrix}$$

$$\sum_{i=1}^{n} y_i = 13.75$$

$$\sum_{i=1}^{n} x_i y_i = 28.12$$

$$\sum_{i=1}^{n} x_i^2 y_i = 61.18$$

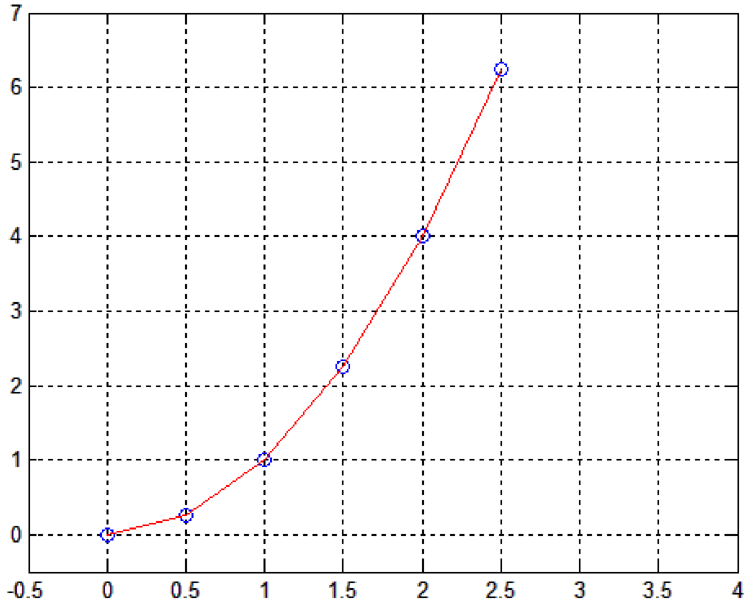
$$B = \begin{bmatrix} 13.75 \\ 28.12 \\ 61.18 \end{bmatrix}$$

Example III

- Finding the solutions $X = A^{-1} \cdot B$

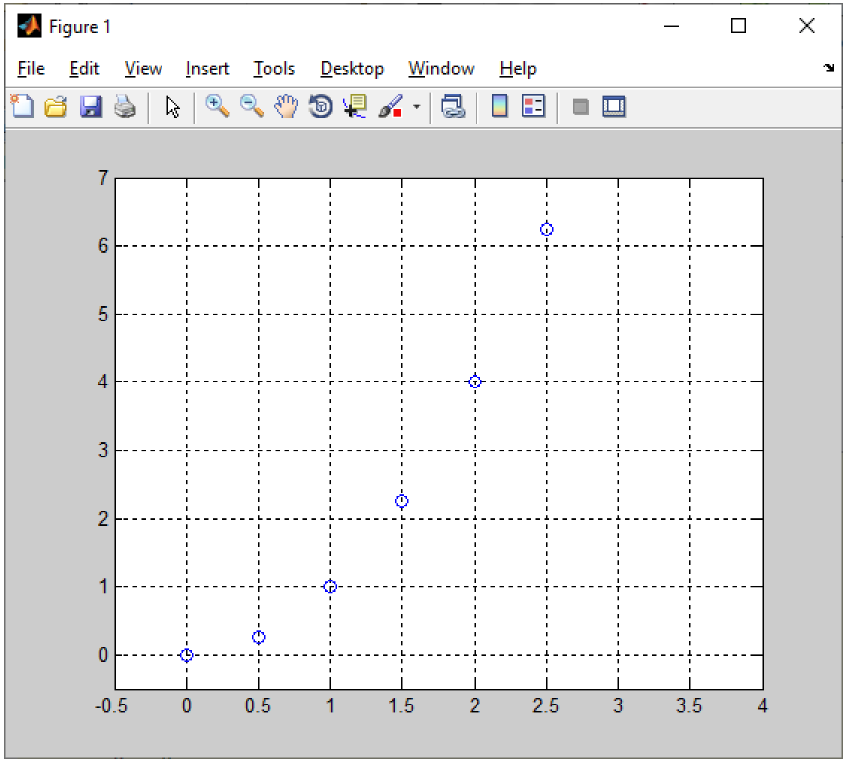
$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = inv \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.12 \\ 13.75 & 28.12 & 61.18 \end{bmatrix} \begin{bmatrix} 13.75 \\ 28.12 \\ 61.18 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}$$

$$f(x) = x^2$$



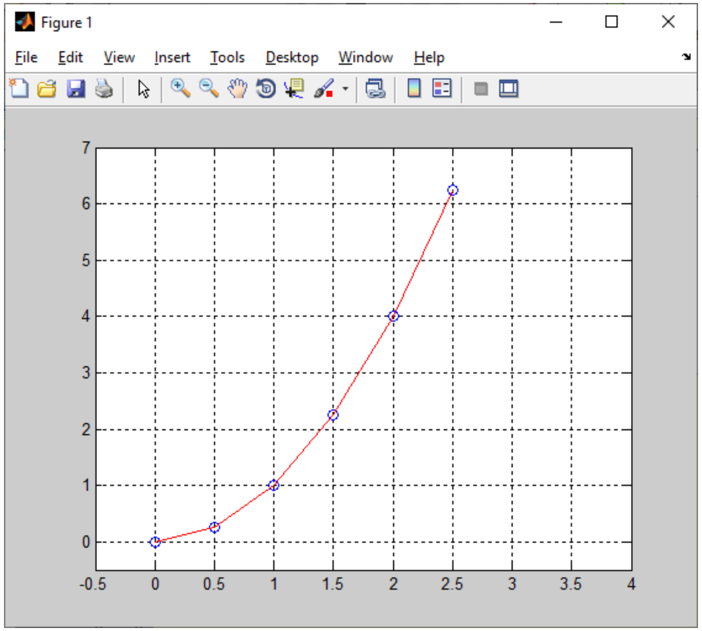
Example III

```
clc
clear all
%% date intrare
x=[0 0.5 1 1.5 2 2.5];
y=[0 0.25 1 2.25 4 6.25];

plot(x,y,'ob');
axis([-0.5 4 -0.5 7]);
grid on
```



Example III

```
%% regresie polinomiala ord. 2
n=numel(x);
sx=sum(x);
sx2=sum(x.^2);
sy=sum(y);
sxy=sum(x.*y);
sx3=sum(x.^3);
sx4=sum(x.^4);
x2=x.^2;
sx2y=sum(x2.*y);
A=[n sx sx2;sx sx2 sx3; sx2 sx3 sx4];
B=[sy; sxy; sx2y];
X=inv(A)*B
rez2=X(1,1)+X(2,1)*x+X(3,1)*x.^2;
hold on
plot(x,rez2,'r');
```



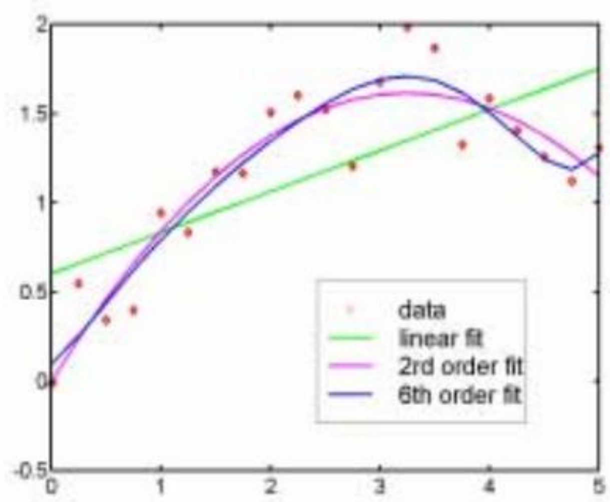
Higher order polynomial

Overfit / Underfit - picking an inappropriate order

Overfit - over-doing the requirement for the fit to 'match' the data trend (order too high)

Consideration #1:

3rd order - 1 inflection point
4th order - 2 inflection points
nth order - n-2 inflection points



Consideration #2:

2 data points - linear touches each point
3 data points - second order touches each point
n data points - n-1 order polynomial will touch each point

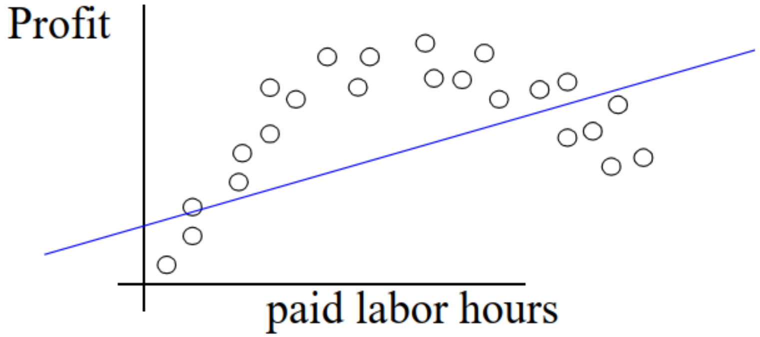
Higher order polynomial

General rule: pick a polynomial form at least several orders lower than the number of data points.
Start with linear and add order until trends are matched.

Underfit - If the order is too low to capture obvious trends in the data

General rule: View data first, then select an order that reflects inflections, etc.

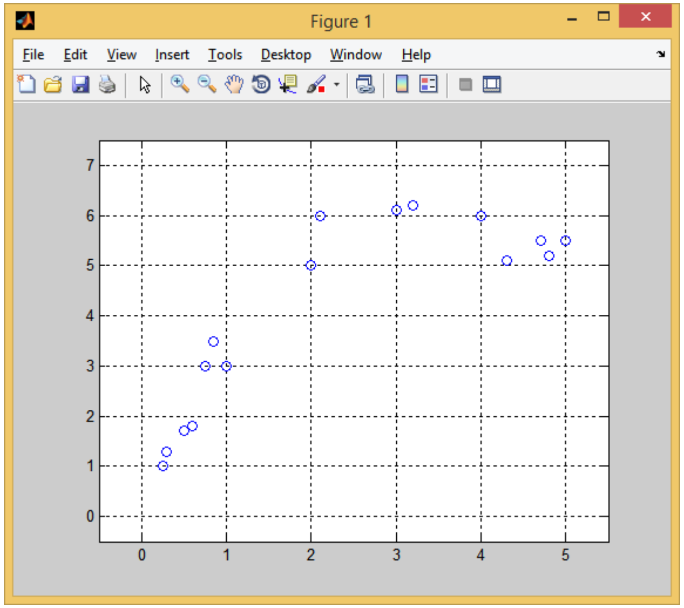
For this example :
1) Obviously nonlinear, so order > 1
2) No inflection points observed as obvious, so order < 3 is recommended



Example IV - Higher order polynomial

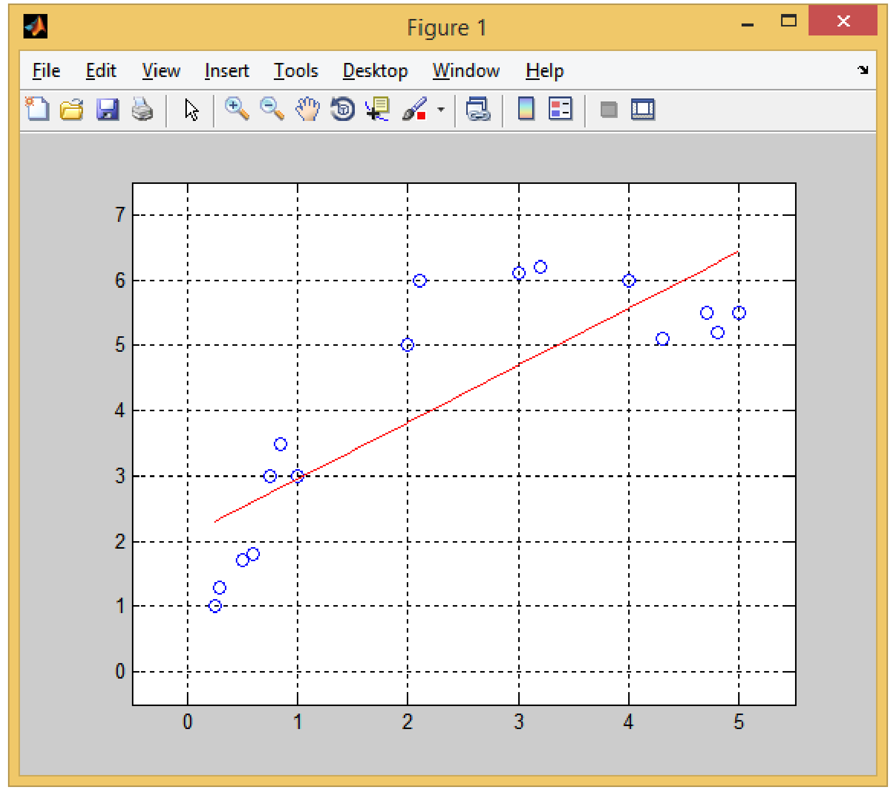
X	0.25	0.3	0.5	0.6	0.75	0.85	1	2	2.1	3	3.2	4	4.3	4.7	4.8	5
Y	1	1.3	1.7	1.8	3	3.5	3	5	6	6.1	6.2	6	5.1	5.5	5.2	5.5

```
%% date intrare
x=[0.25 0.3 0.5 0.6 0.75 0.85 1 2 2.1 3 3.2 4 4.3 4.7
4.8 5];
y=[1 1.3 1.7 1.8 3 3.5 3 5 6 6.1 6.2 6 5.1 5.5 5.2 5.5];

plot(x,y,'ob');
axis([-0.5 5.5 -0.5 7.5]);
grid on
```



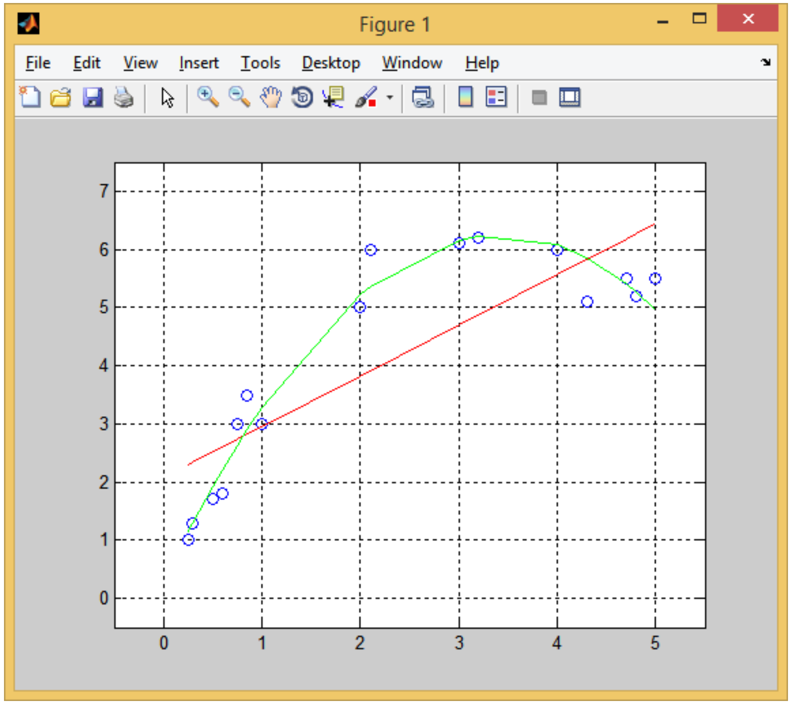
Example IV - Higher order polynomial

```
%% regresie liniara
n=numel(x);
sx=sum(x);
sx2=sum(x.^2);
sy=sum(y);
sxy=sum(x.*y);
A=[n sx; sx sx2];
B=[sy; sxy];

X=inv(A)*B;

rez1=X(1,1)+X(2,1)*x;
hold on
plot(x,rez1,'b');
```



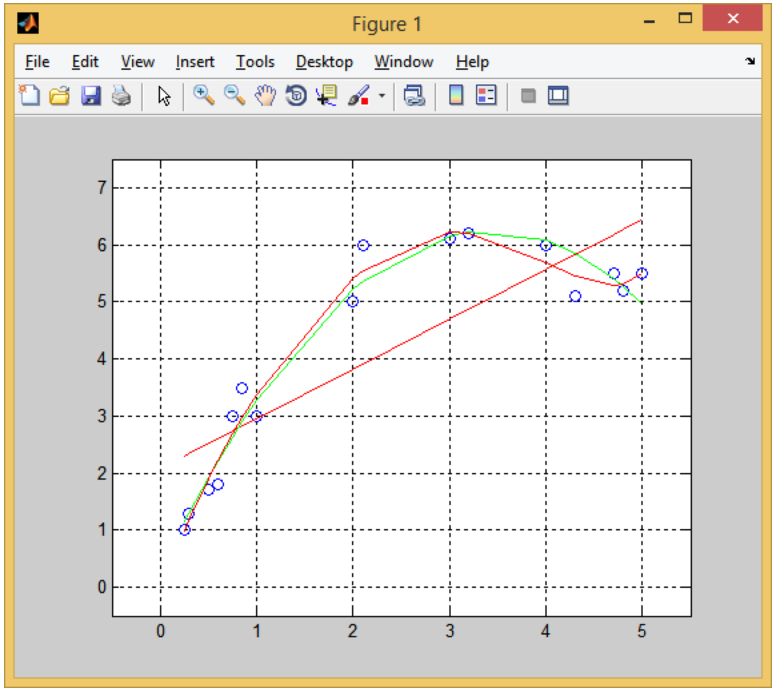
Example IV - Higher order polynomial

```
%% regresie polinomiala ord. 2
sx3=sum(x.^3);
sx4=sum(x.^4);
x2=x.^2;
sx2y=sum(x2.*y);
A=[n sx sx2;
   sx sx2 sx3;
   sx2 sx3 sx4];
B=[sy; sxy; sx2y];
X=inv(A)*B;

rez2=X(1,1)+X(2,1)*x+X(3,1)*x.^2;
plot(x,rez2,'g');
```



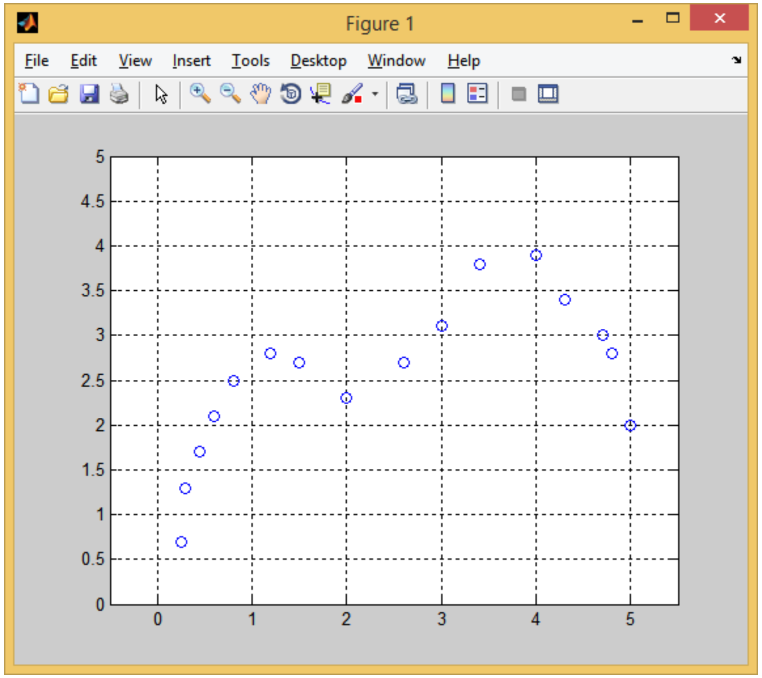
Example IV - Higher order polynomial

```
%% regresie polinomiala ord. 5
sx5=sum(x.^5);
sx6=sum(x.^6);
sx7=sum(x.^7);
sx8=sum(x.^8);
sx9=sum(x.^9);
sx10=sum(x.^10);
x3=x.^3;
sx3y=sum(x3.*y);
x4=x.^4;
sx4y=sum(x4.*y);
x5=x.^5;
sx5y=sum(x5.*y);

A=[n sx sx2 sx3 sx4 sx5;
   sx sx2 sx3 sx4 sx5 sx6;
   sx2 sx3 sx4 sx5 sx6 sx7;
   sx3 sx4 sx5 sx6 sx7 sx8;
   sx4 sx5 sx6 sx7 sx8 sx9;
   sx5 sx6 sx7 sx8 sx9 sx10];
B=[sy; sxy; sx2y; sx3y; sx4y; sx5y];
X=inv(A)*B;

rez3=X(1,1)+X(2,1)*x+X(3,1)*x.^2+X(4,1)*x.^3+X(5,1)*x.^4+X(6,1)*x.^
5;
plot(x,rez3,'r');
```

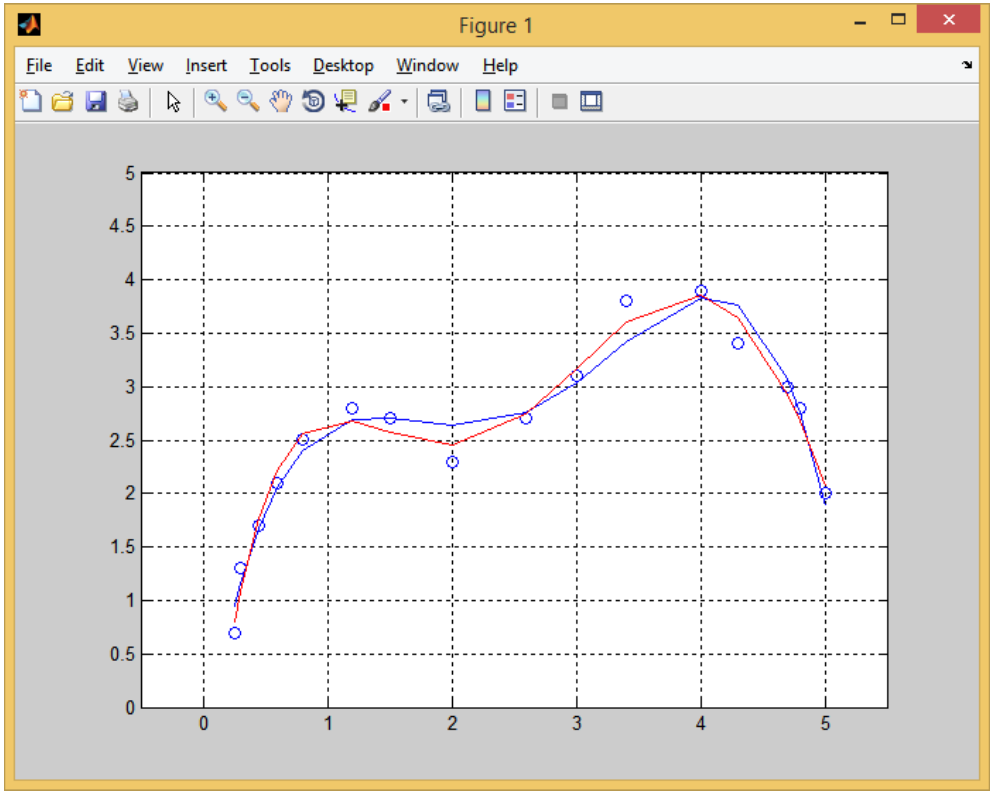


Example V - Higher order polynomial
===

X	0.25	0.3	0.45	0.6	0.8	1.2	1.5	2	2.6	3	3.4	4	4.3	4.7	4.8	5
Y	0.7	1.3	1.7	2.1	2.5	2.8	2.7	2.3	2.7	3.1	3.8	3.9	3.4	3	2.8	2



Example V - Higher order polynomial

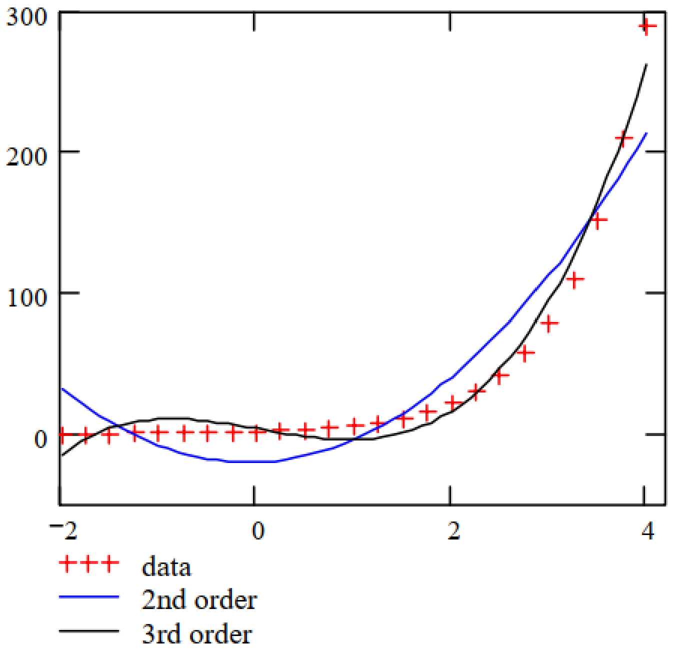


Exponential regression

- general exponential equation

$$f(x) = Ce^{Ax} = C\exp(Ax)$$

- seek C and A such that this equation fits the data as best it can



Error:

$$err = \sum_{i=1}^{n} (y_i - (C\exp(Ax)))^2 \qquad (*)$$

Exponential regression

When eq. * is partial derivate with respect to and set to zero, we get two **nonlinear** equations with respect to C and A

Solution #1: Nonlinear equation solving methods

- use Newton Raphson to solve a system of nonlinear equations

Solution #2: Linearization:

- change of variables to re-cast this as a linear problem
- Find: a function to fit data of the general exponential form

$$y = Ce^{Ax}$$

Exponential regression

Solution #2: Linearization:

- Take logarithm of both sides to get rid of the exponential

$$\ln(y) = \ln(Ce^{Ax}) = Ax + \ln(C)$$

- Introduce the following change of variables:

$$Y = \ln(y), \quad X = x, \quad B = \ln(C)$$

- Now we have the following function, which is linear

$$Y = AX + B$$

Exponential regression

Solution #2: Linearization:

The original data points in the plane *x-y* get mapped into the plane *X-Y*.

The data is transformed as:

$$(x, y) \Rightarrow (X, Y) = (x, \ln(y))$$

Now the method for solving a first order linear curve fit is used, for A and B,

$$\begin{bmatrix} n & \sum X \\ \sum X & \sum X^2 \end{bmatrix} \begin{bmatrix} B \\ A \end{bmatrix} = \begin{bmatrix} \sum Y \\ \sum XY \end{bmatrix}$$

!Obs: $Y = \ln(y)$

$X = x$

Exponential regression

Solution #2: Linearization:

$$B = \ln(C) \Longrightarrow C = e^{B}$$

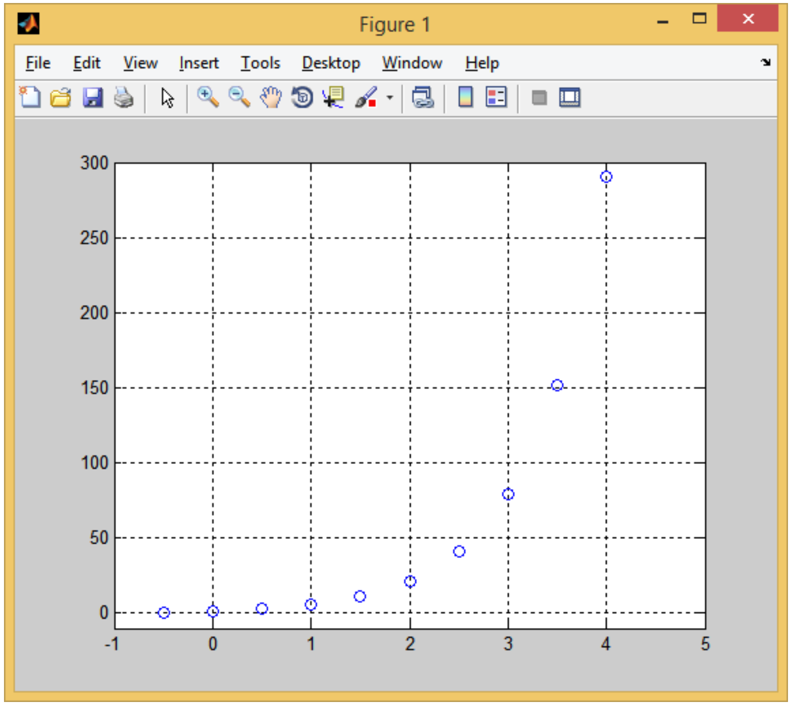
The coefficients for the exponential function are:

$$y = Ce^{Ax}$$

Exponential regression Example

X= -2.00 -1.50 -1.00 -0.50 0 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00

Y=0.1188 0.2276 0.4361 0.8353 1.6000 3.0649 5.8709 11.2459 21.5420 41.2645 79.0439 151.4119 290.0356

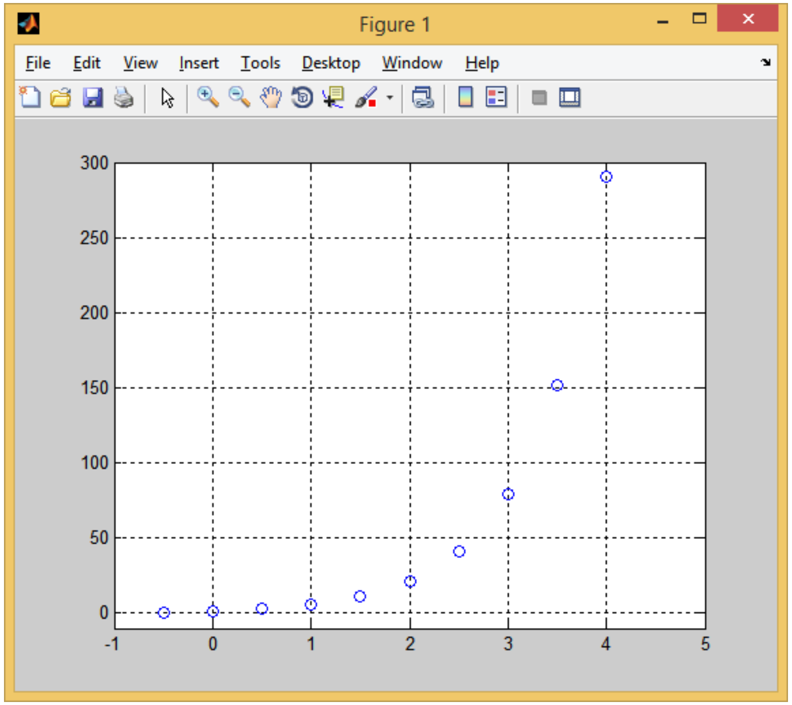


Exponential regression Example

X= -2.00   -1.50   -1.00   -0.50   0   0.50   1.00   1.50   2.00 2.50   3.00   3.50   4.00

Y=0.1188   0.2276   0.4361   0.8353   1.6000   3.0649   5.8709   11.2459   21.5420   41.2645   79.0439 151.4119 290.0356

```
%% initializare date
clc;
clear all;

X=-2:0.5:4;
Y=1.6*exp(1.3*X);

disp (X);
disp(Y);

plot(X,Y,'ob')
grid on
axis([-3 5 -10 300]);
```

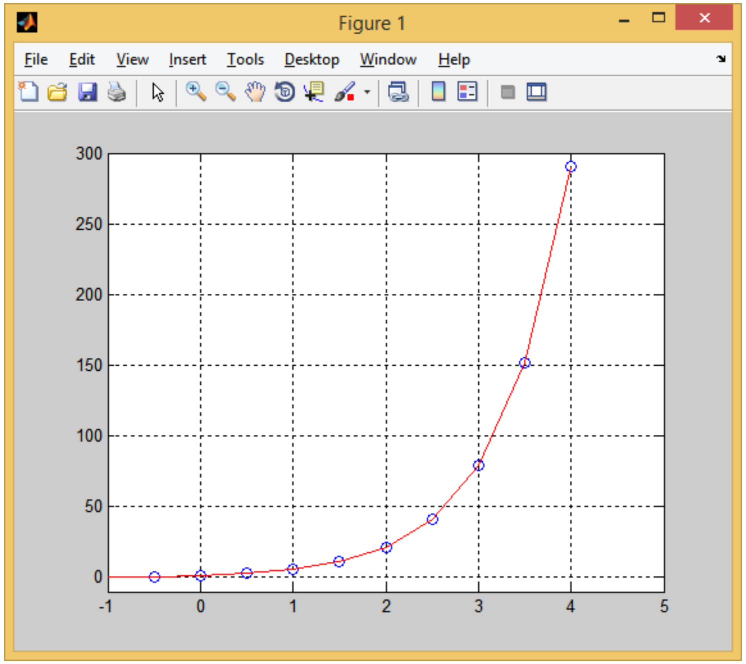


Exponential regression Example

```
%% determinare parametri

n=numel(X)
sX=sum(X)
sX2=sum(X.^2)
Y2=log(Y)
sY=sum(Y2)
XY=X.*Y2
sXY=sum(XY)

A=[n sX; sX sX2]
B=[sY ; sXY]

M=inv(A)*B;
```

Exponential regression Example

```
%% calcul parametri
a=M(2,1)
c=exp(M(1,1))

%%plotare grafic
rasp=c*exp(a*X)

hold on
plot(X,rasp,'r')
```



Evaluation methods for the obtained models

Residuals

- “Left-over” variations in the response after fitting the regression line are called residuals.

- A residual is the difference in the observed value of the response variable and the value predicted by the regression line.

Residual = observed y – predicted y

Evaluation methods for the obtained models

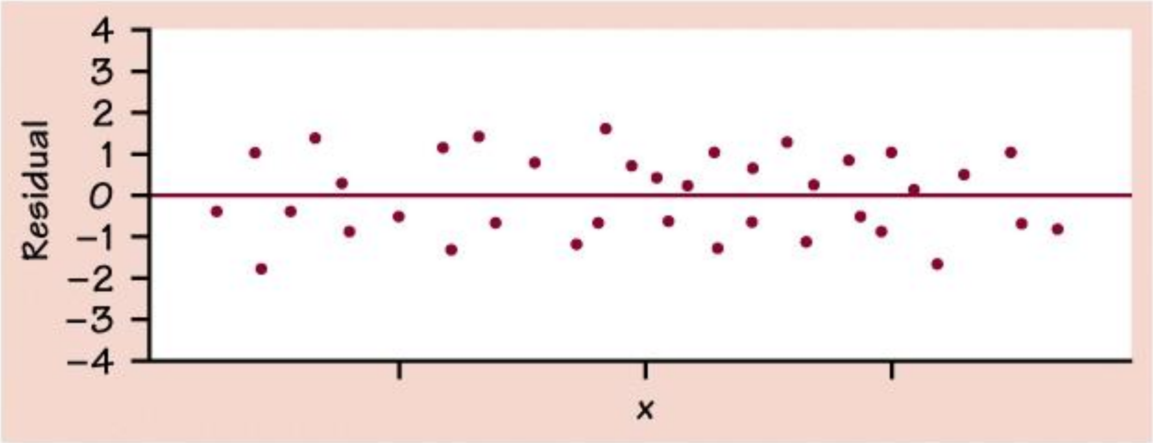
Residuals

- A Residual Plot is a scatterplot of the regression residuals against the explanatory variable.
- They help us assess the fit of a regression line.

- If the regression line captures the overall relationship between x and y, the residuals should have no systematic pattern.

Evaluation methods for the obtained models

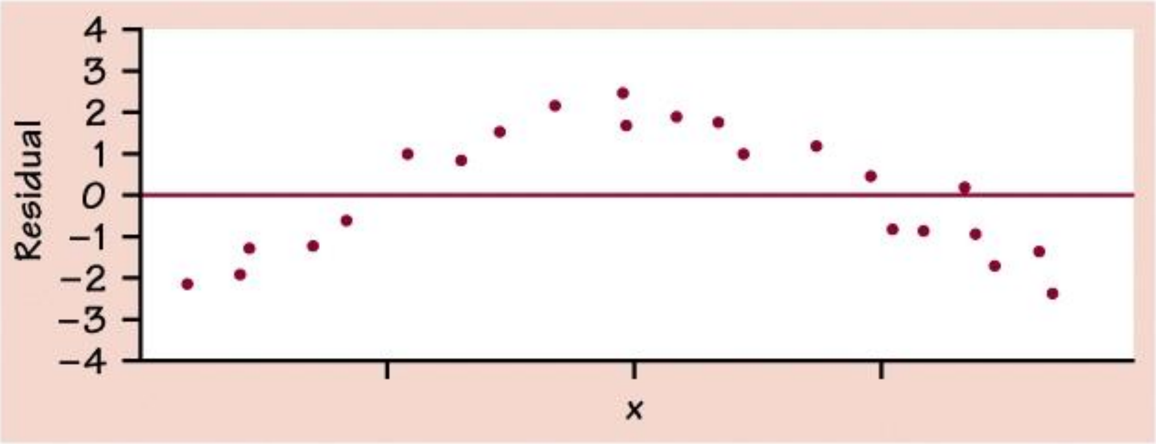
Residuals



- The uniform scatter of points indicates that the regression line fits the data well, so the line is a good model.

Evaluation methods for the obtained models

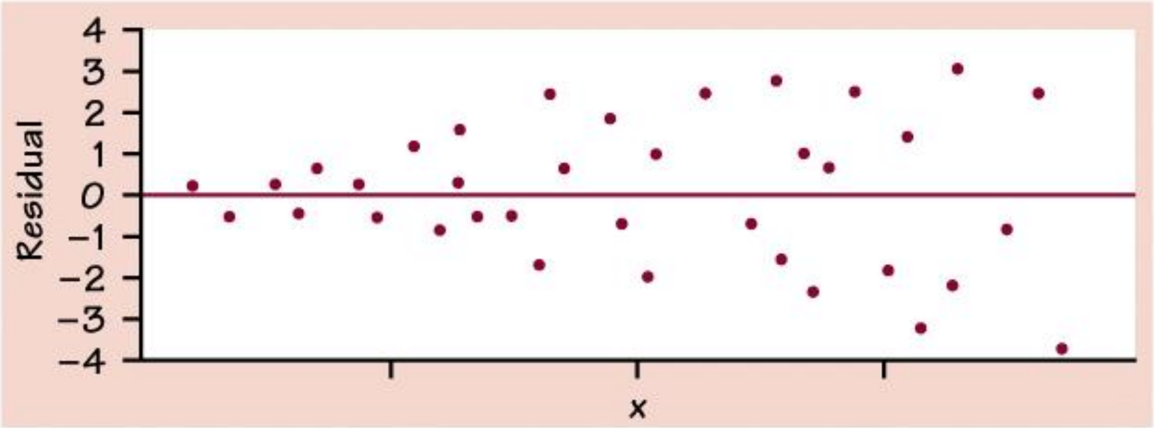
Residuals



- A curved pattern shows that the relationship is not linear.

Evaluation methods for the obtained models

Residuals

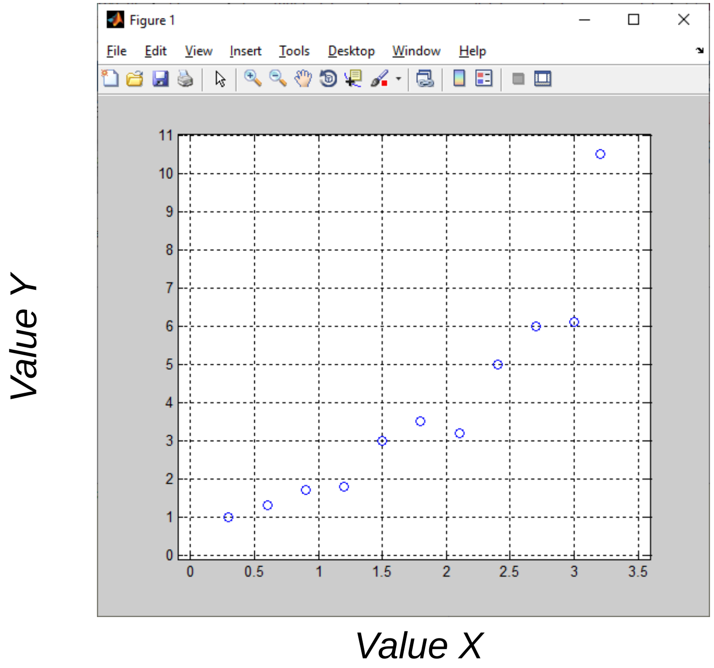


- Increasing or decreasing spread about the line. The response variable y has more spread for larger values of the explanatory variable x, so the prediction will be less accurate when x is large.

Example

- Fit a first order function to the following data

Value X	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3	3.2
Value Y	1	1.3	1.7	1.8	3	3.5	3.2	5	6	6.1	10.5



Example

- Fit a first order function to the following data

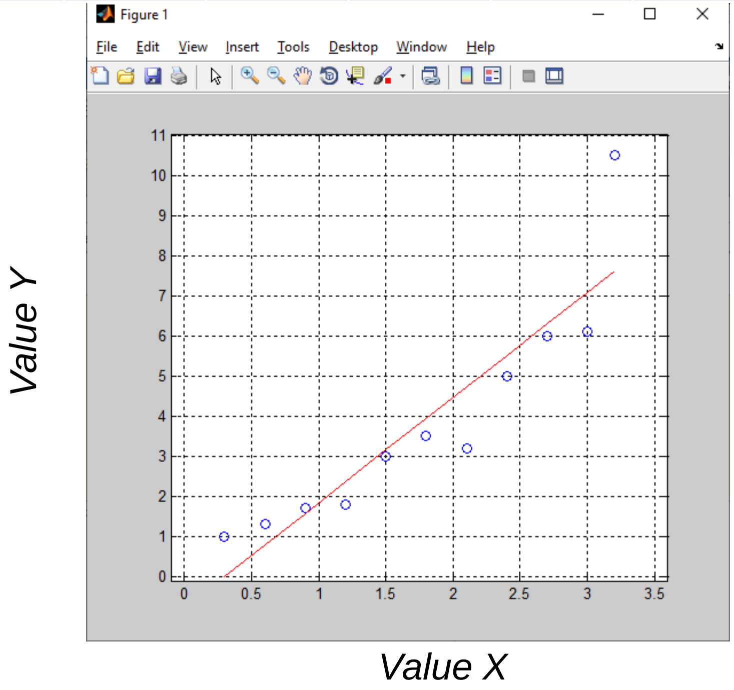
Value X	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3	3.2
Value Y	1	1.3	1.7	1.8	3	3.5	3.2	5	6	6.1	10.5

- Finding the solutions

$$\begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 10.0 & 16.5 \\ 16.5 & 34.5 \end{bmatrix} \begin{bmatrix} 32.6 \\ 68.79 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.0733 \\ 2.0202 \end{bmatrix}$$

$$f(x) = 2.0202\, x - 0.0733$$

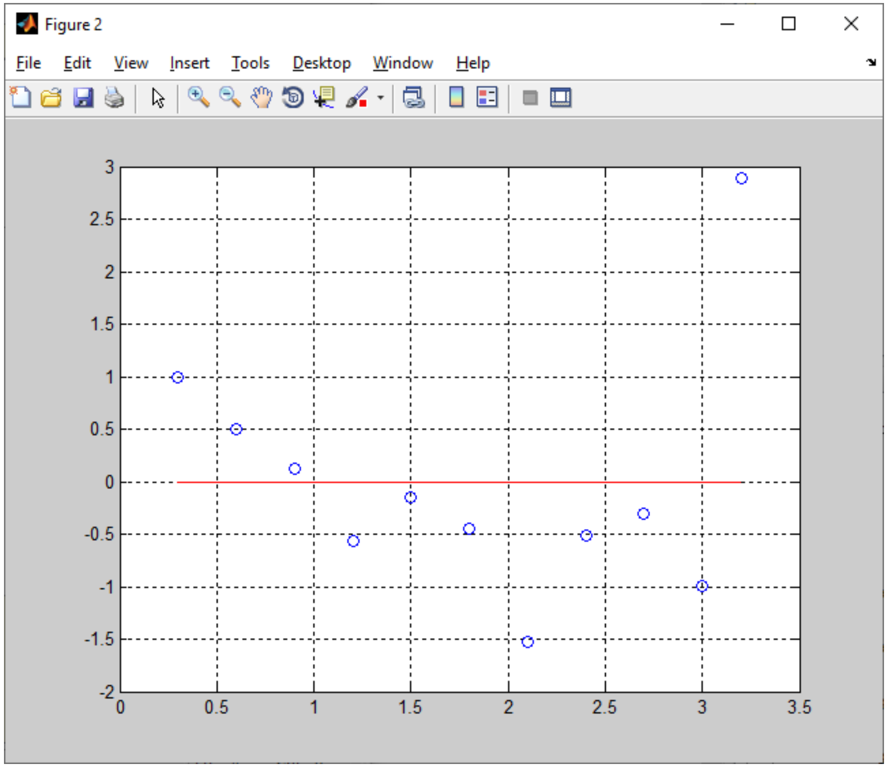


Example – Matlab Implementation

```
clc;
clear all;
%% date intrare
x=[0.3 0.6 0.9 1.2 1.5 1.8 2.1 2.4 2.7 3 3.2];
y=[1 1.3 1.7 1.8 3 3.5 3.2 5 6 6.1 10.5];
figure(1);
hold off
plot(x,y,'ob')
grid on
axis([-0.1 3.6 -0.1 11])
%% determinare valori parametri matrici
n=numel(x)
sx=sum(x)
sx2=sum(x.^2)
sy=sum(y)
sxy=sum(x.*y)
%% definire matrici
A=[ n sx;
    sx sx2]
B=[sy;
   sxy]

%calcul parametri
X=inv(A)*B
estimare=X(2,1)*x+X(1,1)
hold on
plot(x,estimare,'-r');
```

Example – Matlab Implementation

```
%% calcul residuals
residuals=y-estimare;
figure(2);
vzero=zeros(1,numel(x));
plot(x, vzero,'-r',x,residuals,'ob');
grid on
```

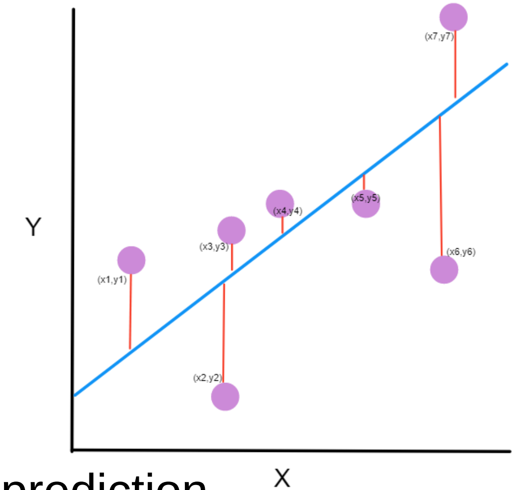


Errors in Regressions

Mean squared error (MSE)

- measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



n - number of data points used to generate the prediction
$\hat{Y}_i$ - predicted values
Y - observed values

Errors in Regressions

Mean squared error (MSE)

- the MSE is a measure of the quality of an estimator
- it is always non-negative,
- values closer to zero are better

Errors in Regressions

Root Mean Squared Error (RMSE)

- It is an estimate of the standard deviation of the random component in the data

$$RMSE = \sqrt{MSE}$$

- value closer to 0 indicates a fit that is more useful for prediction

Errors in Regressions

Sum of Squares Error (SSE)

- This statistic measures the total deviation of the response values from the fit to the response values

$$SSE = \sum (y - y')^2$$

- A value closer to 0 indicates that the model has a smaller random error component, and that the fit will be more useful for prediction.

Errors in Regressions

Sum of Squares Total (SST)

- sum of squares about the mean

$$SST = \sum (y - \bar{y})^2$$

R-Square - *coefficient of determination*

- measures how successful the fit is in explaining the variation of the data.

$$R^2 = 1 - \frac{SSE}{SST}$$

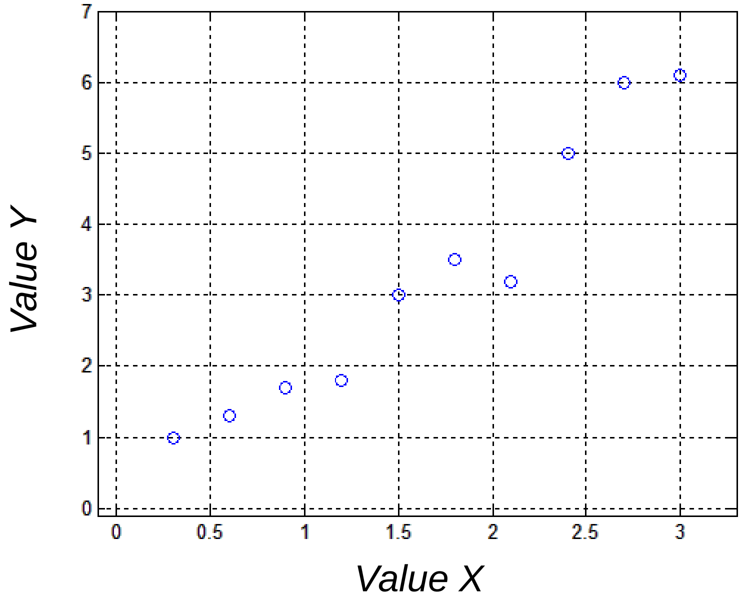
- R-square can take on any value between 0 and 1, with a value closer to 1 indicating that a greater proportion of variance is accounted for by the model.

- For example, an R-square value of 0.8234 means that the fit explains 82.34% of the total variation in the data about the average.

Example I

- Fit a first order function to the following data

Value X	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3
Value Y	1	1.3	1.7	1.8	3	3.5	3.2	5	6	6.1



Example I

- First the summation terms are calculated

$$n = 10$$

$$\sum_{i=1}^{n} x_i = 16.5$$

$$\sum_{i=1}^{n} x_i^2 = 34.65$$

$$\sum_{i=1}^{n} y_i = 32.6$$

$$\sum_{i=1}^{n} x_i y_i = 68.79$$

$$A = \begin{bmatrix} 10.0 & 16.5 \\ 16.5 & 34.5 \end{bmatrix}$$

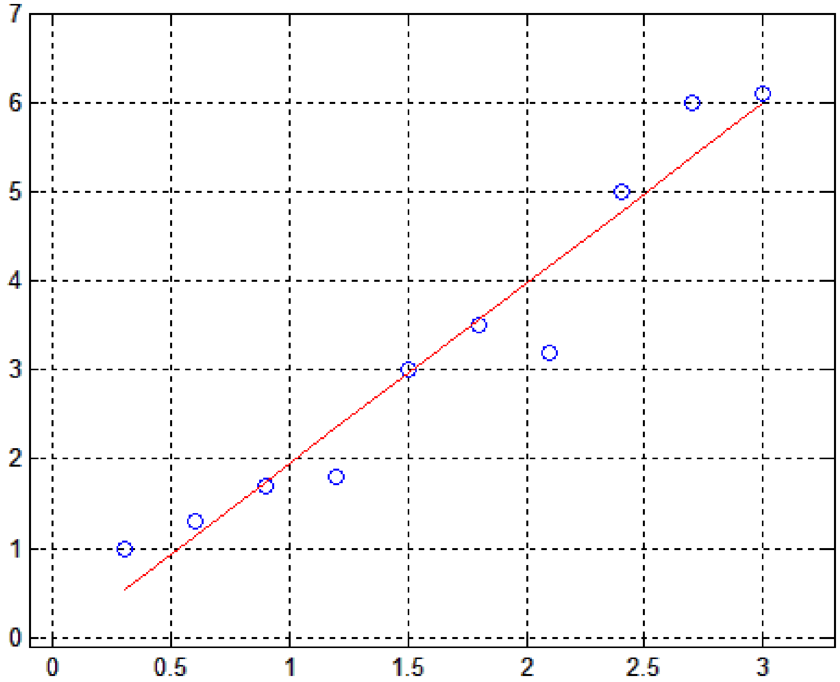
$$B = \begin{bmatrix} 32.6 \\ 68.79 \end{bmatrix}$$

Example I

- Finding the solutions

$$\begin{bmatrix} b \\ a \end{bmatrix} = inv \begin{bmatrix} 10.0 & 16.5 \\ 16.5 & 34.5 \end{bmatrix} \begin{bmatrix} 32.6 \\ 68.79 \end{bmatrix} \Rightarrow \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.0733 \\ 2.0202 \end{bmatrix}$$

$$f(x) = 2.0202\, x - 0.0733$$



Example I

```
clc;
clear all;
%% date intrare
x=[0.3 0.6 0.9 1.2 1.5 1.8 2.1 2.4 2.7 3];
y=[1 1.3 1.7 1.8 3 3.5 3.2 5 6 6.1];

plot(x,y,'ob')
grid on
axis([-0.1 3.3 -0.1 7])

%% determinare valori parametri matrici
n=numel(x)
sx=sum(x)
sx2=sum(x.^2)
sy=sum(y)
sxy=sum(x.*y)
```

Example I

```
%% definire matrici
A=[ n sx;
    sx sx2]
B=[sy;
   sxy]

%calcul parametri
X=inv(A)*B
estimare=X(2,1)*x+X(1,1)
hold on
plot(x,estimare,'-r')

% determinare erori
SSE=sum(y-estimare).^2

MSE=SSE/n
```

Multiple Linear Regression

- Obtain the derivatives with respect to the model parameters $\beta_0, \ldots, \beta_k$ and set them to zero

$$
\begin{aligned}
n\hat{\beta}_0 \quad &+\hat{\beta}_1 \sum_{i=1}^{n} x_{i1} &+\hat{\beta}_2 \sum_{i=1}^{n} x_{i2} &+\cdots &+\hat{\beta}_k \sum_{i=1}^{n} x_{ik} &= \sum_{i=1}^{n} y_i \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} &+\hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 &+\hat{\beta}_2 \sum_{i=1}^{n} x_{i1}x_{i2} &+\cdots &+\hat{\beta}_k \sum_{i=1}^{n} x_{i1}x_{ik} &= \sum_{i=1}^{n} x_{i1}y_i \\
\vdots \quad & \quad \vdots & \vdots & & \vdots & \quad \vdots \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} &+\hat{\beta}_1 \sum_{i=1}^{n} x_{ik}x_{i1} &+\hat{\beta}_2 \sum_{i=1}^{n} x_{ik}x_{i2} &+\cdots &+\hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 &= \sum_{i=1}^{n} x_{ik}y_i
\end{aligned}
$$

Multiple Linear Regression

- The equations presented before are formulated with the help of vectors and matrices

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

- the linear regression model can be written as follow

$$\mathbf{y} = \mathbf{X}\beta$$

- the regression coefficients are

$$\beta = X^{-1}y$$

Multiple Linear Regression

- multiple linear regression model; $k=2$ => predictor variables x_1, x_2 and a response y , can be written as follow:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

- Number of observations n=10

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} \qquad i = 1..10$$

- The error is

$$err = \sum_{i=1}^{10} e_i^2 = \sum_{i=1}^{10} (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2})^2$$

Multiple Linear Regression

- finding the minimum of the function (first derivative is zero)

$$\frac{\partial err}{\partial \beta_0} = n\beta_0 + \beta_1 \sum_{i=1}^{n} x_{i1} + \beta_2 \sum_{i=1}^{n} x_{i2} = \sum_{i=1}^{n} y_i$$

$$\frac{\partial err}{\partial \beta_1} = \beta_0 \sum_{i=1}^{n} x_{i1} + \beta_1 \sum_{i=1}^{n} {x_{i1}}^2 + \beta_2 \sum_{i=1}^{n} x_{i1}x_{i2} = \sum_{i=1}^{n} x_{i1}y_i$$

$$\frac{\partial err}{\partial \beta_2} = \beta_0 \sum_{i=1}^{n} x_{i2} + \beta_1 \sum_{i=1}^{n} x_{i2}x_{i1} + \beta_2 \sum_{i=1}^{n} x_{i2}^2 = \sum_{i=1}^{n} x_{i2}y_i$$

Multiple Linear Regression

$$X = \begin{bmatrix} n & \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i2} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} {x_{i1}}^2 & \sum_{i=1}^{n} x_{i1}x_{i2} \\ \sum_{i=1}^{n} x_{i2} & \sum_{i=1}^{n} x_{i2}x_{i1} & \sum_{i=1}^{n} x_{i2}^2 \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \qquad y = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_{i1}y_i \\ \sum_{i=1}^{n} x_{i2}y_i \end{bmatrix}$$

$$\beta = X^{-1}y$$

Multiple Linear Regression

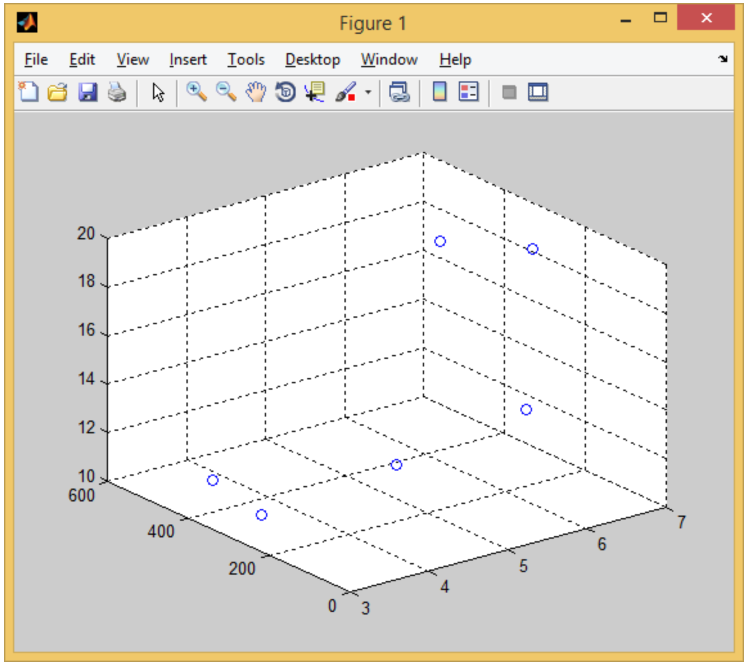
Example

A soft drink bottler is analyzing the vending machine serving routes in his distribution system. He is interested in predicting the time required by the distribution driver to service the vending machines in an outlet. This service activity includes stocking the machines with new beverage products and performing minor maintenance or housekeeping. It has been suggested that the two most important variables influencing delivery time (y in min) are the number of cases of product stocked (x_1) and the distance walked by the driver (x_2 in meters).

Cases (X1)	7	3	3	4	6	7
Distance (X2)	560	220	340	80	150	330
Time (Y)	16.48	11.5	12.03	13.75	13.75	18.11

Multiple Linear Regression

```
x1=[7 3 3 4 6 7];
x2=[560 220 340 80 150 330]
y=[16.48 11.5 12.03 13.75 13.75
18.11]


scatter3(x1,x2,y);
grid on
```



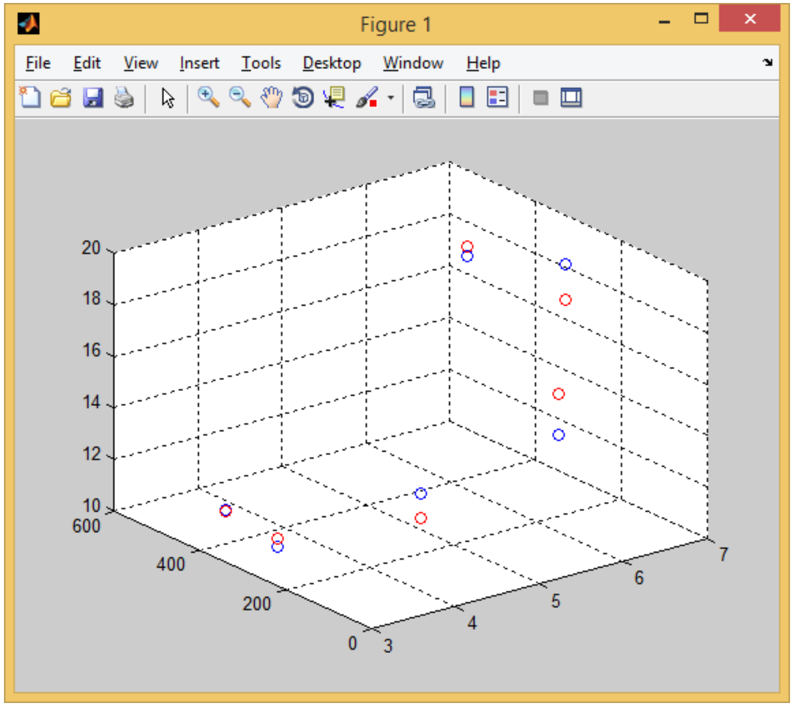
Multiple Linear Regression

```
%% determ param matrici
n=numel(x1);
sx1=sum(x1);
sx2=sum(x2);
sx1_2=sum(x1.^2);
sx1x2=sum(x1.*x2);
sx2_2=sum(x2.^2);

sy=sum(y);
sx1y=sum(x1.*y);
sx2y=sum(x2.*y);

X=[n sx1 sx2;
   sx1 sx1_2 sx1x2;
   sx2 sx1x2 sx2_2];

Y=[sy; sx1y; sx2y];
```

Multiple Linear Regression

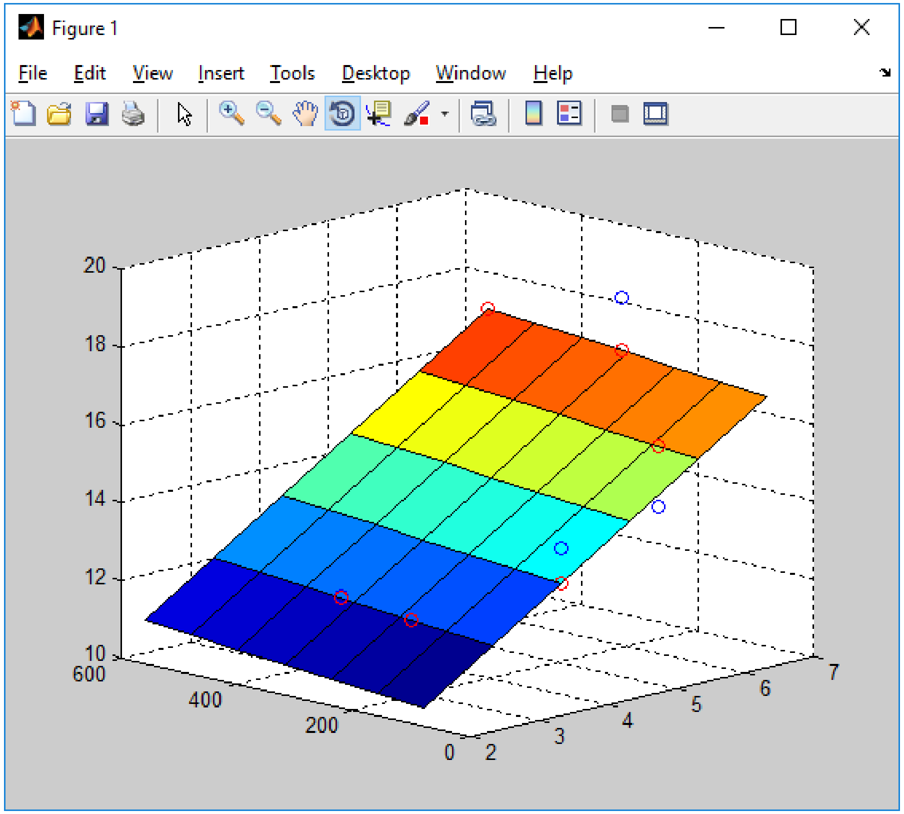
```
%% determinare parametri functie
B=inv(X)*Y

est_val=B(1)+x1.*B(2)+x2.*B(3)

hold on
scatter3(x1,x2,est_val);
```

B = 7.9428
    1.1960
    0.0014



Multiple Linear Regression

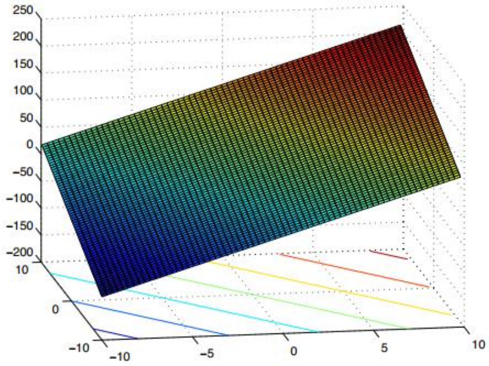
```
%% plot surface
tx1=2:1:7;
tx2=80:80:560
[X1,X2] = meshgrid(tx1,tx2);
f=B(1)+X1.*B(2)+X2.*B(3)


surf(X1,X2,f);
```

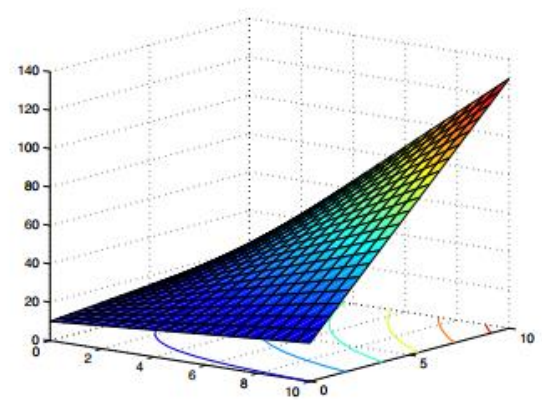


Multiple Linear Regression

$$y = 50 + 10x_1 + 7x_2$$

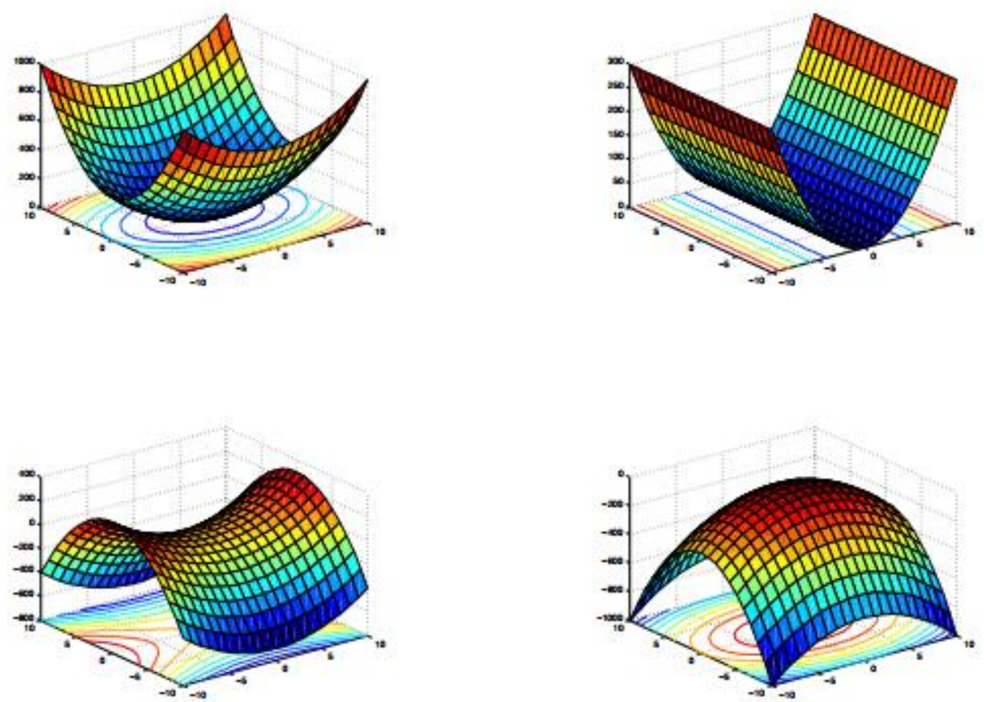
$$y = 10 + x_1 + x_2 + x_1x_2$$





Multiple Linear Regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2$$



Bibliography

1. Gogu, E, Birsan-Pipu, N. - Modele de regresie simpla si multipla - O abordare matriceala, Ed. Universitara, 2017

2. ARLINGHAUS, Sandra. Practical handbook of curve fitting. CRC press, 1994.

3. *SEMATECH e-Handbook of Statistical Methods*, Published by NIST, 2013