# System Identification for Sensing & Process

17-18 Dec 2008

*Prof. Dr Mohd Nasir Taib*

*Advanced Signal Processing (ASP) Research Group*

*Faculty of Electrical Engineering*

*Universiti Teknologi MARA Malaysia*

Email: dr.nasir@ieee.org

Web: http://www.asprg.net

# Content

Introduction
System Identification
Black Box Approach via ANN
Practical Guide - Sensor Modeling
Conclusion

# Introduction

WHAT IS THIS COURSE?

The course provides practical guide on System Identification technique

- Model optical sensor system
- Model of dynamic plant and controller design
- Model nonlinear process

# Introduction

WHAT IS IN THIS COURSE?

The course is divided into three (3) modules

- Black Box (using ANN) SysID
- GUI based SysID
- Command Prompt SysID

# Introduction

OUTCOMES

The course is designed so that participants could:

- Understand & appreciate System Identification
- Apply ANN, linear & nonlinear System Identification Techniques

# Introduction

## SCHEDULE – Day I

**DAY 1 (17-dec-08)**

08.30 am - Registration
09.00 am - Introduction
09.30 am - Black Box ANN System Identification
10.30 am - Tea break
11.00 am - Lab Session 1
12.30 pm - Lunch break

02.00 pm - ARX Modelling
03.30 pm - Lab Session 2
05.00 pm - Tea and break for the day

# Introduction

## SCHEDULE – Day II

**DAY 2 (18-dec-08)**

09.00 am - Continue with Lab Session 2
10.00 am - Tea break
10.30 am - System Identification for Process
12.30 pm - Lunch break

02.00 pm - Lab Session 3
04.00 pm - Closing
04.30 pm - Tea and end

# Introduction

- 1950's, much of control design relied on Bode, Nyquist or Ziegler-Nichols charts, or on step response analyses
- These techniques were limited to control design for single-input single-output (SISO) systems.

# Introduction

- Around 1960, Kalman introduced the state-space representation and laid the foundations for state-space based optimal filtering and optimal control theory, with Linear Quadratic optimal control as a cornerstone for model-based control design.
- This is the starting point for system identification

# Introduction

- System identification arose from the development of data-based techniques that would allow one to develop dynamical models for such diverse fields as process control, environmental systems, biological and biomedical systems, transportation systems, etc.

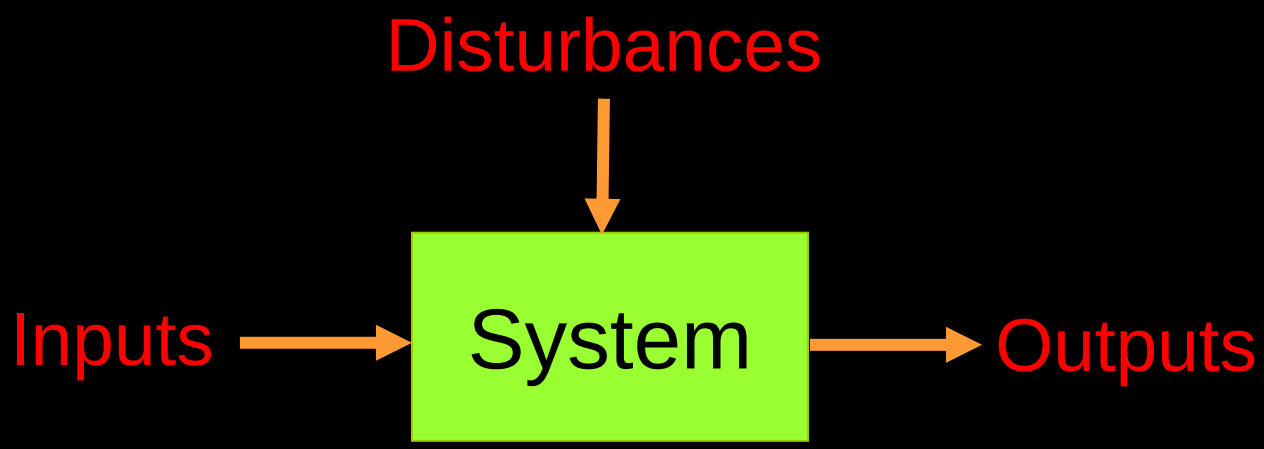
# System Identification

"Identification is the determination, on the basis of input and output, of a system within a specified class of systems, to which the system under test is equivalent."

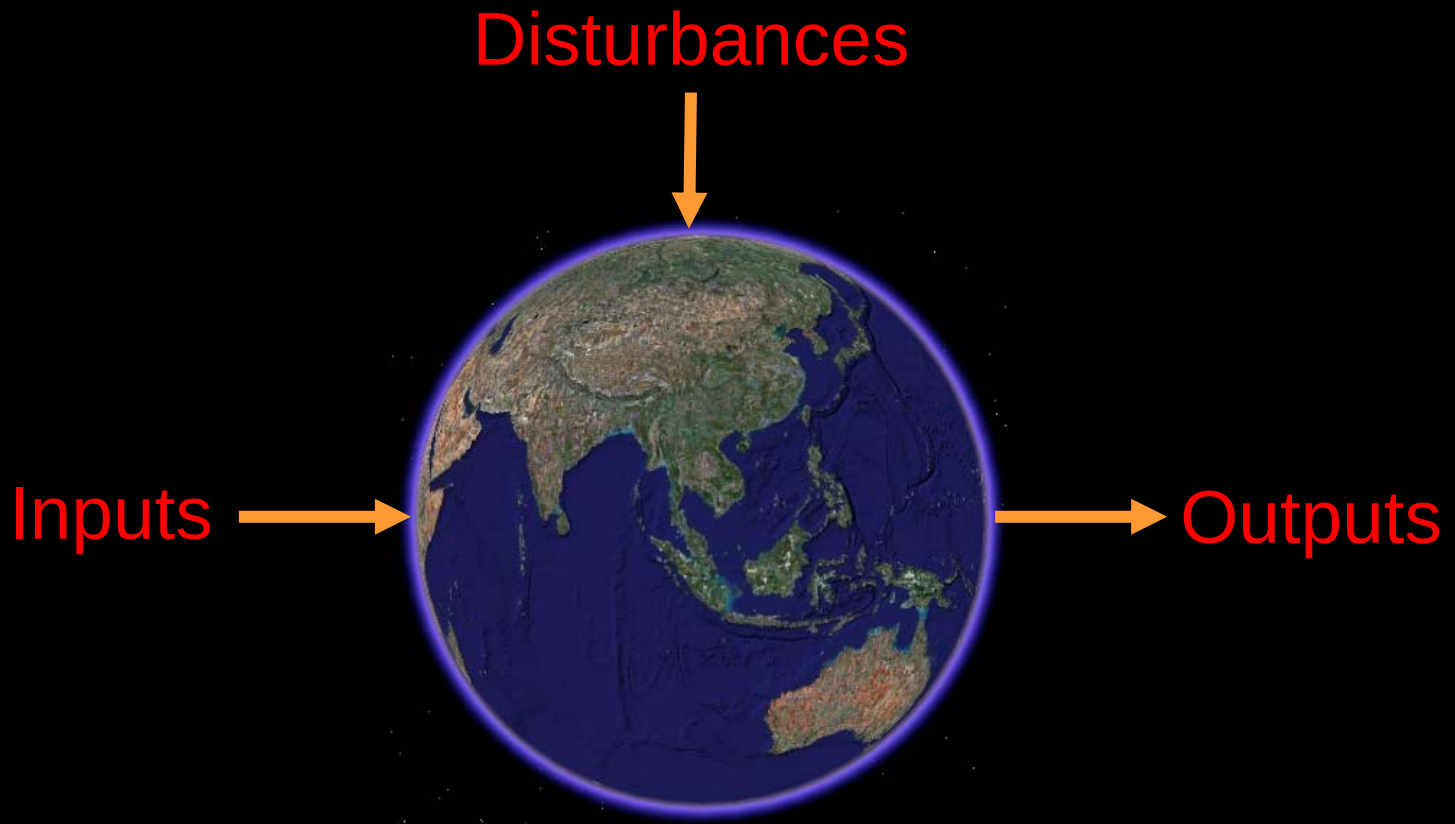
Zadeh (1962)

# System Identification

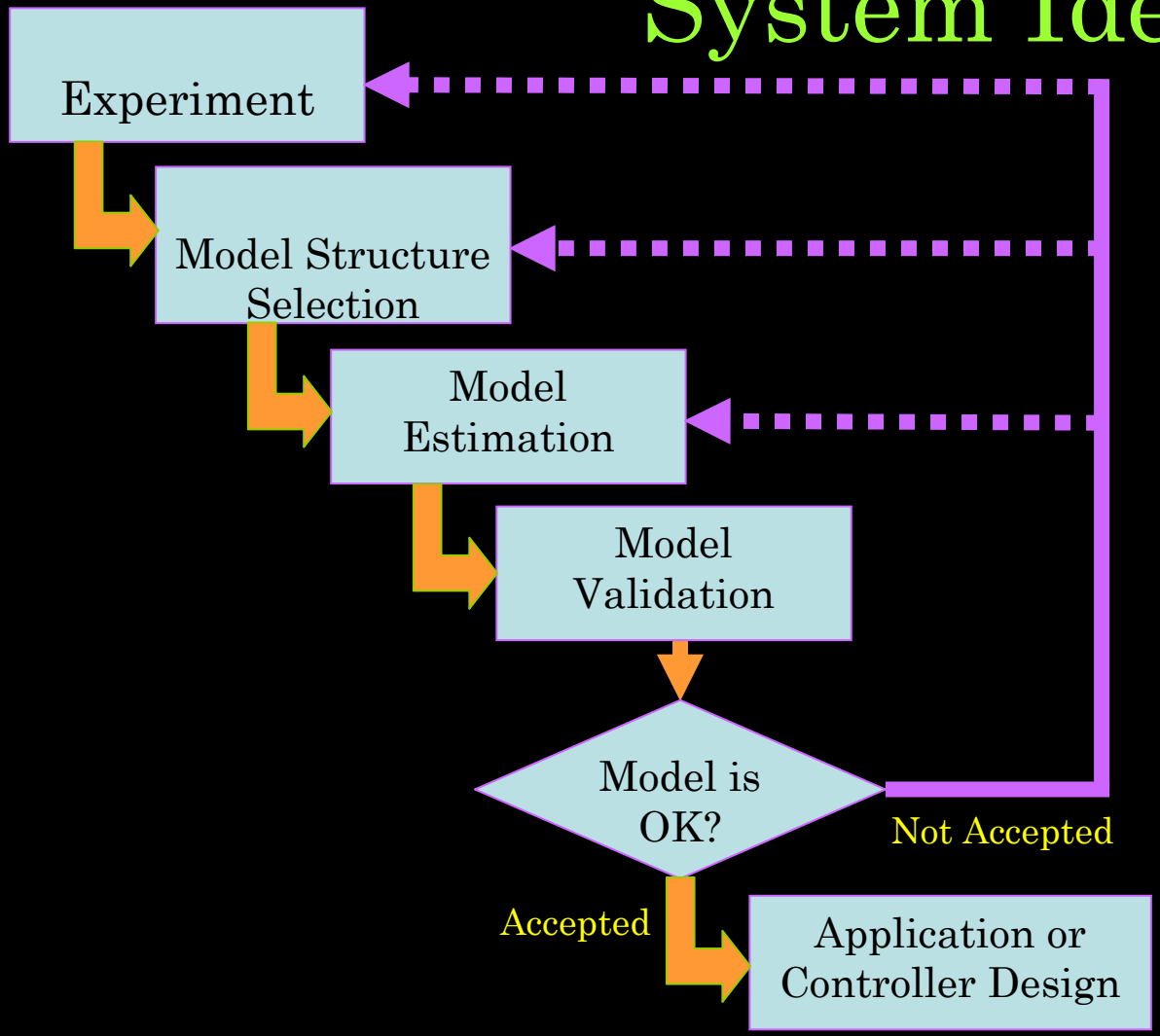
**Focuses on the modeling of dynamical systems from experimental data**

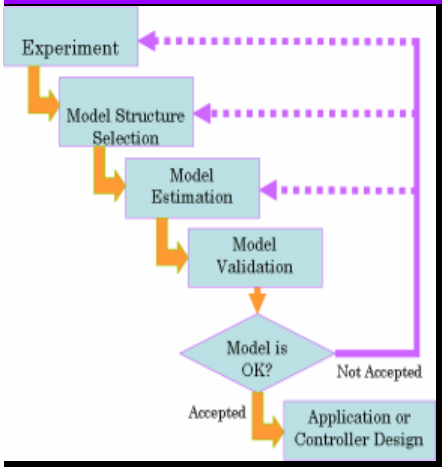


# System Identification

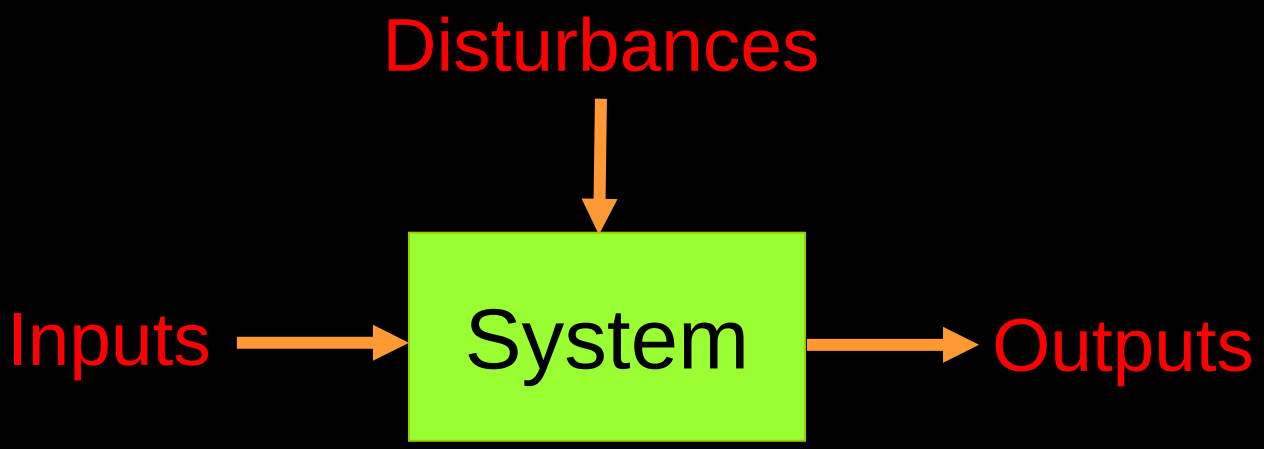


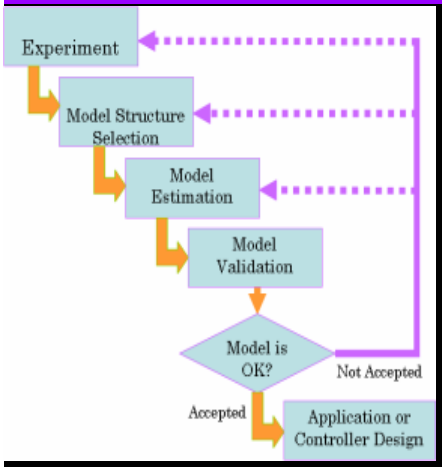
# System Identification



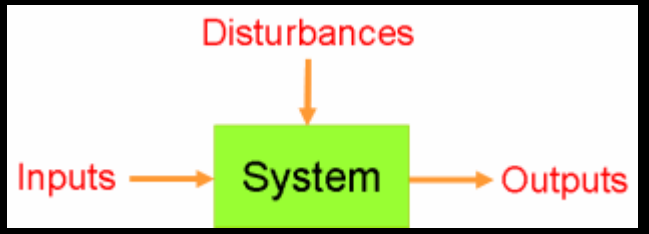



# Step 1: Experiment






# Step 1: Experiment




Step/Pulse Inputs

Gaussian White Noise

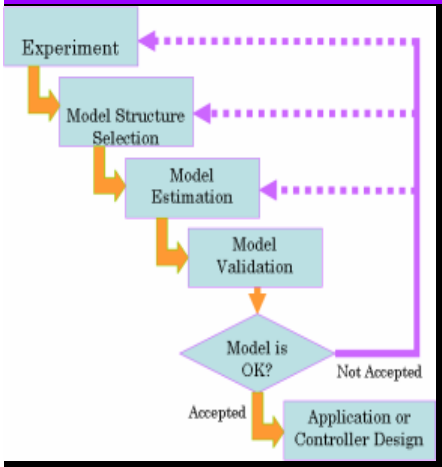
Random Binary Signal (RBS)

Pseudo-Random Binary Signal (PRBS)

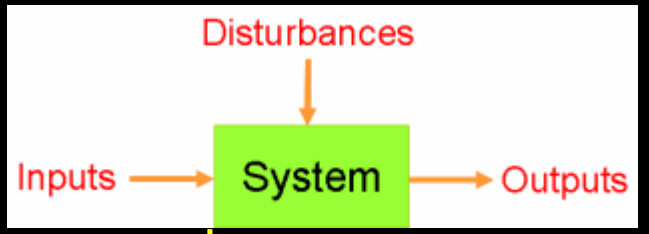
multi-level Pseudo-Random Signals

Multi-sine inputs

Hybrid, etc ...




# Step 1: Experiment

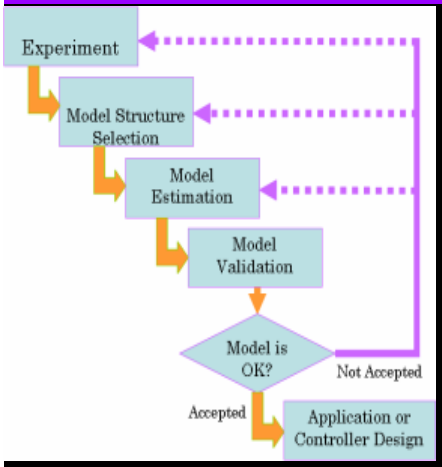



## Data Sampling

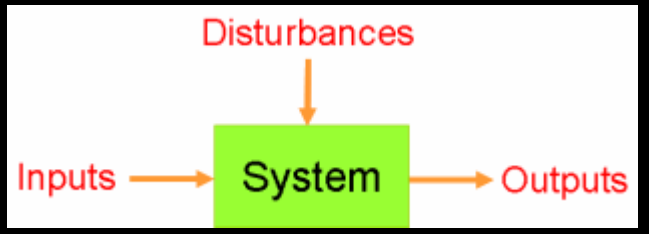
- rule of thumbs ~ 4 to 10 samples per time constant (rise time)

## Pre-process data

- filter noise
- Remove trend
- Remove outliers, *etc.*

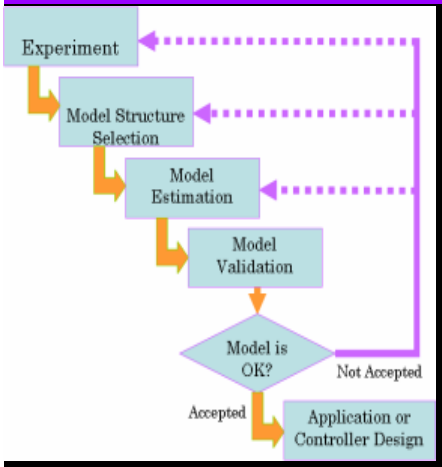



# Step 1: Experiment



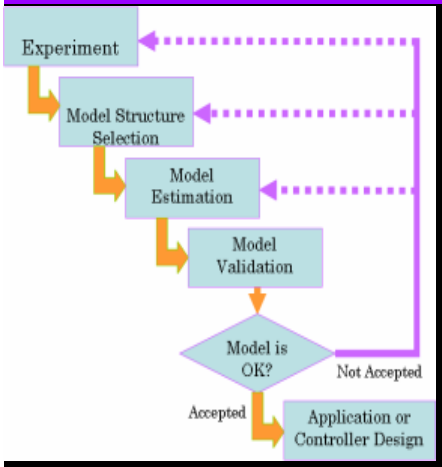

Split data into two parts

- one for estimation/training
- one for validation/testing



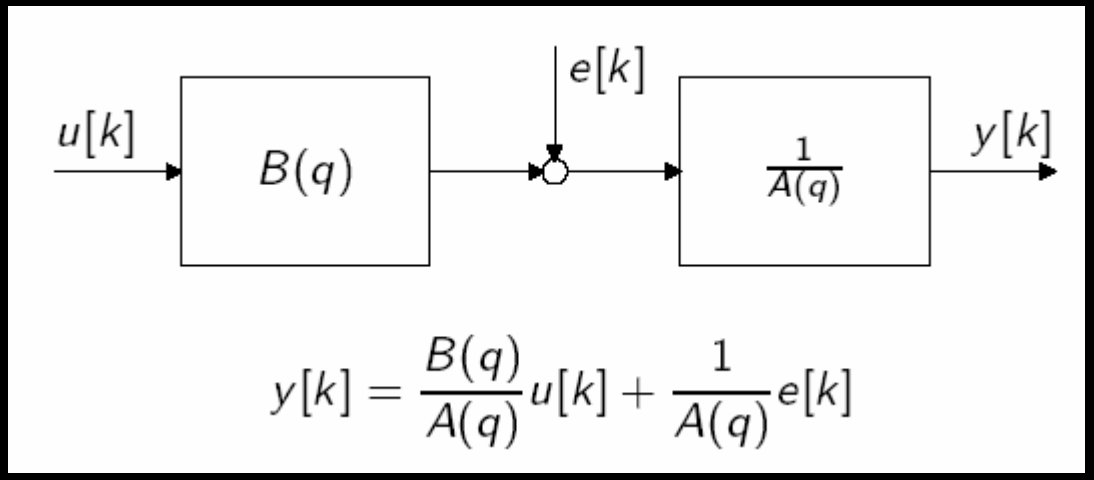

# Step 2: Model Structure Selection

- Linear – AR, ARX, ARMAX
- Nonlinear – NAR, NARX, NARMAX, ANN, Fuzzy, Hybrids, *etc.*

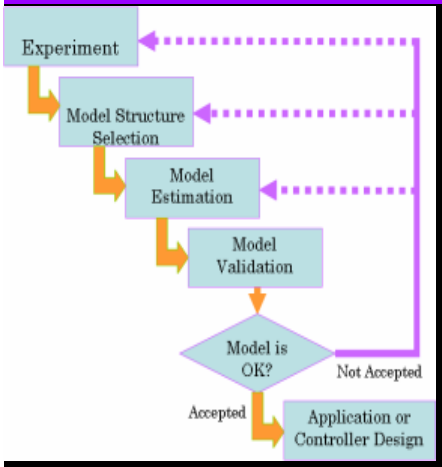



# Step 2: Model Structure Selection

- ARX Model

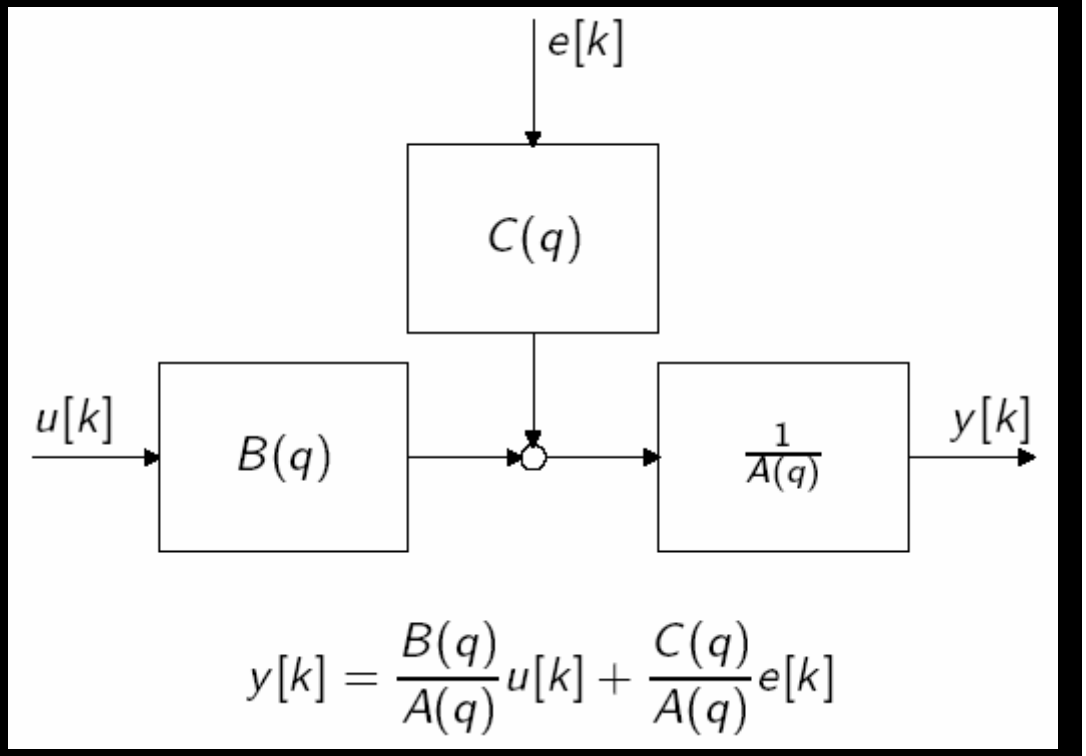



$$y[k] = \frac{B(q)}{A(q)}u[k] + \frac{1}{A(q)}e[k]$$

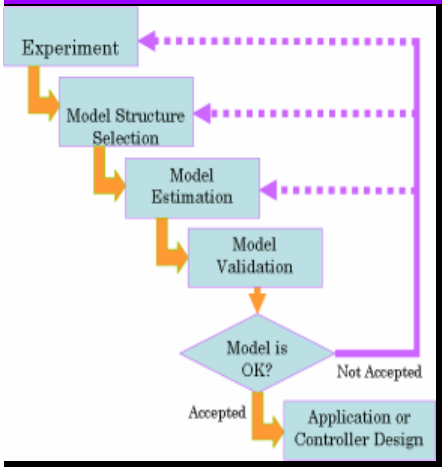



# Step 2: Model Structure Selection

- ARMAX Model

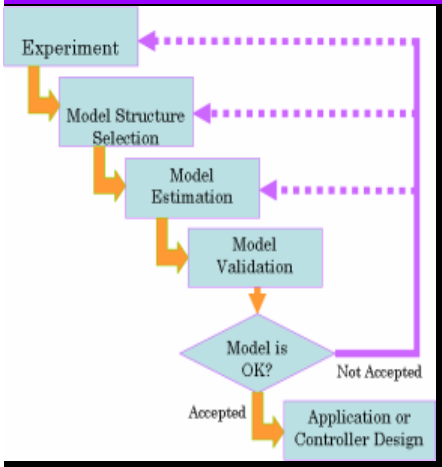



$$y[k] = \frac{B(q)}{A(q)}u[k] + \frac{C(q)}{A(q)}e[k]$$



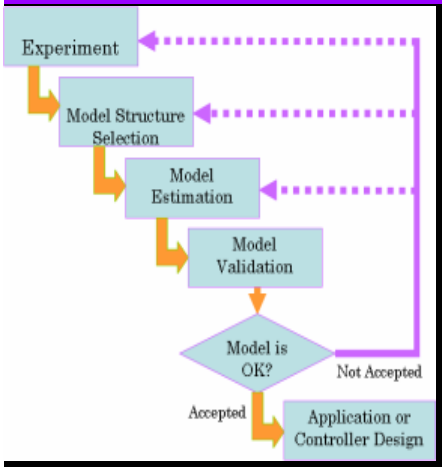

# Step 3: Model Estimation

- Non-parametric
  - Transient response
  - Correlation analysis
  - Frequency response analysis & Fourier analysis
  - Spectral analysis



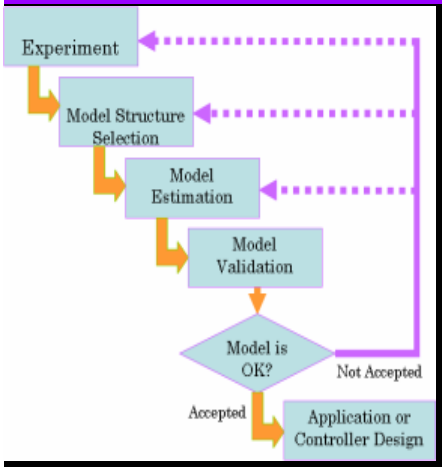

# Step 3: Model Estimation

- Parametric (Non-recursive/Batch methods)
  - Linear regression
  - Prediction error methods
  - Instrumental variable methods
  - etc



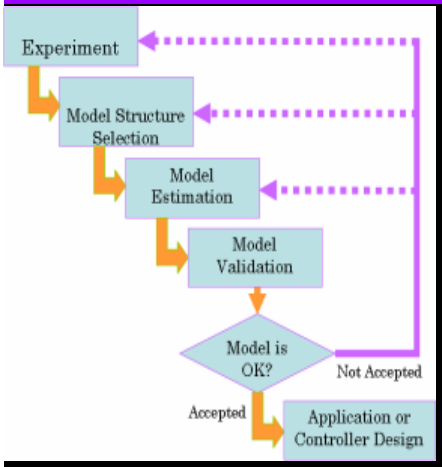

# Step 3: Model Estimation

- Parametric (Recursive methods)
  - Recursive Least Squares (RLS)
  - Recursive prediction error methods
  - Recursive instrumental variable methods
  - Forgetting factor techniques and time-varying systems
  - etc



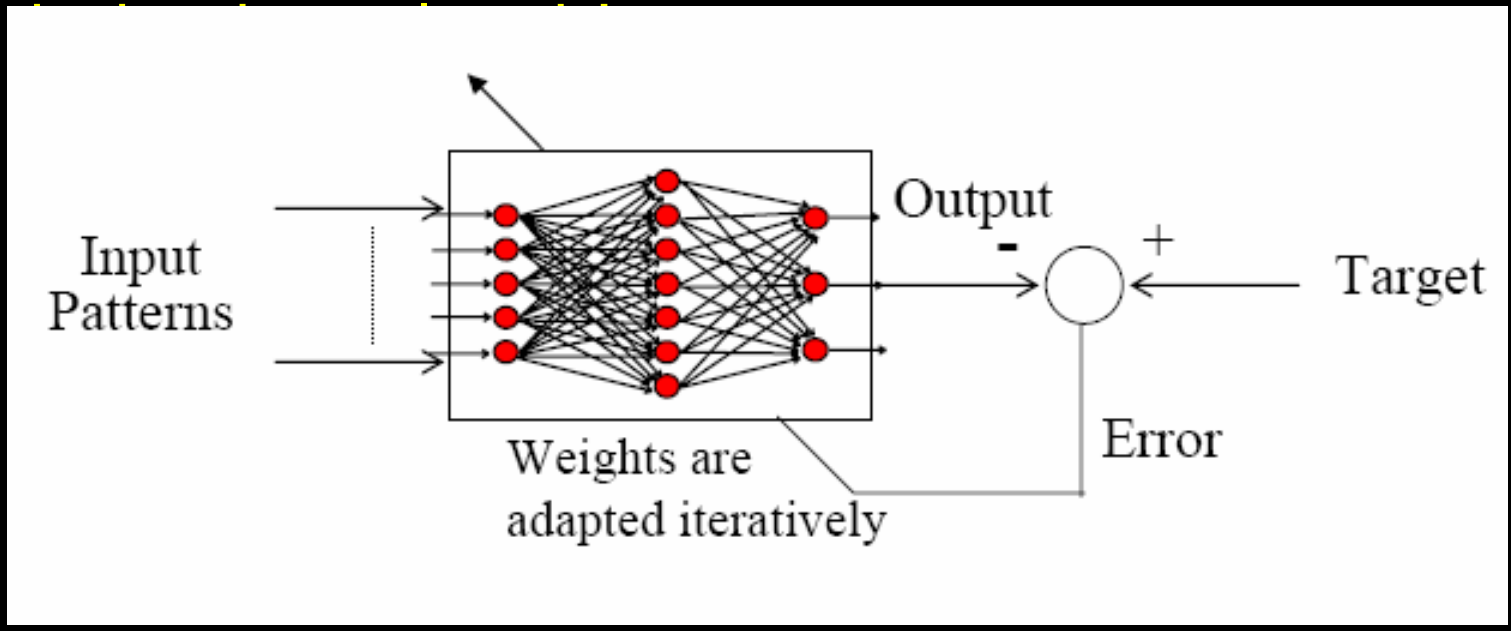

# Step 3: Model Estimation

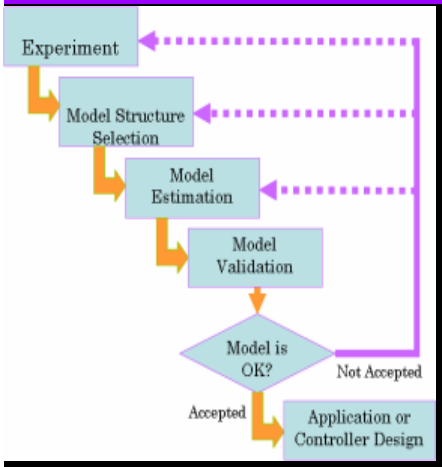
- ANN, Fuzzy, Hybrids
  - Optimisation algorithms




# Step 3: Model Estimation

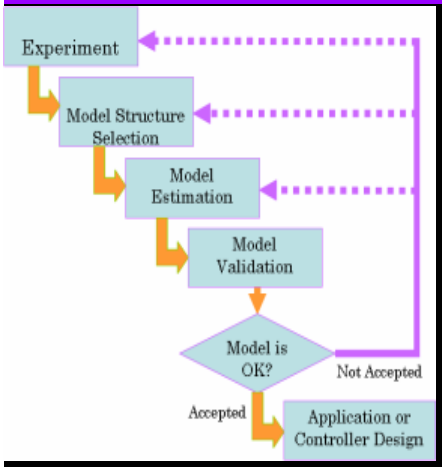
- ANN, Fuzzy, Hybrids
  - Op






# Step 4: Model Validation

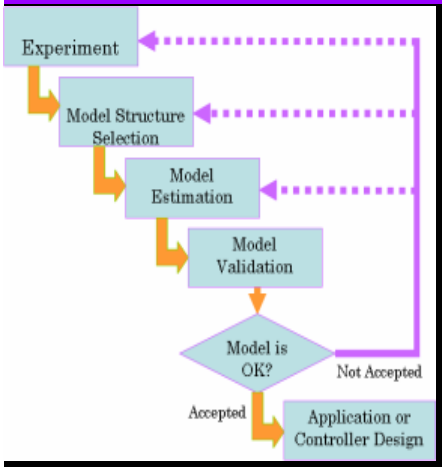
- Check for model 'goodness'
- Compare model simulation/prediction with test data
- Perform statistical tests on prediction errors
- Compare estimated model's frequency response and spectral
- analysis result in frequency domain




# Step 4: Model Validation

Residuals (errors)

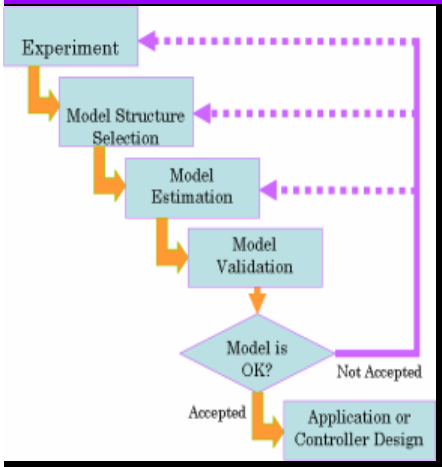
- If the model is correct the residuals should be
  - white
  - uncorrelated with input(s)




# Step 4: Model Validation

Residuals (errors)

- Observe 95% confidence band of auto/cross correlations
  - Components outside bands indicate unmodelled dynamics




# Step 5: Application

- Apply model for on-line/real time application
- Design controller

# System Identification

Best fit line Technique

Linear model Poly1:

f(x) = p1*x + p2

Coefficients (with 95% confidence bounds):

p1 = 0.9905 (0.9695, 1.011)

p2 = 0.5408 (0.4193, 0.6622)

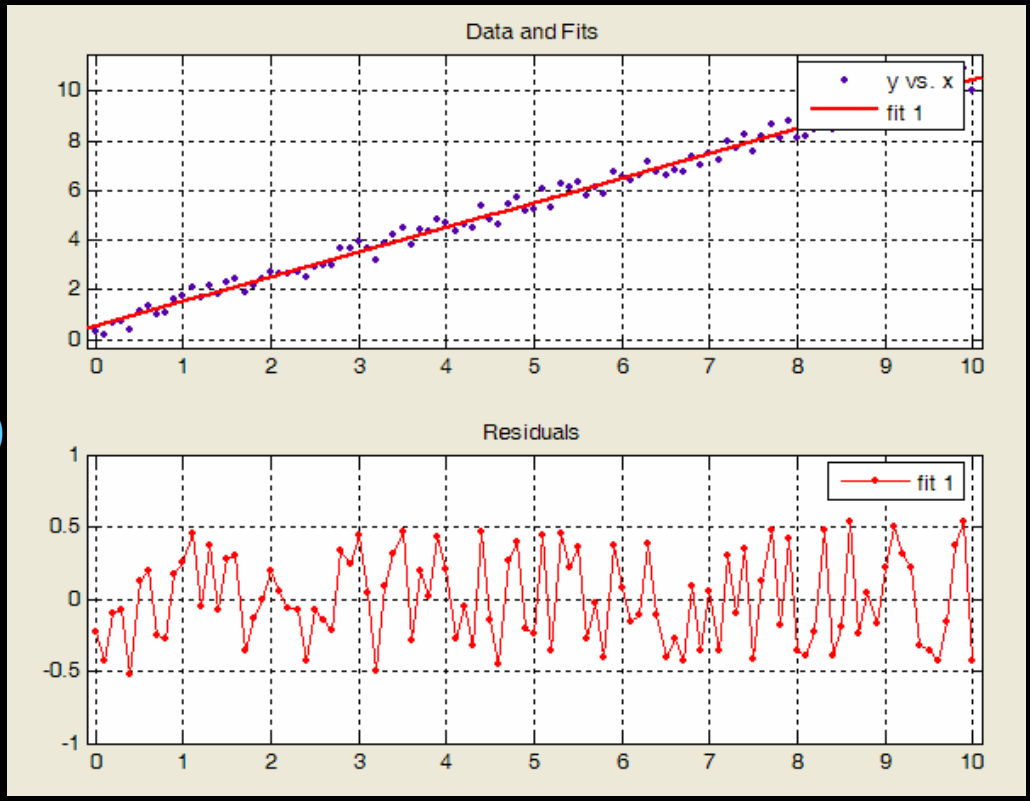
Goodness of fit:

SSE: 9.502

R-square: 0.9888

Adjusted R-square: 0.9887

RMSE: 0.3098



# Black Box Approach via ANN

# Black Box Model

Let $Z^t$ denote all available (input-output) data up to time $t$. A mathematical model for the system is a function from these data to the space where the output at time $t$, $y(t)$ lives, in general

$$\hat{y}(t|t-1) = g(Z^{t-1}, t)$$

The function can be thought of as a predictor of the next output.
A parametric model structure is a parameterized family of such models:

$$g(Z^{t-1}, \theta)$$

# Black Box Model

The general mapping $g(Z^{t-1}, \theta)$ is normally too flexible. Let us split it into one mapping from $Z^{t-1}$ to a regression vector $\varphi(t)$ of fixed dimension $d$ and a mapping $g$ from $R^d$ to $R$ (assuming the output to be scalar):

$$g(Z^{t-1}, \theta) = g(\varphi(t), \theta)$$

$$\varphi(t) = \varphi(Z^{t-1}) \quad (\text{ or } \varphi(t, \theta) = \varphi(Z^{t-1}, \theta))$$

Leaves two problems

1. Choose the mapping $g(\varphi, \theta)$
2. Choose the regression vector $\varphi(t)$

# Black Box Model

First, consider $\varphi$ to be scalar. Basic form

$$g(\varphi, \theta) = \sum_{k=1}^{N} \alpha_k \kappa(\beta_k(\varphi - \gamma_k))$$

- $\kappa(x) = \cos(x)$: Fourier transform
- $\kappa(x) = U(x)$: Unit pulse, gives piecewise constant functions $g$.
  - Soft version: $\kappa(x) = e^{-x^2/2}$
- $\kappa(x) = H(x)$: Step at $x = 0$, gives also piecewise constant functions
  - Soft version: $\kappa(x) = \frac{1}{1+e^{-x}}$
- $\alpha$ coordinates, $\beta$ scale or dilation, $\gamma$ location

# Black Box Model

- ANN: artificial Neural Networks
  - One hidden layer sigmoidal: $\kappa(x) = \frac{1}{1+e^{-x}}$, ridge extension
  - Radial Basis Networks: $\kappa(x) = e^{-x^2/2}$, radial extension
- Wavelets: $\kappa$ is the "mother wavelet" and $\beta_j = 2^j$, $\gamma_k = 2^{-j}k$ (double indexing) as fixed choices
- (Neuro)-Fuzzy models: $\kappa$ are the membership functions

# Black Box Model

- Models without any reference to the physical background (no *apriori* information).
- The model parameters are basically used to fit the model behaviour to the measured system data - often impossible to associate them with physical quantities of the system.

# ANN

- System Identification using ANN started in 1990's
- ANN models belong to a set of 'model' classes that can be linear or nonlinear

# ANN

- An easy and simple but superior signal processing technique can be accomplished using an artificial neural network (ANN).
- ANN has been popularly employed as a novel technique for non-linear system modelling, pattern recognition, forecasting and process control purposes.
- It is a powerful tool particularly suited to the analysis of non-linear, multivariate data. Based on the biological neural network, ANN can be adapted to a given system through training or teaching cycles with a known/measured system's input-output information

# ANN

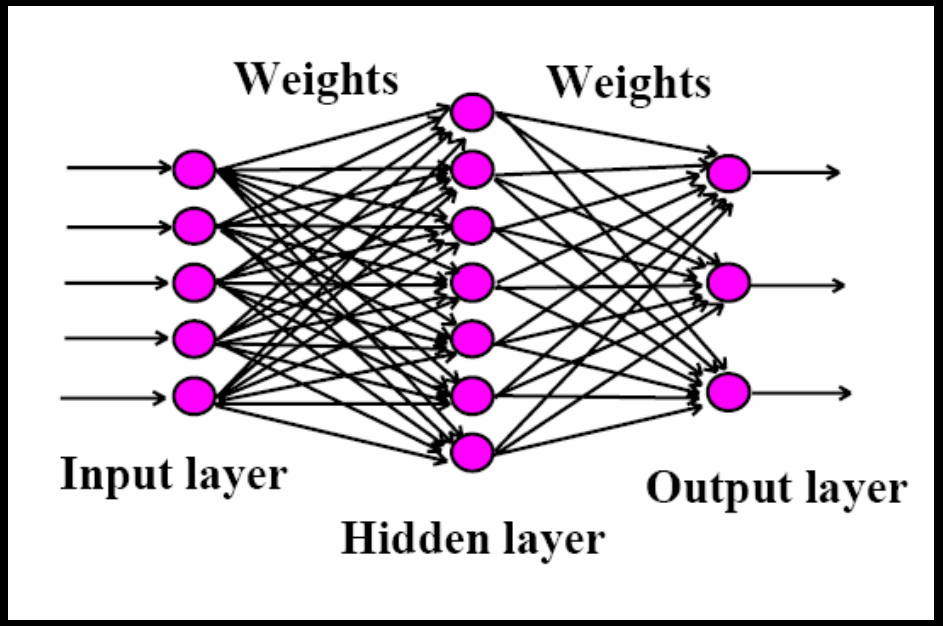
- ANN can be trained to capture and generalise the non-linear mechanism underlying the sensor's input-output relationship, thus memorising the behaviour of the sensor's response after which it will be able to provide an accurate prediction during recall process.
- The training is the hardest part in employing ANN, since this may require a lot of computational power and time. Training an ANN is a process akin to that of an equation building and model fitting process in a conventional system modelling, except that for ANN there is no need to explicitly define the model form or structure.

# ANN

- There are many training algorithms
- There are many different network architectures.
- Once trained, the practical application of ANN is straightforward and fast. This application is realised by a one-step process of forward passing the sensor's measurements through the stored weight matrix.
- The ANN can be implemented using a computer system or solid-state instrumentation embedded with a microprocessor system.

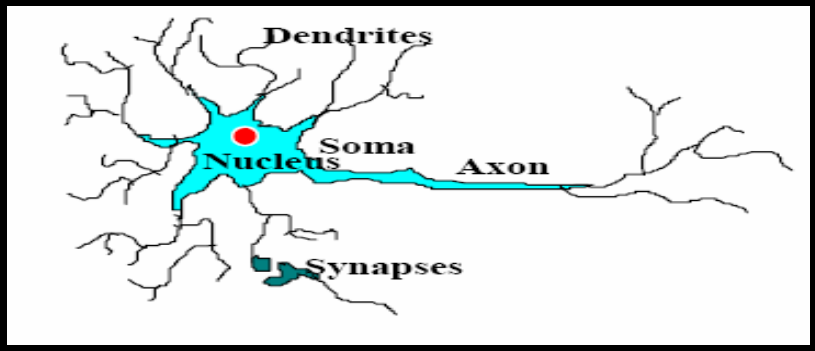
# ANN

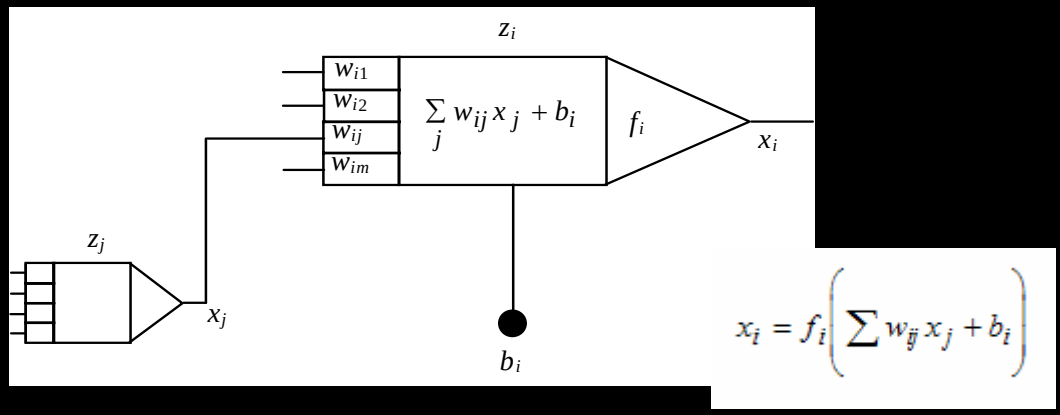
- ANN is a highly parallel processing system
- Consists of interconnected simple processing units called neurons (nodes)



# ANN

- For each interconnection, there is an associated weight (adjustable gains)
- Neurons are activated by a certain (activation) function.



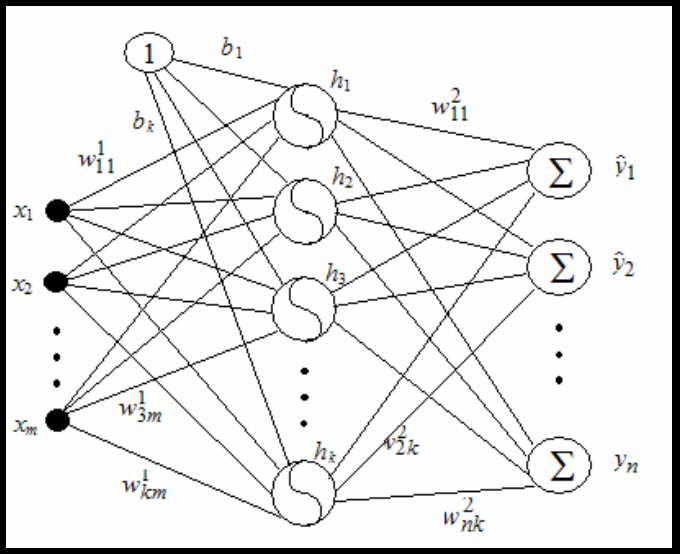



# ANN

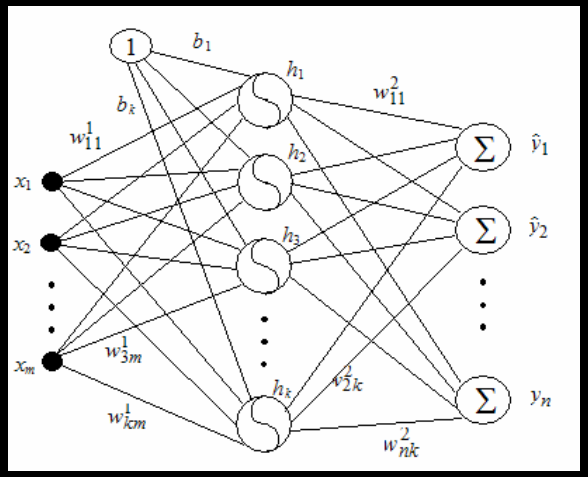
- Signals are processed from input to output by multiplying the weights and activated signals

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix} = f\left( \begin{bmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{1m}^1 \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2m}^1 \\ \vdots & \vdots & & \vdots \\ w_{k1}^1 & w_{k2}^1 & \cdots & w_{km}^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} \right)$$

$$h_j = f\left( \sum_{i=1}^{m} w_{ji}^1 x_i + b_j \right)$$



# ANN




$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} w^2_{11} & w^2_{12} & \cdots & w^2_{1k} \\ w^2_{21} & w^2_{22} & \cdots & w^2_{2k} \\ \vdots & \vdots & & \vdots \\ w^2_{n1} & w^2_{n2} & \cdots & w^2_{nk} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix}$$

$$\hat{y}_l = \sum_{j=1}^{k} w^2_{lj} \; f\left( \sum_{i=1}^{m} w^1_{ji} x_i + b_j \right)$$

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix} = f\left( \begin{bmatrix} w^1_{11} & w^1_{12} & \cdots & w^1_{1m} \\ w^1_{21} & w^1_{22} & \cdots & w^1_{2m} \\ \vdots & \vdots & & \vdots \\ w^1_{k1} & w^1_{k2} & \cdots & w^1_{km} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} \right)$$

$$h_j = f\left( \sum_{i=1}^{m} w^1_{ji} x_i + b_j \right)$$

# ANN

- A learning rule is needed to adapt the weights to solve a particular problem
- Training patterns are needed in order to train the ANN.

$$\Delta w_{ij}^{p}(t) = \eta_w \delta_i^{p}(t) x_j^{p-1}(t) + \alpha_w \Delta w_{ij}^{p}(t-1)$$

$$\Delta b_i^{p}(t) = \eta_b \delta_i^{p}(t) + \alpha_b \Delta b_i^{p}(t-1)$$

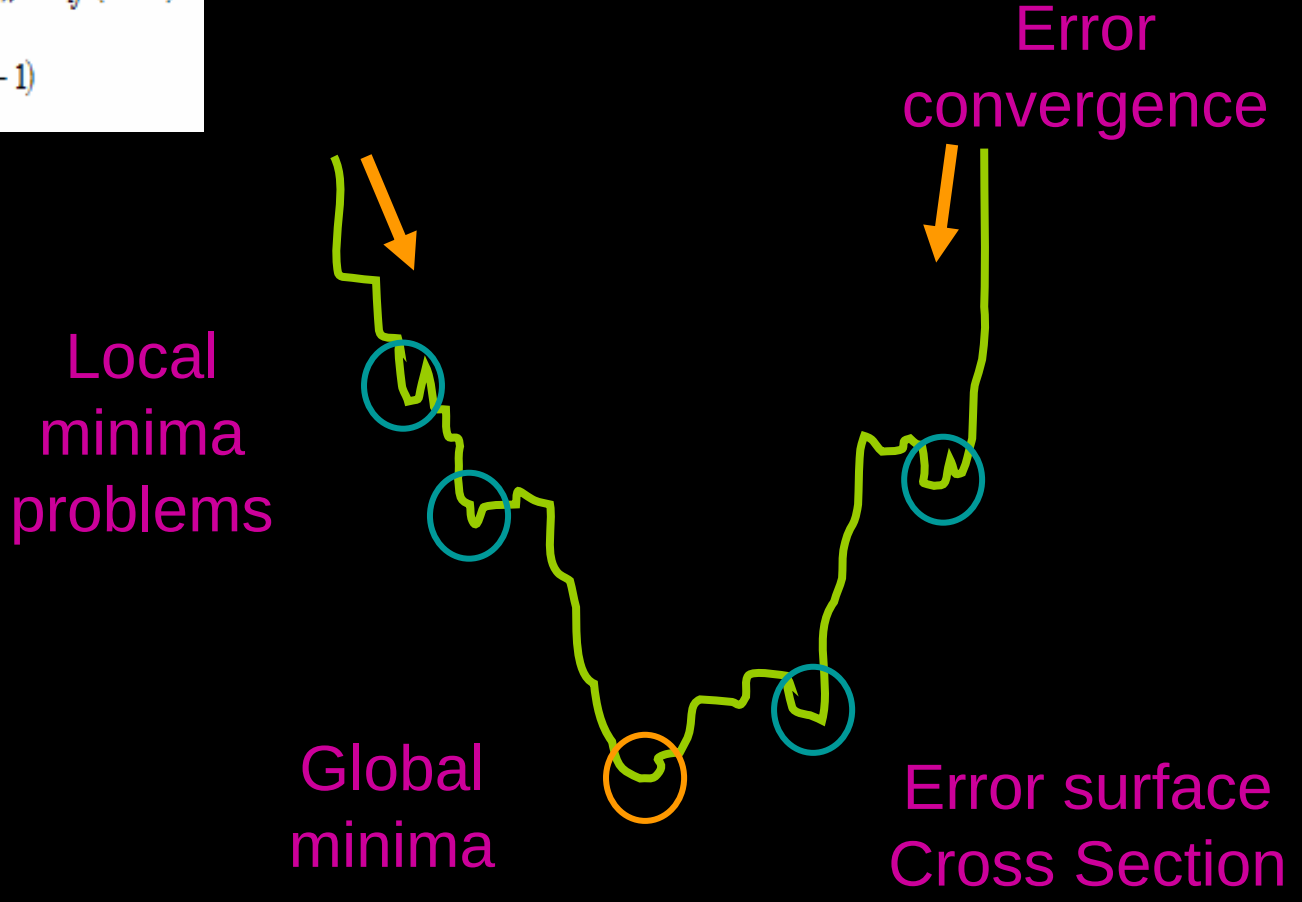
$$J = \frac{1}{2}\sum_i \left(y_i - \hat{y}_i\right)^2$$

# ANN

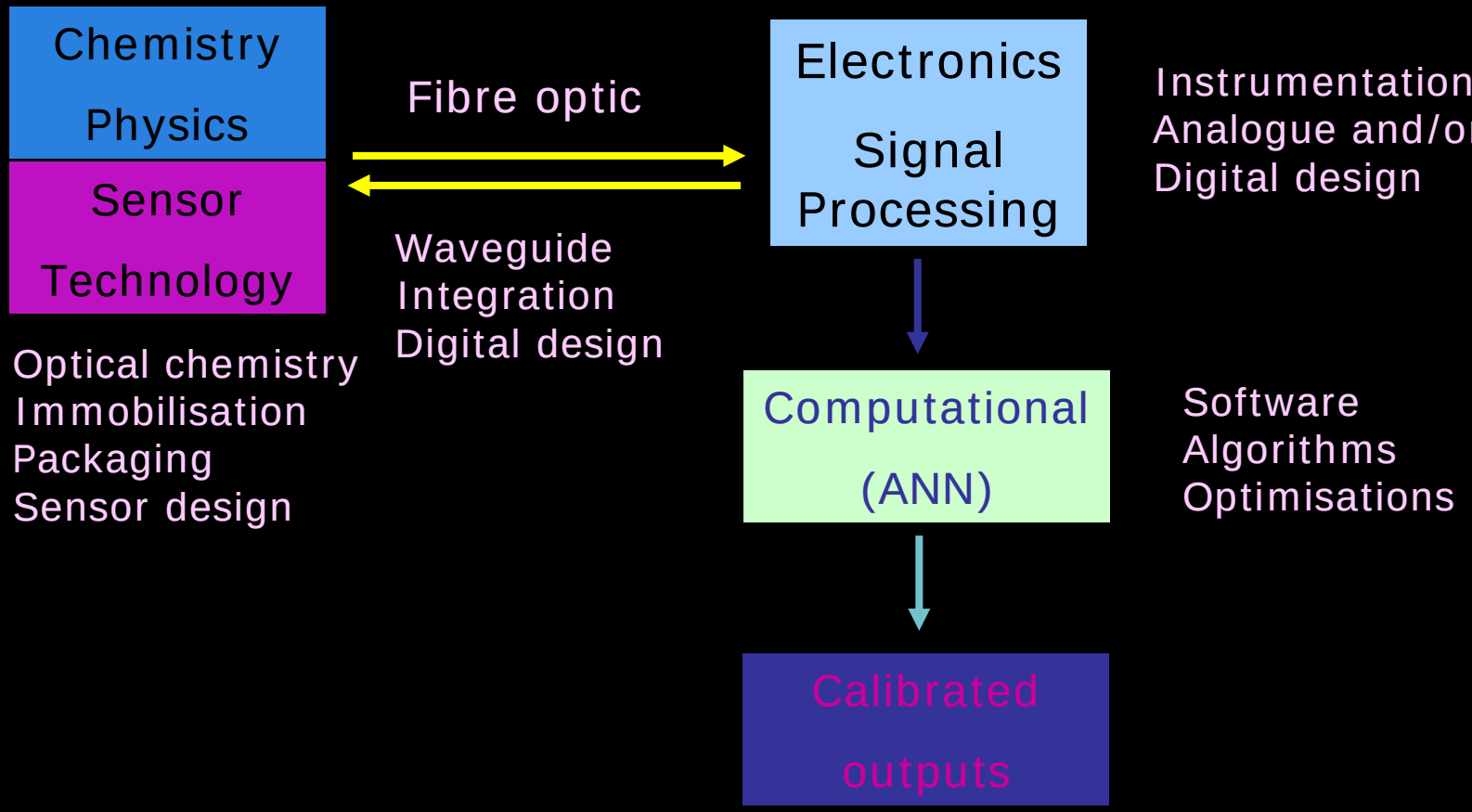
$$\Delta w_{ij}^{p}(t) = \eta_w \delta_i^{p}(t) x_j^{p-1}(t) + \alpha_w \Delta w_{ij}^{p}(t-1)$$

$$\Delta b_i^{p}(t) = \eta_b \delta_i^{p}(t) + \alpha_b \Delta b_i^{p}(t-1)$$

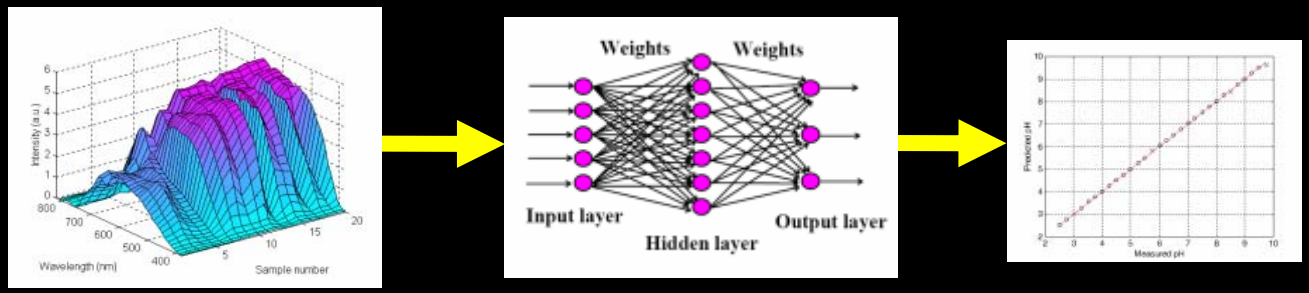
$$J = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2$$



# Application of ANN for OFCS



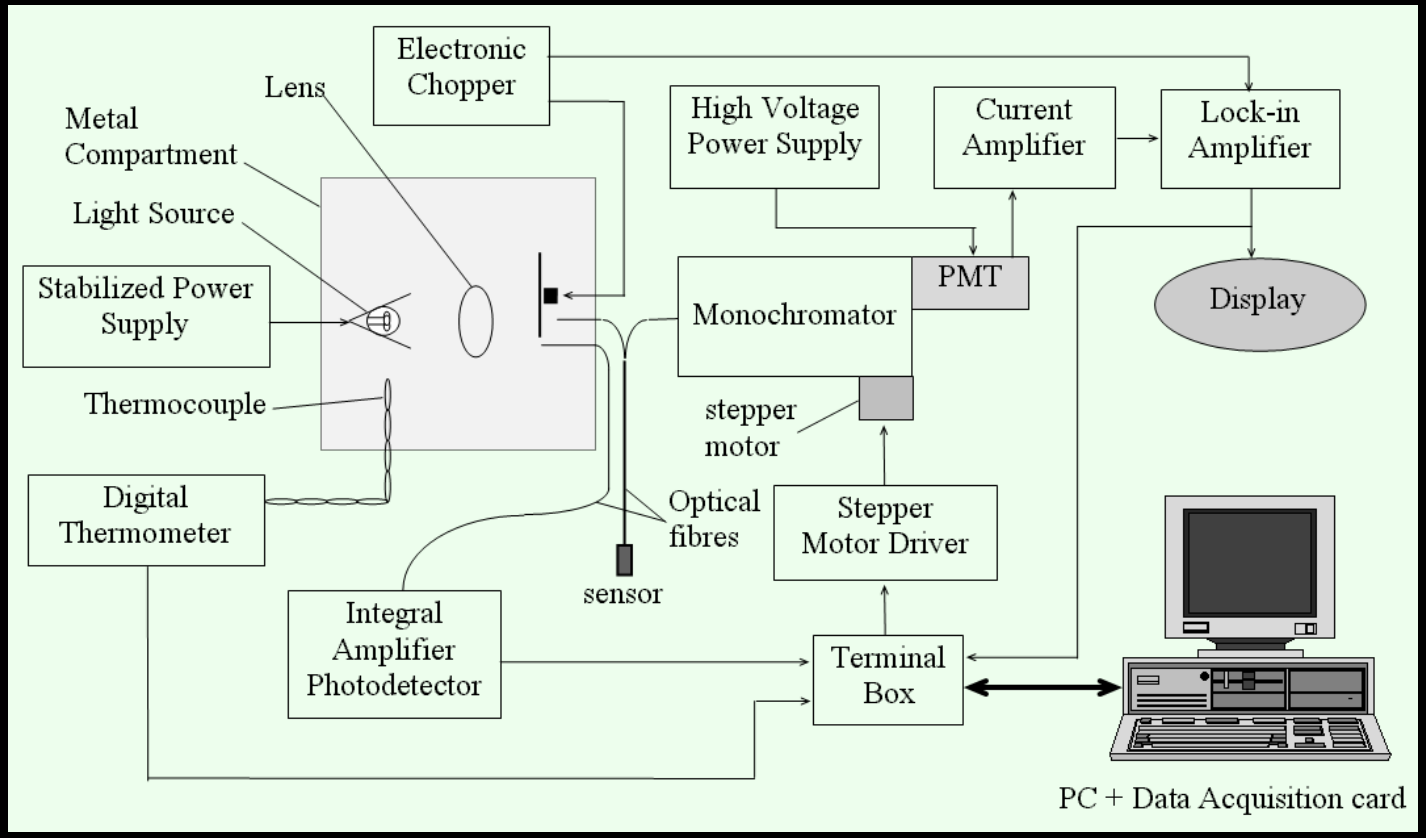
# Application of ANN for OFCS



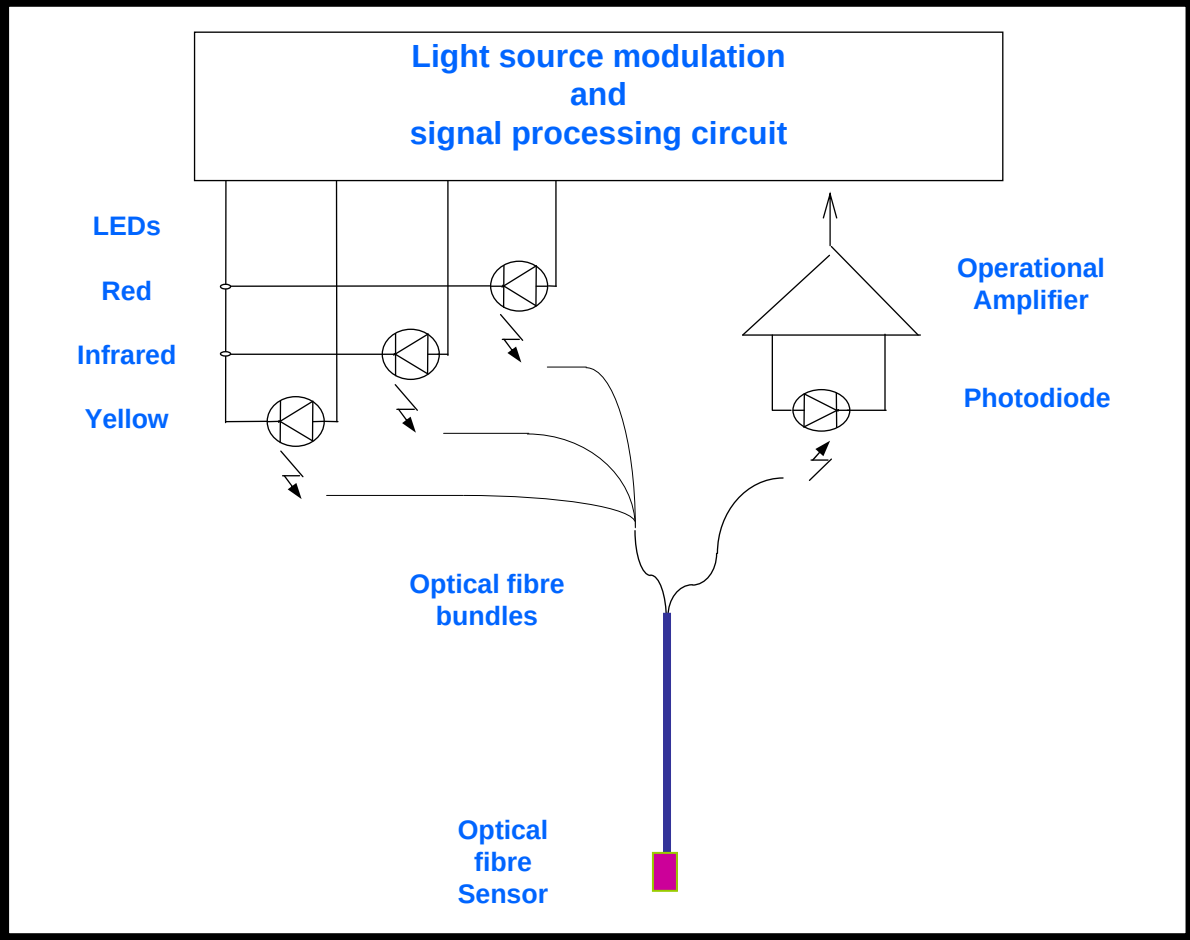
# Optical Sensor

- Works based on changes of light properties such as reflection, dispersion, scattering, interference, absorption, refraction and diffraction
- Optical system measures one or more of: wavelength, amplitude, phase, polarization or color

# Optical Sensor - Measurement

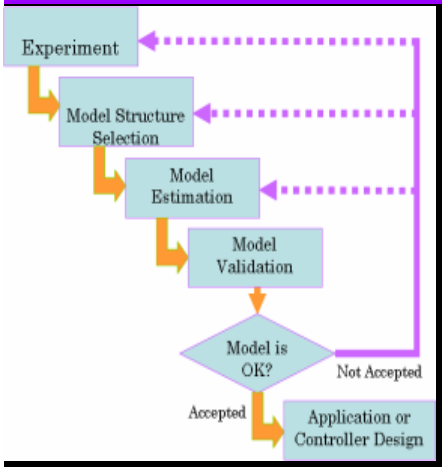


# Optical Sensor - Measurement



# Practical Guide: Sensor Modeling

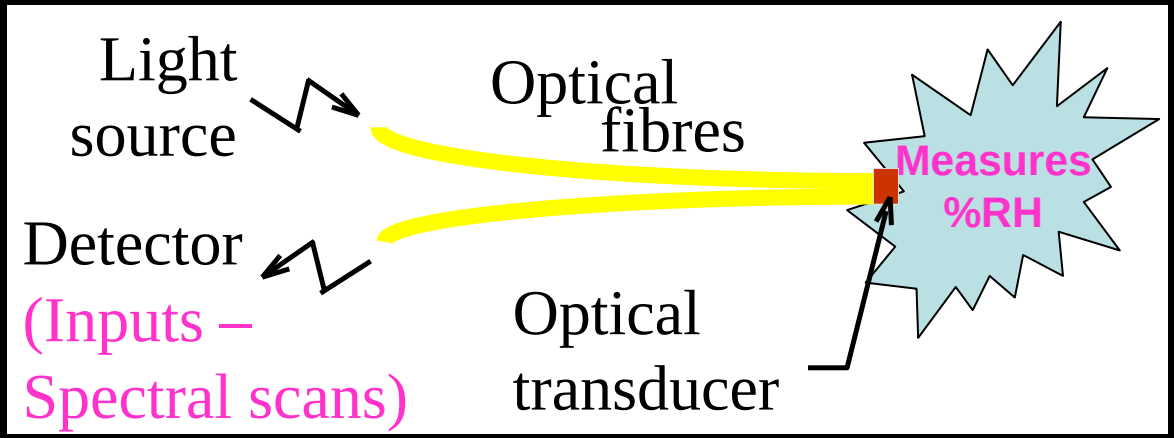
Apply Black Box System Identification for modelling Optical %RH Sensor

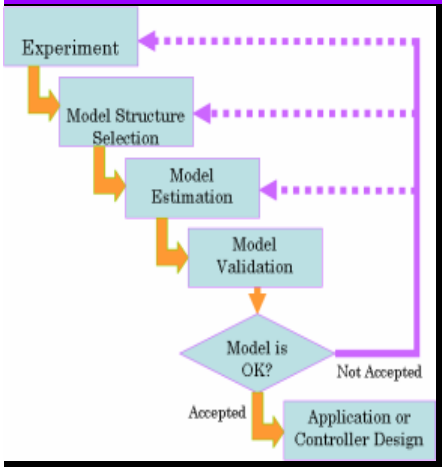



# Practical Guide: Sensor Modeling

## Experiment

### Gather input-output data

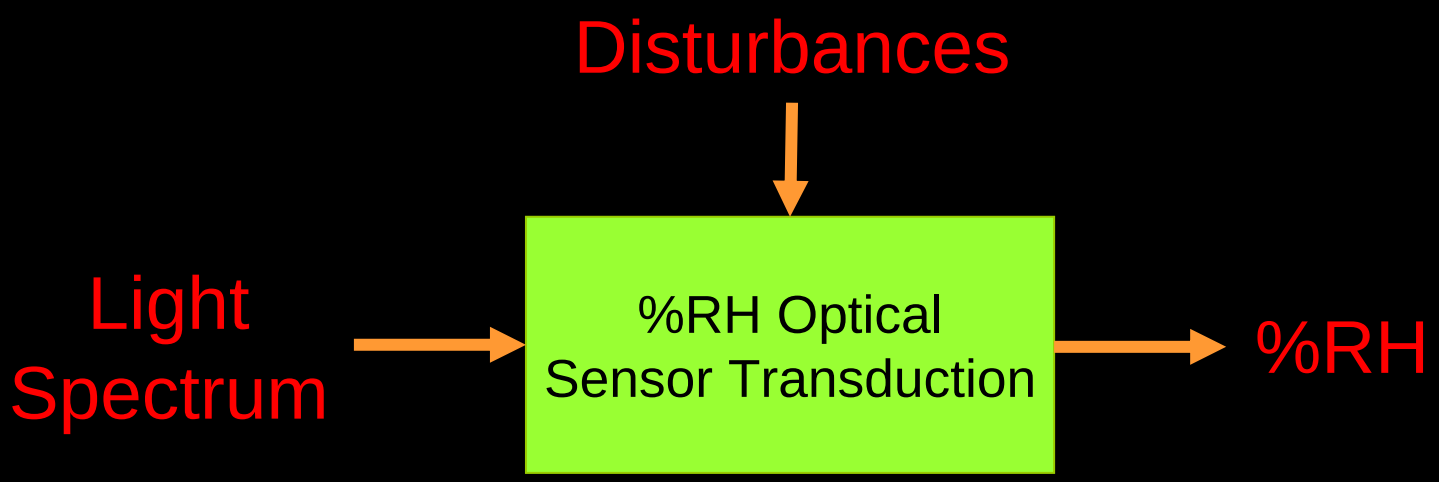


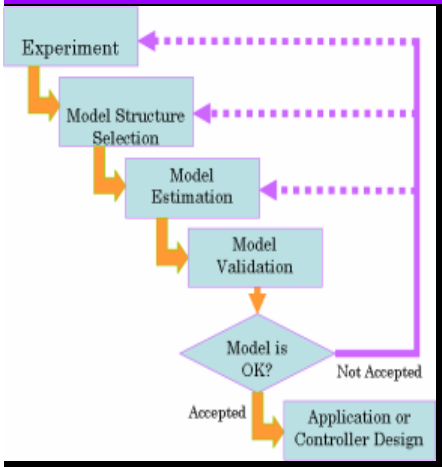



# Practical Guide: Sensor Modeling

## Experiment

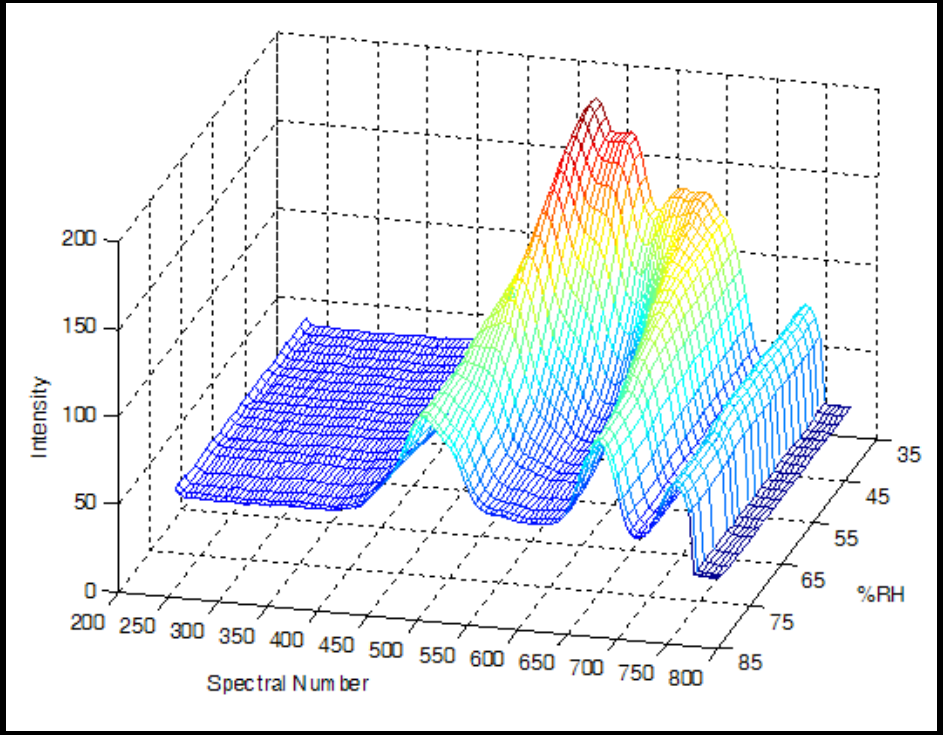
Gather input-output data

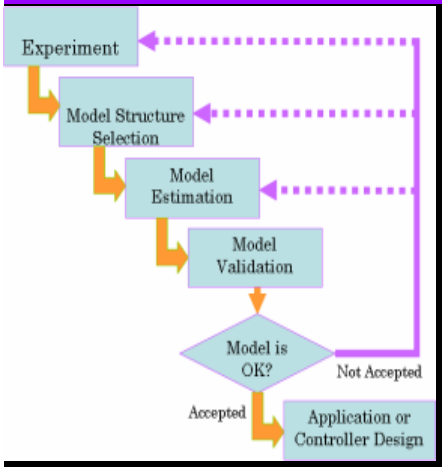





# Practical Guide: Sensor Modeling

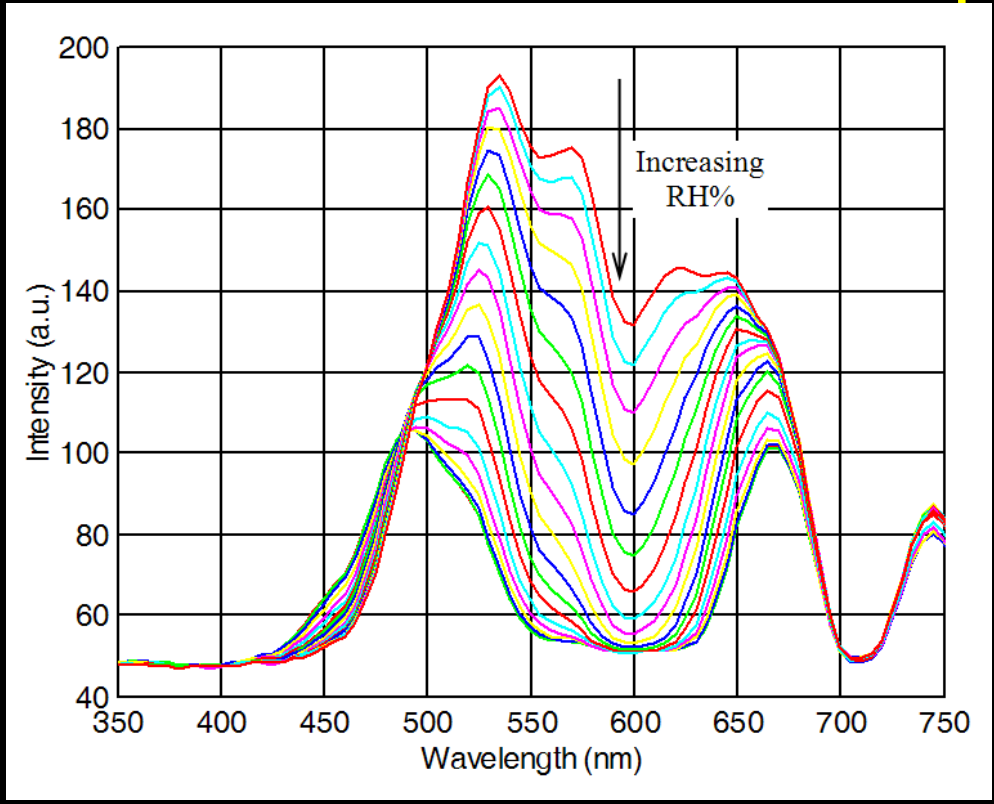
## Experiment

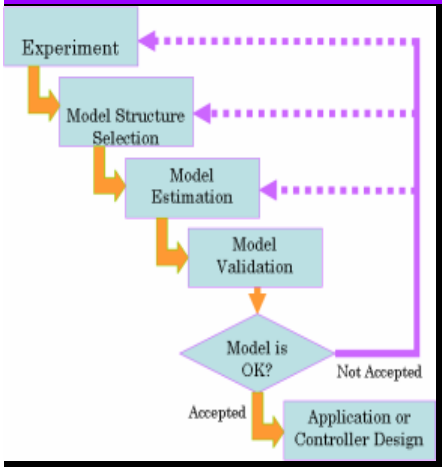





# Practical Guide: Sensor Modeling

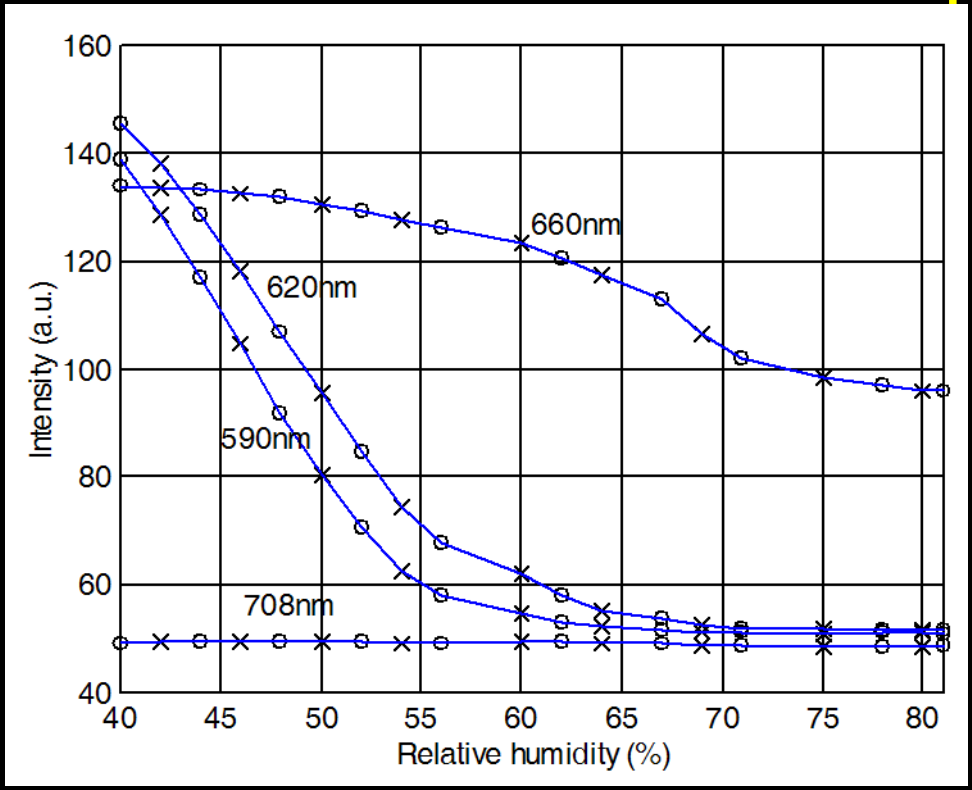
## Data preprocessing

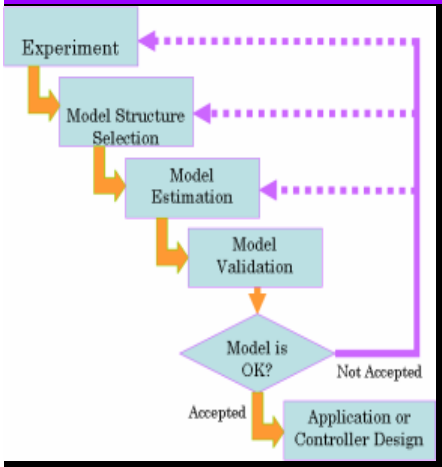





# Practical Guide: Sensor Modeling

## Data preprocessing

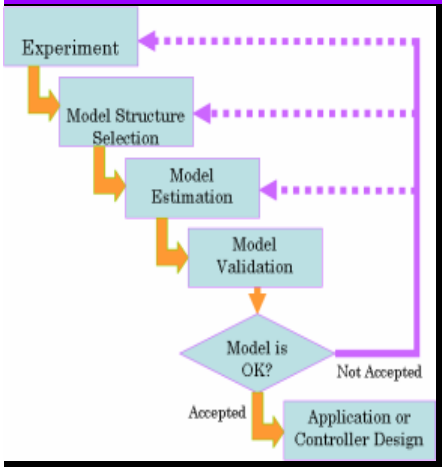





# Practical Guide: Sensor Modeling

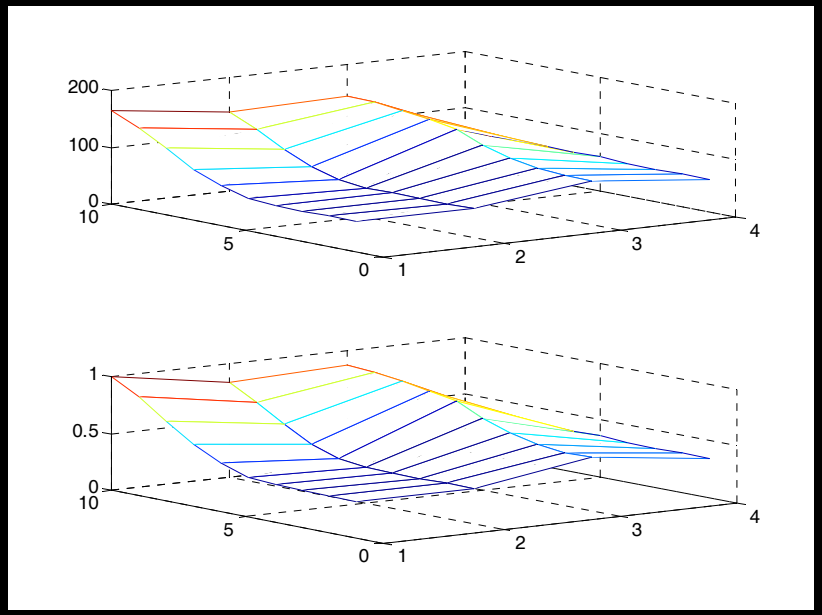
## Data preprocessing

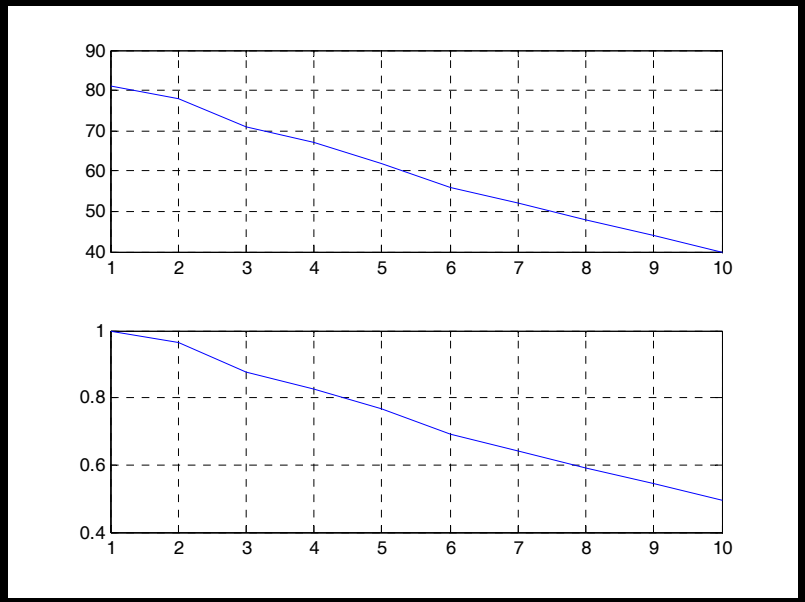
```
13      %load input data
14 -    x=load('trainInputs.m'); % input patterns
15 -    allx=load('AllInputs.m'); % all input patterns = trai
16 -    maxx=max(allx);maxx=max(maxx); % preprocess by scalin
17 -    x1=x/maxx;
18 -    y=load('trainTargets.m'); % targets
19 -    ally=load('AllTargets.m');
20 -    maxo=max(ally);maxo=max(maxo);
21 -    out_target=y/maxo;
```

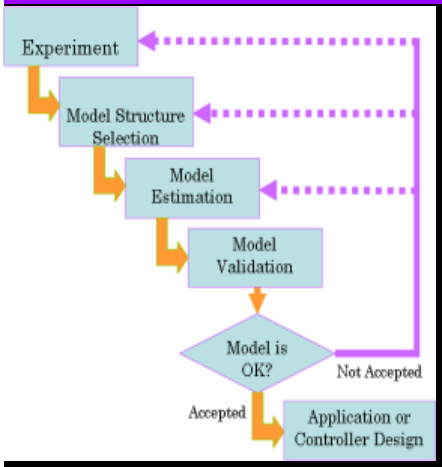



# Practical Guide: Sensor Modeling

## Data preprocessing



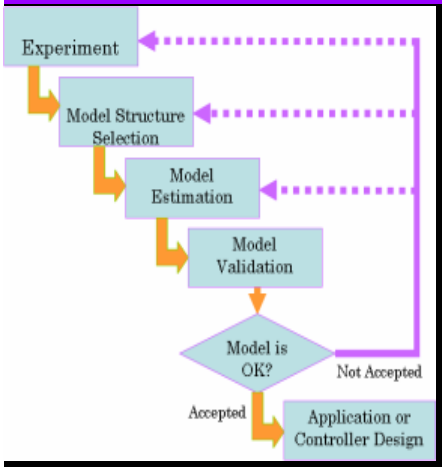



# Practical Guide: Sensor Modeling

## Model Structure

### Define the ANN architecture

```
26      %network parameters
27 -    net=newff(minmax(x1),[10,1],{'tansig','purelin'},'trainlm');
28 -    net.trainParam.show = 10;
29 -    net.trainParam.lr = 0.9;
30 -    net.trainParam.lr_inc = 0.001;
31 -    net.trainParam.mc = 0.7;
32 -    net.trainParam.epochs = 1000;
33 -    net.trainParam.goal = 0.00001;
```

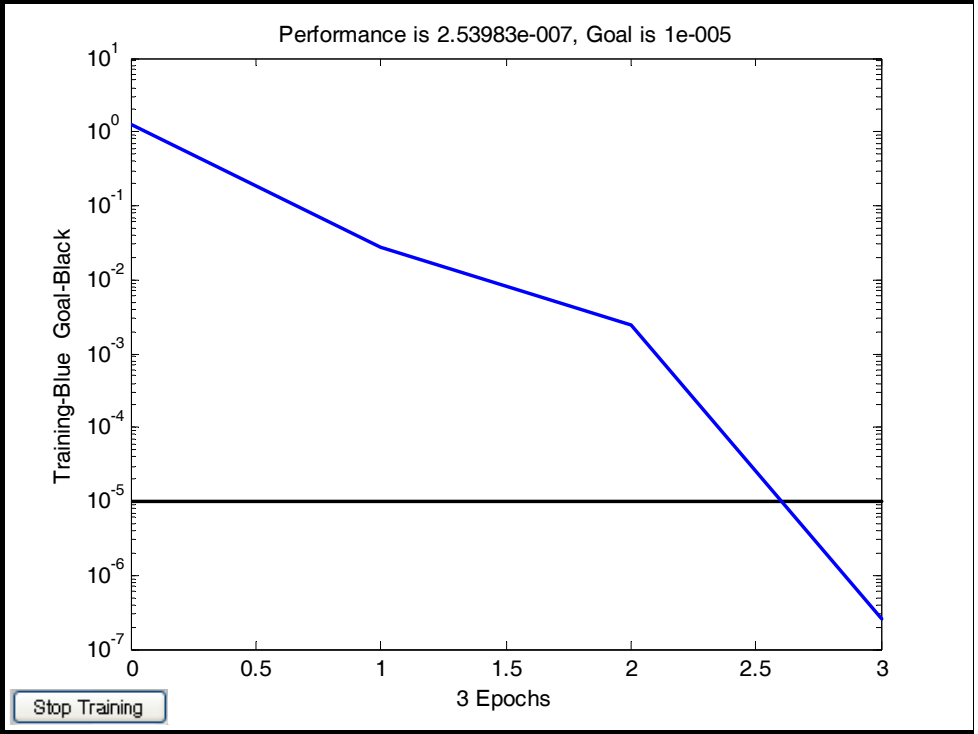


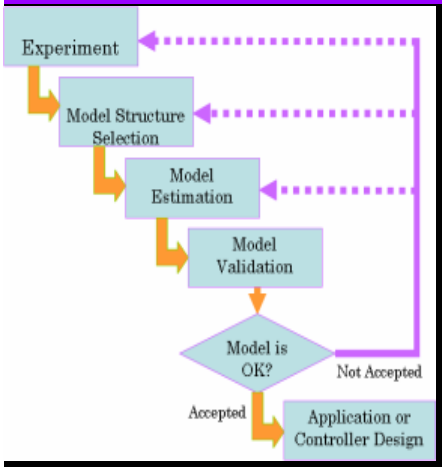

# Practical Guide: Sensor Modeling

## Model Estimation

## ANN Training

```
33 -   net.trainParam.goal = 0.00001;
34 -   [net,tr]=train(net,x1,y1);
35 -   pause
```



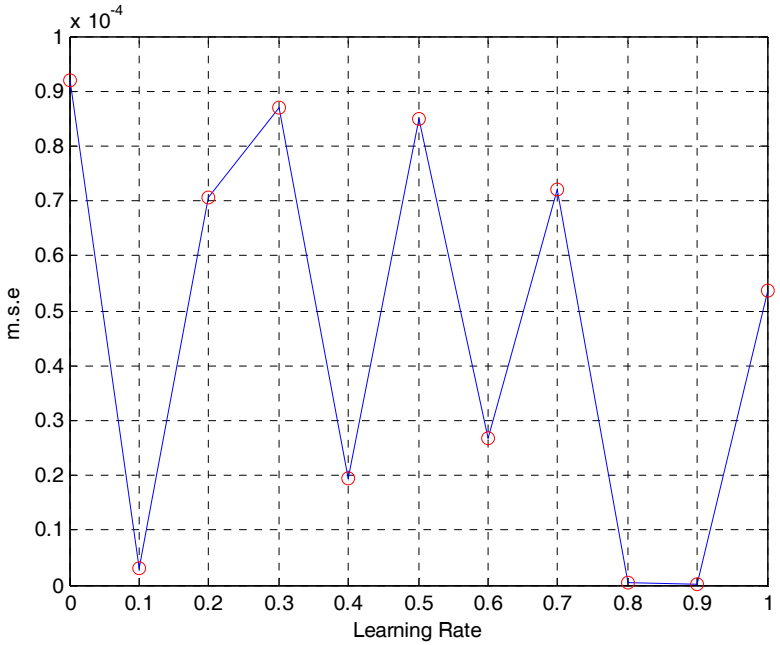


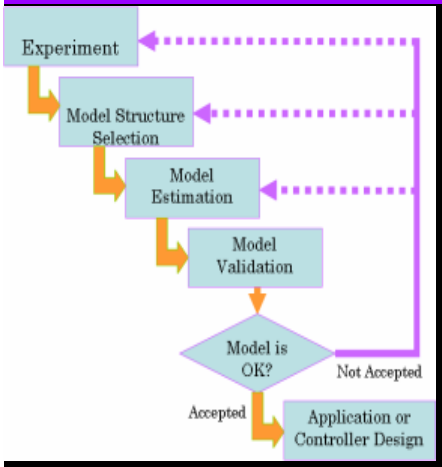

# Practical Guide: Sensor Modeling

## Model Estimation

## ANN Training – optimise learning rate

```
26    %network parameters
27 -  net=newff(minmax(x1),[10,1],{'tansig','purelin'},'t
28 -  net.trainParam.show = 10;
29 -  net.trainParam.lr = 0.9;
30 -  net.trainParam.lr_inc = 0.001;
31 -  net.trainParam.mc = 0.7;
32 -  net.trainParam.epochs = 1000;
33 -  net.trainParam.goal = 0.00001;
```



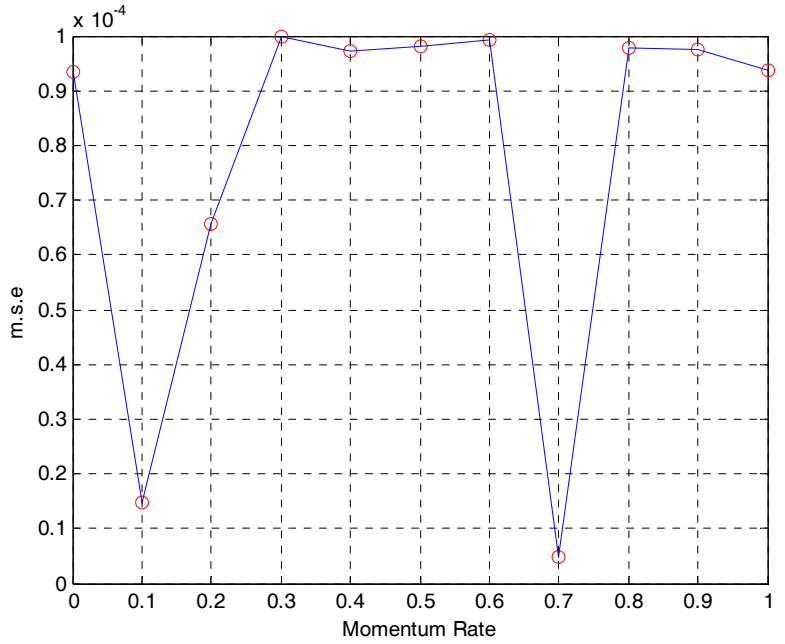


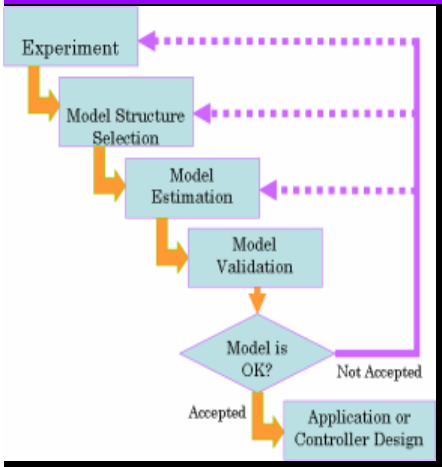

# Practical Guide: Sensor Modeling

## Model Estimation

## ANN Training – optimise momentum

```
26    %network parameters
27 -  net=newff(minmax(x1),[10,1],{'tansig','purelin'},'t
28 -  net.trainParam.show = 10;
29 -  net.trainParam.lr = 0.9;
30 -  net.trainParam.lr_inc = 0.001;
31 -  net.trainParam.mc = 0.7;
32 -  net.trainParam.epochs = 1000;
33 -  net.trainParam.goal = 0.00001;
```



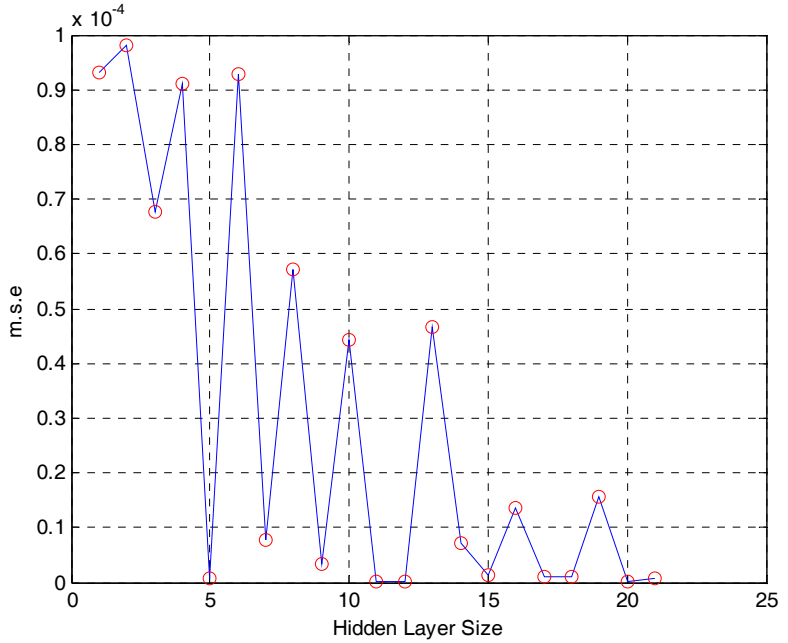


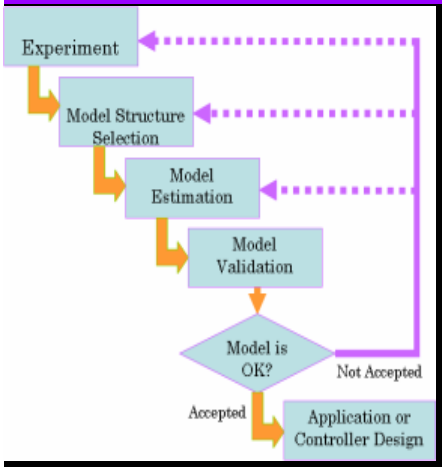

# Practical Guide: Sensor Modeling

## Model Estimation

## ANN Training – optimise hidden layer

```
26    %network parameters
27 -  net=newff(minmax(x1),[10,1],{'tansig','purelin'},'
28 -  net.trainParam.show = 10;
29 -  net.trainParam.lr = 0.9;
30 -  net.trainParam.lr_inc = 0.001;
31 -  net.trainParam.mc = 0.7;
32 -  net.trainParam.epochs = 1000;
33 -  net.trainParam.goal = 0.00001;
```



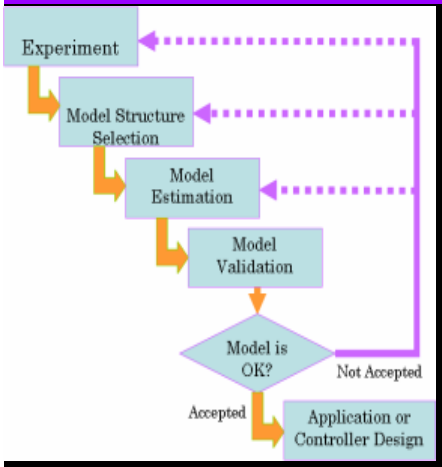



# Practical Guide: Sensor Modeling

## Model Validation

### Check errors

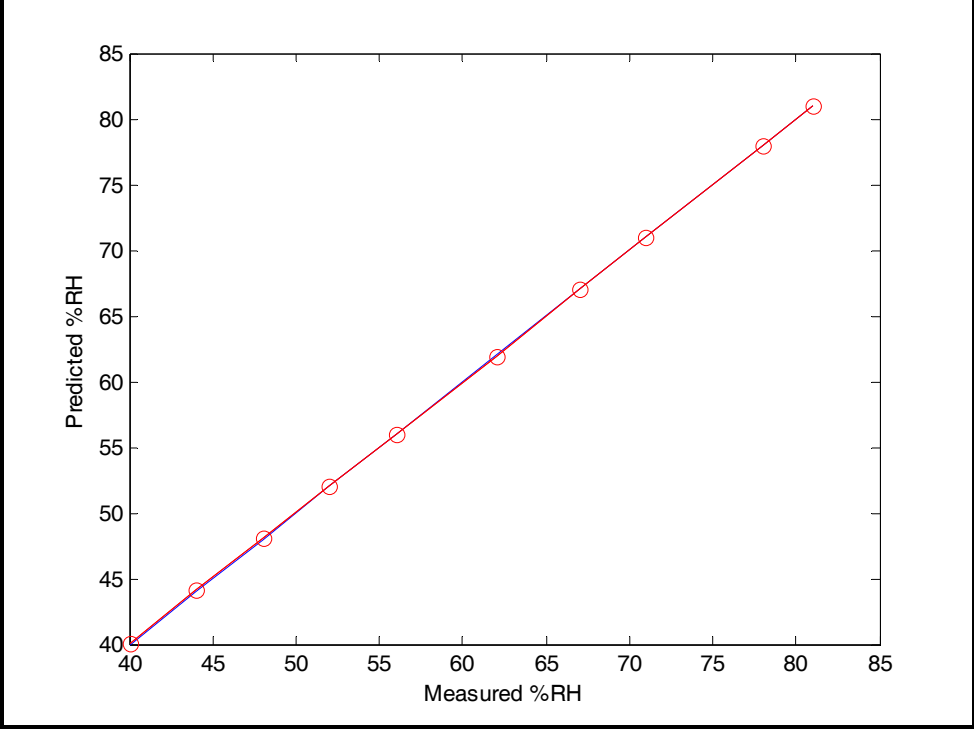
```
37      %check calculation of outputs for training data
38 -    simtrain=sim(net,x1); % forward pass
39 -    etrain=y1-simtrain; % calculate prediction errors
40 -    figure,plot(y,y1*maxo,'-b',y,simtrain*maxo,'-r');
41 -    hold,plot(y,simtrain*maxo,'-ro');
42 -    xlabel('Measured %RH')
43 -    ylabel('Predicted %RH');
44 -    grid
45      %save all data
46 -    save net1.mat
```

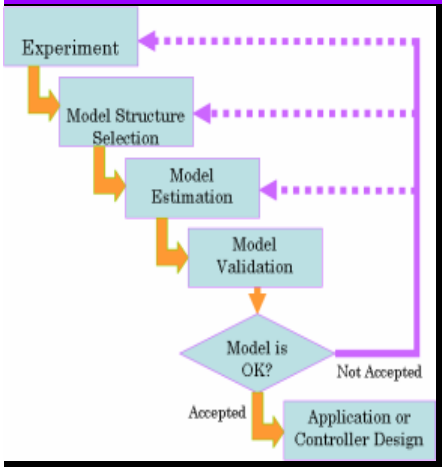



# Practical Guide: Sensor Modeling

## Model Validation

Check errors

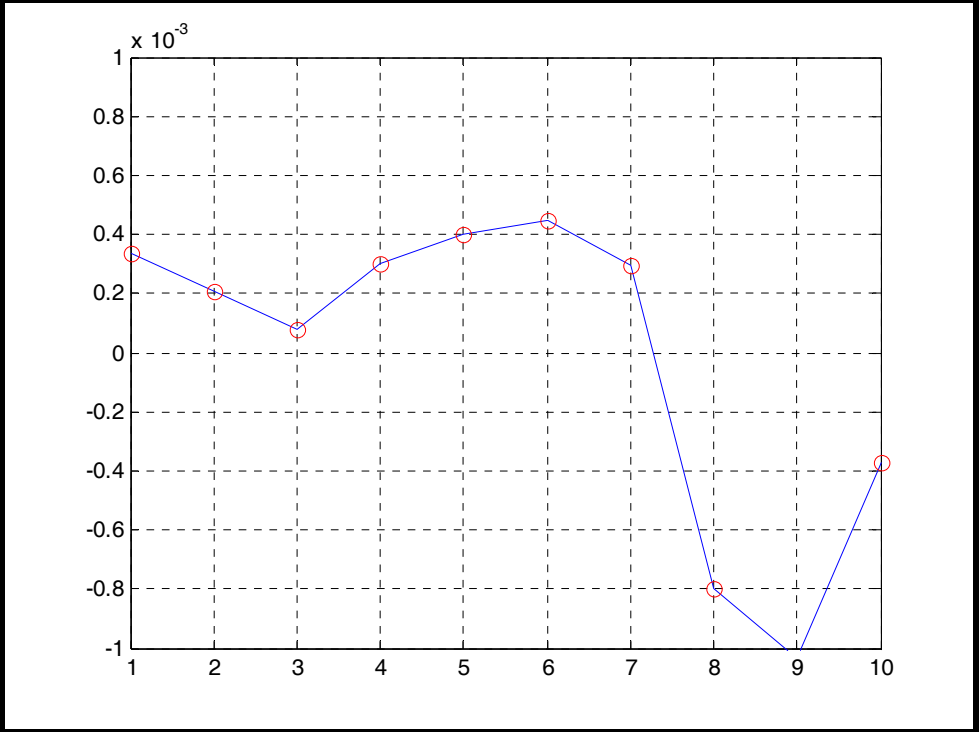





# Practical Guide: Sensor Modeling

## Model Validation

Check errors



# Conclusion

- It has been shown that a sensor system can be easily identified by the black box ANN approach.
- Quick & simple deployment of system identification

# Conclusion

Thank you