System Identification for Sensing & Process

21-22 Nov 2007, UiTM Shah Alam

Prof. Dr Mohd Nasir Taib

Advanced Signal Processing (ASP) Research Group

Faculty of Electrical Engineering

Universiti Teknologi MARA

Email: dr.nasir@ieee.org

Web: http://asprg.org

Content

Introduction

OUTCOMES

The course is designed so that participants could:

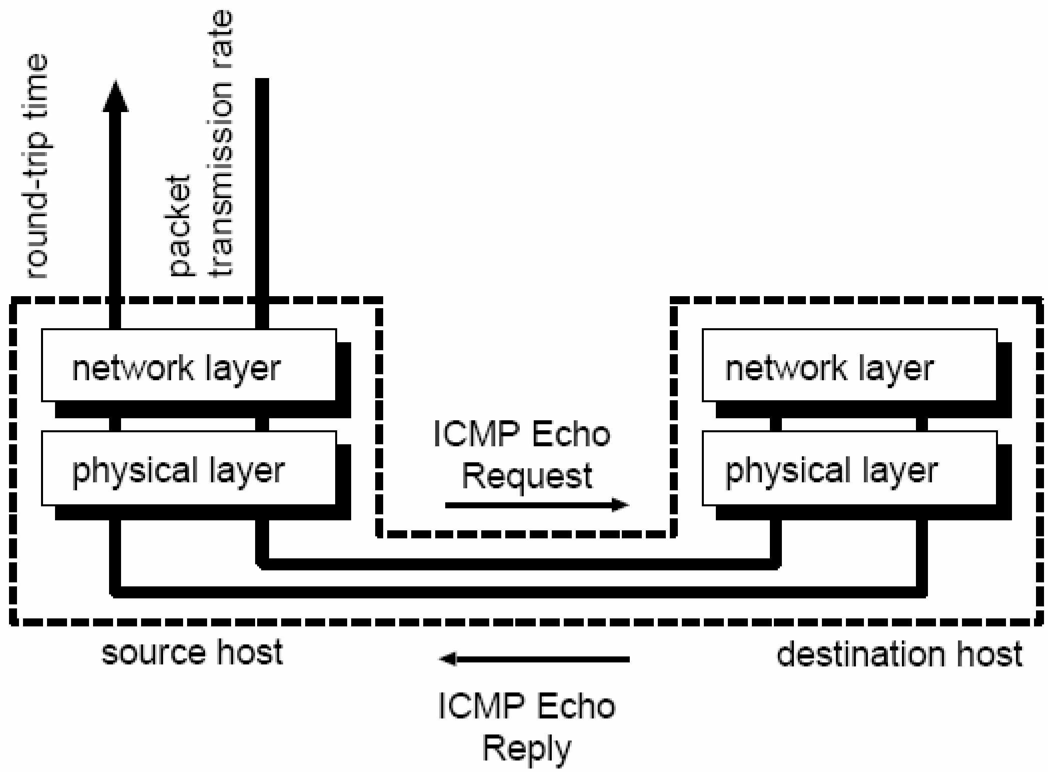
- Understand & appreciate System Identification
- Apply ANN based, linear & nonlinear System Identification Techniques

Introduction

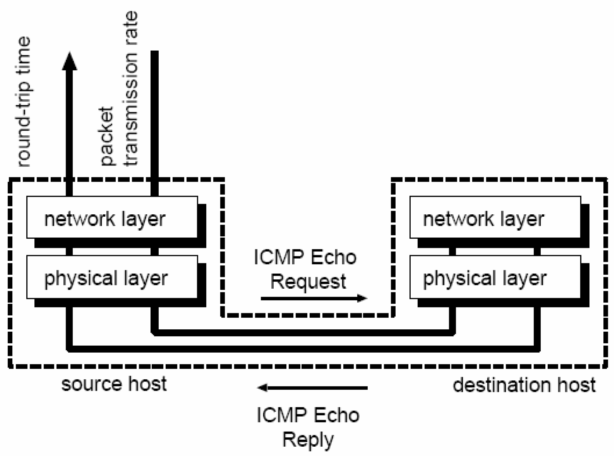
- 1950's, much of control design relied on Bode, Nyquist or Ziegler-Nichols charts, or on step response analyses
- These techniques were limited to control design for single-input single-output (SISO) systems.
- Around 1960, Kalman introduced the state-space representation and laid the foundations for 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

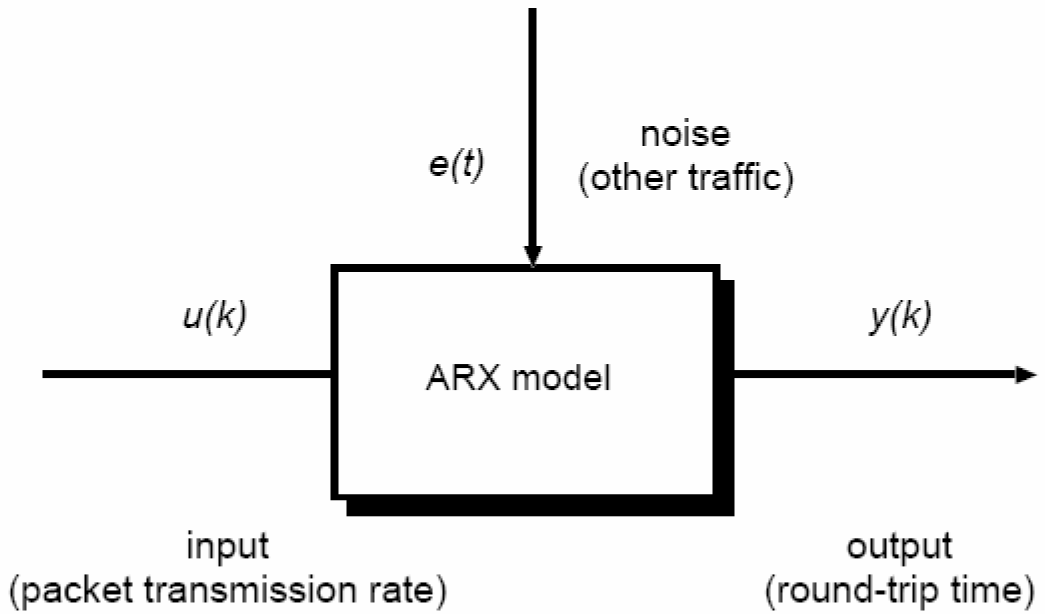
- System identification arose from the development of data-based techniques that would allow one to develop dynamical models for such diverse fields as process control, environmental systems, biological and biomedical systems, transportation systems, etc.



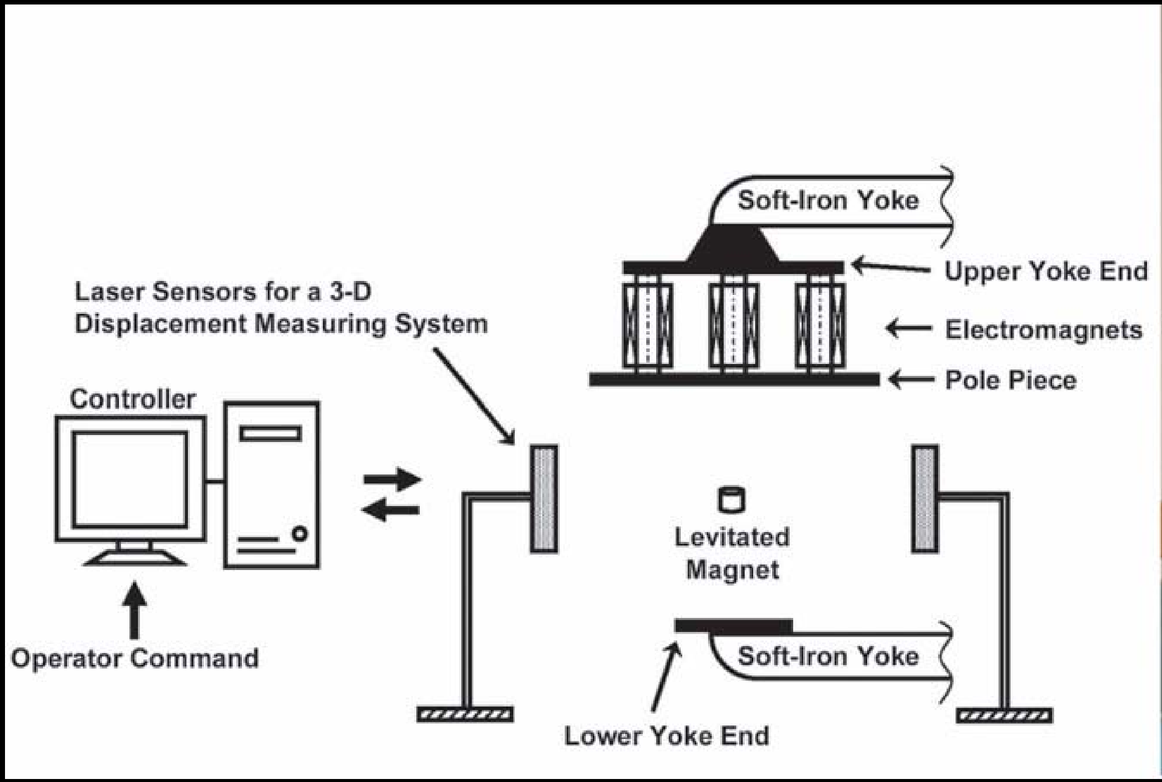
Modeling round-trip time dynamics as SISO system.

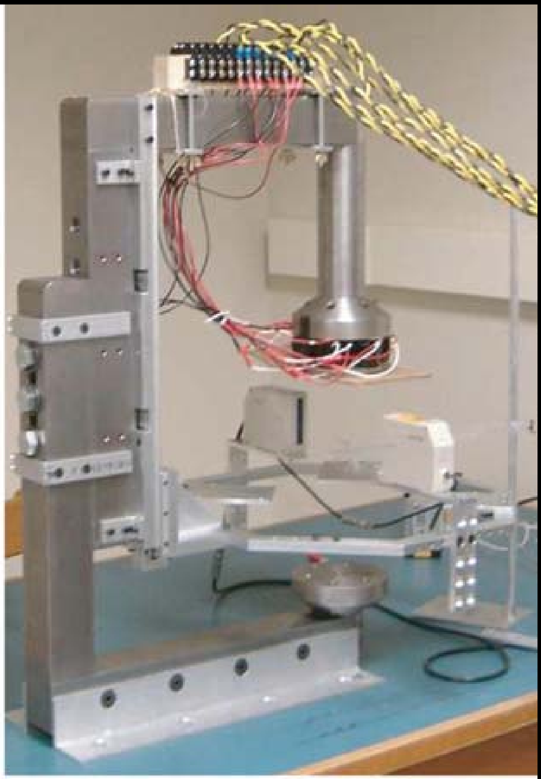


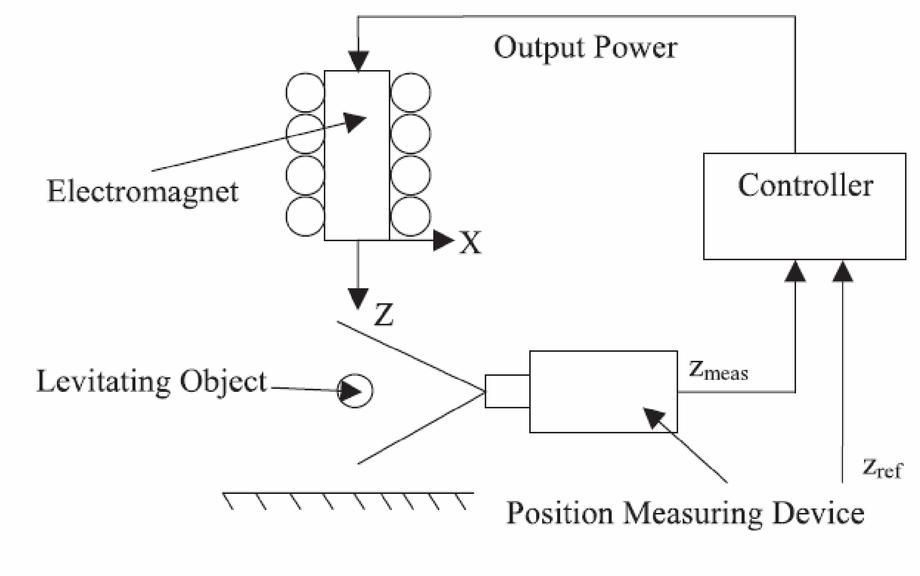
Modeling round-trip time dynamics as SISO system.



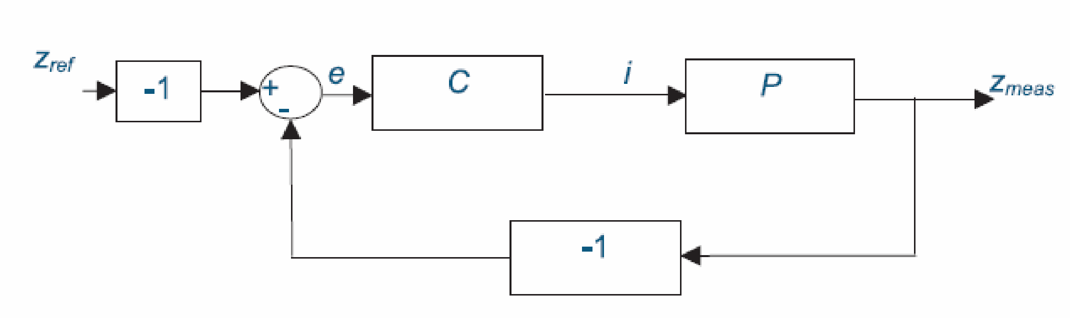
ARX model for modeling round-trip time dynamics.


Soft-Iron Yoke
Upper Yoke End
Electromagnets
Pole Piece
Laser Sensors for a 3-D
Displacement Measuring System
Controller
Levitated
Magnet
Operator Command
Soft-Iron Yoke
Lower Yoke End

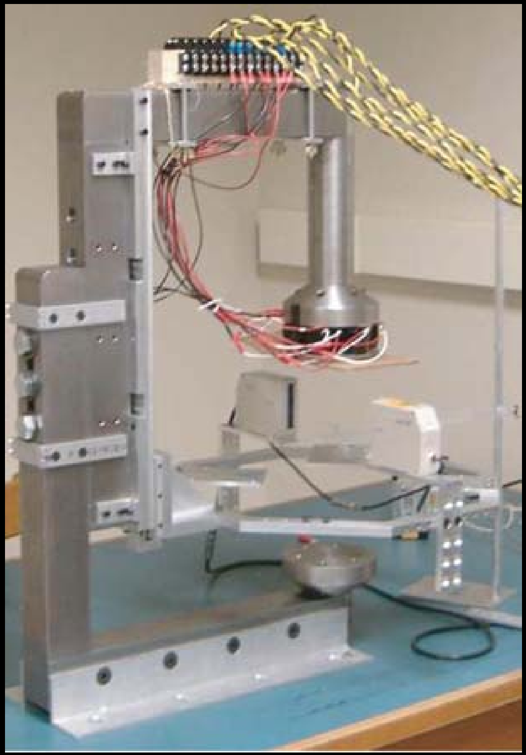


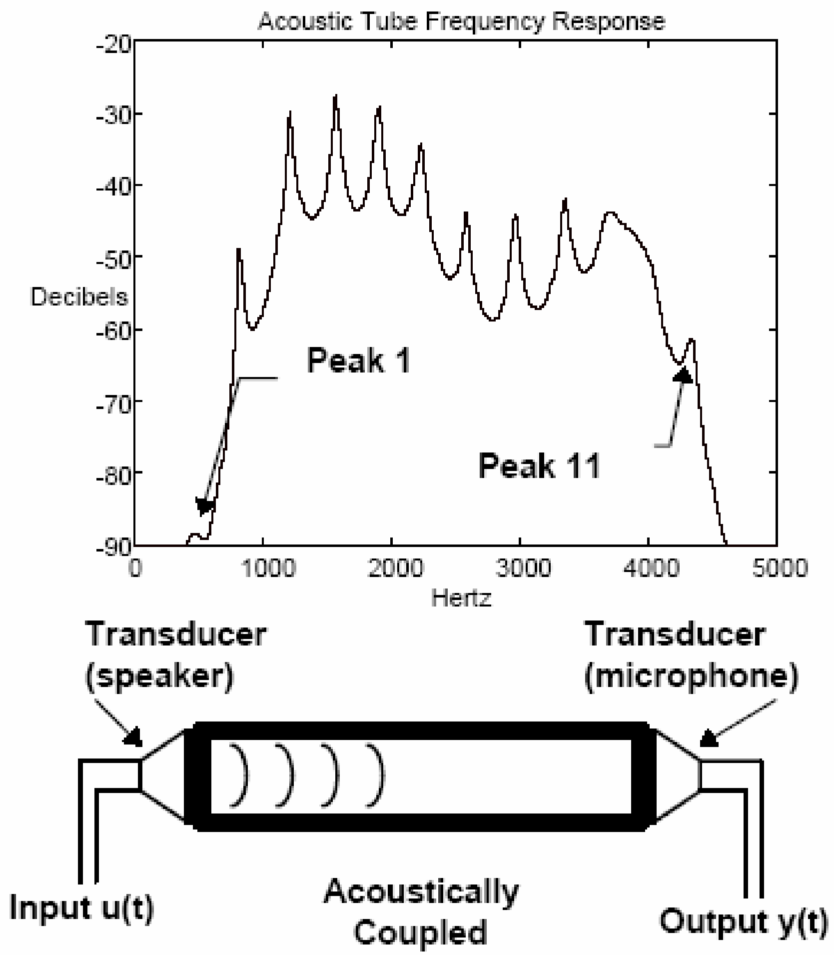


A simple one-dimensional magnetic levitation system.

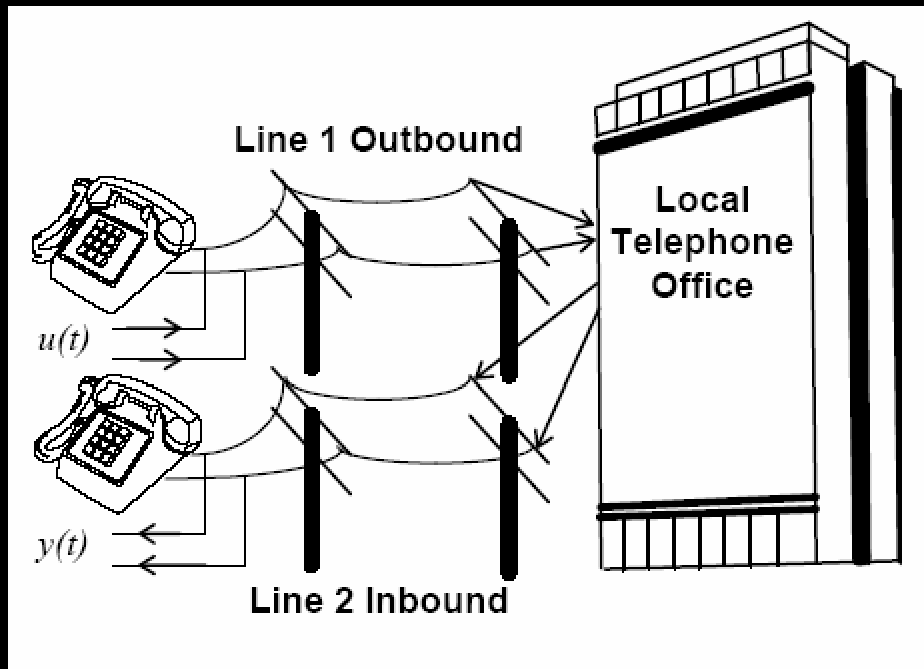


Block diagram of the closed loop system.

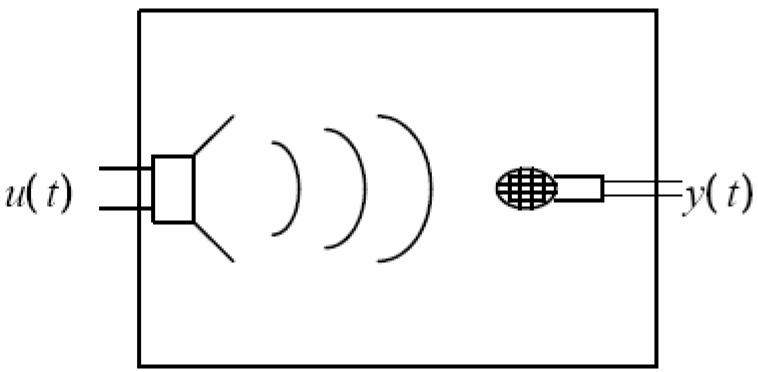




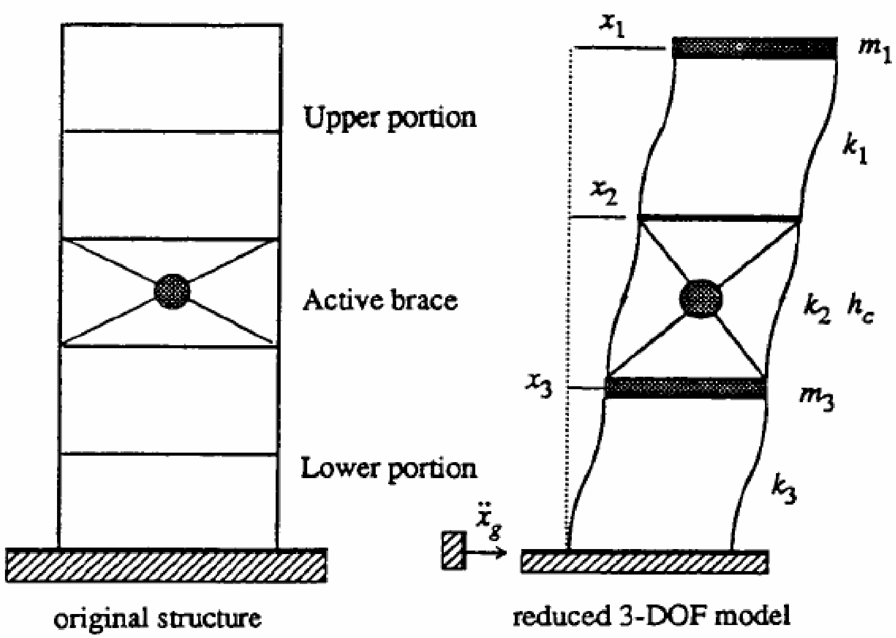
A "Simple" Acoustic System



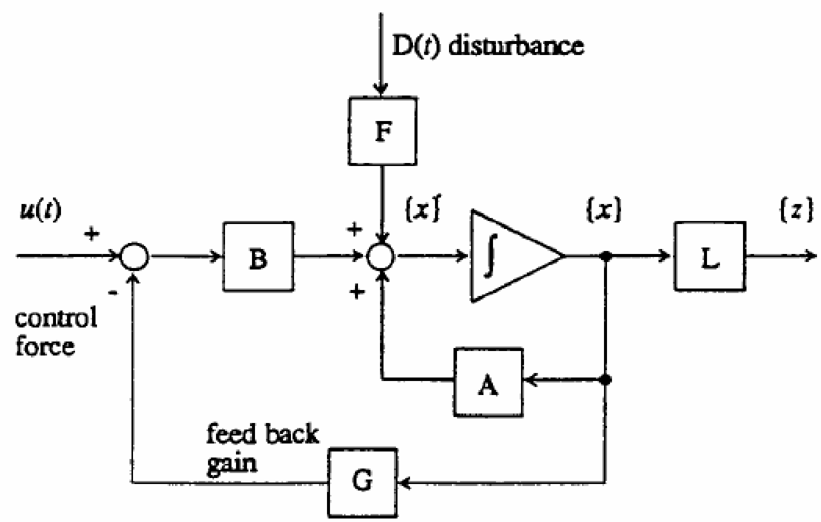
Characterizing Telephone Transmission Dynamics



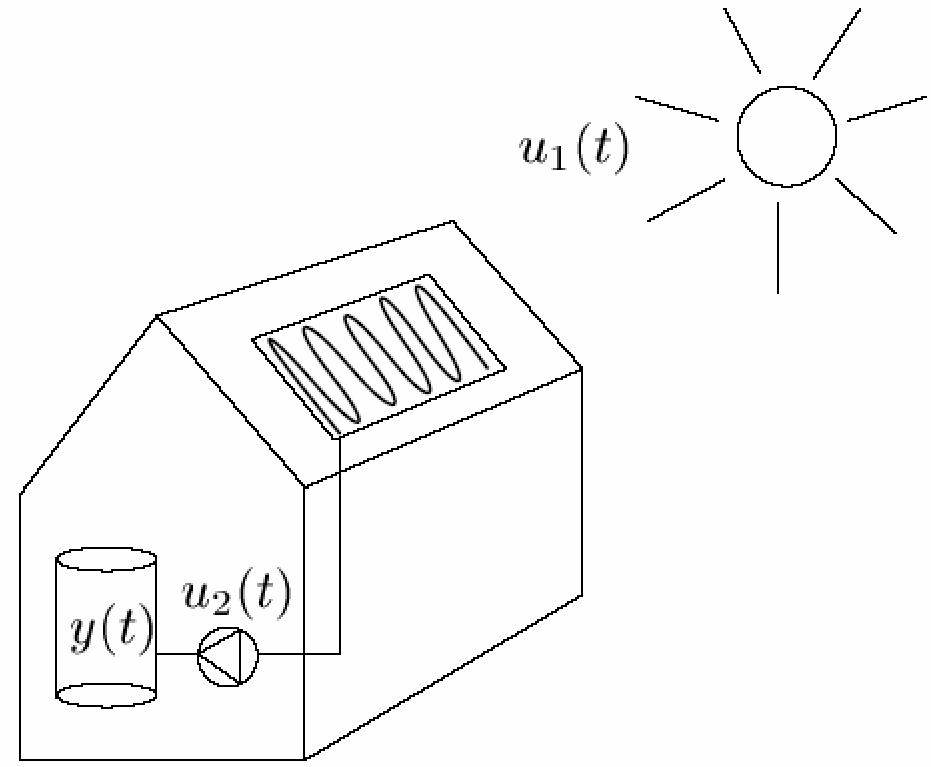
Acoustical signal processing
investigating noise-cancellation techniques



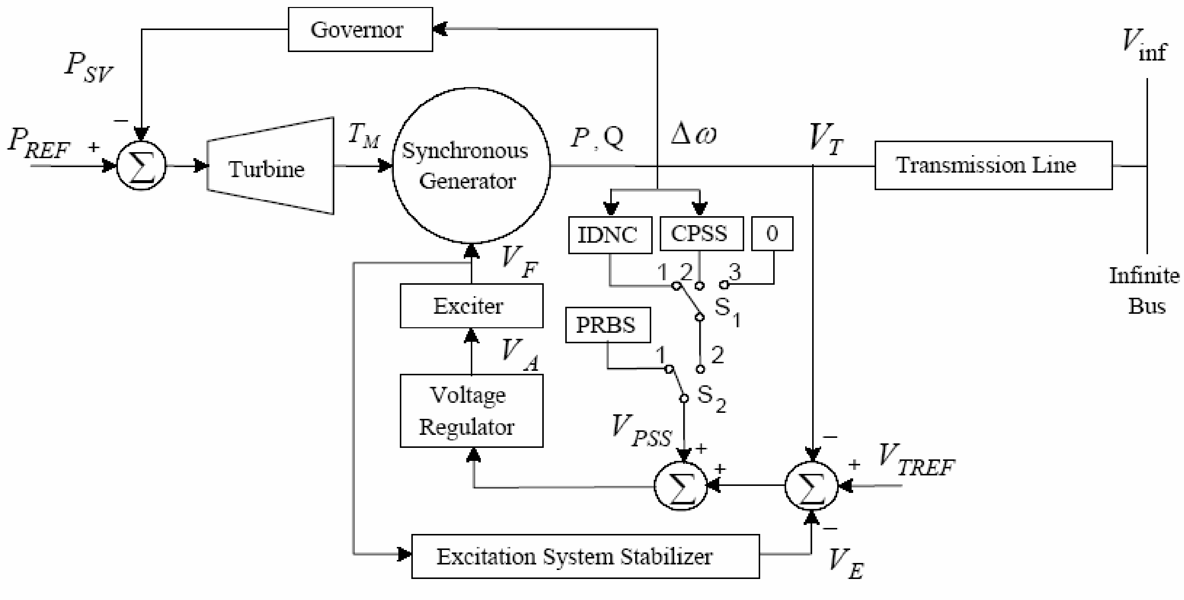
Study of the Active Control of a Building Model



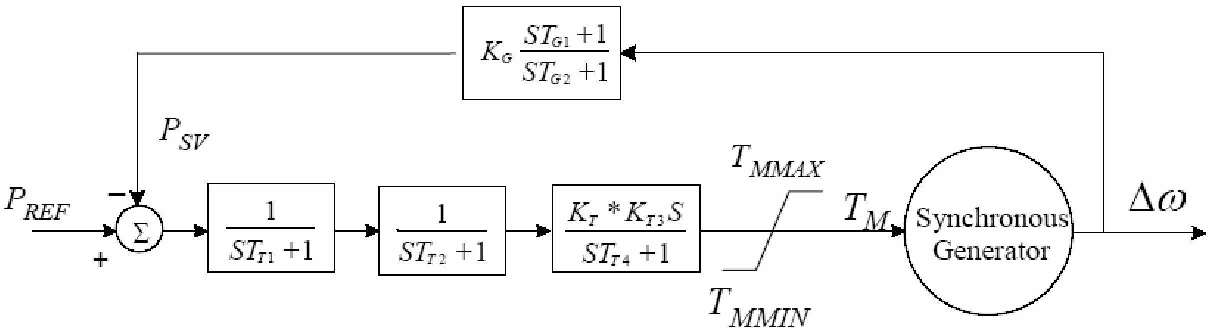
Block diagram of controlled structure.



A schematic picture of the solar heating system, with the measured variables irradiance (solar insolation), $u_1(t)$, fan switch state, $u_2(t)$, and heat storage temperature, $y(t)$.



Power system stabilizers (PSS)



Block diagram of the turbine and the governor

PERGAMON

Renewable Energy 22 (2001) 281–286

RENEWABLE ENERGY

www.elsevier.com/locate/renene

Application of system identification modelling to solar hybrid systems for predicting radiation, temperature and load

S. Sinha[a,*], Sanjay Kumar[b], Tsuyoshi Matsumoto[a], Toshinori Kojima[a]

[a] *Department of Industrial Chemistry, Faculty of Engineering, Seikei University, 3-3-1 Kichijoji, Kitamachi, Musashino-shi, Tokyo, 180-8633, Japan*

[b] *Department of Global Environment Eng, Kyoto University, Sakyo-ku, Kyoto, 606-01, Japan*

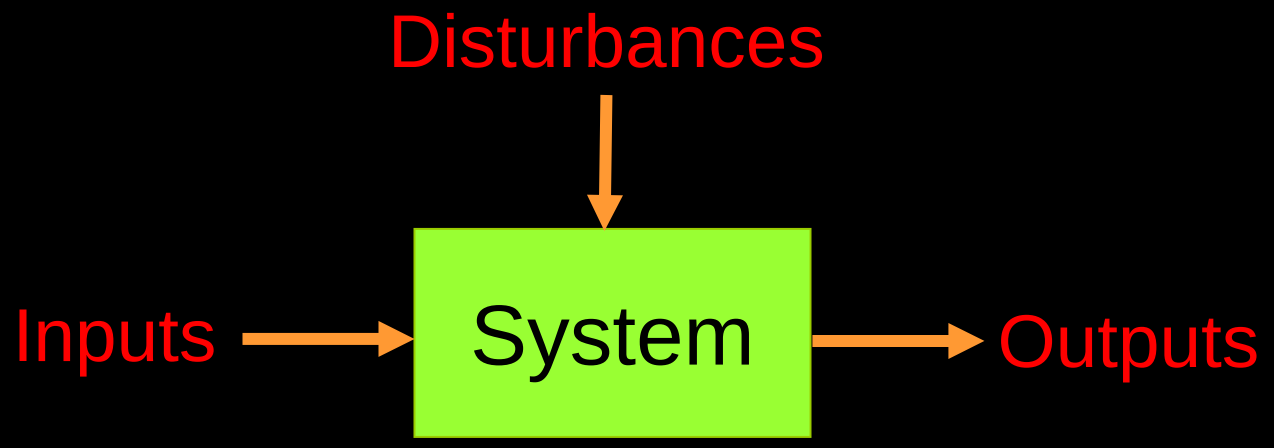
System Identification

"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."

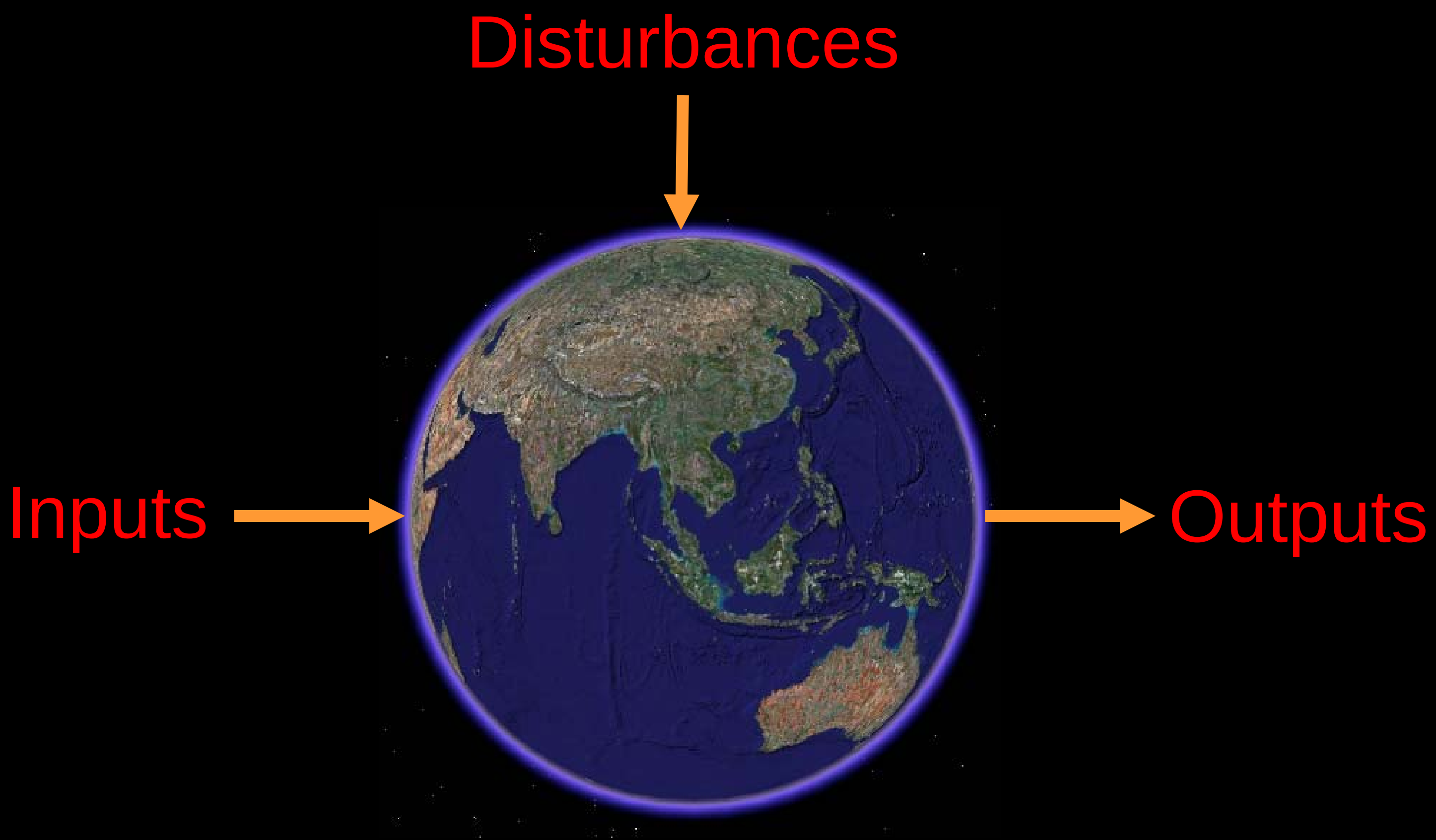
Zadeh (1962)

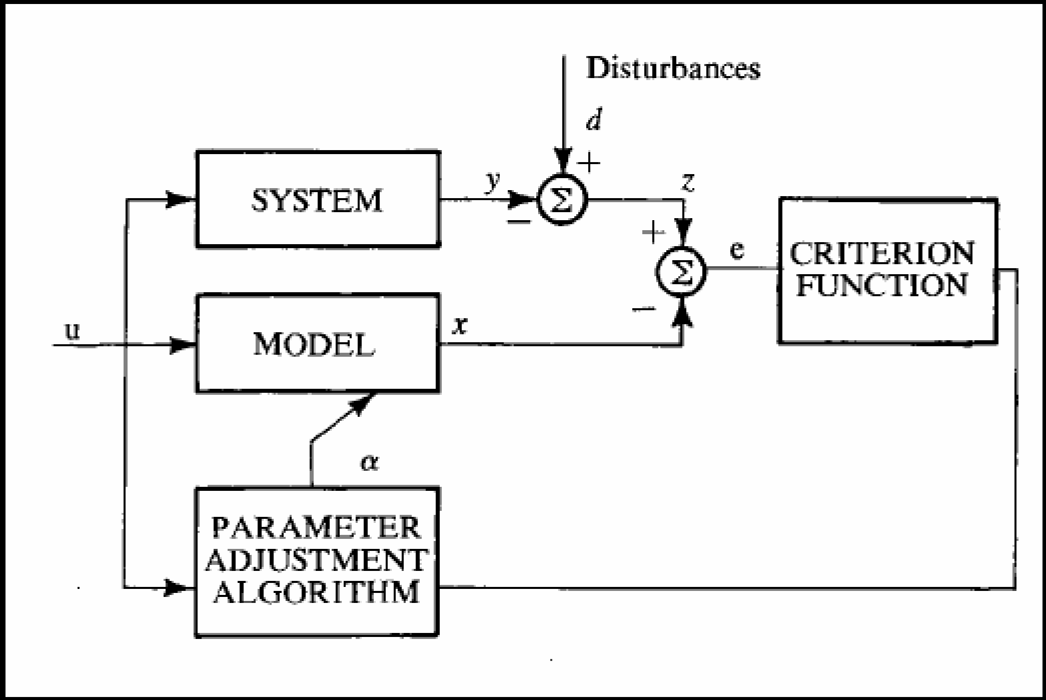
System Identification

Focuses on the modeling of dynamical systems from experimental data

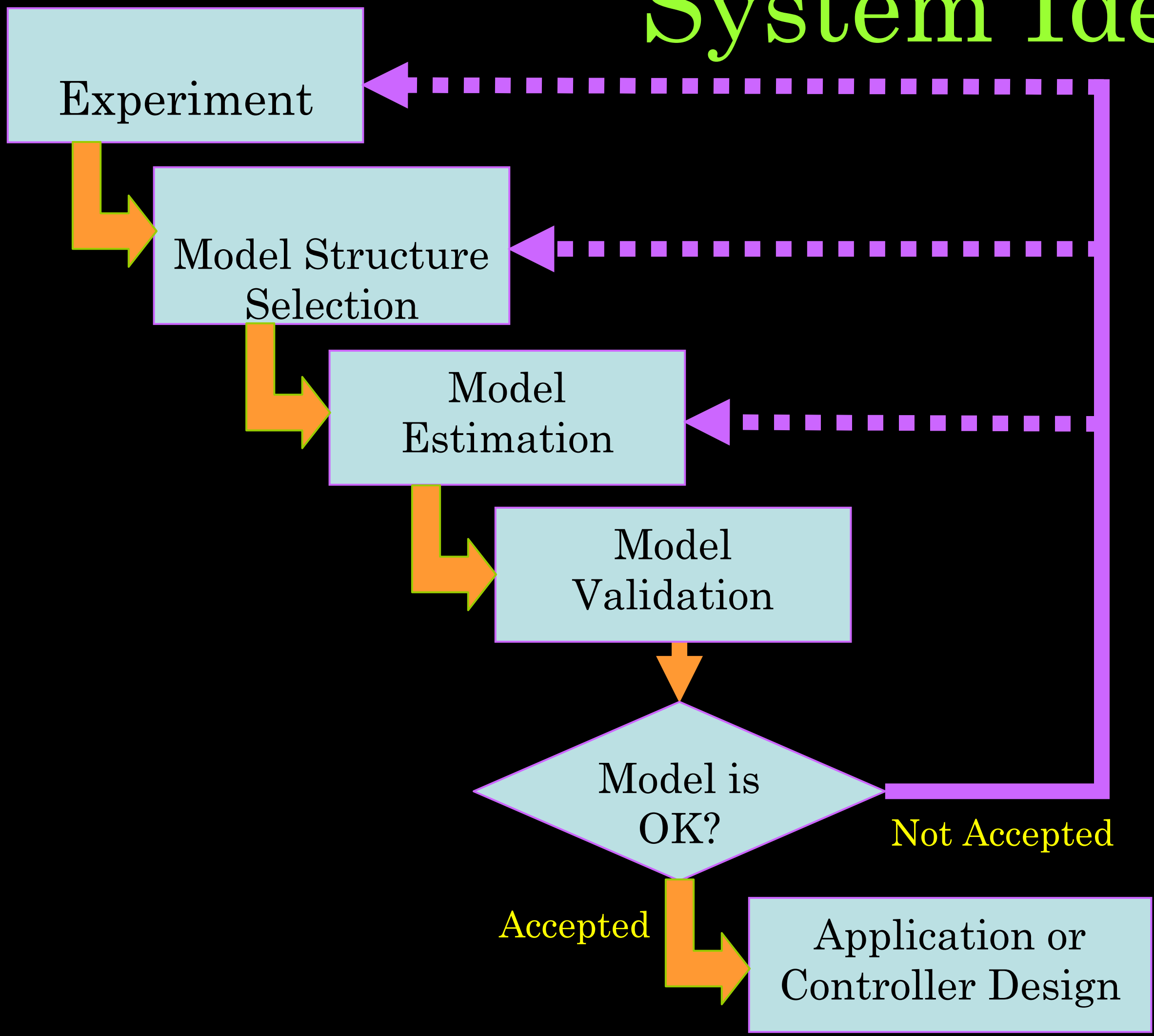


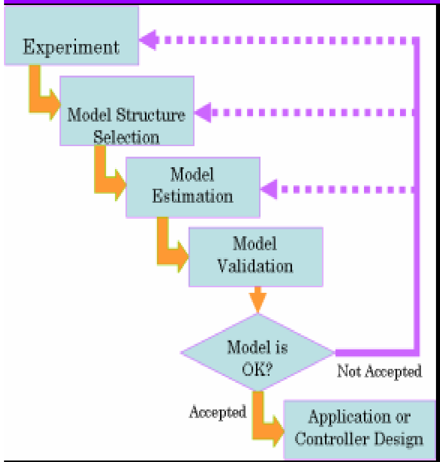
System Identification



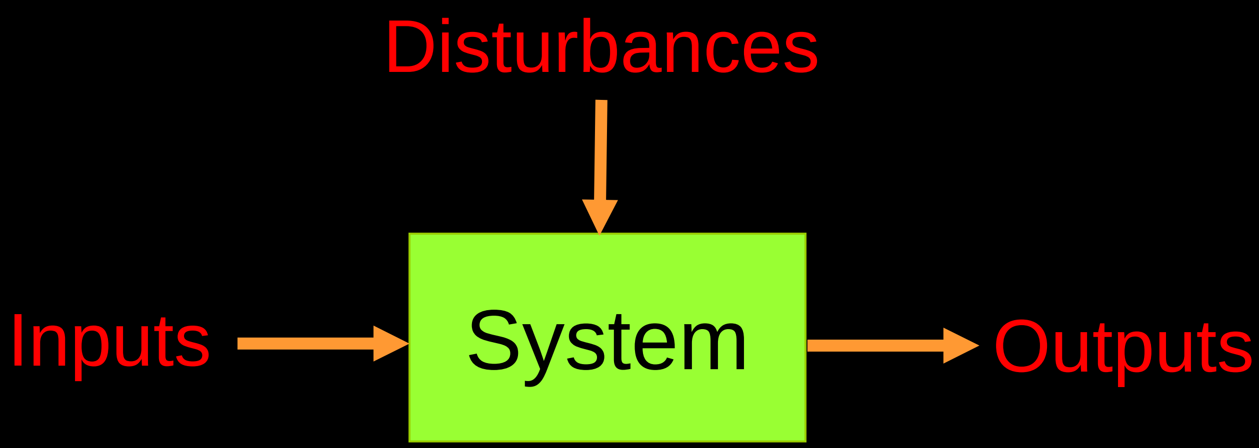

Disturbances
d
+
y
−
z
+
e
−
SYSTEM
MODEL
u
x
α
CRITERION
FUNCTION
PARAMETER
ADJUSTMENT
ALGORITHM

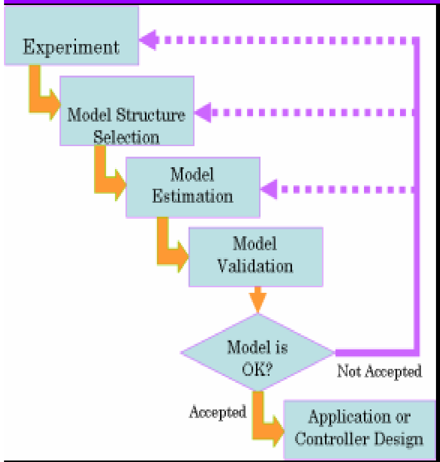
System Identification



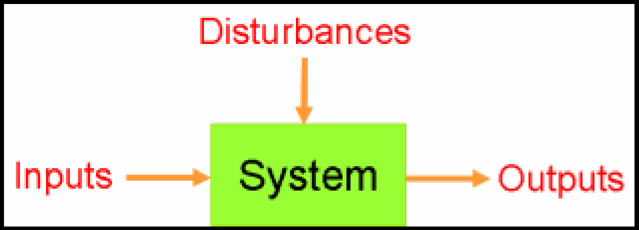


Step 1: Experiment





Step 1: Experiment



Step/Pulse Inputs

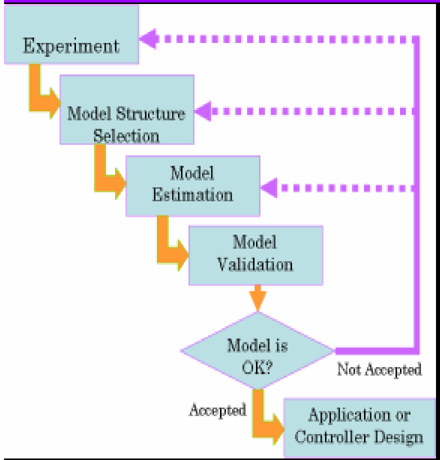
Gaussian White Noise

Random Binary Signal (RBS)

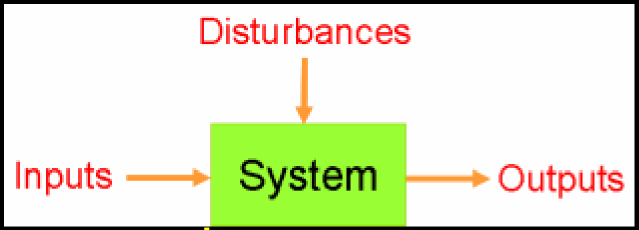
Pseudo-Random Binary Signal (PRBS)

multi-level Pseudo-Random Signals

Multi-sine inputs

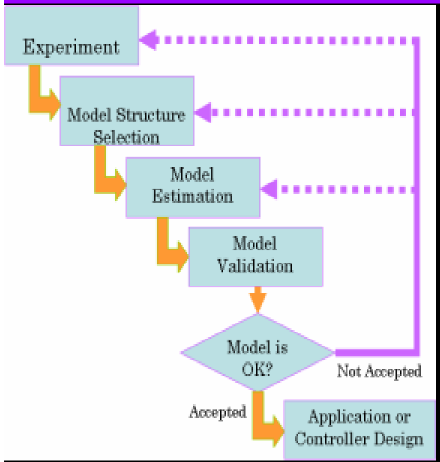


Step 1: Experiment

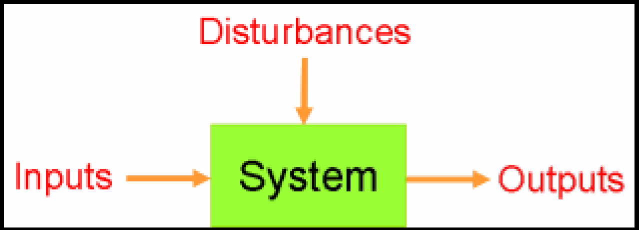


Data Sampling
- rule of thumbs ~ 4 to 10 samples per time constant (rise time)

Pre-process data
- filter noise
- Remove trend
- Remove outliers, *etc.*

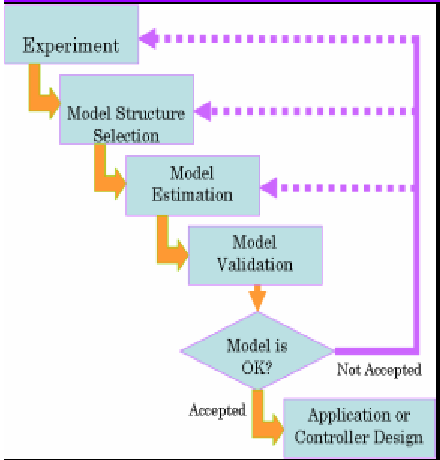


Step 1: Experiment



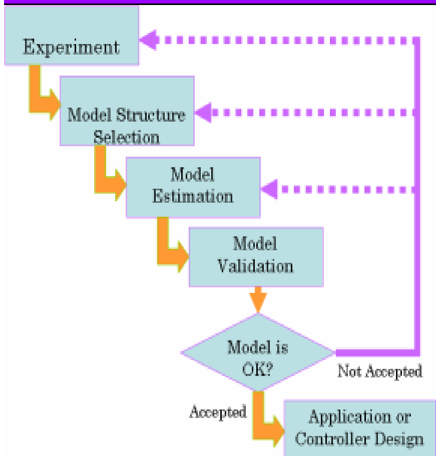
Split data into two parts

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



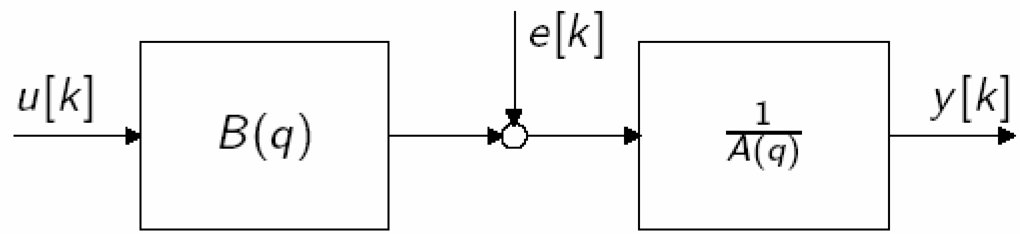
Step 2: Model Structure Selection

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

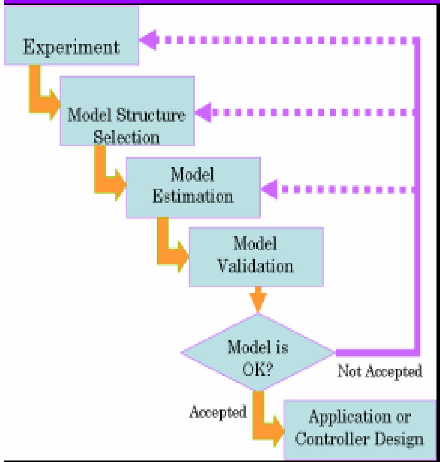


Step 2: Model Structure Selection

- ARX Model

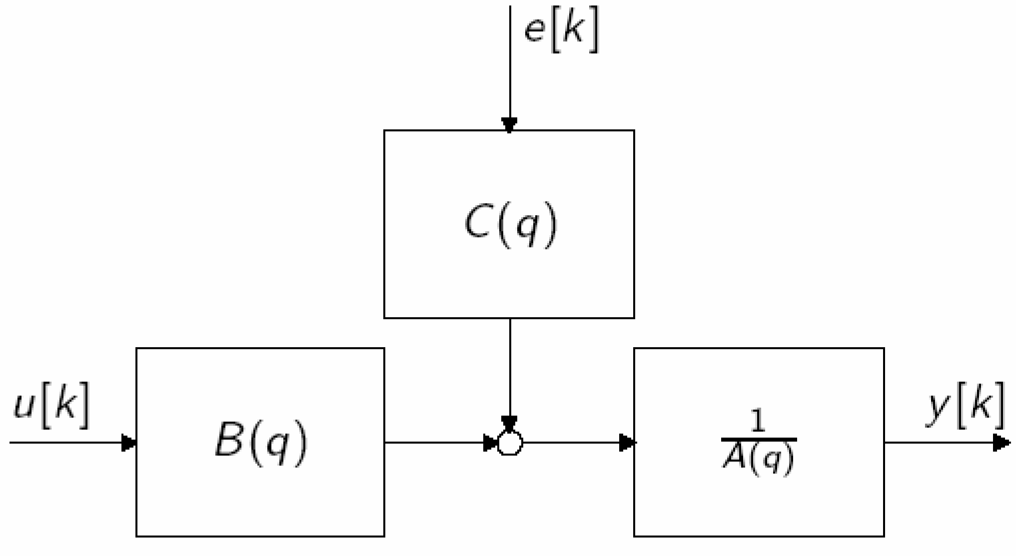


$$y[k] = \frac{B(q)}{A(q)}u[k] + \frac{1}{A(q)}e[k]$$

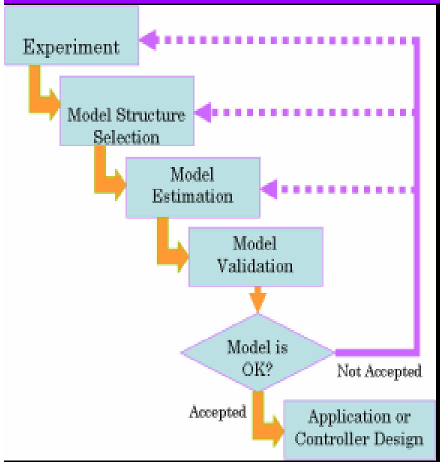


Step 2: Model Structure Selection

- ARMAX Model

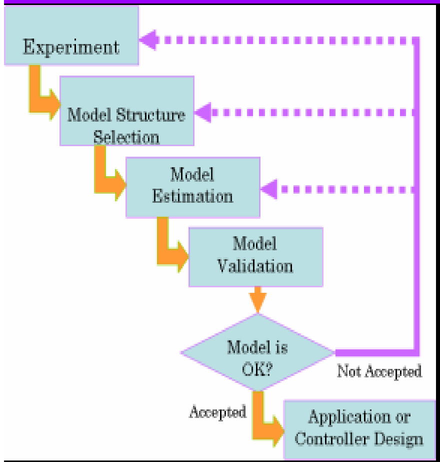


$$y[k] = \frac{B(q)}{A(q)}u[k] + \frac{C(q)}{A(q)}e[k]$$



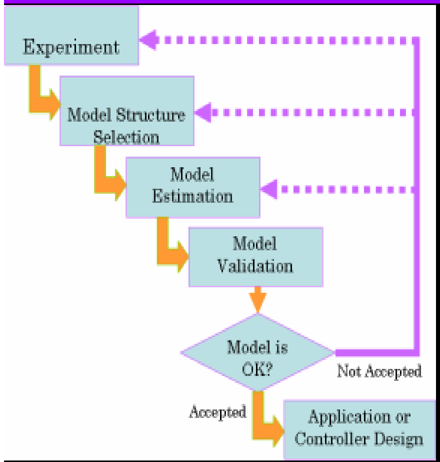
Step 3: Model Estimation

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



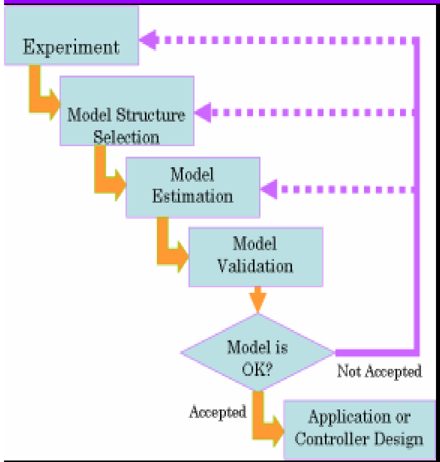
Step 3: Model Estimation

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



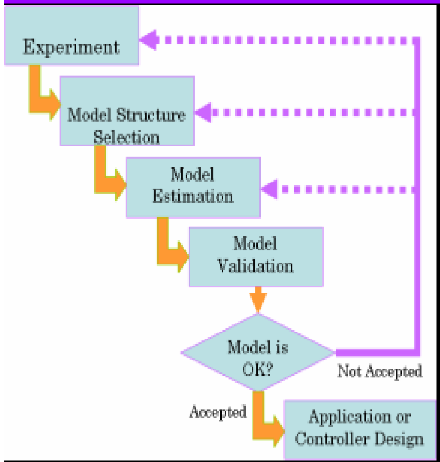
Step 3: Model Estimation

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



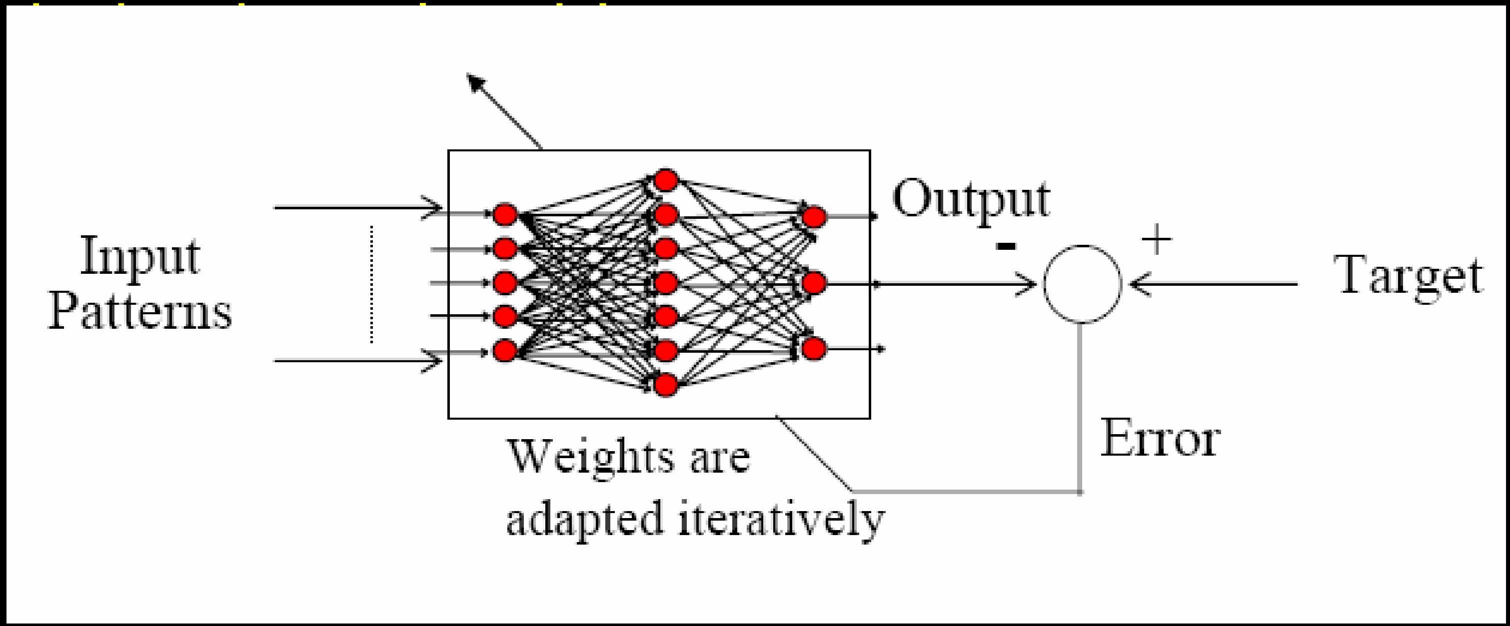
Step 3: Model Estimation

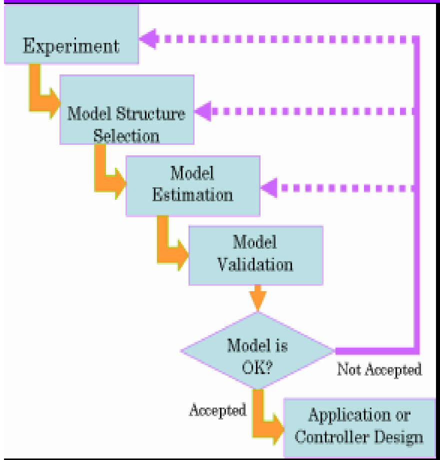
- ANN, Fuzzy, Hybrids
  - Optimisation algorithms



Step 3: Model Estimation

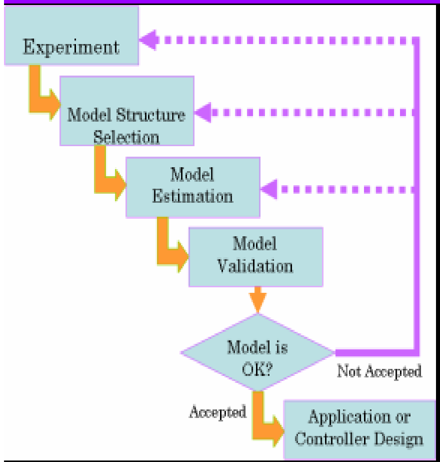
- ANN, Fuzzy, Hybrids
  - Op





Step 4: Model Validation

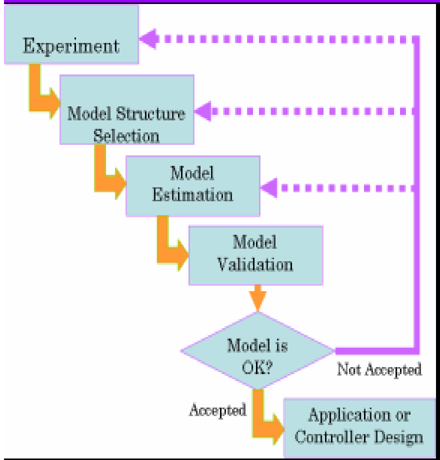
- Check for model ‘goodness’
- Compare model simulation/prediction with test data
- Perform statistical tests on prediction errors
- Compare estimated model's frequency response and spectral
- analysis result in frequency domain



Step 4: Model Validation

Residuals (errors)

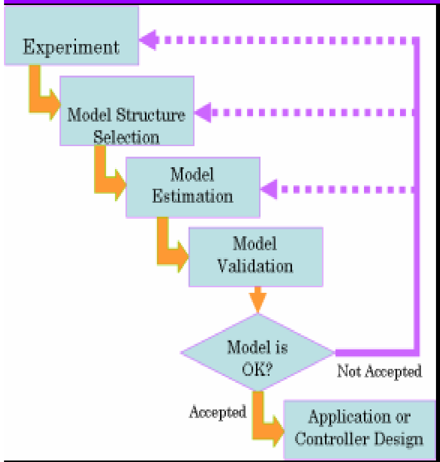
- If the model is correct the residuals should be
  - white
  - uncorrelated with input(s)



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)

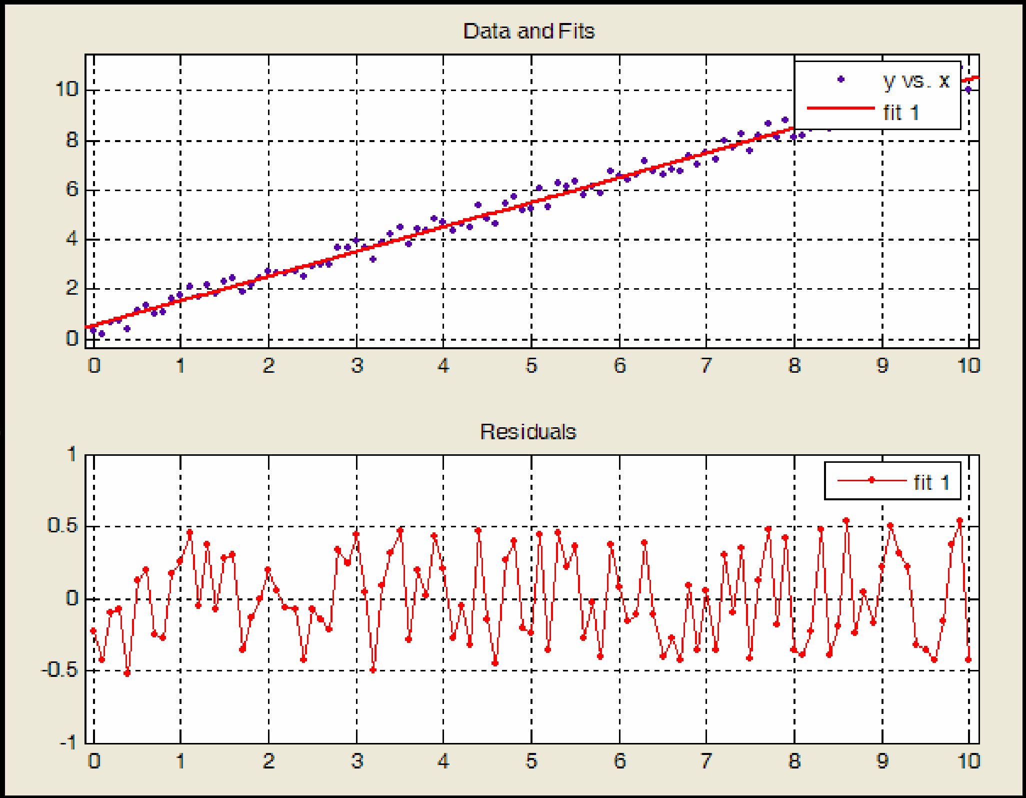
Goodness of fit:

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 φ 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: $\kappa(x) = \frac{1}{1+e^{-x}}$
- α coordinates, β scale or dilation, γ 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: κ is the "mother wavelet" and $\beta_j = 2^j$, $\gamma_k = 2^{-j}k$ (double indexing) as fixed choices
- (Neuro)-Fuzzy models: κ 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

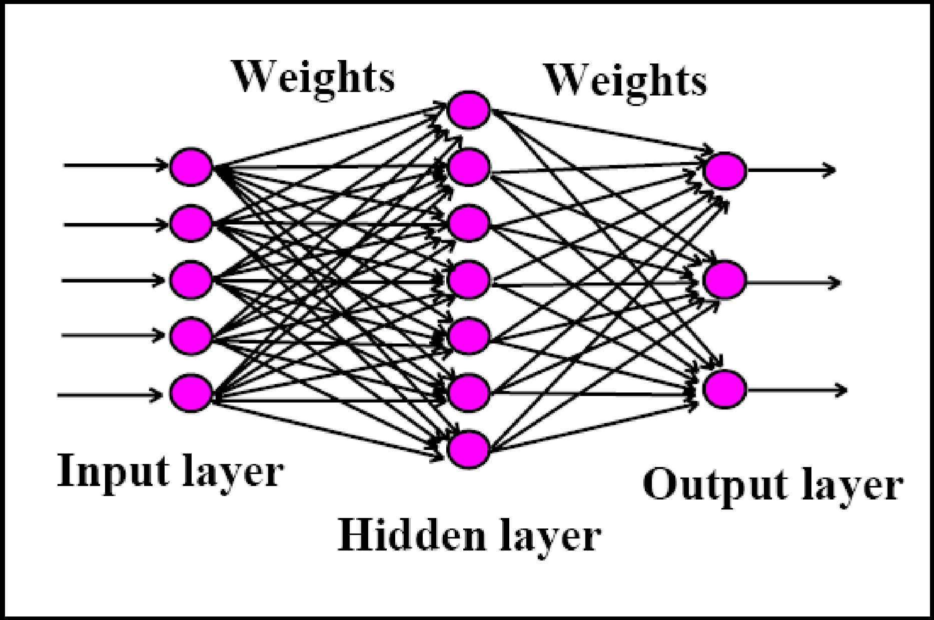
- ANN can be trained to capture and generalise the non-linear mechanism underlying the sensor's input-output relationship, thus memorising the behaviour of the sensor's response after which it will be able to provide an accurate prediction during recall process.
- The training is the hardest part in 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.

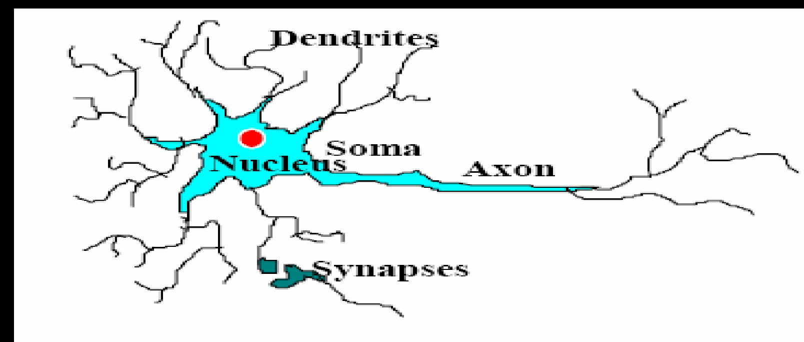
ANN

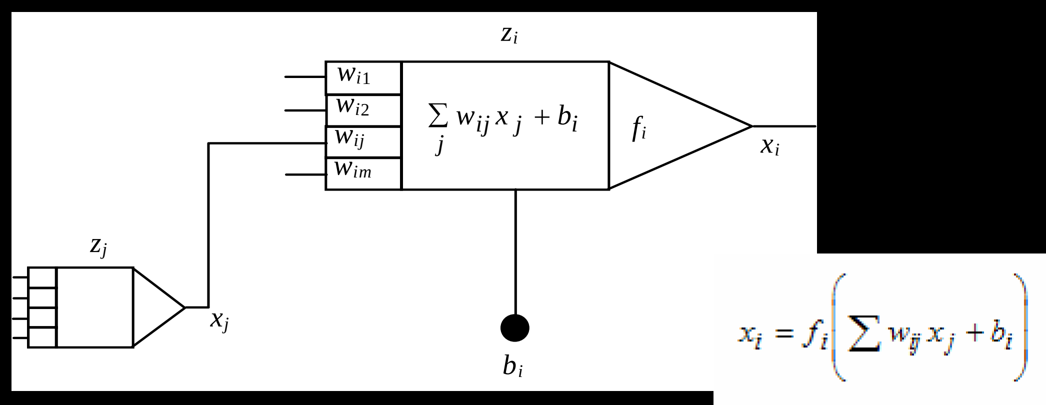
- ANN is a highly parallel processing system
- Consists of interconnected simple processing units called neurons (nodes)



ANN

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



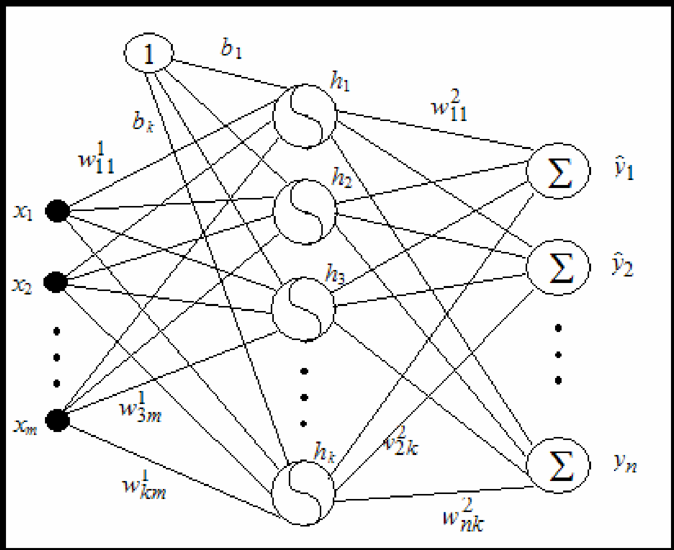


ANN

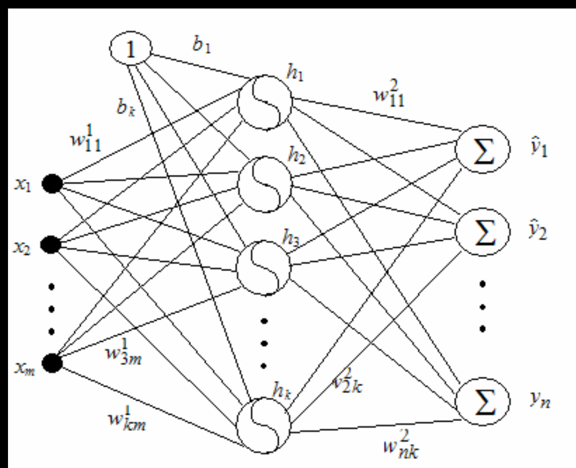
- Signals are processed from input to output by multiplying the weights and activated signals

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

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



ANN



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

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

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

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

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.

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

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

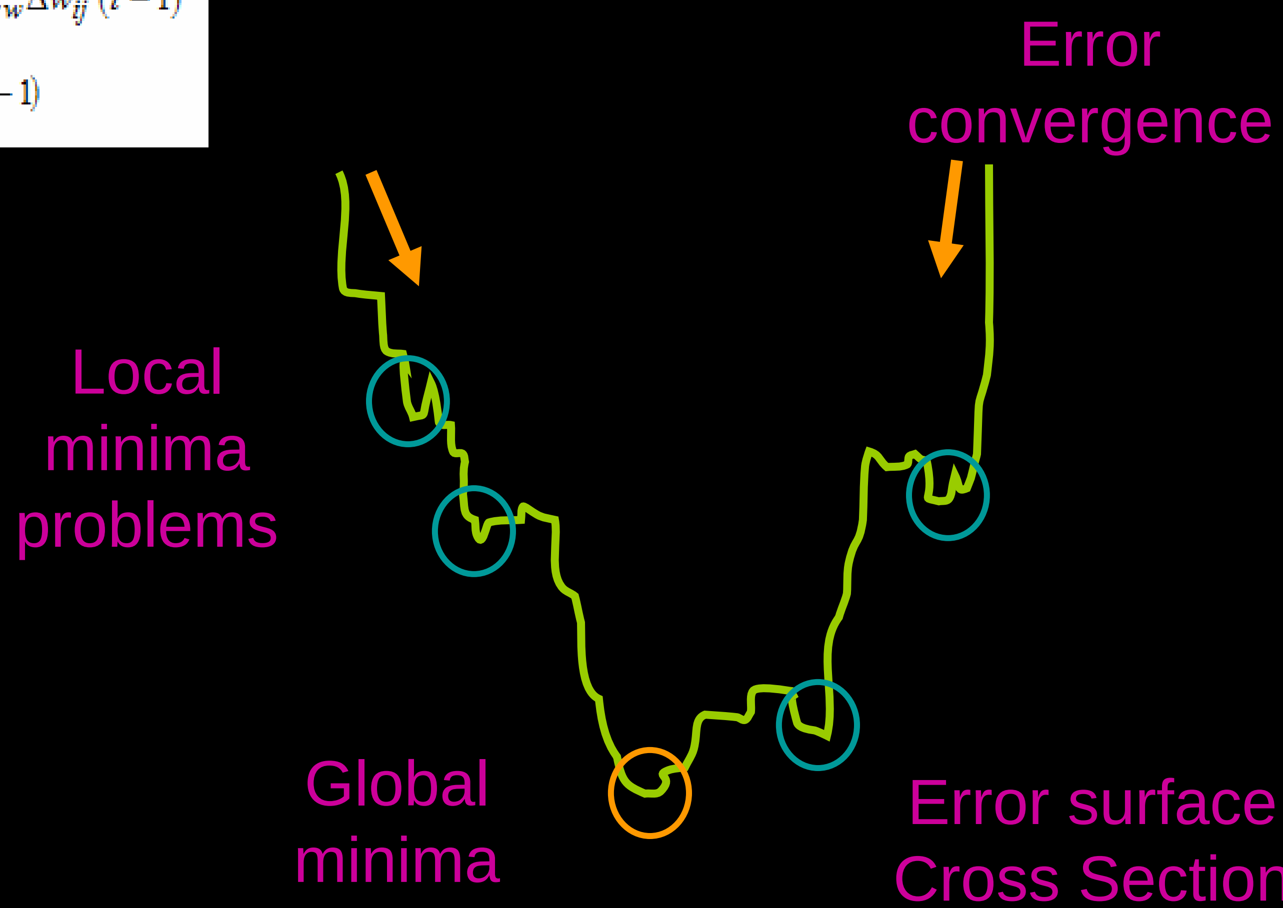
$$J = \frac{1}{2} \sum_i \left(y_i - \hat{y}_i \right)^2$$

ANN

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

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

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

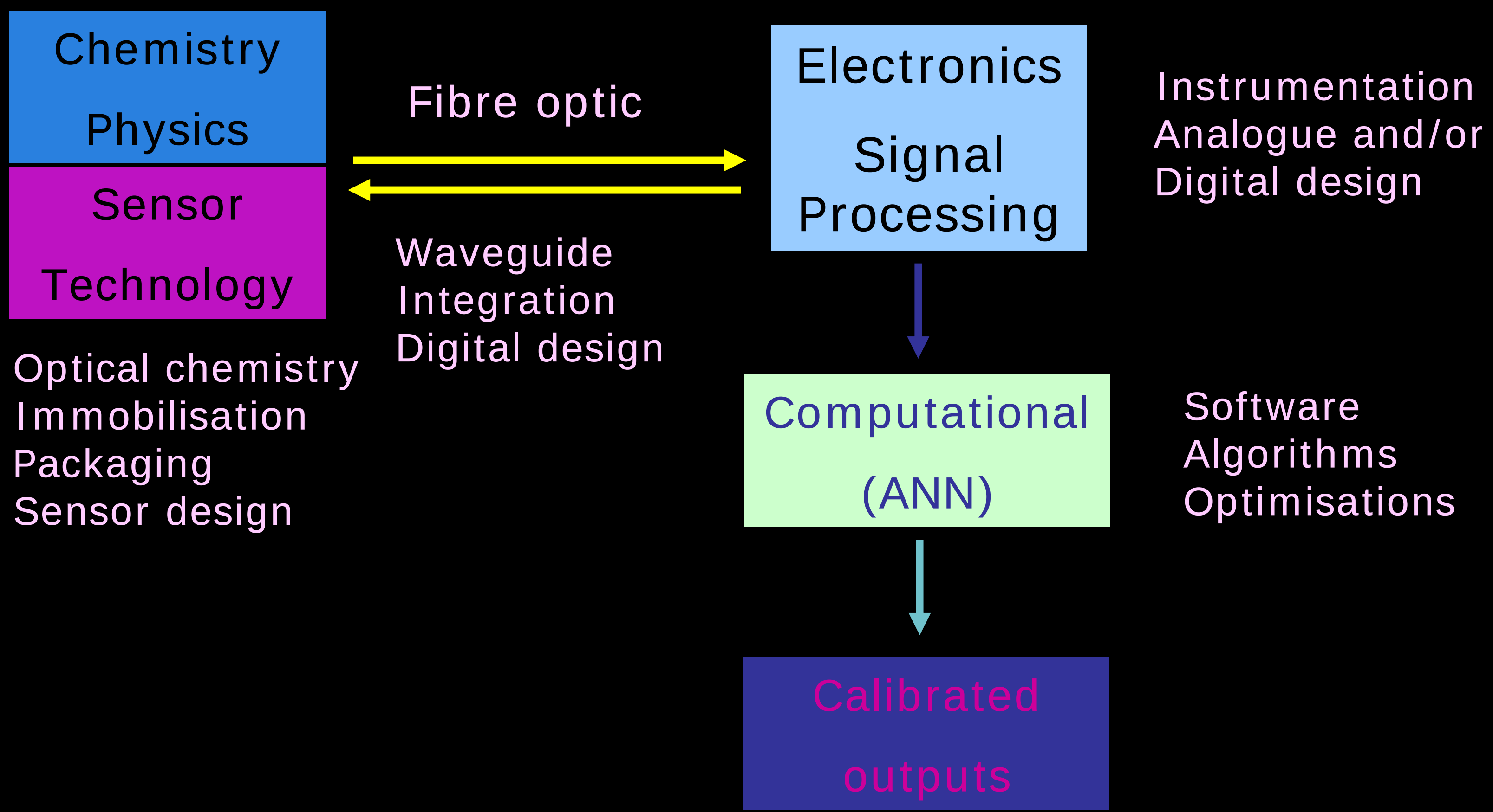


ANN

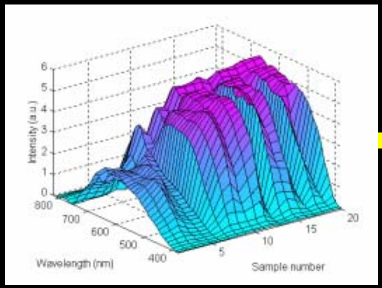
Adjusting a parameterized model structure to data typically is a non-convex problem and several local minima of the criterion function may exist.
This is one of the most pressing problem in non-linear identification, and calls for sophisticated initialization procedures.

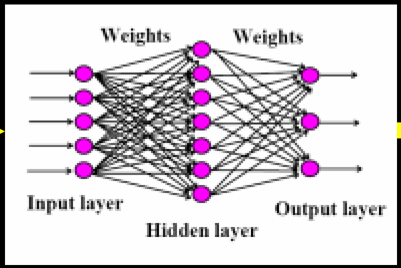
- In Neural Networks, some normalization is first applied to the data, and then a randomized initialization is made. Typically one will have to try several initializations
- Wavenets use an initialization based on fixed location and dilation parameters, which gives a linear regression
- For physical models, algebraic methods may produce linear regressions for initial estimates
- Block oriented models often employ several steps, fixing linear and/or nonlinear block to create smaller problems.

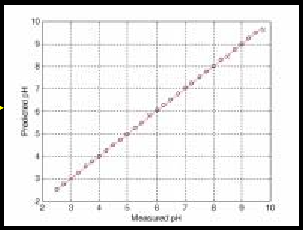
Application of ANN for OFCS



Application of ANN for OFCS

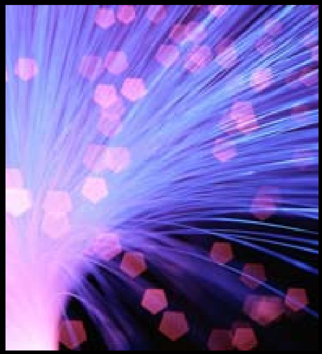




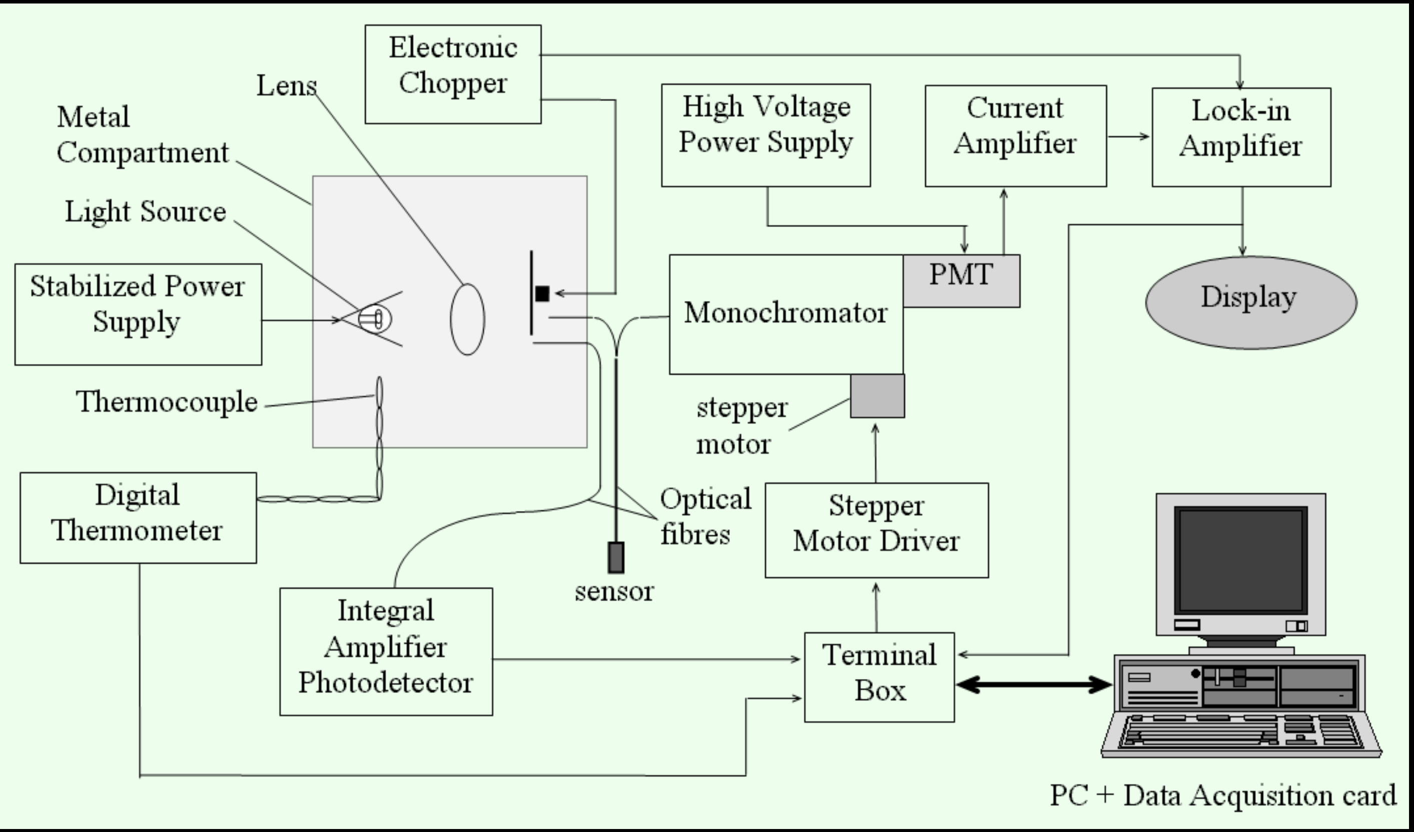


Optical Sensor

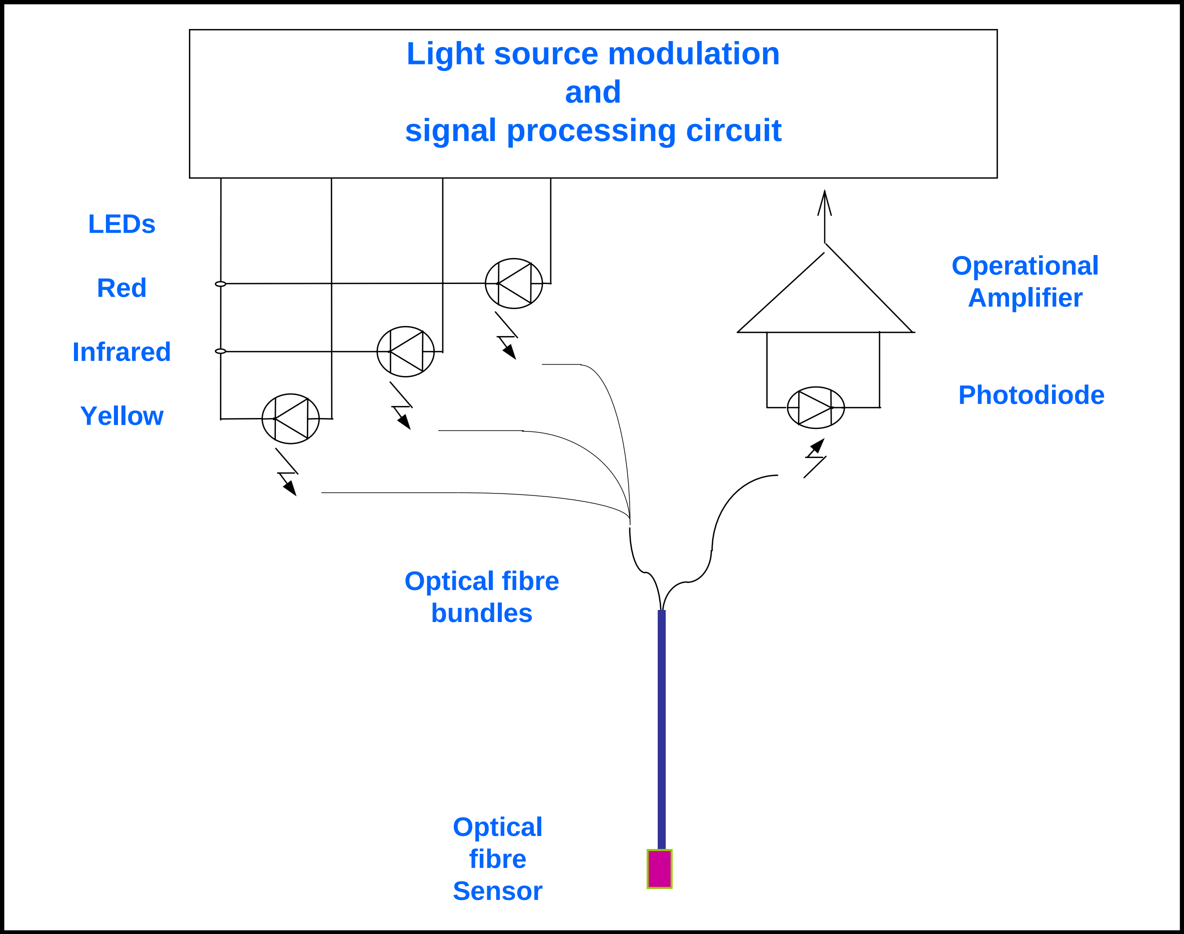
- Works based on changes of light properties such as reflection, dispersion, scattering, interference, absorption, refraction and diffraction
- Optical system measures one or more of: wavelength, amplitude, phase, polarization or color



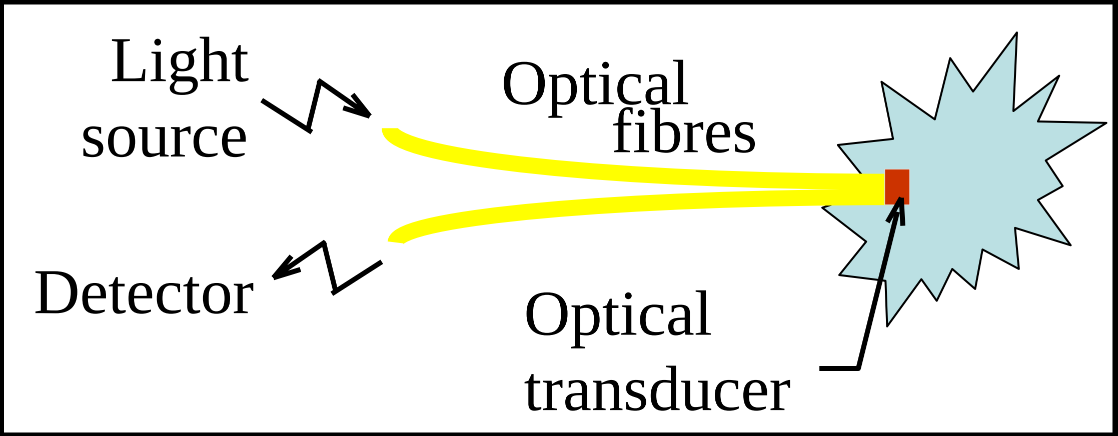
Optical Sensor - Measurement



Optical Sensor - Measurement

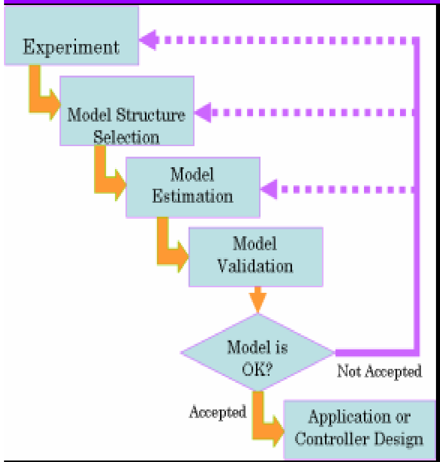


Optical Sensor - Measurement



Practical Guide: Sensor Modeling

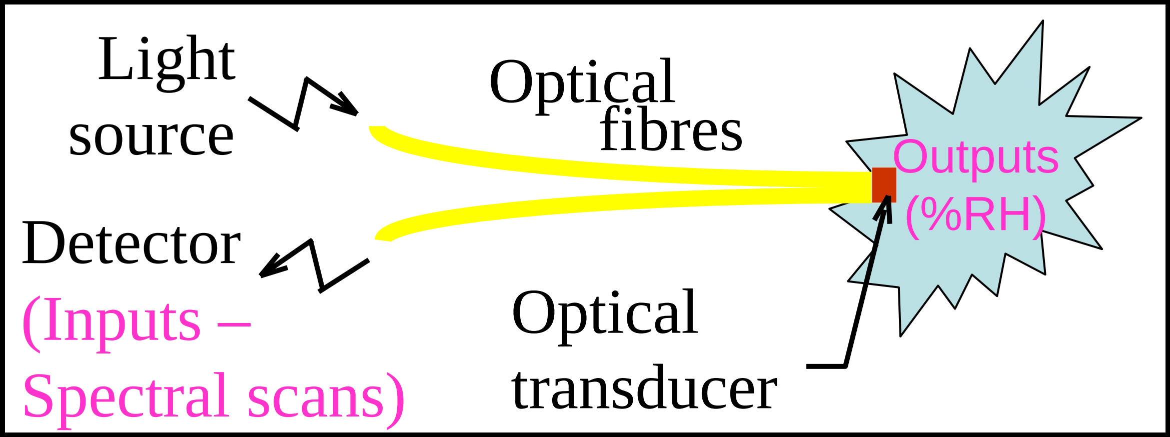
Apply Black Box System Identification for modelling Optical %RH Sensor

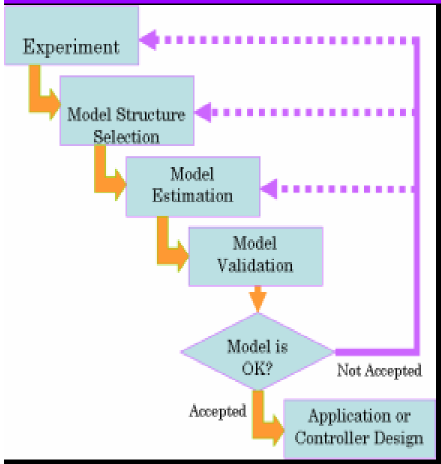


Practical Guide: Sensor Modeling

Experiment

Gather input-output data

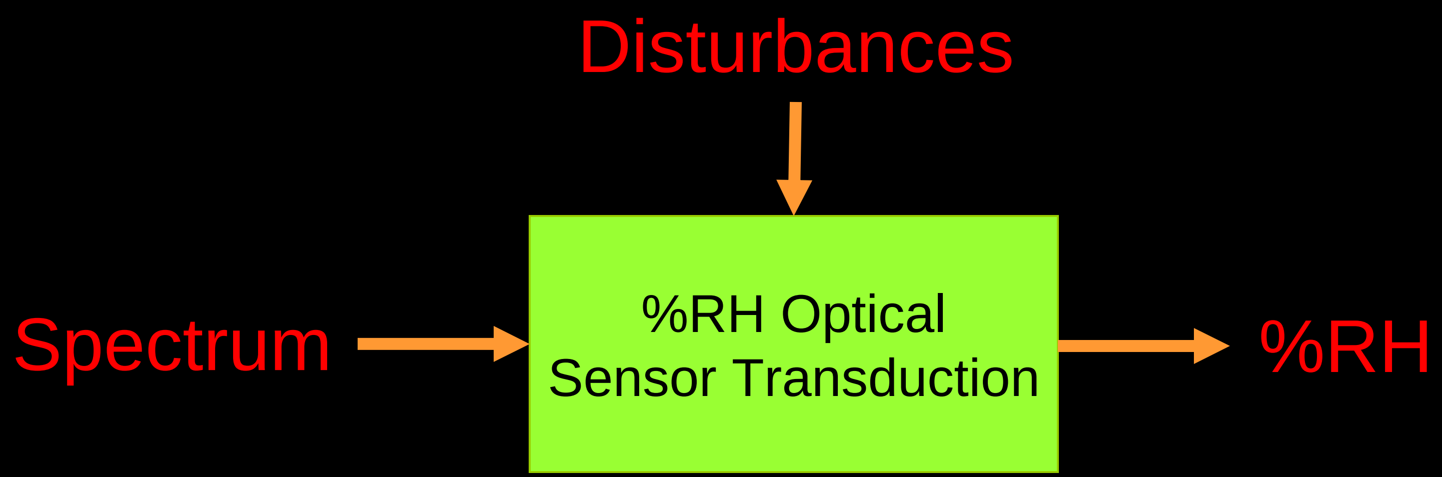


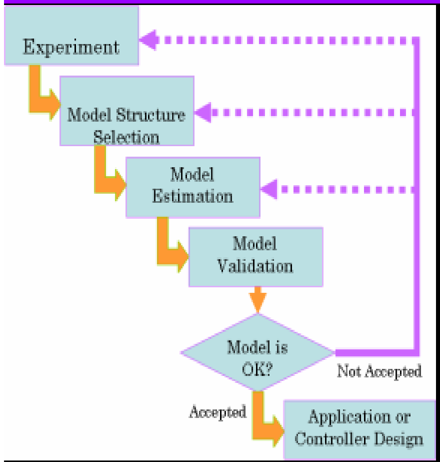


Practical Guide: Sensor Modeling

Experiment

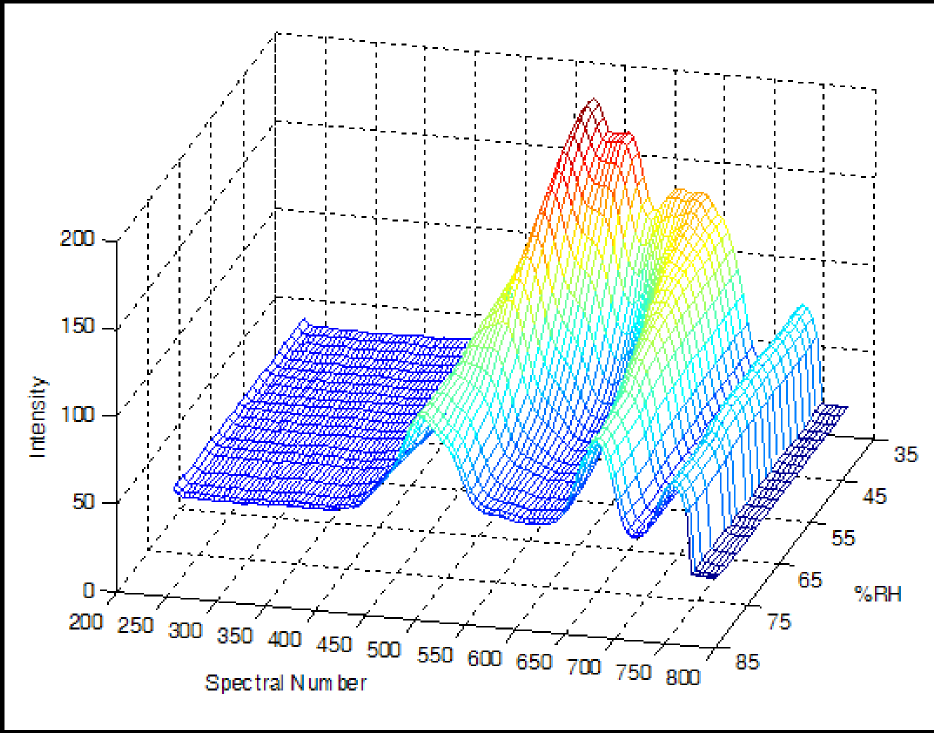
Gather input-output data





Practical Guide: Sensor Modeling

Experiment



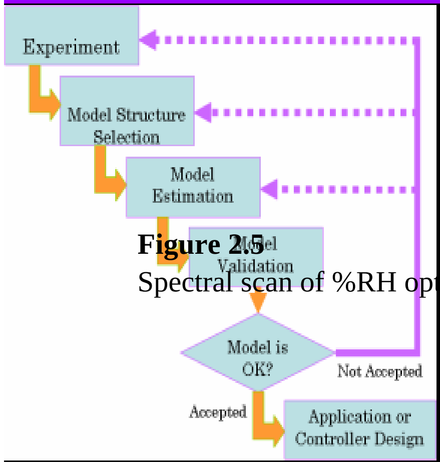
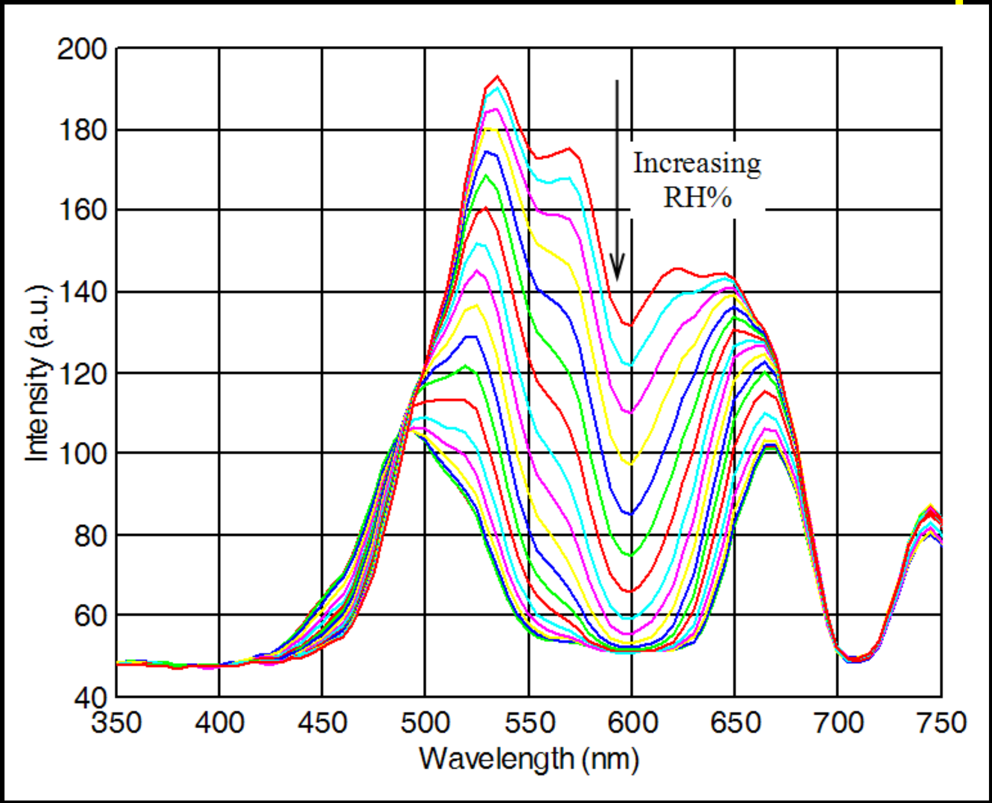
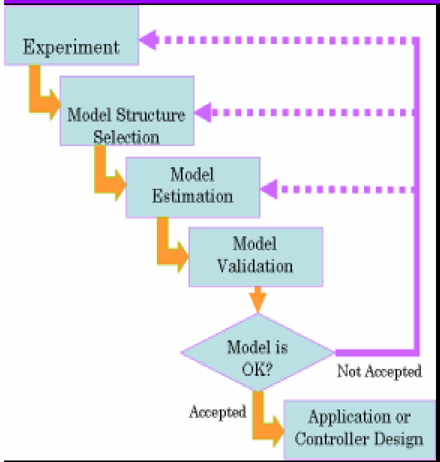


Figure 2.5
Spectral scan of %RH opt

Practical Guide: Sensor Modeling

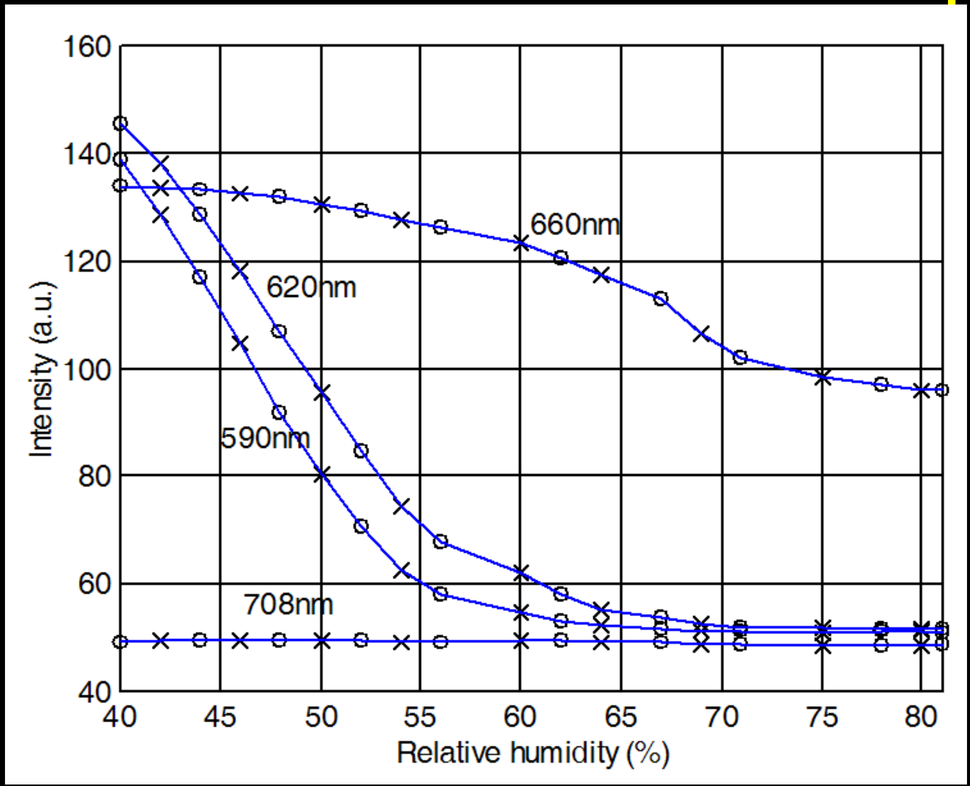
Data preprocessing

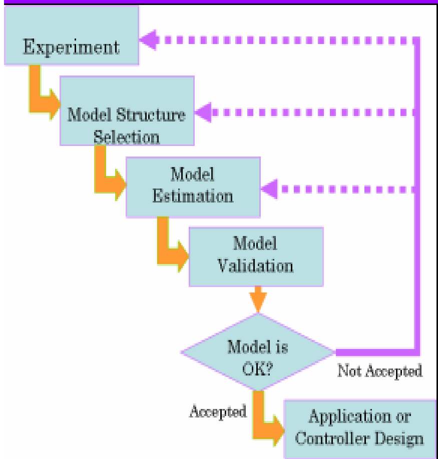




Practical Guide: Sensor Modeling

Data preprocessing

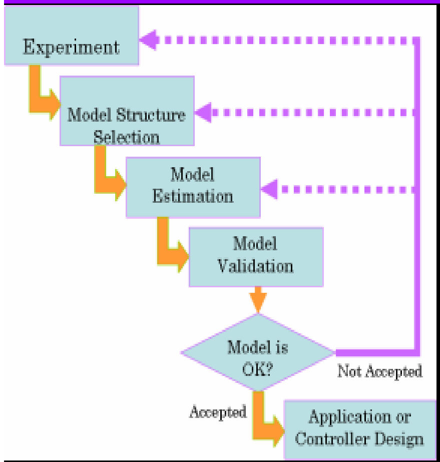




Practical Guide: Sensor Modeling

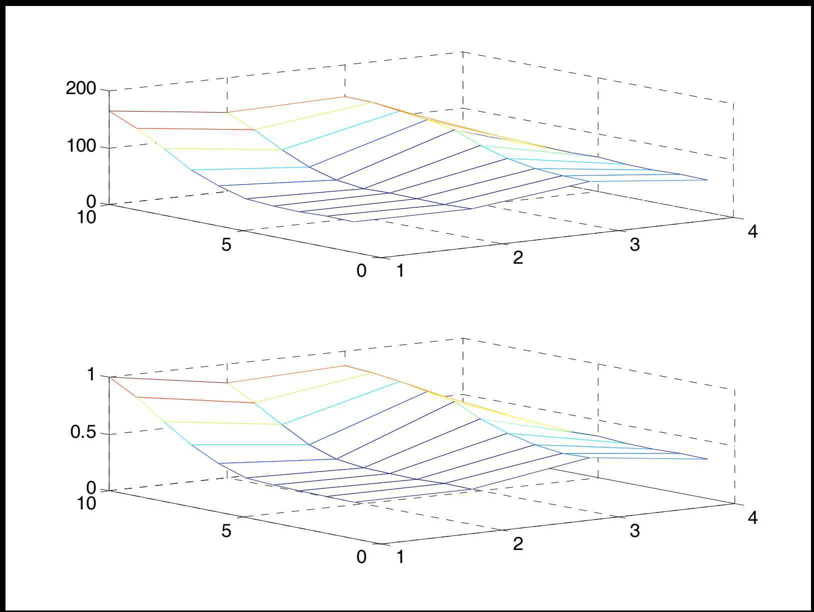
Data preprocessing

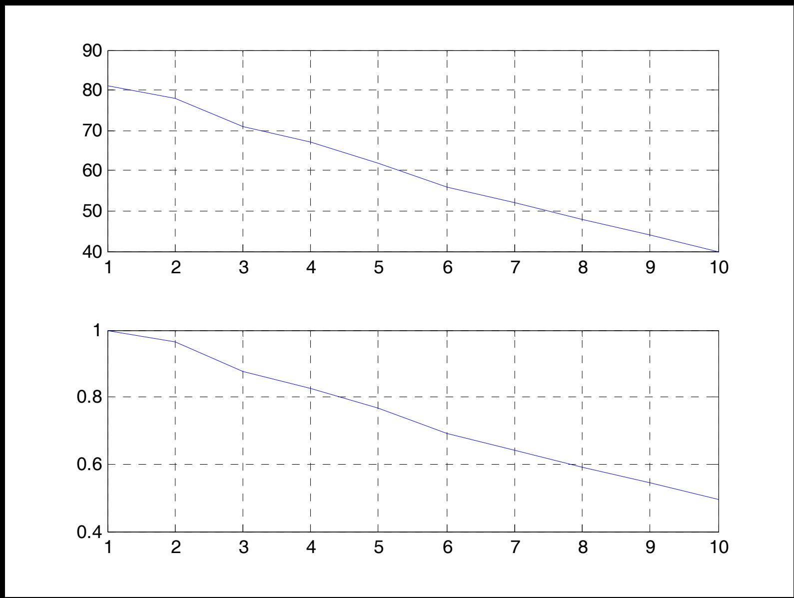
```
%load input data
x=load('trainInputs.m'); % input patterns
allx=load('AllInputs.m'); % all input patterns = trai
maxx=max(allx);maxx=max(maxx); % preprocess by scalin
x1=x/maxx;
y=load('trainTargets.m'); % targets
ally=load('AllTargets.m');
maxo=max(ally);maxo=max(maxo);
out_target=y/maxo;
```

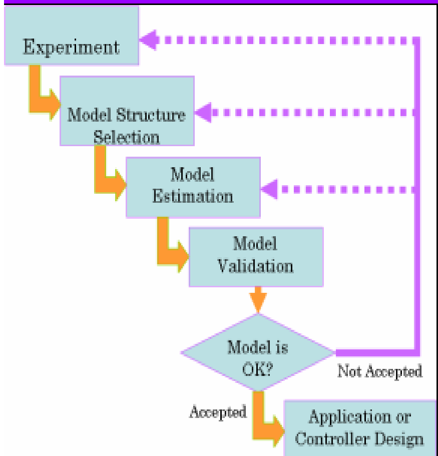


Practical Guide: Sensor Modeling

Data preprocessing





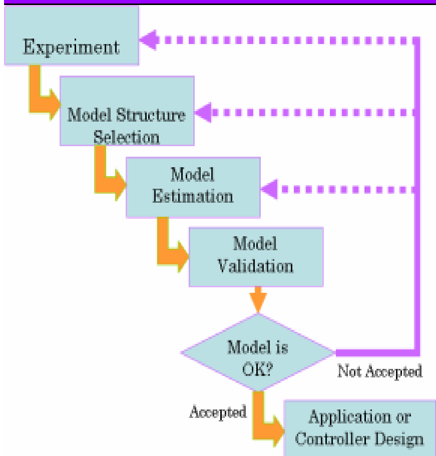


Practical Guide: Sensor Modeling

Model Structure

Define the ANN architecture

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

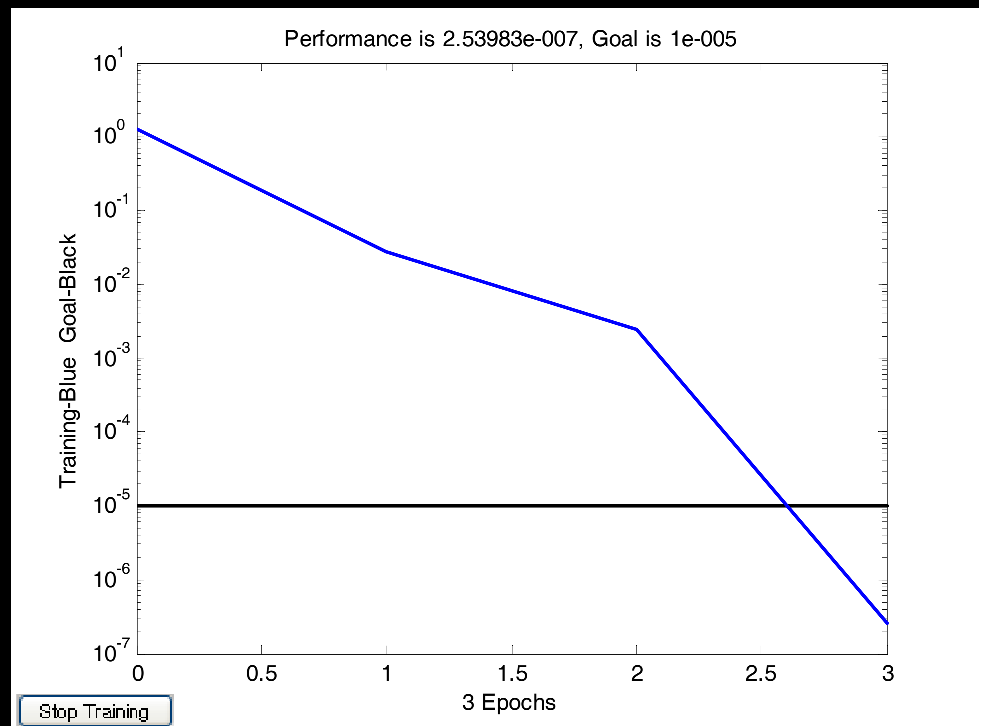


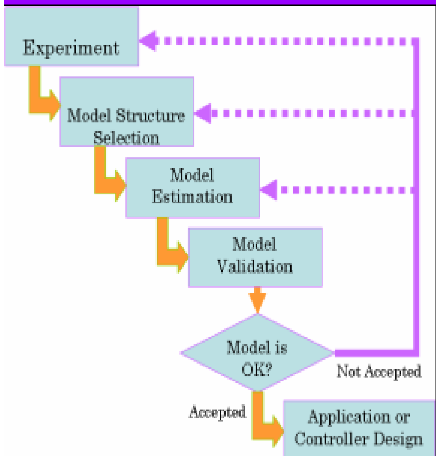
Practical Guide: Sensor Modeling

Model Estimation

ANN Training

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



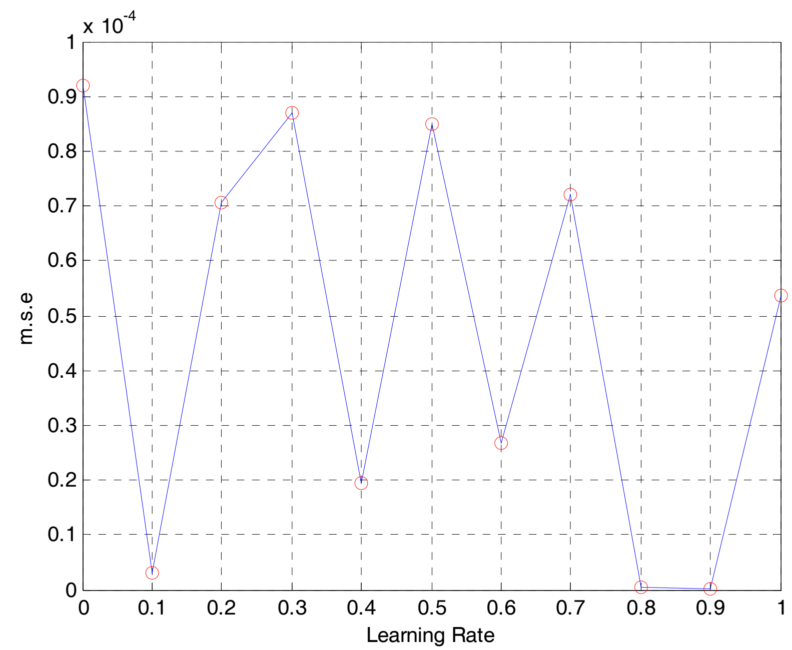


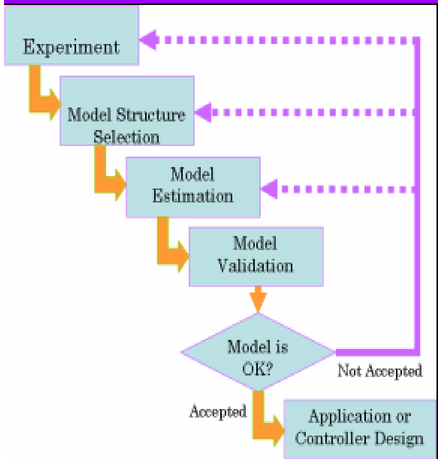
Practical Guide: Sensor Modeling

Model Estimation

ANN Training – optimise learning rate

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



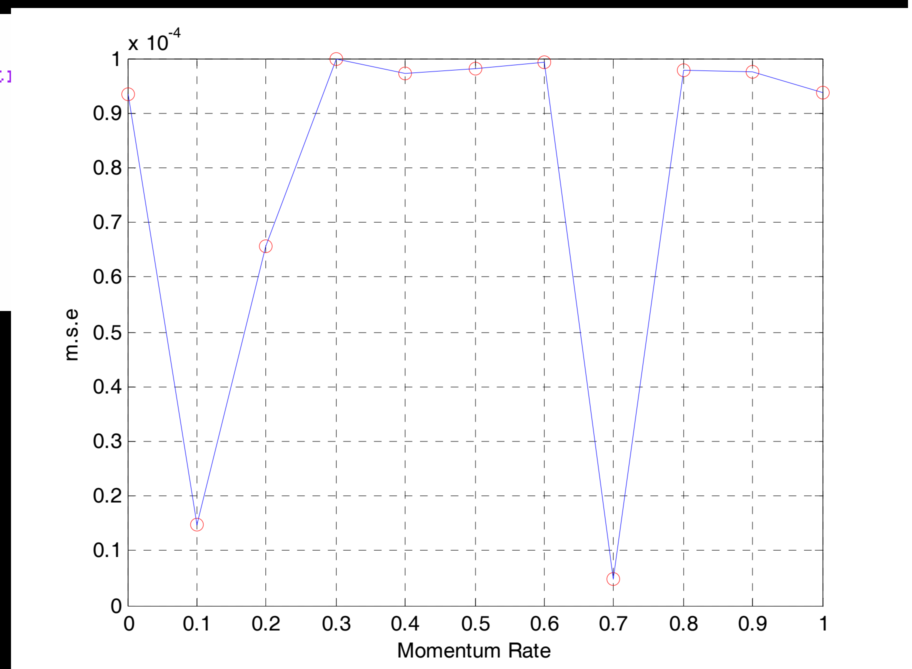


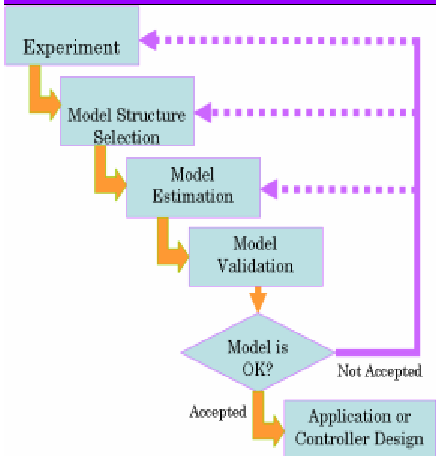
Practical Guide: Sensor Modeling

Model Estimation

ANN Training – optimise momentum

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



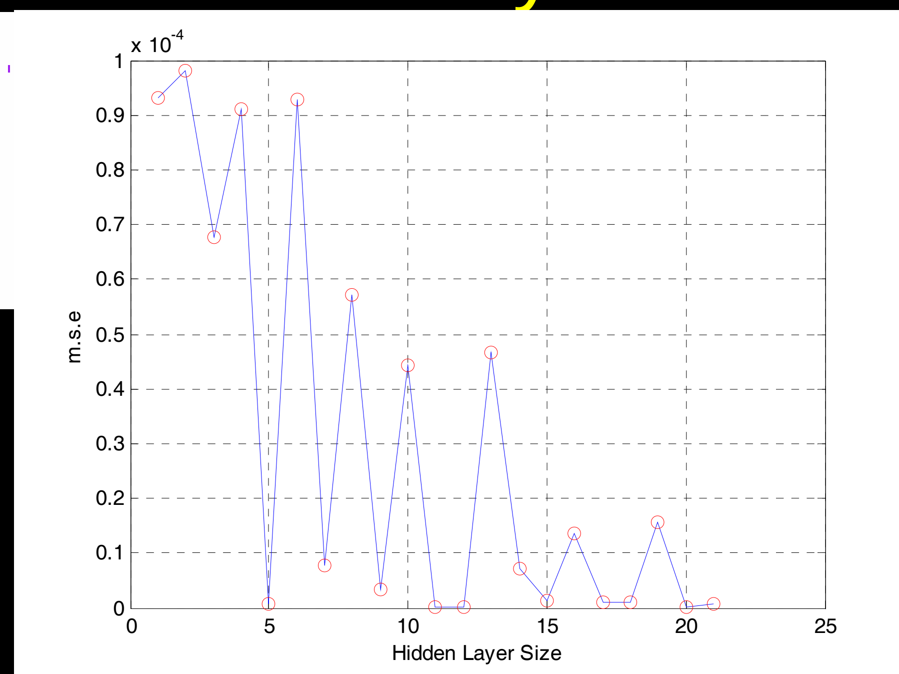


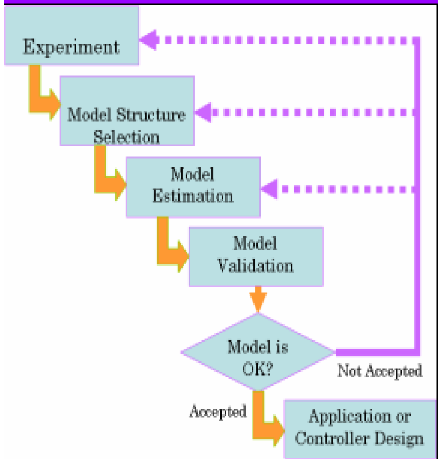
Practical Guide: Sensor Modeling

Model Estimation

ANN Training – optimise hidden layer

```
%network parameters
net=newff(minmax(x1),[10,1],{'tansig','purelin'},'
net.trainParam.show = 10;
net.trainParam.lr = 0.9;
net.trainParam.lr_inc = 0.001;
net.trainParam.mc = 0.7;
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.00001;
```



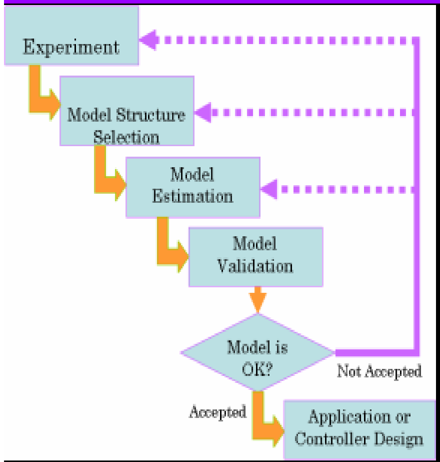


Practical Guide: Sensor Modeling

Model Validation

Check errors

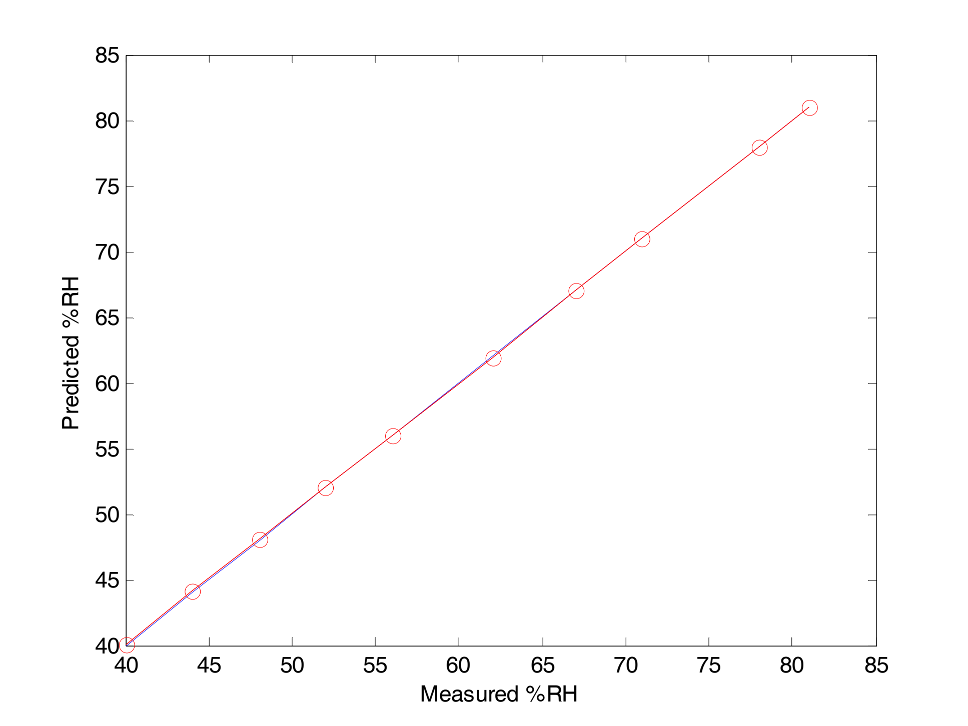
```
%check calculation of outputs for training data
simtrain=sim(net,x1); % forward pass
etrain=y1-simtrain; % calculate prediction errors
figure,plot(y,y1*maxo,'-b',y,simtrain*maxo,'-r');
hold,plot(y,simtrain*maxo,'-ro');
xlabel('Measured %RH')
ylabel('Predicted %RH');
grid
%save all data
save net1.mat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

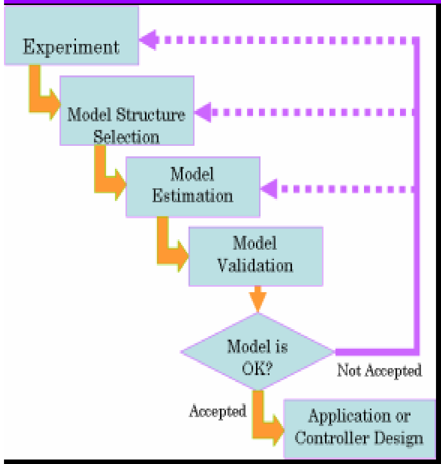


Practical Guide: Sensor Modeling

Model Validation

Check errors

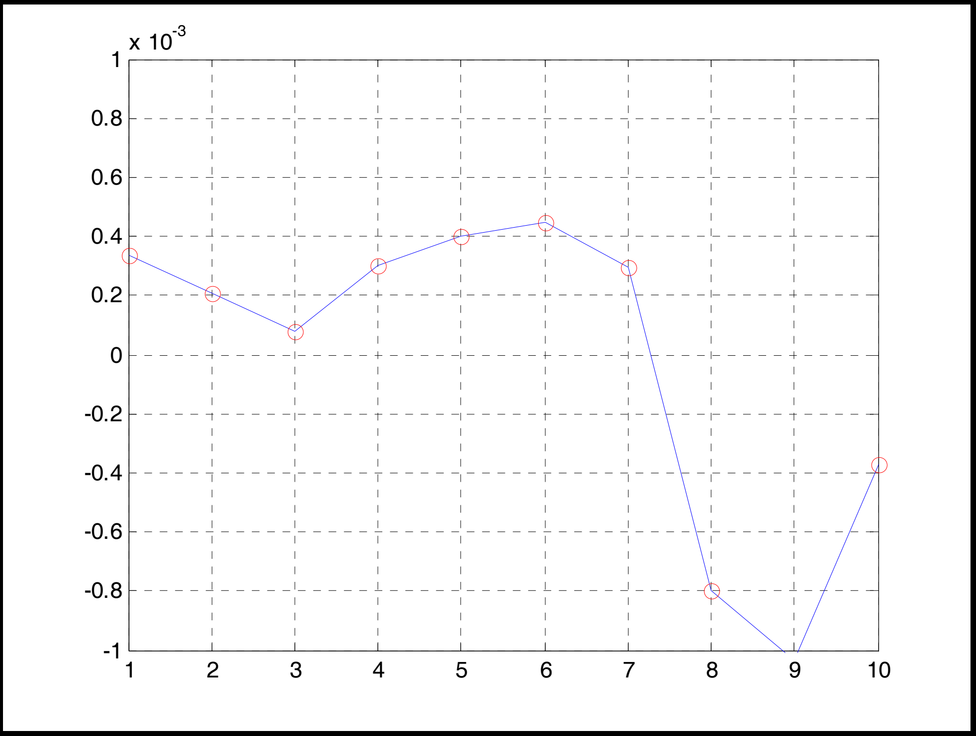




Practical Guide: Sensor Modeling

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