Lecture 13 - Handling Nonlinearity

- Nonlinearity issues in control practice
- Setpoint scheduling/feedforward
  - path planning replay - linear interpolation
- Nonlinear maps
  - B-splines
  - Multivariable interpolation: polynomials/splines/RBF
  - Neural Networks
  - Fuzzy logic
- Gain scheduling
- Local modeling

Nonlinearity in control practice

Here are the nonlinearities we already looked into

- Constraints - saturation in control
  - anti-windup in PID control
  - MPC handles the constraints
- Control program, path planning
- Static optimization
- Nonlinear dynamics
  - dynamic inversion
  - nonlinear IMC
  - nonlinear MPC

One additional nonlinearity in this lecture

- Controller gain scheduling

Dealing with nonlinear functions

- Analytical expressions
  - models are given by analytical formulas, computable as required
  - rarely sufficient in practice
- Models are computable off line
  - pre-compute simple approximation
  - on-line approximation
- Models contain data identified in the experiments
  - nonlinear maps
  - interpolation or look-up tables
- Advanced approximation methods
  - neural networks

Path planning

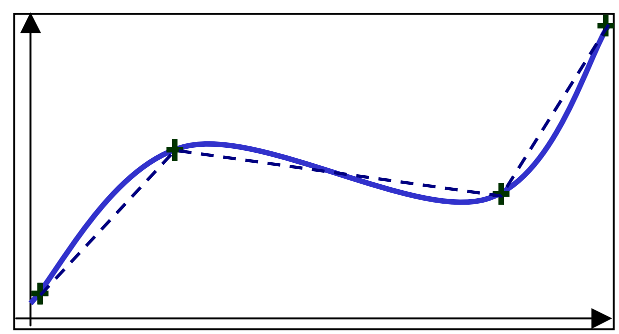
- Real-time replay of a pre-computed reference trajectory $y_d(t)$ or feedforward $v(t)$
- Reproduce a nonlinear function $y_d(t)$ in a control system

$$Y = \begin{bmatrix} Y_1 = y_d(\theta_1) \\ Y_2 = y_d(\theta_2) \\ \vdots \\ Y_n = y_d(\theta_n) \end{bmatrix}, \quad \Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Code:

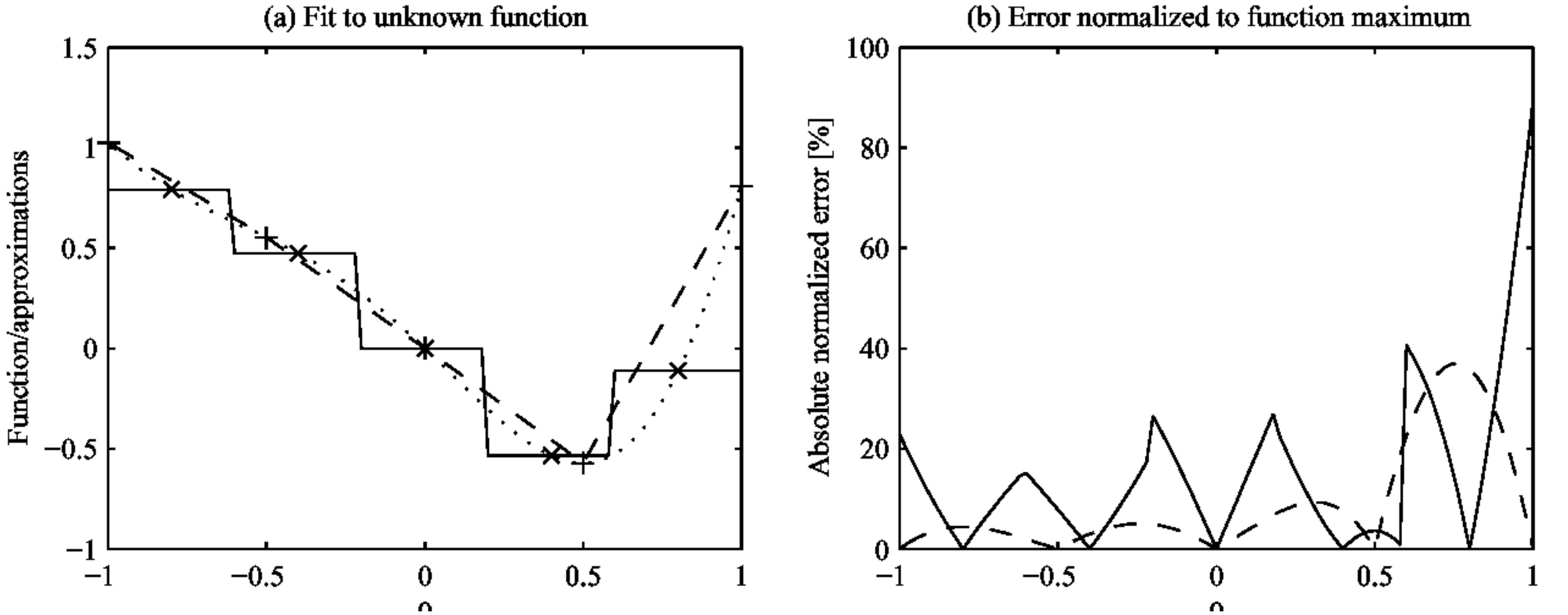
1. Find j, such that $\theta_j \le t \le \theta_{j+1}$
2. Compute

$$y_d(t) = Y_j \frac{\theta_{j+1} - t}{\theta_{j+1} - \theta_j} + Y_{j+1} \frac{t - \theta_j}{\theta_{j+1} - \theta_j}$$



Linear interpolation vs. table look-up

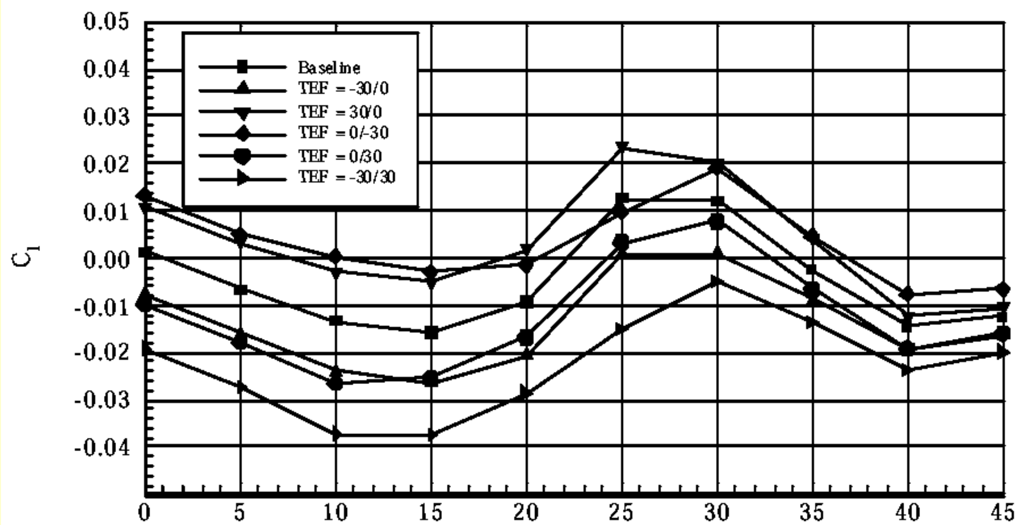
- linear interpolation is more accurate
- requires less data storage
- simple computation



Empirical models

- Aerospace - most developed nonlinear approaches
  - automotive and process control have second place
- Aerodynamic tables
- Engine maps
  - jet turbines
  - automotive
- Process maps, e.g., in semiconductor manufacturing
- Empirical map for a attenuation vs. temperature in an optical fiber

Example

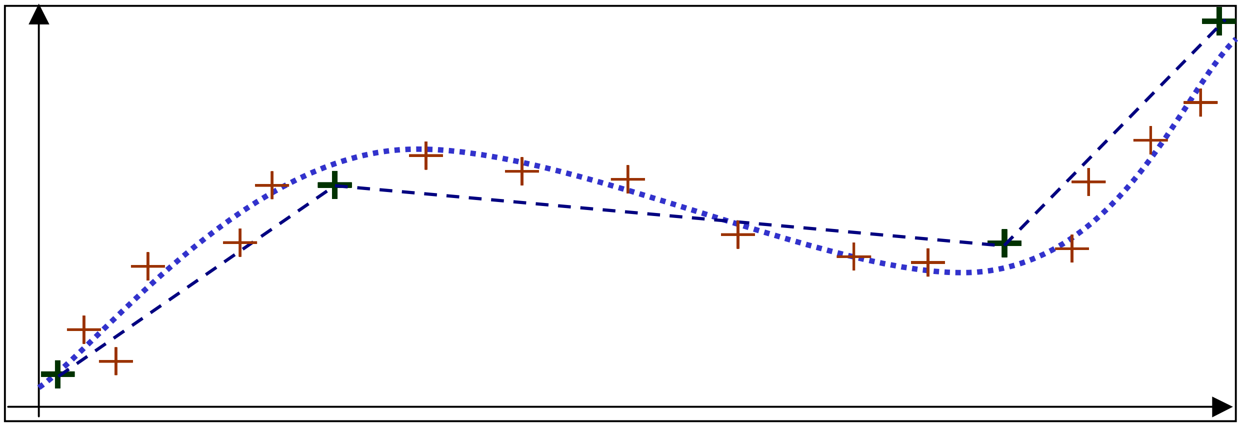


Rolling-Moment Coefficient as a Function of Angle of Attack with TEF Deflections, $\beta = 10°$.

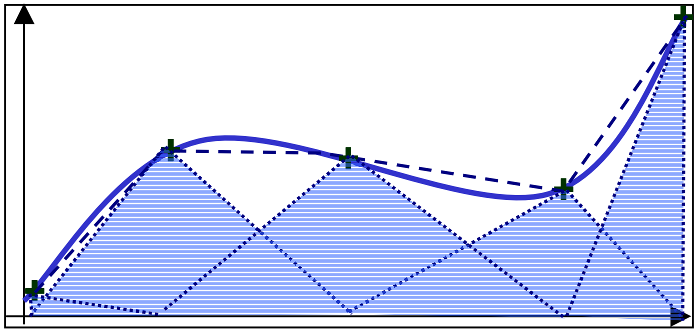
TEF=Trailing Edge Flap

Approximation

- Interpolation:
  - compute function that will provide given values Y_j in the nodes θ_j
  - not concerned with accuracy in-between the nodes
- Approximation
  - compute function that closely corresponds to given data, possibly with some error
  - might provide better accuracy throughout

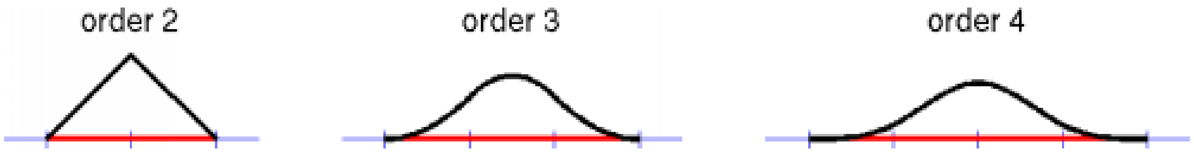


B-spline interpolation



- 1st-order
  - look-up table, nearest neighbor
- 2nd-order
  - linear interpolation

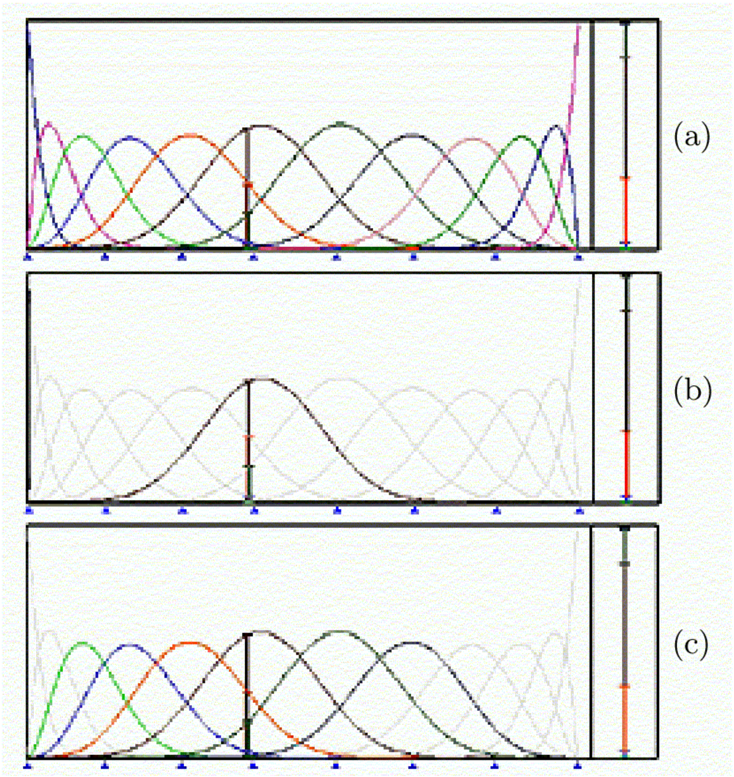
$$y_d(t) = \sum_j Y_j B_j(t)$$



- n-th order:
  - Piece-wise *n*-th order polynomials, matched *n*-2 derivatives
  - zero outside a local support interval
  - support interval extends to *n* nearest neighbors

B-splines

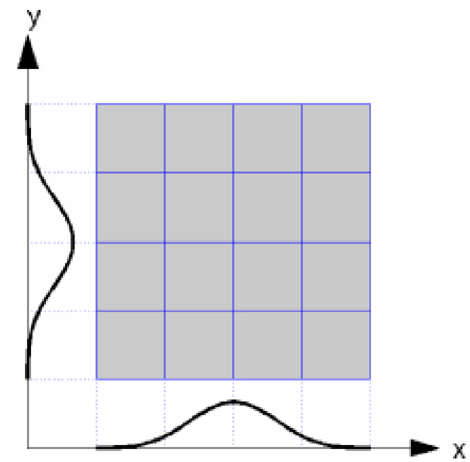
- Accurate interpolation of smooth functions with relative few nodes
- For 1-D function the gain from using high-order B-splines is not worth an added complexity
- Introduced and developed in CAD for 2-D and 3-D curve and surface data
- Are used for defining multidimensional nonlinear maps

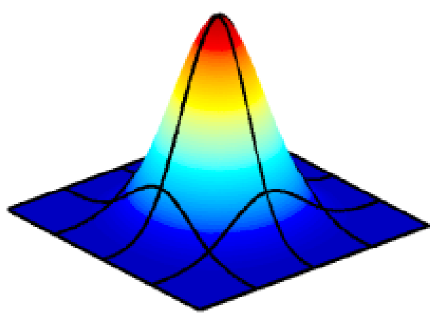


Multivariable B-splines

- Regular grid in multiple variables
- Tensor product B-splines
- Used as a basis of finite-element models

$$y(u,v) = \sum_{j,k} w_{j,k} B_j(u) B_k(v)$$





Linear regression for nonlinear map

- Linear regression

$$y(\bar{x}) = \sum_j \theta_j \varphi_j(\bar{x}) = \theta^T \cdot \phi(\bar{x}) \qquad \bar{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Multidimensional B-splines
- Multivariate polynomials

$$\varphi_j(x_1, \ldots, x_n) = (x_1)^{k_1} \cdot \ldots \cdot (x_n)^{k_n}$$

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 (x_1)^2 + \theta_4 x_1 x_2 + \ldots$$

- RBF - Radial Basis Functions

$$\varphi_j(\bar{x}) = R\left(\left\|\bar{x} - \bar{c}_j\right\|\right) = e^{-a\left\|\bar{x} - \bar{c}_j\right\|^2}$$

Linear regression approximation

- Nonlinear map data
  - available at scattered nodal points

$$Y = \left[y^{(1)} \quad \ldots \quad y^{(N)} \right]$$

$$\bar{x}^{(1)} \quad \ldots \quad \bar{x}^{(N)}$$

- Linear regression map

$$Y = \theta^T \cdot \left[\phi(\bar{x}^{(1)}) \quad \ldots \quad \phi(\bar{x}^{(N)}) \right] = \theta^T \Phi$$

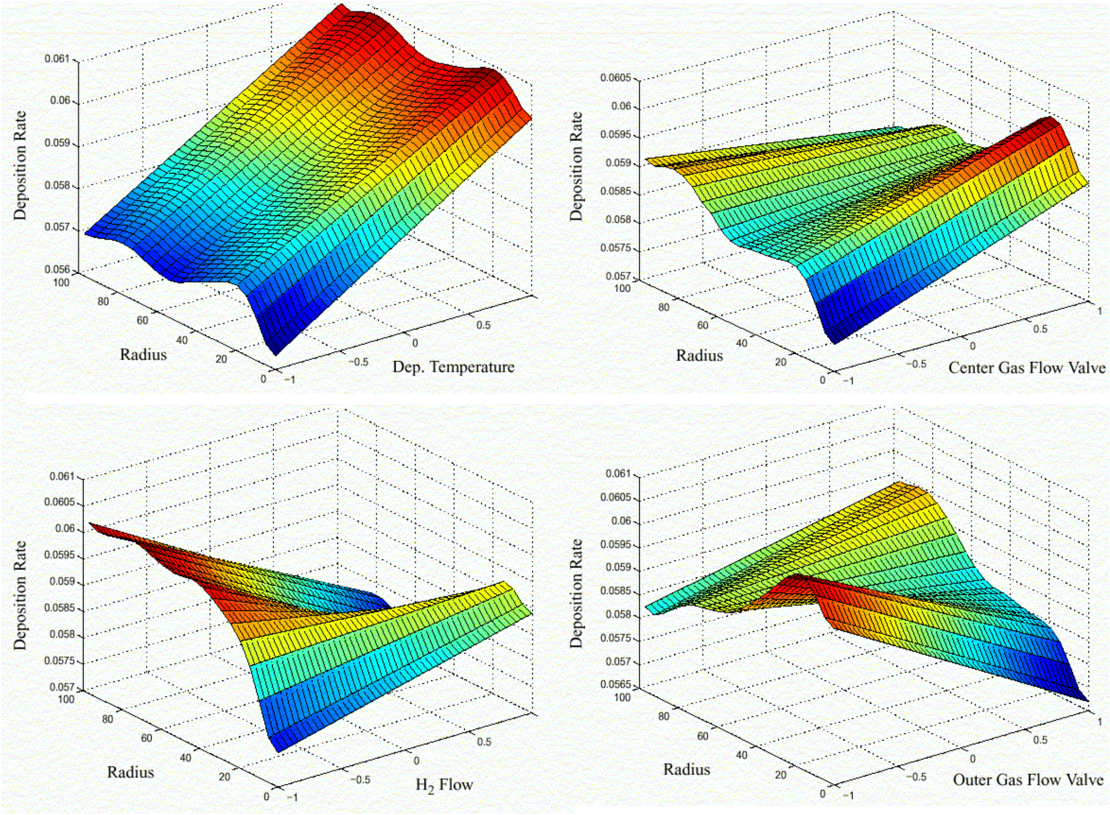
- Linear regression approximation
  - regularized least square estimate of the weight vector

$$\hat{\theta} = \left(\Phi \Phi^T + rI \right)^{-1} \Phi Y^T$$

- Works just the same for vector-valued data!

Nonlinear map example - Epi

- Epitaxial growth (semiconductor process)
  - process map for run-to-run control



Linear regression for Epi map

- Linear regression model for epitaxial grouth

$$y = c_0 x_1 p_1(x_2) + c_1(1 - x_1) p_2(x_2)$$

$$p_1 = w_0 + w_1 x_2 + w_3 (x_2)^2 + w_4 (x_2)^3$$

$$c_0 x_1 p_1 = \underbrace{w_0 c_0}_{\theta_1} x_1 + \underbrace{w_1 c_0}_{\theta_2} x_1 x_2 + \underbrace{w_3 c_0}_{\theta_3} x_1 (x_2)^2 + \underbrace{w_4 c_0}_{\theta_4} x_1 (x_2)^3$$

$$c_1(1 - x_1) p_2(x_2) =$$

$$\underbrace{v_0 c_1}_{\theta_5}(1 - x_1) + \underbrace{v_1 c_1}_{\theta_6}(1 - x_1) x_2 + \underbrace{v_3 c_0}_{\theta_7}(1 - x_1)(x_2)^2 + \underbrace{w_4 c_0}_{\theta_8}(1 - x_1)(x_2)^3$$

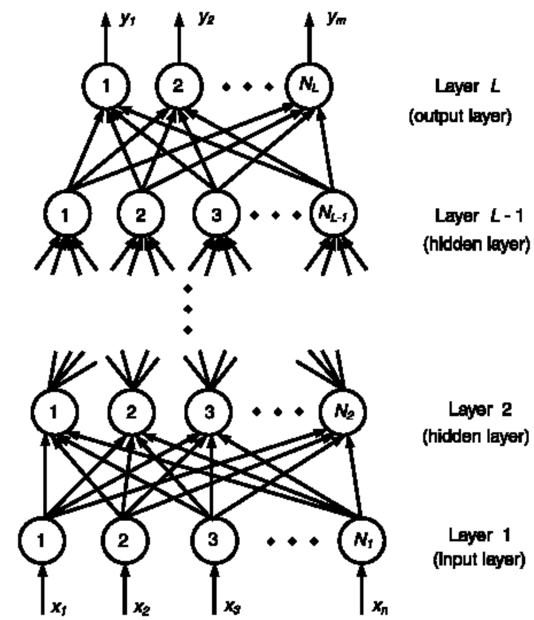
$$y(x_1, x_2) = \sum_j \theta_j \varphi_j (x_1, x_2) = \theta^T \cdot \phi(x_1, x_2)$$

Neural Networks

- Any nonlinear approximator might be called a Neural Network
  - RBF Neural Network
  - Polynomial Neural Network
  - B-spline Neural Network
  - Wavelet Neural Network

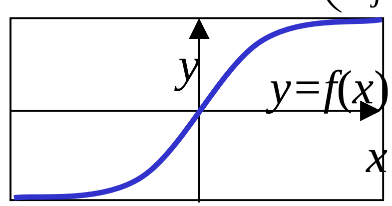
Linear in parameters

- MPL - Multilayered Perceptron
  - Nonlinear in parameters
  - Works for many inputs



$$y(\overline{x}) = w_{1,0} + f\left(\sum_j w_{1,j} y_j^1\right), y_j^1 = w_{2,0} + f\left(\sum_j w_{2,j} x_j\right)$$

$$f(x) = \frac{1}{1+e^{-x}}$$



Multi-Layered Perceptrons

- Network parameter computation
  - training data set
  - parameter identification

$$y(\bar{x}) = F(\bar{x};\theta)$$

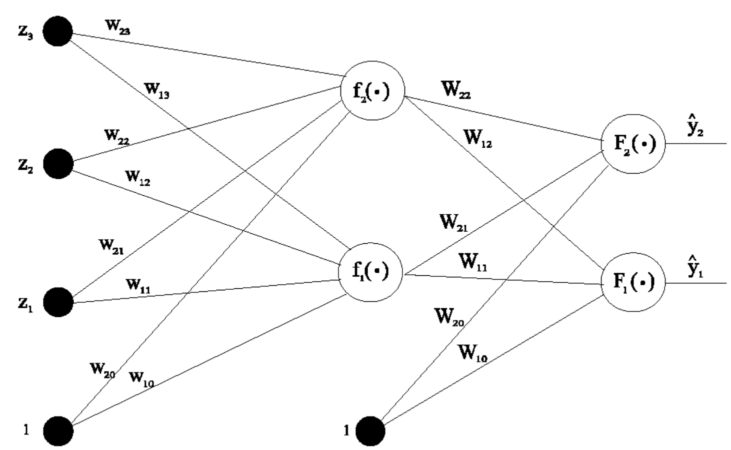
- Noninear LS problem

$$V = \sum_j \left\| y^{(j)} - F(\bar{x}^{(j)};\theta) \right\|^2 \rightarrow \min$$

- Iterative NLS optimization
  - Levenberg-Marquardt
- Backpropagation
  - variation of a gradient descent

$$Y = \left[y^{(1)} \quad \dots \quad y^{(N)} \right]$$

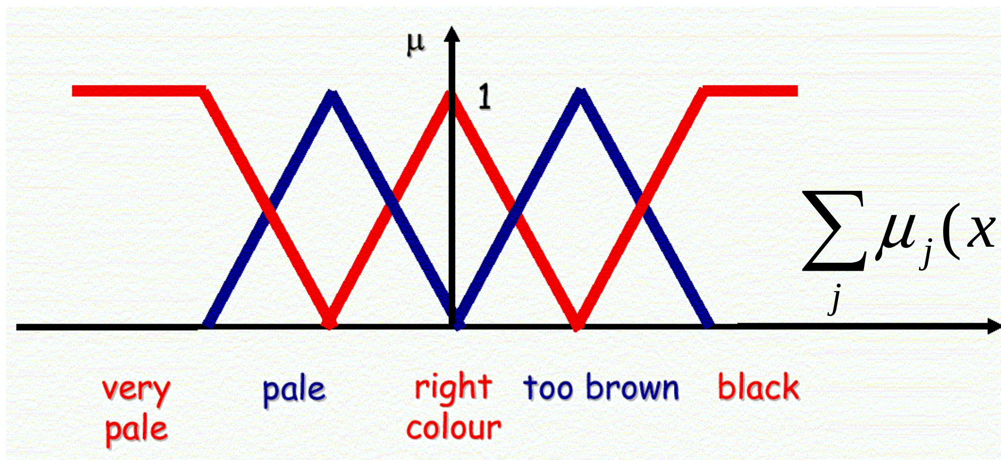
$$\bar{x}^{(1)} \quad \dots \quad \bar{x}^{(N)}$$



Fuzzy Logic

- Function defined at nodes. Interpolation scheme
- Fuzzyfication/de-fuzzyfication = interpolation
- Linear interpolation in 1-D

$$y(x) = \frac{\sum_j y_j \mu_j(x)}{\sum_j \mu_j(x)}$$

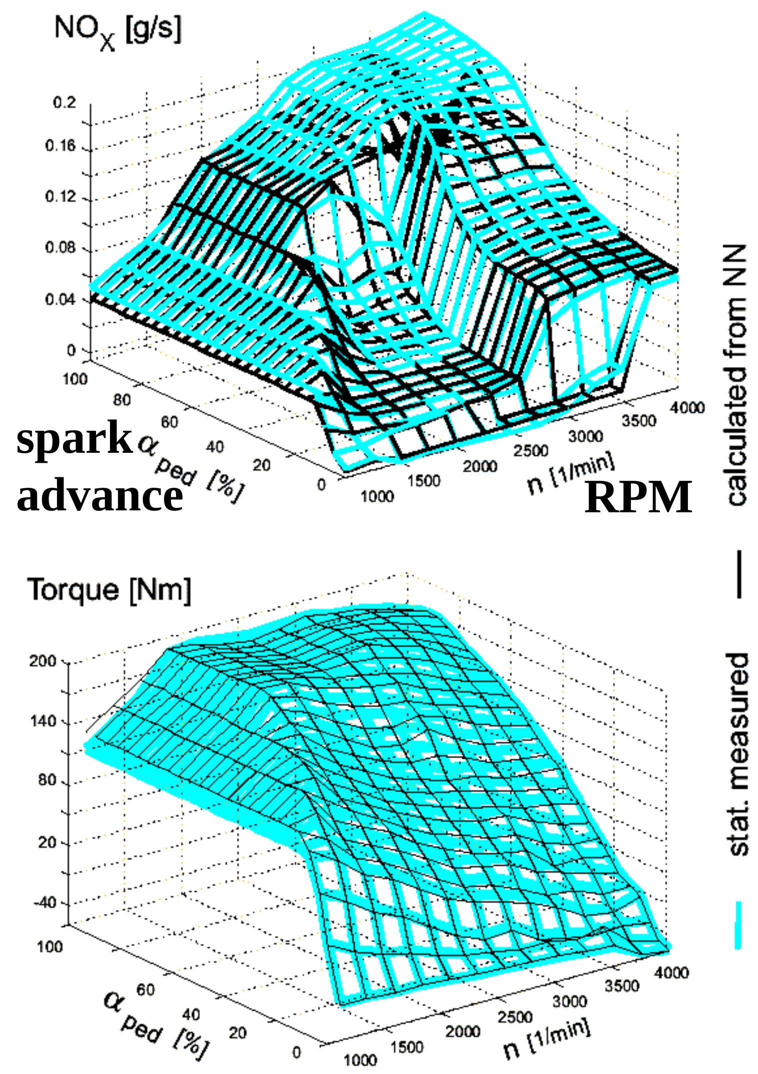


$$\sum_j \mu_j(x) = 1$$

- Marketing (communication) and social value
- Computer science: emphasis on interaction with a user
  - EE - emphasis on mathematical computations

Neural Net application

- Internal Combustion Engine maps
- Experimental map:
  - data collected in a steady state regime for various combination of parameters
  - 2-D table
- NN map
  - approximation of the experimental map
  - MLP was used in this example
  - works better for a smooth surface

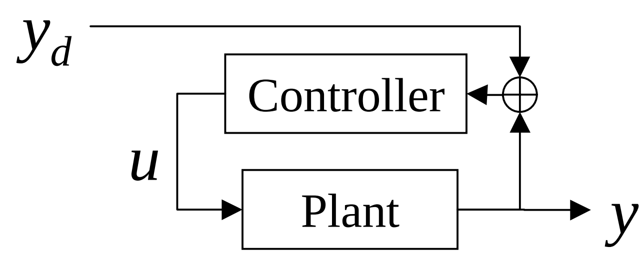


Linear feedback in a nonlinear plant

- Simple example

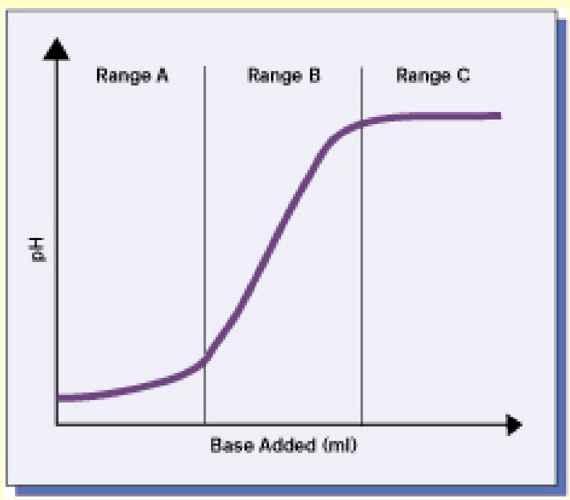
  $y = f(x) + g(x)u$

  $u = -k(x)(y - y_d) + u_{ff}(x)$



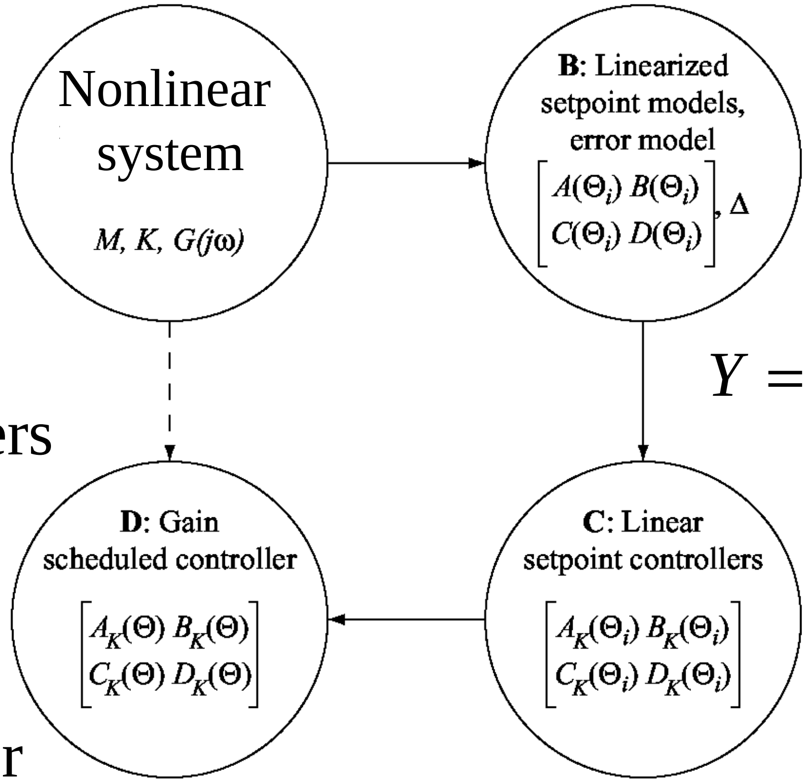
- Control design requires $k(x), u_{ff}(x), y_d(x)$
- These variables are *scheduled* on x

Example: varying process gain



Gain scheduling

- Single out several regimes - model linearization or experiments
- Design linear controllers in these regimes: setpoint, feedback, feedforward
- Approximate controller dependence on the regime parameters



$$\begin{bmatrix} A(\Theta_i) & B(\Theta_i) \\ C(\Theta_i) & D(\Theta_i) \end{bmatrix}, \Delta$$

$$\begin{bmatrix} A_K(\Theta_i) & B_K(\Theta_i) \\ C_K(\Theta_i) & D_K(\Theta_i) \end{bmatrix}$$

$$\begin{bmatrix} A_K(\Theta) & B_K(\Theta) \\ C_K(\Theta) & D_K(\Theta) \end{bmatrix}$$

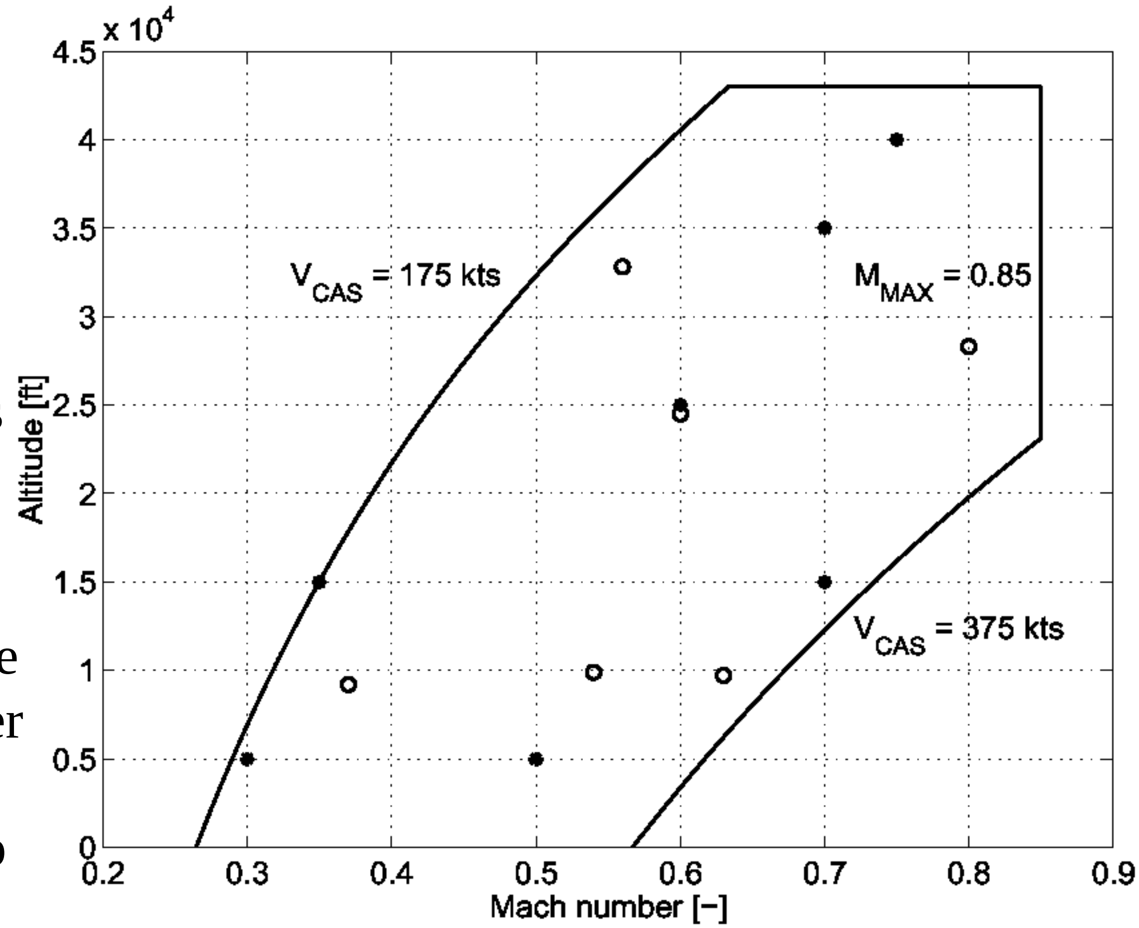
$$Y = \begin{bmatrix} \mathrm{vec}(A) \\ \mathrm{vec}(B) \\ \mathrm{vec}(C) \\ \mathrm{vec}(D) \end{bmatrix}$$

Linear interpolation:

$$Y(\Theta) = \sum_j Y_j \varphi_j(\Theta)$$

Gain scheduling - example

- Flight control
- Flight envelope parameters are used for scheduling
- Shown
  - Approximation nodes
  - Evaluation points
- Key assumption
  - Attitude and Mach are changing much slower than time constant of the flight control loop



Local Modeling Based on Data

- Data mining in the loop
- Honeywell product

Heat Loads