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

\[
\begin{align*}
    \text{Path planner,} \\
    \text{data arrays } Y, \Theta
\end{align*}
\]

\[
\begin{align*}
    t \rightarrow & \quad y_d(t) \\
    & \quad Y = \begin{bmatrix}
        Y_1 = y_d(\theta_1) \\
        Y_2 = y_d(\theta_2) \\
        \vdots \\
        Y_n = y_d(\theta_n)
    \end{bmatrix}, \\
    & \quad \Theta = \begin{bmatrix}
        \theta_1 \\
        \theta_2 \\
        \vdots \\
        \theta_n
    \end{bmatrix}
\end{align*}
\]

Code:
1. Find $j$, such that $\theta_j \leq t \leq \theta_{j+1}$
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, $\beta = 10^\circ$.

TEF=Trailing Edge Flap
Approximation

• Interpolation:
  – compute function that will provide given values $Y_j$ in the nodes $\theta_j$
  – 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
  \[ y(\bar{x}) = \sum_j \theta_j \varphi_j(\bar{x}) = \theta^T \cdot \phi(\bar{x}) \]
  \[ \bar{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \]

• 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
  \[ \varphi_j(\bar{x}) = R(\lVert \bar{x} - \bar{c}_j \rVert) = e^{-a \lVert \bar{x} - \bar{c}_j \rVert^2} \]
Linear regression approximation

- Nonlinear map data
  - available at scattered nodal points

- Linear regression map

\[ Y = \theta^T \cdot [\phi(x^{(1)}) \ldots \phi(x^{(N)})] = \theta^T \Phi \]

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

\[ \hat{\theta} = (\Phi \Phi^T + rI)^{-1} \Phi Y^T \]

- 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 growth

\[ y = c_0 x_1 p_1(x_2) + c_1 (1-x_1) p_2(x_2) \]

\[ p_1 = w_0 + w_1 x_2 + w_3 (x_2)^2 + w_4 (x_2)^3 \]

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

\[ c_1 (1-x_1) p_2(x_2) = \underbrace{v_0 c_1 (1-x_1)}_{\theta_5} + \underbrace{v_1 c_1 (1-x_1) x_2}_{\theta_6} + \underbrace{v_3 c_0 (1-x_1)(x_2)^2}_{\theta_7} + \underbrace{w_4 c_0 (1-x_1)(x_2)^3}_{\theta_8} \]

\[ y(x_1, x_2) = \sum_j \theta_j \phi_j(x_1, x_2) = \theta^T \phi(x_1, x_2) \]
Neural Networks

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

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

\[
y(\bar{x}) = w_{1,0} + f\left(\sum_j w_{1,j} y_j^1\right),\quad y_j^1 = w_{2,0} + f\left(\sum_j w_{2,j} x_j\right)
\]

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

Linear in parameters
Multi-Layered Perceptrons

- Network parameter computation
  - training data set
  - parameter identification
    \[ y(\bar{x}) = F(\bar{x}; \theta) \]
- Nonlinear LS problem
  \[ V = \sum_{j} \| y^{(j)} - F(\bar{x}^{(j)}; \theta) \|^2 \rightarrow \min \]
- Iterative NLS optimization
  - Levenberg-Marquardt
- Backpropagation
  - variation of a gradient descent
Fuzzy Logic

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

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

- 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 combinations 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

\[
\sum_j Y_j \varphi_j (\Theta) = Y(\Theta)
\]
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

Relational Database → Multidimensional Data Cube

Query point (What if?)

Outdoor temperature → Time of day → Heat demand

Forecasted variable

Explanatory variables

Heat Loads

Data mining in the loop
Honeywell product