NONLINEAR SYSTEM IDENTIFICATION

Michele TARAGNA, Carlo NOVARA

Dipartimento di Elettronica e Telecomunicazioni

Politecnico di Torino

`michele.taragna@polito.it`

Master Course in Mechatronic Engineering

Master Course in Computer Engineering

01RKYQW / 01RKYOV "Estimation, Filtering and System Identification"

Academic Year 2019/2020

The nonlinear system identification problem

- Output data are generated by the nonlinear system f_o :

$$y_o(t) = f_o(\varphi_o(t))$$

where:

$$\varphi_o(t) = [\, y_o(t-1) \ \cdots \ y_o(t-n_y) \ \ u(t-1) \ \cdots \ u(t-n_u) \,]^T \text{ : regressor}$$

$u(t)$: known values, $\forall t \geq 1$ (“exogenous” input)

- The nonlinear system f_o is unknown, but a finite number N of noise-corrupted measurements of $y_o(t)$ and $\varphi_o(t)$ are available:

$$y(t) = f_o(\varphi(t)) + v(t)\,, \quad t = 1, \ldots, N$$

where:

$$\varphi(t) = [\, y(t-1) \ \cdots \ y(t-n_y) \ \ u(t-1) \ \cdots \ u(t-n_u) \,]^T \in \mathbb{R}^n, \ n = n_y + n_u$$

$v(t)$ accounts for noises in measurements $y(t)$ and $\varphi(t)$

- **Identification problem:** find an estimate $\hat{f} \approx f_o$

- Related problems:
  - for a given estimate $\hat{f} \approx f_o$, evaluate the **identification error** $\left\| f_o - \hat{f} \right\|$
  - find the optimal estimate $\hat{f} \approx f_o$ that minimizes the identification error
- Main difficulty: the identification error cannot be exactly evaluated, since neither the system f_o nor the noise $v(t)$ are known
- Prior assumptions on f_o and $v(t)$ are necessary for deriving finite bounds on the identification error

Parametric approach to nonlinear identification

- Typical assumptions in literature:
  - on system: $f_o \in \Psi(\theta) = \left\{ f(\varphi, \theta) = \sum_{k=1}^{r} \alpha_k \sigma_k(\varphi, \beta_k) \right\}$
  - on noise $v(t)$: stochastic, i.i.d. (independent and identically distributed)
- Functional form of f_o :
  - derived from physical laws
  - σ_k : k-th “basis” function (polynomial, trigonometric, sigmoid, etc.)
- Parameters θ are estimated by means of the Prediction Error method (PEM), using as predictor of $y(t)$:

$$\hat{y}(t|t-1) = f(\varphi(t), \theta) = \sum_{k=1}^{r} \alpha_k \sigma_k(\varphi(t), \beta_k)$$

- Given N noise-corrupted measurements $y(t)$ and $\varphi(t)$:

$$\begin{cases} y(1) = f(\varphi(1), \theta) + \varepsilon(1) \\ y(2) = f(\varphi(2), \theta) + \varepsilon(2) \\ \quad\vdots \\ \underbrace{y(N)}_{Y} = \underbrace{f(\varphi(N), \theta)}_{F(\theta)} + \underbrace{\varepsilon(N)}_{E_\varepsilon} \end{cases} \Rightarrow$$

Measurement equation:

$$\boxed{Y = F(\theta) + E_\varepsilon}$$

Y : measured output

$F(\theta)$: known nonlinear function of θ

$E_\varepsilon = Y - F(\theta)$: prediction error

- It is possible to estimate θ by means of the Prediction Error method (PEM):

$$\hat{\theta}_{LS} = \arg\min_{\theta} J_N(\theta)$$

$$J_N(\theta) = \frac{1}{N} E_\varepsilon^T E_\varepsilon = \frac{1}{N} [Y - F(\theta)]^T [Y - F(\theta)] \qquad \text{(MSE)}$$

Problem: $J_N(\theta)$ is in general non-convex, since $F(\theta)$ is nonlinear

- If possible, physical laws are used to obtain the parametric representation of $f(\varphi, \theta)$
- When the physical laws are not well known or may result too complex, black-box parametrizations are used:
  - *fixed* basis parametrization, with
    - polynomial basis functions $x^k \Rightarrow x,\ x^2,\ \ldots$
    - trigonometric basis functions $\cos(kx) \Rightarrow \cos(x), \cos(2x),\ \ldots$
    - sigmoidal basis functions

$$\frac{1}{1+e^{-kx}} \Rightarrow \frac{1}{1+e^{-x}}, \frac{1}{1+e^{-2x}},\ \ldots$$

    - hyperbolic tangent basis functions

$$\tanh(kx) = \frac{e^{kx} - e^{-kx}}{e^{kx} + e^{-kx}} = \frac{1 - e^{-2kx}}{1 + e^{-2kx}} \Rightarrow \tanh(x), \tanh(2x),\ \ldots$$

  - *tunable* basis parametrization (neural networks)

Parametric nonlinear ID with fixed basis functions

$$f(\varphi,\theta) = \sum_{k=1}^{r} \alpha_k \sigma_k(\varphi), \qquad \theta = [\,\alpha_1 \;\; \cdots \;\; \alpha_r\,]^T$$

$\sigma_k(\varphi) =$ basis functions

- **Problem:** can σ_k's be found such that:

$$f(\varphi,\theta) \xrightarrow[r \to \infty]{} f_o(\varphi) \text{ ?}$$

- **Result** (Weierstrass)**:** for continuous f_o , bounded regressor space $\Phi \subset \mathbb{R}^n$ and basis function σ_k polynomial of degree k:

$$\lim_{r\to\infty} \sup_{\varphi\in\Phi} \| f_o(\varphi) - f(\varphi,\theta) \| = 0$$

$$\Downarrow$$

polynomial models

- Parameters θ are estimated by means of the Prediction Error method (PEM), using as predictor of $y(t)$:

$$\hat{y}(t|t-1) = f(\varphi(t), \theta) = \sum_{k=1}^{r} \alpha_k \sigma_k(\varphi(t)), \qquad \theta = \begin{bmatrix} \alpha_1 & \cdots & \alpha_r \end{bmatrix}^T$$

- NARX models: estimation of θ with PEM is a linear problem:

$$\boxed{Y = L\theta + E_\varepsilon}$$

$$L = \begin{bmatrix} \sigma_1(\varphi(1))^T & \cdots & \sigma_r(\varphi(1))^T \\ \vdots & \ddots & \vdots \\ \sigma_1(\varphi(N))^T & \cdots & \sigma_r(\varphi(N))^T \end{bmatrix}, \quad Y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}$$

- Least squares solution:

$$\hat{\theta}_{LS} = \left(L^T L\right)^{-1} L^T Y$$

- $NARX\,(n_a, n_b, n_k)$ models can be understood as nonlinear extension of linear $ARX\,(n_a, n_b, n_k)$ models, whose structure is:

$$\begin{aligned} y(t) &= -a_1 y(t-1) - \ldots - a_{n_a} y(t-n_a) + b_1 u(t-n_k) + \ldots + b_{n_b} u(t-n_k-n_b+1) \\ &= \left[y(t-1) \cdots y(t-n_a)\; u(t-n_k) \cdots u(t-n_k-n_b+1) \right] \left[-a_1 \cdots -a_{n_a}\; b_1 \cdots b_{n_b} \right]^T \\ &= \varphi(t)^T\,\theta, \quad \text{where } \theta = [-a_1 \cdots -a_{n_a}\; b_1 \cdots b_{n_b}]^T \in \mathbb{R}^{n_a+n_b} \end{aligned}$$

If $n_a = n_b = 2$ (i.e., $n = n_a + n_b = 4$), $n_k = 1$ and the nonlinear function $f(\varphi, \theta)$ is polynomial of degree $r = 2$, then the nonlinear predictor of $y(t)$ is:

$$\hat{y}(t) = \alpha_1 \sigma_1(\varphi(t)) + \alpha_2 \sigma_2(\varphi(t)) = [\alpha_1, \alpha_2] \begin{bmatrix} \sigma_1(\varphi(t)) \\ \sigma_2(\varphi(t)) \end{bmatrix} = \left[\sigma_1(\varphi(t))^T \; \sigma_2(\varphi(t))^T \right] \theta$$

where:

$$\sigma_1(\varphi(t)) = [y(t-1), y(t-2), u(t-1), u(t-2)]^T = \varphi(t) \in \mathbb{R}^n = \mathbb{R}^4$$

$$\begin{aligned} \sigma_2(\varphi(t)) = [&y(t-1)^2, y(t-2)^2, u(t-1)^2, u(t-2)^2, y(t-1)y(t-2), y(t-1)u(t-1), \\ &y(t-1)u(t-2), y(t-2)u(t-1), y(t-2)u(t-2), u\,(t-1)u(t-2)]^T \in \mathbb{R}^{n(n+1)/2} \end{aligned}$$

$$\alpha_1 = [\alpha_{1,1}, \cdots, \alpha_{1,4}],\; \alpha_2 = [\alpha_{2,1}, \cdots, \alpha_{2,10}] \;\Rightarrow\; \theta = [\alpha_1, \alpha_2]^T \in \mathbb{R}^{n(n+3)/2} = \mathbb{R}^{14}$$

- Model structure choice:
  - kind of basis functions σ_k
  - number r of basis functions $\Longleftarrow$ The model complexity may be exponential in $n = \dim(\Phi)$
  - dimension n of regressors φ

- **Problem:** *"curse of dimensionality"*

  The number of parameters needed to obtain "accurate" models may grow exponentially with the dimension n of regressor space Φ

  $\Uparrow$

  more relevant in the case of fixed basis functions (for NARX, $\dim(\theta) = O(n^r)$)

Parametric nonlinear ID with tunable basis functions

$$f(\varphi,\theta)=\sum_{k=1}^{r}\alpha_k\sigma_k(\varphi,\beta_k)\,,\quad \beta_k=[\,\beta_{k,1}\cdots\beta_{k,q}\,]\in\mathbb{R}^{1,q},\ q\geq n=\dim(\varphi)$$

$$\theta=[\,\alpha_1\ \cdots\ \alpha_r\ \beta_{1,1}\ \cdots\ \beta_{r,q}\,]^T$$

- One of the most common tunable parametrizations is the one-hidden layer hyperbolic tangent neural network $\Rightarrow$ the nonlinear predictor of $y(t)$ is:

$$\hat{y}(t|t-1)=f(\varphi\,(t)\,,\theta)=\sum_{k=1}^{r}\alpha_k\sigma_k(\varphi(t)\,,\beta_k)$$

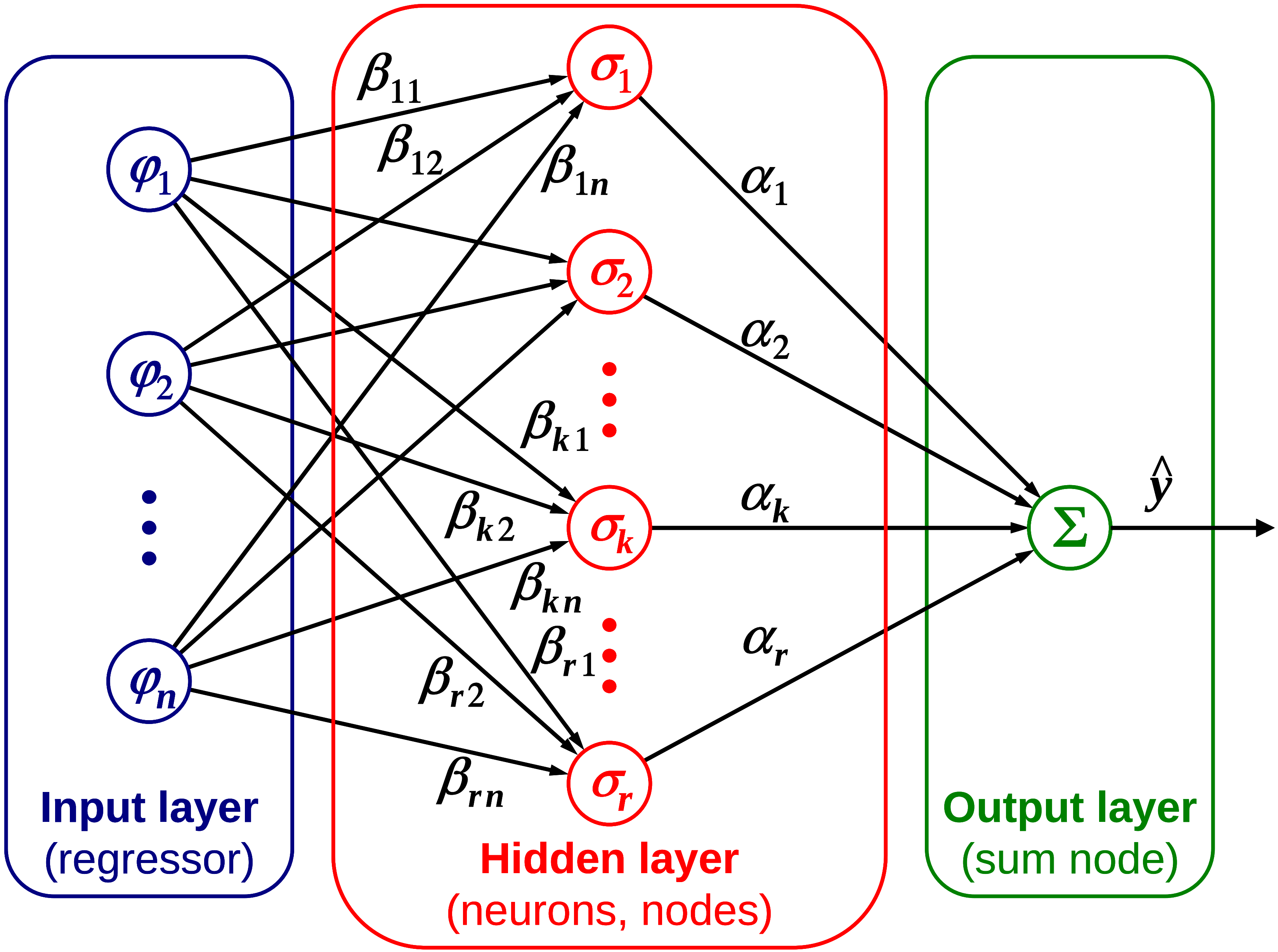
$$\sigma_k(\varphi(t)\,,\beta_k)=\tanh\left(\beta_k\cdot\varphi(t)\right)$$

- Under suitable regularity conditions on the function to approximate, the number of parameters required to obtain “accurate” models grows linearly with n
- **Drawback:** estimation of θ requires to solve a non-convex minimization problem (even for NARX models)

$$\Downarrow$$

Trapping in local minima

One-hidden layer neural network:



Nonlinear identification with NNSYSID Toolbox

- Under MATLAB, the model structure of the neural network having:
  - $n = \dim(\varphi)$ inputs in the input layer
  - r hyperbolic tangent neurons (or nodes or units) in the hidden layer
  - 1 linear neuron (or node or unit) in the output layer

  can be defined as:

$$\texttt{NetDef=['}\underbrace{\texttt{H}\cdots\texttt{H}}_{r\text{ times}}\texttt{';'L}\underbrace{\texttt{-}\cdots\texttt{-}}_{r-1\text{ times}}\texttt{'];}$$

- The maximum number of iterations of the identification algorithm is set to 500 by:

```
trparms=settrain;
trparms=settrain(trparms,'maxiter',500);
```

- The SISO identification of a Neural Network ARX model $\mathbf{NNARX}(n_a, n_b, n_k)$, where $\varphi(t) = [y(t-1) \cdots y(t-n_a)\ u(t-n_k) \cdots u(t-n_k-n_b+1)]^T \in \mathbb{R}^n$, $n = n_a + n_b$, is performed with:

```
[W1,W2]=nnarx(NetDef,[na,nb,nk],[],[],trparms,ye,ue)
```

where: `NetDef` = model structure of the neural network
`na` = number of past measurements in the regressor φ
`nb` = number of past inputs in the regressor φ
`nk` = minimum input-output delay
`trparms` = "training" parameters
`ye` = row-vector output estimation signal ("training" output)
`ue` = row-vector input estimation signal ("training" input)
`W1` = $\mathbb{R}^{r,n+1}$ matrix of row-vector weights $\beta_k = [\,\beta_{k,1} \cdots \beta_{k,n+1}\,]$
`W2` = $\mathbb{R}^{1,r+1}$ row-vector of scalar weights α_k

$\Rightarrow$ the nonlinear predictor of $y(t)$ is:

$$\hat{y}(t) = \sum_{k=1}^{r} [W_2]_k \tanh\Big(\sum_{j=1}^{n} [W_1]_{k,j}\, \varphi_j(t) + [W_1]_{k,n+1}\Big) + [W_2]_{r+1}$$

- The structure of the NNARX model can be plotted by:

```
drawnet(W1,W2)
```

- The predicted output $\hat{y}(t)$ of the NNARX model for the validation dataset:

$$\hat{y}(t) = \sum_{k=1}^{r} [W_2]_k \tanh\left(\sum_{j=1}^{n} [W_1]_{k,j}\, \varphi_j(t) + [W_1]_{k,n+1}\right) + [W_2]_{r+1}$$

defining the regressor φ as a column vector, can be computed as:

```
phi=[yv(t-1:-1:t-na);uv(t-nk:-1:t-nk-nb+1)];
alfa=W2(1:end-1); alfa_0=W2(end);
beta=W1(:,1:end-1); beta_0=W1(:,end);
yp(t)=alfa*tanh(beta*phi+beta_0)+alfa_0;
```

Nonlinear regression systems

- Consider a nonlinear system in regression form:

$$y(t) = f(\varphi(t)) + v(t)\,, \quad t = 1, \ldots, N$$

where:

  - $\varphi(t)$: regressor, that defines the system structure:

$$\varphi(t) = [\; y(t-1) \;\; y(t-2) \;\; \cdots \;\; u(t-1) \;\; u(t-2) \;\; \cdots \;]^T \Leftrightarrow \text{NARX}$$

$$\varphi(t) = [\; f(\varphi(t-1)) \;\; f(\varphi(t-2)) \;\; \cdots \;\; u(t-1) \;\; u(t-2) \;\; \cdots \;]^T \Leftrightarrow \text{NOE}$$

$$\varphi(t) = [\; y(t-1) \;\; y(t-2) \cdots u(t-1) \;\; u(t-2) \cdots v(t-1) \;\; v(t-2) \cdots]^T \Leftrightarrow \text{NARMAX}$$

  - $u(t)$: "exogenous" input signal
  - $v(t)$: noise acting on the system

- The predictor of the system f is defined as:

$$\hat{y}(t) = f(\varphi(t)), \quad t = 1, \ldots, N$$

where:

$\varphi(t) = [\, y(t-1)\; y(t-2) \cdots u(t-1)\; u(t-2) \cdots]^T \Leftrightarrow$ NARX

$\varphi(t) = [\, \hat{y}(t-1)\; \hat{y}(t-2) \cdots u(t-1)\; u(t-2) \cdots]^T \Leftrightarrow$ NOE

$\varphi(t) = [\, y(t-1)\; y(t-2) \cdots u(t-1)\; u(t-2) \cdots \varepsilon(t-1)\; \varepsilon(t-2) \cdots]^T \Leftrightarrow$ NARMAX

$\varepsilon(t) = y(t) - \hat{y}(t)$: prediction error