NonLinear System Identification

Jonas Sjöberg

May 13, 2020
Linear systems:

\[ y(t) = \int_{-\infty}^{\infty} h(t-\tau)u(\tau)d\tau \Leftrightarrow Y(j\omega) = H(j\omega)U(j\omega) \]

If \( u(t) = \sin \omega_0 t \) then \( y(t) = H(j\omega_0)\sin \omega_0 t \). The same frequency as the input.

Nonlinear, for example

\[ y(t) = u^2(t) \]

and \( u(t) = \sin \omega_0 t \) gives

\[ y(t) = \sin^2 \omega_0 t = \frac{1 - \cos 2\omega_0 t}{2} \]

The two frequencies \( e^{j\omega_0 t} \) and \( e^{-j\omega_0 t} \) mix with each other and themselves.

Generally, non nice frequency domain interpretation available in nonlinear case.
Stability

Linear systems: system property, are the poles in the left half-plane?

Nonlinear systems: generally stability depends on the input signal.

Example

$$\frac{dy}{dt} = u \cdot y$$

consider positive and negative step responses

$$u = -1 \iff y(t) = y(0)e^{-t},$$

which converges. On the other hand

$$u = 1 \iff y(t) = y(0)e^{t}$$

diverges.
Stationary points

Linear system

\[ \frac{dx}{dt} = Ax(t) + Bu(t) \]

has one single stationary point \( x = 0 \) (if \( A \) not singular).

Nonlinear system

\[ \frac{dx}{dt} = f(x(t), u(t)) \]

might have many stationary points \( x_1, \ldots, x_n \) so that

\[ \frac{dx}{dt} = f(x_i, 0) = 0, \; i = 1, \ldots, n \]

The system can be linearly approximated at these stationary points and it can be locally stable or unstable depending on the properties at the stationary point.
Limit cycles

Can be stable or unstable.
Example: system with two states:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2) \\
\dot{x}_2 &= f_2(x_1, x_2)
\end{align*}
\]

Phase plot: linear system nonlinear system
Finite escape time

States and outputs of linear system are finite for any finite $t$. This is not necessary true for nonlinear systems.

**Linear system:**

\[ y(t) \]

**Nonlinear system:** $\dot{x} = x^2$

\[ y(t) \]
In a control perspective the behavior from the *input* signal to the output signal is the important issue.

\[ Y(s) = H(s)U(s) \]

Easier to analyze system behavior and to design controller if the system is linear.

From a parameter estimation point of view it is the *parameterization* which matters. Estimation much easier for linear parameterized models (linear regression).

\[ y(t) = \theta^T \phi(t) \]
Try to Stay Linear!

Many nonlinear modeling and identification problems can be re-defined so that linear techniques can be used. For example, the power $y(t)$ in a resistor as a function of the applied voltage

$$y(t) = \frac{u^2(t)}{R}$$

If the $u^2(t)$ is considered as input instead of $u(t)$ the system becomes linear. If $k = 1/R$ is used as parameter instead of $R$ then one has a linear parameterized model.
Linearize around certain points \((x_0, u_0)\):

\[
\frac{dx}{dt} = f(x(t), u(t))
\]

\[
\approx f(x_0, u_0) + A(x(t) - x_0) + B(u(t) - u_0)
\]

Possible to work with a series of linear models, each used in a domain of the \(x(t)\) space.
Linear system identification: all frequencies have to be present in the input signal. Amplitude is not of importance.

Nonlinear system identification: more complicated – also amplitude important. In practice: excite all levels and frequencies of interest. Choose “typical trajectory”.
Black-box model

Things Go Nonlinear
Stability
Stationary points
Limit cycles
Finite escape time
Two Types of Linearity
Try to Stay Linear!
Input signal

Black-box model
Training – Minimization
So, what are the problems?
Regressors for input-output models
Nonlinear Input-Output Models
Choices of mapping
Mixed structures

\[ \varphi(t) = \text{Vector of measured signals} \]

\( \theta: \) Parameters

\( \hat{\theta}: \) Parameter estimate
Problem: Find relationship between $y$ and $\varphi$

Regression:

$\varphi(t) = \text{Regressor}$

$\theta = \text{Parameter vector}$

Linear regression:

$\hat{y}(t) = \theta^T \varphi(t) = \sum_{i=1}^{n} \theta_i \varphi_i(t)$

Non-linear regression:

$\hat{y}(t) = g(\varphi(t), \theta)$

where $g$ can be “any” non-linear function - as in the first part of the course “statistical learning”.

Things Go Nonlinear
Stability
Stationary points
Limit cycles
Finite escape time
Two Types of Linearity
Try to Stay Linear!
Input signal

Black-box model
Training – Minimization
So, what are the problems?
Regressors for input-output models
Nonlinear Input-Output Models
Choices of mapping
Mixed structures
Example:

\( p(t) = \text{Heat produced in a resistance.} \)

\( \varphi(t) = [u(t) \ u(t-1)] \)

where \( u(t) \) is the applied voltage
Nonlinear models by choosing nonlinear regressor, e.g.,
\[ \varphi(t) = [u(t) \ u^2(t) \ \ldots] \]

This is still linear regression! Linear in the *parameters*:
\[ \hat{y}(t) = \theta^T \varphi(t) \]

Is this not enough to describe non-linearities?
Training – Minimization

\( V_N(\theta) = \frac{1}{2N} \sum_{t=1}^{N} |y(t) - g(\varphi(t), \theta)|^2 \)

Minimization algorithm

\( \hat{\theta}(i+1) = \hat{\theta}(i) - \mu_i R_i^{-1} V_N'(\theta(i)) \)

\( \hat{\theta}(0) \): Initial parameter guess.
\( \mu_i \): Step length to ensure descent updates.
\( R_i \): Matrix changing the search direction.
So, what are the problems?

1. Select $\varphi$ – regressor
2. Select (the model structure) $g(\theta, \varphi)$
   - $g(\theta, \varphi) = \sum_{k=1}^{m} \theta(k)g_k(\varphi)$
   - select the $g_k$
3. An algorithm to find $\hat{\theta}_N$
4. To understand the model’s properties
High dimension is a problem

If some directions are more important one can project the data and estimate a function in a space of lower dimension.

One sigmoid (hidden unit)
Nonlinear black-box models do not have to be “very nonlinear”.

Sum of two sigmoids in two dimensions – a feedforward neural net.

A radial basis function model with two basis functions

These “universal approximators” can be seen as a smoothed local linear model respectively a linear model with local deviations.
Consider the mapping from data to regressor

\[ \{ y_{t-1}, u_{t-1} \} \rightarrow \varphi(\theta, t), \]

where \( \varphi(\theta, t) \in R^d \).

Assuming data generated by linear system

\[ y(t) = G_o(q^{-1})u(t) + H_o(q^{-1})e(t) \]

Given system model \( G(q^{-1}) \) and noise model \( H(q^{-1}) \) the predictor becomes

\[ \hat{y}(t) = H^{-1}(q^{-1})G(q^{-1})u(t) + (1 - H(q^{-1}))y(t) \]
\[ = \theta^T \varphi(\theta, t) \]

Hence, linear black box models can be written

\[ \hat{y}(t) = \theta^T \varphi(\theta, t). \]
The components of $\varphi(\theta, t)$ are:

**FIR:** $u(t - k)$,  
$\varphi(t) = [u(t - 1) \ u(t - 2) \ldots]$ 

**ARX:** $y(t - k), u(t - k), \varphi(t) = [y(t - 1) \ y(t - 2) \ldots u(t - 1) \ldots]$
\[ \hat{y}(\theta, t-1), u(t-k), \varphi(\theta, t) = [\hat{y}(\theta, t-1) \hat{y}(\theta, t-2) \ldots u(t-1) \ldots] \]

or as a “recurrent network”
Nonlinear Input-Output Models

The linear standard models can all be described as

\[ \hat{y}(t) = \theta^T \varphi(t) \] – linear regression (ARX, FIR) or

\[ \hat{y}(t) = \theta^T \varphi(\theta, t) \] – pseudo-linear regression (OE, ARMAX).

Nonlinear models are obtained by choosing a nonlinear mapping:

\[ \hat{y}(t) = g(\theta, \varphi(t)) \] – nonlinear regression (NARX, NFIR),

\[ \hat{y}(t) = g(\theta, \varphi(\theta, t)) \] – pseudo-nonlinear regression (NOE, NARMAX, state-space) also called recurrent models.
Linear models specified after that $\varphi(\theta, t)$ is defined. In nonlinear identification also the mapping

$$g : \varphi \longrightarrow \hat{y}(t)$$

also has to be specified.

Possibilities:

- Any black-box structure (neural net, wavelet, ...). Gives “universal approximators”.
- A “mixed” model somewhere between linear (or physical nonlinear) and general nonlinear model.
Mixed structures

In cases where linear models have been rejected it might still be interesting to try a intermediate model somewhere between linear and fully nonlinear models.

This leads to models with less parameters than a fully nonlinear model and, hence, to a lower variance error.

For example, assume additive noise

\[ y(t) = g(u^t) + v(t). \]

where \( v(t) \) can be described as filtered white noise

\[ v(t) = H(q)e(t) \]
This gives the predictor

$$\hat{y}(t) = (1 - H^{-1}(q))y(t) + H^{-1}(q)g(u^t).$$

which is linear in past outputs.

with $g_1$ is linear.

This structure has some nice properties:

- Stability is easy to determine.
- ”Easy” to design a controller from this model.
A NARMAX model linear in $e(t)$:

$$y(t) = g(\theta, \varphi(t)) + C(q)e(t)$$

The regressors $\varphi(t)$ contains past $y$ and past $u$. The predictor becomes

$$\hat{y}(t) = g(\theta, \varphi(t)) + (C(q) - 1)\varepsilon(t)$$

with $\varepsilon(t) = y(t) - \hat{y}(t)$.

The ”NARX” part $g$ can be further divided with a linear part as discussed previously.
Traditional nonlinear models like the Hammerstein

\[
\begin{align*}
& u(t) \quad \xrightarrow{\int} \quad \text{OE/ARX} \quad \hat{y}(t) \\
& G \\
\end{align*}
\]

and the Wiener model

\[
\begin{align*}
& u(t) \xrightarrow{G} \hat{y}(t) \\
\end{align*}
\]

are also “restricted” nonlinear models. The static nonlinear element can be realized with, eg, a neural net.