02_System Identification and Linearization

1. Parameter Estimation in Nonlinear Models (white box)

Steps:

1. construct predict output and generate the **prediction error vector**
2. Define the **performance index** $J(\theta)$ with **loss function** g
3. Compute the optimal value and **optimal** θ (By nonlinear optimization methods)

Nonlinear Optimization Methods

- Newton's Method: gradient and Hessian
- Gauss-Newton: neglects second order terms in Hessian
- Levenberg-Marquardt (GN plus regularization)

`lsqnonlin` Function

Syntax: `x=lsqnonlin('fun',x0,lb,ub,options)` with

- Denoting by $x = \theta$, $F_k(x) = \varepsilon_k(\theta)$
- `fun` : m-file function returning $F(x)$ and its Jacobian J_F (optional)

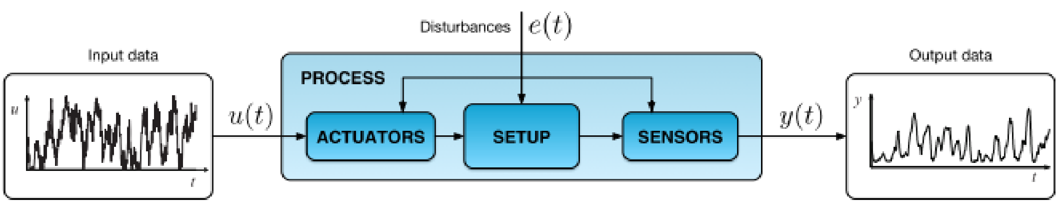
```
[F,J]=fun(x)
F=...
% Compute Jacobian if required.
if (nargout > 1)
    J=...
end;
```

- `x0` : initial starting point
- `lb`, `ub` : lower and upper bounds for `x` (optional)
- `options` : options structure (optional)

- `LargeScale` : use a large-scale algorithm (`'on'`) or medium-scale algorithm (`'off'`).
- `Display` : controls display of (intermediate) values. Possible values: `'off'`, `'iter'`, and `'final'`
- `Jacobian` : indicates whether Jacobian is defined by user
- `MaxIter` : maximum number of iterations allowed
- `TolFun`, `TolX` : termination tolerance on the function value and on `x`
- `LevenbergMarquardt` : Choose Levenberg-Marquardt over Gauss-Newton algorithm (default: `'on'`)

2. Linear System Identification in Time Domain (black box)

Overall Method: PE method



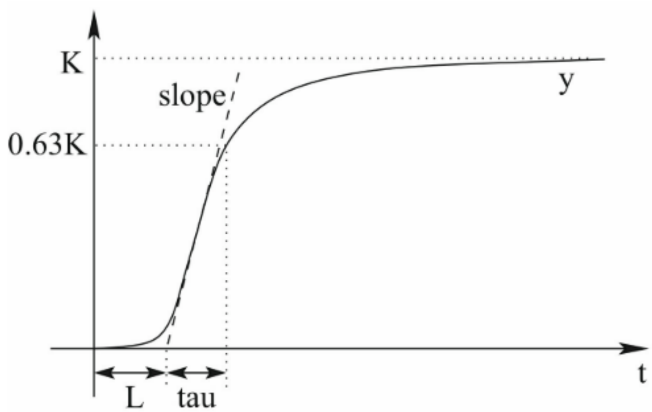
Data	$u(1), u(2), \dots, u(N) \qquad y(1), y(2), \dots, y(N)$
Model	Ideally, a transfer function $y(k) = \frac{B(q)}{A(q)} u(k)$
Parameter	$\theta(k) \triangleq [a_1\ a_2\ \dots\ a_n\ b_0\ b_1\ \dots\ b_m]^\top$
Problem	How do we assume e is affecting the dynamics? This lead to different methods (ARX, ARMAX, OE, BJ)

Identification of 1^{st}-order models

$$G_1(s) = \frac{Ke^{-Ls}}{\tau s + 1}$$

- L is a delay in the action of the input

$$y(t) \approx K(1 - \exp(-(t-L)/\tau))$$



Identification of 2^{nd}-order models

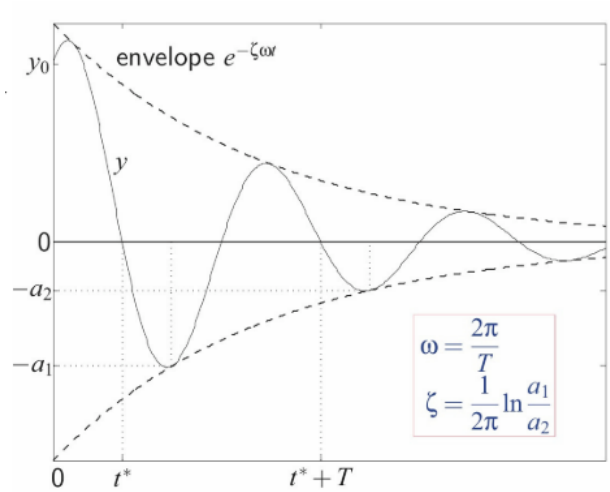
$$\ddot{x} + 2\omega_N \zeta \dot{x} + \omega_N^2 x = K_{ss}\omega_N^2 u, \quad y = x$$

Transfer function:

$$G_2(s) = \frac{K_{ss}\omega_N^2}{s^2 + 2\zeta\omega_N s + \omega_N^2}$$

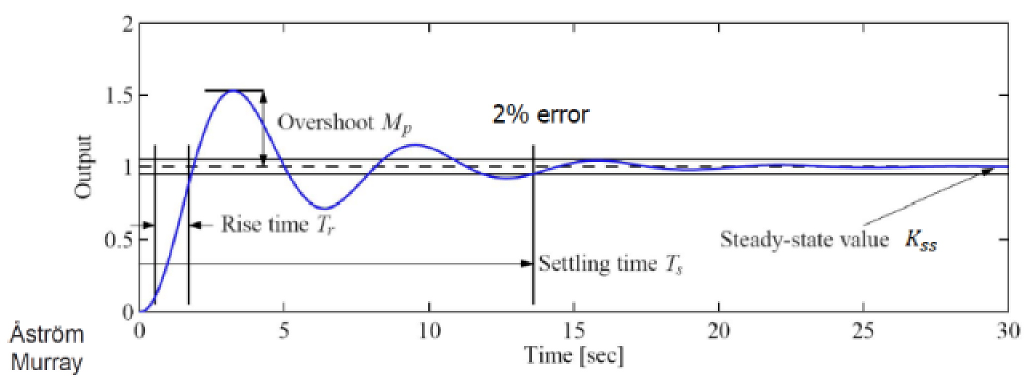
In this experiment $u = 0$.

$$\omega = \omega_N\sqrt{1-\zeta^2}$$

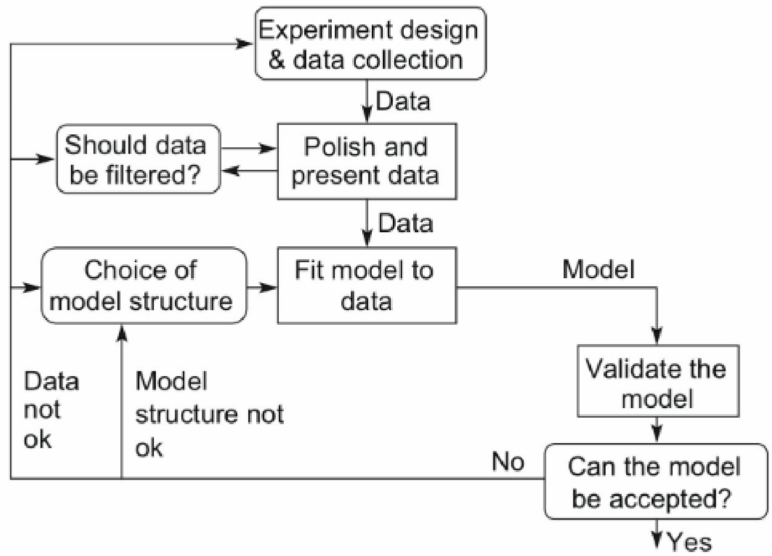


Step Response Model:

- $M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \Rightarrow \text{ compute } \zeta$
- $T_s \approx \frac{4}{\zeta\omega_N} \Rightarrow \text{ compute } \omega_N$



3. Experiment Design and the Identification Procedure



Choice of Input Signal

Selection

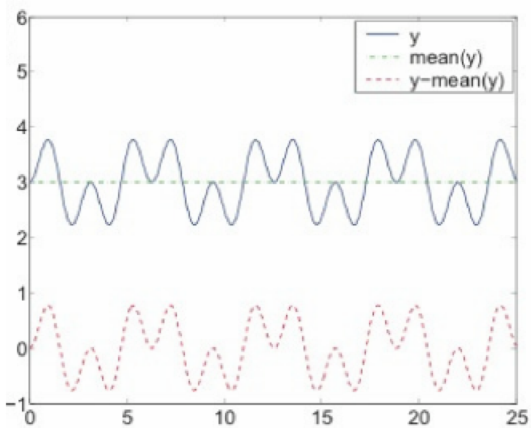
- For **linear identification**: do not use full range of input signals!
  - relatively small amplitude around an operating point
- For **nonlinear identification**: two levels (H, L) may not be sufficient
  - use multiple levels or smoothly varying signals (multi-sine)
- Take care that the output signal amplitude is significantly larger than the noise amplitude
  - **signal-to-noise ratio should be large enough**

Suggestions

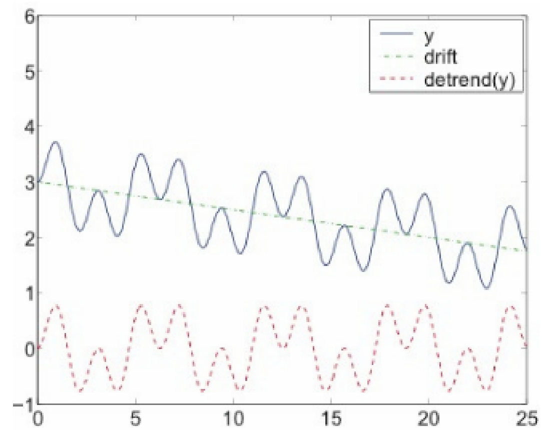
- It is a good idea to first generate the **input sequence off-line**, and to **examine its properties** before applying it to the system → first try it with the **simulation model**
- **Binary signals** (telegraph, PRBN) often suitable to **identify linear systems**
  - For **nonlinear models** → **multi-level telegraph signal**

Post-Treatment of Data

Subtract mean (for identification around operating point)



Removing drift: `detrend(y)`



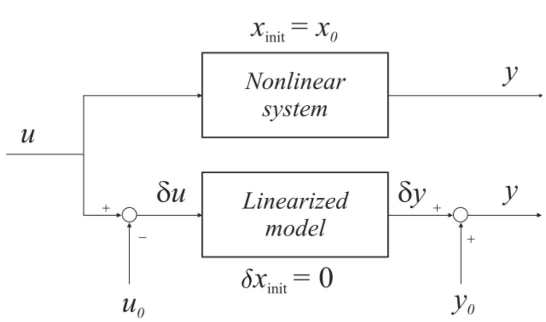
If a filter is used for identification → use it in the control design

Overfitting

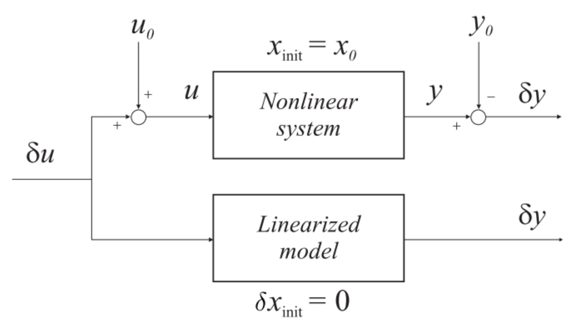
Solution

1. Cross-Validation
2. Penalize the number of parameters (e.g. Akaike)

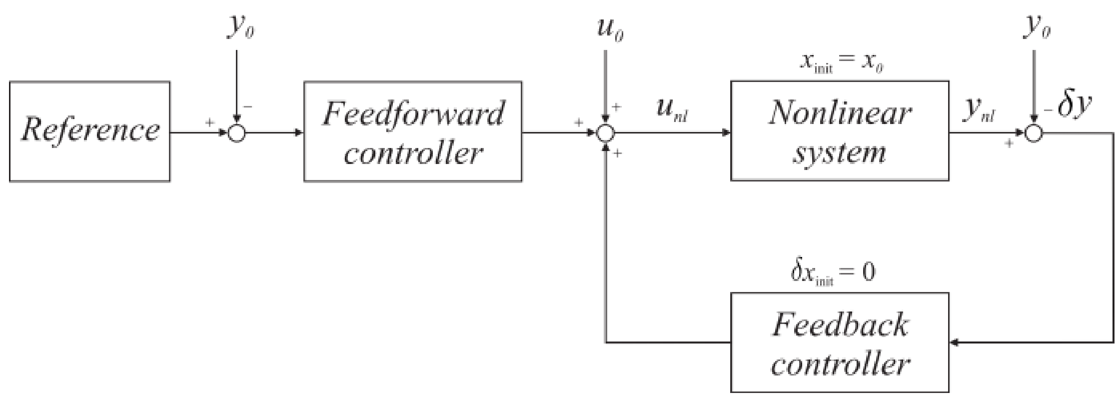
4. Linearization of Nonlinear Models



$\delta u = u_{nl} - u_0$ $\qquad$ $y_{nl} = \delta y + y_0$



$u_{nl} = \delta u + u_0$ $\qquad$ $\delta y = y_{nl} - y_0$



Linear controllers must use signals δu, δy, δx!

Summary

- White Box Parameter Estimation: known model structure, estimate parameter
- Black Box Parameter Estimation: classical parameter estimation
- Experiment Design and Overfitting
- Linearization on Nonlinear Model