

Simulering og eksperimentel modelbestemmelse

Klaus Trangbæk

`ktr@es.aau.dk`

Automation & Control

Aalborg University

Denmark

Dagens program

- SENSTOOLS til parameter-estimation
 - Navnekonventioner
 - Procedure for parameter-estimation
- Parameter-følsomhed
 - Følsomhedsmål
 - Følsomhedsellipse

SENSTOOLS til parameter-estimation

- **Senstools** er en samling af Matlab-programmer, der implementerer følsomhedsmetoden for direkte parameter-estimation, eksperiment-design og modelvalidering.
- Programmerne kan fordelagtigt organiseres som en **Matlab Toolbox**.
- Alt hvad brugeren behøver at programmere er **simuleringsprogrammet for den specifikke proces**.
- Programmerne er organiseret som main programmer (script filer), der kalder underprogrammer (funktioner) og bruger data (mat filer).

Navnekonventioner

- Program- og data-filnavnene indeholder information om programtype og navnet på den aktuelle proces. Begyndelses bogstaverne indikerer typen:
 - `main` main program (script fil)
 - `sim` simulerings program for processen (funktion)
 - `meas` input/output måledata (mat fil)
 - `prog` program data (mat fil)
- Navne på filer tilknyttet en specifik proces skal indeholde proces navnet. Eksempel:
`process='motor' simmotor.m` og `measmotor.mat`
- Programnavnene indeholder også info om funktionen. Eksempel: `mainest.m` (main program for estimation)

Procedure for parameter-estimation

For en proces med navn `xxx`:

1. **Opret simuleringsprogrammet** som en Matlab funktion: `y = simxxx.m`
2. **Gem de målte data** `t`, `u` og `y`: `save measxxx t u y`
3. **Indtast nødvendige program data**, en af de tre måder:
 - a) Direkte i workspace: `process='xxx' ;`
`par0=[1 2] ;`
 - b) Indlæses fra en mat fil (`progdatabxxx.mat`). Sker automatisk hvis filen eksisterer. progdata-filen oprettes med `progprogxxx.m` fil.
 - c) Brug default værdier for main programmet.
4. **Kør `mainest.m`** for parameter-estimation.

Hvordan de tre inputmetoder skelnes

Programkoden for mainest.m:

```
% mainest is the main program for parameter estimation
:
% 20/9-94,MK. 26/11-02,MK

% Default values:
if ~exist('process'), process='ktau'; end % Process name
if ~exist('no'), no=''; end % Measurement number
if exist(['progdata',process,no,'.mat'])==2 & ~exist('par0')
    load(['progdata',process,no]), end % progdata loades
if exist(['meas',process,no,'.mat'])==2, load(['meas',process,no]),
else
    disp(['data: meas',process,no,'.mat missing !']), break, end
if ~exist('ploty'), ploty=2; end
if ~exist('par0'), par0=[1.5 3]; end
simmod=['sim',process];
:
```

Eksempel: Par. estim. med `mainest.m`

System `kutau`

Model: $\frac{Y(s)}{U(s)} = \frac{K(u)}{1 + s\tau}$ hvor $K(u) = k_0(1 + \frac{k_1}{0.5+u^2})$

`measkutau.mat` eksisterer (målinger u, y, t)

Eksempel: DC-motor (SIMO)

```
» help simdcml
```

```
y=[i,w]=simdcml(u,t,par) simulates a linear dc-motor with  
input u and outputs i and w.
```

```
w/u = K/R/(J*s+B+K^2/R), i/u=(J*s+B)/R/(J*s+B+K^2/R)
```

```
par=[R K J B]
```

```
27/11-02,MK
```


Evaluering af model fit

Parameter estimat: $\theta_N = \underset{\theta}{\operatorname{argmin}} P(u_N, y_N, \theta)$

Performance funktion: $P(\theta) = \frac{1}{2N} \sum_{k=1}^N \epsilon^2(k, \theta)$

Minimum værdien $P(\theta_N)$ giver **ikke** brugbar information om hvor godt fit der er opnået.

Parameterfølsomhed

Følsomhedsellipse

Næste Forelæsning

Næste gang ser vi på:

- Parameter-nøjagtighed
- Input-signal design