Modeling and Learning for Dynamical Systems

Lecture 7

Martin Enqvist

LINKÖPING UNIVERSITY

Parametric Identification

Two Types of Parameterized Models

A model that depends on a parameter vector θ (of fixed size) is referred to as a **parameterized model**. It is common to distinguish two types of parameterized models:

- **Grey-box models**: The model structure is derived using first-principles and the parameters have often a physical meaning (or are functions of some physical quantities). However, some of the parameters are unknown and have to be estimated from data.
- **Black-box models**: The model structure is generic and can be applied in many domains and settings. The parameters have no physical interpretation and are estimated from data such that the generic model can give an as accurate description as possible of a particular system. The choice of model structure is done based on data.

N.B. Sometimes people use the term *white-box models* for first-principles models without unknown parameters. It is also possible to talk about models having *different shades of grey* when the amount of physical insight differs.

Parameterized Transfer-Function Models

A generic parameterized linear transfer-function model:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t),$$

- θ: parameter vector
- $u(t)$: input signal to the system
- $y(t)$: output signal from the system
- $e(t)$: white noise signal
- We will here assume that the sampling time is $T = 1$

Rational Transfer Function Model Structures

Rational transfer function models

$$y(t) = \frac{B(q)}{A(q)F(q)}u(t) + \frac{C(q)}{A(q)D(q)}e(t)$$

can be defined using the five polynomials

$$A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}$$
$$B(q) = b_{n_k} q^{-n_k} + b_{n_k+1} q^{-n_k-1} + \ldots + b_{n_k+n_b-1} q^{-n_k-n_b+1}$$
$$C(q) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}$$
$$D(q) = 1 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d}$$
$$F(q) = 1 + f_1 q^{-1} + \ldots + f_{n_f} q^{-n_f}$$

N.B.:

- Despite the notation/terminology, these functions are polynomials of q^{-1} (and rational functions of q).
- The parameters n_a, n_b, n_c, n_d and n_f denote the number of parameters in each polynomial.
- The parameter n_k denotes the model delay.

Rational Transfer Function Model Structures...

Special cases for particular subsets of polynomials used:

- FIR: $B(q)$ (results in a linear regression $\hat{y}(t|\theta) = \varphi(t)^T\theta$)
- ARX: $A(q)$, $B(q)$ (results in a linear regression $\hat{y}(t|\theta) = \varphi(t)^T\theta$)
- ARMAX: $A(q)$, $B(q)$, $C(q)$
- OE: $B(q)$, $F(q)$
- BJ: $B(q)$, $C(q)$, $D(q)$, $F(q)$

Here, $\varphi(t)$ contains past and present input components and (in the ARX case) past output components and θ contains polynomial coefficients.

Prediction Errors

For a particular choice of predictor model and parameter vector θ_*, the *prediction error*

$$\varepsilon(t, \theta_*) = y(t) - \hat{y}(t|\theta_*)$$

describes the difference between the measured output and the output predicted (as well as possible) using the particular model obtained with $\theta = \theta_*$.

A Key Idea: Minimizing the Prediction Error

One approach to parameter estimation: Select θ such that (some measure of) the size of the prediction errors is minimized (underlying idea: the true system should give the best prediction of the output)

The prediction-error method (PEM):

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta)$$

$$\text{where } V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} l(\varepsilon(t,\theta))$$

A common choice when $y(t)$ is scalar:

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\theta)^2$$

(For this choice, the PEM estimator coincides with the maximum-likelihood (ML) estimator when $e(t)$ is normally distributed.)

A Family of Minimization Approaches

In most cases, the minimization problem in PEM has to be solved in an iterative way using numerical optimization methods (exceptions: ARX and FIR models with quadratic cost function).

A family of minimization approaches:

$$\hat{\theta}_N^{(i+1)} = \hat{\theta}_N^{(i)} - \mu_N^{(i)} (R_N^{(i)})^{-1} V_N'(\hat{\theta}_N^{(i)})$$

Here, $\mu_N^{(i)}$ is a *step size* and $R_N^{(i)}$ is a matrix that can be chosen in different ways. Setting this matrix to the Hessian ($R_N^{(i)} = V_N''(\hat{\theta}_N^{(i)})$) results in a *Newton method*. Other popular choices result in the *Gauss-Newton method* (approximate the Hessian using the gradient) or the *Levenberg-Marquardt method*.

Model Fit

The **model fit** M_f is a common measure of model quality:

$$M_f = 100 \left(1 - \frac{\|Y_m - \hat{Y}\|_2}{\|Y_m - \bar{Y}_m\|_2} \right)$$

where Y_m and $\hat{Y}$ are vectors containing N output measurements and predictions, respectively, and where $\bar{Y}_m$ is a vector with the mean of Y_m repeated N times.

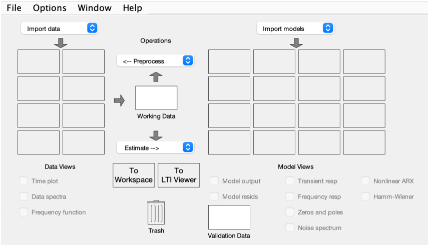
- $M_f = 100\%$: perfect model
- $M_f = 0\%$: poor model, similar to just taking the mean of the output
- $M_f < 0\%$: very poor model

The model fit should primarily be computed for a separate validation dataset that was not used for model estimation.

Matlab Example

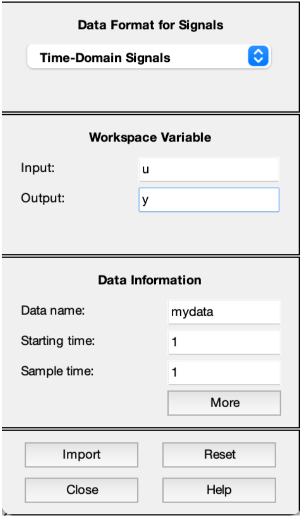
Matlab Example

Start the graphical user interface to the System Identification Toolbox in Matlab with the command `systemIdentification`



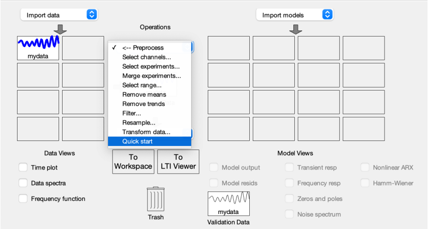
Matlab Example. . .

Import data from workspace



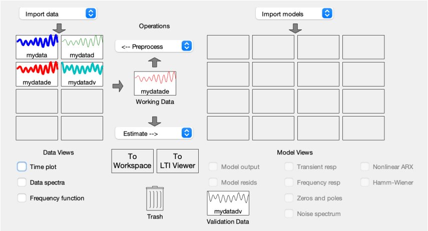
Matlab Example. . .

Preprocess data. The quick start option removes means and splits the data into estimation and validation data.



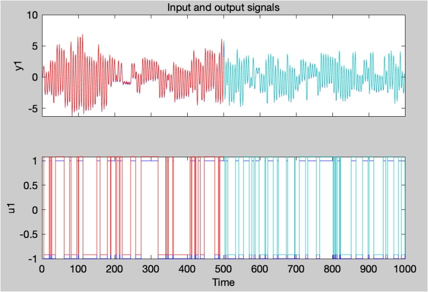
Matlab Example. . .

Result of quick start preprocessing



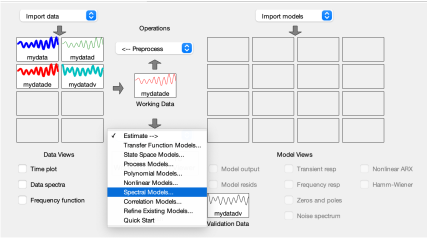
Matlab Example. . .

Look at the data. No remaining trends or strange outliers here.



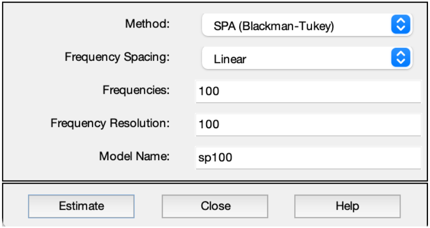
Matlab Example. . .

Estimate nonparametric spectral models



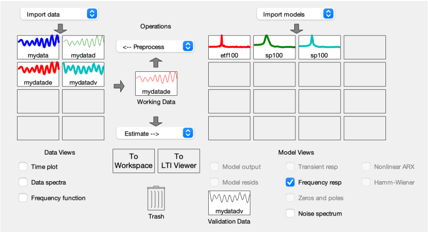
Matlab Example...

Try SPA and ETFE with different frequency resolutions (window widths)



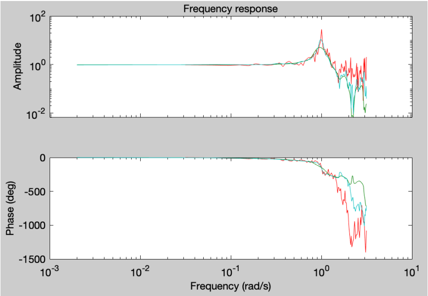
Matlab Example. . .

The estimated models end up to the right



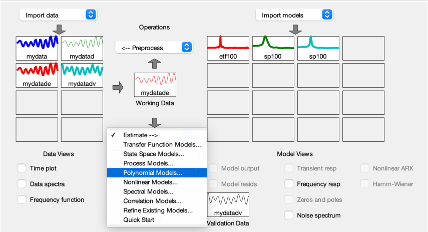
Matlab Example. . .

Look at the estimated frequency responses. There seems to be a clear resonance peak at $\omega = 1$.



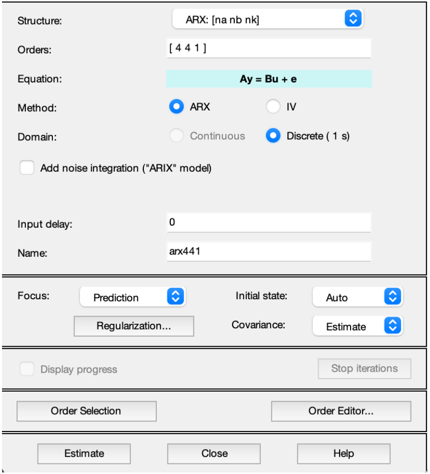
Matlab Example. . .

Estimate rational transfer-function models (here called polynomial models)



Matlab Example. . .

ARX and the order selection option is a good starting point

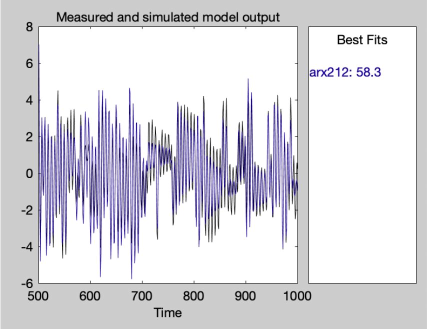


Matlab Example. . .

Many ARX models are estimated and compared. Select one right after the biggest drop in unexplained output variance. Here: ARX212.

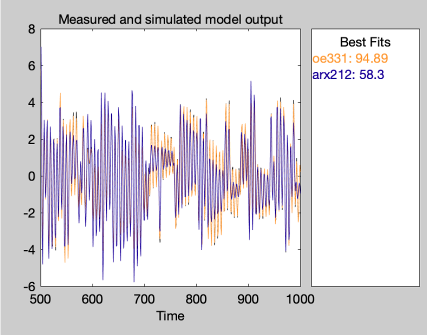
Matlab Example. . .

Look at the model output. The model has picked up part of the dynamics but it might be possible to find a better one. Let us try some other model structures (OE, ARMAX, BJ, etc.) and model orders.



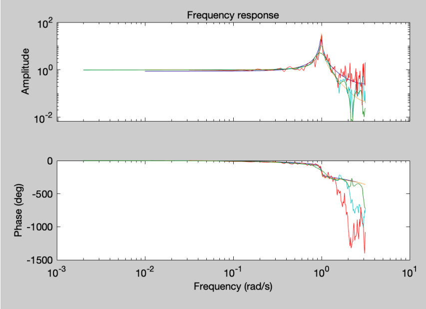
Matlab Example...

After some testing, an OE331 model that gives a high model fit is found



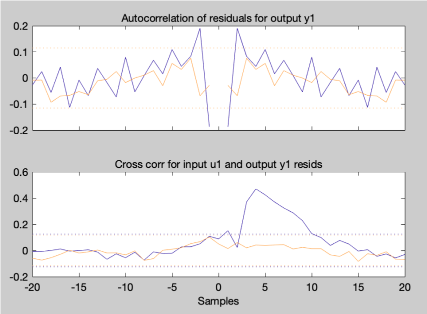
Matlab Example. . .

The parametric models have frequency responses that are similar to the nonparametric ones



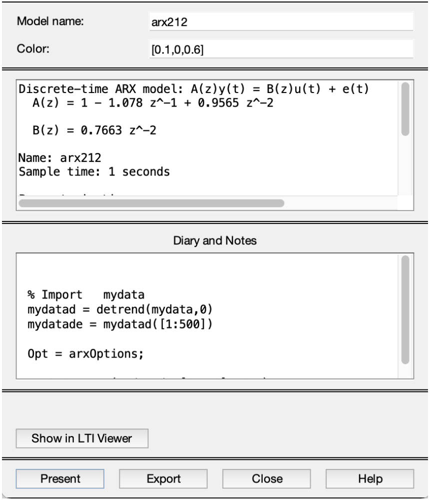
Matlab Example. . .

The model residuals show how white $\varepsilon(t)$ is for each model and if there is correlation between $\varepsilon(t)$ and $u(t - \tau)$. Here, the OE model is clearly better than the ARX model.



Matlab Example. . .

Right-clicking on a model gives details about the parameter values and how the model was obtained. The present option gives more details in Matlab's workspace.



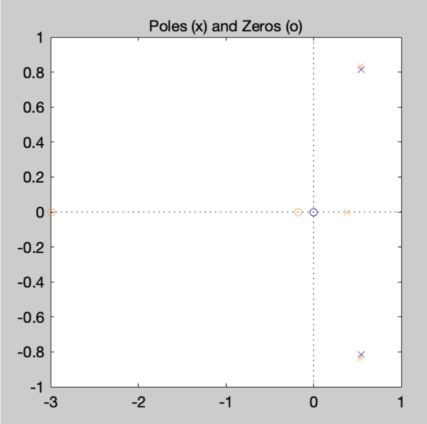
Matlab Example. . .

In particular, the workspace information contains estimated standard deviations for the parameters, which can be used to find unneccessary parameters (the standard deviation should ideally be small compared to the parameter value). This seems OK here.

```
arx212 =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
A(z)= 1 - 1.078 (+/- 0.006591) z^-1 + 0.9565 (+/- 0.006448) z^-2
B(z)= 0.7663 (+/- 0.01351) z^-2
Name: arx212
```

Matlab Example. . .

A zeros and poles plot can be used to find possible pole-zero cancellations. No obvious ones in this case.



Matlab Example. . .

In this case, the estimated OE331 model seems like a good choice.

```
oe331 =
Discrete-time OE model: y(t) = [B(z)/F(z)]u(t) + e(t)
B(z)= 0.1205 (+/- 0.006321) z^-1 + 0.382 (+/- 0.009364) z^-2
  + 0.06379 (+/- 0.01485) z^-3
F(z)= 1 - 1.449 (+/- 0.01355) z^-1 + 1.386 (+/- 0.01449) z^-2
  - 0.3721 (+/- 0.01326) z^-3

Name: oe331
Sample time: 1 seconds
```

Summary

Parametric Identification

- Parameterized models
- Rational transfer-function model structures: FIR, ARX, ARMAX, OE, BJ
- Predictions of the output
- The prediction-error method
- Example using the GUI of the System Identification Toolbox in Matlab

www.liu.se

LINKÖPING UNIVERSITY