Project 2: Estimating Transfer Functions

SSY 230, System Identification
Jonas Sjöberg
Electrical Engineering, Chalmers
April 13, 2020

In this project you will develop a functions for estimating transfer function. You will use the functions from the first project as building blocks. You will also write functions for analysing the model and for using it for prediction and simulation.

1 Write an ARX estimator

(a) Write a command \texttt{arxfit(z,[na nb nk])} that takes a data data set \texttt{z}, and three integer parameters as inputs and that delivers an ARX model. \texttt{na} indicates the number of past outputs in the regressor, \texttt{nb} number of past inputs, and \texttt{nk} delays in the input signal. To validate the function, generate some noise-free data, and verify that you re-obtain the correct parameter values. Store the result in an object like the ones you produced in \textit{project 1}. Add fields to the object, one indicating that it is an ARX model and one with the indeces. These fields will be useful to recognise what type of model it is.

(b) Write a function \texttt{id2tf(arx)} that convert an ARX model to Matlab’s transfer function (see \texttt{tf}). Using that function, you can evaluate your estimated models using \texttt{ltiview}.

(c) Write functions \texttt{idpredict(model,z,horizon)} and \texttt{idsimulate(model,z)}. For \texttt{idsimulate} you can have good help of \texttt{lsim}.

(d) (optional - but easy and useful) Write a function \texttt{idcompare(z,model,horizon)} that plots the simulated/predicted output together with the true output. Plot also the uncertainty. This is an excellent function for validation where you can test the estimated model on validation data and evaluate the performance. If you don’t implement this function, consider at least how you could have evaluated the uncertainty and write this in the report. Discuss it also at some of the meetings.

2 Write an OE estimator

You will now write a function \texttt{oeefit(z,[nb nf nk])} that returns an OE-model.

(a) Write a function \texttt{oeefit(z,[nb nf nk])} that performs the following steps
• Estimate a high order ARX model of order \( na=4*nf \), \( nb=4*nb \) and \( nk \) unchanged. This gives you an ARX model and let us call the numerator \( B_h \) and the denominator \( A_h \).

• Simulate the ARX model using the input data from the identification of it, giving the output \( y_s \).

• Here you have two options:
  
  (1) (approximate method) Estimate a new ARX model using the original input, the simulated output, \( y_s \), and the original model order. The result is your OE model.

  (2) (optimal method, motivation found in appendix) Filter both the original input and simulated output, \( y_s \), through the FIR filter \( A_h \). Estimate a new ARX model using the filtered signals and the original model order. The result is your OE model.

  Change the appropriate field so that it indicates that it is an OE model.

• Explain why you obtain an OE model in this way. Discuss at a meeting if you are lost.

Verify your estimator. Add white noise to the output signal and verify that you re-obtain the true parameter values when the number of data is high. Verify also that this is not the case if you identify with arxfit.

(b) Modify id2tf so it works for OE-models too.

(c) Modify idpredict so it works for OE-models too.

(d) Modify idsimulate so it works for OE-models too.

(e) (Optional) Modify idcompare so it works for OE-models too.

3 Identify two systems

Now it is time to use your functions on some data sets from two different systems.

For the first system you find the data in the file exercise1 and the input is in the variable \( u \) and the output in \( y \).

The second system data you find in exercise2 and the data is placed in \( z1 \), estimation data and \( z2 \), validation data. The input is in the second column and the output is in the first column.

For both systems, estimate good models taking the following into account.

• Try different orders of ARX and OE models and evaluate the model quality.

• Investigate model properties by looking at, eg, poles and zeros, and impulse response (using ltiview).

• Try prediction with different horizon and simulation on validation data. Is the same model best in all aspects?
Illustrate prediction performance using your command idcompare.

Appendix: Explanation of the filtered version

The assumption is that data was generated by an OE structure

\[ y(t) = G_0 u(t) + e(t) \]

and we want to estimate \( G_0 \) with a two stage method. First step is to estimate a high order ARX model. The order of the model is assumed to be so high that we can consider the plant and disturbance model to be independent parameterized and we have the following expression for the asymptotic estimate as \( N \to \infty \). Open loop, \( G(q, \rho) \) and \( H(q, \eta) = 1/A(q) \), \( \theta = [\rho, \eta] \)

\[ \rho^* = \arg \min_{\rho} \int_{-\pi}^{\pi} |G_0(e^{i\omega}) - G(e^{i\omega}, \rho)|^2 Q_*(\omega, \eta^*) d\omega \]

where

\[ Q_*(\omega, \eta^*) = \Phi_u(\omega)|A(e^{i\omega})|^2 \]

The frequency weighting is due to that we used an ARX model and its disturbance model. In time domain this corresponds to minimizing

\[ \frac{1}{N} \sum_{t=1}^{N} \left( \left[ \frac{B_h}{A_h} - G_0(q) \right] A_h u(t) \right)^2 \]

(1)

Now, given that we estimated the high order model

\[ \frac{B_h}{A_h} \]

and we want to estimate a low order model

\[ \frac{B}{A}, \]

with the same frequency weighting, then we minimize (1) again but now over

\[ G_0(q) = \frac{B}{A}. \]

This can be rewritten as

\[ \frac{1}{N} \sum_{t=1}^{N} \left( \left[ \frac{B_h}{A_h} - \frac{B}{A} \right] A_h u(t) \right)^2 = \frac{1}{N} \sum_{t=1}^{N} \left( \left[ y_{pf} - \frac{B}{A} u_{pf} \right] \right)^2 \]

(2)

where \( y_{pf} = A_h y_f, y_f = \frac{B_h}{A_h} u \) and, \( u_{pf} = A_h u \).

Remark: This filtering should not change the estimate very much since the high order ARX model is approximating an OE process, and if that holds then we expect \( 1/A_h \approx 1 \) and \( y_{pf} \approx y_f \). We could use the same method to estimate the plant model in cases where the true system would be of Box-Jenkin structure, then the true disturbance model would not be 1 and the pre-filtering would give a larger impact on the estimate.