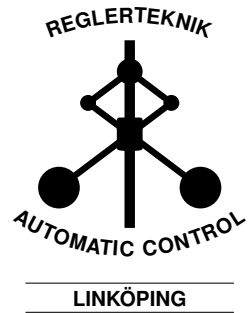


# Computer Exercises in System Identification

## Part 2

This version: 2023-08-28



# 1 Parametric Identification of Black-Box Models

Parametric black-box models are discussed in [2, Chapter 12]. For parametric identification, it is important to have *different data sets for estimation and validation (evaluation) of the model*. If only one data set is available, it is typically split into two parts. This can be done in the GUI by clicking the menu **Preprocess** and choosing **Select range**. Usually, it is often desirable to remove the means of the signals, this can be done in the GUI by clicking the menu **Preprocess** and choosing **Remove means**. A quick way to both split the data set and remove the means, is to use **Quick start** in the **Preprocess** menu.

## 1.1 ARX Models

We will first study how ARX models can be used in order to get an initial understanding of the model dynamics and model order. In Figure 1, the prompt shown by clicking **Estimate** and then choosing **Polynomial Models**. In **Structure**, different model classes may be chosen. We will only consider ARX models for now. In **Orders**, you can enter the desired ARX model order. The order is a space separated list of integers  $n_a, n_b, n_k$ . A model is estimated when one selects **Estimate**. An icon representing this model is created in the main GUI window under **Models**.

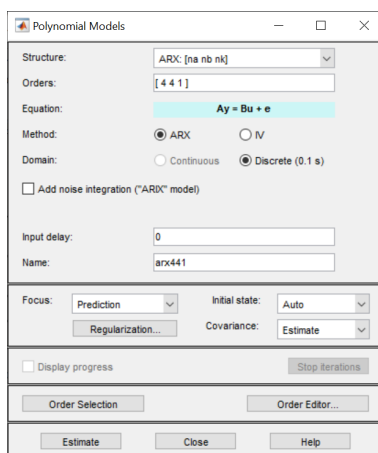


Figure 1: Prompt for estimation of polynomial models.

It may be useful to try many different model orders at once. By clicking **Order Selection** and then **Estimate**, all models with polynomial orders and time delays between 1 and 10 are estimated. The best model for each model order between 2 and 20 (computed from the value of the loss function on validation data) is then shown, see Figure 2. A rough estimate of the model order can be taken as the model order where the loss function stops decreasing significantly (“flattens out”). In Figure 2, this occurs in at around the fifth bar. By clicking on the bar, we can see that  $n_a = 4$  and  $n_b = 1$ , which corresponds to model order 5. This is the model with 5 total parameters which gives the best fit to validation data. The models which you find useful can be put into the main interface by clicking at the bar and then **Insert**.

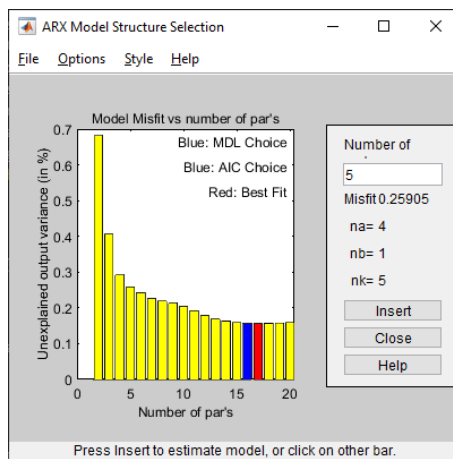


Figure 2: Result of estimating all models of order 2 to 20.

## Exercises for Section 1.1

We now return to the electric motor with load from Exercise 4, nonparametric identification. The description is briefly repeated here.

An approximate physical model for the motor is given by

$$G(s) = \frac{1}{s^3 + 0.9s^2 + 1.1s + 0.46} \quad (1)$$

In an experiment, 1000 data points have been collected with a sample time of 0.3 seconds. The data set has been saved in `elmotor.mat`, where the input signal (voltage, V) is `u` and the output signal (angular velocity, rad/s) is `y`.

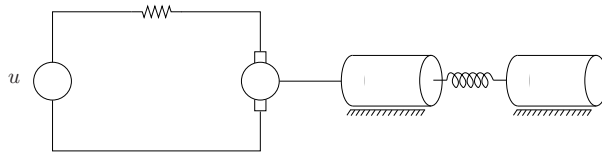


Figure 3: Electric motor with load

1. Import the data set into SITB and study its time and frequency properties.
2. Preprocess the data, and split it up into estimation and validation data.
3. Plot the data in the time and frequency domains and ensure that everything looks normal; i.e., that no trends are in the data and that no outliers are included.
4. By looking at the model (1), what model order should your (discrete-time) ARX model have?

**Hint:** The model can be constructed and discretized in MATLAB by writing:

```
G = tf([1],[1 0.9 1.1 0.46]);
Gd = c2d(G,0.3);
```

5. Use `Order selection` to estimate many ARX models with different model orders. What model orders seem good?
6. Try a few more model orders, and see if you can find a good model!
7. Choose one or a few models which you think are better than the others, and study their properties more closely. Does the model behave like you would expect?

## 1.2 Validation

Model validation is an important part of modeling, see [2, Chapter 16]. The GUI offers many tools for this in `Model Views`:

- `Model output`. Predicted or simulated output signal for validation data.

- **Model resid.** Plots of the prediction error autocorrelation function  $\hat{R}_\varepsilon(\tau)$ , and the prediction error and input cross-correlation function  $\hat{R}_{\varepsilon u}(\tau)$ , computed from the validation data.
- **Frequency resp.** Bode plot of the estimated model.
- **Zeros and poles.** Poles and zeros of the model.

(See [2, Chapter 16] for more information of what to look at for the various choices.) Note in particular that one can see the uncertainty of the quantities by choosing **Show 99% confidence intervals** in **Options** (other percentages can also be chosen). For instance, this is relevant for **Zeros and poles**, since overlapping uncertainty intervals for poles and zeros may indicate a too high model order (pole and zeros are cancelling, and can just as well be removed by reducing the model order). The estimated model parameters with uncertainties can be shown by right-clicking on the model icon. A window according to Figure 4 is then shown. The window also shows code

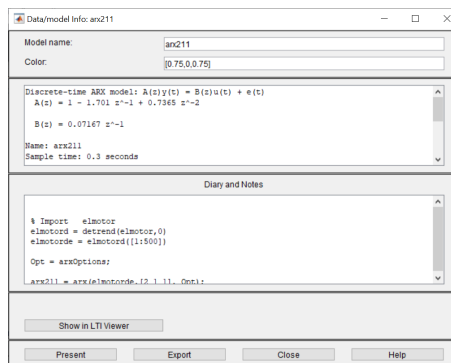


Figure 4: Information about an estimated ARX(2,1,1) model.

to compute the estimated model programmatically. Clicking **Present** prints a model description to the MATLAB command prompt. For the model in Figure 4 (where  $n_a = 2$ ,  $n_b = 1$ ,  $n_k = 1$ ), this is

```
arx211 =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 - 1.701 (+/- 0.02681) z^-1 + 0.7365 (+/- 0.02573) z^-2

  B(z) = 0.07167 (+/- 0.008238) z^-1

Name: arx211
```

Sample time: 0.3 seconds

Parameterization:

Polynomial orders: na=2 nb=1 nk=1

Number of free coefficients: 3

Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using ARX on time domain data "elmotorde".

Fit to estimation data: 93.2% (prediction focus)

FPE: 0.01821, MSE: 0.01785

More information in model's "Report" property.

The printout of the  $B$  polynomial should be interpreted as follows.  $B(q)$  consists of only one parameter, which has an estimated value of 0.07167 and an estimated standard deviation (uncertainty) of 0.008238. Information about the  $A$  polynomial, the value of the loss function, the sampling interval of the data set, and the fit to estimation data is also shown.

Sometimes, it is better to have a model with good predictive performance, and in other cases, it is better to have a good simulation model. This can be studied in `Model` output by clicking `Options` and then choosing `Simulated Output` or `k Step Ahead Predicted Output` (where  $k$  is an integer that you must choose in `Set prediction horizon`).

## Exercises for Section 1.2

1. Compare the frequency estimates (`Frequency resp`) of the black-box models with the spectral models and your knowledge of the system.
2. Analyze the estimated prediction error autocorrelation function  $\hat{R}_{\varepsilon}(\tau)$ , and the prediction error-input cross-correlation function  $\hat{R}_{\varepsilon u}(\tau)$ , in `Model resids`. Can you see any problems with your best models?
3. Study the uncertainty in the estimated parameters of your best models. Are the estimates accurate? Are the parameters significantly different from zero?
4. What model is the best one?

## 1.3 General Black-box Models

We continue to study the electric motor which we discussed earlier. Now, we will use more general model structures than before. The model structures ARMAX, OE and BJ can be estimated in the same prompt as the ARX models. Click `Estimate, Polynomial models`, and then select the desired model structure in `Structure`.

### Exercises for Section 1.3

1. If you saved the experiment from the previous section, load it. Otherwise, import and preprocess the data as before, and re-estimate the best ARX model you found.
2. Try finding even better models by testing ARMAX, OE, BJ, and state-space models. Start your search by using the polynomial orders that the best ARX model had.

## 1.4 An Industrial Application of System Identification

The identification of the electric motor which we looked at before was relatively simple. The process is fairly linear, the measurement noise is low, and the system has low order. For a trickier problem, we will now study a data set from an industrial robot <sup>1</sup>. Data has been collected when the robot is performing a movement about axle 1, see Figure 5.

During the movement, motor torque (input) and motor angle (output) has been measured. Estimation and validation data are stored as the `iddata` objects `ze` and `zv` in the file `irb1400.mat`. The sample time is 0.00504 s (downsampling and preprocessing has already been performed) and the data sets each consist of 397 data points. In both cases, the input signal has been low-pass filtered white noise, and the controller of the robot has been activated. You know that the controller has high gain at low frequencies (static errors are unwanted). You also know that the system is oscillatory due to its construction.

---

<sup>1</sup>The robot is an IRB1400 from ABB

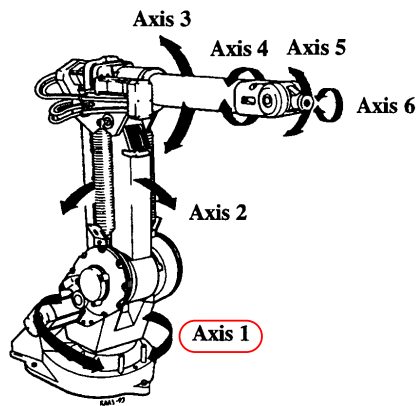


Figure 5: The ABB IRB1400. Axle 1 is encircled.

### Exercises for Section 1.4

1. The data set has been collected in feedback operation. How does this influence the nonparametric spectral methods?
2. The system is oscillatory. What does this mean for nonparametric methods?
3. Use the skills you learned from the electric motor modelling to find as good as possible a model for this process.

## Solutions with Discussion

The solutions below are not necessarily the only correct solution. Depending on choices during data processing and estimation, your results may be perfectly correct even if they do not exactly agree with the discussion below.

### Solutions for Section 1.1

2. It may be a good idea to let the estimation and validation parts of the data set be equally large. Since we do not have much knowledge about the data collection, in this case it suffices to subtract the mean of the signals.



3. Looks OK.
4. The discrete time model has order  $n_a = 3$ ,  $n_b = 3$  and  $n_k = 1$  (thus [3 3 1] in SITB notation) if we assume that the input is constant between each sample instant.
5. You can get different results here, depending on how estimation and validation data are created. **Order selection** indicates that orders [3 1 4], [4 1 5] and [4 2 4] may be suitable choices.
6. The models of order [2 2 1] and [3 1 4] have decent results in **Model output**. By choosing a much higher order, for instance [8 2 3], a much higher model fit is achieved in **Model output**. However, these models have too many poles in comparison to the true system.

### Solutions for Section 1.2

1. The Bode plots for low order models ([2 2 1] and [3 1 4]) seem OK, but have two main flaws in comparison to the true model. First, the resonance peak is off. It should be at 0.9 rad/s, but is at 0.6 rad/s. Second, the slope of the bode plot is too low at high frequencies. The model with order [8 2 3] has a resonance peak at 0.8 rad/s and the slope at high frequencies is higher. This indicates that this model better describes the true system.
2. Looking at the residual analysis of the selected models, the models of order [2 2 1], [3 1 4] and [8 2 3] are similar. There are peaks in both  $\hat{R}_\epsilon(\tau)$  and  $\hat{R}_{\epsilon u}(\tau)$  which are outside the corresponding confidence intervals. This indicates that there is information in the output signal which cannot be explained well by any of these models.
3. All models have parameters which are significant (uncertainty smaller than parameter value), but the model with order [8 2 3] has a few parameters where the confidence interval has the same order of magnitude as the parameter.
4. From the discussion above, the model with  $n_a = 8$ ,  $n_b = 2$  and  $n_k = 3$  appears to best describe the system. However, it has too many poles in comparison to the true system.

### Solutions for Section 1.3

2. We try out different model structures. The search can be carried out as follows:
  - Try increasing model orders to get a good value of model fit.
  - When the prediction error autocorrelation function  $\hat{R}_\epsilon(\tau)$  is outside the confidence interval, increase the noise model order(s).
  - When the prediction error-input cross-correlation function  $\hat{R}_{\epsilon u}(\tau)$  is outside the confidence interval, increase the input model order(s).
  - When pole-zero cancellations appear to occur in the input or noise model, or when certain parameters do not appear to be significantly different from 0, reduce the corresponding model orders. Look at confidence intervals!
  - Ensure that reduced model orders do not yield notable reductions in (prediction or simulation) model fit on validation data.

These objectives may sometimes be contradictory, so choose a good trade-off!

We find that an ARMAX model with  $n_a = 3$ ,  $n_b = 1$ ,  $n_c = 3$  and  $n_k = 2$ , or a state space model of order 3, appear to describe the system very well.

### Solutions for Section 1.4

1. The nonparametric methods we study here normally don't work well in feedback situations. Their performance depends on the strength of the dependence between noise and input. In this case, the dependence is strong for low frequencies and weaker for higher, which indicates that we shouldn't trust these estimates for low frequencies.
2. Since the system is oscillatory, the bode diagram should have clear and narrow resonance peaks. Thus, a high frequency resolution should be used.
3. The system can be fairly well modeled by a BJ model of order [8 4 4 10 1]. The fit to the nonparametric estimate is good for high frequencies, model fit for validation data is very good, and by looking at correlation functions, the residuals appear close to white.

## References

- [1] T. Glad and L. Ljung. *Reglerteknik. Grundläggande teori*. Studentlitteratur, 2006.
- [2] L. Ljung, T. Glad and A. Hansson. *Modeling and identification of Dynamic Systems*. Studentlitteratur, 2021.