Faculty of Automation and Computer Science

NONLINEAR ARX IDENTIFICATION

System Identification semester project

2022 - 2023

Students:
Nagy Timea
Nemes Raluca
Oprea Florin-Octavian

Coordinating teachers:

Prof. dr. ing. Lucian Busoniu

As. dr. ing. Zoltan Nagy

Datafile indexes: 16/19

Contents

Problem Description

ARX black-box model for a **dynamic system** from adjustable model orders and degree

Linear regression to determine the parameters

One-step ahead **Prediction** & **Simulation** implementations

Tune model orders and degree to obtain **best performance**

Approximator Structure

ARX

Easily implementable method that produces parametric, polynomial models

Output y(k) computed based on previous inputs and outputs

Orders na & nb are not needed to be higher than m

NARX

Nonlinear dependence between previous outputs and inputs

Combines multiple regressors of different powers

Vector of delayed inputs & outputs: d

Polynomial of degree m: p

Parameters: coefficients of the polynomial

General ARX model structure

$$y(k) = -a_1y(k-1)-a_2y(k-2)- \dots - a_{na}y(k-na) + b_1u(k-1)+b_2u(k-2) + \dots + b_{nb}u(k-nb) + e(k)$$

Our Solution

1. Preparation

Load the data – index: 19

Build **'Comb_inator'** function

2. Find MSE optim

3. Prediction and Simulation

4. Representative plots

Prediction vs Simulation

Prediction: predict solution from the output and the calculated parameters

- Create vector of combinations for each row
- Create identification and validation matrices for combinations
- Calculate approximator, Theta based on phi_id
- Calculate approximation y_hat based on validation matrix and Theta

Simulation: solution based on previously predicted outputs and parameters

- Initialise y_hat_sim for simulation
- Create combination vector for each row
- Calculate each y_hat_sim based on combinations
- Store all combinations in matrix

Forming the Combinations

Function ‘Comb_inator’ – key features

Creates vector of delayed inputs/outputs

Makes all combinations based around the polynomial degree, m.

Adjusts vector d, so that we don’t have null elements in between - first na rows

Creates an auxiliar vector to filter out the null elements – correct length

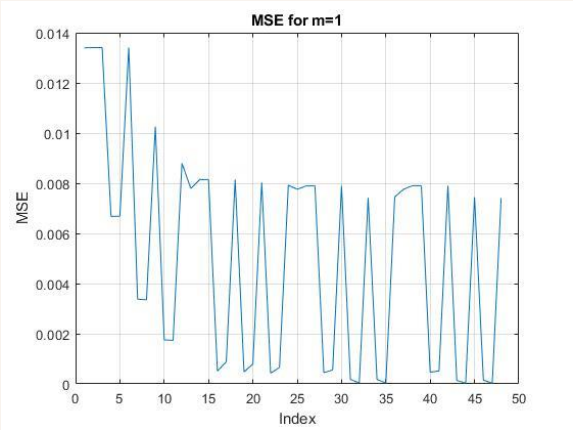
Creates ‘combination’ vector based on algorithm

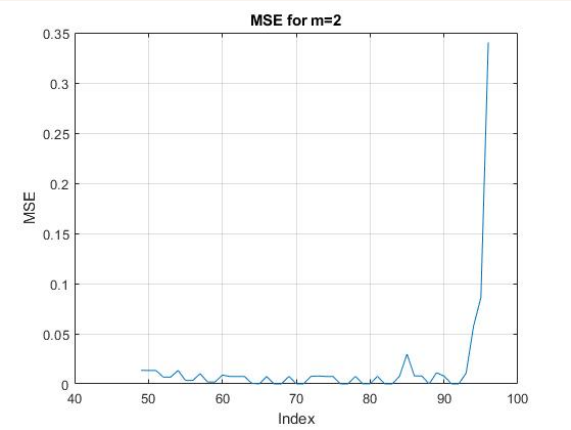
Tuning results

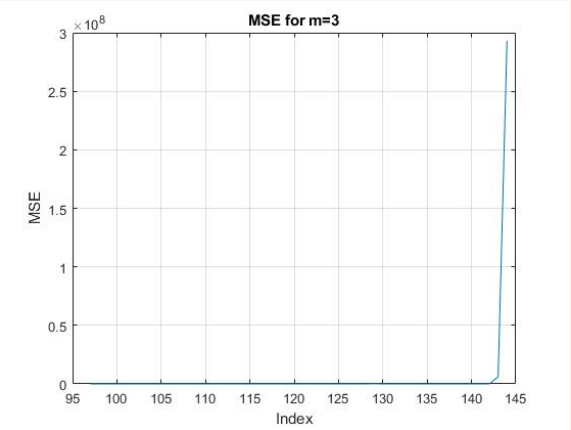
Results vary depending on input data

Depending on our data we calculated the mean-squared error

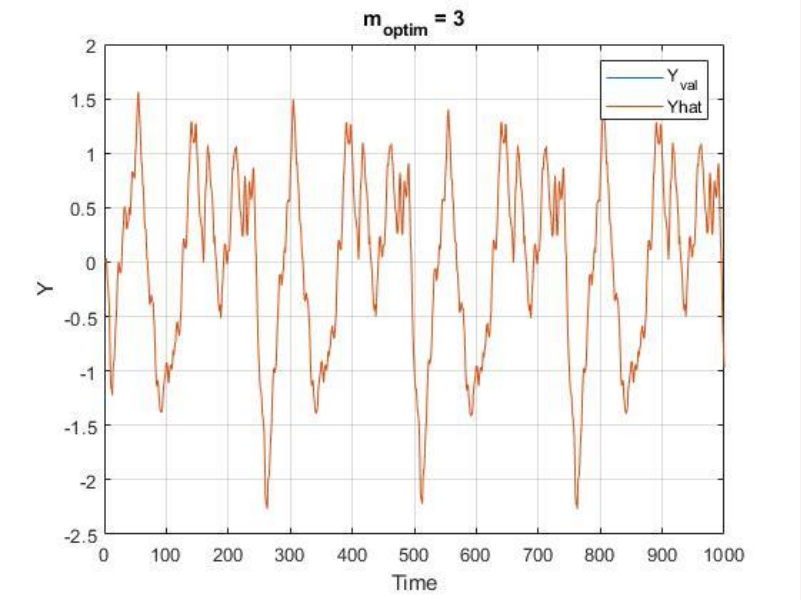
Best solution: for m = 3 , na = nb = 2, nk = 2



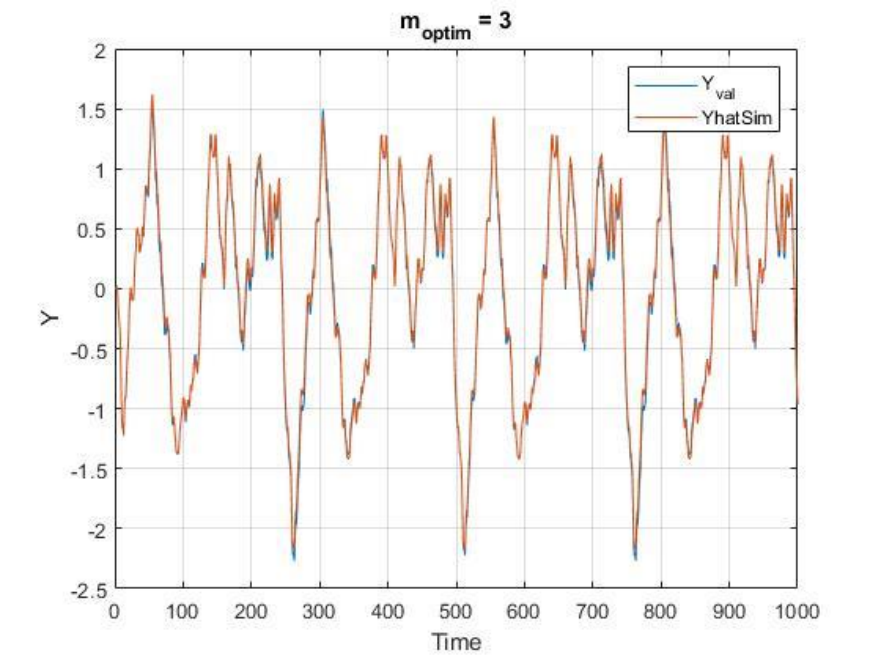




Representative plot for Prediction



Representative plot for Simulation



Conclusion

Optimal results: case m = 3 and na = nb = nk = 2

For m = 1 and na, nk increasing, the model will remain stable.

Our model becomes more unstable, reaches extreme values, as m – also na, nb, nk - increase.

Prediction more accurate than simulation

Thanks!

Do you have any questions?

Bibliography

~ System Identification 2022 (busoniu.net)

~ L. Ljung, System Identification, Wiley Encyclopedia of Electrical and Electronics Engineering, 2007. Section 4, nonlinear models