Nonlinear ARX identification

Table of contents

Description of the problem

What is an ARX model?

- Autoregressive with exogeneous input model:

previous outputs & knowledge about the input → current output

- Black-box model
- Dynamical system (initial conditions)
- For the inputs u and outputs y, with an order of the dynamics of n_a and n_b and the parameters a_{n_a} and b_{n_b}, the general form of the ARX can be written as:

$$y(k) = -y(k-1) - y(k-2) \dots - y(k-n_a) + u(k-1) + u(k-2) \dots + u(k-n_b) + e(k)$$

Delay vector

- Therefore, the delayed outputs and inputs vector can be computed:

$$d(k) = [y(k-1), y(k-2), \dots, y(k-n_a), u(k-n_k), u(k-n_k-1), \dots, u(k-n_k-n_b+1)]^T$$

- By applying the polynomial function to $d(k)$, we get our solution:

$$\hat{y}(k) = f\big(y(k-1), y(k-2), \dots, y(k-n_a), u(k-1), u(k-2), \dots, u(k-n_b)\big)$$

Polynomial approximator

- For example, having an order of dynamics of $n_a = n_b = n_k = 1$, and an order of the polynomial of $m = 2$, we will have 2 inputs for our function. Using the notation $y(k-1) = x_1$ and $u(k-1) = x_2$, the polynomial becomes:

$$\hat{y}(k) = [\,1 \;\; x_1 \;\; x_2 \;\; x_1^2 \;\; x_2^2 \;\; x_1 x_2] \cdot \theta$$

where $\theta = \left[a_1 \;\; \dots \;\; a_{n_a} \; b_1 \dots \; b_{n_b}\right]^T$

Approximator function

Functions

- *delay* function – computes the past input and outputs using the given mathematical formula and creates a matrix for each step k
- *powers* function – generates the powers for every input of our polynomial;
  - returns a matrix of N x no. of inputs, where N – number of all possible combinations with repetitions

For 2 inputs and m=2, we get:

$$\begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix}$$

Functions

- *reg* function – the columns of the delay vector are raised at each row of the power matrix (element-wise):

$$[x_1 \;\; x_2] \,.\!\wedge [2 \;\; 0] = [x_1^2 \; x_2^0]$$

$$[x_1 \;\; x_2] \,.\!\wedge [1 \;\; 1] = [x_1^1 \; x_2^1]$$

$$[x_1 \;\; x_2] \,.\!\wedge [0 \;\; 2] = [x_1^0 \; x_2^2]$$

- Therefore, $\varphi_{power=2} = [x_1^2 x_2^0 \;\; x_1^1 x_2^1 \;\; x_1^0 \, x_2^2]$ for each step k.

- *regressor* function – creates the regressor matrix (calls reg function for power 1 to m)

Parameters

- Nonlinear model & linear parameters
- Linear regression can still be used, so, following the further equation:

$$Y = \phi \cdot \theta$$

- Y – the output (in this case, taken from the training dataset)
- ϕ – the regressor (computed using the training dataset)
- θ – the parameter vector (unknown)

- we can perform left multiplication with $(\phi^T \phi)^{-1} \phi^T$, hence:

$$\theta = (\phi^T \phi)^{-1} \phi^T Y$$

Key features

Approximator function

- Multiply inputs at each iteration:

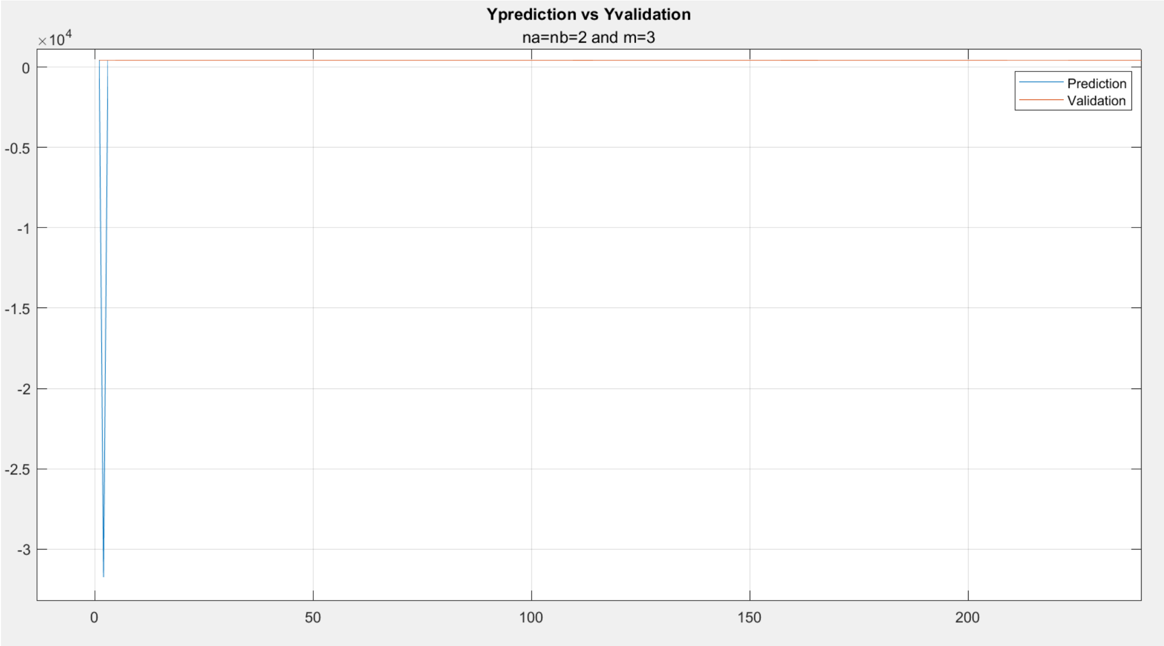
$$x_1x_2x_3 \;\; x_1x_2x_3 \;\; \dots \; x_1x_2x_3$$

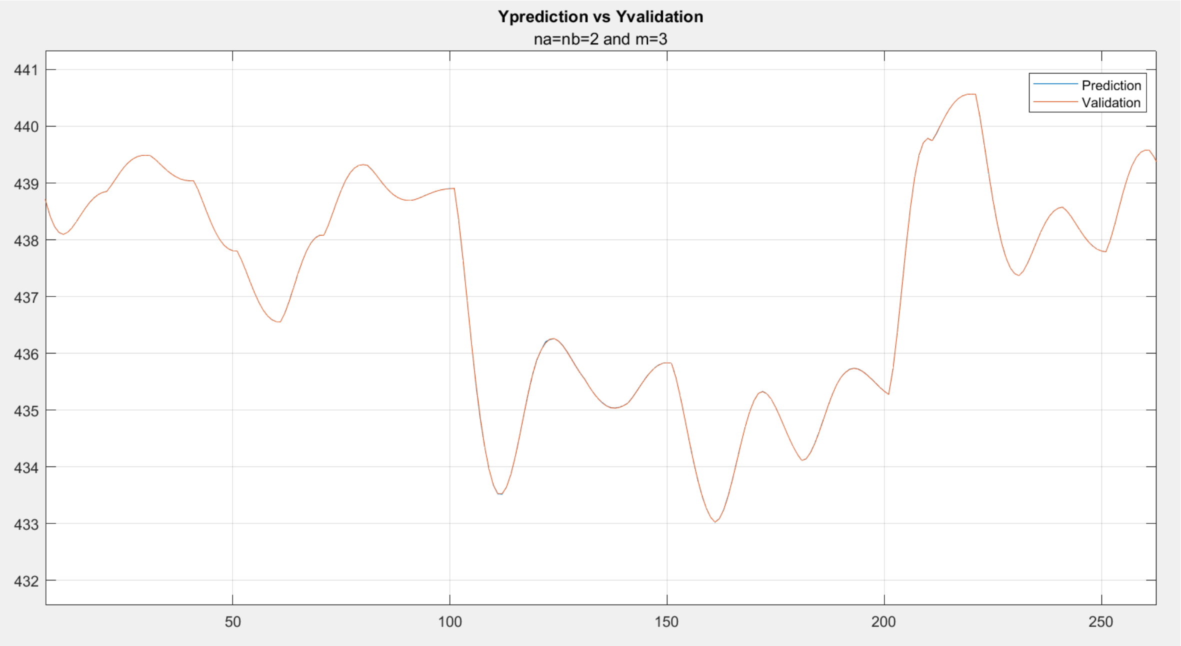
- *nchoosek* (Matlab function) – combinations of elements from a given vector, sizing them as specified
- Assuring repetition – copying the vector a number of times (*number of columns* times)
  - If the number of columns is 3 and m=2, then:

$$v = [0\ 1\ 2] \;\; \rightarrow \;\; v = [0\ 1\ 2\ 0\ 1\ 2\ 0\ 1\ 2]$$

- Imposing 2 conditions: unique rows and sum of the elements equal to m

Corrections



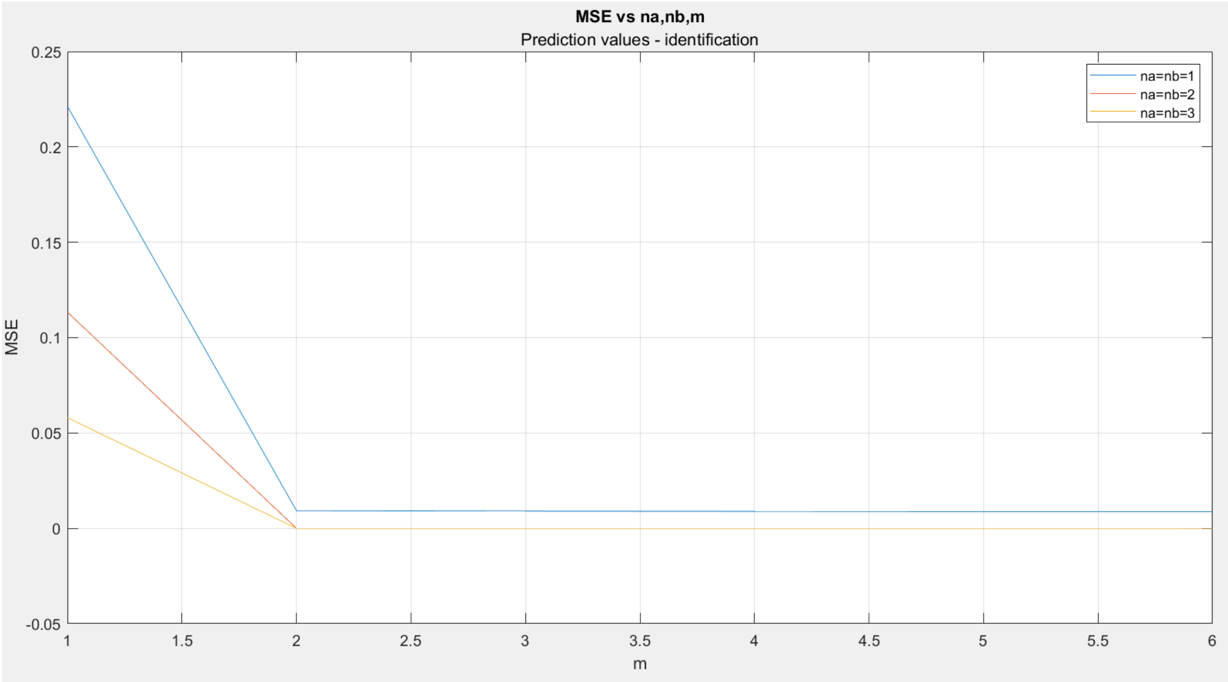


- Observations:
  - few out of range value could be neglected
  - very similar values with the true ones
  - ensuring corrections

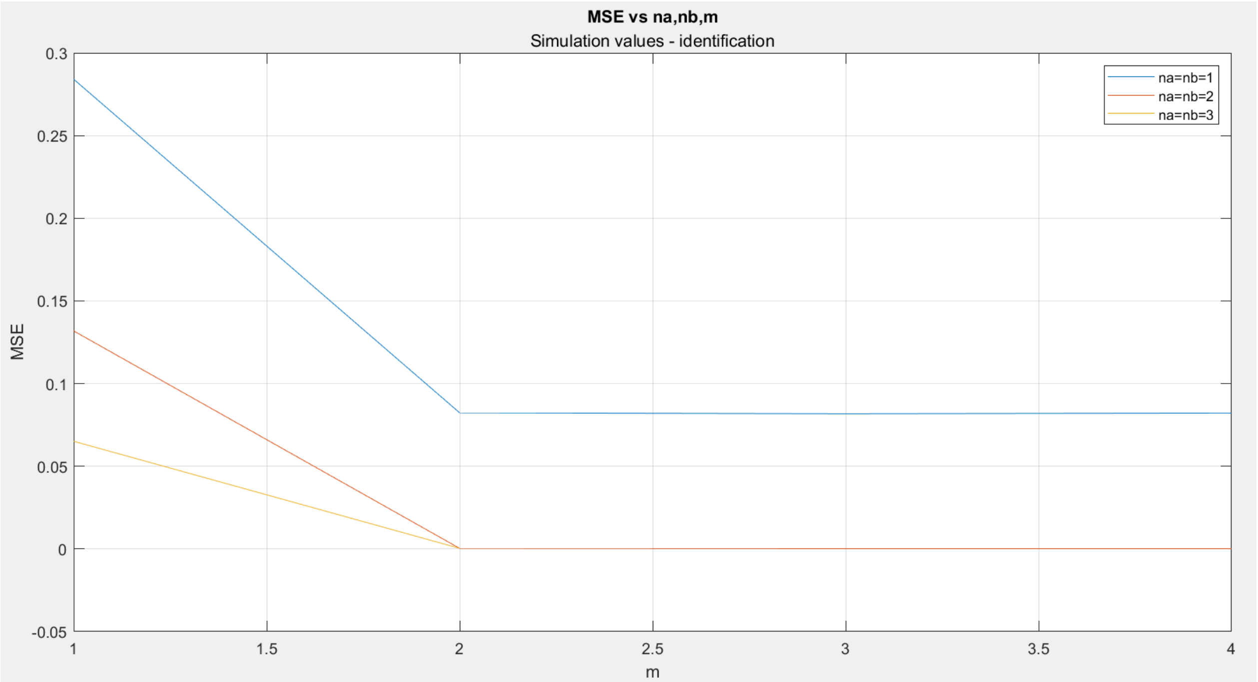
Tuning results

Identification dataset

Prediction



Simulation

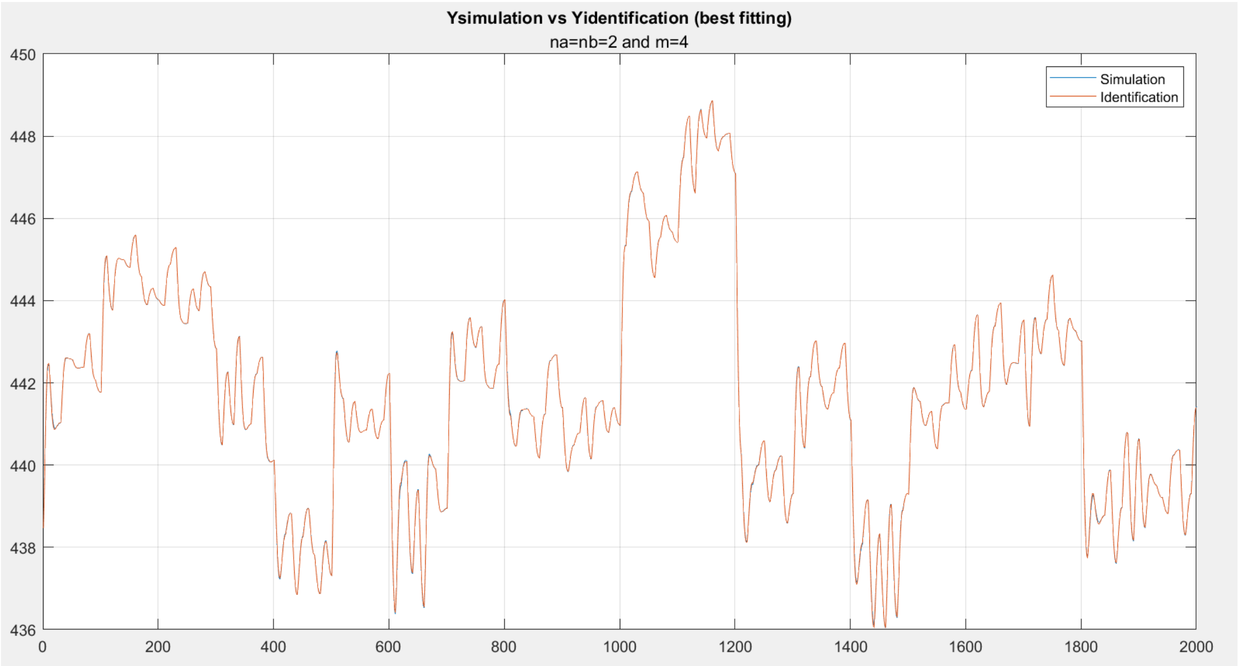


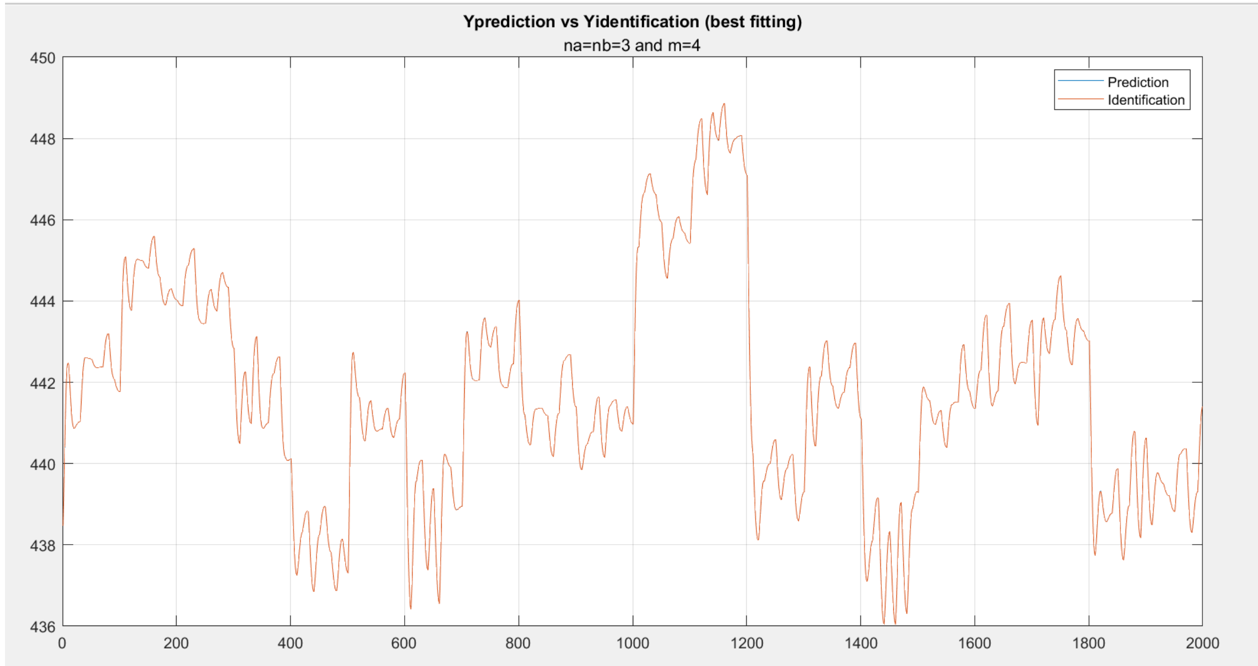
Best values for:

$$\text{MSE} = 1.53 \cdot 10^{-6}\ (n_a = n_b = 3\ and\ m = 4)$$

$$\text{MSE} = 1.65 \cdot 10^{-4}\ (n_a = n_b = 2\ and\ m = 4)$$

Identification dataset



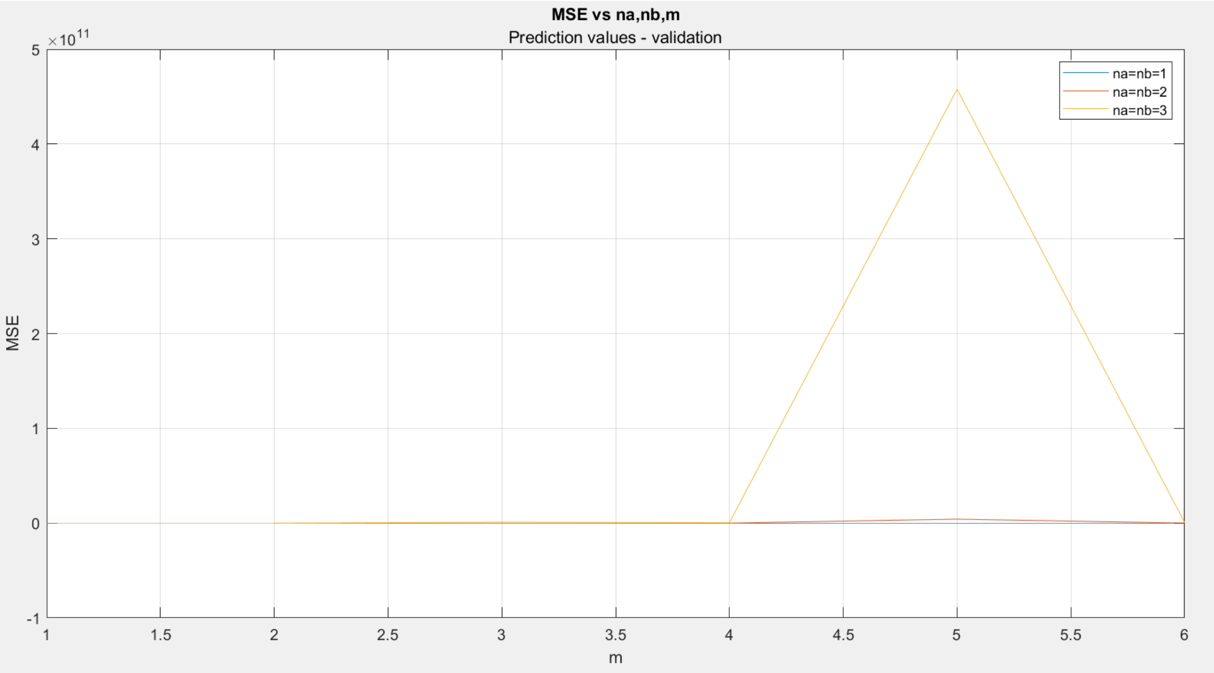


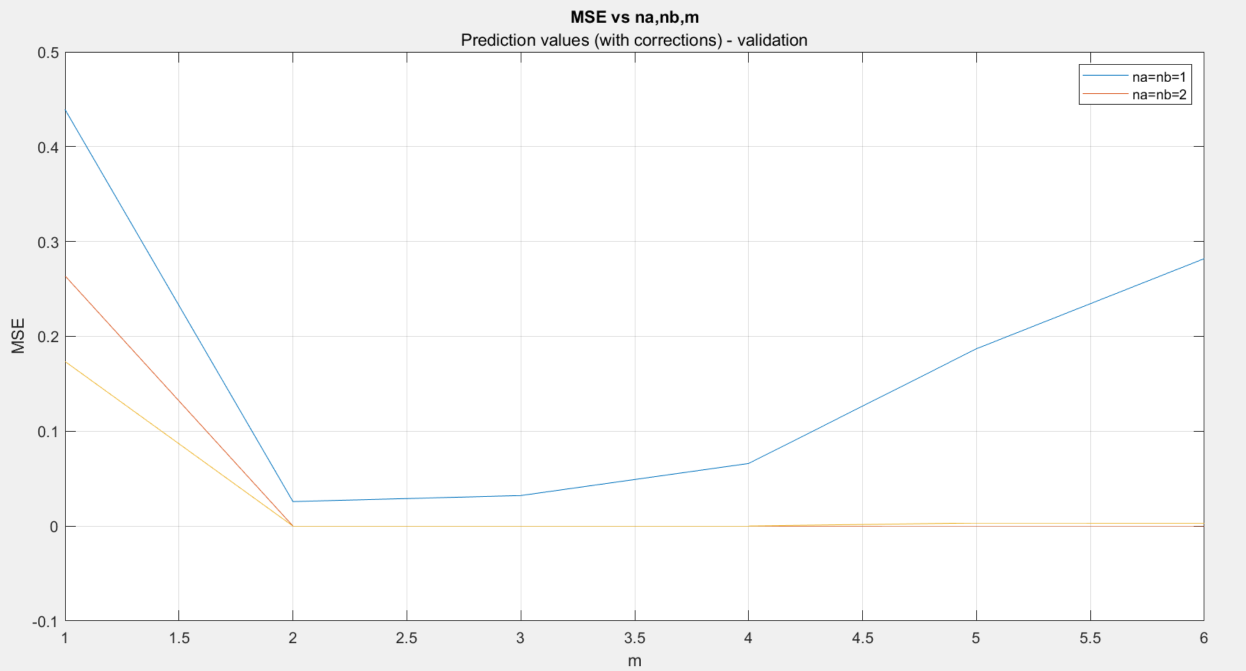
- Observations:
  - overfitting
  - similarity
  - approximately large order

Validation dataset

Prediction without corrections

Prediction with corrections





Best values for:

$\mathrm{MSE} = 0.0316\ (n_a = n_b = 1\ and\ m = 2)$

$\mathrm{MSE} = 2.06 \cdot 10^{-5}\ (n_a = n_b = 3\ and\ m = 2)$

Validation dataset

Simulation

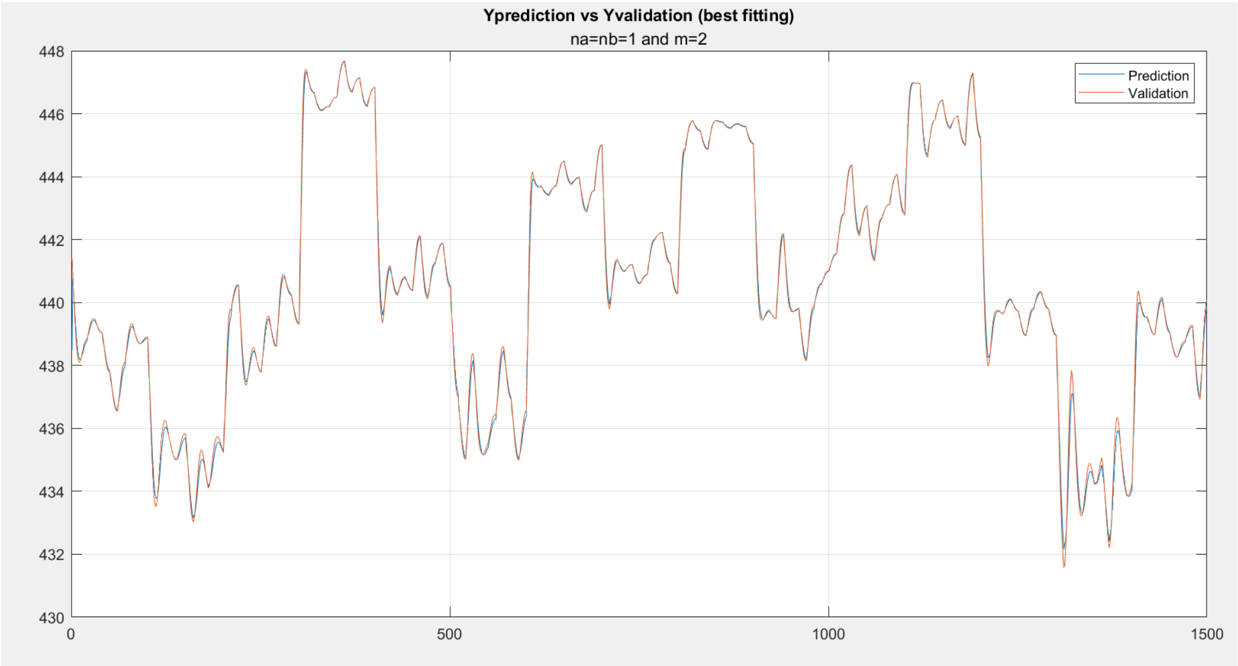
	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$
$n_a = n_b = 1$	0.57177	0.26917	0.82594	NaN	NaN	NaN
$n_a = n_b = 2$	0.32082	NaN	NaN	NaN	NaN	NaN
$n_a = n_b = 3$	0.21043	NaN	NaN	NaN	NaN	NaN

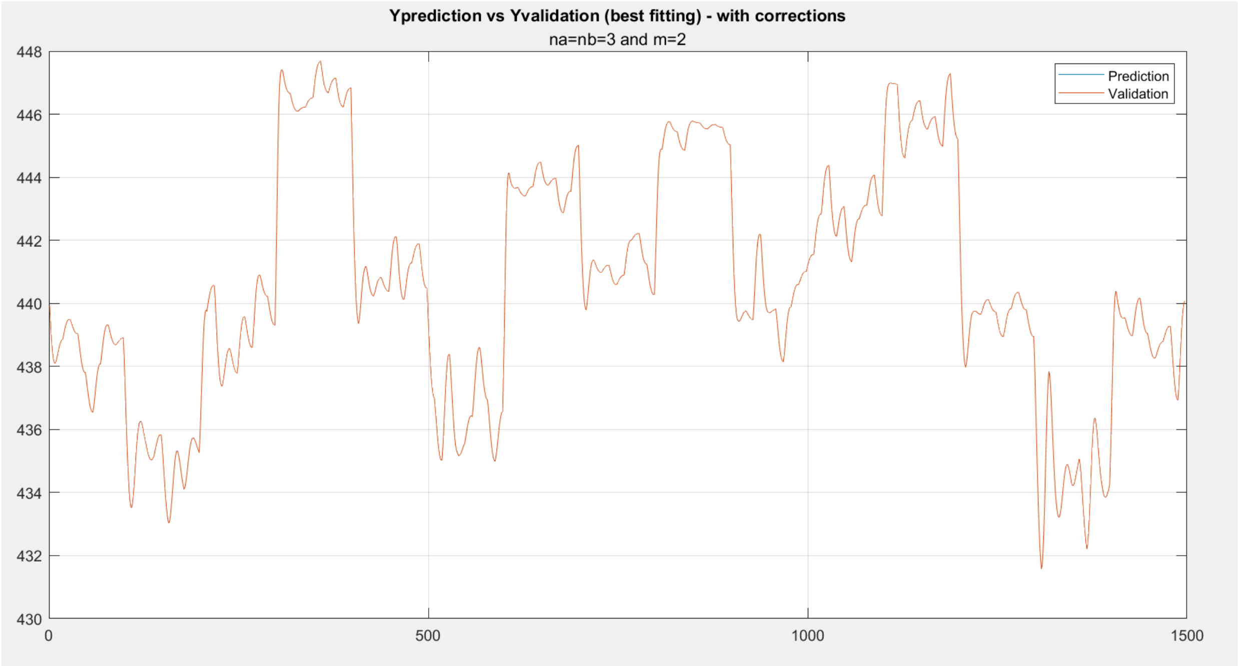
- Observations:
  - unstable model
  - very small values of the parameters

Best values for:

$$\mathrm{MSE} = 0.21043\ (n_a = n_b = 3\ and\ m = 1)$$

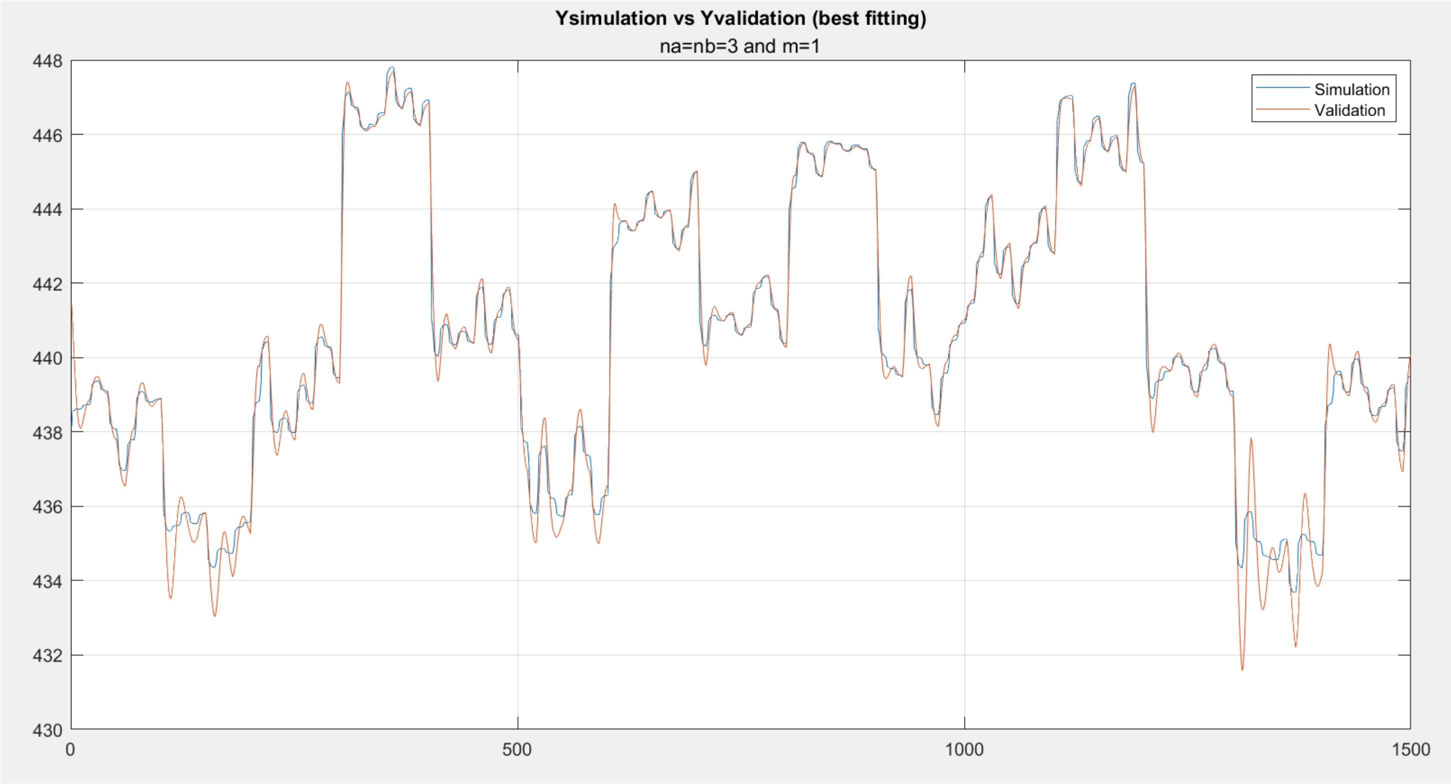
Validation dataset





- Observations:
  - corrections – if the value is not in the range => exclude (plot 2)
  - overfitting (plot 2)
  - small order system (for both orders) (plot 1)

Validation dataset



- Observations:
  - noticeable difference between prediction and simulation
  - large order of dynamics, small order of the polynomial

Conclusion

Prediction vs. Simulation

- Prediction (more knowledge about the model)
  - open-loop system
  - “memory” of the ARX model
  - uses true past output values in order to predict the output at time k
  - precise values (maybe even cases of overfitting)
  - more accuracy => more certainty about the real-time output

- Simulation (less knowledge about the model)
  - closed-loop system
  - uses previously computed values in order to predict the output at time k (dynamical model)
  - approximately good results for small orders
  - very poor results for large orders (leading to instability)
  - can be used even if we know just the inputs of our model

Nonlinear ARX model

- Generalization of the ARX model
- Applied to various dynamical systems
- The prediction itself can be used in a wide range of domains