Linear Regression

Group Index: 38

Group members: Bârzan Emma, Cozac Antonio, Mureșan Mihai

Problem at hand

The aim of this project is to predict a system's response to the relationship between two independent variables such that the error between the approximation and system's real output is minimal.

We have portrayed the implementation of multiple linear regression with two input variables using polynomials as basis functions.

The approximator

The approximator must have the following shape:

$$\hat{y}(k) = \phi^{\mathsf{T}}(k)\theta$$

In a system form, it will be:

$$\begin{bmatrix} y(1) \\ y(2) \\ \dots \\ y(N) \end{bmatrix} = \begin{bmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_n(1) \\ \phi_1(2) & \phi_2(2) & \dots & \phi_n(2) \\ \dots & \dots & \dots & \dots \\ \phi_1(N) & \phi_2(N) & \dots & \phi_n(N) \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \dots \\ \theta_n \end{bmatrix}$$

The regressors $\phi(k)$ will be determined individually for each dataset, but the parameters θ will be determined via:

$$\hat{\theta} = \left(\theta^{\mathsf{T}}\theta\right)^{-1}\theta^{\mathsf{T}}\, Y \quad \equiv \quad \texttt{theta = phi \ id.Y}$$

Key Feature – Regressor implementation

The sum of the powers of the two independent variables in each polynomial are less or equal to the current degree, leading to a combinations list that must be trimmed:

DEGREE	POLYNOMIAL FORM OF APPROXIMATOR		POWERS OF REGRESSOR
$m = 1,$	$\hat{g}(x) = [1,\ x_1,\ x_2] \cdot \theta$	$\Longrightarrow$	$\begin{bmatrix} \text{pow}_{x_1} \\ \text{pow}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$m = 2,$	$\hat{g}(x) = [1,\ x_1,\ x_2,\ x_1^2,\ x_2^2,\ x_1 x_2] \cdot \theta$	$\Longrightarrow$	$\begin{bmatrix} \text{pow}_{x_1} \\ \text{pow}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \end{bmatrix}$
$m = 3,$	$\hat{g}(x) = [1,\ x_1,\ x_2,\ x_1^2,\ x_2^2,\ x_1^3,\ x_2^3,\ x_1 x_2,\ x_1^2 x_2,\ x_1 x_2^2] \cdot \theta$	$\Longrightarrow$	$\begin{bmatrix} \text{pow}_{x_1} \\ \text{pow}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 3 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 2 & 0 & 3 & 1 & 1 & 2 \end{bmatrix}$

Example of trimming:

COMBINATIONS (n=2)	TRIMMED
$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \end{bmatrix}$

Key Feature – Reshaping the dataset output

Output prior to reshape

$$\begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,N} \\ y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,N} \\ y_{3,1} & y_{3,2} & y_{3,3} & \cdots & y_{3,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{N,1} & y_{N,2} & y_{N,3} & \cdots & y_{N,N} \end{bmatrix}$$

Output post reshape

$$\begin{bmatrix} y_{1,1} \\ y_{1,2} \\ y_{1,3} \\ \cdots \\ y_{N,N} \end{bmatrix}$$

Key Feature – Runtime Optimization

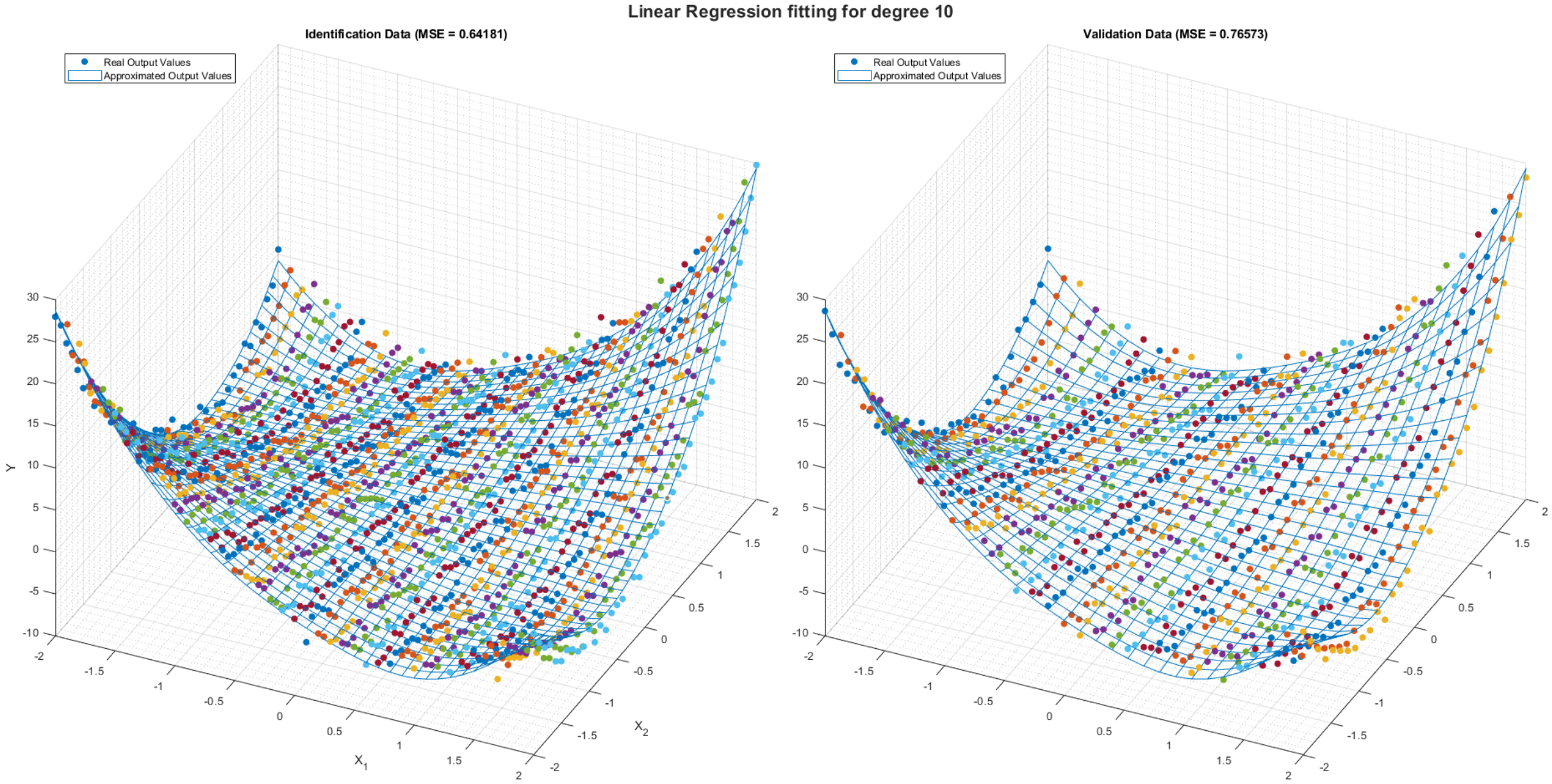
Problem: Large given dataset ($N \times N$) results large regressors ($1 \times N^2$).

Solutions:

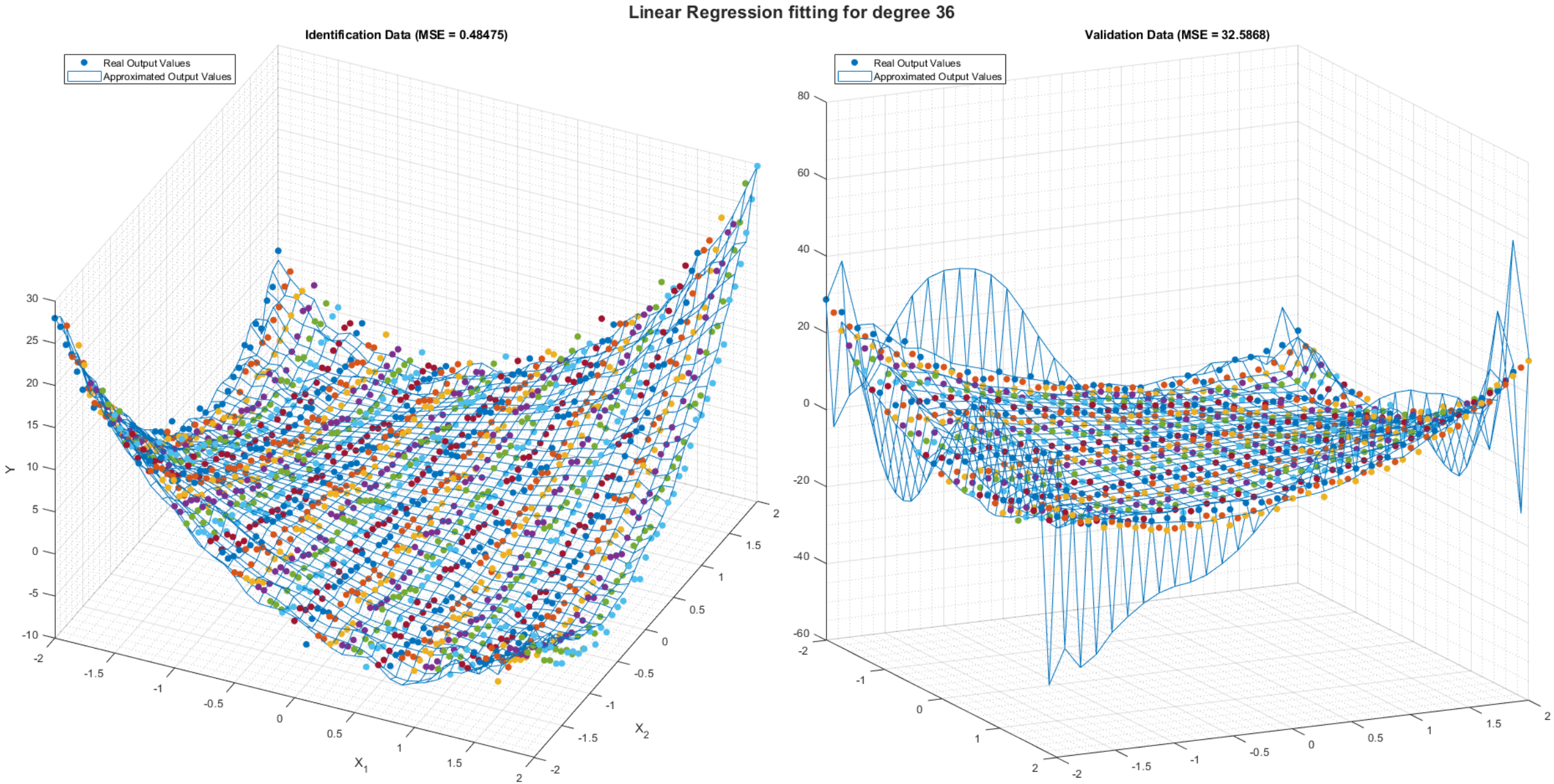
- Utilizing MATLAB's computation speed for vectorization;
- Possible use of multithreading (via parallel pools), but for a small dataset, the performance increase is not significant.

Vectorization only	Vectorization and multithreading
elapsed time: 9 seconds	elapsed time: $6 - 7$ seconds

Tuning results



Tuning results – overfitting



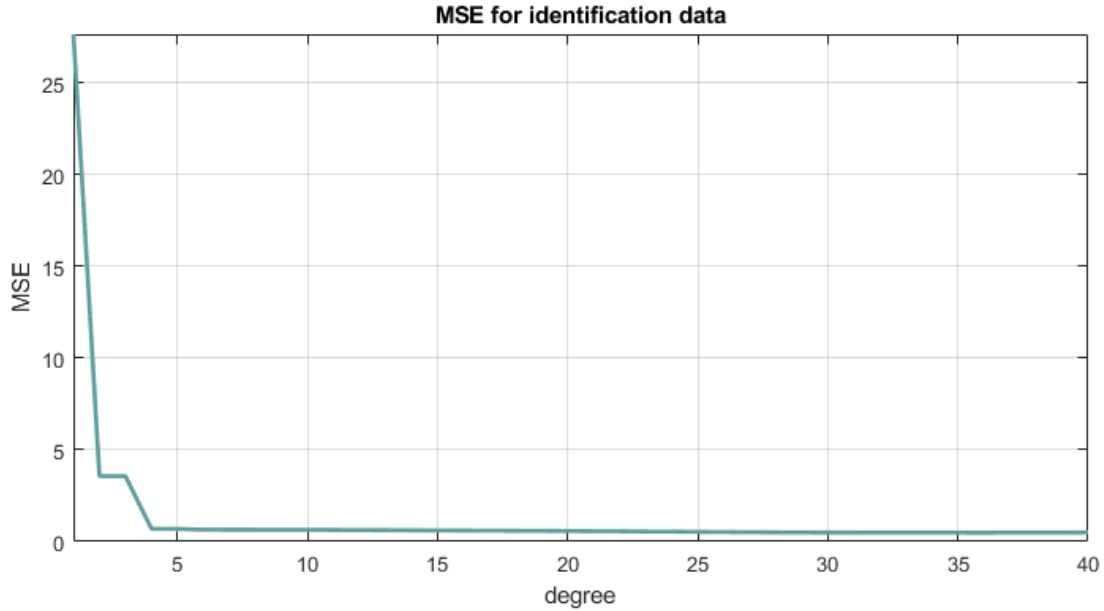
Tuning results

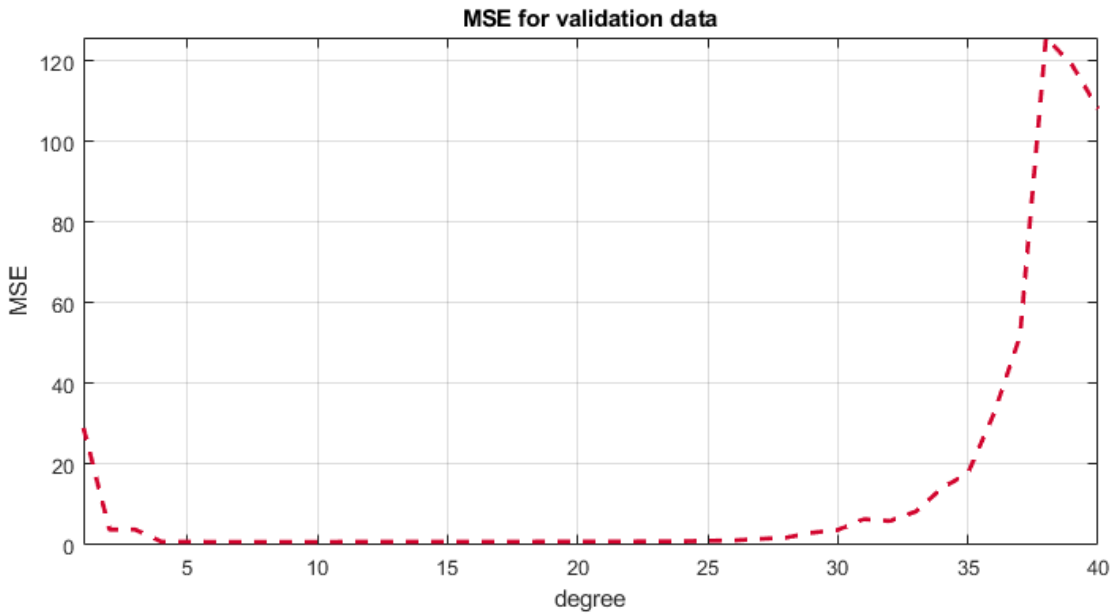
- 1 to 3: unreliable model
- 6 to 10: best cases
- ≥25: major increase in the validation dataset's MSE

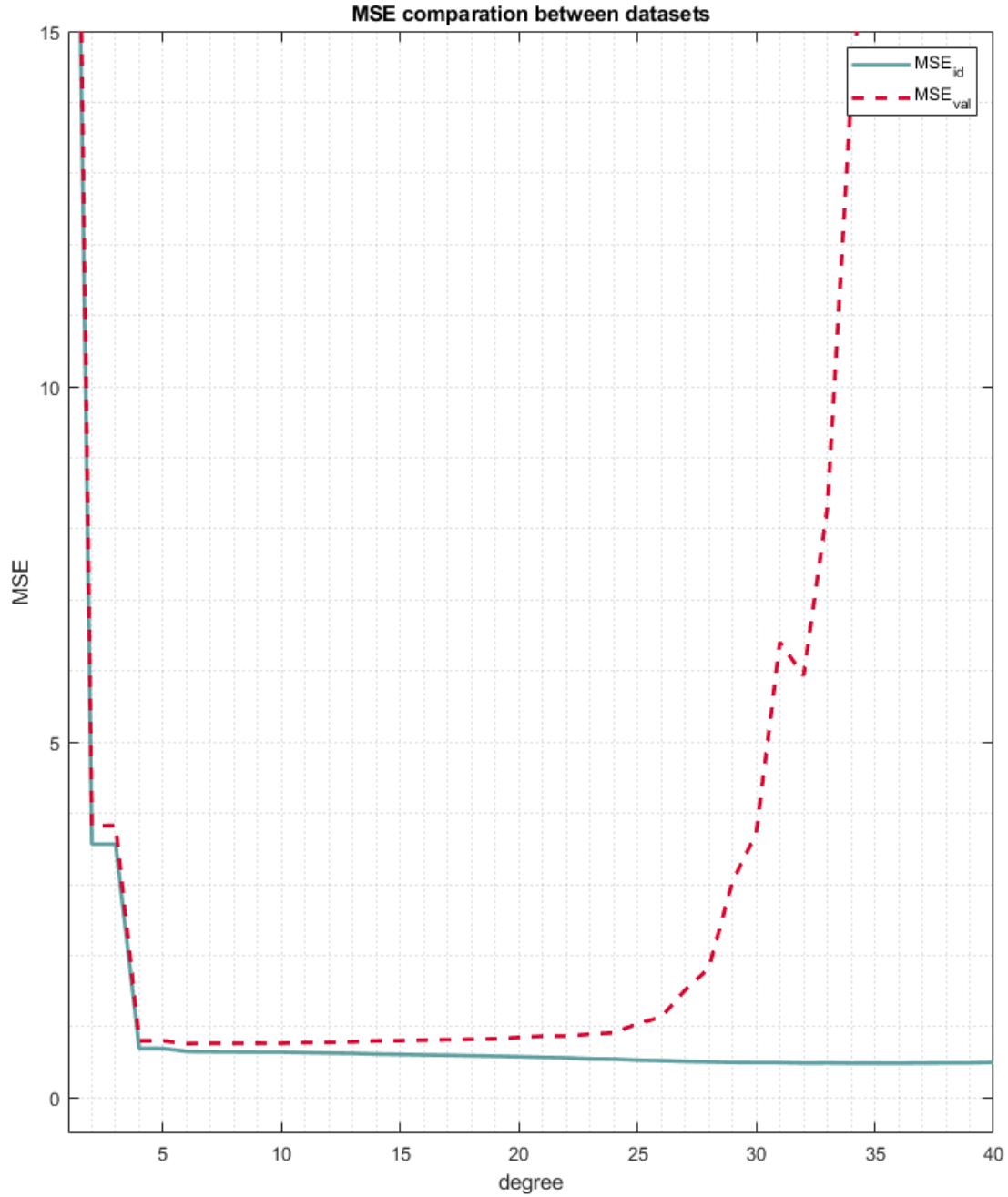
Degree	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MSE_id	27.629	3.567	3.566	0.694	0.693	0.650	0.646	0.644	0.643	0.642	0.635	0.628	0.623	0.614	0.609	0.602	0.597	0.591	0.585	0.577
MSE_val	29.088	3.825	3.829	0.800	0.801	0.761	0.767	0.766	0.767	0.766	0.777	0.780	0.787	0.800	0.802	0.810	0.817	0.821	0.829	0.848

Degree	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
MSE_id	0.568	0.561	0.547	0.544	0.528	0.521	0.509	0.504	0.497	0.496	0.494	0.487	0.490	0.486	0.487	0.485	0.488	0.490	0.491	0.498
MSE_val	0.866	0.867	0.894	0.912	1.041	1.135	1.512	1.820	3.041	3.715	6.405	5.939	8.303	14.107	17.807	32.587	51.706	125.699	118.946	108.066

MSE for varying degree







Conclusion

Albeit this model can accurately determine the response of the system, this implementation is slow and there is a substantial error between the expected output and the actual response.

Any degrees from 6 to 10 result in a low approximation error. Anything above this will decrease the MSE for the training data but will be disastrous for the testing data.