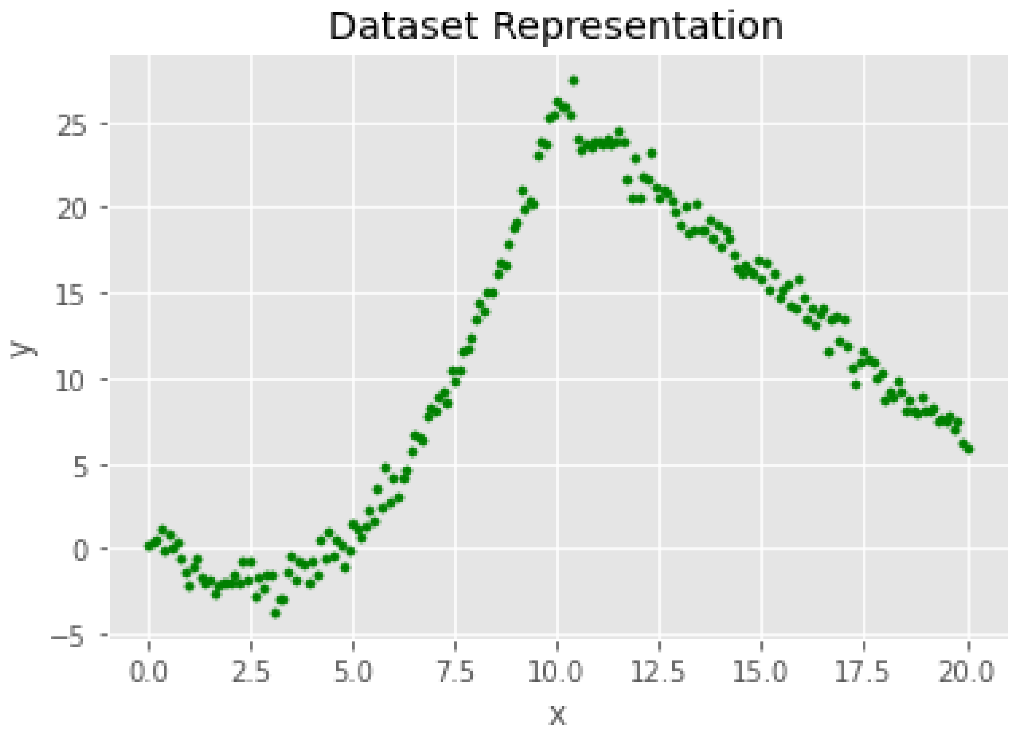
Advanced Subjects in Signal Processing

Project 3 Presentation
Evangelos Siatiras
AM: EN2190001

First Goal: Estimating the Unknown f(·)

- Investigate through the data
- In several Dimensions the goal is to produce a Curve that fits the Data (Curve Fitting)
- We will do Regression to produce a Decision Surface f(x)
- The Regression Method Chosen is the Least Squares Method



Least Squares Method (1)

- In Least Squares the goal is to find a curve that minimizes the sum of the Squared Residuals.
- We Define a cost function representing the above as:

$$\boldsymbol{J(\theta) = \sum_n (y_n - \underline{\theta^T}\,\underline{x_n})^2} \Leftrightarrow$$

$$J(\theta) = \sum_n y_n^2 - 2\left(\theta^{\mathrm{T}} x_n\right) y_n + \left(\underline{\theta^{\mathrm{T}}}\ \underline{x_n}\right)\left(\underline{\theta^T}\ \underline{x_n}\right) = \sum_n [y_n^2 - 2\left(\underline{\theta^{\mathrm{T}}}\ \underline{x_n}\right) + \underline{\theta^{\mathrm{T}}}(\underline{x_n}\ \underline{x_n^{\mathrm{T}}})\underline{\theta}]$$

- So the goal is to find the θ_s such minimizing the cost function, so we take the derivative of the quantity above equal to zero

$$\frac{dJ}{d\theta} = 0 \Leftrightarrow \sum_n y_n \underline{x_n} - \sum_n \underline{x_n}\ \underline{x_n^T \theta} = 0 \Leftrightarrow \sum_n y_n \underline{x_n} = \sum_n \underline{x_n}\ \underline{x_n^T}\ \underline{\theta}$$

- Note that : All quantities are known except the vector θ so the system is solvable.

Remarks:

$\underline{\theta^{\mathrm{T}}}$:number

$\underline{x_n}$:number

So

$\left(\underline{\theta^{\mathrm{T}}}\ \underline{x_n}\right) = (\underline{x_n}\ \underline{\theta^{\mathrm{T}}})$

Least Squares Method (2)

- In Matrix form we have to compare $y \rightarrow x \cdot \underline{\theta}$

$$J(\theta) = \left\| \underline{y} - x\underline{\theta} \right\|^2 = \left(\underline{y} - \underline{\theta^T} x^T\right)\left(\underline{y} - x\underline{\theta}\right) = y^T y + \underline{\theta^T} x^T x \underline{\theta} - \underline{y^T} x \underline{\theta} - \underline{\theta^T} x^T \underline{y}$$

- And by minimizing the above quantity we get the closed form solution:

$$\frac{dJ}{d\theta} = 0 \Leftrightarrow x^T x \underline{\theta} - x^T y = 0 \Leftrightarrow$$

$$\boldsymbol{\hat{\theta} = (x^T x)^{-1} x^T y}$$

Polynomial Regression (1)

- Goal: Fit a n_{th} degree polynomial to data to establish and general relationship between the independent variable x and dependent variable y.
- Objective: Minimizing the cost function given by:
- $J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(h_\theta(\chi^\iota) - y^i)^2$
- The hypothesis h_θ is given by the linear model:
- $h_\theta(x) = \theta^T x = \theta_0 \mathrm{x} + \theta_1 x^2 + \theta_2 x^3 + \cdots + \theta_n x^n$
- Note that in general there is no particular way to know if the hypothetical model been chosen in the beginning is the right or the optimal. It is a decision-based parameter very critical for the performance of the model.
- The $PolynomialRegression$ class can perform polynomial regression using the closed form solution to linear regression.

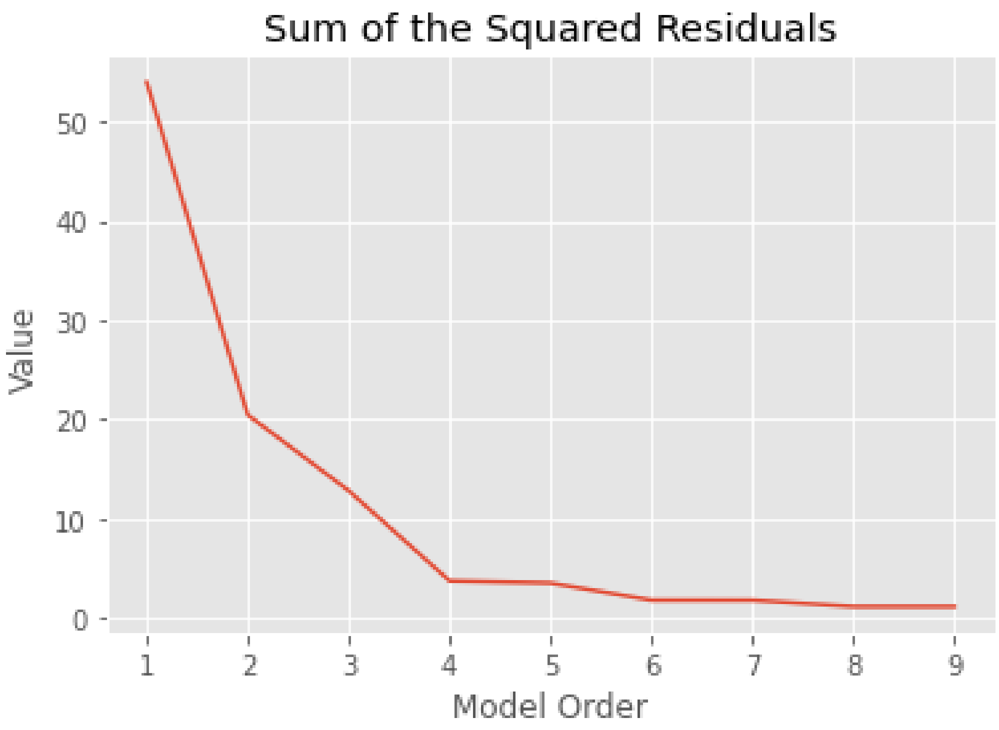
$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T\boldsymbol{y}$$

Polynomial Regression (2)

- First Step is to train our model => to find the function reproduces with the correct way and then with the optimal way the behavior of our dataset.

Polynomial Regression (3)

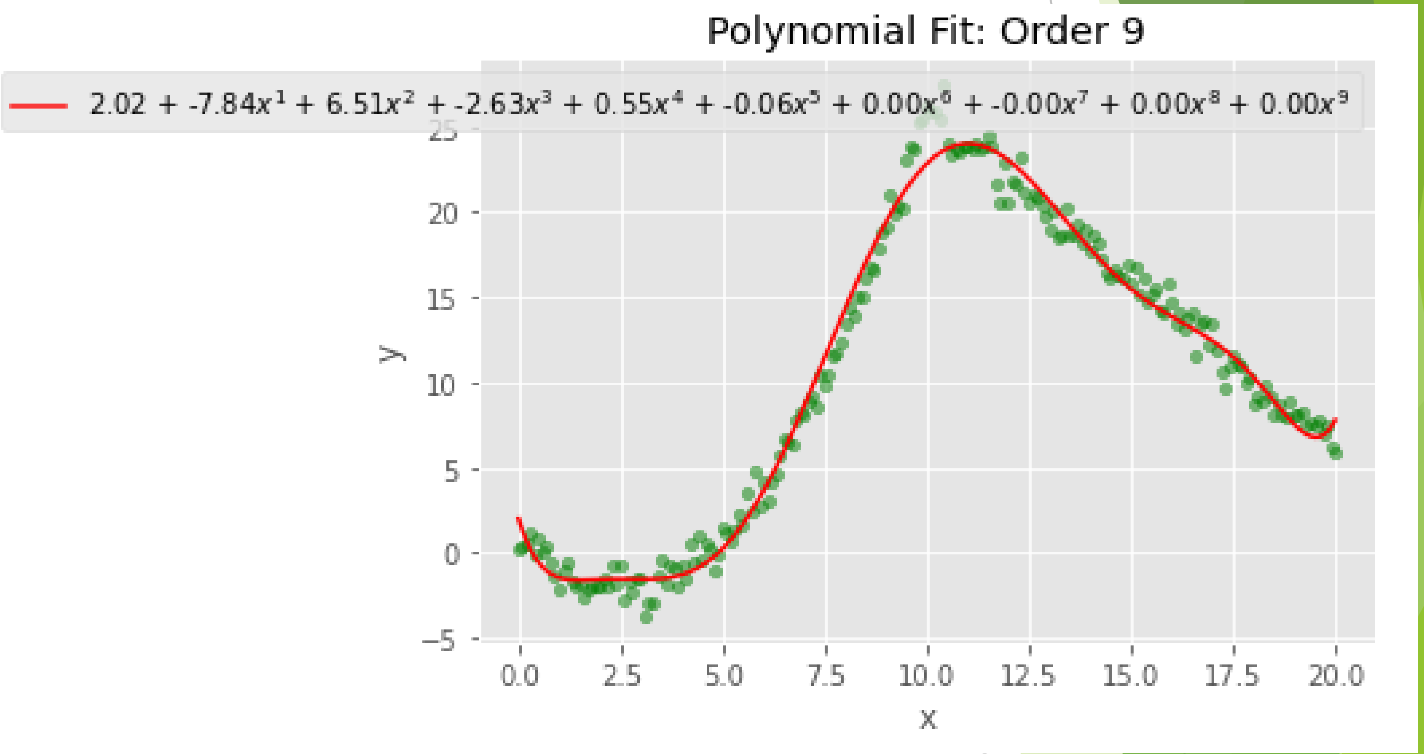
- Second Step is to evaluate the performance of the cost function for different orders of our Model.
- Note that: The best model (in terms of performance) for given order is the one with the minimum value of the cost function.
- In our case the Min value of the cost function is achieved for Polynomial of Order 9 and is: 1.142



Polynomial Regression (4)

Remarks:

- Sometimes by adjusting the parameters of the model in order to achieve the minimum MSE from evaluating only from the training set is not recommended.
- If we introduce some new data to the model (a test set) we will see that the model will underfit the data and will perform very bad.
- By increasing the model order after 5 the model starts to overfit the data and that’s why the θ coefficients have very small values.



Channel Estimation Using Least Square Algorithm (1)

- Consider a MIMO channel characterized by H,S as the training sequence, Y as the related received signal and N representing the Additive White Gaussian Noise.
- Assuming: $Y = SH + N$
- LS Estimator finds a prediction of the H called H_{pred} in such a way that minimizes the Euclidean Distance of $S \cdot H\_pred\ - \ Y$
- Assuming $H_{pred} = H$ the Euclidian distance is $\left\| S\hat{H} - \ Y \right\|^2$
- For the minimization we expand the above quantity
- $(S\hat{H} - Y)^{H}(S\hat{H} - Y) = \ (S\hat{H})^{H}(S\hat{H}) - Y^{H}S\hat{H} - (S\hat{H})^{H}Y + Y^{H}Y$
- And taking the derivative with respect to H and set it equal to zero and thus constructing the closed form equation to linear regression
- $2S^{H}SH\ - 2S^{H}Y = 0 \rightarrow S^{H}S\hat{H} = S^{H}Y \Leftrightarrow \boldsymbol{\hat{H} = \ (S^{H}S)^{-1}S^{H}Y}$

Channel Estimation Using Least Square Algorithm (2)

Parameters:

- 4 Transmit and 4 Receive Antennas ⇒ 4 Channels
- The Dimensions of the Predicted H are $N_R \times \ N_T = 4 \ \times 4$
- 500 transmitted Symbols Per Antenna so totally 2000

Implementation:

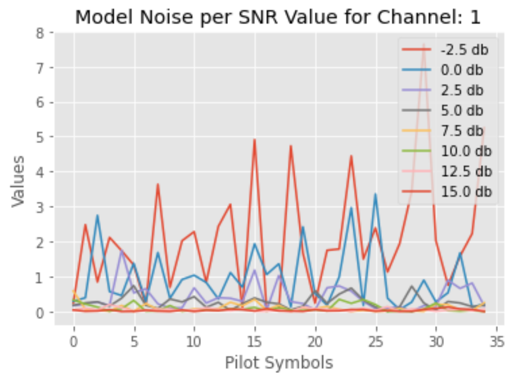
- $LS\_Estimate$ performs regression using the closed form equation and thus constructs the coefficient vector H for given number of points. The H constructed from all the 500 symbols per Channel is called H_500 and Similarly H_300 and H_100.
- The H_* is a dictionary with 8 keys (for every received signal) and with values 4×4 matrices, the coefficient vectors constructed using the symbols of every received signal

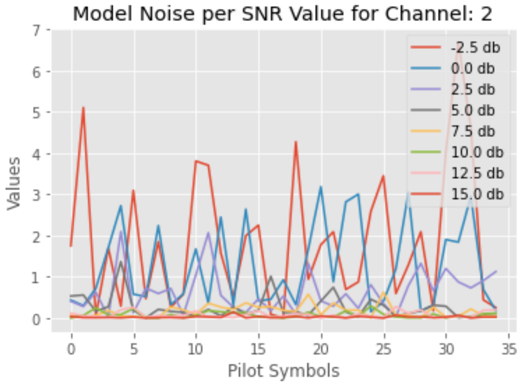
Channel Estimation Using Least Square Algorithm (3)

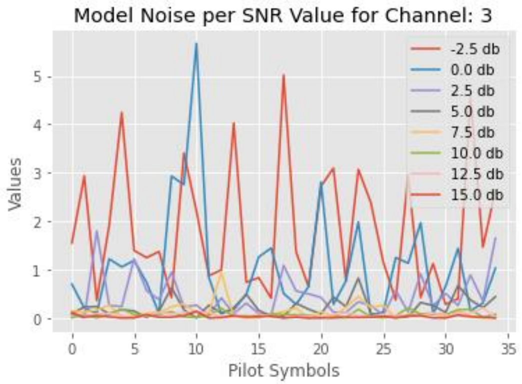
- Recap: The 8 Receiving signals are a product of the transmitted signal added by a supplementary noise depending on the SNR value.
- SNR value is small ⇒A lot of noise in the model.
- SNR value is big

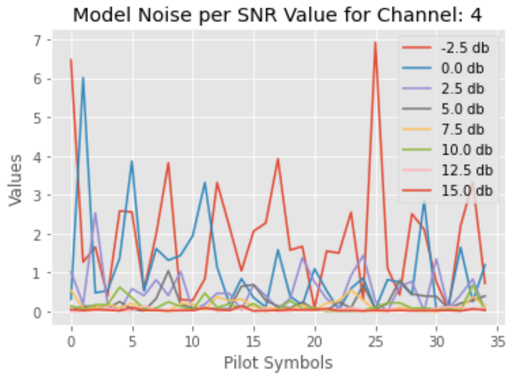
  ⇒Transmitted signal is close to the received signal

  ⇒noise in the model tends to be zero









Channel Estimation Using Least Square Algorithm (4)

- Performance metric to Evaluate the whole System:

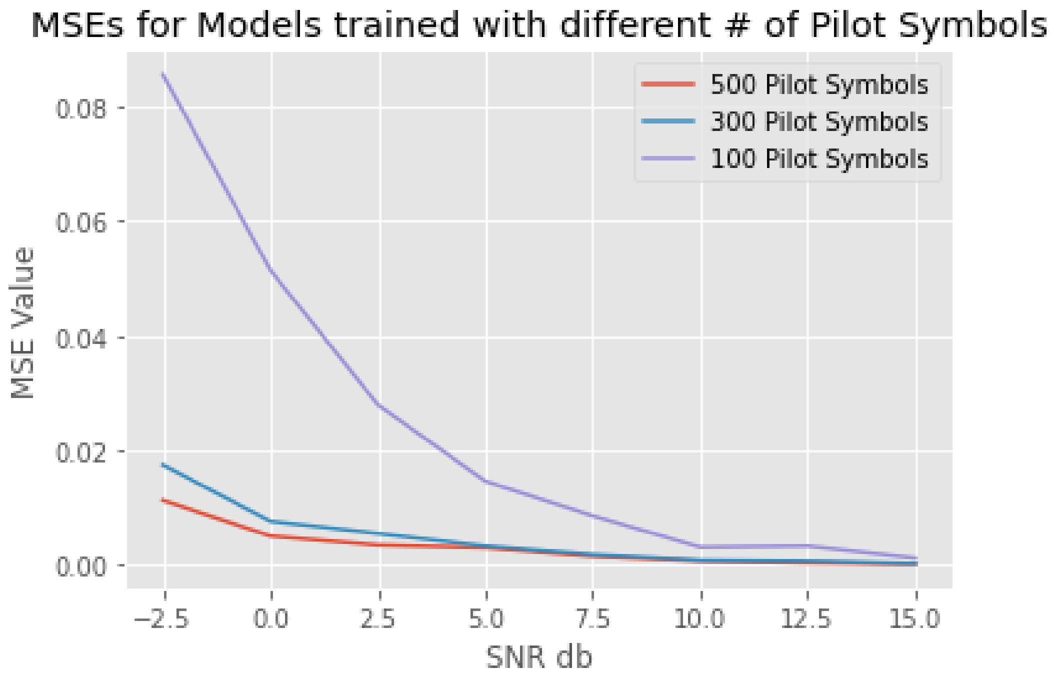
  MSE (Mean Squared Error) $= \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(|\boldsymbol{H_{true}S} - \boldsymbol{H_{pred}S}|)^{\boldsymbol{2}}$

- Note that H_{pred} has been calculated for every received signal Y_i denoted by different values of SNR.
- An MSE Value will be calculated for the corresponding SNR Value.

Channel Estimation Using Least Square Algorithm (5)

Remarks:

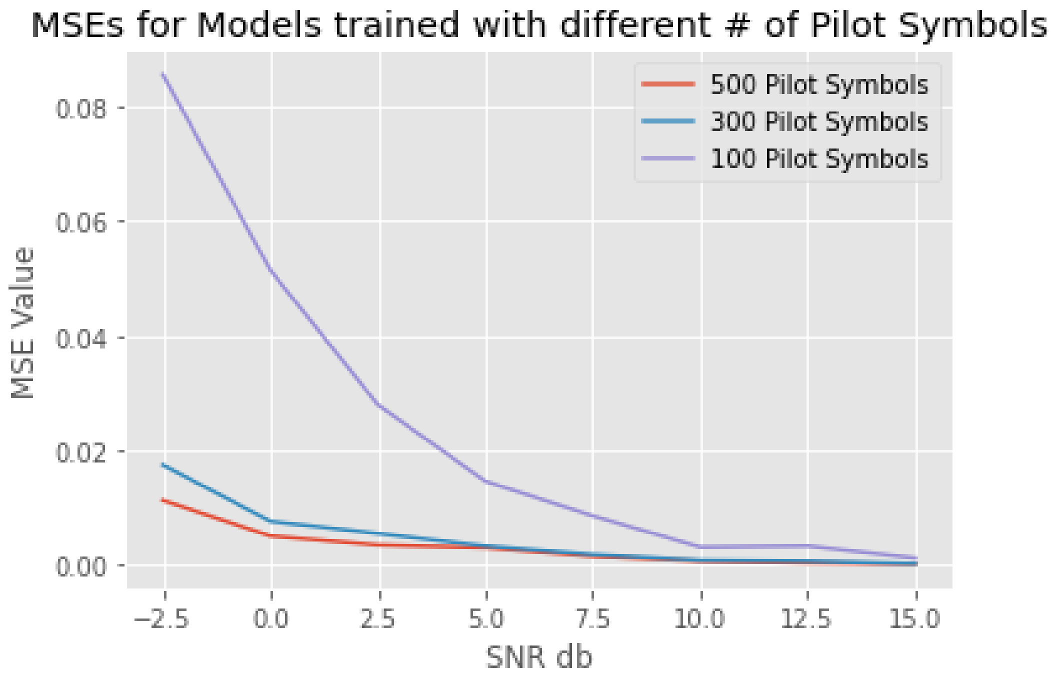
- We could evaluate our system using the received signals Y .In such case the results corresponds to how close is the prediction to the received signal with the added white gaussian noise due to transmission.
- We want to evaluate our system with the transmitted signal.
- In the training phase our goal is to be able to construct a parameter vector H in order to predict the received signal and thus we use Y.



Channel Estimation Using Least Square Algorithm (6)

Remarks:

- For the three models, for increasing SNR value the MSE is decreasing thus the performance of the model is Increasing.
- In low signal to noise ratio (SNR) regime, the channel estimates are overwhelmed by random noise.
- The number of received symbols that the model is trained is important. The Smaller the dataset the Bigger the underfitting of the data thus Bigger the MSE.



Noise Cancellation Using the Wiener Filter (1)

- Goal: Design a Wiener filter that filters the provided noise $u2$ and produces an estimate of the noise $u1$. This noise estimate is subtracted from the corrupted signal y to produce an estimate of the signal of interest.
- The estimate $\widehat{u1} = \sum_{l=0}^{p-1} w(l)u_2(n-l)$ p: # Filter Coefficients
- The autocorrelation function of a WSS signal is defined as:

$$r_{u2}(k) = E\left\{u_2(n)u_2^*(n-k)\right\}$$

- The cross-correlation function for jointly WSS signals is defined as

$$r_{u1\,u2} = E\left\{u_1(n)u_2^*(n-k)\right\}$$

- The desired signal can be defined as the difference between the corrupted signal and the estimate of the added noise signal as

$$\mathrm{s}(n) = y(n) - \widehat{u1}(n)$$

Noise Cancellation Using the Wiener Filter (2)

- The optimal Wiener Filter Coefficients by solving a linear set of equations defined as

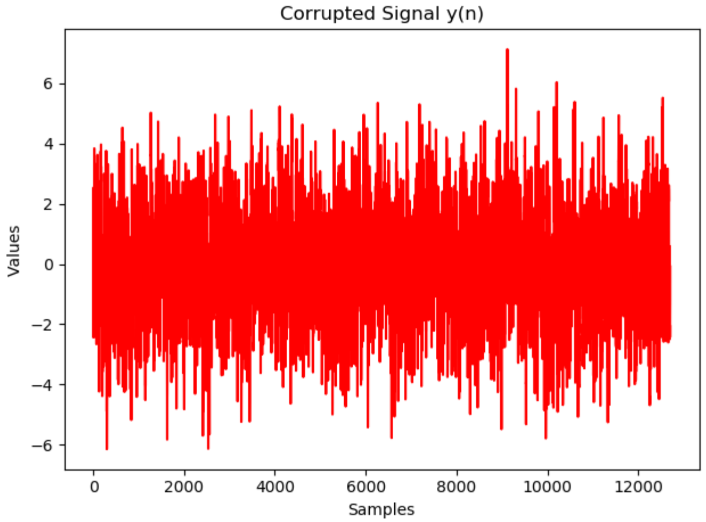
$$w_{opt} = R_{u2}^{-1} \cdot r_{y\,u2}$$

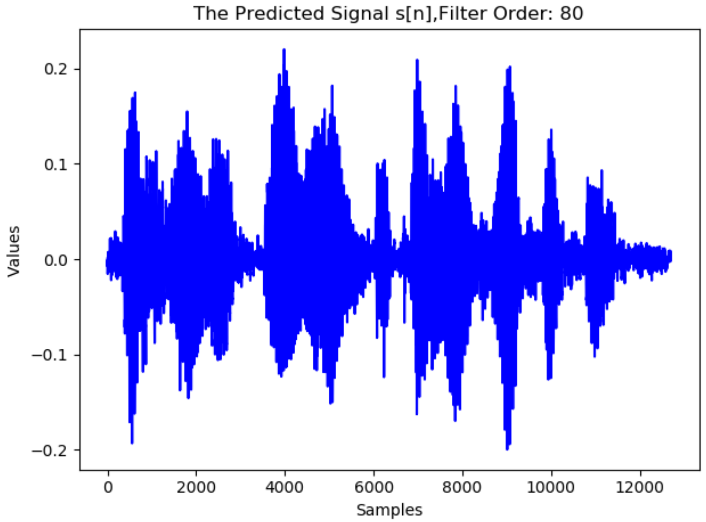
- Note that R_{u2} is the Toeplitz matrix of the autocorrelation matrix $r_{u2}(k)$

Noise Cancellation Using the Wiener Filter (3)

Remarks

- As increasing the order of the wiener filter there is a significant improvement.
- For Filter of order 80 the voice can be heard clearly but there is still some added noise in the signal.
- A Performance indicator is still needed in order to be able to evaluate and compare the results especially in applications require high sensitivity.





Thank You!