Sharif University of Technology

Masoud Tahmasbi Fard

Student ID: 402200275

EE181: Stochastic Processes

Computer Assignment #3

January 27, 2024

Table of Contents

This assignment for the Stochastic Processes course consisted of 3 questions focused on different applications of stochastic signal processing. The first question examined spectral estimation of a random noise process. This required analyzing properties of the autocorrelation function to estimate the power spectral density.

The second question looked at filtering - specifically, estimating a signal $s[n]$ from observations of a related process $x[n]$. Filtering techniques allow signal extraction in the presence of noise.

Finally, the third question involved implementation of a Kalman filter to dynamically estimate the state of a random process over time. The Kalman filter recursively updates the state estimate by incorporating predictions as well as corrective feedback from noisy observations.

1 Spectral Estimation

1.1 Part A.

Part A focused on estimating the autocorrelation function of a provided noise data.

Figure 1 illustrates the empirical autocorrelation calculated directly from the noise measurements over time. As observed, the autocorrelation peaks at zero lag - indicating maximum correlation between a noise sample and itself.

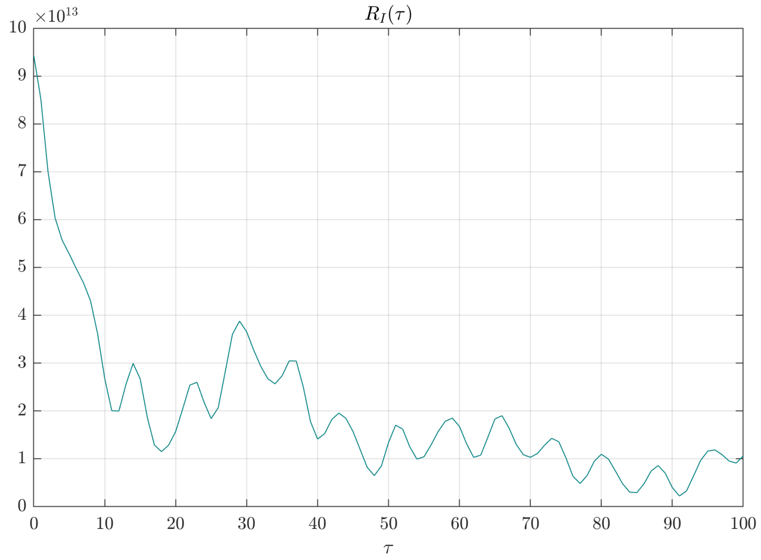


Figure 1: Autocorrelation function of the noise

1.2 Part B.

Part B involved fitting autoregressive (AR) models to the noise data to characterize the stochastic process. Specifically, we modeled the noise using $AR(2)$ and $AR(4)$ models, which estimate each sample as a linear function of the previous 2 and 4 samples respectively.

To extract the AR coefficients, we utilized the `ar(data,order)` function in MATLAB. By default, the *ar* function in MATLAB utilizes a forward-backward linear prediction approach to minimize error. It first fits an auto-regressive model in the forward direction by minimizing the prediction error for future values based on past samples. This results in a least-squares fit over the measured time interval.

Additionally, *ar* fits a reverse linear model predicting each point from future points, again minimizing the least-squares error. This backward linear prediction helps control error growth outside the measured interval.

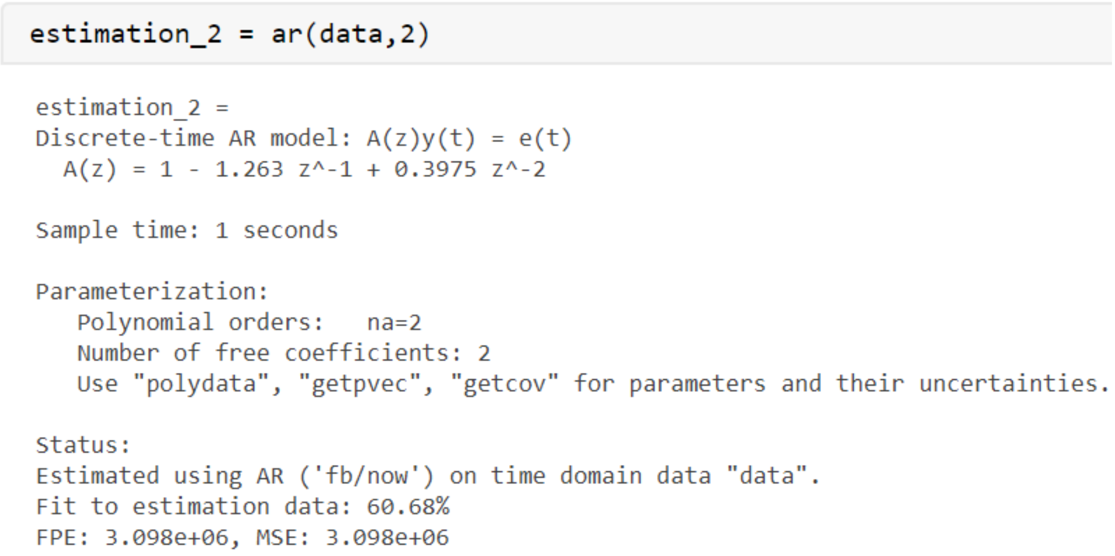
Finally, the forward and backward AR models are combined to produce a final result that accounts for past and future samples.

Furthermore, the *ar* function supports pre-windowing and post-windowing options to apply tapering weights prior to and after the measured time series. Adding such tapers reduces discontinuities at the boundaries. This complements the forward-backward approach for controlling edge effects.

Figures 2 and 3 illustrate the resulting coefficient values for the 2nd order and 4th order models. Examining the relative coefficient magnitudes provides insight into the correlation structure, indicating the extent of linear dependence between samples at different lags. The AR modeling complements the direct autocorrelation analysis in visualizing the noise process dynamics.

1.3 Part C.

After fitting AR models in Part B, Part C focused on examining the spectral properties of the estimated models. Figures 4 and 5 illustrate the spectral density (PSD) derived from the $AR(2)$ and $AR(4)$ coefficients respectively.



```
estimation_2 = ar(data,2)
```

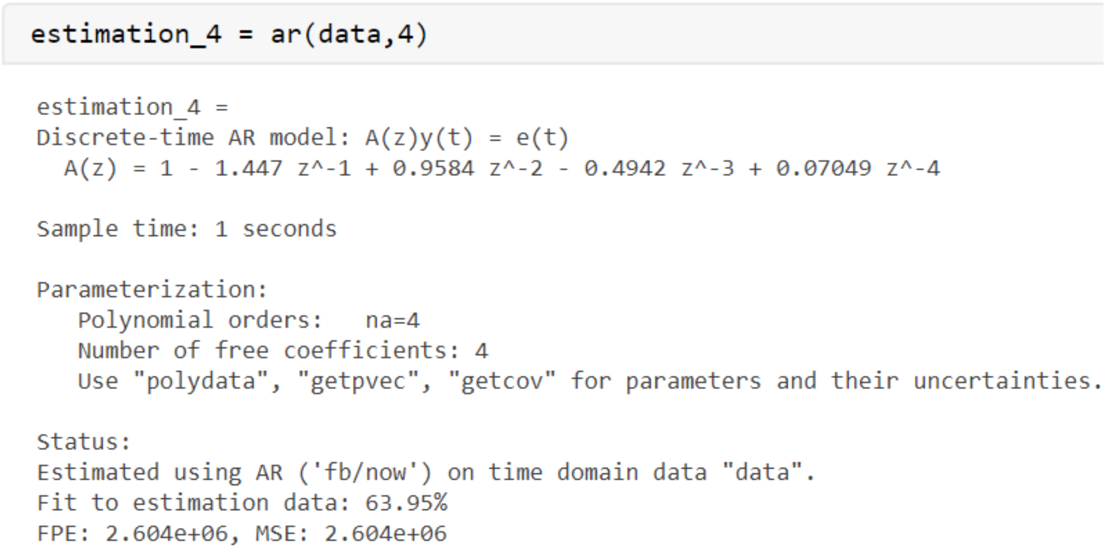
```
estimation_2 =
Discrete-time AR model: A(z)y(t) = e(t)
  A(z) = 1 - 1.263 z^-1 + 0.3975 z^-2

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   na=2
   Number of free coefficients: 2
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using AR ('fb/now') on time domain data "data".
Fit to estimation data: 60.68%
FPE: 3.098e+06, MSE: 3.098e+06
```

Figure 2: $AR(2)$ Estimation



```
estimation_4 = ar(data,4)
```

```
estimation_4 =
Discrete-time AR model: A(z)y(t) = e(t)
  A(z) = 1 - 1.447 z^-1 + 0.9584 z^-2 - 0.4942 z^-3 + 0.07049 z^-4

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   na=4
   Number of free coefficients: 4
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using AR ('fb/now') on time domain data "data".
Fit to estimation data: 63.95%
FPE: 2.604e+06, MSE: 2.604e+06
```

Figure 3: $AR(4)$ Estimation

Importantly, the provided noise is not white, as evidenced by the non-flat spectrums. The $AR(4)$ spectrum is "tighter" than the $AR(2)$ version, with more pronounced resonances within a lower bandwidth. The richer harmonic content of the $AR(4)$ model provides a better fit, coming closer to capturing the true noise spectrum characteristics.

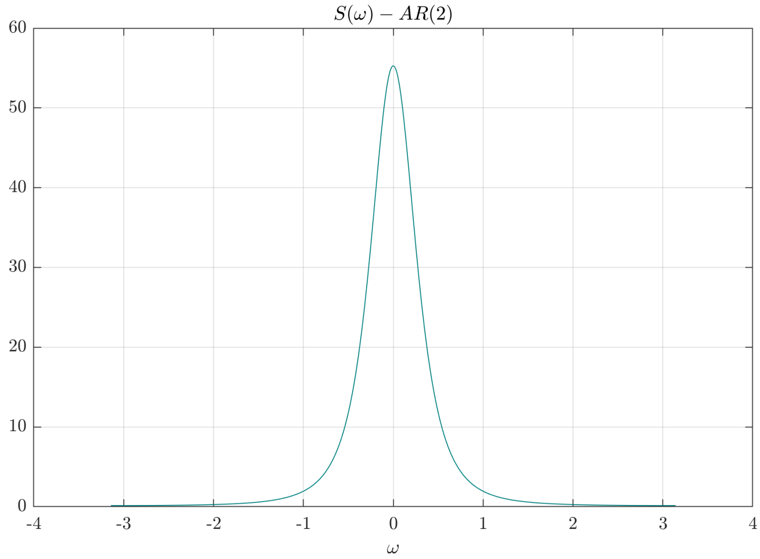


Figure 4: Spectrum of $AR(2)$ estimation

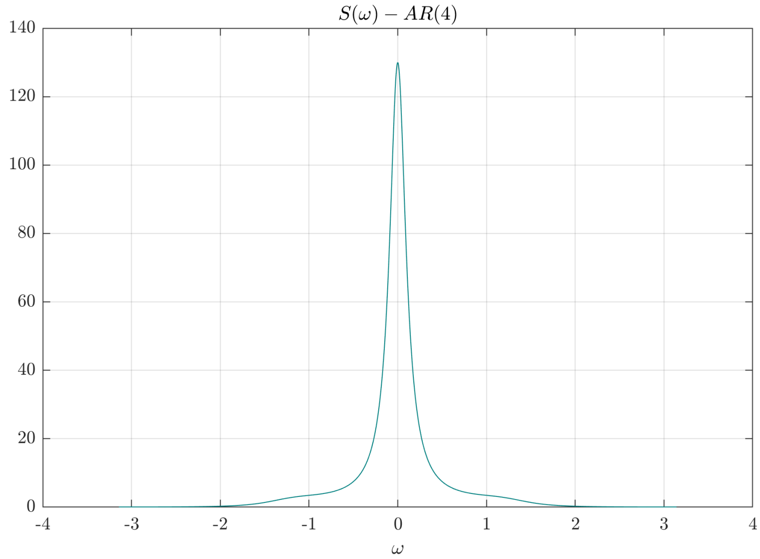


Figure 5: Spectrum of $AR(4)$ estimation

1.4 Part D.

Part D involved estimating the power spectral density (PSD) of the noise process via the periodogram method. The periodogram is a simple nonparametric spectral estimation technique based on the Fourier transform.

Mathematically, the periodogram is defined as:

$$S(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j2fn} \right|^2$$

Where $S(f)$ is the PSD estimate at frequency f, $x[n]$ is the noise time series data, and N is the number of samples. Essentially, it calculates the discrete Fourier transform and takes the modulus squared to derive a spectral estimate.

To obtain the periodogram visualization in Figure 6, MATLAB's built-in *periodogram* function was utilized on the noise data.

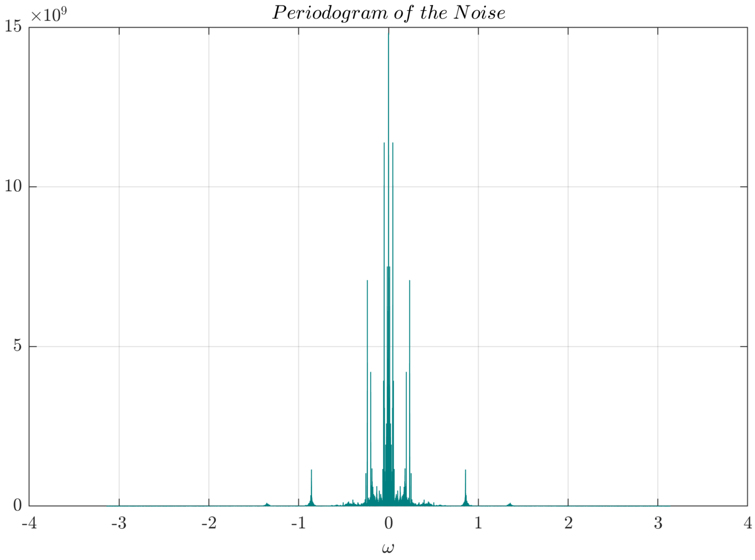


Figure 6: Periodogram of the noise

As seen in Figure 6, the periodogram produces a noisier PSD estimate compared to the AR-based approach in Part C. However, it does capture a similar overall distribution with resonance peaks and nulls indicating the presence of correlation over time.

1.5 Part E.

Part E explored Bartlett's method - a time-averaged periodogram approach for power spectral density estimation. In contrast to the basic periodogram in Part D, Bartlett's method aims to reduce variance in the PSD estimate by dividing the data into overlapping segments and averaging periodograms across the segments.

Mathematically, Bartlett's method is defined as:

$$\hat{S}(\omega) = \frac{1}{M}\sum_{m=0}^{M-1} I_m(\omega)$$

$$I_m(\omega) = \frac{1}{L}\left|\sum_{l=0}^{L-1} x_m[l]e^{-j\omega l}\right|^2$$

$$x_m[l] = x[l + mL]; 0 \leq l \leq L - 1$$

Where M is the number of segments, L is the segment length, and $x_m[n]$ indexes the m^{th} segment of data.

Figures 7-10 illustrate the PSD estimates for different segment lengths L, showing "tighter" spectra for larger L (more averaging). For L = full data length, Bartlett's method reduces to the basic periodogram, as expected.

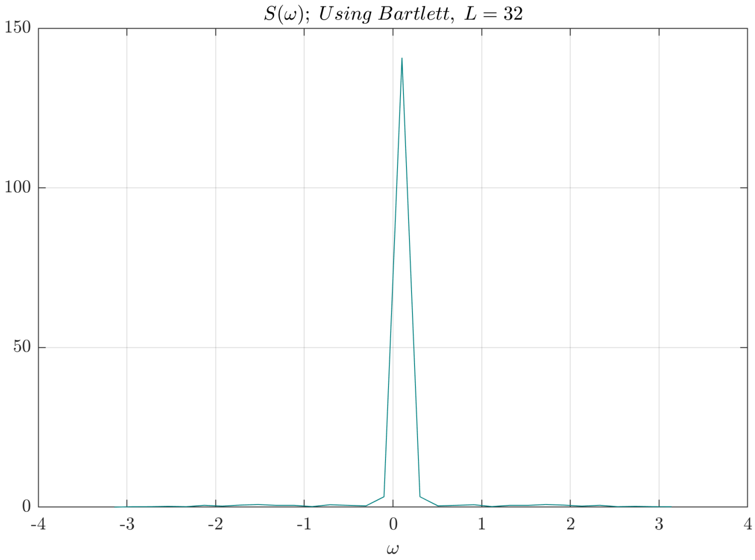


Figure 7: Periodogram - $L = 32$

To verify accuracy, Figures 11 and 13 plot the PSD obtained by Fourier transforming the empirical auto-correlation function. We see Bartlett's estimate correctly converging to the true PSD. Both non-parametric approaches (Barlett's method and Periodogram) reveal similar harmonic detail about the correlation structure.

In summary, Bartletts method trades variance reduction for resolution by controlling the time-frequency localization.

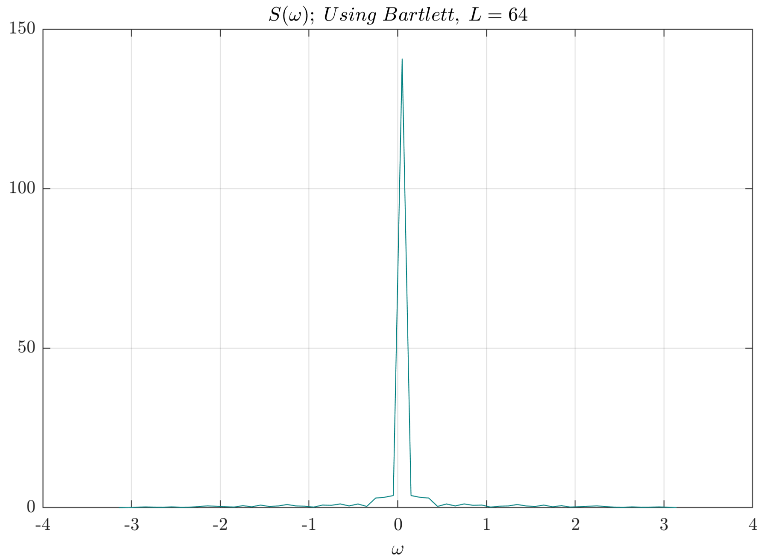


Figure 8: Periodogram - $L = 64$

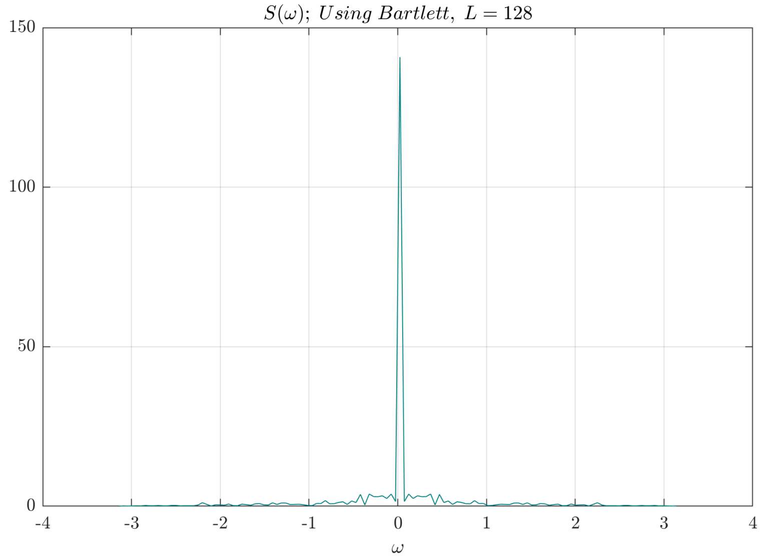


Figure 9: Periodogram - $L = 128$

Overall, both the Bartlett and periodogram methods produced highly accurate PSD plots, correctly revealing the harmonic content as verified against the true Fourier-based spectrum. The AR modeling approach also provided reasonable spectral plots, communicating the overall distribution even if specific resonances were smoothed over due to model order limitations. So all approaches had merits in exposing the spectral properties of the noise process. However, the Fourier domain methods (Bartlett, periodogram) aligned best with ground truth when extensive data was leveraged.

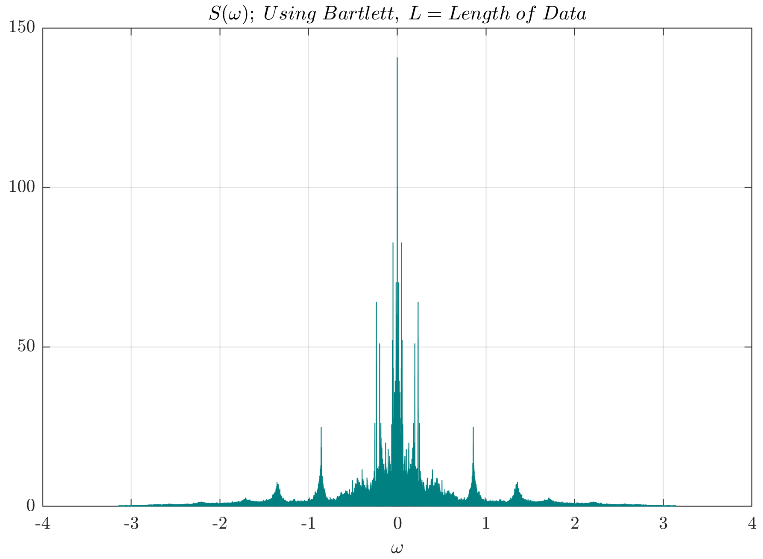


Figure 10: Periodogram - L = full data length

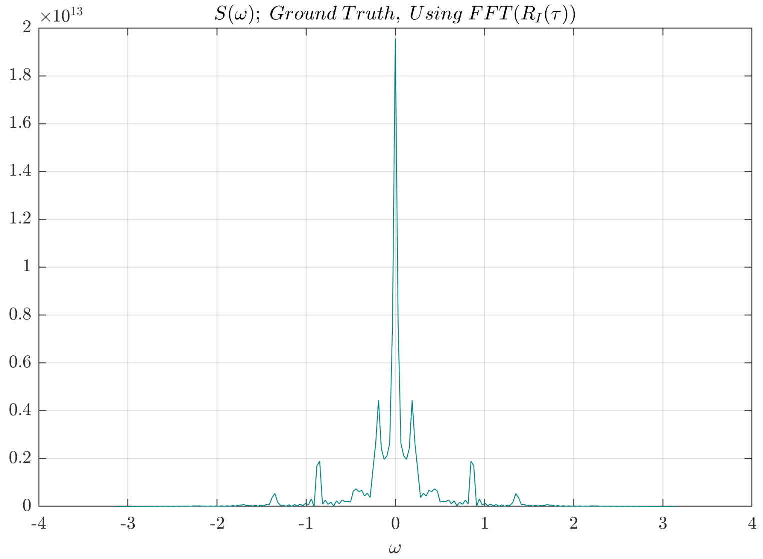


Figure 11: FFT of auto-correlation function

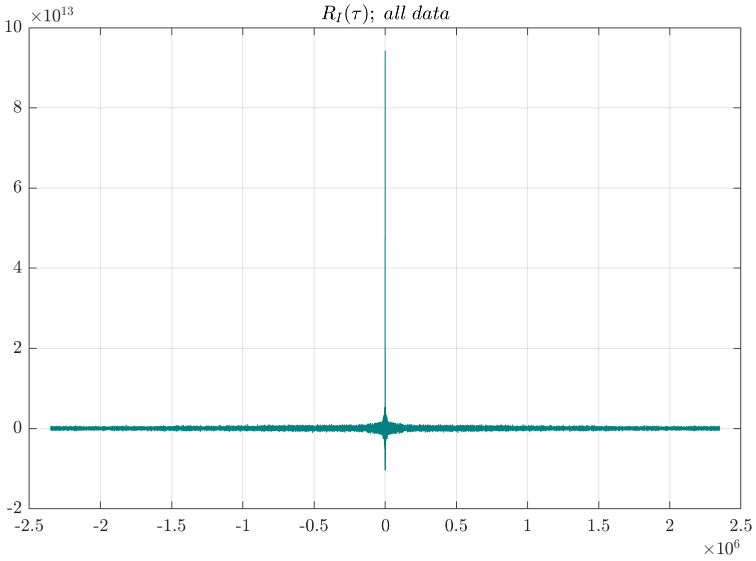


Figure 12: Auto-correlation function using all data

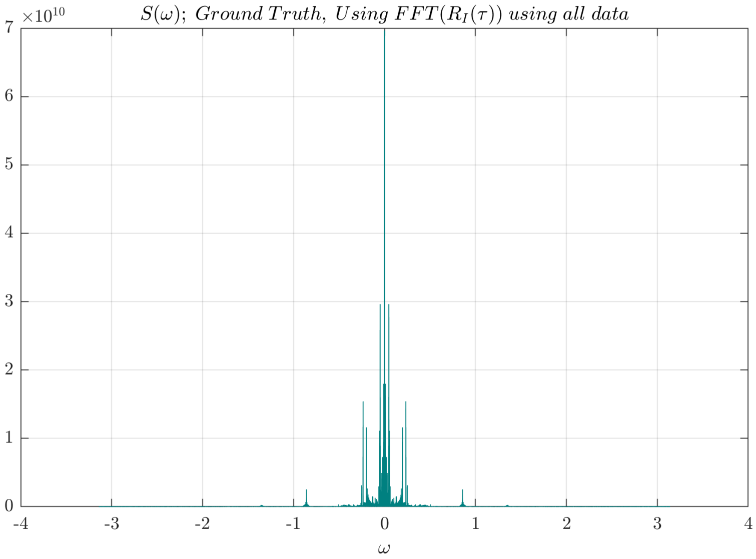


Figure 13: FFT of auto-correlation function, obtained by full data length

2 Mean Square Estimation

In this question, the goal is to estimate $s[n]$ optimally from the observations $x[n]$. We formulate multiple estimators for $\hat{s}[n]$, leveraging different amounts of knowledge about the signal and noise.

To derive the estimator equations, we minimize the mean-squared error between the true signal $s[n]$ and its estimate $\hat{s}[n]$. Invoking the orthogonality principle in mean square estimation leads to formulas for calculating optimal coefficient values that best recover $s[n]$ from $x[n]$.

2.1 Part A.

Orthogonality principle:

$$s[n] - \hat{s}[n] \perp x(n-m)\ ; 0 \leq m \leq M-1$$

$$s[n] - \sum_{i=0}^{M-1} h_i x[n-i] \perp x(n-m)\ ;\ 0 \leq m \leq M-1$$

$$\Longrightarrow \mathrm{E}\{(s[n] - \sum_{i=0}^{M-1} h_i x[n-i])(x(n-m))\} = 0$$

$$\Longrightarrow R_{sx}[m] - \sum_{i=0}^{M-1} h_i R_x[m-i] = 0$$

$$R_{sx}[m] = \mathrm{E}\{s[m+n]x[n]\} = \mathrm{E}\{s[m+n](s[n]+w[n])\} = R_s[m] + \underbrace{R_{sw}[m]}_{=0} \Longrightarrow$$

$$\boxed{R_{sx}[m] = R_s[m]}$$

$$R_x[m-i] = \mathrm{E}\{x[m]x[i]\} = \mathrm{E}\{(s[m]+w[m])(s[i]+w[i])\} = R_s[m-i] + \underbrace{R_{ws}[m-i]}_{=0} + \underbrace{R_{sw}[m-i]}_{=0} + R_w[m-i]$$

$$\Longrightarrow \boxed{R_x[m-i] = R_s[m-i] + R_w[m-i]}$$

The equation to obtain optimal coefficients:

$$\sum_{i=0}^{M-1} h_i \left(R_s[n-i] + R_w[n-i]\right) = R_s[n];\ 0 \leq n \leq M-1 \tag{1}$$

2.2 Part B.

$$s[n] = 0.6s[n-1] + \nu[n]$$

$$S(z) = 0.6z^{-1}S(z) + N(z)$$

$$H(z) = \frac{S(z)}{N(z)} = \frac{1}{1-0.6z^{-1}}$$

$$S_s(z) = H(z)H(1/z)S_\nu(z)$$

$$S_\nu(z) = 0.64$$

$$\begin{aligned}\implies S_s(z) &= \frac{1}{1-0.6z^{-1}} \times \frac{1}{1-0.6z} \times 0.64\\ &= \frac{0.6}{z-0.6} + \frac{1/0.6}{z-1/0.6}\\ &= \frac{0.6z^{-1}}{1-0.6z^{-1}} - \frac{1/0.6z^{-1}}{1-1/0.6z^{-1}}\end{aligned}$$

$$\implies \boxed{R_s[n] = (0.6)^n u[n-1] + (0.6)^{n-1}u[-n]} \tag{2}$$

To validate the derived equation for $R_s[n]$, Figures 14 and 15 present both theoretical and empirical values of $R_s[n]$, illustrating a close alignment between the theoretical predictions and simulation observations.

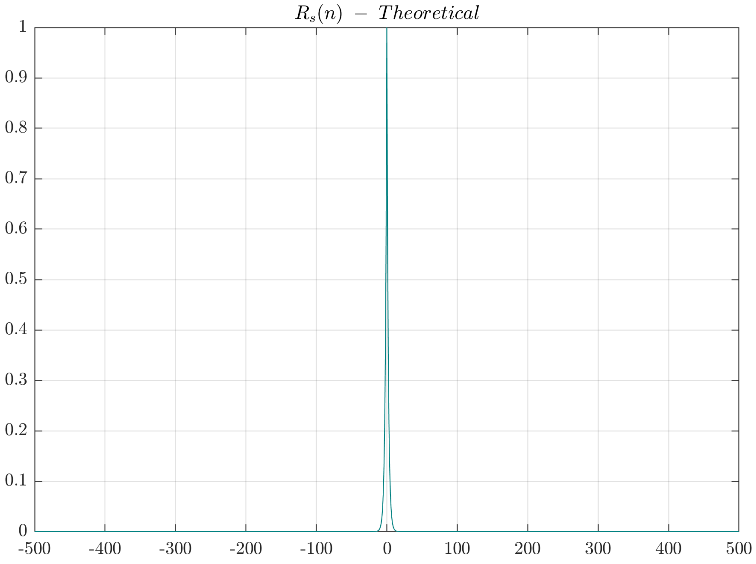


Figure 14: $R_s[n]$ - theoretical

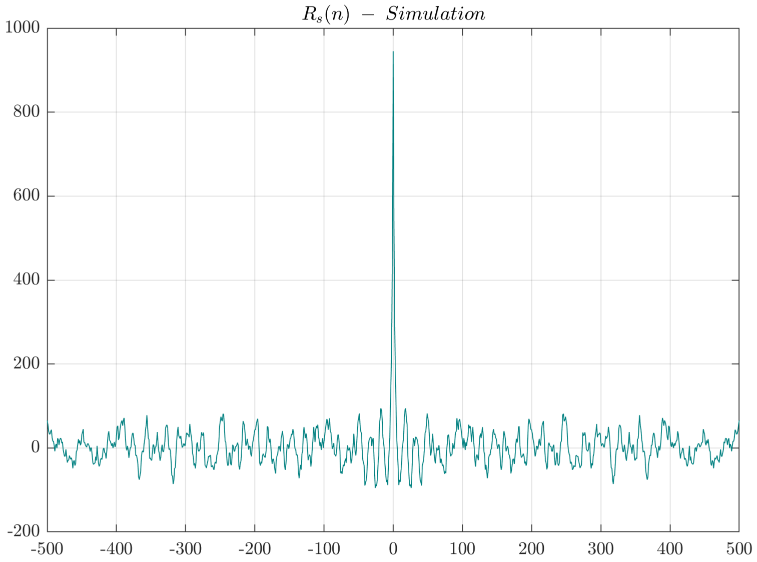


Figure 15: $R_s[n]$ - simulation

For a second order estimation:

$$\sum_{i=0}^{1} h_i \left(R_s[n-i] + R_w[n-i]\right) = R_s[n];\ 0 \leq n \leq 1$$

$$\begin{cases} h_0 \left(R_s[0] + R_w[0]\right) + h_1 \left(R_s[1] + R_w[1]\right) = R_s[0] \\ h_0 \left(R_s[1] + R_w[1]\right) + h_1 \left(R_s[0] + R_w[0]\right) = R_s[1] \end{cases} \tag{3}$$

Note that since the signals in question are real-valued processes, an important property is that their auto-correlation functions are even functions with respect to the origin. That is:

$$R_s[-m] = R_s[m]$$

To solve the set of linear equations defined in (3), we need the auto-correlation values R_s and R_w.

R_s can be directly obtained via equation (3).

As for R_w, since $w[n]$ is specified as white noise with zero mean and unit variance, its auto-correlation function takes the form:

$$R_w[m] = \delta[m]$$

This reflects the fact that white noise values are uncorrelated across any non-zero lag m.

Therefore, the auto-correlation functions required to evaluate the optimal estimators per (3) can be constructed.

In alignment with the outlined procedure, the optimal coefficients in (1) can be derived for any given order of M through the following equation:

$$\begin{bmatrix} R_s(0)+R_w(0) & \dots & R_s(M-1)+R_w(M-1) \\ \vdots & \vdots & \ddots \\ R_s(M-1)+R_w(M-1) & \dots & R_s(0)+R_w(0) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{M-1} \end{bmatrix} = \begin{bmatrix} R_s(0) \\ R_s(1) \\ \vdots \\ R_s(M-1) \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{M-1} \end{bmatrix} = \begin{bmatrix} R_s(0)+R_w(0) & \dots & R_s(M-1)+R_w(M-1) \\ \vdots & \vdots & \ddots \\ R_s(M-1)+R_w(M-1) & \dots & R_s(0)+R_w(0) \end{bmatrix}^{-1} \begin{bmatrix} R_s(0) \\ R_s(1) \\ \vdots \\ R_s(M-1) \end{bmatrix} \tag{5}$$

Using (5), all coefficients can be obtained. To elaborate, for $M = 2$:

$$\begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} 2 & 0.6 \\ 0.6 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.6 \end{bmatrix} \tag{6}$$

which gives:

$$h_0 = 0.4505 \; \& \; h_1 = 0.1648$$

$$\implies \boxed{\hat{s}[n] = 0.4505x[n] + 0.1648x[n-1]}$$

2.3 Part C.

The actual signal $s[n]$ and the estimated signal $\hat{s}[n]$ for second order estimation are depicted in Figure 16.

2.4 Part D.

Figures 17 - 23 illustrate the actual and estimated signals and error of estimation for varying values of M.

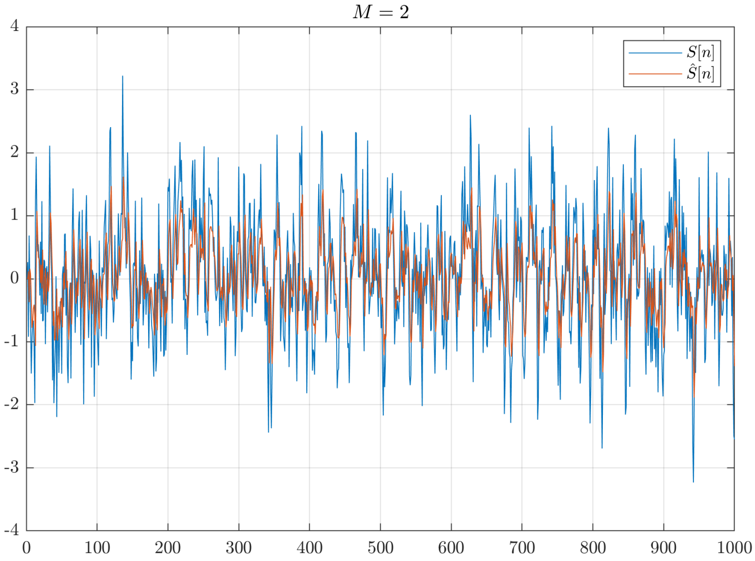


Figure 16: Actual and estimation signals, $M = 2$

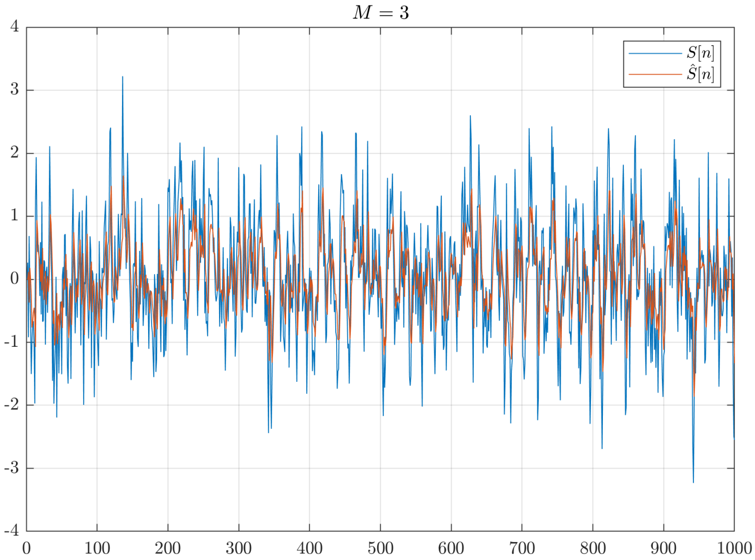


Figure 17: Actual and estimation signals, $M = 3$

As evident from the observation, increasing the order of estimation, denoting the utilization of more samples from previous data, exhibits only a marginal impact on error reduction. While this might seem counter-intuitive, a mathematical deduction elucidates this phenomenon. The underlying reason can be attributed to the auto-correlation function of $s[n]$. As previously demonstrated, the auto-correlation function $R_s[m]$ undergoes exponential decay. For instance, when $m = 9$, $R_s[9] = 0.01$, signifying a weak correlation among time samples of $s[n]$, particularly as the temporal gap between two points widens. Consequently, incorporating more past samples

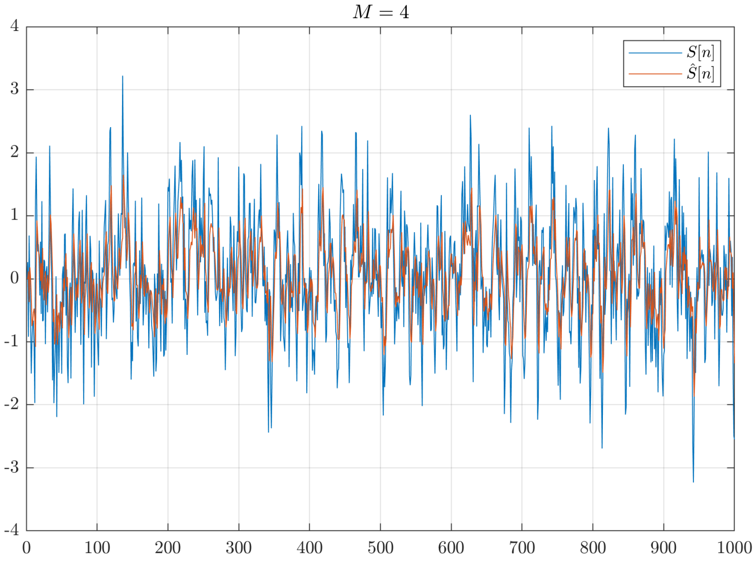


Figure 18: Actual and estimation signals, $M = 4$

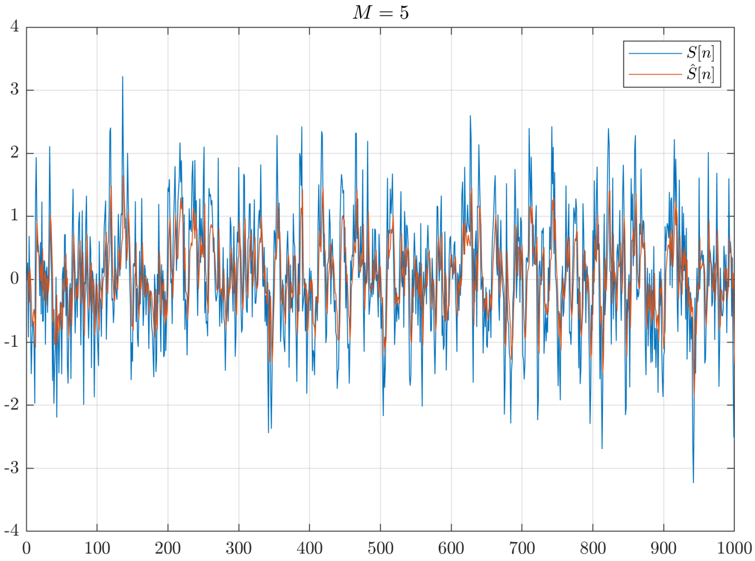


Figure 19: Actual and estimation signals, $M = 5$

fails to contribute significantly to our estimation, a characteristic also reflected in the values of the h_i coefficients. For instance, considering $M = 5$, the values of h_i are:

$$h_0 = 0.4445;\ h_1 = 0.1482;\ h_2 = 0.0495;\ h_3 = 0.0167; h_4 = 0.0061;$$

Clearly, the h_i values diminish and become negligible for samples distant in time from the current instance.

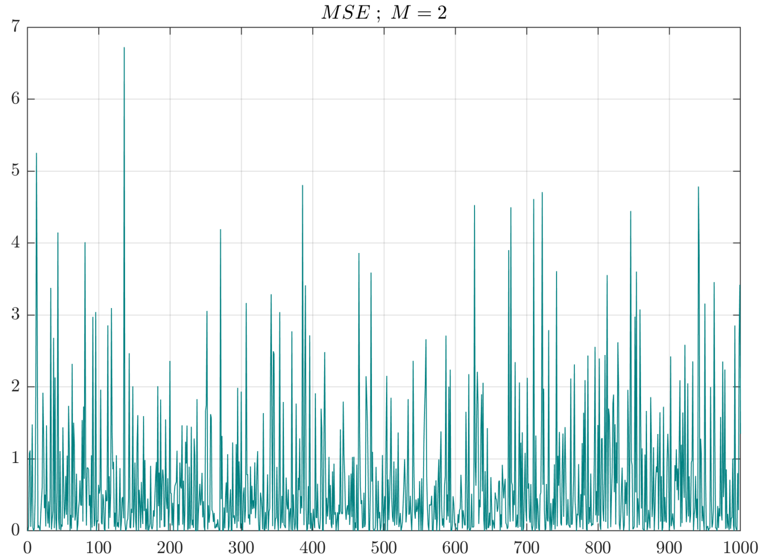


Figure 20: MSE, $M = 2$

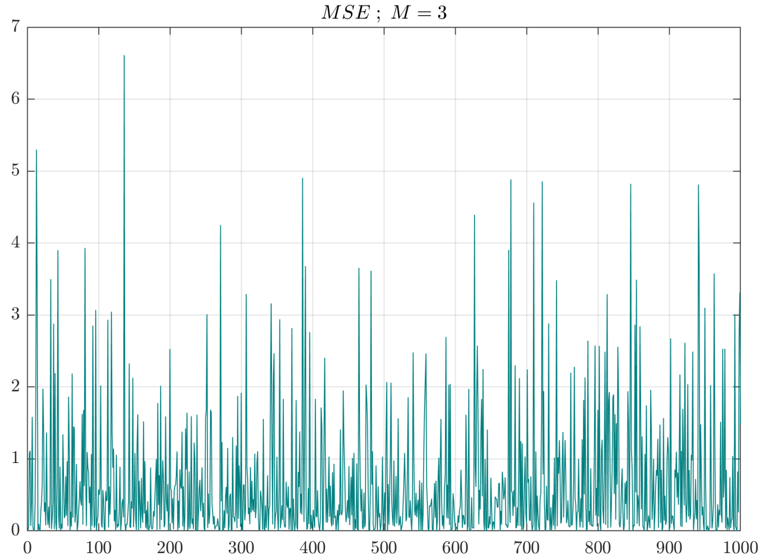


Figure 21: MSE, $M = 3$

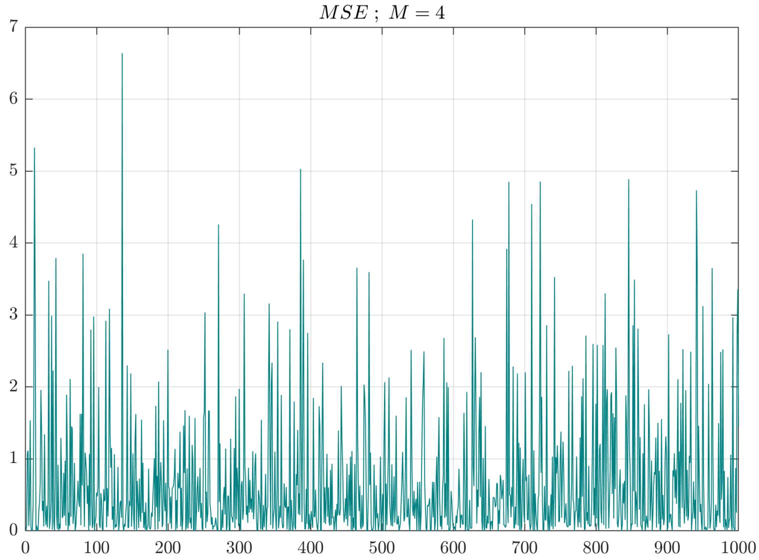


Figure 22: MSE, $M = 4$

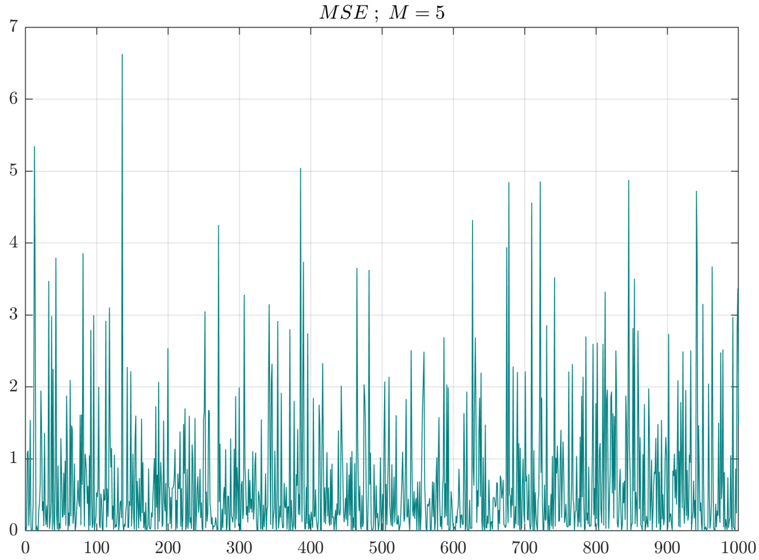


Figure 23: MSE, $M = 5$

3 Kalman Filter

In this question, we aim to estimate $y[n]$ based on observations of $x[n]$. The relation between $y[n]$ and $x[n]$ is given by:

$$x[n] = y[n] + w[n]$$

Note that for the sake of simplicity, we incorporate the coefficient associated with $w[n]$ into the variance. This implies that we treat $w[n]$ as white noise with a variance of $\sigma_w^2 = 10$.

A second order auto-regressive model $AR(2)$ is defined as:

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + \nu(n)$$

Like before, we treat $\nu(n)$ as white noise with a variance of $\sigma_\nu^2 = 0.01$.

The second order auto-regressive model can be reformulated in a state model. The state vector to be considered includes the actual value of signal at n and $n-1$, $\boldsymbol{y}(n) = [y(n), y(n-1)]^T$ and $\hat{y}(n) = [\hat{y}(n), \hat{y}(n-1)]^T$. The state transition matrix is $\mathbf{M} = \begin{bmatrix} a_1 & a_2 \\ 1 & 0 \end{bmatrix}$ and the state noise vector is : $\mathbf{u}(n) = [u(n), 0]^T$. The selection vector of size 1×2 is $\mathbf{s}^T = [1, 0]$. Therefore:

$$\hat{\boldsymbol{y}}(n) = \mathbf{M}\hat{\boldsymbol{y}}(n-1) + \mathbf{u}(n) \tag{7}$$

$$x(n) = \mathrm{s}^T \boldsymbol{y}(n) + w(n) \tag{8}$$

Regarding the state-space formulation (7) and (8), the two stages of the filter are :

Time update equations:

$$\hat{\boldsymbol{y}}(n \mid n-1) = \mathbf{M}\hat{\boldsymbol{y}}(n-1 \mid n-1)$$

$$\mathbf{P}(n \mid n-1) = \mathbf{M}\mathbf{P}(n-1 \mid n-1)\mathbf{M}^H + \mathbf{U}$$

Measurement update equations:

$$\mathbf{K}(n) = \frac{\mathbf{P}(n \mid n-1)\mathbf{s}}{\mathbf{s}^T\mathbf{P}(n \mid n-1)\mathbf{s} + \sigma_w^2}$$

$$\hat{\boldsymbol{y}}(n \mid n) = \hat{\boldsymbol{y}}(n \mid n-1) + \mathbf{K}(n)\left(x(n) - \mathbf{s}^T\hat{\boldsymbol{y}}(n \mid n-1)\right)$$

$$\mathbf{P}(n \mid n) = \left(\mathbf{I}_2 - \mathbf{K}(n)\mathbf{s}^T\right)\mathbf{P}(n \mid n-1)$$

where $\mathbf{K}(n) = \begin{bmatrix} K_1(n) \\ K_2(n) \end{bmatrix}$ is the Kalman gain vector, $\mathbf{U} = \begin{bmatrix} \sigma_\nu^2 & 0 \\ 0 & 0 \end{bmatrix}$, $\mathbf{I}_2$ 2×2 identity matrix and $\mathbf{P}(n \mid n)$ and $\mathbf{P}(n \mid n-1)$

Given the values $a_1 = 1.8$ and $a_2 = -0.81$, we can employ the Kalman filter utilizing the aforementioned equations. Figure 24 showcases the outcome of applying the Kalman filter. Despite the presence of substantial noise obscuring the state, the primary pattern of the signal remains concealed. From a visual assessment of the observed signal, discerning the underlying pattern becomes challenging. However, the implemented Kalman filter effectively operates, successfully extracting the original signal. Furthermore, for additional investigations, the Kalman filter is tested in a scenario with a higher Signal-to-Noise Ratio (SNR). As illustrated in Figure 25, the elevated SNR results in a more accurate estimation of the original signal, as expected. In this context, the observation signal proves to be more informative.

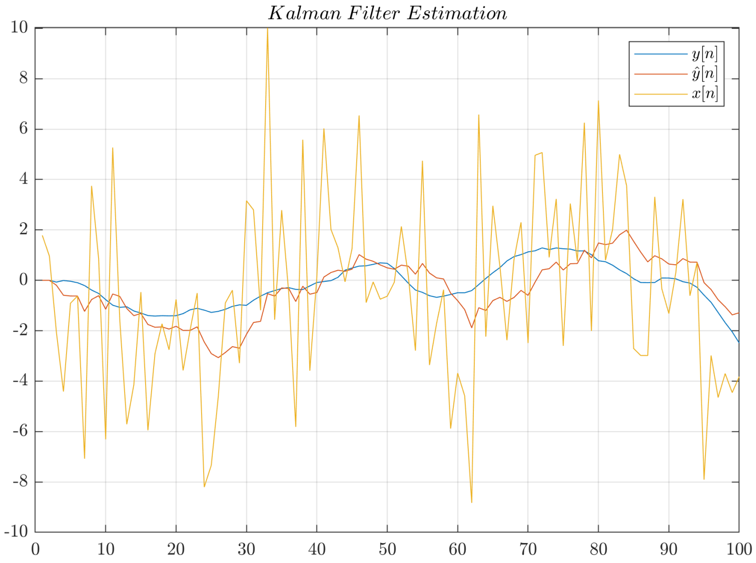


Figure 24: Kalman filter estimation

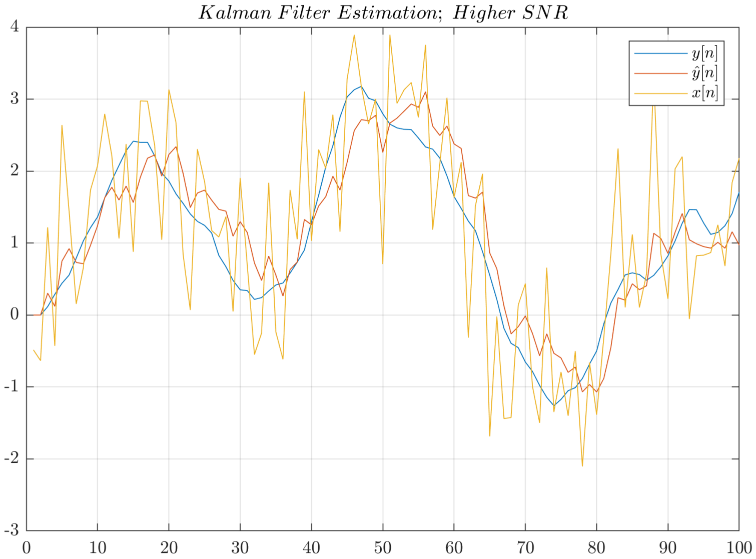


Figure 25: Kalman filter estimation in highe SNR