Kalana Gayal Abeywardena
1009505867
kalana.abeywardena@mail.utoronto.ca

Project 02 - Results and Discussion

JEB 1444 - Neural Engineering

Part 1: Estimating Wiener-Bose Model Parameters

In this part of the project, I am estimating a second-order discrete-time Wiener-Bose non-parametric model for the coupled FitzHugh-Nagumo oscillator system based on input-output characteristics using Laguerre-Expansion Technique (LET).
To estimate the Wiener Kernels (and the static non-linear coefficients) I first pass a band-limited white Gaussian noise (WGN) as the input to the coupled FZN model to get the model output. First, we should make sure that the bandwidth of the WGN is such that:

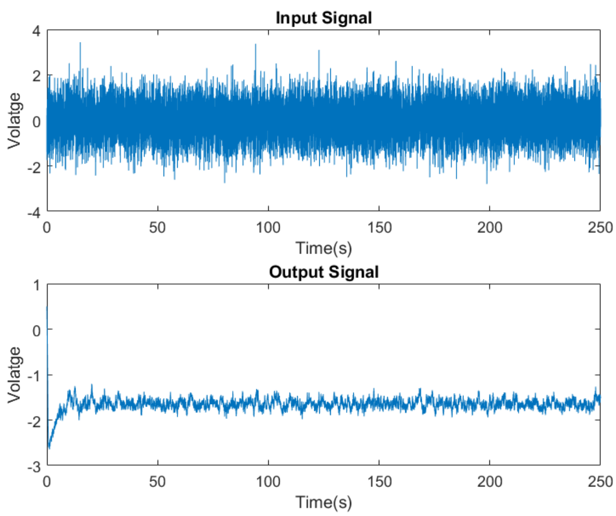
- it is sufficiently larger than the system bandwidth and
- follows the sampling theory to avoid any aliasing effects on the system output.

For this purpose, the generated and cubic-interpolated WGN is first passed through a low-pass filter with a cut-off frequency that satisfies the above requirements. Once the band-limited WGN input is constructed, we can use it as the input to the FZN model to obtain the system output by solving the following dynamic equations using *ODE15s* solver.

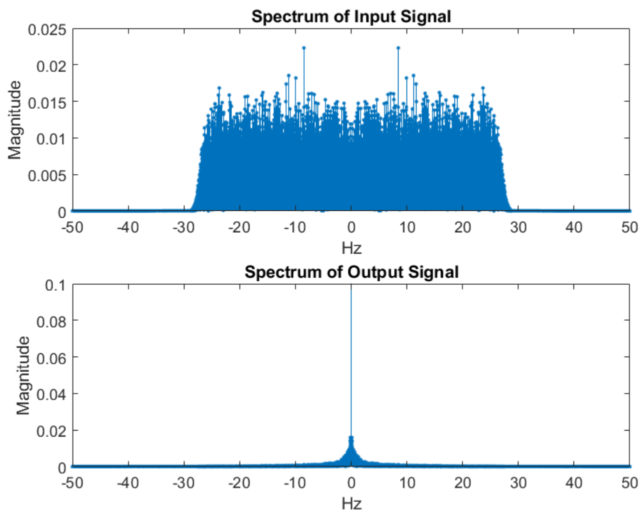
$$\dot{x}_1 = \alpha\left[y_1 + x_1 - \frac{x_1^3}{3} + (k_1 + cx_2) + z_{in}^1\right] \qquad \dot{y}_1 = -\frac{1}{\alpha}\left[\omega^2 x_1 - a + by_1\right]$$

$$\dot{x}_2 = \alpha\left[y_2 + x_2 - \frac{x_2^3}{3} + (k_2 + cx_1)\right] \qquad \dot{y}_2 = -\frac{1}{\alpha}\left[\omega^2 x_2 - a + by_2\right]$$

The system outputs and their spectral characteristics are shown in Figure 1 for a non-oscillatory system response and in Figure 2 for an oscillatory system response. As the spectrums show, the bandwidth of the input WGN is sufficiently large compared to the system output bandwidth and there is no apparent aliasing present.



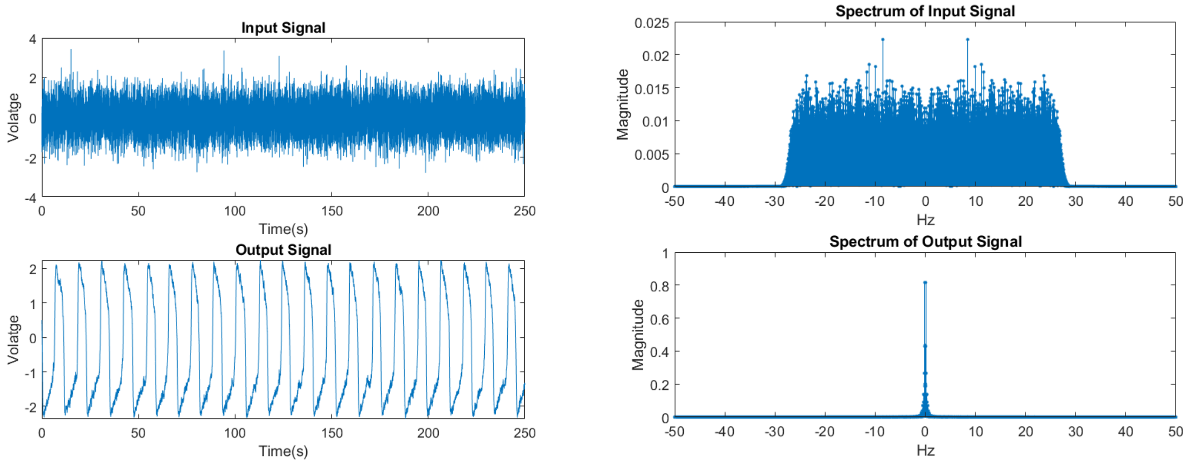
(a) Temporal characteristics



(b) Spectral characteristics

Figure 1: Filtered GWN Input with F_p = 25Hz and System Output
(Non-Oscillatory Region: c = 0.95, k_1 = -1.41, k_2 = -0.68)

Now that we have the system input and output, the next step is to estimate the Block Wiener-Bose Model parameters. We can do this in two ways:



(a) Temporal characteristics

(b) Spectral characteristics

Figure 2: Filtered GWN Input with F_p = 25Hz and System Output
(Oscillatory Region: c = 0.32, k_1 = -1.41, k_2 = -0.84)

- Directly estimating the Wiener Kernels as the kernels are *orthogonal* with WGN as the input to the system
- Estimate the static non-linear coefficients that construct the Wiener Kernels using the Laguerre function decomposition

Both these estimations can be obtained from the function LET_1 from the *Lysis* package. Before using the package to estimate Wiener-Bose model coefficients, we require the following to be estimated to the best of our knowledge:

- First-Order Kenerl ($K_1(\tau)$) and System Memory (in samples)
- System Bandwidth (per sample)
- L (No. of Laguerre Functions) and α.

Estimation of First-Order Kernel and System Memory

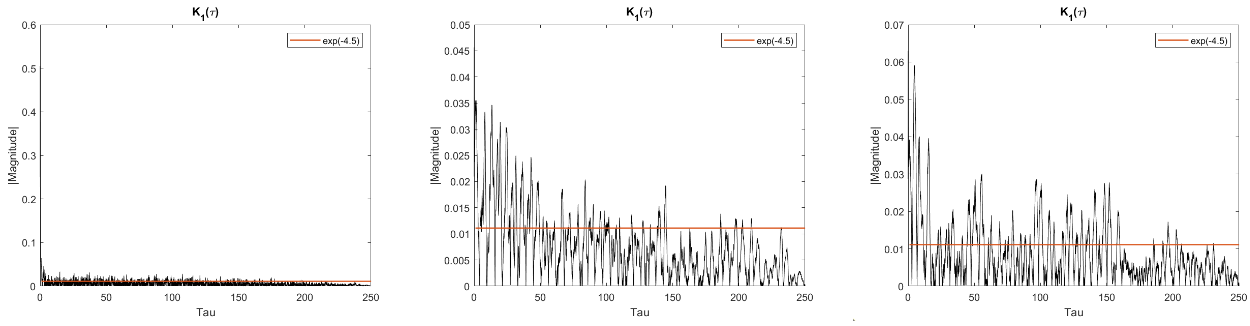
For this purpose, I am using the cross-correlation between the input and the output as opposed to the Fourier Transformation method due to its lack of resolution. Let $x[n]$ = WGN and $y[n] = x_1$.

$$\therefore K_1(\tau) = \frac{1}{\mathrm{P}_x}\mathbb{E}\Big\{\big[y[n] - \mathbb{E}(y[n])\big]x[n]\Big\} \text{ where } \mathbb{E}(y[n]) = \bar{y} = \frac{1}{N}\sum_{n=1}^{N} y[n] \text{ is the mean of output signal.}$$

After estimating $K_1(\tau)$, the next is to estimate the system memory which is the length of the kernel. I selected a sample that is approximately equal to 5-time constants ($\tau_{Memory} \approx 5RC$) i.e., the sample at which the kernel response falls to $\approx |K_1(\tau)|_{max} \exp(-5)$. In my implementation, I used $\tau_{Memory} = 4.5RC$ as the time constant based on the simulations run on the regions I have selected.
Figure 3 illustrates three $K_1(\tau)$ estimated for three system behaviors and the system memories estimated: (a) shows the kernel response for $c = 0.95,\ k_1 = -1.41,\ k_2 = -0.68$, (b) shows the kernel response for $c = 0.32,\ k_1 = -1.41,\ k_2 = -0.84$, and (c) shows the kernel response for $c = 0.01,\ k_1 = -1.41,\ k_2 = -0.61$.

Estimation of System Bandwidth

When estimating the system bandwidth, I took the frequency range of the system output where 99% of its spectral energy is contained. The frequencies in the discrete domain are of the unit ”per sample”. For the three systems shown in Figure 3, the corresponding bandwidths I estimated are (a) 0.0015 per sample , (b) 0.0077 per sample, and (c) 0.0075 per sample.



(a) Non-Oscillatory Region (N = 21245) (b) Oscillatory Region (N = 23193) (c) Chaotic Region (N = 23065)

Figure 3: $K_1(\tau)$ estimation for three system behaviours

Now using the estimated system bandwidth and system memory, for each system behavior, I calculate the *Memory-Bandwidth Product* as:

$$\mathrm{M} = \tau_{sys} \times \mathrm{B}_{sys}$$

Using this, we can then estimate the final parameters required to run LET_1.

Estimation of L and α

The number of Laguerre functions and α are coupled estimations that require simultaneous estimation based on *Memory-Bandwidth Product (M)* such that the Laguerre functions estimated will span the system memory sufficiently.

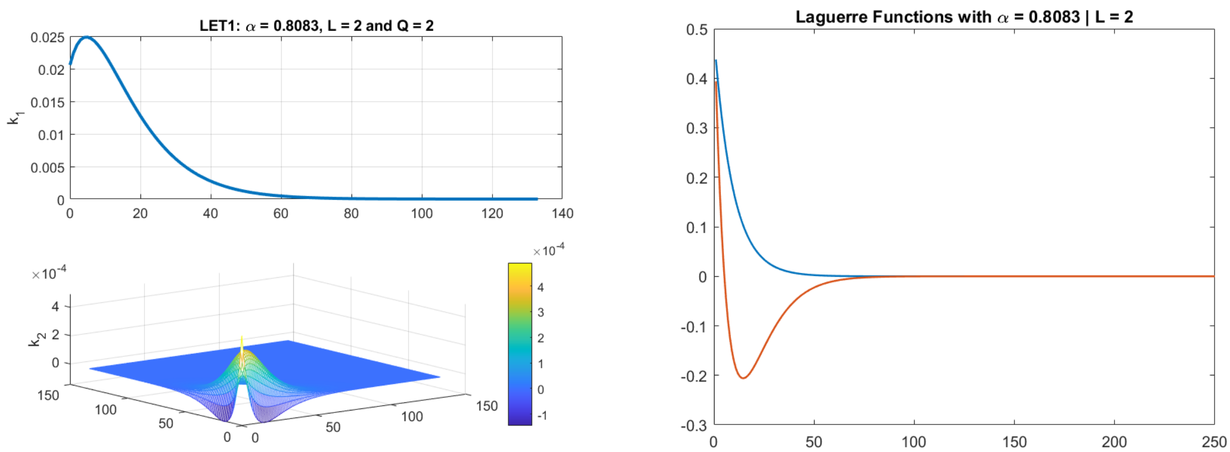
However, in this part, I estimated the two values heuristically as follows:

$$\mathrm{L} = \max\left(1, \left\lceil \frac{\mathrm{M}}{\mathrm{L}_{\mathrm{Thresh}}} \right\rceil\right) \text{ where } \mathrm{L}_{\mathrm{Thresh}} = 25,$$

and

$$\alpha = \exp\left(-\frac{\alpha_{Thresh}}{\tau_{Sys} \times T_s - 1}\right) \text{ where } T_s = 0.01s \text{ and } \alpha_{Thresh} = 45.$$

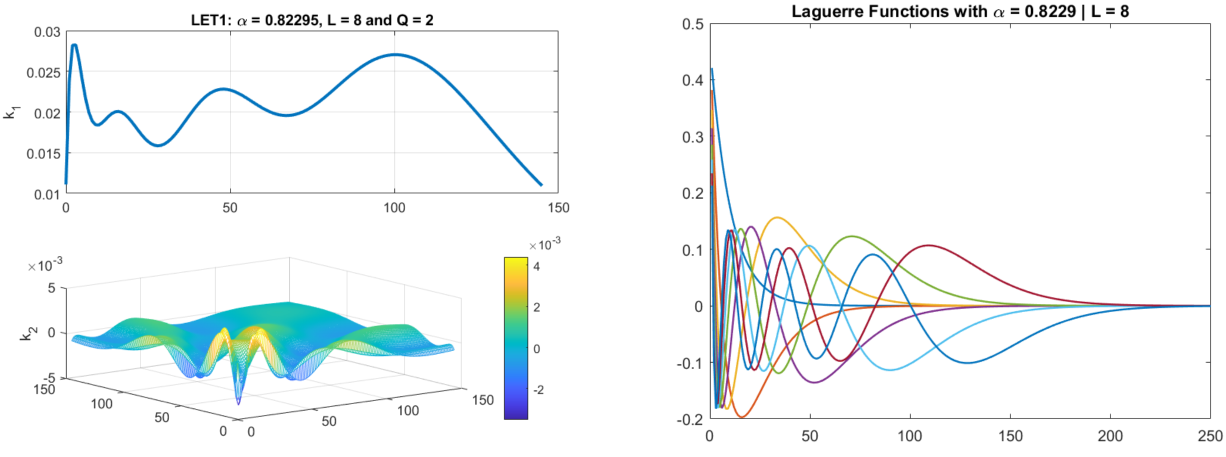
Now using these results, we can estimate the Wiener-Bose Model parameters that depict the non-parametric model for the given input-output characteristics. Some of the results are shown below for the three system behaviors.



(a) Estimated Wiener Kernels (b) Laguerre Functions in Linear Part

Figure 4: Estimated Wiener-Bose Kernels and Laguerre Functions for Non-Oscillatory System Behavior

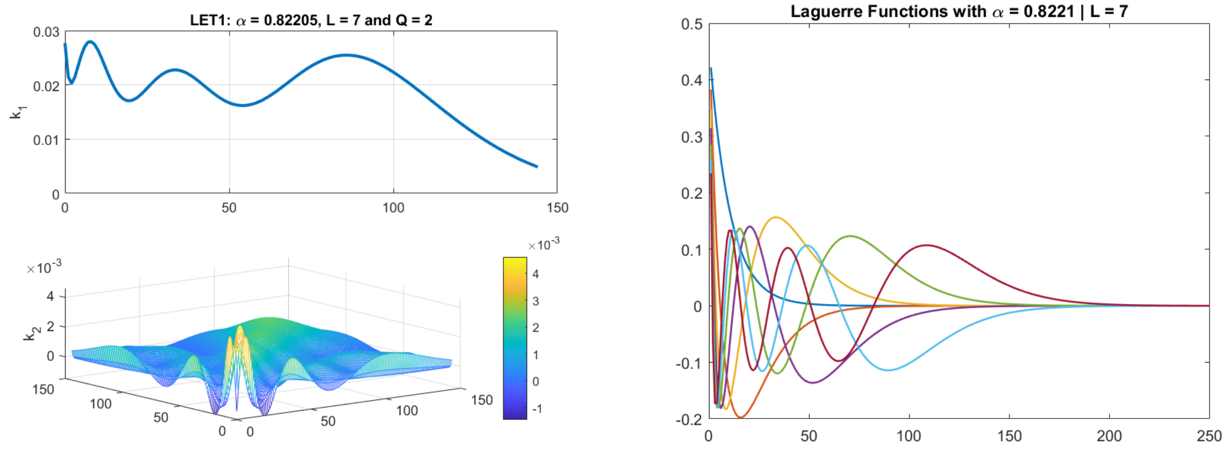
Based on the estimated Wiener-Bose Models for each system behavior, the responses for the training WGN sequence are obtained and the Mean Squared Error between the ground truth and the predicted is calculated.



(a) Estimated Wiener Kernels

(b) Laguerre Functions in Linear Part

Figure 5: Estimated Wiener-Bose Kernels and Laguerre Functions for Oscillatory System Behavior

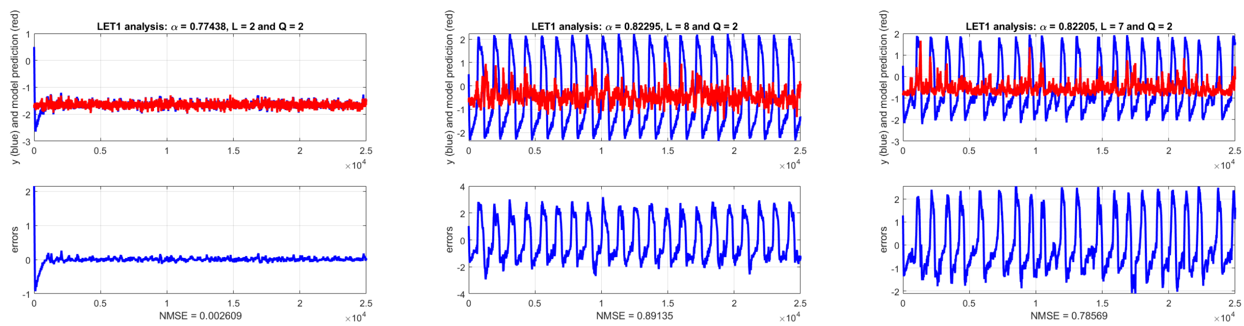


(a) Estimated Wiener Kernels

(b) Laguerre Functions in Linear Part

Figure 6: Estimated Wiener-Bose Kernels and Laguerre Functions for Chaotic System Behavior

However, since the parametric model output is sensitive to the initial conditions, this metric is not used to determine the goodness of estimation. This will be discussed in Part 3.



(a) Non-Oscillatory System

(b) Oscillatory System

(c) Chaotic System

Figure 7: Waveform comparison between the Parametric Model and Wiener-Bose Model

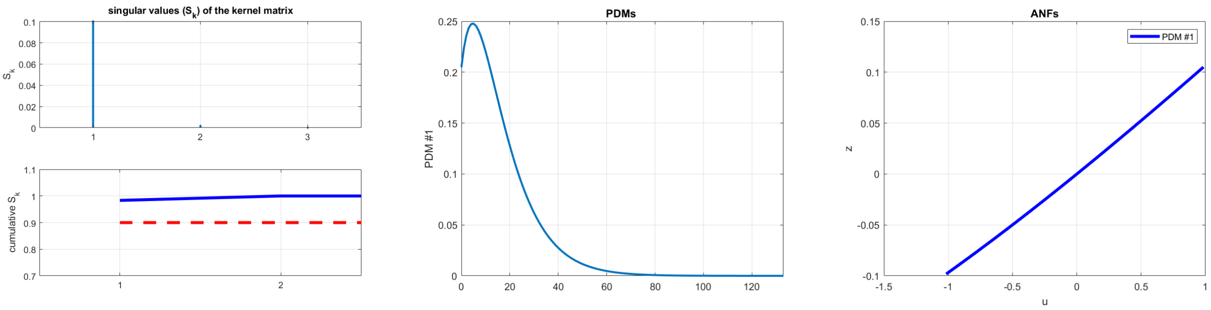
By looking at the responses, we can conclude that the Wiener-Bose model is capable of predicting the non-oscillatory responses with sufficient accuracy. However, we cannot compare the two waveforms for Oscillatory system behavior as the Wiener-Bose model is not capable of producing spikes and requires a threshold trigger to generate the action potentials based on a heuristic threshold.

Part 2: Estimating Principal Dynamic Modes & identify Modular Volterra Model

Now that we have the Wiener-Kernels, we can construct the **Q** matrix.

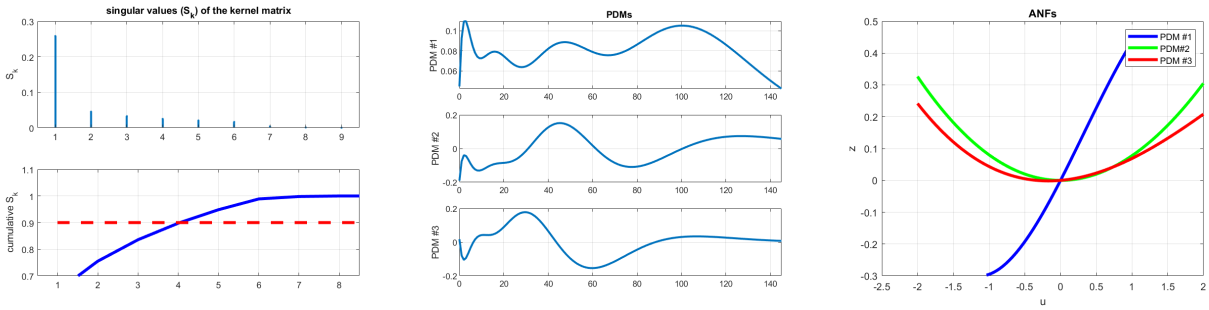
$$\mathbf{Q} = \begin{bmatrix} K_0 & \frac{1}{2}\mathbf{K_1}^T \\ \frac{1}{2}\mathbf{K_1} & \mathbf{K_2} \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \text{ where } \mathbf{K_1} \in \mathbb{R}^{d\times 1}, \mathbf{K_2} \in \mathbb{R}^{d\times d}$$

Now using the eigendecomposition, we can select the number of Principle Dynamic Modes (PDMs) $K = \left|\left\{\lambda_i \;\middle|\; \frac{|\lambda_i|}{\sum|\lambda_j|} \geq \epsilon\right\}\right|$ where I chose $\epsilon = 0.01$. Once K is estimated, I used the PDM_1 function of *Lysis* package to estimate the PDMs, and their Associated Non-linear Functions (ANFs). The evaluations for the three systems from Part 1 are illustrated below.



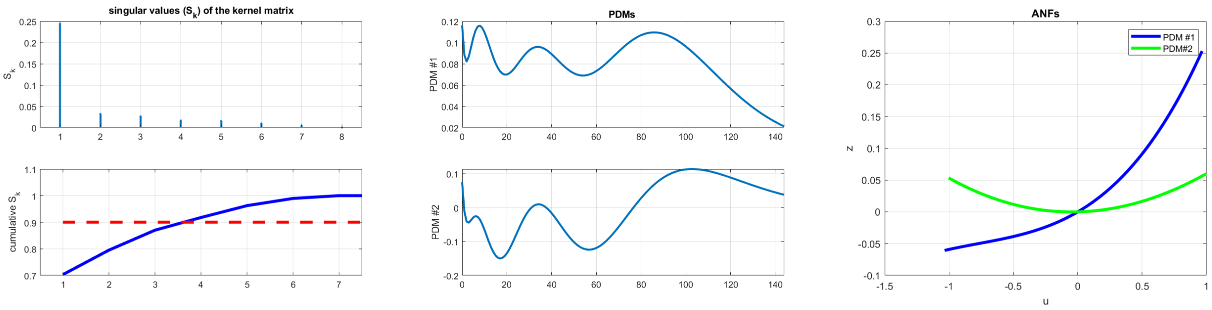
(a) Eigen Values of **Q** matrix (b) Estimated PDMs (c) Estimated associated ANFs

Figure 8: PDM Estimation for Non-Oscillatory System Behavior



(a) Eigen Values of **Q** matrix (b) Estimated PDMs (c) Estimated associated ANFs

Figure 9: PDM Estimation for Oscillatory System Behavior

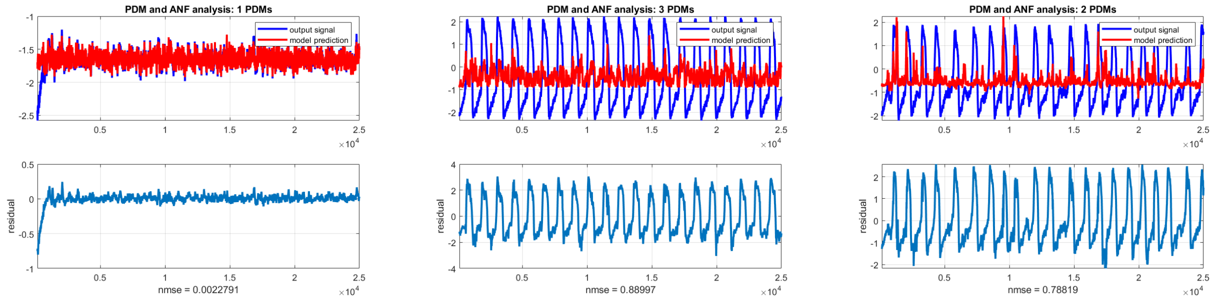


(a) Eigen Values of **Q** matrix (b) Estimated PDMs (c) Estimated associated ANFs

Figure 10: PDM Estimation for Chaotic System Behavior

From Figure 8, for the non-oscillatory system behavior, we only have 1 PDM which turns out to be an integrating PDM. For other system behaviors (illustrated in Figure 9 and Figure 10) we have multiple PDMs that are complex.

Using the estimated PDMs and corresponding ANFs, we can construct a Modular Volterra Model. Figure 11 illustrate a waveform comparison between the FZN model output and the Modular Volterra Model for the three system behaviors.



(a) Non-Oscillatory System (b) Oscillatory System (c) Chaotic System

Figure 11: Waveform comparison between the Parametric Model and Modular Volterra Model

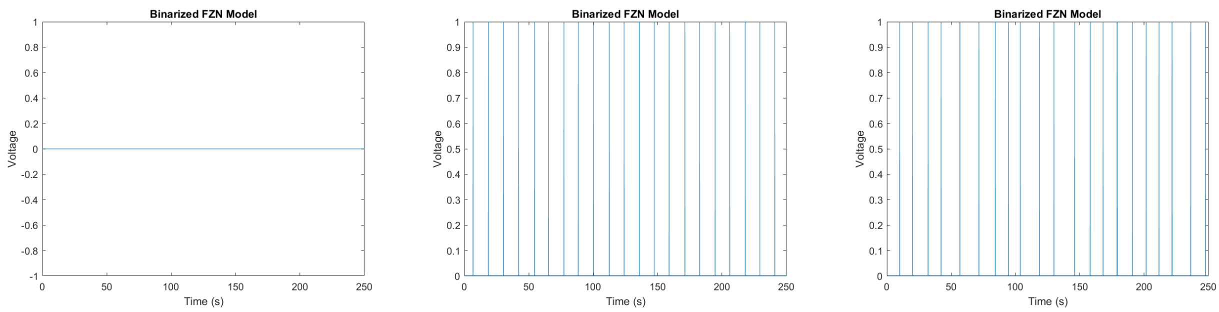
Again for oscillatory system behavior (and chaotic system behavior), we need the threshold trigger to convert the output of the Modular Volterra Model to a spike train. However, we can directly compare the two waveforms for non-oscillatory system behavior where we can observe a significantly low NMSE value.

Part 3: Comparing and Evaluating System Responses

Now that we have estimated the non-parametric models for each system's behaviors for specific model parameters, the next is to evaluate their performances. For an accurate comparison between the output from the parametric model and the outputs from the Wiener-Bose model and Modular Volterra Model, we need to consider the binarized/thresholded versions of the waveforms.

Model Evaluation (Training)

First I evaluated the model performances with the training noise sequence. The PZN model outputs are first binarized to $\{0, 1\}$ by thresholding the output waveforms for each system behavior. Figure 12 shows the binarized waveforms for each system behavior.



(a) Non-Oscillatory System (b) Oscillatory System (c) Chaotic System

Figure 12: Binarized Waveforms from Parametric Model

By visual inspections, the binarization is accurate where in (a) there is no action potentials since it is related to the non-oscillatory system, and in (b) we have periodical action potentials as expected.

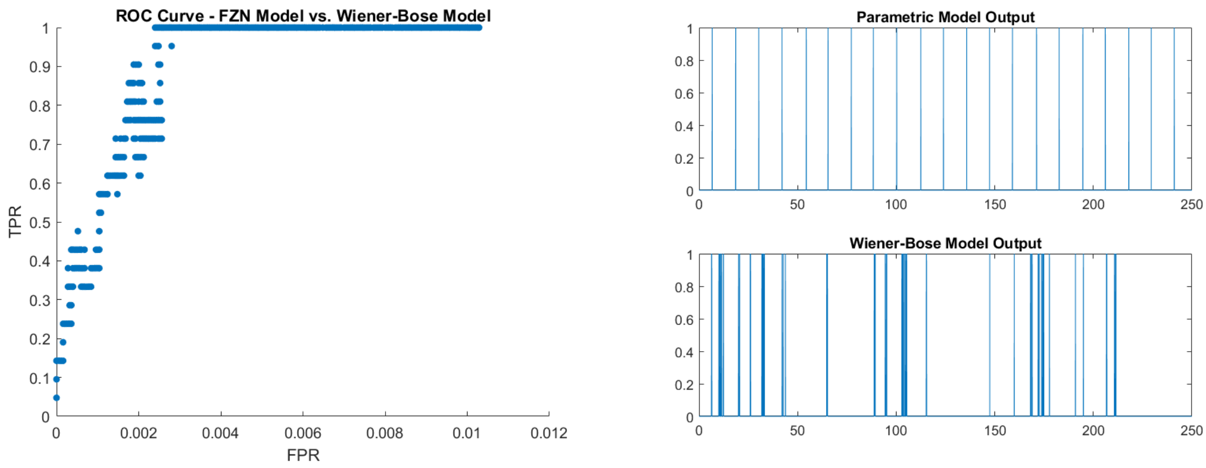
Then, the output waveforms from the Wiener-Bose model and Modular-Volterra model are passed through a threshold trigger to create the action potentials as follows:

$$y_{trig}[n] = 0.5 + 0.5 \cdot \mathrm{sgn}\big(y[n] - \beta\big)$$

where

$$\mathrm{sgn}\big(y[n] - \beta\big) = \begin{cases} +1, & \text{for } y[n] \geq \beta \\ -1, & \text{for } y[n] < \beta \end{cases} ,$$

the threshold $\beta \in [\min(y[n]), \max(y[n])]$, and $y[n]$ is the output from non-parametric models (Wiener-Bose or Modular Volterra models). This is specifically required when the system is oscillatory and/or chaotic as we need the action potentials to be generated from the model output waveform to compare with FZN output. For the non-oscillatory systems, we can select a threshold such that no spikes are generated such that it matches the parametric system behavior. Hence, for the non-oscillatory systems, it is enough to compare the two waveforms using NMSE.
Now for the oscillatory system, I pass the non-parametric system outputs through the threshold trigger with different thresholds. For each threshold, I evaluated *True Positive Rate (TPR)* and *False Positive Rate (FPR)* and constructed the ROC Curve. The optimal threshold for each non-parametric model is selected by selecting the threshold that provided the best TPR.
Figure 13 illustrates the ROC curve obtained for the oscillatory system with the training noise sequence and the thresholded model output of the Wiener-Bose model at the optimal threshold selected.

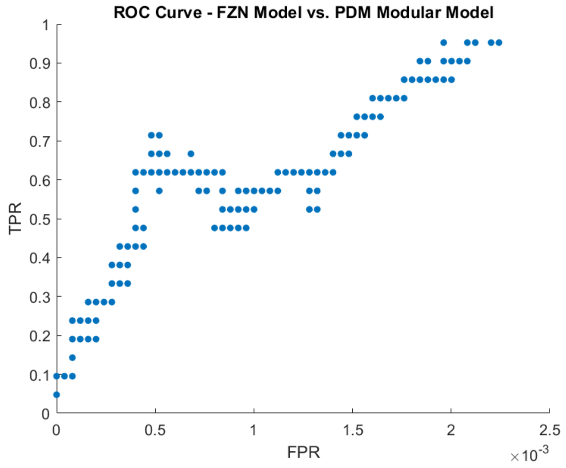


(a) ROC Curve (AUC = 0.00914) (b) Threshold Triggered output ($\beta = 0.2070$)

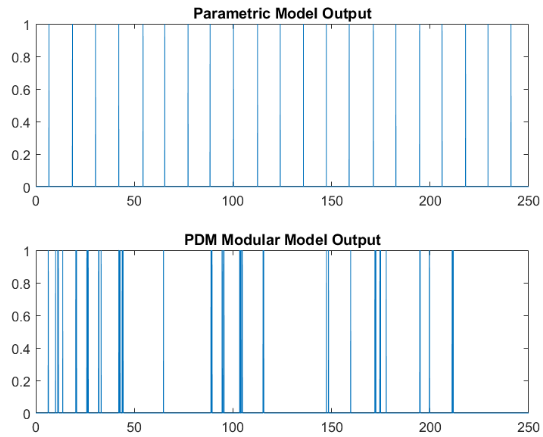
Figure 13: Wiener-Bose Model evaluation for Oscillatory System Response

Compared to the random system response (i.e., fair coin tosses) which gives an AUC = $0.00005 (\ll 0.00914)$, our system produces better system performance. When visually comparing the optimally thresholded signal, we still have significant false positives that impact the goodness of the system performance.

Similarly, Figure 14 illustrates the results obtained for Modular Volterra Model. Compared to the Wiener-Bose Model, the performance of the Modular Volterra Model is lower. However, it is still better compared to the random system response which is AUC = $0.000002 (\lll 0.00139)$ that makes the Modular Volterra Model a better system than the random model to predict the spike generation for the given input WGN signal.



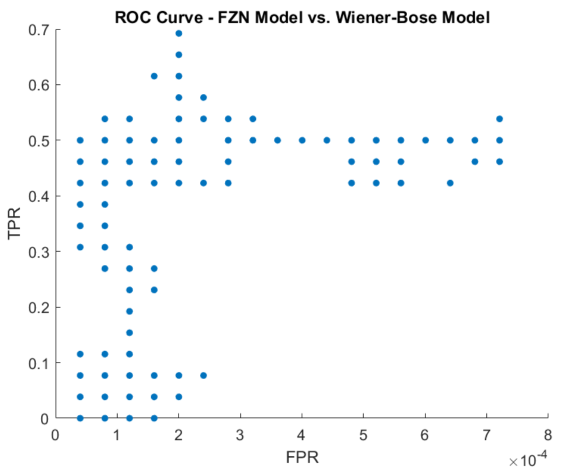
(a) ROC Curve (AUC = 0.00139)



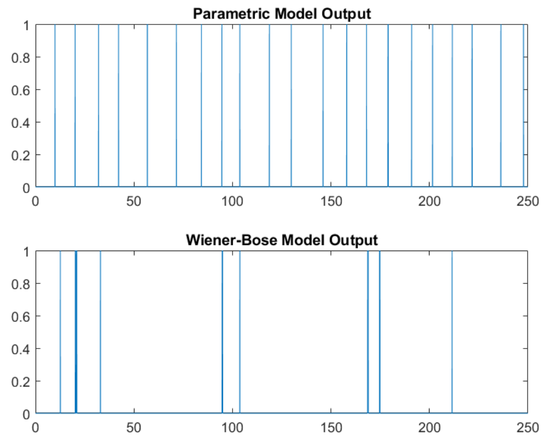
(b) Threshold Triggered output (β = 0.2501)

Figure 14: Modular Volterra Model evaluation for Oscillatory System Response

Figure 15 illustrates the ROC curve obtained for the chaotic system with the training noise sequence and the thresholded model output of the Wiener-Bose model at the optimal threshold selected. When compared with the oscillatory system, the ROC curve obtained for the chaotic system response has a weird shape and achieves an AUC of 0.00027 which is better than the random system response.

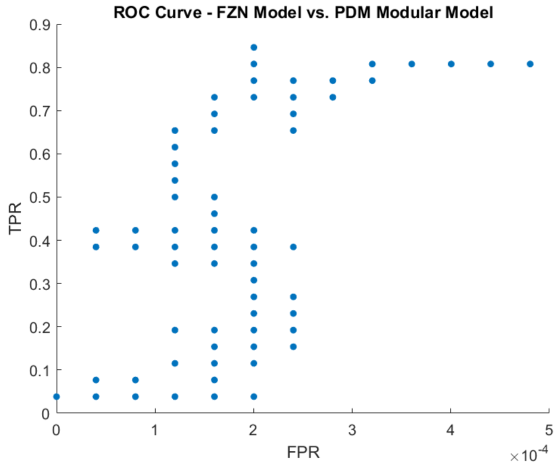


(a) ROC Curve (AUC = 0.00027)

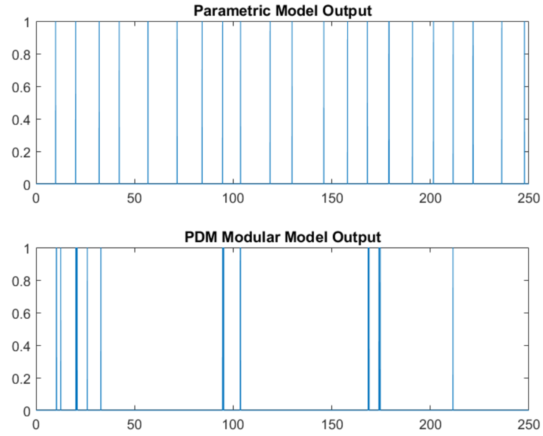


(b) Threshold Triggered output (β = 0.5373)

Figure 15: Wiener-Bose Model evaluation for Chaotic System Response



(a) ROC Curve (AUC = 0.00027)



(b) Threshold Triggered output (β = 0.5373)

Figure 16: Modular Volterra Model evaluation for Chaotic System Response

Similarly, Figure 16 illustrates the results obtained for Modular Volterra Model. The model performances are not significantly changed for this system response and the resolution of the triggered output is better at the optimal threshold compared to the oscillatory system behavior. Now that the system is evaluated with the train noise sequence, next the models are validated using a test noise sequence.

Model Validation

To validate the model, we use a different noise sequence and conduct a similar evaluation to what I did with the training noise sequence. We use the estimated non-parametric models to compute the model outputs for the given validation noise sequence. Figure 17 shows the validation noise sequence, the corresponding parametric model outputs, and the non-parametric model outputs for oscillatory system behavior.

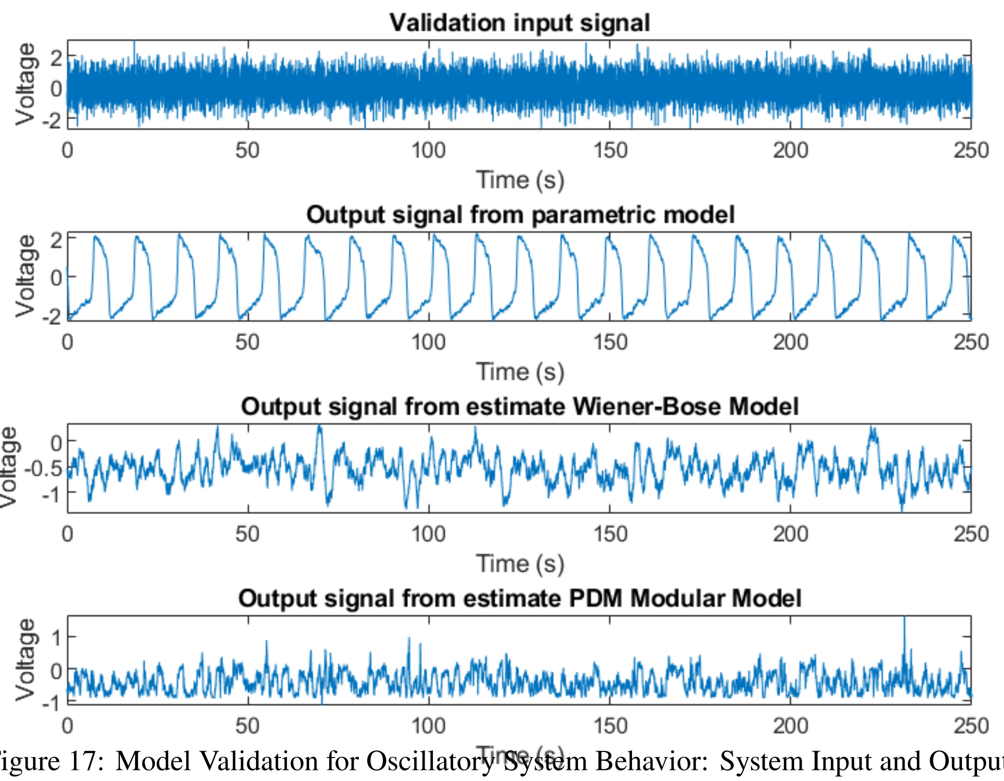
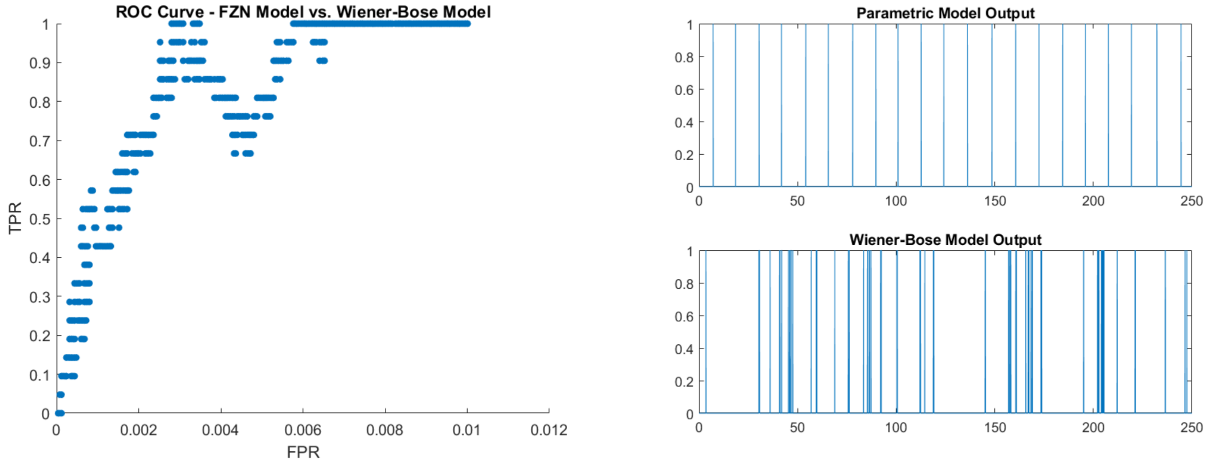


Figure 17: Model Validation for Oscillatory System Behavior: System Input and Outputs

Figure 18 illustrates the ROC curve obtained for the oscillatory system with the validation noise sequence and the thresholded model output of the Wiener-Bose model at the optimal threshold selected. Based on the ROC curve and the AUC, the performance of the Wiener-Bose model is validated and is similar to the training performance.

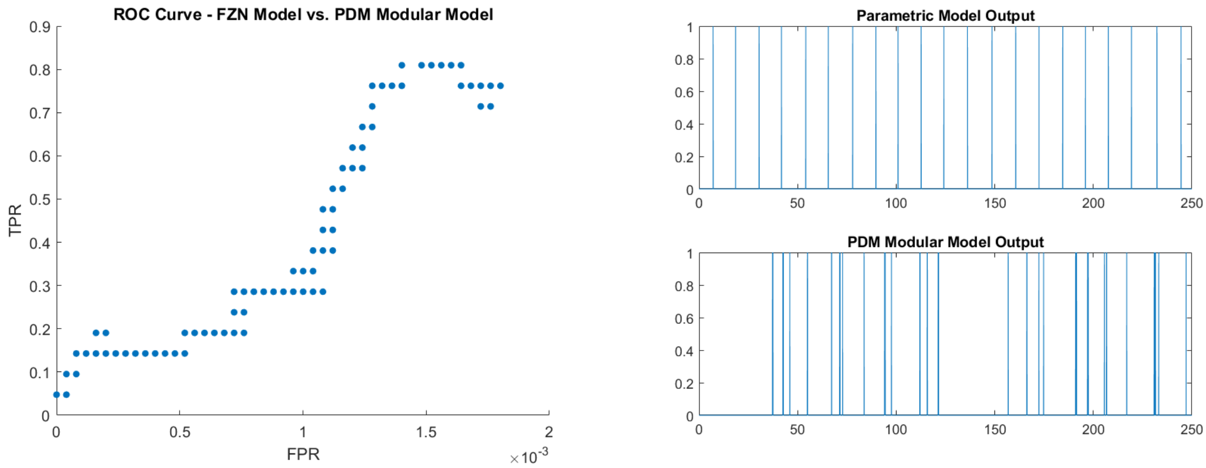


(a) ROC Curve (AUC = 0.00822)

(b) Threshold Triggered output (β = -0.1467)

Figure 18: Wiener-Bose Model model validation for Oscillatory System Response

Similar validation can be seen for the estimated Modular Volterra Model in Figure 19. The model performance is consistent with the training phase evaluation, where the Wiener-Bose Model is performing better at predicting spikes compared to the Modular Volterra Model for a given WGN.



(a) ROC Curve (AUC = 0.00073)

(b) Threshold Triggered output (β = 0.2949)

Figure 19: Modular Volterra Model model validation for Oscillatory System Response

Figure 20 shows the validation noise sequence, the corresponding parametric model outputs, and the non-parametric model outputs for the chaotic system behavior.

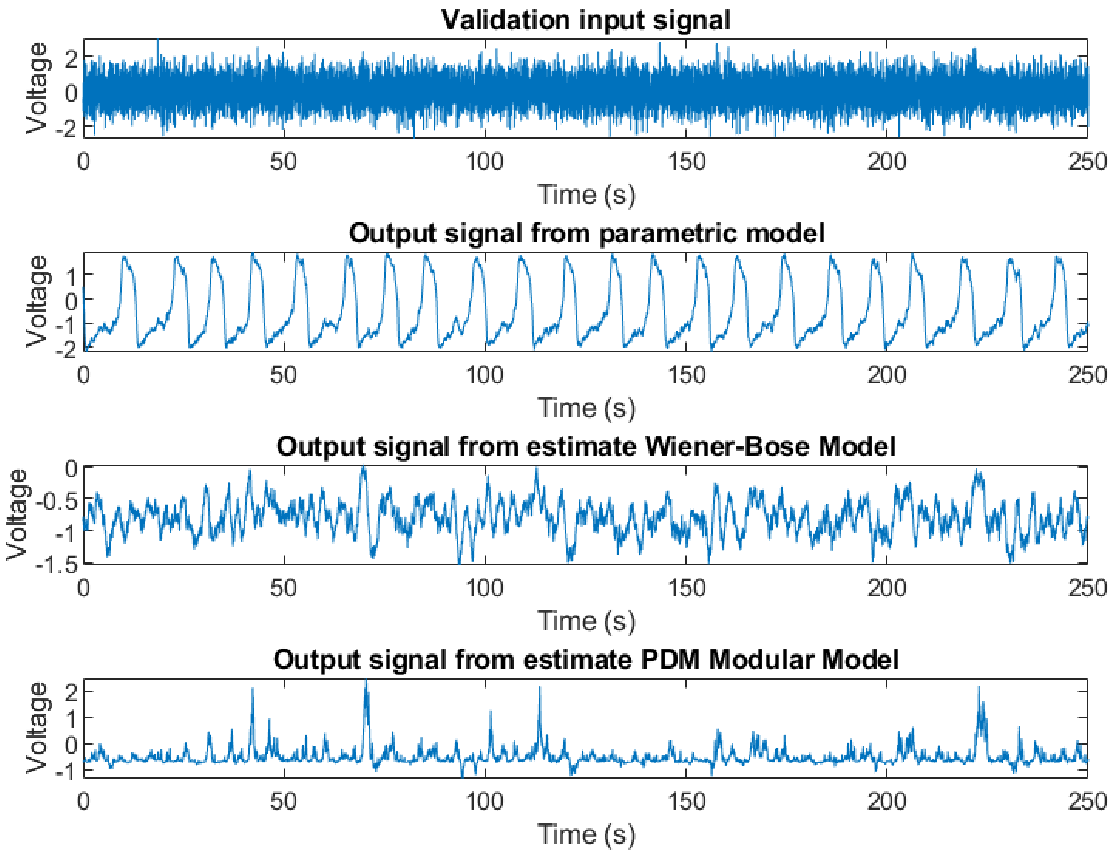
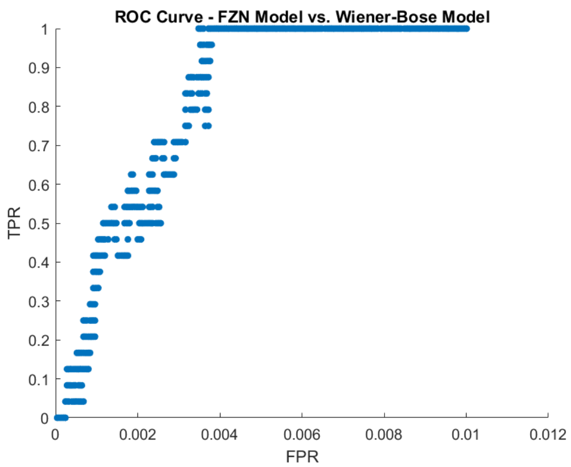
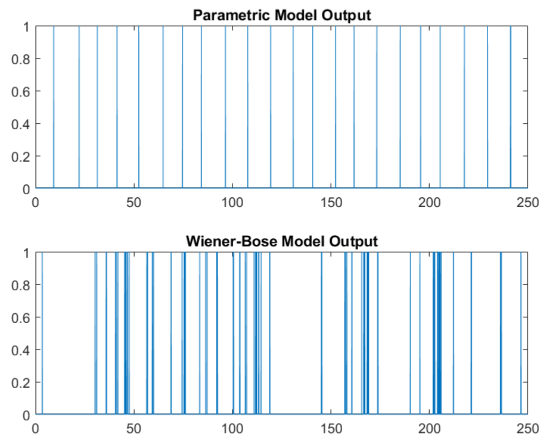


Figure 20: Model Validation for Chaotic System Behavior: System Input and Outputs

Figure 21 illustrates the ROC curve obtained for the chaotic system with the validation noise sequence and the thresholded model output of the Wiener-Bose model at the optimal threshold selected. Based on the ROC curve and the AUC, we can determine that the Wiener-Bose model estimated from the training noise sequence is a good estimate and thus validates the model.



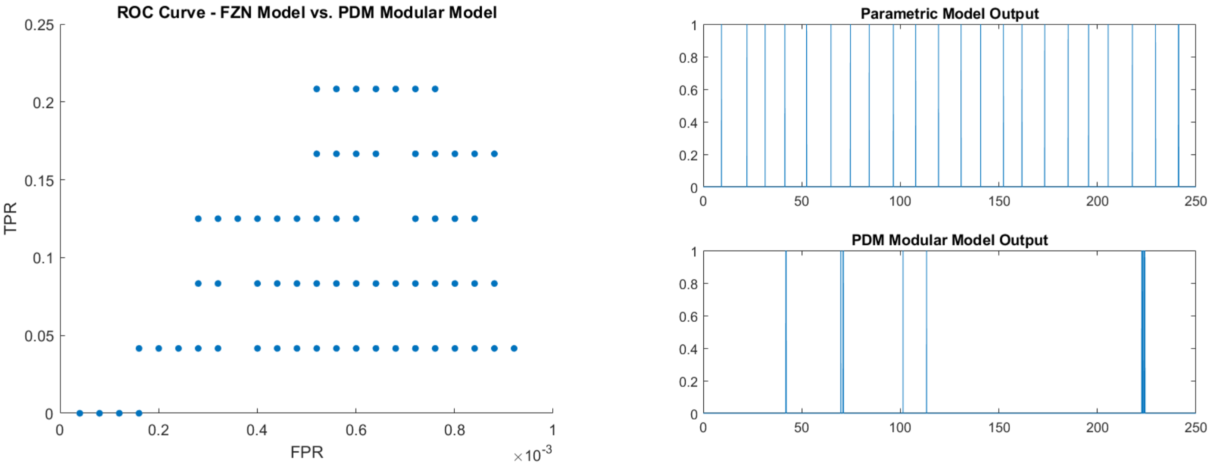
(a) ROC Curve (AUC = 0.00808)



(b) Threshold Triggered output (β = -0.4289)

Figure 21: Wiener-Bose Model model validation for Chaotic System Response

Similarly, the estimated Modular Volterra Model is validated which provides an AUC value better than the random system. However, the ROC curve does not provide a strong base to conclude the performance of the model.



(a) ROC Curve (AUC = 0.00007)
(b) Threshold Triggered output (β = 0.9498)

Figure 22: Modular Volterra Model model validation for Oscillatory System Response