Danmarks
Tekniske
Universitet

Digital Predistortion of Optical Transmitters

October 5, 2022

Contents

1 Task01

1.1 Exercise 01

We simulate three different kinds of Nonlinear functions in the exercise one. Firstly, the sigmoid function is expressed as equation (1),

$$S(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

and the shape of ideal sigmoid Nonlinear function is shown in Figure 1.

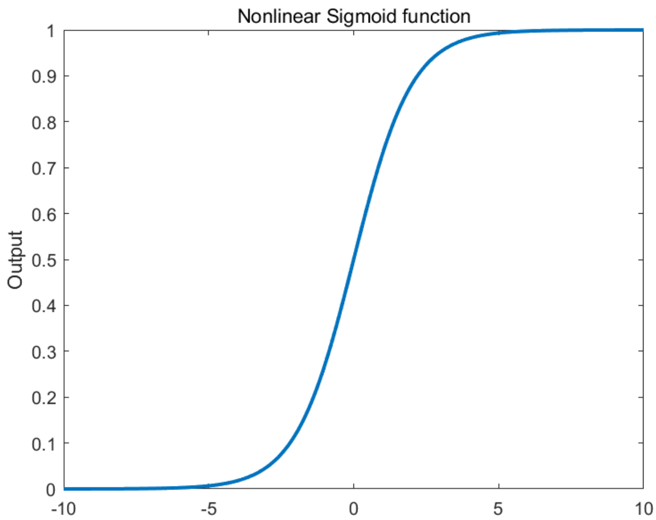


Figure 1: Nonlinear Sigmoid function

The Gaussian nonlinear function can be expressed as equation (2),

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

in this exercise, we set the variance $\sigma = 1$, and the mean value $\mu = 0$, the shape of this Gaussian function is shown in the Figure 2.

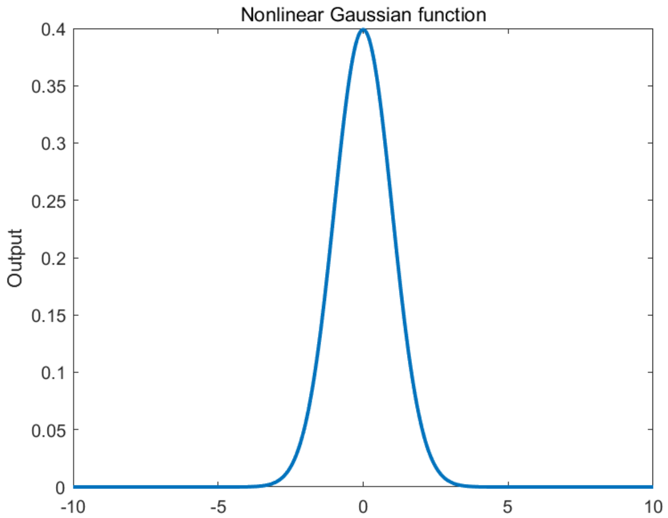
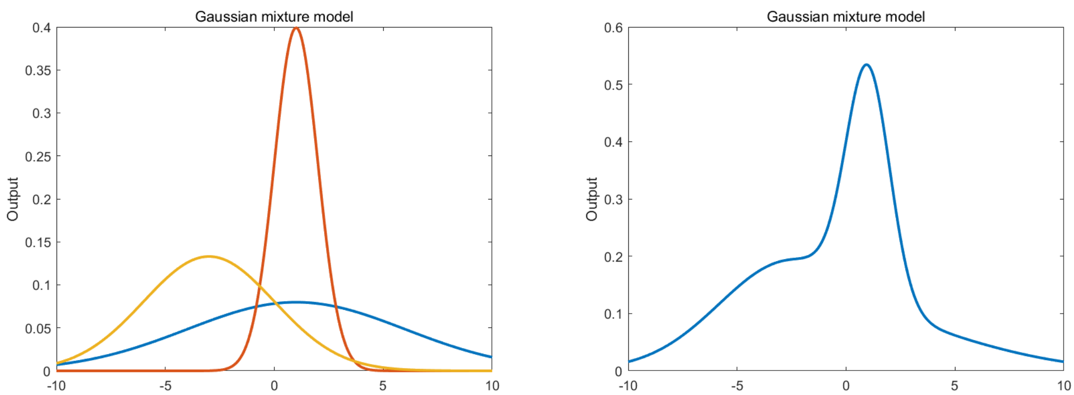


Figure 2: Nonlinear Gaussian function

Finally, We also simulate the Gaussian mixture model (GMM), the GMM can be expressed as equation (3).

$$y = \sum_{k=1}^{K} \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp -\frac{(x-\mu_k)^2}{2\sigma_k^2} \tag{3}$$

The K is equal to three in this example, the variance and the mean value are both random value, we use *randi function* to generate variance and mean value, the range of variances is $[1, 5]$ and the range of mean values is $[-5, 5]$. The random Gaussian mixture Nonlinear function is shown in Figure 3.



(a) Random three Gaussian function for GMM

(b) Gaussian Mixture Model

Figure 3: Gaussian Mixture Model and its components

1.2 Exercise 02

In this section, we show the result of the Volterra series model, we try to get the same nonlinear function just like we show in Figure 1 to 3 by using Volterra series, and we use first and second order Volterra model respectively to fit the nonlinear function. The iteration number is 1000 in this example, and the memory depth of these Volterra series are both 8.

Firstly, we introduce the simple theory of Volterra series. In traditional linear system, output can be expressed as equation (4),

$$y(t) = h \cdot x(t) \tag{4}$$

this linear function may not flexible enough for the system in realistic, so we consider the first-order Volterra series which can be regarded as a linear system with memory, in order to simulate the nonlinear function, the second order Volterra series is a better choice that can be simulated not only for memory system but also for nonlinear system, the second-order Volterra series can be expressed as equation (5)

$$y(n) = \sum_{j=0}^{n} \sum_{i=0}^{n} h_2 \left(p_i, p_j\right) x \left(n - p_i\right) \cdot x \left(n - p_j\right) \tag{5}$$

Where the $h_2 \left(p_i, p_j\right)$ is called Volterra kernels, and the parameter n may called memory length, and in this exercise n is equal to 8.

The architecture we train Volterra series is shown in Figure 4.

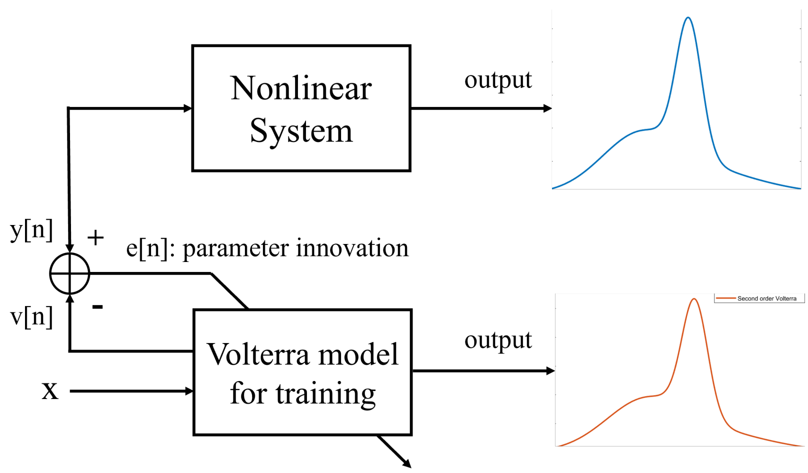


Figure 4: The training architecture of the Volterra series model nonlinear fitting

We generate a random number sequence called x as the input sequence of the Volterra series, and the parameter of Volterra series is initial as whole one matrix. For every iteration, the result of Volterra series $v[n]$ is compared with the nonlinear function, and the error $e[n]$ between nonlinear function and Volterra series will be utilized to renew the parameter of

Volterra series in a very simple way as equation (6).

$$h(n) = h(n-1) + e[n] \cdot \alpha \tag{6}$$

Where α is equal to 0.001 in this exercise, after the whole iteration, the final result of Volterra series will show in the output, the Mean square error (MSE) of training is shown in Figure 5.

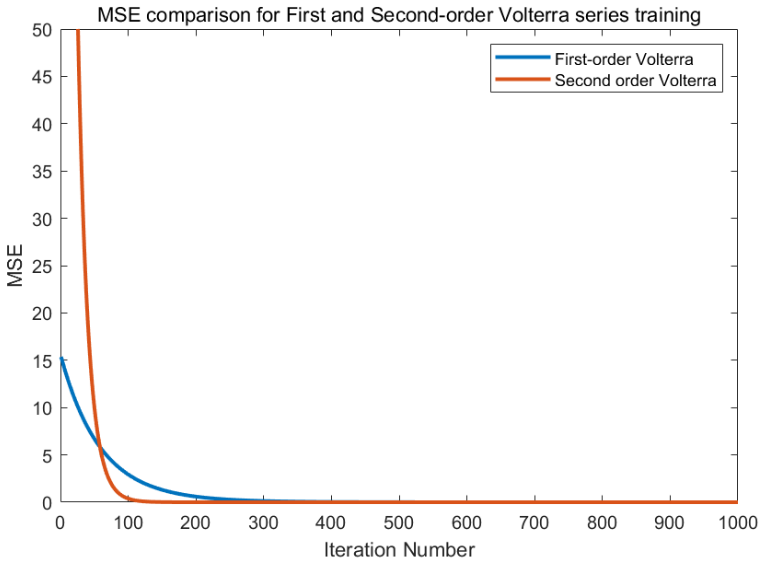
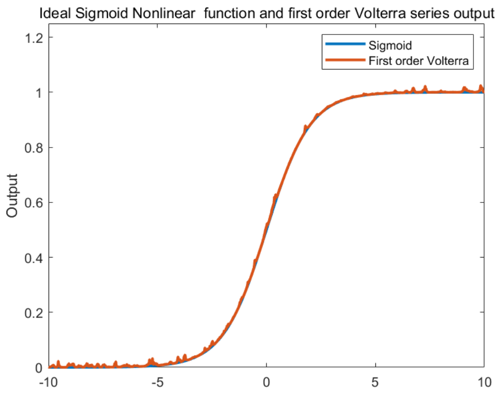
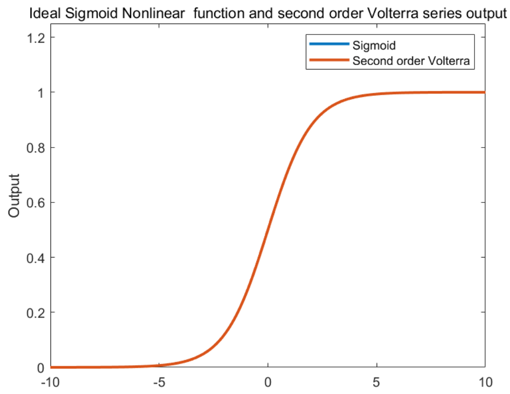


Figure 5: The MSE of training different Volterra series

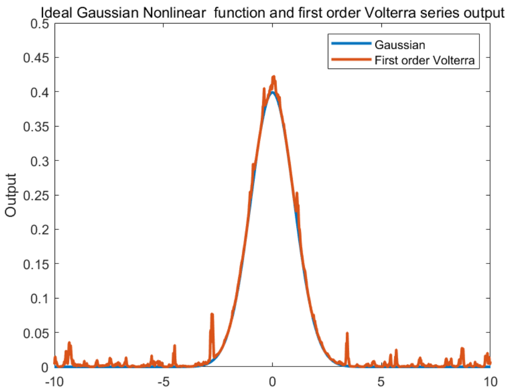
From Figure 5, We can see the MSE of Second-Volterra series is larger than First-order Volterra series at the beginning, but it converge faster with training, which may be the Second-order Volterra series have more parameters than the First one, and it can fit nonlinear function better with the iteration. The training result of different nonlinear function is shown in Figure 6. From these sub-figures in Figure 6, we can clearly observe the Second-order Volterra series has better performance than the First-order Volterra series.



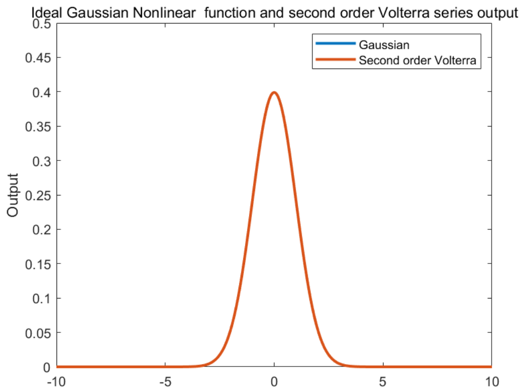
(a) Comparison of First-order Volterra sequence output and objective Sigmoid function



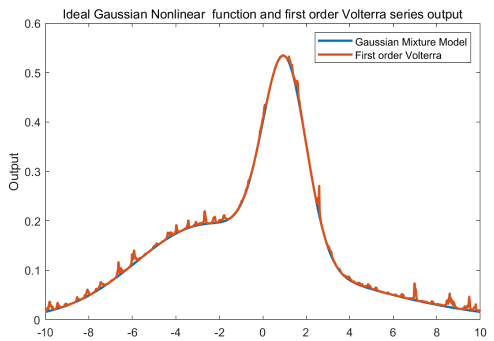
(b) Comparison of Second-order Volterra sequence output and objective Sigmoid function



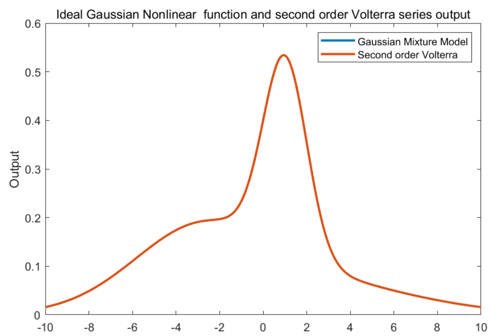
(c) Comparison of First-order Volterra sequence output and objective Gaussian function



(d) Comparison of Second-order Volterra sequence output and objective Gaussian function



(e) Comparison of First-order Volterra sequence output and objective GMM



(f) Comparison of Second-order Volterra sequence output and objective GMM

Figure 6: Comparison of Volterra sequence output and objective Nonlinear function