The Auditory System and Human Sound-Localization Behavior
Short Answers to the Exercises of Chapter 4: Nonlinear Systems

Exercise 4.1:

(a) With the input given by \( x(t) = \sin(\omega_1 t) + \sin(\omega_2 t) \), the output is found to be:

\[
y(t) = a \cdot (\sin(\omega_1 t) + \sin(\omega_2 t)) + b \cdot (\sin(\omega_1 t) + \sin(\omega_2 t))^2 + c \cdot (\sin(\omega_1 t) + \sin(\omega_2 t))^3
\]

Collecting terms yields the following 13 frequency components with their amplitudes (n.b.: all at phase 0 (positive) or \( \pi \) (negative)):

<table>
<thead>
<tr>
<th>frequency</th>
<th>amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( b )</td>
</tr>
<tr>
<td>( \omega_1 )</td>
<td>( (a + 5c/4) )</td>
</tr>
<tr>
<td>( \omega_2 )</td>
<td>( (b + 5c/4) )</td>
</tr>
<tr>
<td>( 2\omega_1 )</td>
<td>( -b/2 )</td>
</tr>
<tr>
<td>( 2\omega_2 )</td>
<td>( -b/2 )</td>
</tr>
<tr>
<td>( \omega_1 - \omega_2 )</td>
<td>( b )</td>
</tr>
<tr>
<td>( \omega_1 + \omega_2 )</td>
<td>( b )</td>
</tr>
<tr>
<td>( \omega_1 + 2\omega_2 )</td>
<td>( -3c/4 )</td>
</tr>
<tr>
<td>( \omega_1 - 2\omega_2 )</td>
<td>( -3c/4 )</td>
</tr>
<tr>
<td>( \omega_2 + 2\omega_1 )</td>
<td>( -3c/4 )</td>
</tr>
<tr>
<td>( \omega_2 - 2\omega_1 )</td>
<td>( -3c/4 )</td>
</tr>
<tr>
<td>3( \omega_1 )</td>
<td>( -c/4 )</td>
</tr>
<tr>
<td>3( \omega_2 )</td>
<td>( -c/4 )</td>
</tr>
</tbody>
</table>

(b) The Matlab script is found on the Book’s web page as *Chapter4-Exc4-1.m*

Exercise 4-2:

(a) The function \( y(t) = \log(x(t) + 1) \) is defined on \( |x(t)| < 1 \), for which the Taylor expansion thus yields

\[
y(t) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} \cdots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n}
\]

The associated Volterra kernels for this nonlinear function are thus determined by:

\[
k_n(\tau_1, \tau_2, \cdots, \tau_n) = \frac{(-1)^{n+1}}{n} \prod_{k=1}^{n} \delta(\tau_k)
\]
(b) For the exponential function $y = exp(x(t))$, the Volterra series reads

$$k_0 = 1 \quad \text{and} \quad k_n(\tau_1, \tau_2, \cdots, \tau_n) = \frac{1}{n!} \prod_{k=1}^{n} g(\tau_k)$$

(c) The cosine function $y(t) = cos(x(t))$ is expanded as

$$k_0 = 1 \quad \text{and} \quad k_n(\tau_1, \tau_2, \cdots, \tau_n) = \frac{(-1)^n}{(2n)!} \prod_{k=1}^{2n} \delta(\tau_k) \quad \text{for} \quad n \geq 1$$

Exercise 4-3:
(a) The following holds: $\int_0^T f^2(x) \cdot dx < \infty$ ($f(x)$ is an absolute integrable function on the interval $[0, T]$). We substitute the orthogonality condition, to find:

$$I_N = \int_0^T f^2(x)dx - 2 \sum_{n=1}^{N} a_n \int_0^T f(x)w_n(x)dx + \sum_{n=1}^{N} a_n^2$$

We rewrite:

$$I_N = \int_0^T f^2(x)dx + \sum_{n=1}^{N} \left[ a_n - \int_0^T f(x)w_n(x)dx \right]^2 - \sum_{n=1}^{N} \left[ \int_0^T f(x)w_n(x)dx \right]^2$$

We thus have to minimize this term, which happens when:

$$a_n = \int_0^T f(x)w_n(x)dx$$

(b) The mimimum integrated error is:

$$I_N(\text{min}) = \int_0^T f^2(x)dx - \sum_{n=1}^{N} a_n^2$$

And the following must hold

$$\sum_{n=1}^{N} a_n^2 \leq \int_0^T f^2(x)dx$$

This is the so-called Bessel inequality. When the orthogonal basis set is complete, it follows that

$$\lim_{N \to \infty} I_N(\text{min}) = 0$$
Exercise 4-4:
The general second-order inhomogeneous Volterra functional reads:
\[ G_2[x(t), P] = k_{0,2} + \int_0^\infty k_{12}(\tau) \cdot x(t - \tau) d\tau + \int_0^\infty \int_0^\infty k_{22}(\tau_1, \tau_2) \cdot x(t - \tau_1) \cdot x(t - \tau_2) d\tau_1 d\tau_2 \]
and the first two Wiener functionals are
\[ G_0 = h_0 \quad \text{and} \quad G_1[x(t), P] = \int_0^\infty h_1(\tau) \cdot x(t - \tau) d\tau \]
We have to determine \( k_{0,2}, k_{12}(\tau) \) and \( k_{22}(\tau_1, \tau_2) \). We demand orthogonality by setting the expectation values of the inner products to zero:
\[ G_0 \cdot G_2 = 0 \quad \text{and} \quad G_1 \cdot G_2 = 0 \]
It then follows (using the autocorrelation properties of GWN, Ewn. 4.16):
\[ G_2[x(t), P] = \int_0^\infty \int_0^\infty h_2(\tau_1, \tau_2) \cdot x(t - \tau_1) x(t - \tau_2) d\tau_1 d\tau_2 - P \cdot \int_0^\infty h_2(\tau, \tau) d\tau \]

Exercise 4-5:
The first-order Wiener functional is:
\[ G_1[h_1; x(t), P] = \int h_1(\tau) \cdot x(t - \tau) d\tau \]
and the third-order Wiener functional reads:
\[ G_3[h_3; x(t), P] = \int \int h_3(\tau_1, \tau_2, \tau_3) x(t - \tau_1) \cdot x(t - \tau_2) x(t - \tau_3) d\tau_1 d\tau_2 d\tau_3 - 3P \cdot \int \int h_3(\tau_1, \tau_2, \tau_2) x(t - \tau_1) d\tau_1 \]
We calculate the expectation value of the inner product:
\[ G_1[h_1; x(t), P] \cdot G_3[h_3; x(t), P] \]
and note that the two terms of \( G_1 \cdot G_3 \) will contain a double product and a four-product of GWN samples. It follows that
\[ G_1 \cdot G_3 = 3P^2 \int \int h_1(\tau_p) h_3(\tau_p, \tau_q, \tau_q) d\tau_p d\tau_q - 3P^2 \int \int h_1(\tau_1) h_3(\tau_1, \tau_2, \tau_2) d\tau_1 d\tau_2 = 0 \]

Exercise 4-6:
The output of the system in Fig 4.5 is described as follows:

\[
y(t) = a \cdot w(t) + b \cdot w^2(t) = \int_0^\infty a g(\tau) \cdot x(t - \tau) d\tau + \int_0^\infty \int_0^\infty b g(\tau_1) g(\tau_2) x(t - \tau_1) x(t - \tau_2) d\tau_1 d\tau_2
\]

from which the Volterra kernels are given by:

\[
k_0 = 0, \\
k_1(\tau) = a g(\tau), \\
k_2(\tau_1, \tau_2) = b g(\tau_1) g(\tau_2)
\]

The Lee and Schetzen method (Eqns. 4.30-4.31) applies crosscorrelation between input and output to determine the Wiener kernels. This is how this works (we omitted all odd-numbered products in \(x(t)\), as these yield zero):

\[
h_0 = E[y(t)] = P b \int_0^\infty g^2(\tau_1) d\tau_1
\]

\[
h_1(\sigma) = \frac{1}{P} E[y(t)x(t - \sigma)] = a g(\sigma)
\]

\[
h_2(\sigma_1, \sigma_2) = \frac{1}{2P^2} E[(y(t) - h_0)x(t - \sigma_1)x(t - \sigma_2)] = b g(\sigma_1) g(\sigma_2)
\]

Indeed, the first-order and second-order Volterra and Wiener kernels are identical.

**Exercise 4-7:**

The fourth Hermite polynomial with respect to a GWN signal with power \(P\) reads:

\[
H_4 = x^4 - 6P x^2 + 3P^2
\]

From the analogy in Eqn. 4.18, the fourth-order Wiener functional can thus be constructed:

\[
G_4[h_4; x(t), P] = \int \int \int h_4(\tau_1, \tau_2, \tau_3, \tau_4) \cdot \prod_{n=1}^4 x(t - \tau_n) d\tau_n + \int \int \int \int \int h_4(\tau_1, \tau_2, \tau_3, \tau_4) x(t - \tau_1) x(t - \tau_2) x(t - \tau_3) x(t - \tau_4) d\tau_1 d\tau_2 d\tau_3 d\tau_4 + \int \int \int \int \int \int h_4(\tau_1, \tau_2, \tau_3) x(t - \tau_1) x(t - \tau_2) x(t - \tau_3) x(t - \tau_4) d\tau_1 d\tau_2 d\tau_3 d\tau_4 + \int \int \int \int \int \int \int \int \int \int \int \int \int h_4(\tau_1, \tau_2, \tau_3, \tau_4) x(t - \tau_1) x(t - \tau_2) x(t - \tau_3) x(t - \tau_4) d\tau_1 d\tau_2 d\tau_3 d\tau_4
\]

**Exercise 4-8:**
From Fig. 4.11, we obtain the following relationships:

\[
y(t) = \int_0^\infty \int_0^\infty a_k(\sigma)h(\tau)x(t-\sigma-\tau)d\sigma d\tau + \\
+ \int_0^\infty \int_0^\infty b_k(\sigma)h(\tau_1)h(\tau_2)x(t-\sigma-\tau_1) \cdot x(t-\sigma-\tau_2)d\sigma d\tau_1 d\tau_2
\]

For the Wiener kernels we take \( x(t) = \text{GWN}, \text{power} \ P \) and mean zero.

Taking the expectation value of the output yields the zero-order Wiener kernel (we leave out the odd-numbered products in \( x(t) \)):

\[
h_0 = E[y(t)] = P \cdot b \cdot \left[ \int_0^\infty h(\tau)d\tau \right]^2 \cdot \left[ \int_0^\infty k(\sigma)d\sigma \right]
\]

The first-order Wiener kernel is found by cross-correlation:

\[
h_1(\lambda) = aP \cdot \int_0^\infty k(\sigma)h(\lambda-\sigma)d\sigma
\]

Finally, we find the second-order Wiener kernel from the third-order cross-correlation:

\[
\phi_{yxx}(\lambda_1, \lambda_2) = E[y(t)x(t-\lambda_1)x(t-\lambda_2)]
\]

Corrected for the average, \( h_0 \), the second-order Wiener kernel is given by:

\[
h_2(\lambda_1, \lambda_2) = \frac{\phi_{y-h_0}xx}{2P^2} = \frac{b}{2} \int_0^\infty k(\sigma)h(\sigma-\lambda_1)h(\sigma-\lambda_2)d\sigma
\]

Exercise 4-9:
In Fig. 4.13 we calculate:

\[
u(t) = \int_0^\infty g(\tau)x(t-\tau)d\tau \quad \text{and} \quad y(t) = [u(t)]^2
\]

so that

\[
y(t) = \int_0^\infty \int_0^\infty g(\tau_1)g(\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2
\]

from which it follows that \( k_0 = 0 \), \( k_1(\tau) = 0 \) and \( k_n(\tau_1, \tau_2, \ldots, \tau_n) = 0 \) for \( n \geq 3 \), and

\[
k_2(\tau_1, \tau_2) = A^2 \cdot \exp(-k(\tau_1 + \tau_2)) \cdot \sin(m\tau_1) \cdot \sin(m\tau_2)
\]

with \( A = 6.67, k = 0.08 \) and \( m = 0.3 \).

Exercise 4-10:
For this exercise, the reader is referred to the Matlab section for Chapter4-Exc4-10.m
Exercise 4-11:
We use the ‘hint’ to determine the $n$-th order expectation value of the output $y(t)$ (or: the $n$-th order autocorrelation function of $y(t)$), which is given by taking the following time-average:

$$
y(t - \sigma_1) \cdots y(t - \sigma_n) = \int_0^\infty \cdots \int_0^\infty h(\tau_1) \cdots h(\tau_n) x(t - \tau_1 - \sigma_1) \cdots x(t - \tau_n - \sigma_n) d\tau_1 \cdots d\tau_n
$$

We assume that $x(t)$ is a Gaussian process, with average zero. For odd values of $n$, i.e., when it can be written as $n = 2m + 1$ for $m = 0, 1, 2, 3, \ldots$, we see that

$$
\frac{y(t - \sigma_1)y(t - \sigma_2) \cdots y(t - \sigma_{2m+1})}{y(t - \sigma_1) \cdots y(t - \sigma_n)} = 0
$$

For even values of $n$, when we write $n = 2m$, the following holds (using Eqn. 4.16):

$$
\frac{y(t - \sigma_1) \cdots y(t - \sigma_{2n})}{y(t - \sigma_1) \cdots y(t - \sigma_{2n})} = \Sigma \Pi y(t - \tau_i - \sigma_i)y(t - \tau_j - \sigma_j)
$$

We conclude that also $y(t)$ has to be a Gaussian process!