3 Volterra Series

3.1 Introduction

Most physiological systems cannot be modeled successfully as linear systems. At best, a linear model can be considered an approximation of physiological activity in cases where the output of a physiological system behaves (almost) linearly over a limited range of the input. In the following, we extend the convolution integral that describes the behavior of linear devices to the convolution-like Volterra series, which can be used to represent nonlinear systems. Because the expressions for higher-order nonlinear terms require significant computational resources and become very complex to deal with, we will demonstrate the general principles for second-order systems. See Schetzen (2006) if you are interested in details of higher-order systems.

In a linear time invariant (LTI) system, the convolution integral links output $y(t)$ and input $x(t)$ by means of its weighting function $h(t)$ (Fig. 3.1) (Chapter 8 in van Drongelen, 2007):

$$y(t) = h(t) \otimes x(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau) \mathrm{d}\tau \tag{3.1}$$

Here $\otimes$ symbolizes the convolution operation and the system's weighting function $h(t)$ is its unit impulse response (UIR). This role of $h(t)$ can be verified by using a unit impulse $\delta(t)$ as the input. In this case we obtain (using the sifting property):

$$\int_{-\infty}^{\infty} h(\tau)\delta(t-\tau)\mathrm{d}\tau = h(t) \tag{3.2}$$

Signal Processing for Neuroscientists, A Companion Volume. DOI: 10.1016/B978-0-12-384915-1.00003-6

Note: In the following we will use the sifting property of the unit impulse repeatedly (for a discussion, see section 2.2.2 in van Drongelen, 2007). The sifting property is defined as:

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(\tau - t)\mathrm{d}\tau = \int_{-\infty}^{\infty} x(\tau)\delta(t - \tau)\mathrm{d}\tau$$

The unit impulse δ has properties of a function with even symmetry; therefore, the evaluation of the integral above is the same for $\delta(t - \tau)$ and $\delta(\tau - t)$. You can also see that this must be the case since the outcome is $\delta(0)$ for $t = \tau$ in both cases, $\delta(t - \tau)$ and $\delta(\tau - t)$.

Such an LTI system shows superposition and scaling properties. For instance, if we introduce a **scaled delta function** $C\delta(t)$ (C—constant) at the input, we get a **scaled UIR function** $Ch(t)$ at the output:

$$\int_{-\infty}^{\infty} h(\tau)C\delta(t - \tau)\mathrm{d}\tau = C\int_{-\infty}^{\infty} h(\tau)\delta(t - \tau)\mathrm{d}\tau = Ch(t) \tag{3.3}$$

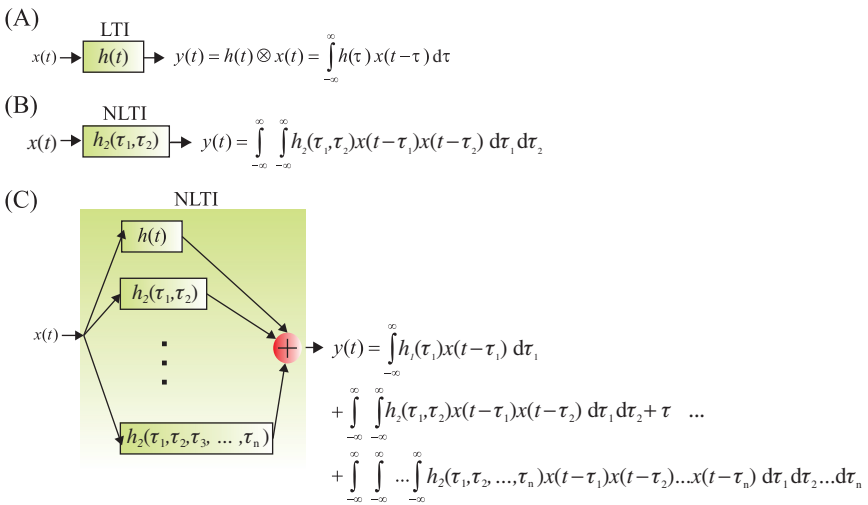

Figure 3.1 Example of LTI and NLTI systems. (A) A linear system. (B) A second-order system and (C) a combined *n*th order system. The output of the first-order components is determined by the convolution integral, and the output of higher-order components is obtained from convolution-like integrals. The output of the *n*th order system is represented by a Volterra series consisting of the sum of the individual components, each determined by a convolution-like expression.

Now consider a system that is governed by an equation that is convolution-like:

$$y(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \tag{3.4}$$

Unlike the convolution in Equation (3.1), this system works on two copies of input $x(t)$ instead of only one. As we discussed in Section 2.5, such a system is an example of a so-called second-order Volterra system. Note that the double integral in Equation (3.4) is identical to the last term in the expression in Equation (2.24). If we determine the UIR for the system in Equation (3.4), we get:

$$h_2(t, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) \delta(t - \tau_1) \delta(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \tag{3.5}$$

Here we applied the sifting property twice: once for each of the delays τ_1 and τ_2. The result $h_2(t,t)$ is the diagonal of kernel h_2.

> *Note*: You can see that in a second-order Volterra system, the UIR $h_2(t,t)$ does not fully characterize the system (unlike the situation in an LTI system). Instead it only characterizes the 2D function $h_2(\tau_1, \tau_2)$ along the diagonal $\tau_1 = \tau_2$ in the τ_1, τ_2 plane. As we will see in Section 3.3 we need sets of paired impulses to fully characterize h_2.

The system in Equation (3.4) is nonlinear because scaling does not hold. For example, the response to a scaled delta function $C\delta(t)$ at the input is:

$$\begin{aligned} &\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) C\delta(t - \tau_1) C\delta(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 = \\ &C^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) \delta(t - \tau_1) \delta(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 = C^2 h_2(t, t) \end{aligned} \tag{3.6}$$

By comparing Equations (3.5) and (3.6) we can see that in this system the UIR $h_2(t,t)$ scales with C^2 instead of C. As we will show in Section 3.2.1, superposition does not hold for this system either, but showing that scaling does not hold is sufficient to negate linearity of the system. In the remainder of this chapter we will continue our introduction of Section 2.5 by studying the properties of the Volterra series and applying it for the characterization of higher-order systems.

3.2 Volterra Series

The mathematician Vito Volterra used series of convolution-like expressions to define the input−output relationship of NLTI systems:

$$y(t) = \overbrace{\int_{-\infty}^{\infty} h_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1}^{\text{1st order term}} + \overbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2}^{\text{2nd order term}} + \cdots + \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h_n(\tau_1,\tau_2,\ldots,\tau_n)x(t-\tau_1)x(t-\tau_2)\ldots x(t-\tau_n)\mathrm{d}\tau_1\,\mathrm{d}\tau_2\ldots\mathrm{d}\tau_n}_{n\text{th order term}} \tag{3.7}$$

The output $y(t)$ of an nth order system depends on multiple copies of the input and is the sum of the 1st, 2nd, …, nth order convolution-like expressions. The functions h_1, h_2, …, h_n are called the 1st, 2nd, …, nth order **Volterra kernels**. In some texts, a zero-order kernel (h_0) representing a DC term, or offset, is added to Equation (3.7). Just as in an LTI system, $y(t)$ is the UIR if the input $x(t)$ is a unit impulse $\delta(t)$. In higher-order systems, the contribution of the nth order Volterra kernel to the UIR is a so-called diagonal slice through the kernel, that is, a section through the kernel with all delays τ_1, τ_2, …, τ_n equal. An example for a second-order system ($n = 2$) is shown in Equation (3.5).

> *Note*: We can refer to h_1 as the UIR only if we deal with a first-order Volterra system without a DC term—that is, a (linear) system where h_1 is the only term of $y(t)$. In all other cases, the UIR is determined by the contributions of all of the system's Volterra kernels and not just by h_1.

If we represent the 1st, 2nd, …, nth order convolution-like terms in Equation (3.7) as H_1, H_2, …, H_n we get an alternative, simplified notation:

$$y(t) = H_1[x(t)] + H_2[x(t)] + \cdots + H_n[x(t)] \tag{3.8}$$

Equation (3.8) can be generalized for an nth order system:

$$y(t) = \sum_{n=1}^{N} H_n[x(t)] \tag{3.9a}$$

In some cases a DC term $H_0[x(t)] = h_0$ (with h_0 being a constant) is added. This allows one to further generalize Equation (3.8) for the NLTI system to:

$$y(t) = \sum_{n=0}^{N} H_n[x(t)] \tag{3.9b}$$

Just as with any series, we should consider the convergence of the Volterra series. In our case, we approach this by optimistically assuming that any system we consider will be associated with a converging series. We can afford this optimism because we will apply the Volterra series only to known, relatively low-order systems and because we would immediately notice if the output predicted by the Volterra series would poorly match the measured output of the system it represents.

Recall that we can consider the Volterra series' approximation of output as a Taylor series with the addition of memory (Section 2.5). The Taylor series links output with instantaneous input (no memory, Equation (2.16)), whereas the Volterra series includes a memory component in the convolution-like integrals. These integrals show that the output at time t depends not only on current input signal $x(t)$, but on multiple copies of the delayed input, represented by $x(t-\tau_1)$, $x(t-\tau_2), \ldots, x(t-\tau_n)$ in the integrals in Equation (3.7).

3.2.1 Combined Input to a Second-Order Volterra System

In general, the input−output relationship of a second-order Volterra system without lower-order components can be specified by Equation (3.4). We also demonstrated above that a second-order Volterra system does not scale as an LTI system (compare Equations (3.3) and (3.6)). We can next show that the **superposition property** of an LTI system does not hold in the second-order Volterra system either. To accomplish this, we will determine the system's response to the sum of two inputs $x(t) = x_1(t) + x_2(t)$ relative to its responses to $x_1(t)$ and $x_2(t)$ individually. The response to the combined inputs is:

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)[x_1(t-\tau_1)+x_2(t-\tau_1)][x_1(t-\tau_2)+x_2(t-\tau_2)]\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \end{aligned} \tag{3.10}$$

In Equation (3.10) we have the following four terms:

$$H_2[x_1(t)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)x_1(t-\tau_1)x_1(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \tag{3.11a}$$

$$H_2[x_2(t)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x_2(t - \tau_1) x_2(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \tag{3.11b}$$

$$\left.\begin{aligned} H_2[x_1(t), x_2(t)] &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x_1(t - \tau_1) x_2(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \\ H_2[x_2(t), x_1(t)] &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x_1(t - \tau_2) x_2(t - \tau_1) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \end{aligned}\right\} \text{cross-terms} \tag{3.11c}$$

Note that Equations (3.11a) and (3.11b) represent the expressions for the system's response when its input would be x_1 and x_2, respectively. The two cross-terms in expression (3.11c) are determined by both x_1 and x_2 and can be considered equal because **the second-order kernel is symmetric**—that is, $h(\tau_1, \tau_2) = h(\tau_2, \tau_1)$.

Note: The symmetry of $h(\tau_1, \tau_2, \ldots, \tau_n)$: Recall that the kernel h of a linear system is obtained from the system's response to a unit impulse. As we will see in the following section, h can be determined in higher-order systems from the responses to multiple unit impulses. Since kernel h can be obtained from responses to combinations of unit impulses, the symmetry assumption makes sense. This is the case because there is no reason to assume that a system would be able to distinguish (i.e., react differently) between unit impulse 1 followed by unit impulse 2 as compared to unit impulse 2 followed by unit impulse 1 (the unit impulses are indistinguishable because they are identical). For a formal explanation see chapter 3 in Schetzen (2006). If you have problems following this reasoning, you may come back to it after studying the concrete example in Pr3_1.m and Section 3.3.

Based on the symmetry, we can rewrite the second equation in (3.11c) as:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_2, \tau_1) x_1(t - \tau_2) x_2(t - \tau_1) \mathrm{d}\tau_2 \, \mathrm{d}\tau_1 \tag{3.11d}$$

If we now interchange the dummy variables τ_1 and τ_2 this becomes:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x_1(t - \tau_1) x_2(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \tag{3.11e}$$

This result indicates that the two expressions in (3.11c) are equal, so we may combine the cross-terms into:

$$2H_2[x_1(t), x_2(t)] = 2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x_1(t - \tau_1) x_2(t - \tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \tag{3.11f}$$

By combining Equations (3.10), (3.11a), (3.11b), and (3.11f), we get the following expression for the output $y(t)$ for the sum of the inputs $x_1(t) + x_2(t)$:

$$y(t) = H_2[x(t)] = H_2[x_1(t) + x_2(t)] = H_2[x_1(t)] + H_2[x_2(t)] + 2H_2[x_1(t), x_2(t)] \tag{3.12}$$

The cross-terms $2H_2[x_1(t), x_2(t)]$ in Equation (3.11f) represent the deviation of the second-order Volterra system's response from the response to $x_1(t) + x_2(t)$ if superposition were to hold, that is, in the second-order Volterra system the total response $y(t)$ to $x_1(t) + x_2(t)$ is **not** equal to the sum (superposition) of the responses to $x_1(t)$ and $x_2(t)$ individually: $H_2[x_1(t)] + H_2[x_2(t)]$.

3.3 A Second-Order Volterra System

As we discussed in Chapter 2, we can create a dynamical nonlinear system by combining a dynamical linear system (L) and a static nonlinear (N) one (Figs. 2.2 and 3.2A); the type of system that emerges from this combination is often indicated as an LN cascade. In the example in Fig. 3.2A, the dynamical linear system is a simple low-pass filter consisting of a resistor (R) and capacitor (C) ($h_{RC} = 1/(RC)e^{-t/RC}$). If you need to refresh your basic knowledge about RC filters, see chapters 10 and 11 in van Drongelen (2007). The static nonlinear component in Fig. 3.2A relates an input to the square of the output, that is, output $y(t)$ is the square of its input $f(t)$: $y(t) = f(t)^2$. From this relationship, the static nonlinear component is considered a squarer. Following the procedure described in Section 2.5, we can establish that this cascade is a second-order Volterra system with a second-order kernel:

$$h_2(\tau_1, \tau_2) = h_{RC}(\tau_1) h_{RC}(\tau_2) = \left(\frac{1}{RC} e^{-\tau_1/RC}\right)\left(\frac{1}{RC} e^{-\tau_2/RC}\right) = \left(\frac{1}{RC}\right)^2 e^{-(\tau_1 + \tau_2)/RC} \tag{3.13}$$

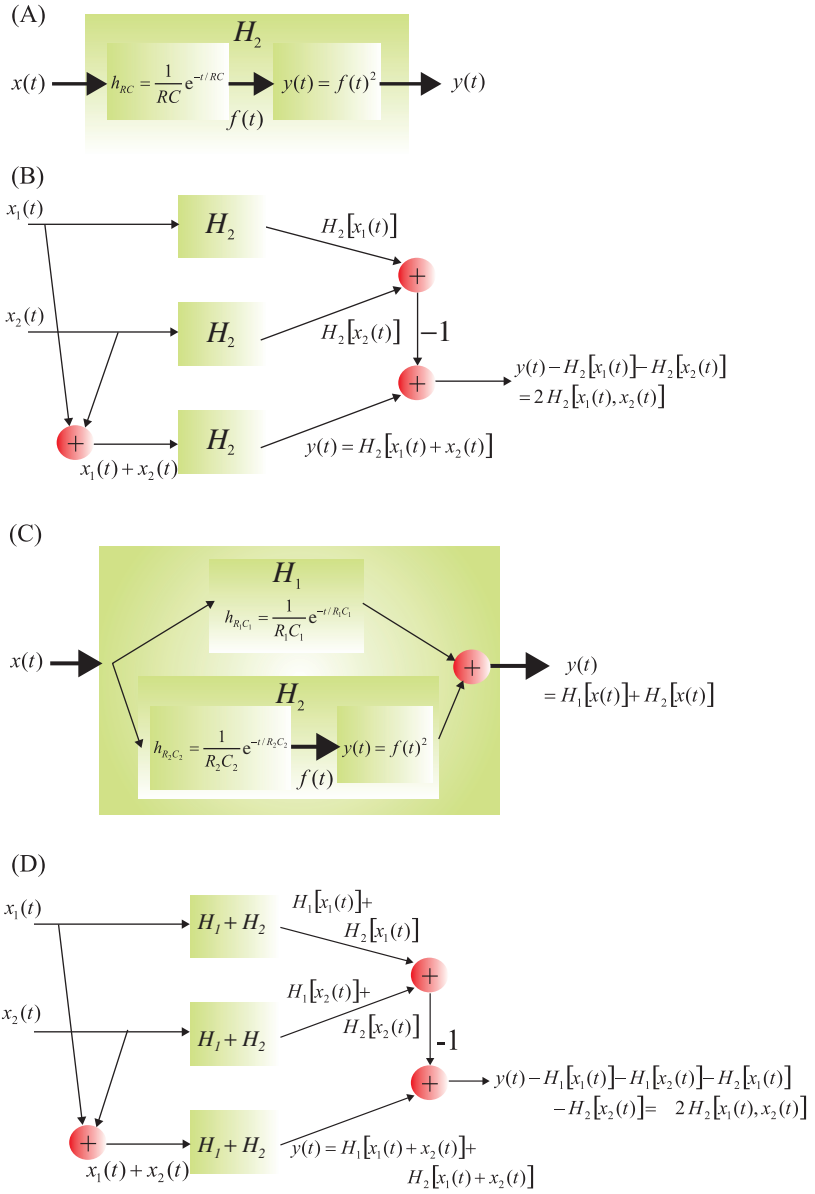

Figure 3.2 (A) Nonlinear system consisting of a cascade of a linear filter (a dynamical system) and a squarer (a static system). (B) Procedure to compute H_2 following Equation (3.12). This procedure is one of the approaches used to determine kernel h_2 in Pr3_1.m. (C) A general second-order Volterra system with a first-order (H_1) and second-order (H_2) operator. (D) The same procedure as in (B) applied to the second-order system with a first-order component. This procedure is followed in script Pr3_2.m.

The following MATLAB code (the first part of Pr3_1.m available on http://www.elsevierdirect.com/companions/9780123849151) will create an image of the 2D Volterra kernel (Fig. 3.3) based on the known structure of the nonlinear cascade (Equation (3.13)).

```
% Linear component is a low-pass RC circuit
% we use R=10k and C=3.3uF
R=10e3;
C=3.3e-6;
RC=R*C;

% Timing parameters
sample_rate=1000;
dt=1/sample_rate;
time=0.1;
A=RC/dt;
T=100;              % The setting for timing and the length of correlation
                    % calculations for both Volterra and Wiener kernels
% Step 1. The analog continuous time approach using the square of
% the unit impulse response of the filter: h(t)=(1/RC)*exp(-t/RC)
% to compare with discrete time approach (in the following Steps) we assume
% the discrete time steps (dt) and interval (time)
j=1;
for tau1=0:dt:time;
      i=1;
      r1(j)=(1/RC)*exp(-tau1/RC);   % 1st-order response in the cascade
      for tau2=0:dt:time
         y(i,j)=((1/RC)*exp(-tau1/RC))*((1/RC)*exp(-tau2/RC));
                    % Output y is h2 (=2nd order Volterra kernel)
                    % which is the square of the filter response
         i=i+1;
      end;
      j=j+1;
end;
% plot the surface of h2
y=y*dt^2;           % scale for the sample value dt
figure; surf(y);
axis([0 T 0 T min(min(y)) max(max(y))])
view(100,50)
title('2nd order Volterra kernel (h2) of an LN cascade')
xlabel('tau1');ylabel('tau2');zlabel('h2');
```

Now we validate our (nonparametric) approach with the Volterra series by using a parametric model, the Wiener cascade, depicted in Fig. 3.2A. So far we assumed that the internal structure of the second-order system is known. In this example we study the LN

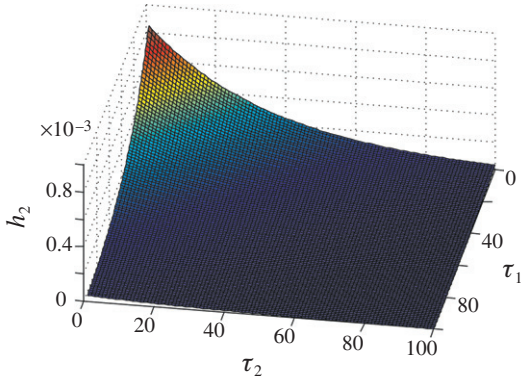

Figure 3.3 Example of a second-order Volterra kernel $h_2(\tau_1, \tau_2)$ determined by Equation (3.13) in MATLAB script Pr3_1.m.

cascade system in Fig. 3.2A but we pretend to know only that it is a second-order Volterra system and that we do **not** know that it is a cascade or what its components are. Now we can use Equation (3.12) and the procedure described in Steps 1−6 below to find the second-order kernel h_2. Finally we will compare this result with the one we can obtain analytically (Equation (3.13)), which is shown in Fig. 3.3.

(1) Using the approach in the example in Section 3.2.1, we use a pair of unit impulses occurring at times T_1 ($\delta(t - T_1)$) and T_2 ($\delta(t - T_2)$) as inputs $x_1(t)$ and $x_2(t)$, respectively.

(2) We determine the system's response to each of the inputs individually: the response to $x_1(t) = \delta(t - T_1)$ is $H_2[\delta(t - T_1)]$ and the response to $x_2(t) = \delta(t - T_2)$ is $H_2[\delta(t - T_2)]$) (Fig. 3.2B).

(3) We determine the response to the sum of both impulses $x_1(t) + x_2(t) = \delta(t - T_1) + \delta(t - T_2)$, which is $y(t) = H_2[\delta(t - T_1) + \delta(t - T_2)]$ and according to Equation (3.12):

$$y(t) = H_2[\delta(t - T_1)] + H_2[\delta(t - T_2)] + 2H_2[\delta(t - T_1), \delta(t - T_2)].$$

(4) From the responses obtained in Steps 2 and 3 above, we can solve for H_2:

$$H_2[\delta(t - T_1), \delta(t - T_2)] = \frac{y(t) - H_2[\delta(t - T_1)] - H_2[\delta(t - T_2)]}{2} \tag{3.14}$$

(5) We use Equation (3.11f) (divided by two) and substitute $\delta(t - T_1)$ for $x_1(t)$ and $\delta(t - T_2)$ for $x_2(t)$:

$$H_2[\delta(t - T_1), \delta(t - T_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2)\delta(t - T_1 - \tau_1)\delta(t - T_2 - \tau_2)\mathrm{d}\tau_1\, \mathrm{d}\tau_2$$

Using the sifting property twice, the double integral evaluates to:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2)\delta(t - T_1 - \tau_1)\delta(t - T_2 - \tau_2)\mathrm{d}\tau_1\, \mathrm{d}\tau_2 = h_2(t - T_1, t - T_2) \tag{3.15}$$

This is the second-order Volterra kernel we are looking for.

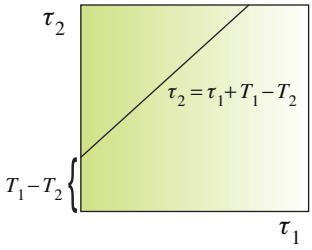

Figure 3.4 The $\tau_1 - \tau_2$ plane and the section represented by $\tau_2 = \tau_1 + T_1 - T_2$.

(6) To relate Equations (3.14) and (3.15) to the definition of the second-order kernel $h_2(\tau_1, \tau_2)$, we set $\tau_1 = t - T_1$ and $\tau_2 = t - T_2$. By using the common variable t, we can relate the delays by: $\tau_1 + T_1 = \tau_2 + T_2 \rightarrow \tau_2 = \tau_1 + T_1 - T_2$. In the $\tau_1 - \tau_2$ plane, this represents a line at 45° with an intercept at $T_1 - T_2$.

Therefore, the response obtained in Equation (3.15) is a slice of the second-order kernel along the line $\tau_2 = \tau_1 + T_1 - T_2$.

Following this procedure, we can obtain the second-order kernel by repeatedly probing the system with pairs of unit impulses at different times T_1 and T_2. By varying the timing of the impulses, we can determine h_2 in both dimensions τ_1 and τ_2, that is, we fill in the plane in Fig. 3.4.

3.3.1 Discrete Time Implementation

Now we are ready to implement the procedure described in Steps 1−6 above for the cascade in Fig. 3.2A. A diagram of this procedure is shown in Fig. 3.2B. Because in this example we know the parameters of the LN cascade, we can compare the result we obtain following Steps 1−6 with the earlier analytically derived result based on our knowledge of the system (depicted in Fig. 3.3).

Recall that for the discrete time solution in the MATLAB file below it is assumed that the sample interval dt is much smaller than the time constant of the filter, that is, $RC/\mathrm{d}t \gg 1$ (see section 11.2.2 in van Drongelen, 2007). If this assumption is violated too much, the approximation of the differential equation by the difference equation will be compromised.

Equation (3.14) can be used to determine the second-order kernel of the system. The following MATLAB code (the second part in Pr3_1.m available on http://www.elsevierdirect.com/companions/9780123849151) will create an image of the 2D kernel shown in Fig. 3.5.

```
i=1; j=0;
delay1=1;

for delay2=delay1:1:length(x);
      j=j+1;
      x1=zeros(1,100);x1(delay1)=1;     % unit impulse train with delay 1
      x2=zeros(1,100);x2(delay2)=1;     % unit impulse train with delay 2

% The summed input xs, containing two unit impulses
```

```
    if (delay1==delay2);
       xs=zeros(1,100);xs(delay1)=2;   % delays are equal
    else
       xs=zeros(1,100);xs(delay1)=1;xs(delay2)=1;
                                       % sum of two unit impulses if delays
                                       % are NOT equal
    end;

% Compute the system outputs to individual and combined unit impulses
    y1_previous=0;
    y2_previous=0;
    ys_previous=0;

    for n=1:length(x);
       % response to delay1
       y1(n)=(A*y1_previous+x1(n))/(A+1);      % the linear component
       y1_previous=y1(n);
       z1(n)=y1(n)^2;                          % the squarer
       % response to delay2
       y2(n)=(A*y2_previous+x2(n))/(A+1);      % the linear component
       y2_previous=y2(n);
       z2(n)=y2(n)^2;                          % the squarer
       % response to the sum of both delays
       ys(n)=(A*ys_previous+xs(n))/(A+1);      % the linear component
       ys_previous=ys(n);
       zs(n)=ys(n)^2;                          % the squarer
       end;

    h=(zs-z1-z2)/2;                  % A slice of the kernel h2
                                     % in the tau1-tau2 plane this is a line
                                     % at 45 degrees with intersection
                                     % delay1-delay2
    tau1=delay2:1:length(x);
    tau2=tau1+(delay1-delay2);
    h=h(delay2:length(h));

    plot3(tau1,tau2,h);
end;

axis([0 T 0 T])
view(100,50)
% Only half is shown because kernel h2 is symmetric
title('half of 2nd order Volterra kernel (h2) of an LN cascade')
xlabel('tau1');ylabel('tau2');zlabel('h2');
grid on
```

3.4 General Second-Order System

The example of the cascade in Fig. 3.2A has a second-order operator only. Generally a second-order system consists of both a first- and second-order operator (assuming again that there is no H_0 component). Following the notation in Equation (3.9a) with $N = 2$ we get:

$$y(t) = H_1[x(t)] + H_2[x(t)] \tag{3.16}$$

An example of such a system where the H_2 operator (representing an LN cascade) is extended with a first-order component H_1 is shown in Fig. 3.2C.

3.4.1 Determining the Second-Order Kernel

For determining h_2 in a system such as that described by Equation (3.16), we can still use the procedure discussed in Steps 1−6 (Section 3.3) and depicted in Fig. 3.2D. The method still works because superposition holds for the contribution of the first-order operator—that is, for input $x(t) = x_1(t) + x_2(t)$, the contribution of the first-order operator is simply the sum of the contributions for $x_1(t)$ and $x_2(t)$ separately:

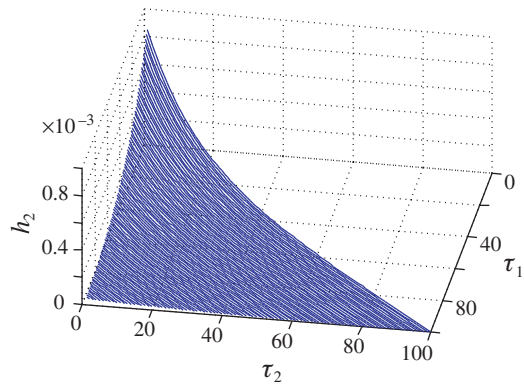

Figure 3.5 By following the procedures in the first and second parts in script Pr3_1.m we can compare the second-order Volterra kernels obtained from the parametric LN cascade model (Fig. 3.3, based on Equation (3.13)) and the one obtained using the nonparametric approach in which we determined h_2 by the procedure outlined in Steps 1−6 (Section 3.3) and represented in Fig. 3.2B. The result of the latter procedure (obtained in the second part of Pr3_1) is shown here. As can be seen by comparing this result with the earlier one in Fig. 3.3, both approaches agree. Because of the symmetry in h_2, only half of the kernel is depicted.

$$\begin{aligned}\int_{-\infty}^{\infty} h_1(\tau)x(t-\tau)\mathrm{d}\tau &= \int_{-\infty}^{\infty} h_1(\tau)[x_1(t-\tau)+x_2(t-\tau)]\mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} h_1(\tau)x_1(t-\tau)\mathrm{d}\tau + \int_{-\infty}^{\infty} h_1(\tau)x_2(t-\tau)\mathrm{d}\tau\end{aligned} \tag{3.17a}$$

or in a more compact notation:

$$H_1[x(t)] = H_1[x_1(t)+x_2(t)] = H_1[x_1(t)] + H_2[x_2(t)] \tag{3.17b}$$

If we apply the same procedure (as shown in Fig. 3.2B) to a system that obeys $y(t) = H_1[x(t)] + H_2[x(t)]$ (e.g., the system in Fig. 3.2C), the contribution of the first-order kernel will cancel because of the superposition property (Equations (3.17a) and (3.17b)). Just as in the previous example, the output will be $2H_2[x_1(t), x_2(t)]$, allowing us to find the second-order kernel by dividing the output of the procedure by 2 (see Equation (3.14)). In program Pr3_2.m (available on http://www.elsevierdirect.com/companions/9780123849151), the procedure depicted in Fig. 3.2D is followed for the second-order system shown in Fig. 3.2C.

3.4.2 Determining the First-Order Kernel

After we determined the system's second-order kernel, we can also find its first-order kernel via the system's UIR. The UIR of the system in Fig. 3.2C will consist of a first- and second-order component (Fig. 3.6). Therefore, if we determine the system's UIR and subtract its second-order component, we have the first-order Volterra kernel h_1. The second-order component of the system's UIR is the slice through h_2 for $\tau_1 = \tau_2$ (i.e., the diagonal of the second-order kernel). This approach is now feasible, since we determined h_2 in the previous procedure. To summarize, we find h_1 by:

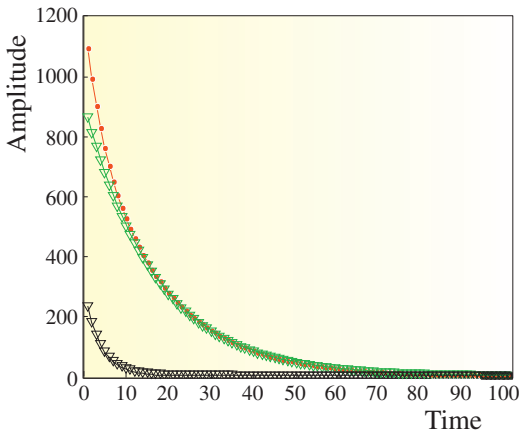

Figure 3.6 An example of a UIR (red dots, upper curve) of a second-order system such as in Fig. 3.2C. The response consists of a first-order component (black triangles, lower curve) and a second-order part (green triangles, middle curve). This result was obtained with MATLAB script Pr3_2.m, albeit with different parameters than the version available on http://www.elsevierdirect.com/companions/9780123849151.

$$h_1 = \text{UIR} - h_2(\tau_1, \tau_2) \quad \text{for } \tau_1 = \tau_2 \tag{3.18}$$

3.5 System Tests for Internal Structure

Nonlinear systems are usually complex, and to facilitate their characterization, one may attempt to simplify their structure by presenting it as a cascade of basic modules. As we discussed in this chapter and in Section 2.5, we often represent dynamical nonlinear systems with cascades of dynamical linear systems and static nonlinearities. In neuroscience, such cascades are frequently used to model neurons and their networks. For example, the integrate-and-fire neuronal model (e.g., Izhikevich, 2007) combines a linear low-pass filter (RC circuit) to mimic subthreshold integration of the biological membrane combined with a static nonlinearity that generates an action potential when the membrane potential exceeds a threshold. Models for neuronal networks also frequently use the cascade approach. For example, in a model to explain the EEG's alpha rhythm, Lopes da Silva et al. (1974) model synaptic function in the thalamo-cortical network with linear filters and a static nonlinearity to model action potential generation (see their fig. 7). Examples of systems that are frequently used to represent nonlinear systems are depicted in Fig. 3.7; in this section we will discuss how these basic configurations may be recognized by examination of their Volterra kernels.

3.5.1 The LN Cascade

The LN cascade (linear system followed by a nonlinear system, Fig. 3.7A), also called a Wiener system (not to be confused with the Wiener series we will discuss in Chapter 4), was also used in Section 2.5 when we demonstrated that the system's input−output relationship fits the Volterra series representation (Equation (2.24)). This result is repeated here:

$$\begin{aligned} z(t) &= a_0 + \int_{-\infty}^{\infty} a_1 h(\tau) x(t-\tau) \mathrm{d}\tau + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_2 h(\tau_1) h(\tau_2) x(t-\tau_1) x(t-\tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \\ &= h_0 + \int_{-\infty}^{\infty} h_1(\tau) x(t-\tau) \mathrm{d}\tau + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \end{aligned} \tag{3.19}$$

From Equation (3.19) we can see that the second-order Volterra kernel h_2 is related to the first-order kernel h_1. The first-order kernel is proportional with the UIR of the Wiener system's linear component: $h_1(\tau) = a_1 h(\tau)$, while the second-order kernel is given by $h_2(\tau_1, \tau_2) = a_2 h(\tau_1) h(\tau_2)$. If we keep one of the variables τ_1 or τ_2 constant, we obtain a section (slice) through the second-order kernel, which is also proportional with the linear component's UIR h. Let us keep τ_2 constant so that $h(\tau_2)$ is a constant value b; we then obtain the expression for a slice through the second-order kernel parallel to the τ_1 axis: $h_2(\tau_1) = b a_2 h(\tau_1)$. It is straightforward

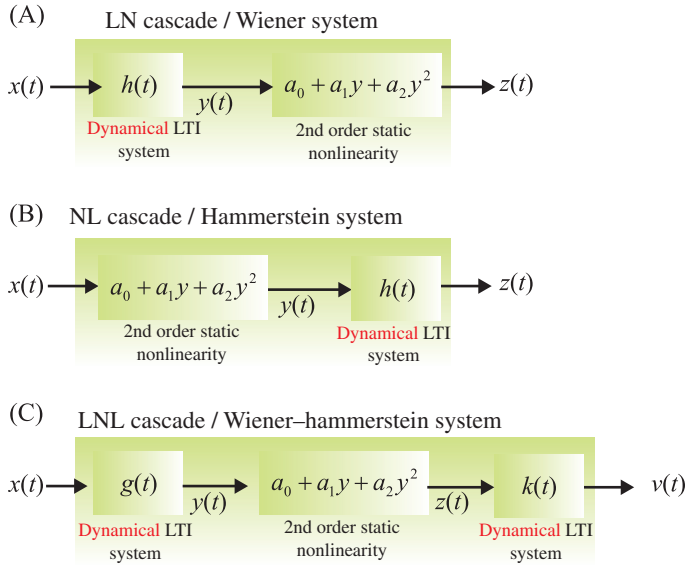

Figure 3.7 Frequently used cascade models to analyze nonlinear dynamical systems. (A) Cascade of a dynamical linear system followed by a static nonlinearity. (B) A similar cascade, but compared with (A) the order has changed: first the static nonlinearity followed by the linear component. (C) A static nonlinearity sandwiched in between two dynamical linear systems.

to show that the ratio between the first-order kernel and a slice (parallel to the τ_1 axis) of the second-order kernel is the constant a_1/ba_2. It is important to note here that this constant may be negative or positive; this should be taken into account when looking for proportionality. A similar result can be obtained for a slice parallel to the τ_2 axis when we hold τ_1 constant. It should be noted that this condition must be satisfied for a Wiener system but there are other configurations that may show the same property. Therefore, strictly speaking, the condition of proportionality of first-order kernels and second-order slices can be used only to exclude the Wiener structure of a nonlinear system. Optimistically, one might say that if the condition is satisfied for a particular nonlinear system, we may use the Wiener structure to model the system.

3.5.2 The NL Cascade

The cascade shown in Fig. 3.7B, also called a Hammerstein system, is a cascade of a nonlinear static component followed by a linear dynamic one. The output y of the first (nonlinear) component becomes the input of the linear dynamical system. The output from this final dynamical component is then the system's output z:

$$z(t) = \int_{-\infty}^{\infty} h(\tau)y(t-\tau)\mathrm{d}\tau = \int_{-\infty}^{\infty} h(\tau)[a_0 + a_1x(t-\tau) + a_2x(t-\tau)^2]\mathrm{d}\tau \tag{3.20}$$

If we separate the three terms in Equation (3.20), we can identify the three Volterra kernels h_0, h_1, and h_2.

$$\text{First term: } \underbrace{\int_{-\infty}^{\infty} h(\tau)a_0\,\mathrm{d}\tau}_{h_0} \tag{3.21a}$$

$$\text{Second term: } \int_{-\infty}^{\infty} \underbrace{h(\tau)a_1}_{h_1(\tau)}\, x(t-\tau)\,\mathrm{d}\tau \tag{3.21b}$$

$$\begin{aligned}\text{Third term: } &\int_{-\infty}^{\infty} h(\tau)a_2x(t-\tau)^2\,\mathrm{d}\tau \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{h(\tau_1)a_2\delta(\tau_1-\tau_2)}_{h_2(\tau_1,\tau_2)}\, x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2\end{aligned} \tag{3.21c}$$

To obtain the Volterra formalism, we rewrote the single integral expression in Equation (3.21c) as a double integral by separating the product $x(t-\tau)^2$ into $x(t-\tau_1)x(t-\tau_2)$. To make sure this product is only nonzero for $\tau_1 = \tau_2$, we added the $\delta(\tau_1 - \tau_2)$ function. The diagonal slice in the $\tau_1 - \tau_2$ plane of the Hammerstein's second-order kernel ($h(\tau)a_2$) is proportional to the UIR of the cascade's linear component ($h(\tau)$). It can be seen in Equation (3.21b) that the first-order Volterra kernel ($h(\tau)a_1$) is also proportional to the linear component's impulse response. Consequently, the diagonal slice of the second-order kernel is proportional with the first-order kernel. Both characteristics discussed above (nonzero second-order kernel along the diagonal and its proportionality with the first-order kernel) may be used to test an unknown nonlinear system for an underlying Hammerstein structure.

3.5.3 The LNL Cascade

A combination of both the cascades discussed above is shown in Fig. 3.7C. Such a system is an LNL cascade, also called a Wiener−Hammerstein model. We obtain the output $z(t)$ of the static nonlinearity inside the cascade by following the same procedure we used to determine the Wiener system's output (see Equations (2.24) and (3.19)):

$$\begin{aligned} z(t) &= a_0 + a_1 y(t) + a_2 y(t)^2 \\ &= a_0 + a_1 \int_{-\infty}^{\infty} g(\tau)x(t-\tau)\mathrm{d}\tau + a_2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tau_1)g(\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \end{aligned} \tag{3.22}$$

The LNL cascade's final output v is then the convolution of the expression above with the UIR k of the second linear system $v(t) = \int_{-\infty}^{\infty} k(\lambda)z(t-\lambda)\mathrm{d}\lambda$ (we use λ here for the delay). Using Equation (3.22) for $z(t-\lambda)$ gives:

$$\begin{aligned} v(t) = a_0 &\int_{-\infty}^{\infty} k(\lambda)\mathrm{d}\lambda + a_1 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(\lambda)g(\tau)x(t-\tau-\lambda)\mathrm{d}\tau\,\mathrm{d}\lambda\ldots \\ &+ a_2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(\lambda)g(\tau_1)g(\tau_2)x(t-\tau_1-\lambda)x(t-\tau_2-\lambda)\mathrm{d}\tau_1\,\mathrm{d}\tau_2\,\mathrm{d}\lambda \end{aligned} \tag{3.23}$$

To simplify, the first-order part of this expression can be rewritten using $\omega = \lambda + \tau$:

$$a_1 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(\lambda)g(\omega-\lambda)x(t-\omega)\mathrm{d}\omega\,\mathrm{d}\lambda = \int_{-\infty}^{\infty} \underbrace{\left[a_1 \int_{-\infty}^{\infty} k(\lambda)g(\omega-\lambda)\mathrm{d}\lambda\right]}_{h_1(\omega)} x(t-\omega)\mathrm{d}\omega \tag{3.24a}$$

Similarly, using $\upsilon = \lambda + \tau_1$ and $\omega = \lambda + \tau_2$, the second-order part becomes:

$$\begin{aligned} &a_2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(\lambda)g(\upsilon-\lambda)g(\omega-\lambda)x(t-\upsilon)x(t-\omega)\mathrm{d}\upsilon\,\mathrm{d}\omega\,\mathrm{d}\lambda \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{\left[a_2 \int_{-\infty}^{\infty} k(\lambda)g(\upsilon-\lambda)g(\omega-\lambda)\mathrm{d}\lambda\right]}_{h_2(\upsilon,\omega)} x(t-\upsilon)x(t-\omega)\mathrm{d}\upsilon\,\mathrm{d}\omega \end{aligned} \tag{3.24b}$$

We can see that the second-order kernel is the integral expression in between the brackets: $a_2 \int_{-\infty}^{\infty} k(\lambda)g(\upsilon-\lambda)g(\omega-\lambda)\mathrm{d}\lambda$. From this expression we can obtain the so-called second-order marginal kernel K_2^{m} (the sum of all kernel slices over one of the variables). If we integrate this expression with respect to one of its variables, say υ, we get:

$$K_2^m = a_2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(\lambda) g(\upsilon - \lambda) g(\omega - \lambda) \mathrm{d}\lambda \, \mathrm{d}\upsilon \tag{3.25a}$$

Now we make a change of timing variables again, $\xi = \upsilon - \lambda$, $\mathrm{d}\xi = \mathrm{d}\upsilon$, and rearrange the integral operation:

$$a_2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(\lambda) g(\upsilon - \lambda) g(\omega - \lambda) \mathrm{d}\lambda \, \mathrm{d}\upsilon = a_2 \int_{-\infty}^{\infty} k(\lambda) \underbrace{\left[\int_{-\infty}^{\infty} g(\xi) \mathrm{d}\xi \right]}_{A} g(\omega - \lambda) \mathrm{d}\lambda$$

$$= a_2 \underbrace{A}_{\mathrm{I}} \underbrace{\left[\int_{-\infty}^{\infty} k(\lambda) g(\omega - \lambda) \mathrm{d}\lambda \right]}_{\mathrm{II}} \tag{3.25b}$$

In the first step, we regrouped the integral operations and defined the outcome of the integral with respect to $\mathrm{d}\xi$ as a number A. Subsequently we separated the expression into two parts. Part I is equal to A and Part II is proportional with the expression for the first-order kernel; this relationship can be seen by comparing Part II with the expression for h_1 in Equation (3.24a): $a_1 \int_{-\infty}^{\infty} k(\lambda) g(\omega - \lambda) \mathrm{d}\lambda$. This latter term is simply Part II scaled by a_1. Of course, we would have obtained a similar outcome had we integrated the second-order kernel with respect to ω. This reasoning leads us to conclude that in an LNL sandwich, the marginal kernel K_2^{m} (the summation [integral] of all slices of the second-order kernel h_2 parallel to one of the axes) is proportional with the first-order kernel h_1. We can use the above proportionality between the marginal second-order kernel and the first-order one to test for a potential underlying sandwich structure of an unknown system. Because other types of cascade may show a similar property, this will allow us to exclude an LNL sandwich structure or to make it likely that we are dealing with one.

3.6 Sinusoidal Signals

When we use a sinusoidal signal as the input to a linear system, we get a sinusoidal signal at its output. At the output, the amplitude of the sinusoidal signal may be amplified/attenuated and the waveform may have a changed phase, but the frequencies of the input and output of an LTI system are identical. We can use this property to completely characterize an LTI system, such as the RC filter, with a set of sinusoidal inputs (see section 10.3 in van Drongelen, 2007). Since the frequency at the output does not change relative to the input frequency, we can describe the LTI system by depicting change of amplitude and phase for each frequency with a Bode plot or an Nyquist plot (see section 12.3, fig. 12.4 in van Drongelen, 2007).

As you may have guessed, this simple relationship between input and output frequency is not valid for nonlinear systems. Let us investigate the response of the second-order nonlinear system introduced in Equation (3.5) by feeding it a cosine

with amplitude A and angular frequency ω_0. Further, let us use Euler's relationship ($e^{\pm j\phi} = \cos\phi \pm j \sin\phi$) to express the input in terms of two complex exponentials:

$$x(t) = A\cos\omega_0 t = \underbrace{\frac{A}{2}e^{j\omega_0 t}}_{x_1(t)} + \underbrace{\frac{A}{2}e^{-j\omega_0 t}}_{x_2(t)} \tag{3.26}$$

Note that the two components of input x (x_1 and x_2) are complex conjugates. Now we can treat this input as we did in Section 3.2.1 and repeat the result from Equation (3.10) for the system's output y:

$$y(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)[x_1(t-\tau_1)+x_2(t-\tau_1)][x_1(t-\tau_2)+x_2(t-\tau_2)]d\tau_1\, d\tau_2 \tag{3.27}$$

In short notation we can write:

$$y(t) = H_2[x_1(t)] + H_2[x_2(t)] + H_2[x_1(t), x_2(t)] + H_2[x_2(t), x_1(t)] \tag{3.28}$$

The only difference between Equation (3.28) and the Equation (3.12) obtained in Section 3.2.1 is that we did not use the symmetry property $H_2[x_1(t), x_2(t)] = H_2[x_2(t), x_1(t)]$. Let us then evaluate each of the four terms in Equation (3.28). The first term is:

$$\begin{aligned} H_2[x_1(t)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)x_1(t-\tau_1)x_1(t-\tau_2)d\tau_1\, d\tau_2 \\ &= \left(\frac{A}{2}\right)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)e^{j\omega_0(t-\tau_1)}\, e^{j\omega_0(t-\tau_2)}\, d\tau_1\, d\tau_2 \end{aligned} \tag{3.29}$$

Combining both exponential expressions we get:

$$\left(\frac{A}{2}\right)^2 e^{j2\omega_0 t} \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)e^{-j\omega_0\tau_1}\, e^{-j\omega_0\tau_2}\, d\tau_1\, d\tau_2}_{\Psi} = \left(\frac{A}{2}\right)^2 e^{j2\omega_0 t}\, \Psi(-j\omega_0, -j\omega_0) \tag{3.30a}$$

Here we use the variable Ψ to symbolize the double integral, a complex function of ω_0.

Note: Comparing the function Ψ above (symbolizing the double integral) with equation (6.4) in van Drongelen (2007), it can be seen that the expression is

the 2D Fourier transform of the second-order kernel.

Similarly, substituting the exponential expression for x_2, the second term $H_2[x_2(t)]$ in Equation (3.28) becomes:

$$\left(\frac{A}{2}\right)^2 \mathrm{e}^{-\mathrm{j}2\omega_0 t} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)\mathrm{e}^{\mathrm{j}\omega_0\tau_1}\,\mathrm{e}^{\mathrm{j}\omega_0\tau_2}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 = \left(\frac{A}{2}\right)^2 \mathrm{e}^{-\mathrm{j}2\omega_0 t}\,\Psi(\mathrm{j}\omega_0,\mathrm{j}\omega_0) \tag{3.30b}$$

Note that both Equations (3.30a) and (3.30b) include an exponent in which the frequency is doubled ($2\omega_0$ instead of ω_0).

The third term in Equation (3.28) is:

$$\begin{aligned} H_2[x_1(t),x_2(t)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)x_1(t-\tau_1)x_2(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \\ &= \left(\frac{A}{2}\right)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)\mathrm{e}^{\mathrm{j}\omega_0(t-\tau_1)}\,\mathrm{e}^{-\mathrm{j}\omega_0(t-\tau_2)}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \end{aligned} \tag{3.31}$$

Combining the exponentials in the expression above, we get:

$$\left(\frac{A}{2}\right)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)\mathrm{e}^{-\mathrm{j}\omega_0\tau_1}\,\mathrm{e}^{\mathrm{j}\omega_0\tau_2}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 = \left(\frac{A}{2}\right)^2 \Psi(-\mathrm{j}\omega_0,\mathrm{j}\omega_0) \tag{3.32a}$$

Using the same approach the fourth term becomes:

$$\left(\frac{A}{2}\right)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)\mathrm{e}^{\mathrm{j}\omega_0\tau_1}\,\mathrm{e}^{-\mathrm{j}\omega_0\tau_2}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 = \left(\frac{A}{2}\right)^2 \Psi(\mathrm{j}\omega_0,-\mathrm{j}\omega_0) \tag{3.32b}$$

Substituting the results for all four terms obtained in (3.30a), (3.30b), (3.32a), and (3.32b) into Equation (3.28) we now have:

$$\begin{aligned} y(t) \quad &= \left[\left(\frac{A}{2}\right)^2 \mathrm{e}^{\mathrm{j}2\omega_0 t}\,\Psi(-\mathrm{j}\omega_0,\,-\mathrm{j}\omega_0) + \left(\frac{A}{2}\right)^2 \mathrm{e}^{-\mathrm{j}2\omega_0 t}\,\Psi(\mathrm{j}\omega_0,\mathrm{j}\omega_0)\right] \\ &+ \left[\left(\frac{A}{2}\right)^2 \Psi(-\mathrm{j}\omega_0,\mathrm{j}\omega_0) + \left(\frac{A}{2}\right)^2 \Psi(\mathrm{j}\omega_0,-\mathrm{j}\omega_0)\right] \end{aligned} \tag{3.33}$$

It can be seen that the first two terms and the second two terms (grouped by brackets) are the complex conjugates of each other. Therefore, we may conclude that the expression in Equation (3.33) is real since the sum of two complex conjugates is

real (the sum of imaginary numbers $a + jb$ and $a - jb$ is $2a$). Consequently we get the following result:

$$y(t) = 2\left(\frac{A}{2}\right)^2 Re(e^{j2\omega_0 t}\Psi(-j\omega_0, -j\omega_0)) + 2\left(\frac{A}{2}\right)^2 Re(\Psi(-j\omega_0, j\omega_0)) \tag{3.34}$$

in which Re(...) denotes the real component. Using Euler's relationship again, we can see that the output contains a sinusoid:

$$y(t) = 2\left(\frac{A}{2}\right)^2 Re[(\cos 2\omega_0 t + j \sin 2\omega_0 t)\Psi(-j\omega_0, -j\omega_0)] + 2\left(\frac{A}{2}\right)^2 Re[\Psi(-j\omega_0, j\omega_0)] \tag{3.35}$$

The output contains a constant (the second term in Equation (3.35)) and a sinusoid with a frequency $2\omega_0$ (the first term).

The expression in Equation (3.35) is an important result for the analysis of higher-order systems: a certain frequency at the system's input may result in a higher-frequency component at the output. **When we digitize the output of a higher-order system as the result of some input signal, it is important to estimate the highest frequency at the output to avoid aliasing** (Section 2.2.2 in van Drongelen, 2007). With a linear system, this problem does not occur; the highest frequency of the input is the highest frequency possible at the output. But with nonlinear systems, the maximum output frequency may be a multiple (as shown above, in a second-order system it is a factor of two, and in an nth order system it is a factor of n) of the input's highest frequency value. A practical approach here is to first sample the system's output at a much higher sample rate than would be used routinely (one to a few orders of magnitude higher) and then compute a power spectrum to estimate the highest frequency component. The outcome of this preliminary experiment can be used to establish an appropriate sample rate.