5 Poisson–Wiener Series

5.1 Introduction

In the previous chapter we considered systems with continuous input signals. One such continuous input is Gaussian white noise (GWN), which allows us to create a series with orthogonal terms that can be estimated sequentially with the Lee–Schetzen cross-correlation method (also shown in the previous chapter). This approach can be adapted when the system's natural input consists of impulse trains such as a spike train. Identifying a system with an impulse train as input will be the topic of this chapter. We will elaborate on the approach that was described by Krausz (1975) and briefly summarized in Marmarelis (2004).[1] Our task at hand is to develop a Wiener series-like approach that describes the input–output relationship of a nonlinear system when an impulse train is at its input. To create randomness at the input, we use an impulse sequence that follows a Poisson process (see Section 14.2 in van Drongelen, 2007).

5.2 Systems with Impulse Train Input

The approach is to create a set of operators that are orthogonal to all lower-order Volterra operators, which is analogous to the development of the Wiener series with a GWN input. We will call these operators "Poisson–Wiener operators" to distinguish our current development of operators (using impulses as input) from that of Chapter 4 (using GWN as input). For each order n, we will symbolize these Poisson–Wiener operators as P_n. Similar to the Wiener series, we define the output z of a nonlinear system as the sum of a set of these operators, each depending on kernel p_n and impulse train input x. For a system of order N we have:

$$z(t) = \sum_{n=0}^{N} P_n[p_n; x(t)]$$

This equation for the Poisson–Wiener series is similar to the ones for the Volterra and Wiener series, but as we will see there are important differences.

As we described in the previous chapter, the approach of the Wiener series works so well because of the specific characteristics of the GWN input signal: $\langle x(t-\tau_1)\rangle = 0$, $\langle x(t-\tau_1)x(t-\tau_2)\rangle = \sigma^2\delta(\tau_2-\tau_1)$, etc. (see Appendix 4.1). When

[1] If you compare the following with Krausz' original work, please note that the derivation in Krausz (1975) contains minor scaling errors (as was also noted by Marmarelis, 2004).

Signal Processing for Neuroscientists, A Companion Volume. DOI: 10.1016/B978-0-12-384915-1.00005-X

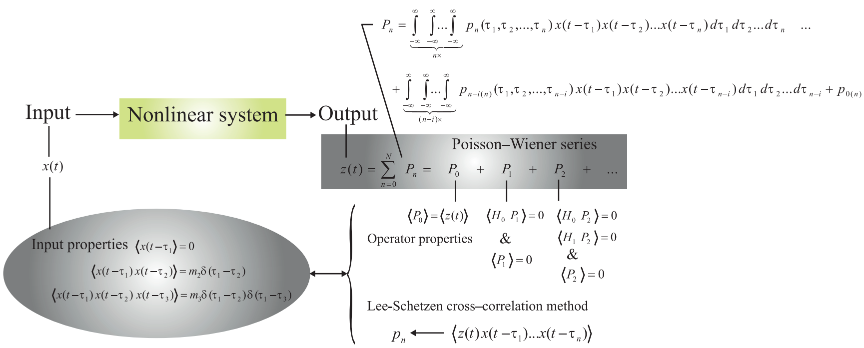

Figure 5.1 Diagram of the procedures used here to develop the Poisson–Wiener series, the properties of its operators, and the method to determine the kernels. Just as for the Wiener series, the input signal's properties play a crucial role in the development of the Poisson–Wiener approach.

the system's input changes to a series of impulses, these relationships no longer hold and we can no longer apply the equations we derived previously. To resolve this, we must start from scratch and first determine expressions for the averaged products $\langle x(t-\tau_1)\rangle$, $\langle x(t-\tau_1)x(t-\tau_2)\rangle$, $\ldots$ for the Poisson process. Subsequently we must use these new results to redo the Gram–Schmidt procedure for the derivation of our series' orthogonal terms, as was done in Section 4.2. Finally we must redevelop Lee–Schetzen's cross-correlation method in a similar fashion as the procedure described in Section 4.3.

A schematic overview of the procedures we develop in this chapter is depicted in Fig. 5.1. Similar to the properties of Wiener series, the output z of a nonlinear system can be described by a (Poisson–Wiener) series in which:

(1) Operators P_n are heterogeneous (top-right in Fig. 5.1)
(2) Each operator is orthogonal to all lower-order Volterra operators
(3) Except for P_0, the Expectation (or time average) of all operators will vanish
(4) Except for p_0, the kernels can be determined from the cross-correlation of input and output (see also the Lee–Schetzen method introduced in Chapter 4).

In each of the above properties, it is important to know the Expectation or time average for the input and its cross products (see Input properties in Fig. 5.1). Therefore we will first determine these time averages associated with the input in Section 5.2.1 before we derive the Poisson–Wiener kernels in Section 5.2.2 and adapt Lee–Schetzen's cross-correlation method for determining the kernels from recorded data in Section 5.3.

5.2.1 Product Averages for the Poisson Impulse Train

Let us use signal χ, a train of Diracs with amplitude A that follows a Poisson process with rate ρ (Fig. 5.2A). The **first moment** or mean μ of impulse train χ can be established by a time average over a sufficiently long interval T. In such

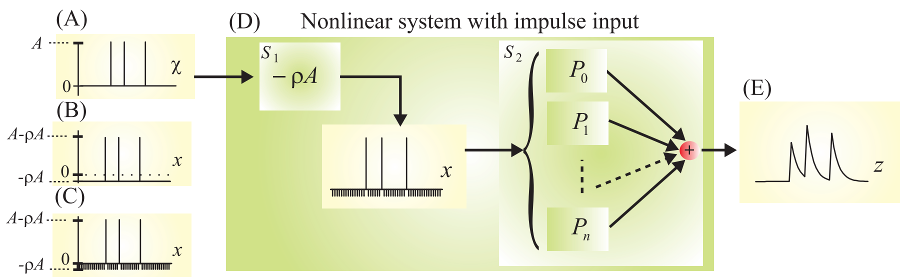

Figure 5.2 Impulse train inputs following a Poisson process can be used to identify nonlinear systems. A standard impulse train χ with amplitude A is shown in (A). A demeaned version of this time series x is depicted in (B). The signal in (C) is the same demeaned series x but is now presented as a series of weighted unit impulses (each impulse is represented by a vertical line). The procedure depicted in (D) shows the steps we use to identify a nonlinear system with such a train of impulses. First we pretend that the input is demeaned by part of the system (subsystem S_1) by subtracting ρA, the mean of χ. This demeaned series x is then used as input to subsystem S_2. We actually determine the operators P_n and kernels for S_2 instead of the whole system $S_1 + S_2$, but if we can characterize S_2 we have characterized the whole system, since S_1 is a simple subtraction. (E) depicts the output z of the system to the impulse input.

an interval we expect to find $N = \rho T$ impulses in the input signal $\chi = \sum_{i=1}^{N=\rho T} A\delta(t - t_i)$. The time average of the input signal is $\langle \chi \rangle = (1/T) \int_0^T \sum_{i=1}^{N=\rho T} A\delta(t - t_i)\mathrm{d}t$. Assuming we can interchange summation and integration we get:

$$\mu = \langle \chi \rangle = \frac{A}{T} \sum_{i=1}^{N=\rho T} \underbrace{\int_0^T \delta(t - t_i)\mathrm{d}t}_{=1} = \frac{A}{T}\rho T = \rho A \tag{5.1a}$$

The integral in Equation (5.1a) evaluates to one if the delta function is located within the interval T (i.e., $0 \le t_i \le T$). We could have done the computation of the mean in a simpler way because we know how many impulses we expect in epoch T and the amplitude of each impulse. The number of impulses (each with amplitude A) during this interval is ρT, resulting in the following expression for the first moment:

$$\mu = \langle \chi \rangle = \frac{1}{T} \int_0^T \underbrace{\rho T}_{\mathrm{I}} \underbrace{A\delta(t)}_{\mathrm{II}} \mathrm{d}t = \frac{1}{T}\rho T A = \rho A \tag{5.1b}$$

Part I in the equation above is the number of expected impulses over interval T and Part II is the amplitude for each impulse. Unlike the first moment for the GWN signal we used in the previous chapter, this result is **not** zero. The following step is therefore critical for the rest of our approach: **because the nonzero result for the first moment would complicate matters, we create a new signal x, which is the demeaned version of χ** (Fig. 5.2B):

$$\boxed{x(t) = \chi(t) - \rho A} \tag{5.1c}$$

We can check that this generates a zero first moment for time series $x(t)$:

$$\begin{aligned}\langle x\rangle &= \frac{1}{T}\int_0^T \left[\sum_{i=1}^{N=\rho T} A\delta(t-t_i) - \rho A\right]\mathrm{d}t = \frac{1}{T}\left(\sum_{i=1}^{N=\rho T}\underbrace{\int_0^T A\delta(t-t_i)\mathrm{d}t}_{A} - \int_0^T \rho A\,\mathrm{d}t\right)\\ &= \frac{1}{T}\left(\rho AT - [\rho At]_0^T\right) = \frac{1}{T}(\rho AT - \rho AT) = 0\end{aligned} \tag{5.1d}$$

Here we interchanged the integration and summation operations. Subsequently, we evaluate the integral with the delta function and find that it is equal to the constant A if the delta function falls within epoch T. Alternatively, we can also approach the estimation of $\langle x\rangle$ a bit differently. As you can see in Fig. 5.2C, we can consider the demeaned signal as a series of Diracs (a sampled version of the signal) with amplitude $A - A\rho$ for each spike, and amplitude $-A\rho$ in between the spikes. Over interval T the number of spike occurrences is again ρT and the number of nonspike occurrences is $(1-\rho)T$.

$$\begin{aligned}\langle x\rangle &= \frac{1}{T}\int_0^T [\underbrace{\rho T}_{\mathrm{I}}\ \underbrace{(A - A\rho)\delta(t)}_{\mathrm{II}} + \underbrace{(1-\rho)T}_{\mathrm{III}}\ \underbrace{(-A\rho)\delta(t)}_{\mathrm{IV}}]\mathrm{d}t\\ &= \frac{1}{T}\int_0^T \underbrace{[\rho AT - \rho^2 AT - \rho AT + \rho^2 AT]}_{0}\ \delta(t)\mathrm{d}t = 0\end{aligned} \tag{5.1e}$$

Parts I and III above are the expected number of spiking and nonspiking events, respectively, and Parts II and IV are their respective amplitudes. We will use the approach in Equation (5.1e) to compute the higher-order products in the following. The bottom line is that by using the impulse series x as input, we have (just as for GWN) zero for the first moment m_1:

$$\boxed{m_1 = \langle x\rangle = 0} \tag{5.1f}$$

The next expression we must evaluate is the cross-correlation $\langle x(t-\tau_1)x(t-\tau_2)\rangle$. To start, we can look into the *second moment* $\langle x^2\rangle$ of the impulse train in Fig. 5.2C. As shown above in Equation (5.1e), the number of events N is the event probability ρ times the interval T, and the nonevent probability equals $(1-\rho)T$ (Parts I and III, respectively). For the second-order moment, we will square the associated amplitudes (Parts II and IV):

$$\langle x^2\rangle = \frac{1}{T}\int_0^T [\underbrace{\rho T}_{\mathrm{I}}\ \underbrace{(A-A\rho)^2\delta(t)}_{\mathrm{II}} + \underbrace{(1-\rho)T}_{\mathrm{III}}\ \underbrace{(-A\rho)^2\delta(t)}_{\mathrm{IV}}]\mathrm{d}t \tag{5.2a}$$

Note that by squaring the amplitudes we weight the unit impulse function $\delta(t)$, but we do not need to square the delta function itself. It is relatively simple to see why this is not required. Imagine the input as the series of Dirac deltas weighted with different amplitudes shown in Fig. 5.2C. The sum of all amplitudes x divided by the epoch length T is the first moment, the sum of all x^2 divided by T is the second moment, the sum of all x^3 divided by T is the third moment, and so on (see Section 3.2 in van Drongelen, 2007). To sample the amplitudes of x, x^2, x^3, ... we only have to weight a single Dirac with the desired amplitude (if you need to review the properties of the Dirac δ, see section 2.2.2 in van Drongelen, 2007). Simplifying Equation (5.2a), we get:

$$= \frac{1}{T}\int_0^T \underbrace{[\rho A^2T - 2\rho^2A^2T + \rho^3A^2T + \rho^2A^2T - \rho^3A^2T]}_{\rho A^2T-\rho^2A^2T}\delta(t)\mathrm{d}t$$

$$= \frac{1}{T}\int_0^T T\rho A^2(1-\rho)\delta(t)\mathrm{d}t = \frac{1}{T}\left[T\rho A^2(1-\rho)\right]$$

Finally, the expression for the second moment m_2 becomes:

$$m_2 = \langle x^2\rangle = \rho A^2(1-\rho) \tag{5.2b}$$

The next step is to determine the second-order cross-correlation using a time average of the product $x(t-\tau_1)x(t-\tau_2)$:

$$\begin{aligned}\langle x(t-\tau_1)x(t-\tau_2)\rangle = \frac{1}{T}\int_0^T &[\underbrace{\rho T}_{\mathrm{I}}\ \underbrace{(A-A\rho)^2\delta(t-\tau_1)\delta(t-\tau_2)}_{\mathrm{II}} \\ &+ \underbrace{(1-\rho)T}_{\mathrm{III}}\ \underbrace{(-A\rho)^2\delta(t-\tau_1)\delta(t-\tau_2)}_{\mathrm{IV}}]\mathrm{d}t\end{aligned} \tag{5.3a}$$

Parts I–IV are similar to the ones in Equation (5.2a). The product of I and II, the first term in the integral in Equation (5.3a), evaluates to:

$$\frac{1}{T}\left[\rho T(A - A\rho)^2\right]\delta(\tau_1 - \tau_2)$$

and the product of III and IV, the second term in Equation (5.3a), becomes:

$$\frac{1}{T}\left[(1-\rho)TA^2\rho^2\right]\delta(\tau_1 - \tau_2)$$

Combining the two terms above, we get the result for the *second-order cross-correlation*:

$$\boxed{\langle x(t-\tau_1)x(t-\tau_2)\rangle = \rho A^2(1-\rho)\delta(\tau_1-\tau_2) = m_2\delta(\tau_1-\tau_2)} \tag{5.3b}$$

This result is not unexpected since Equation (5.3b) becomes the expression we derived for the second moment m_2 (Equation (5.2b)) when we have the case $\tau_1 = \tau_2$. Just as was the case for GWN, this expression will evaluate to zero otherwise.

For computing the *third moment* m_3, we can use the same approach as in Equation (5.2a):

$$\langle x^3\rangle = \frac{1}{T}\int_0^T [\underbrace{\rho T}_{\mathrm{I}}\,\underbrace{(A - A\rho)^3\delta(t)}_{\mathrm{II}} + \underbrace{(1-\rho)T}_{\mathrm{III}}\,\underbrace{(-A\rho)^3\delta(t)}_{\mathrm{IV}}]\mathrm{d}t \tag{5.4a}$$

If you do the algebra, you will find that this results in:

$$m_3 = \langle x^3\rangle = \rho A^3(1-\rho)(1-2\rho) \tag{5.4b}$$

The third-order cross-correlation is:

$$\begin{aligned}&\langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\rangle\\ &\quad= \frac{1}{T}\int_0^T [\underbrace{\rho T}_{\mathrm{I}}\,\underbrace{(A-A\rho)^3\delta(t-\tau_1)\delta(t-\tau_2)\delta(t-\tau_3)}_{\mathrm{II}}\\ &\qquad+ \underbrace{(1-\rho)T}_{\mathrm{III}}\,\underbrace{(-A\rho)^3\delta(t-\tau_1)\delta(t-\tau_2)\delta(t-\tau_3)}_{\mathrm{IV}}]\mathrm{d}t\end{aligned} \tag{5.5a}$$

in which Parts I−IV can be evaluated similarly to the ones in Equation (5.3a). Accordingly, the result becomes:

$$\langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\rangle = \rho A^3(1-\rho)(1-2\rho)\delta(\tau_1-\tau_2)\delta(\tau_1-\tau_3)$$

Combined with Equation (5.4b), we get:

$$\boxed{\langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\rangle = m_3\delta(\tau_1-\tau_2)\delta(\tau_1-\tau_3)} \tag{5.5b}$$

for the *third-order cross-correlation*. As you can see, due to the presence of two Diracs, the third-order product is only nonzero for $\tau_1 = \tau_2 = \tau_3$.

In the above cases, things are relatively simple because we set the first moment to zero by demeaning the input impulse train. This approach ensures that any product that contains $E\{x\}$ or $\langle x\rangle$ (the Expectation or time average of x) vanishes (see Appendix 5.1). Appendix 5.1 explains that for the *fourth moment* m_4, we have to deal with additional terms that include $E\{x^2\}$. If you are mainly interested in how we will next make Poisson−Wiener operators orthogonal, you can accept the results for m_4 and the fourth-order product below and skip the appendix. The expression for m_4 is obtained in the same manner as the lower-order moments above.

$$m_4 = \langle x^4\rangle = \rho A^4[\rho(1-\rho)^2 + (1-\rho)(1-2\rho)^2] \tag{5.6a}$$

The time averaged *fourth-order cross-correlation* critically depends on the values of the delays $\tau_1 - \tau_4$ in a piece-wise manner (see Appendix 5.1, Equation (A5.1.5)):

$$\boxed{\begin{array}{l}\langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x(t-\tau_4)\rangle \\ \begin{cases} \tau_1=\tau_2=\tau_3=\tau_4 & : m_4\delta(\tau_1-\tau_2)\delta(\tau_1-\tau_3)\delta(\tau_1-\tau_4) \\ \tau_1=\tau_2 \text{ and } \tau_3=\tau_4 & : m_2^2\delta(\tau_1-\tau_2)\delta(\tau_3-\tau_4) \\ \tau_1=\tau_3 \text{ and } \tau_2=\tau_4 & : m_2^2\delta(\tau_1-\tau_3)\delta(\tau_2-\tau_4) \\ \tau_1=\tau_4 \text{ and } \tau_2=\tau_3 & : m_2^2\delta(\tau_1-\tau_4)\delta(\tau_2-\tau_3) \\ 0 & \text{otherwise} \end{cases}\end{array}} \tag{5.6b}$$

5.2.2 Orthogonal Terms of the Poisson−Wiener Series

In this section we use the same procedure (Gram−Schmidt orthogonalization, see Arfken and Weber, 2005) as in Chapter 4 to derive the orthogonal series that can characterize a nonlinear system given our impulse input. As depicted in Fig. 5.2D, the Poisson−Wiener series represents an output signal z consisting of the sum of operators P_n:

$$z(t) = P_0[p_0; x(t)] + P_1[p_1; x(t)] + P_2[p_2; x(t)] + \cdots + P_n[p_n; x(t)] \tag{5.7a}$$

in which the heterogeneous operator P_n is defined as:

$$
\begin{aligned}
&P_n[p_n;x(t)] \\
&= \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}}_{n\times} p_n(\tau_1,\tau_2,\ldots,\tau_n)x(t-\tau_1)x(t-\tau_2)\ldots x(t-\tau_n)\mathrm{d}\tau_1\,\mathrm{d}\tau_2\ldots\mathrm{d}\tau_n \\
&+ \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}}_{(n-1)\times} p_{n-1(n)}(\tau_1,\tau_2,\ldots,\tau_{n-1})x(t-\tau_1)x(t-\tau_2)\ldots x(t-\tau_{n-1})\mathrm{d}\tau_1\,\mathrm{d}\tau_2\ldots\mathrm{d}\tau_{n-1} \\
&+ \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}}_{(n-i)\times} p_{n-i(n)}(\tau_1,\tau_2,\ldots,\tau_{n-i})x(t-\tau_1)x(t-\tau_2)\ldots x(t-\tau_{n-i})\mathrm{d}\tau_1\,\mathrm{d}\tau_2\ldots\mathrm{d}\tau_{n-i}+p_{0(n)}
\end{aligned}
\tag{5.7b}
$$

Here we have Poisson–Wiener kernel p_n and derived Poisson–Wiener kernels $p_{n-i(n)}$ ($i = 1, 2, \ldots, n$). In Sections 5.2.2.1–5.2.2.3, we will derive the expressions for the Poisson–Wiener operators in a similar fashion we did for the Wiener series in Chapter 4.

5.2.2.1 The Zero-Order Poisson–Wiener Operator

Similar to the zero-order Wiener operator, we define the zero-order Poisson–Wiener operator P_0 as the output's DC component p_0:

$$\boxed{P_0[p_0;x(t)] = p_0} \tag{5.8}$$

In this equation, we use p_0 to symbolize the zero-order Poisson–Wiener kernel to distinguish it from the zero-order Volterra and Wiener kernels h_0 and k_0, respectively.

5.2.2.2 The First-Order Poisson–Wiener Operator

Now we use the orthogonality between Poisson–Wiener operators and lower-order Volterra operators to derive the expression for the first-order Poisson–Wiener kernel p_1. Similar to Equation (4.5), we have:

$$\langle H_0[x(t)]P_1[p_1;x(t)]\rangle = \left\langle h_0 \underbrace{\left[\int_{-\infty}^{\infty} p_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1 + p_{0(1)}\right]}_{P_1} \right\rangle = 0$$
$$= h_0 \underbrace{\left[\int_{-\infty}^{\infty} p_1(\tau_1)\big\langle x(t-\tau_1)\big\rangle \mathrm{d}\tau_1 + p_{0(1)}\right]}_{\langle P_1\rangle} = 0 \tag{5.9}$$

The subscript 0(1) indicates that $p_{0(1)}$ is a derived kernel: a zero-order member of the first-order operator P_1. Note that we took all constants out of the time average operation, and only the (time dependent) input time series x remains within the time average brackets $\langle ... \rangle$. Since input x is a demeaned impulse train following a Poisson process, we know that $\langle x(t-\tau_1)\rangle = 0$ (see Equation (5.1f)). Consequently the integral evaluates to zero, and we therefore conclude that the orthogonality requirement demands that:

$$\boxed{p_{0(1)} = 0} \tag{5.10}$$

Substituting this result in the general expression for our first-order Poisson—Wiener operator $P_1[p_1;x(t)] = \int_{-\infty}^{\infty} p_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1 + p_{0(1)}$, we obtain:

$$\boxed{P_1[p_1;x(t)] = \int_{-\infty}^{\infty} p_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1} \tag{5.11}$$

Note that this result is very similar to the first-order Wiener operator (Equation (4.7)). Furthermore, we see that $E\{P\} = \langle P_1\rangle = 0$: that is, the Expectation or time average of P_1, $\langle \int_{-\infty}^{\infty} p_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1\rangle$, evaluates to zero because $\langle x(t-\tau_1)\rangle = 0$.

You can also see in Fig. 5.2D that this kernel is not the first-order kernel for our system but for the subsystem indicated by S_2 (the whole system is $S_1 + S_2$). Because we know that the other part, subsystem S_1, is a simple subtraction ($-\rho A$), we have effectively characterized the first-order component of the system under investigation.

5.2.2.3 The Second-Order Poisson—Wiener Operator

To establish the expression for the second-order operator we follow the same procedure as for the Wiener kernels: we demand both orthogonality between the second-order Poisson—Wiener operator and a zero-order Volterra operator plus orthogonality between the second-order operator and a first-order Volterra operator.

First, for orthogonality between H_0 and P_2: using the orthogonality condition we get:

$$\langle H_0[x(t)]P_2[p_2; x(t)]\rangle = 0$$

That is:

$$= \left\langle \underbrace{h_0}_{H_0} \underbrace{\left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2 + \int_{-\infty}^{\infty} p_{1(2)}(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1 + p_{0(2)} \right]}_{P_2} \right\rangle = 0$$

$$= h_0 \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)\langle x(t-\tau_1)x(t-\tau_2)\rangle \mathrm{d}\tau_1\mathrm{d}\tau_2 \right.$$

$$\left. + \int_{-\infty}^{\infty} p_{1(2)}(\tau_1)\langle x(t-\tau_1)\rangle \mathrm{d}\tau_1 + p_{0(2)} \right] = 0 \tag{5.12}$$

Similar to the composition of the Wiener operator G_2, the components $p_{0(2)}$ and $p_{1(2)}$ are derived zero-order and first-order members of operator P_2. As we did in Equation (5.9), we took all constants out of the time average $\langle ... \rangle$ and only kept the time series x within it. Again, because the input is a zero mean impulse train following a Poisson process, the term with the single integral in the expression above is zero (since $\langle x(t-\tau_1)\rangle = 0$, Equation (5.1f)). The term with the double integral is dictated by the averaged product of both inputs $\langle x(t-\tau_1)x(t-\tau_2)\rangle$, which is given by Equation (5.3b). Therefore the above expression becomes:

$$h_0 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_2(\tau_1, \tau_2)\big\langle x(t-\tau_1)x(t-\tau_2)\big\rangle \mathrm{d}\tau_1\, \mathrm{d}\tau_2 + h_0 p_{0(2)}$$
$$= m_2 h_0 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_2(\tau_1, \tau_2)\delta(\tau_1 - \tau_2)\mathrm{d}\tau_1\, \mathrm{d}\tau_2 + h_0 p_{0(2)} = 0$$

This equation can be evaluated by using the sifting property for one of the time constants; here we integrate with respect to τ_2 and get:

$$m_2 h_0 \int_{-\infty}^{\infty} p_2(\tau_1, \tau_1)\mathrm{d}\tau_1 + h_0 p_{0(2)} = 0 \rightarrow \boxed{p_{0(2)} = -m_2 \int_{-\infty}^{\infty} p_2(\tau_1, \tau_1)\mathrm{d}\tau_1} \tag{5.13}$$

As you can see, the kernel $p_{0(2)}$ is derived from p_2.

To further express our second-order Poisson–Wiener operator, we will next demand orthogonality between second-order operator P_2 and first-order Volterra operator H_1. Similar to Equation (4.10), we have:

$$\langle H_1[x(t)]P_2[p_2; x(t)]\rangle = 0,$$

which can be written as:

$$\left\langle \begin{array}{c} \left[\int_{-\infty}^{\infty} h_1(\upsilon)x(t-\upsilon)\mathrm{d}\upsilon\right] \times \\ \left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2 + \int_{-\infty}^{\infty} p_{1(2)}(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1 + p_{0(2)}\right] \end{array} \right\rangle = 0 \tag{5.14}$$

Equation (5.14) contains three terms that we will consider separately.
The **first** term is:

$$\left\langle \left[\int_{-\infty}^{\infty} h_1(\upsilon)x(t-\upsilon)\mathrm{d}\upsilon\right]\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2\right]\right\rangle$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_1(\upsilon)p_2(\tau_1,\tau_2)\underbrace{\langle x(t-\upsilon)x(t-\tau_1)x(t-\tau_2)\rangle}_{m_3\delta(\upsilon-\tau_1)\delta(\upsilon-\tau_2)}\mathrm{d}\upsilon\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2$$

In the Wiener series development, for systems with GWN input, the odd product $\langle x(t-\upsilon)x(t-\tau_1)x(t-\tau_2)\rangle = 0$. Here, however, the odd product is given by Equation (5.5b). This gives:

$$\begin{aligned} & m_3 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_1(\upsilon)p_2(\tau_1,\tau_2)\delta(\upsilon-\tau_1)\delta(\upsilon-\tau_2)\mathrm{d}\upsilon\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \\ & = m_3 \int_{-\infty}^{\infty} h_1(\upsilon)p_2(\upsilon,\upsilon)\mathrm{d}\upsilon \end{aligned} \tag{5.15a}$$

The **second** term in Equation (5.14) is:

$$\left\langle \left[\int_{-\infty}^{\infty} h_1(\upsilon)x(t-\upsilon)\mathrm{d}\upsilon\right]\left[\int_{-\infty}^{\infty} p_{1(2)}(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1\right]\right\rangle$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_1(\upsilon)p_{1(2)}(\tau_1)\underbrace{\langle x(t-\upsilon)x(t-\tau_1)\rangle}_{m_2\delta(\upsilon-\tau_1)}\mathrm{d}\upsilon\,\mathrm{d}\tau_1$$

Using the expression for the second-order correlation in Equation (5.3b) we can simplify to:

$$m_2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_1(\upsilon) p_{1(2)}(\tau_1) \delta(\upsilon - \tau_1) \mathrm{d}\upsilon \, \mathrm{d}\tau_1 = m_2 \int_{-\infty}^{\infty} h_1(\upsilon) p_{1(2)}(\upsilon) \mathrm{d}\upsilon \tag{5.15b}$$

Note that we used the sifting property of the Dirac to simplify the double integral.

Finally, the **third** term in Equation (5.14) is:

$$\left\langle \left[\int_{-\infty}^{\infty} h_1(\upsilon) x(t-\upsilon) \mathrm{d}\upsilon \right] p_{0(2)} \right\rangle = \left[\int_{-\infty}^{\infty} h_1(\upsilon) \langle x(t-\upsilon) \rangle \mathrm{d}\upsilon \right] p_{0(2)} = 0 \tag{5.15c}$$

This evaluates to zero because $\langle x(t-\upsilon) \rangle = 0$ (Equation (5.1f)).

Substituting the results from Equations (5.15a), (5.15b), and (5.15c) into Equation (5.14), we have:

$$m_3 \int_{-\infty}^{\infty} h_1(\upsilon) p_2(\upsilon, \upsilon) \mathrm{d}\upsilon + m_2 \int_{-\infty}^{\infty} h_1(\upsilon) p_{1(2)}(\upsilon) \mathrm{d}\upsilon = 0$$

From this we may conclude that the derived first-order member of the second-order operator is:

$$\boxed{p_{1(2)}(\upsilon) = -\frac{m_3}{m_2} p_2(\upsilon, \upsilon)} \tag{5.16}$$

Again, you can see that the derived kernel $p_{1(2)}$ is indeed derived because it fully depends on p_2. Using the results in Equations (5.13) and (5.16), we get the expressions for the second-order Poisson–Wiener operator in terms of the second-order Poisson–Wiener kernel p_2:

$$\boxed{\begin{aligned} P_2[p_2; x(t)] = & \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_2(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2) \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \\ & \underbrace{- \frac{m_3}{m_2} \int_{-\infty}^{\infty} p_2(\tau, \tau) x(t-\tau) \mathrm{d}\tau}_{+\int p_{1(2)} x(t-\tau) \mathrm{d}\tau} \underbrace{- m_2 \int_{-\infty}^{\infty} p_2(\tau, \tau) \mathrm{d}\tau}_{+p_{0(2)}} \end{aligned}} \tag{5.17}$$

Note that the above result for the second-order Poisson−Wiener operator differs from the second-order Wiener operator (Equation (4.13)) (here $p_{1(2)}$ is nonzero). This difference is due to the fact that the cross-correlation results for a demeaned train of impulses following a Poisson process are different from a GWN signal (see Section 5.2.1 and compare Appendices 4.1 and 5.1). Using the expressions for $\langle x(t-\tau_1)\rangle$ and $\langle x(t-\tau_1)x(t-\tau_2)\rangle$, it is straightforward to show that $E\{P_2\} = \langle P_2\rangle = 0$.

5.3 Determination of the Zero-, First- and Second-Order Poisson−Wiener Kernels

In this section we will compute the Poisson−Wiener kernels using the same cross-correlation method first described for the Wiener kernels (Lee and Schetzen, 1965). If we deal with a nonlinear system of order N, and we present a demeaned impulse train x following a Poisson process at its input, we obtain output z as the sum of the Poisson−Wiener operators (Fig. 5.2D):

$$z(t) = \sum_{n=0}^{N} P_n[p_n; x(t)] \tag{5.18}$$

In the following example we will describe how to determine the zero-, first- and second-order Poisson−Wiener kernels.

5.3.1 Determination of the Zero-Order Poisson−Wiener Kernel

Similar to the Wiener operators, the Expectation of all Poisson−Wiener operators P_n, except the zero-order operator P_0, is zero. Therefore, assuming an ergodic process (time averages are allowed for estimating the Expectations), we find the average of output signal z:

$$\boxed{\langle z(t)\rangle = \sum_{n=0}^{N} \langle P_n[p_n; x(t)]\rangle = P_0[p_0; x(t)] = p_0} \tag{5.19}$$

Thus the zero-order Poisson−Wiener kernel is equal to the mean output $\langle z(t)\rangle$.

5.3.2 Determination of the First-Order Poisson−Wiener Kernel

Similar to the procedure for the Wiener kernels depicted in Fig. 4.2, the first-order Poisson−Wiener kernel of a system can be obtained from the cross-correlation between its input and output:

$$\begin{aligned}\langle z(t)x(t-\upsilon_1)\rangle &= \langle P_0[p_0;x(t)]x(t-\upsilon_1)\rangle + \langle P_1[p_1;x(t)]x(t-\upsilon_1)\rangle \\ &\quad + \langle P_2[p_2;x(t)]x(t-\upsilon_1)\rangle + \cdots \\ &= \sum_{n=0}^{N}\langle P_n[p_n;x(t)]x(t-\upsilon_1)\rangle\end{aligned} \tag{5.20}$$

Recall that Poisson–Wiener kernels are defined to be orthogonal to lower-order Volterra kernels, and recall that the delay operator $x(t-\upsilon_1)$ can be presented as a first-order Volterra operator (see Appendix 4.2). Therefore, all Poisson–Wiener operators P_n with $n \geq 2$ are orthogonal to $x(t-\upsilon_1)$, and we only have to deal with operators of order $n = 0$ and 1.

For $\boldsymbol{n = 0}$:

$$\langle P_0[p_0;x(t)]x(t-\upsilon_1)\rangle = p_0 \underbrace{\langle x(t-\upsilon_1)\rangle}_{0} = 0 \tag{5.21a}$$

For $\boldsymbol{n = 1}$:

$$\begin{aligned}\langle P_1[p_1;x(t)]x(t-\upsilon_1)\rangle &= \int_{-\infty}^{\infty} p_1(\tau_1)\underbrace{\langle x(t-\tau_1)x(t-\upsilon_1)\rangle}_{m_2\delta(\tau_1-\upsilon_1)}\,d\tau_1 \\ &= m_2\int_{-\infty}^{\infty} p_1(\tau_1)\delta(\tau_1-\upsilon_1)d\tau_1 = m_2 p_1(\upsilon_1)\end{aligned} \tag{5.21b}$$

Here we used Equation (5.3b) to simplify $\langle x(t-\tau_1)x(t-\upsilon_1)\rangle$ and then used the sifting property of the Dirac to evaluate the above integral. From the results in Equation (5.21b), we conclude that the only nonzero part in Equation (5.20) is the term for $n = 1$; therefore, the first-order Poisson–Wiener kernel becomes:

$$\langle z(t)x(t-\upsilon_1)\rangle = m_2 p_1(\upsilon_1) \rightarrow \boxed{p_1(\upsilon_1) = \frac{1}{m_2}\langle z(t)x(t-\upsilon_1)\rangle} \tag{5.22a}$$

Therefore, the first-order Poisson–Wiener kernel is the cross-correlation between input and output weighted by the second moment m_2 of the input.

We can use the properties of the Dirac to rewrite the cross-correlation expression, because the input is an impulse train. If we substitute the expression for the input in Equation (5.22a) with a sum of Diracs and present the time average $\langle ... \rangle$ with an integral notation $(1/T)\int_0^T \cdots$, we get:

$$p_1(\upsilon_1) = \underbrace{\frac{1}{m_2}\frac{1}{T}\int_0^T z(t)\overbrace{\left[A\sum_{i=1}^{N=\rho T}\delta(t-t_i-\upsilon_1)-\rho A\right]}^{\text{input: } x(t-\upsilon_1)}\,dt}_{\text{Time average}}$$

Assuming we may interchange the integration and summation and separating the terms for the impulse train (the Diracs) and the DC correction (ρA), this evaluates into two integral terms:

$$p_1(\upsilon_1) = \frac{A}{m_2}\frac{1}{T}\sum_{i=1}^{N=\rho T}\underbrace{\int_0^T z(t)\delta(t-t_i-\upsilon_1)\mathrm{d}t}_{z(t_i+\upsilon_1)} - \frac{\rho A}{m_2}\underbrace{\frac{1}{T}\int_0^T z(t)\mathrm{d}t}_{\langle z\rangle}$$

When using the sifting property it can be seen that the first term is a scaled average of $z(t_i+\upsilon_1)$ and may be rewritten as:

$$\frac{A}{m_2}\frac{\rho T}{T}\underbrace{\frac{1}{\rho T}\sum_{i=1}^{N=\rho T} z(t_i+\upsilon_1)}_{C_{zx}(\upsilon_1)} = \frac{\rho A}{m_2}C_{zx}(\upsilon_1) = \frac{\mu}{m_2}C_{zx}(\upsilon_1)$$

Note that we used the first moment $\mu = \rho A$ of the original train of impulses χ here (Equation (5.1a)). Combining the above we get:

$$\boxed{p_1(\upsilon_1) = \frac{\mu}{m_2}[C_{zx}(\upsilon_1) - \langle z\rangle]} \tag{5.22b}$$

The average $(1/\rho T)\sum_{i=1}^{N=\rho T} z(t_i+\upsilon_1)$ is the cross-correlation $C_{zx}(\upsilon_1)$ between the input impulse train x and the system's output z. **Unlike the reverse correlation we discussed in section 14.5 in van Drongelen (2007) and applied in Section 4.6, we deal with the forward-correlation here (see Fig. 5.5E).** In the examples in Chapter 4, we used reversed correlation because the impulse train was the output caused by the input and we had to go back in time to reflect this causality. In this case the role is reversed: the impulse train is the input causing the output.

5.3.3 Determination of the Second-Order Poisson–Wiener Kernel

Using a procedure analogous to that developed for the Wiener kernel in Section 4.3.3, we find the second-order Poisson–Wiener kernel by using a second-order cross-correlation between output and input:

$$\begin{aligned}\langle z(t)x(t-\upsilon_1)x(t-\upsilon_2)\rangle &= \langle P_0[p_0; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle \\ &\quad + \langle P_1[p_1; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle \\ &\quad + \langle P_2[p_2; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle + \cdots \\ &= \sum_{n=0}^{N}\langle P_n[p_n; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle\end{aligned} \tag{5.23}$$

Since $x(t-\upsilon_1)x(t-\upsilon_2)$ can be presented as a second-order Volterra operator (see Appendix 4.2), all Poisson–Wiener operators P_n with $n \geq 3$ are orthogonal to $x(t-\upsilon_1)x(t-\upsilon_2)$ (because all Poisson–Wiener operators are orthogonal to lower-order Volterra operators). Furthermore, since we use a Poisson process as input, we will not allow impulses to coincide. Therefore, we neglect all results for equal delays $\upsilon_1 = \upsilon_2$ in the evaluation of Equation (5.23). Taking into account the considerations above, we now analyze the second-order cross-correlation for $n = 0$, 1, 2, and $\upsilon_1 \neq \upsilon_2$.

For $\boldsymbol{n = 0}$:

$$\langle P_0[p_0; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle = p_0 \underbrace{\langle x(t-\upsilon_1)x(t-\upsilon_2)\rangle}_{m_2\delta(\upsilon_1-\upsilon_2)} = m_2 p_0 \delta(\upsilon_1 - \upsilon_2) \tag{5.24a}$$

We can neglect this term because, due to the Dirac, it evaluates to zero for $\upsilon_1 \neq \upsilon_2$.

For $\boldsymbol{n = 1}$:

$$\begin{aligned}\langle P_1[p_1; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle &= \int_{-\infty}^{\infty} p_1(\tau_1) \underbrace{\langle x(t-\tau_1)x(t-\upsilon_1)x(t-\upsilon_2)\rangle}_{m_3\delta(\tau_1-\upsilon_1)\delta(\tau_1-\upsilon_2)} \mathrm{d}\tau_1 \\ &= m_3 p_1(\upsilon_1)\delta(\upsilon_1-\upsilon_2)\end{aligned} \tag{5.24b}$$

Due to the Dirac, this expression also evaluates to zero for $\upsilon_1 \neq \upsilon_2$ and can therefore be ignored.

For $\boldsymbol{n = 2}$, we compute $\langle P_2[p_2; x(t)]x(t-\upsilon_1)x(t-\upsilon_2)\rangle$ using Equation (5.17) and we get:

$$\begin{aligned}&\overbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2) \underbrace{\langle x(t-\tau_1)x(t-\tau_2)x(t-\upsilon_1)x(t-\upsilon_2)\rangle}_{\text{Equation (5.6b)}} \mathrm{d}\tau_1\,\mathrm{d}\tau_2}^{\mathrm{I}} \\ &\overbrace{-\frac{m_3}{m_2}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_1) \underbrace{\langle x(t-\tau_1)x(t-\upsilon_1)x(t-\upsilon_2)\rangle}_{m_3\delta(\tau_1-\upsilon_1)\delta(\tau_1-\upsilon_2)} \mathrm{d}\tau_1}^{\mathrm{II}} \\ &\overbrace{-\,m_2\int_{-\infty}^{\infty} p_2(\tau_1,\tau_1) \underbrace{\langle x(t-\upsilon_1)x(t-\upsilon_2)\rangle}_{m_2\delta(\upsilon_1-\upsilon_2)} \mathrm{d}\tau_1}^{\mathrm{III}}\end{aligned} \tag{5.24c}$$

Term I in Equation (5.24c) is the most complex one and potentially consists of four terms (Equation (5.6b)). Given that we have four delays τ_1, τ_2, υ_1, υ_2 and taking into account the condition $\upsilon_1 \neq \upsilon_2$, there are only two combinations that remain to be considered: $\tau_1 = \upsilon_1$ and $\tau_2 = \upsilon_2$ and $\tau_1 = \upsilon_2$ and $\tau_2 = \upsilon_1$. The first term can now be rewritten as:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)\underbrace{\langle x(t-\tau_1)x(t-\tau_2)x(t-\upsilon_1)x(t-\upsilon_2)\rangle}_{m_2^2\delta(\tau_1-\upsilon_1)\delta(\tau_2-\upsilon_2)+m_2^2\delta(\tau_1-\upsilon_2)\delta(\tau_2-\upsilon_1)}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)[m_2^2\delta(\tau_1-\upsilon_1)\delta(\tau_2-\upsilon_2)+m_2^2\delta(\tau_1-\upsilon_2)\delta(\tau_2-\upsilon_1)]\mathrm{d}\tau_1\,\mathrm{d}\tau_2$$

$$= m_2^2\underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)\delta(\tau_1-\upsilon_1)\delta(\tau_2-\upsilon_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2}_{p_2(\upsilon_1,\upsilon_2)}$$

$$+ m_2^2\underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_2)\delta(\tau_1-\upsilon_2)\delta(\tau_2-\upsilon_1)\mathrm{d}\tau_1\,\mathrm{d}\tau_2}_{p_2(\upsilon_2,\upsilon_1)}$$

The double integral above can be evaluated by sifting for τ_1 and τ_2. Because we assume that the kernel is symmetric around its diagonal, we can use $p_2(\upsilon_1,\upsilon_2) = p_2(\upsilon_2,\upsilon_1)$, and the above evaluates to:

$$\boxed{2m_2^2 p_2(\upsilon_1,\upsilon_2)} \tag{5.24d}$$

Term II in Equation (5.24c) can be written as:

$$-\frac{m_3}{m_2}\int_{-\infty}^{\infty} p_2(\tau_1,\tau_1)m_3\delta(\tau_1-\upsilon_1)\delta(\tau_1-\upsilon_2)\mathrm{d}\tau_1 = -\frac{m_3^2}{m_2}p_2(\upsilon_1,\upsilon_1)\delta(\upsilon_1-\upsilon_2) \tag{5.24e}$$

Due to the delta function $\delta(\upsilon_1 - \upsilon_2)$, this part can be neglected since it is zero for $\upsilon_1 \neq \upsilon_2$.

Term III in Equation (5.24c) evaluates to:

$$-m_2\int_{-\infty}^{\infty} p_2(\tau_1,\tau_1)m_2\delta(\upsilon_1-\upsilon_2)\mathrm{d}\tau_1 = -m_2^2\int_{-\infty}^{\infty} p_2(\tau_1,\tau_1)\delta(\upsilon_1-\upsilon_2)\mathrm{d}\tau_1 \tag{5.24f}$$

This term can also be ignored because it equals zero for $\upsilon_1 \neq \upsilon_2$.
To summarize Equation (5.24), the only nonzero term for $\upsilon_1 \neq \upsilon_2$ is the result in Equation (5.24d). Substituting this result into Equation (5.23), we get an expression for our second-order Poisson–Wiener kernel p_2:

$$\boxed{\begin{aligned}&\langle z(t)x(t-\upsilon_1)x(t-\upsilon_2)\rangle = 2m_2^2 p_2(\upsilon_1,\upsilon_2)\rightarrow\\ &p_2(\upsilon_1,\upsilon_2) = \frac{1}{2m_2^2}\langle z(t)x(t-\upsilon_1)x(t-\upsilon_2)\rangle \quad \text{for } \upsilon_1 \neq \upsilon_2\end{aligned}} \tag{5.25a}$$

Using the fact that the input x is a train of impulses, we can employ the same treatment as for Equation (5.22a) and rewrite Equation (5.25a) as:

$$p_2(\upsilon_1,\upsilon_2) = \frac{1}{2m_2^2}\frac{1}{T}\int_0^T z(t)\overbrace{\left[A\sum_{i=1}^{N=\rho T}\delta(t-t_i-\upsilon_1)-\rho A\right]}^{\text{1st copy of the input: } x(t-\upsilon_1)} \times \overbrace{\left[A\sum_{j=1}^{N=\rho T}\delta(t-t_j-\upsilon_2)-\rho A\right]}^{\text{2nd copy of the input: } x(t-\upsilon_2)}\mathrm{d}t$$

This expression generates four terms:

$$\text{I:}\quad \frac{A^2}{2m_2^2}\frac{1}{T}\int_0^T z(t)\left[\sum_{i=1}^{N=\rho T}\delta(t-t_i-\upsilon_1)\right]\left[\sum_{j=1}^{N=\rho T}\delta(t-t_j-\upsilon_2)\right]\mathrm{d}t$$

$$\text{II:}\quad -\frac{\rho A^2}{2m_2^2}\frac{1}{T}\int_0^T z(t)\sum_{i=1}^{N=\rho T}\delta(t-t_i-\upsilon_1)\mathrm{d}t = \frac{\mu^2}{2m_2^2}C_{zx}(\upsilon_1), \text{ with:}$$

$$C_{zx}(\upsilon_1) = \frac{1}{\rho T}\sum_{i=1}^{N=\rho T} z(t_i+\upsilon_1) \text{ and } \mu = \rho A$$

$$\text{III:}\quad -\frac{\rho A^2}{2m_2^2}\frac{1}{T}\int_0^T z(t)\sum_{j=1}^{N=\rho T}\delta(t-t_j-\upsilon_2)\mathrm{d}t = \frac{\mu^2}{2m_2^2}C_{zx}(\upsilon_2), \text{ with:}$$

$$C_{zx}(\upsilon_2) = \frac{1}{\rho T}\sum_{j=1}^{N=\rho T} z(t_j+\upsilon_2) \text{ and } \mu = \rho A$$

$$\text{IV:}\quad \frac{\rho^2A^2}{2m_2^2}\frac{1}{T}\int_0^T z(t)\mathrm{d}t = \frac{\mu^2}{2m_2^2}\langle z\rangle$$

Terms II–IV were evaluated in a similar fashion as in the first-order case in Equation (5.22). After changing the order of the integration and summations, term I above evaluates to:

$$\frac{A^2}{2m_2^2}\frac{1}{T}\sum_{i=1}^{N=\rho T}\sum_{j=1}^{N=\rho T}\underbrace{\int_0^T z(t)[\delta(t-t_i-\upsilon_1)][\delta(t-t_j-\upsilon_2)]\mathrm{d}t}_{z(t_i+\upsilon_1)\delta(t_i-t_j+\upsilon_1-\upsilon_2)}$$

$$=\frac{A^2}{2m_2^2}\frac{\rho^2T^2}{T}\underbrace{\frac{1}{\rho^2T^2}\sum_{i=1}^{N=\rho T}\sum_{j=1}^{N=\rho T}z(t_i+\upsilon_1)\delta(t_i-t_j+\upsilon_1-\upsilon_2)}_{C_{zxx}(\upsilon_1,\upsilon_2)} = \frac{\mu^2}{2m_2^2}TC_{zxx}(\upsilon_1,\upsilon_2)$$

In the above expression we substituted μ for ρA (Equation (5.1a)); this is the first moment of the original impulse train χ. Combining the results from the four terms I–IV above, we have:

$$\boxed{p_2(\upsilon_1,\upsilon_2)=\frac{\mu^2}{2m_2^2}\{TC_{zxx}(\upsilon_1,\upsilon_2)-[C_{zx}(\upsilon_1)+C_{zx}(\upsilon_2)-\langle z\rangle]\}\quad \text{for } \upsilon_1\neq\upsilon_2} \tag{5.25b}$$

The second-order correlation $C_{zxx}(\upsilon_1,\upsilon_2)$ is the average of $(1/\rho^2T^2)$ $\sum_{i=1}^{N=\rho T}\sum_{j=1}^{N=\rho T} z(t_i+\upsilon_1)$ under the condition set by the Dirac $\delta(t_i-t_j+\upsilon_1-\upsilon_2)$. This condition is equivalent to sampling the values of output signal z when $t_i-t_j+\upsilon_1-\upsilon_2=0$. This indicates that:

(1) The delay between the copies of the input is $\Delta=\upsilon_2-\upsilon_1=t_i-t_j$, which means that the delays under consideration for creating the averages are equal to the differences Δ between spike times t_i, t_j.

(2) There is a relationship between the individual delays given by $\upsilon_2=t_i-t_j+\upsilon_1$, which represents a line in the υ_1,υ_2 plane at 45° and with an intercept at t_i-t_j.

This conditional average is therefore a slice through $p_2(\upsilon_1,\upsilon_2)$ defined by this line. The delays we consider are strictly given by t_i-t_j and the input to the averaging procedure is $z(t_i+\upsilon_1)$. A representation of C_{zxx} is shown in Fig. 5.5F. To keep Fig. 5.5F compatible with the symbols in the other panels in this figure, the delay υ_1 is replaced by τ_1 in the diagram.

5.4 Implementation of the Cross-Correlation Method

Because there is no standard command in MATLAB to create a series of randomly occurring impulses following a Poisson process, we include an example function Poisson.m to create such an impulse train (for details see Appendix 5.2). In MATLAB script Pr5_1.m, we use this function to create the input (in

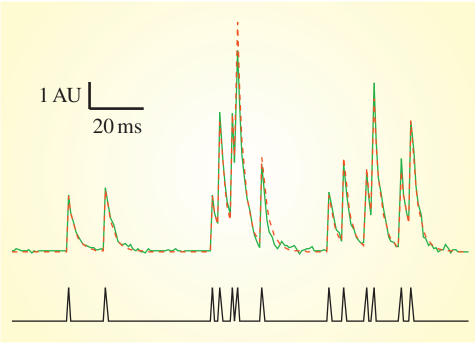

Figure 5.3 Example of input (pulses, lower line (black)) and output (dashed line (red)) traces. The (green) line, following the output closely, is the output contribution from the Poisson−Wiener kernels. The vertical scale is in arbitrary units (AU). The VAF by the model output in this example was 97.6%. All traces were generated by Pr5_1.m.

this example, impulses with amplitude of 2 units) to a nonlinear system consisting of a first-order component (a low-pass filter) and a second-order component (a low-pass filter amplifier with 5× amplification plus a squarer), similar to the system in Fig. 3.2C. Typical traces for input and output are shown in Fig. 5.3. By following the same steps depicted in Fig. 4.2 for the Wiener kernels, but now using Equations (5.19), (5.22b), and (5.25b), we find the Poisson−Wiener kernels for the system. Note that the cross-correlations are impulse-triggered averages in this case.

The following MATLAB code is part of script Pr5_1.m and shows the computation of the first-order cross-correlation and first-order kernel p1 according to Equation (5.22b) (Step 3 of the Lee−Schetzen method depicted in Fig. 4.2).

```
% Step 3. Create the first order average (see Fig. 4.2)
%
Czx=zeros(T,1);
for i=1:length(time)-10                % to avoid problems by ignoring last
                                       % 10 impulses
    Czx=Czx+v0(time(i):time(i)+T-1);
end;
% Now we scale Czx by the # of spikes (i.e. length(time) − 10, which is the
% # of trials in the average. Using Equation (5.22b):
p1=(u1/m2)*((Czx/(length(time)-10))-mean(v0));   % Note that all scaling
                                                  % parameters
                                                  % u1, m2, and mean(v0)
                                                  % are at the ms - scale !

figure;
plot(p1)
title('first order Poisson-Wiener kernel')
xlabel('Time (1 ms)')
ylabel('Amplitude')
```

The percentage of variance accounted for (VAF, see Section 4.4 for its definition) by the output from the Poisson–Wiener kernels in this example is typically in the high 90 s. This VAF number is fairly optimistic because, as can be seen in the output trace in Fig. 5.3, a large number of points with a good match between output (dashed red line) and predicted output (green line) are zero or close to zero; the predicted output we mainly care about is (of course) the activity caused by the input (impulses) and not the rest state.

5.5 Spiking Output

In Chapter 4, we considered continuous input to nonlinear systems with both continuous and spiking output. So far in this chapter, we have analyzed nonlinear systems with spike train input and continuous output. The possible cases one might encounter in neuroscience are summarized in Fig. 5.4. As you can see, the only case remaining for our discussion is a nonlinear system with both spike input and output (Fig. 5.4D). We can compute the Poisson–Wiener kernels by using the previously found expressions (Equations (5.19), (5.22b), and (5.25b)). In this case, kernel p_0 can be determined by the time average of the spike output. Just as in Equation (4.30) p_0 evaluates to the mean firing rate of the output. In Equations (5.22b) and (5.25b)

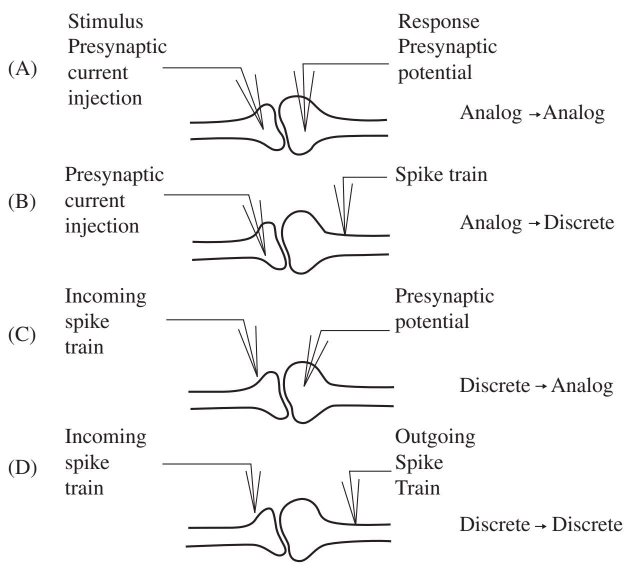

Figure 5.4 Types of signals one may encounter for a system's input and output in neuroscience. In this example a synapse is used to symbolize the four different possibilities (A–D). The incoming signal may be a GWN signal (analog–presynaptic current injection) or a train of impulses following a Poisson process (discrete–incoming spike train). The output can be a postsynaptic potential (analog) or a spike train (discrete). *(Fig. 11.1 from Marmarelis and Marmarelis (1978). With kind permission of Springer Science and Business Media.)*

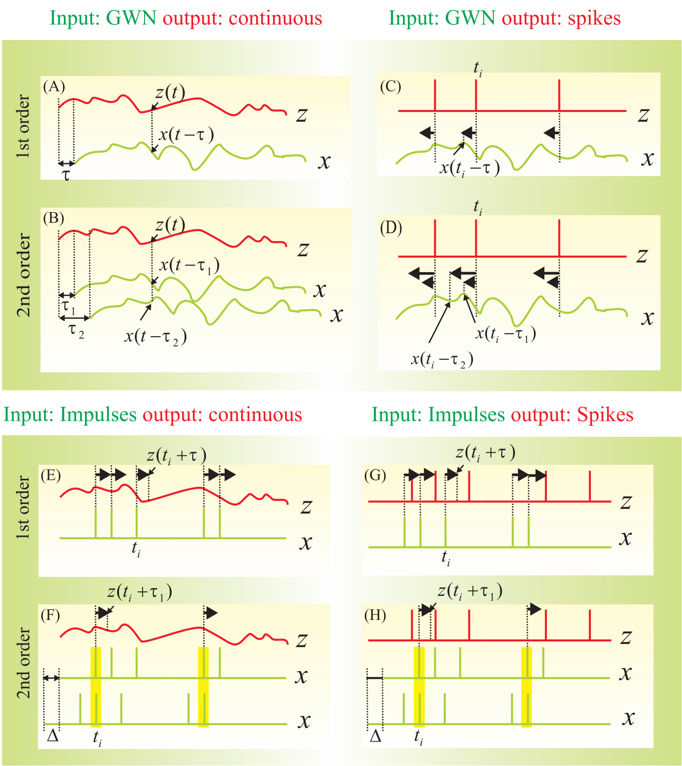

Figure 5.5 Summary of the procedures for determining first- and second-order cross-correlation for the different scenarios depicted in Fig. 5.4. In all panels, x (green) is the input and z (red) is the output. The panels for GWN input (A–D) are identical to Fig. 4.6A, B, E, F. Panels E–H show the procedures for impulses as input. See text for further explanation.

we can see that computing first- and second-order kernels require first- and second-order cross-correlations C_{zx} and C_{zxx} (in this case spike-triggered averages). The procedures for obtaining these cross-correlations are depicted in Fig. 5.5G and H. The first-order cross-correlation is a spike-triggered average; we use the input spikes as the trigger (Fig. 5.5G). The second-order cross-correlation is triggered by coinciding spikes of two copies from the input, one of which is shifted by amount Δ (Fig. 5.5H). The procedures for obtaining these cross-correlation functions are very similar to the ones discussed for a system with a spike input and continuous output, as you can see by comparing panels E with G and F with H in Fig. 5.5.

5.6 Summary

The procedures for determining the first- and second-order cross-correlations for the four scenarios in Fig. 5.4 are summarized in Fig. 5.5. The part of this figure for GWN input is identical to the overview in Fig. 4.6. The panels for spike input show the procedures discussed in this chapter. In practice, the cross-correlations required for computation of the Poisson–Wiener kernels can all be obtained from spike-triggered averages (Fig. 5.5E–H). As such it is very similar to the procedure we followed for nonlinear systems with GWN input and spike output in Chapter 4 (Fig. 5.5C and D). The difference is that here we use the input spikes, instead of the output spike train, to trigger the average; hence, we determine forward cross-correlation instead of reversed correlation. This reflects that the systems are considered causal (output is caused by input). Thus, a system's output shows reversed correlation with the input (Fig. 5.5C and D) and its input is forward-correlated with its output (Fig. 5.5E–H). The procedures followed to obtain the cross-correlations for systems with both continuous input and output are depicted in Fig. 5.5A and B. Here the correlation products are not spike-triggered and the delays of the copies of the input are determined for each sample of the output $z(t)$ (Chapter 4).

Appendix 5.1

Expectation and Time Averages of Variables Following a Poisson Process

The results for time averages of GWN are well known and were briefly summarized in Appendix 4.1. For the application of impulse trains we use a different input signal, the Poisson process (see section 14.2 in van Drongelen, 2007). Products of variables following a Poisson process are important for determining the Poisson–Wiener kernels when impulse trains are used as input to a nonlinear system. A similar derivation was described by Krausz (1975) in his appendix A.[2] Assuming that $x(t)$ follows a Poisson process, we can define the **first moment** as the Expectation of x: $E\{x\}$. Because the signal is ergodic, we may replace this with a time average $\langle x\rangle = (1/T)\int_0^T x(t)\mathrm{d}t$ (see section 3.2 in van Drongelen, 2007, if you need to review ergodicity and time averages). To simplify things further down the road, we start from a demeaned impulse train so that (see Equation (5.1)):

$$\boxed{E\{x\} = \langle x\rangle = 0} \tag{A5.1.1}$$

The Expectation of the **second-order product**, or cross-correlation, of variable x is $E\{x(t-\tau_1)x(t-\tau_2)\}$ (for cross-correlation, see section 8.4 in van Drongelen, 2007). Note that the expression we use here is slightly different from Equation (8.13) in van Drongelen (2007): we substituted $t-\tau_1$ and $t-\tau_2$ for t_1 and t_2, respectively. Because x follows a Poisson process, the factors $x(t-\tau_1)$ and

[2] Please note that the derivation by Krausz contains minor errors for the moments m_2 and m_3, leading to differences in the scaling of several of the derived expressions.

$x(t-\tau_2)$ are independent if $\tau_1 \neq \tau_2$; in this case we may replace the Expectation with two separate ones—that is:

$$E\{x(t-\tau_1)x(t-\tau_2)\} = E\{x(t-\tau_1)\}E\{x(t-\tau_2)\} = 0 \quad \text{for } \tau_1 \neq \tau_2$$

The above product evaluates to zero, because the first moment of our impulse train is zero. The expression $E\{x(t-\tau_1)x(t-\tau_2)\}$ is only nonzero if $\tau_1 = \tau_2$, and (again) because x is ergodic we may apply a time average $\langle x(t-\tau_1)x(t-\tau_2)\rangle$. In Section 5.2.1 you can see that the final result for the Expectation/time average of the second-order product becomes:

$$\boxed{E\{x(t-\tau_1)x(t-\tau_2)\} = \langle x(t-\tau_1)x(t-\tau_2)\rangle = m_2\delta(\tau_1-\tau_2)} \tag{A5.1.2}$$

The Expectation of the **third-order product** $E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\}$ equals zero by independence if $\tau_1 \neq \tau_2 \neq \tau_3$, since in this case we can rewrite the expression as:

$$E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\} = E\{x(t-\tau_1)\}E\{x(t-\tau_2\}E\{x(t-\tau_3)\} = 0$$
$$\text{for } \tau_1 \neq \tau_2 \neq \tau_3$$

If only one pair of τ's is equal (i.e., $\tau_1 = \tau_2 \neq \tau_3$ or $\tau_1 \neq \tau_2 = \tau_3$), we can make the substitutions $\tau_1 = \tau_2$ or $\tau_2 = \tau_3$ and then separate the Expectation into two factors:

$$\begin{aligned} E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\} &= E\{x(t-\tau_1)x(t-\tau_1)x(t-\tau_3)\} \\ &= E\{x(t-\tau_1)^2x(t-\tau_3)\} \qquad \text{for } \tau_1 = \tau_2 \neq \tau_3 \\ &= E\{x(t-\tau_1)^2\}E\{x(t-\tau_3)\} = 0 \end{aligned}$$

and,

$$E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\} = E\{x(t-\tau_2)^2\}E\{x(t-\tau_1)\} = 0 \quad \text{for } \tau_1 \neq \tau_2 = \tau_3$$

In all of the above cases, the expressions evaluate to zero because $E\{x\} = 0$, and the only instance where the Expectation of the third-order product is nonzero is for $\tau_1 = \tau_2 = \tau_3$. In this case (due to ergodicity), it may be replaced by $\langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\rangle$ (see Equation (5.5b)). The final nonzero result is:

$$\boxed{\begin{aligned} E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\} &= \langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\rangle \\ &= m_3\delta(\tau_1-\tau_2)\delta(\tau_1-\tau_3) \end{aligned}} \tag{A5.1.3}$$

The Expectation of the **fourth-order product** $E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)$ $x(t-\tau_4)\}$ is zero by independence if:

I. $\tau_1 \neq \tau_2 \neq \tau_3 \neq \tau_4$

and nonzero if all delays are equal:

II. $\tau_1 = \tau_2 = \tau_3 = \tau_4$

Using the time average approach we use in Section 5.2, we find the following for the fourth moment:

$$m_4 = \langle x^4 \rangle = \frac{1}{T}\int_0^T [\rho T(A-A\rho)^4 \delta(t) + (1-\rho)T(-A\rho)^4 \delta(t)]\mathrm{d}t$$

This can be written as:

$$m_4 = \langle x^4 \rangle = \rho A^4(1-4\rho+6\rho^2-3\rho^3) = \rho A^4[\rho(1-\rho)^2 + (1-\rho)(1-2\rho)^2]$$

Including the condition $\tau_1 = \tau_2 = \tau_3 = \tau_4$, we find that the averaged product is:

$$\boxed{\langle x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x(t-\tau_4)\rangle = m_4\delta(\tau_1-\tau_2)\delta(\tau_1-\tau_3)\delta(\tau_1-\tau_4)} \tag{A5.1.4}$$

in which the δ functions represent the condition that all delays must be equal for a nonzero result. Three alternatives with three equal delays are:

III. $\tau_1 \neq \tau_2 = \tau_3 = \tau_4$
IV. $\tau_1 = \tau_2 \neq \tau_3 = \tau_4$
V. $\tau_1 = \tau_2 = \tau_3 \neq \tau_4$

In all three cases **III–V**, the Expectation of the fourth-order product evaluates to zero. For instance in case **V** we have:

$$E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x(t-\tau_4)\} = E\{x(t-\tau_1)^3 x(t-\tau_4)\}$$
$$= \underbrace{E\{x(t-\tau_1)^3\}}_{m_3}\underbrace{E\{x(t-\tau_4)\}}_{0} = 0$$
$$\text{for } \tau_1 = \tau_2 = \tau_3 \neq \tau_4$$

Finally, we have three cases in which delays are equal in pairs:

VI. $\tau_1 = \tau_2$ and $\tau_3 = \tau_4$
VII. $\tau_1 = \tau_3$ and $\tau_2 = \tau_4$
VIII. $\tau_1 = \tau_4$ and $\tau_2 = \tau_3$

These cases evaluate to a nonzero value. For instance, in case **VI** we get:

$$E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x(t-\tau_4)\} = E\{x(t-\tau_1)^2 x(t-\tau_3)^2\}$$
$$= \underbrace{E\{x(t-\tau_1)^2\}}_{m_2}\underbrace{E\{x(t-\tau_3)^2\}}_{m_2} = m_2^2$$
$$\text{for } \tau_1=\tau_2 \text{ and } \tau_3=\tau_4$$

If we represent the conditions $\tau_1=\tau_2$ and $\tau_3=\tau_4$, with Dirac delta functions, we get the final result for case **VI**:

$$m_2^2\delta(\tau_1-\tau_2)\delta(\tau_3-\tau_4)$$

To summarize the results for the Expectation of the fourth-order product:

$$\boxed{\begin{array}{l} E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x(t-\tau_4)\} \\ \begin{cases} \tau_1=\tau_2 \text{ and } \tau_3=\tau_4 & : m_4\delta(\tau_1-\tau_2)\delta(\tau_1-\tau_3)\delta(\tau_1-\tau_4) \\ \tau_1=\tau_2 \text{ and } \tau_3=\tau_4 & : m_2^2\delta(\tau_1-\tau_2)\delta(\tau_3-\tau_4) \\ \tau_1=\tau_3 \text{ and } \tau_2=\tau_4 & : m_2^2\delta(\tau_1-\tau_3)\delta(\tau_2-\tau_4) \\ \tau_1=\tau_4 \text{ and } \tau_2=\tau_3 & : m_2^2\delta(\tau_1-\tau_4)\delta(\tau_2-\tau_3) \\ 0 & \ \ \text{otherwise} \end{cases} \end{array}} \tag{A5.1.5}$$

In Section 5.3.3 we have to evaluate a case where we know that one pair of delays cannot be equal. Note that in such a case we have to combine from alternatives **VI–VIII**. For example if $\tau_2 \neq \tau_3$, we have two possibilities for pair forming:

(a) $\tau_1=\tau_2$ and $\tau_3=\tau_4$ in which pair τ_1, τ_2 is independent from pair τ_3, τ_4
(b) $\tau_1=\tau_3$ and $\tau_2=\tau_4$ in which pair τ_1, τ_3 is independent from pair τ_2, τ_4.

Now we can write the Expectation for $\tau_2 \neq \tau_3$ as the sum of (a) and (b):

$$E\{x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x(t-\tau_4)\}_{\ \tau_2\neq\tau_3}$$
$$= E\{x(t-\tau_1)^2 x(t-\tau_3)^2\} + E\{x(t-\tau_1)^2 x(t-\tau_2)^2\}$$
$$= \overbrace{\underbrace{E\{x(t-\tau_1)^2\}}_{m_2}\underbrace{E\{x(t-\tau_3)^2\}}_{m_2}}^{\text{case } a} + \overbrace{\underbrace{E\{x(t-\tau_1)^2\}}_{m_2}\underbrace{E\{x(t-\tau_2)^2\}}_{m_2}}^{\text{case } b}$$
$$= m_2^2\delta(\tau_1-\tau_2)\delta(\tau_3-\tau_4) + m_2^2\delta(\tau_1-\tau_3)\delta(\tau_2-\tau_4) \tag{A5.1.6}$$

Appendix 5.2

Creating Impulse Trains Following a Poisson Process

For the generation of a series of random numbers following a Gaussian or a uniform distribution, we use MATLAB commands `randn` and `rand`, respectively. A standard MATLAB command for generating a series of intervals according to a Poisson process does not exist. Therefore, we will apply a Monte Carlo technique to create such an impulse train according to a Poisson process. Our target is to follow a Poisson process characterized by probability density function (PDF) $\rho e^{-\rho x}$ (see Chapter 14 in van Drongelen, 2007). This works as follows. First we generate pairs of independent random numbers x,y with the MATLAB `rand` command. Because the `rand` command generates numbers between 0 and 1, x is multiplied with the maximal epoch value we want to consider, in order to rescale it between 0 and the maximum interval. Second, for each trial we compute $p = \rho e^{-\rho x}$, which is the probability p for interval x to occur according to the Poisson process. So far we will generate intervals x where all intervals have an equal probability because the MATLAB `rand` command is uniformly distributed. The second random number y associated with the randomly generated interval will also be evenly distributed between 0 and 1. We now only include pairs x,y in our series if $y < p$ and discard all others (Fig. A5.2.1); by following this procedure, the accepted intervals x obey the Poisson process because the probability that they are retained is proportional with $\rho e^{-\rho x}$, which is the desired probability. This procedure can, of course, be used for other distributions as well; it is known as the accept–reject algorithm.

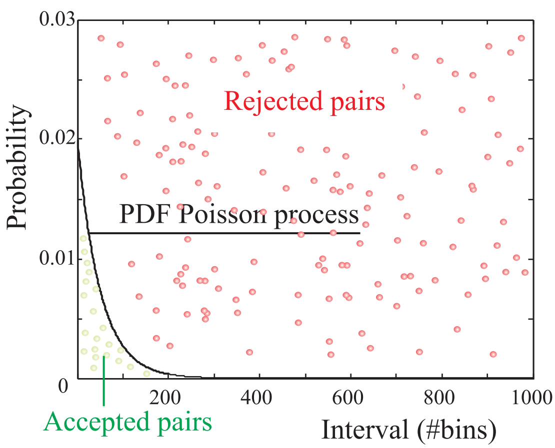

Figure A5.2.1 The Poisson process PDF can be used to create series of intervals obeying a Poisson process. Pairs of random uniformly distributed numbers x,y are generated: x is scaled between 0 and the maximum epoch length (1000 in this example) and y between 0 and 1. Each pair is then plotted in the X–Y plane. If $y < \rho e^{-\rho x}$ the point is accepted (green); otherwise it is rejected (red). If sufficient numbers are evaluated, the result is that epochs are retained according to the PDF describing the Poisson process.

The following MATLAB snippet of the function Poisson.m shows an implementation of the procedure to generate a series of intervals following a Poisson process. This function is applied in pr5_1.m. Note that this routine also avoids intervals that are smaller than one bin because we do not allow for superimposed impulses.

```
i=1;
while (i < len)
        x=rand;y=rand;              % two random numbers scaled 0-1
        x=x*epoch;                  % the interval x is scaled 0-epoch
        p=rate*exp(-rate*x);        % the probability associated with the interval
                                    % using the second random number using the
                                    % Poisson process PDF
        if (y < p);                 % Is the probability below the random # ?
        if x > 1;                   % Avoid intervals that are too small ( < 1 bin)
        series(i)=x;                % else the interval is included
        i=i+1;
        end;
        end;
end;
```