In *The Wigner Distribution - Theory and Applications in Signal Processing*
W. Mecklenbräuker and F. Hlawatsch, Eds., Amsterdam: Elsevier, 1997, pp. 135-209.

Signal synthesis algorithms for bilinear time-frequency signal representations

F. Hlawatsch[a] and W. Krattenthaler[b]

[a]Inst. für Nachrichtentechnik u. Hochfrequenztechnik, Technische Universität Wien, Gusshausstrasse 25/389, A-1040 Vienna, Austria (fhlawats@email.tuwien.ac.at)

[b]Forschungszentrum Seibersdorf, A-2444 Seibersdorf, Austria

Abstract

In the application of bilinear time-frequency representations (BTFRs) to signal design and signal processing, the problem of *signal synthesis* plays an important role. This chapter discusses signal synthesis algorithms for various BTFRs and BTFR classes. First, a general theory of signal synthesis is formulated for the class of subspace-unitary BTFRs. For the discrete-time Wigner distribution (WD), algorithms for both global and halfband-constrained signal synthesis are presented. Extensions of these algorithms yield iterative signal synthesis algorithms for the class of smoothed WDs and the spectrogram, as well as a recursive "on-line" algorithm for the pseudo-WD. Various algorithms for removing the phase ambiguities of signal synthesis are considered, Finally, the application of signal synthesis to time-frequency filtering is discussed.

1. INTRODUCTION

This chapter discusses the signal synthesis problem for bilinear time-frequency signal representations, with emphasis placed on the Wigner distribution (WD) and smoothed WD versions. In the Introduction, we start with a general formulation of bilinear time-frequency representations, give a list of important representations, and state the basic problem of (optimal) signal synthesis. We finally give an overview of the material covered in the subsequent sections.

BTFRs. *Bilinear time-frequency representations* (BTFRs) of signals, like the WD, ambiguity function, and spectrogram, seek to combine the notions of time-domain description and frequency-domain description into a single two-dimensional signal representation [1-9]. While the primary application of BTFRs is the analysis of signals, BTFRs can also be used for signal design and signal processing. Here, the problem of *signal synthesis* plays a central role [1, 10-16].

Let x(n) and y(n) be two discrete-time signals with Fourier transforms $X(\Theta)$ and $Y(\Theta)$, respectively. The variables n and Θ denote the discrete time index and the normalized frequency, respectively. We note that the Fourier transform of a signal,

$$X(\Theta) = \sum_n x(n)\, e^{-j2\pi\Theta n} , \tag{1.1}$$

is a 1-periodic function of the normalized-frequency variable Θ. (In (1.1) and subsequent equations, summations are from $-\infty$ to ∞ unless explicitly stated otherwise.)

Any BTFR of the signals x(n) and y(n) can be written as

$$T_{x,y}(n,\Theta) = \sum_k \sum_l u_T(n,\Theta;k,l)\, q_{x,y}(k,l) , \tag{1.2}$$

where

$$q_{x,y}(k,l) \triangleq x(k)\, y^*(l)$$

is the *outer signal product* of x(n) and y(n), and $u_T(n,\Theta;k,l)$ is a kernel function which specifies the respective BTFR T [17,9]. This kernel function is in fact the BTFR's "impulse response" since

$$u_T(n,\Theta;k,l) = T_{x,y}(n,\Theta) \quad \text{for} \quad x(n) = \delta(n-k) \quad \text{and} \quad y(n) = \delta(n-l) , \tag{1.3}$$

where $\delta(n)$ denotes the unit sample. Eq. (1.3) provides a convenient way for calculating the impulse response for a given BTFR.

Important BTFRs which will be considered in later sections are the *Wigner distribution (WD)* [18,6,1]

$$WD_{x,y}(n,\Theta) = 2 \sum_m x(n+m)\, y^*(n-m)\, e^{-j4\pi\Theta m} ; \tag{1.4}$$

a modified definition of the WD (not equivalent to (1.4)) given by [19]

$$WD'_{x,y}(n,\Theta) = \sum_m x(m)\, y^*(n-m)\, e^{-j2\pi\Theta(2m-n)} ; \tag{1.5}$$

the *Rihaczek distribution* [2,6]

$$RD_{x,y}(n,\Theta) = \sum_m x(n+m)\, y^*(n)\, e^{-j2\pi\Theta m} ;$$

the *pseudo Wigner distribution (PWD)* [18,6]

$$PWD_{x,y}(n,\Theta) = 2 \sum_m x(n+m)\, y^*(n-m)\, h^2(m)\, e^{-j4\pi\Theta m} \tag{1.6}$$

where h(m) is a real-valued and even analysis window; the *smoothed pseudo Wigner distribution (SPWD)* [20-22,6]

$$SPWD_{x,y}(n,\Theta) = PWD_{x,y}(n,\Theta) \underset{n}{*} g(n) \tag{1.7}$$

where g(n) is a time smoothing window and $\underset{n}{*}$ denotes convolution with respect to the time index n; the *exponential distribution (ED)* [23,24,22,6,1]

$$ED_{x,y}(n,\Theta) = 2 \sum_m \sqrt{\frac{\sigma}{4\pi m^2}} \left[\sum_{n'} \exp\left(-\frac{\sigma(n-n')^2}{4m^2}\right) x(n'+m)\, y^*(n'-m) \right] e^{-j4\pi\Theta m} \qquad (1.8)$$

where $\sigma > 0$ is a parameter; and the *spectrogram* [2, 25, 6]

$$S_{x,y}(n,\Theta) = \left[\sum_{n_1} x(n_1)\, h(n_1-n)\, e^{-j2\pi\Theta n_1} \right] \left[\sum_{n_2} y(n_2)\, h(n_2-n)\, e^{-j2\pi\Theta n_2} \right]^* \qquad (1.9)$$

where h(n) is an analysis window. Furthermore, there exists a class of BTFRs with correlative interpretation [1,3], among which the following two definitions of the *ambiguity function (AF)* are the most important [2, 26, 6, 1]:

$$AF1_{x,y}(m,\zeta) = \sum_n x(n+m)\, y^*(n-m)\, e^{-j2\pi\zeta n} \qquad (1.10)$$

$$AF2_{x,y}(m,\zeta) = \sum_n x(n+m)\, y^*(n)\, e^{-j2\pi\zeta n} \; .$$

Here, m and ζ denote a time lag and (normalized) frequency lag, respectively. Correlative BTFRs, too, can be written in the general form (1.2) if n and Θ are replaced by m and ζ, respectively.

In most practical applications, the BTFR of a *single* signal x(n) is of interest; this is defined as $T_x = T_{x,x}$. We shall call T_x an *auto-BTFR* and $T_{x,y}$ (with $x(n) \neq y(n)$) a *cross-BTFR*.

The signal synthesis problem. Suppose, now, that T is a given BTFR defined, for example, by its impulse response $u_T(n,\Theta;k,l)$. Furthermore, let $\hat{T}(n,\Theta)$ be a given function which is 1-periodic with respect to the frequency variable Θ. Then, in most cases, $\hat{T}(n,\Theta)$ will not be a valid auto-BTFR of a signal x(n). This means that there does not exist a signal x(n) such that $T_x(n,\Theta) = \hat{T}(n,\Theta)$. Thus, if we want to specify a signal in the time-frequency domain, this cannot be done by simply constructing a desired BTFR outcome (time-frequency function) $\hat{T}(n,\Theta)$ and "inverting" the BTFR T, i.e., solving the equation $T_x(n,\Theta) = \hat{T}(n,\Theta)$. Instead, it is natural to resort to a least-squares approach [11] and derive the signal x(n) as the solution to the minimization problem

$$\varepsilon_x = \| \hat{T} - T_x \| \longrightarrow \min_x , \qquad (1.11)$$

where the error norm ε_x is defined by

$$\varepsilon_x^2 = \| \hat{T} - T_x \|^2 \triangleq \sum_n \int_{-1/2}^{1/2} | \hat{T}(n,\Theta) - T_x(n,\Theta) |^2 \, d\Theta \; .$$

The minimization (1.11) will be called *(optimal) signal synthesis*, and the time-frequency function $\hat{T}(n,\Theta)$ will be termed a *model*. In later sections, we will depart from strict optimality when considering signal synthesis algorithms which do not minimize the synthesis error ε_x but, rather, local versions of the synthesis error.

Also, we will consider *subspace-constrained* signal synthesis which incorporates a signal subspace constraint.

Already at this point, we can derive a general ambiguity of the signal synthesis solution x(n). From the general BTFR formulation (1.2), it follows that an auto-BTFR is invariant to a constant phase factor:

$$\breve{x}(n) = e^{j\varphi}\, x(n) \qquad \Rightarrow \qquad T_{\breve{x}}(n,\Theta) = T_x(n,\Theta) \qquad \text{for all } \varphi\ . \tag{1.12}$$

From this invariance, we further conclude that the solution of the signal synthesis problem (1.11) is *ambiguous* with respect to a constant phase factor. Indeed, if x(n) is a solution of optimal signal synthesis, then $\breve{x}(n) = e^{j\varphi}\, x(n)$ must be a solution as well since it achieves the same (minimal) synthesis error,

$$\varepsilon_{\breve{x}} = \|\breve{T}-T_{\breve{x}}\| = \|\breve{T}-T_x\| = \varepsilon_x\ . \tag{1.13}$$

This phase ambiguity can be resolved by means of a phase matching algorithm using a reference signal (see Section 7). An additional, more troublesome phase ambiguity will be seen to exist in the case of the WD, smoothed WD versions, and the AF version AF1.

Survey of chapter. The subsequent sections are organized as follows. Section 2 considers a subspace-constrained version of the signal synthesis problem in the general framework of BTFRs. In Subsection 2.1, it is shown that a linear signal subspace "induces" a corresponding BTFR-domain subspace for which an orthonormal basis is easily constructed provided that the BTFR satisfies a unitarity property on the given signal subspace. Based on this fact, a general algorithm for subspace-constrained optimal signal synthesis is derived in Subsection 2.2. This algorithm involves an eigenvalue-eigenvector problem for whose solution the well-known power algorithm is proposed in Subsection 2.3.

Section 3 discusses the signal synthesis problem for the case of the WD. The WD's unitarity properties are considered in Subsection 3.1. The well-known algorithm for *global signal synthesis* (i.e., synthesis without a subspace constraint) is then reviewed in Subsection 3.2. It is shown that this algorithm is equivalent to performing two independent subspace-constrained signal synthesis procedures. Subsection 3.3 shows that the problems of aliasing, non-unitarity, and relative phase invariance of the WD can be avoided by means of a specific signal subspace constraint. Accordingly, a modified signal synthesis algorithm termed *halfband signal synthesis algorithm* is then derived by specializing the general algorithm of Subsection 2.2. Finally, other approaches to WD signal synthesis are briefly discussed in Subsection 3.4.

Sections 4 through 6 consider BTFRs which are inherently non-unitary such that the standard algorithm of Subsection 2.2 cannot be applied. The class of *smoothed WD versions* (containing the PWD, SPWD, and ED) is considered in Section 4. After a general discussion of smoothed WD versions in Subsection 4.1, iterative algorithms for global and halfband signal synthesis are presented in Subsections 4.2 and 4.3, respectively. Since for smoothed WDs a closed-form solution to the signal synthesis

problem does not seem to exist, these algorithms are partly heuristic. In Section 5, the iterative method developed for smoothed WDs is reformulated for the spectrogram. Section 6 modifies the iterative scheme to derive "on-line" algorithms for global PWD signal synthesis. In contrast to the methods considered before, these algorithms permit the synthesis of signals with arbitrary length by recursively synthesizing individual signal blocks or samples, using local model segments.

Section 7 discusses phase matching algorithms for resolving the phase ambiguities of signal synthesis. In Subsection 7.1, a distinction is drawn between the "absolute" phase ambiguity affecting the entire signal and the "relative" phase ambiguity of even-indexed and odd-indexed signal samples relative to each other. Subsection 7.2 considers phase matching algorithms using a reference signal. Subsection 7.3 presents algorithms resolving the particularly troublesome relative phase ambiguity without using a reference signal. On-line versions of all phase matching algorithms are discussed in Subsection 7.4.

Finally, Section 8 studies the application of signal synthesis to the problem of time-frequency filtering. It is shown that WD-based signal synthesis suffers from potential "interference term effects," and various ways to cope with this problem are pointed out.

2. SUBSPACE-UNITARY BILINEAR TIME-FREQUENCY REPRESENTATIONS

In this section, we consider a general BTFR T defined by its impulse response $u_T(n,\Theta;k,l)$ as

$$T_{x,y}(n,\Theta) = \sum_k \sum_l u_T(n,\Theta;k,l)\, q_{x,y}(k,l) \qquad \text{where} \qquad q_{x,y}(k,l) = x(k)\, y^*(l)\ . \tag{2.1}$$

We wish to derive a solution to the following *subspace-constrained* version of signal synthesis [13]:

$$\varepsilon_x = \|\hat{T}-T_x\| \longrightarrow \min_{x\in\mathcal{G}}\ . \tag{2.2}$$

Here, $\mathcal{G}$ is a given N-dimensional linear signal space [27-29] spanned by an orthonormal basis $\{e_k(n)\}$, k=1,2,..,N. (We shall also consider infinite-dimensional signal spaces for which N=∞.) As a difference from the original signal synthesis problem (1.11), the formulation (2.2) incorporates a signal space constraint $x(n)\in\mathcal{G}$ which forces the signal synthesis solution x(n) to be an element of the signal space $\mathcal{G}$. Note, however, that the *unconstrained* signal synthesis problem (1.11) is a special case of the constrained version (2.2); it is obtained by choosing $\mathcal{G}$ to be the total space $l_2(-\infty,\infty)$ of square-integrable (finite-energy) signals. The unconstrained version of signal synthesis (with $\mathcal{G}=l_2(-\infty,\infty)$) will be called *global signal synthesis*; in contrast, subspace-constrained signal synthesis (where $\mathcal{G}\subset l_2(-\infty,\infty)$ is a proper subspace of $l_2(-\infty,\infty)$), will be called *subspace signal synthesis.*

The advantage of the subspace constraint $x(n)\in\mathcal{G}$ is that it provides a convenient way for enforcing certain properties of the synthesis result x(n). By prescribing a

suitable signal subspace $\mathcal{S}$, x(n) can be forced to be bandlimited, time-limited, analytic, causal, symmetric etc. In particular, we shall see in Subsection 3.3 that a specific bandlimitation constraint is important in the case of the WD, smoothed WD versions, and AF1 since it avoids aliasing in these representations and at the same time avoids the relative phase ambiguity which would occur in the case of global signal synthesis. Furthermore, the signal subspace $\mathcal{S}$ can be constructed such that it possesses a time-frequency concentration property [30,31]; this results in a "time-frequency selective" type of signal synthesis which effectively combines signal synthesis with joint time-frequency filtering (see Subsection 3.4).

2.1 Subspace Unitarity and Induced Subspaces

We shall now develop the concepts of subspace unitarity and induced spaces as a theoretical background for subspace signal synthesis.

Subspace unitarity. A BTFR T is said to be *unitary on the signal space* $\mathcal{S}$ [13] if it satisfies Moyal's formula [18,9]

$$(T_{x_1,y_1}, T_{x_2,y_2}) = (x_1,x_2)(y_1,y_2)^* \tag{2.3}$$

for arbitrary signals $x_1(n)$, $x_2(n)$, $y_1(n)$, $y_2(n) \in \mathcal{S}$ which are elements of the signal space $\mathcal{S}$. Here, the signal-domain and BTFR-domain inner products are defined as

$$(x_1,x_2) = \sum_n x_1(n)\, x_2^*(n) \qquad \text{and} \qquad (T_1,T_2) = \sum_n \int_{-1/2}^{1/2} T_1(n,\Theta)\, T_2^*(n,\Theta)\, d\Theta \; .$$

A BTFR is said to be *globally unitary* if Moyal's formula holds on the total signal space $l_2(-\infty,\infty)$. If a BTFR is globally unitary, then it is obviously unitary on arbitrary subspaces $\mathcal{S} \subset l_2(-\infty,\infty)$. Among the BTFRs presented in the Introduction, only WD', RD, and AF2 are globally unitary.

The BTFR's impulse response $u_T(n,\Theta;k,l)$ can be used to check whether or not the BTFR is unitary on a given space $\mathcal{S}$. Indeed, it can be shown [13] that a necessary and sufficient condition for a BTFR's unitarity on $\mathcal{S}$ is

$$\sum_n \int_{-1/2}^{1/2} u_{T\mathcal{S}}^*(n,\Theta;k_1,l_1)\, u_{T\mathcal{S}}(n,\Theta;k_2,l_2)\, d\Theta = z_{\mathcal{S}}(k_1,l_1;k_2,l_2) \; . \tag{2.4}$$

Here,

$$u_{T\mathcal{S}}(n,\Theta;k,l) \triangleq \sum_{k'} \sum_{l'} u_T(n,\Theta;k',l')\, z_{\mathcal{S}}(k',l';k,l) \tag{2.5}$$

with

$$z_{\mathcal{S}}(k_1,l_1;k_2,l_2) \triangleq P_{\mathcal{S}}(k_1,k_2)\, P_{\mathcal{S}}^*(l_1,l_2) \; , \tag{2.6}$$

where $P_{\mathcal{S}}(n_1,n_2)$, the kernel of the orthogonal projection operator [27,28] of $\mathcal{S}$, is

derived from the basis signals $e_k(n)$ spanning $\mathcal{S}$ according to

$$P_{\mathcal{S}}(n_1,n_2) = \sum_{k=1}^{N} e_k(n_1)\, e_k^*(n_2) \ . \tag{2.7}$$

Specializing these results to $\mathcal{S}=l_2(-\infty,\infty)$ and the orthonormal basis of $l_2(-\infty,\infty)$ given by $e_k(n) = \delta(n-k)$, $-\infty < k < \infty$, a necessary and sufficient condition for *global unitarity* is easily found as

$$\sum_{n} \int_{-1/2}^{1/2} u_T^*(n,\Theta;k_1,l_1)\, u_T(n,\Theta;k_2,l_2)\, d\Theta = \delta(k_1-k_2)\, \delta(l_1-l_2) \ . \tag{2.8}$$

Induced spaces. For a given BTFR T, the linear signal space $\mathcal{S}$ "induces" a corresponding linear space $\mathcal{S}_T$ of two-dimensional functions $\tilde{T}(n,\Theta)$ which are 1-periodic with respect to the frequency variable Θ [9,13]. We shall call $\mathcal{S}_T$ the *induced T-domain space* associated with the signal space $\mathcal{S}$. Loosely speaking, the induced T-domain space $\mathcal{S}_T$ consists of all linear combinations of BTFR outcomes $T_{x,y}(n,\Theta)$ with $x(n), y(n) \in \mathcal{S}$. The spaces $\mathcal{S}$ and $\mathcal{S}_T$ are associated in the sense that

$$x(n),\ y(n) \in \mathcal{S} \quad \Rightarrow \quad T_{x,y}(n,\Theta) \in \mathcal{S}_T \ ,$$

i.e., if the signals x(n) and y(n) are elements of the signal space $\mathcal{S}$, then their BTFR outcome $T_{x,y}(n,\Theta)$ is an element of the induced T-domain space $\mathcal{S}_T$. Note, however, that $\mathcal{S}_T$ contains not only valid BTFR outcomes $T_{x,y}(n,\Theta)$ but also functions $\tilde{T}(n,\Theta)$ which are not BTFR outcomes.

If the signal space $\mathcal{S}$ is spanned by orthonormal basis signals $e_k(n)$ (k=1,2,..,N), and if the BTFR T is *unitary on* $\mathcal{S}$, then the induced T-domain space $\mathcal{S}_T$ is spanned by an "induced T-domain basis" $T_{kl}(n,\Theta)$ (k,l=1,2,..,N), where the induced basis functions are simply the cross-BTFR outcomes of the basis signals $e_k(n)$ spanning $\mathcal{S}$,

$$T_{kl}(n,\Theta) = T_{e_k,e_l}(n,\Theta) \ . \tag{2.9}$$

The induced basis functions $T_{kl}(n,\Theta)$ are again orthonormal. In other words, the cross-BTFRs of the orthonormal basis signals spanning $\mathcal{S}$ form a complete orthonormal basis of the induced T-domain space $\mathcal{S}_T$. For this, the unitarity of the BTFR T on $\mathcal{S}$ is an essential requirement [9].

We shall now prove some of the statements made above. Let x(n) and y(n) be two elements of a signal space $\mathcal{S}$, and let $e_k(n)$ be a complete orthonormal basis of $\mathcal{S}$. Then x(n) and y(n) can be represented in terms of the basis $e_k(n)$ as

$$x(n) = \sum_{k} \alpha_k\, e_k(n) \ , \qquad y(n) = \sum_{l} \beta_l\, e_l(n) \tag{2.10}$$

with coefficients α_k and β_l. Inserting (2.10) into the BTFR expression (2.1) yields

$$T_{x,y}(n,\Theta) = \sum_{k}\sum_{l} \alpha_k \beta_l^*\, T_{e_k,e_l}(n,\Theta) \ . \tag{2.11}$$

Since the BTFR T is assumed to be unitary on $\mathcal{G}$, and the basis signals $e_k(n)$ are themselves elements of $\mathcal{G}$, Moyal's formula (2.3) holds for the basis signals $e_k(n)$:

$$(T_{e_{k_1},e_{l_1}}, T_{e_{k_2},e_{l_2}}) = (e_{k_1}, e_{k_2})(e_{l_1}, e_{l_2})^* = \delta(k_1-k_2)\,\delta(l_1-l_2)\,, \qquad (2.12)$$

where the orthonormality of the $e_k(n)$ has been used. Eq. (2.12) shows that the BTFR outcomes $T_{e_k,e_l}(n,\Theta)$ are orthonormal; the functions $T_{e_k,e_l}(n,\Theta)$ are thus an orthonormal basis of a space $\mathcal{G}_T$. Finally, (2.11) shows that $T_{x,y}(n,\Theta)$ can be represented in terms of the basis $T_{e_k,e_l}(n,\Theta)$ spanning $\mathcal{G}_T$; hence, $T_{x,y}(n,\Theta)$ must be an element of $\mathcal{G}_T$. This shows that $T_{x,y} \in \mathcal{G}_T$ for $x,y \in \mathcal{G}$.

2.2 A General Algorithm for Subspace-Constrained Signal Synthesis

Using the concept of the induced T-domain space and induced T-domain basis, a solution to the subspace signal synthesis problem (2.2),

$$\varepsilon_x = \|\hat{T}-T_x\| \longrightarrow \min_{x\in\mathcal{G}}\,,$$

can now be found rather easily [13]. *In the following, we assume that the BTFR T is unitary on the space $\mathcal{G}$ on which signal synthesis is to be performed.*

We first take account of the signal space constraint $x(n) \in \mathcal{G}$ by representing $x(n)$ in terms of an orthonormal basis $e_k(n)$ spanning $\mathcal{G}$,

$$x(n) = \sum_{k=1}^{N} \alpha_k\, e_k(n)\,;$$

the BTFR outcome $T_x(n,\Theta)$ is then an element of the induced T-domain space $\mathcal{G}_T$ and can hence be represented in terms of the induced orthonormal basis $T_{kl}(n,\Theta) = T_{e_k,e_l}(n,\Theta)$ as

$$T_x(n,\Theta) = \sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_k\alpha_l^*\, T_{kl}(n,\Theta)\,. \qquad (2.13)$$

The model $\hat{T}(n,\Theta)$, on the other hand, need not be an element of $\mathcal{G}_T$; however, it can be decomposed as [28]

$$\hat{T}(n,\Theta) = \hat{T}_{\mathcal{G}}(n,\Theta) + \hat{T}_{\perp}(n,\Theta)\,,$$

where $\hat{T}_{\mathcal{G}}(n,\Theta)$ is the orthogonal projection of the model $\hat{T}(n,\Theta)$ on $\mathcal{G}_T$ (and thus an element of $\mathcal{G}_T$), and $\hat{T}_{\perp}(n,\Theta)$ is orthogonal on $\mathcal{G}_T$. The (squared) synthesis error ε_x^2 can then be decomposed as

$$\varepsilon_x^2 = \|\hat{T}-T_x\|^2 = \|(\hat{T}_{\mathcal{G}}+\hat{T}_{\perp})-T_x\|^2 = \|(\hat{T}_{\mathcal{G}}-T_x)+\hat{T}_{\perp}\|^2 =$$

$$= \|\hat{T}_{\mathcal{G}}-T_x\|^2 + \|\hat{T}_{\perp}\|^2 = \varepsilon_{\mathcal{G},x}^2 + \varepsilon_{\perp}^2$$

where we have applied the Pythagorean Theorem [28,29] to the orthogonal "vectors" $(\tilde{T}_{\mathcal{C}}-T_x) \in \mathcal{C}_T$ and $\tilde{T}_\perp \perp \mathcal{C}_T$ (see *Figure 1*). Since the "orthogonal" error component $\varepsilon_\perp = \| \tilde{T}_\perp \|$ does not depend on the signal x(n), it can be disregarded for minimization. Note that $\varepsilon_\perp$ is altogether zero if the model $\tilde{T}(n,\Theta)$ is an element of $\mathcal{C}_T$.

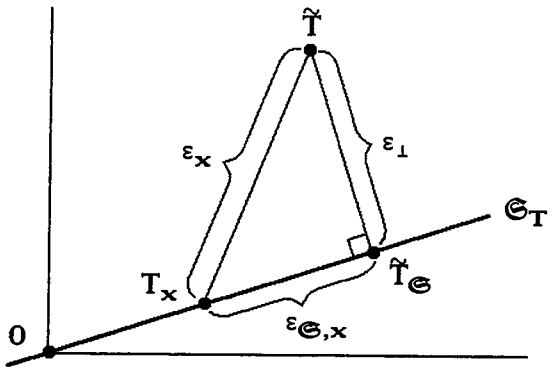

Figure 1. Decomposition of model and synthesis error.

The error component $\varepsilon_{\mathcal{C},x} = \| \tilde{T}_{\mathcal{C}}-T_x \|$ has now to be minimized. Since the model projection $\tilde{T}_{\mathcal{C}}(n,\Theta)$ is an element of the induced T-domain space $\mathcal{C}_T$, it can be represented in terms of the induced T-domain basis $T_{kl}(n,\Theta)$,

$$\tilde{T}_{\mathcal{C}}(n,\Theta) = \sum_{k=1}^{N} \sum_{l=1}^{N} \gamma_{kl}\, T_{kl}(n,\Theta) \qquad \text{with} \qquad \gamma_{kl} = (\tilde{T}, T_{kl}) \quad . \tag{2.14}$$

Inserting the expansions (2.13) and (2.14), and using the orthonormality of the induced T-domain basis $T_{kl}(n,\Theta)$, the (squared) synthesis error can be developed as

$$\varepsilon^2_{\mathcal{C},x} = \| \tilde{T}_{\mathcal{C}}-T_x \|^2 = \left\| \sum_{k=1}^{N} \sum_{l=1}^{N} (\gamma_{kl} - \alpha_k \alpha_l^*)\, T_{kl} \right\|^2 =$$

$$= \sum_{k=1}^{N} \sum_{l=1}^{N} |\gamma_{kl} - \alpha_k \alpha_l^*|^2 = \| \boldsymbol{\Gamma} - \boldsymbol{\alpha}\boldsymbol{\alpha}^+ \|^2 = \varepsilon^2_{\mathcal{C},\alpha} \tag{2.15}$$

with the coefficient matrix $\boldsymbol{\Gamma}=(\gamma_{kl})$ of size NxN and the N-dimensional coefficient vector $\boldsymbol{\alpha} = (\alpha_k)$; $\|\cdot\|$ denotes the Euclidean matrix norm (Frobenius norm), and "+" stands for complex conjugate transposition.

With (2.15), the synthesis error is now formulated in a *coefficient domain*. The dyadic-product matrix $\boldsymbol{\alpha}\boldsymbol{\alpha}^+$ is Hermitian and rank-1; the matrix $\boldsymbol{\Gamma}$, on the other hand, is generally not Hermitian but can be split into a Hermitian component $\boldsymbol{\Gamma}_H$ and an anti-Hermitian component $\boldsymbol{\Gamma}_A$. With this, it can be shown that the squared synthesis error can be further decomposed as

$$\varepsilon^2_{\mathcal{C},\alpha} = \| \boldsymbol{\Gamma} - \boldsymbol{\alpha}\boldsymbol{\alpha}^+ \|^2 = \| (\boldsymbol{\Gamma}_H + \boldsymbol{\Gamma}_A) - \boldsymbol{\alpha}\boldsymbol{\alpha}^+ \|^2 = \| (\boldsymbol{\Gamma}_H - \boldsymbol{\alpha}\boldsymbol{\alpha}^+) + \boldsymbol{\Gamma}_A \|^2 =$$

$$= \| \boldsymbol{\Gamma}_H - \boldsymbol{\alpha}\boldsymbol{\alpha}^+ \|^2 + \| \boldsymbol{\Gamma}_A \|^2 = \varepsilon^2_{\mathcal{C}H,\alpha} + \varepsilon^2_{\mathcal{C}A} \ ,$$

where the "anti-Hermitian" error component $\varepsilon_{\mathcal{E}A}=\|\boldsymbol{\Gamma}_A\|$ does not depend on $\boldsymbol{\alpha}$ and can hence be disregarded for minimization. Thus, there remains to minimize the "Hermitian" error component

$$\varepsilon_{\mathcal{E}H,\alpha} = \|\boldsymbol{\Gamma}_H - \boldsymbol{\alpha}\boldsymbol{\alpha}^+\| \ . \tag{2.16}$$

This minimization amounts to the approximation of the Hermitian matrix $\boldsymbol{\Gamma}_H$ by a dyadic product $\boldsymbol{\alpha}\boldsymbol{\alpha}^+$. Setting the gradient of $\varepsilon^2_{\mathcal{E}H,\alpha}$ with respect to the coefficient vector $\boldsymbol{\alpha}$ equal to zero, we obtain the following necessary condition for a minimum:

$$\boldsymbol{\Gamma}_H \boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|^2 \boldsymbol{\alpha} \ . \tag{2.17}$$

This is an eigenvalue-eigenvector equation, and we conclude that the $\boldsymbol{\alpha}$ minimizing $\varepsilon_{\mathcal{E}H,\alpha}$ must be some eigenvector of the matrix $\boldsymbol{\Gamma}_H$, scaled such that $\|\boldsymbol{\alpha}\|^2$ equals the corresponding eigenvalue. We note that the eigenvalues and eigenvectors of $\boldsymbol{\Gamma}_H$ are real-valued and orthogonal, respectively since the matrix $\boldsymbol{\Gamma}_H$ is Hermitian. With λ_k and $\mathbf{u}_k$ denoting the eigenvalues and the corresponding normalized eigenvectors of $\boldsymbol{\Gamma}_H$, respectively, the solution to (2.17) can be written as

$$\boldsymbol{\alpha} = e^{j\varphi}\sqrt{\lambda_k}\,\mathbf{u}_k \ , \tag{2.18}$$

where the phase ambiguity of eigenvectors has been explicitly incorporated via the unknown phase constant φ.

We finally have to specify *which* eigenvector $\mathbf{u}_k$ has to be taken in (2.18) in order to achieve the minimum of $\varepsilon_{\mathcal{E}H,\alpha}$. Inserting (2.18) and the spectral decomposition

$$\boldsymbol{\Gamma}_H = \sum_{l=1}^{N} \lambda_l \mathbf{u}_l \mathbf{u}_l^+$$

of the Hermitian matrix $\boldsymbol{\Gamma}_H$ into (2.16), the residual synthesis error $\varepsilon_{\mathcal{E}H,\alpha}$ is derived as

$$\varepsilon^2_{\mathcal{E}H,\alpha} = \left\| \sum_{l=1}^{N} \lambda_l \mathbf{u}_l \mathbf{u}_l^+ - \lambda_k \mathbf{u}_k \mathbf{u}_k^+ \right\|^2 = \sum_{l=1}^{N} \lambda_l^2 - \lambda_k^2 \ , \tag{2.19}$$

where the orthonormality of the eigenvectors $\mathbf{u}_l$ has been used. From (2.19), it appears that the index k minimizing $\varepsilon^2_{\mathcal{E}H,\alpha}$ has to be chosen such that λ_k is the eigenvalue with maximum magnitude. However, (2.17) requires $\|\boldsymbol{\alpha}\|^2 = \lambda_k$, so that λ_k is constrained to be nonnegative. Thus, we conclude that the optimal λ_k is the largest positive eigenvalue. If the eigenvalues λ_k and the corresponding eigenvectors $\mathbf{u}_k$ of $\boldsymbol{\Gamma}_H$ are arranged in decreasing order, $\lambda_1 \geq \lambda_2 \geq \ldots$, and if the largest eigenvalue λ_1 is nonnegative, then the optimal coefficient vector $\boldsymbol{\alpha}_{opt}$ is finally obtained as

$$\boldsymbol{\alpha}_{opt} = e^{j\varphi}\sqrt{\lambda_1}\,\mathbf{u}_1 \ , \qquad \lambda_1 \geq 0 \ . \tag{2.20}$$

Algorithm summary. As a result of the above derivation, we have obtained a general algorithm for subspace signal synthesis. This algorithm is applicable to BTFRs which are unitary on the signal subspace on which signal synthesis is performed. It can be summarized as follows:

1a) Calculate the expansion coefficients of the model projection $\tilde{T}_{\mathcal{G}}(n,\Theta)$ according to

$$\gamma_{kl} = (\tilde{T}, T_{kl}) = \sum_n \int_{-1/2}^{1/2} \tilde{T}(n,\Theta)\, T_{kl}^*(n,\Theta)\, d\Theta , \qquad 1 \le k,l \le N .$$

1b) Form the matrix $\mathbf{\Gamma} = (\gamma_{kl})$ and take its Hermitian component

$$\mathbf{\Gamma}_H = \tfrac{1}{2}(\mathbf{\Gamma} + \mathbf{\Gamma}^+) .$$

2) Calculate the largest eigenvalue λ_1 and the associated (normalized) eigenvector $\mathbf{u}_1$ of $\mathbf{\Gamma}_H$.

3) If $\lambda_1 \ge 0$, the synthesis solution is given by

$$x_{opt}(n) = \sum_{k=1}^{N} \alpha_{opt,k}\, e_k(n) \tag{2.21}$$

with

$$\boldsymbol{\alpha}_{opt} = e^{j\varphi} \sqrt{\lambda_1}\, \mathbf{u}_1 , \tag{2.22}$$

where φ is an arbitrary phase constant.

Global signal synthesis. We have stated before that global (i.e., unconstrained) signal synthesis can be considered a special case of signal synthesis with a signal space constraint $x(n) \in \mathcal{G}$ if we choose the space $\mathcal{G}$ to be the entire space of finite-energy signals, $\mathcal{G} = l_2(-\infty,\infty)$. From the identity

$$x(n) = \sum_k x(k)\, \delta(n-k) , \qquad x(n) \in l_2(-\infty,\infty) ,$$

it follows that $e_k(n) = \delta(n-k)$ is an (orthonormal) basis of $l_2(-\infty,\infty)$; the expansion coefficients α_k are equal to the signal samples $x(k)$. If we assume the BTFR T to be *globally* unitary, i.e., unitary on $l_2(-\infty,\infty)$, then the induced T-domain space $[l_2(-\infty,\infty)]_T$ is spanned by the orthonormal basis

$$T_{kl}(n,\Theta) = T_{x,y}(n,\Theta)\,\big|_{x(n)=\delta(n-k),\ y(n)=\delta(n-l)} = u_T(n,\Theta;k,l) ,$$

where (1.3) has been used. Specializing the general results summarized above, we obtain the following algorithm for global signal synthesis in the case of a globally unitary BTFR:

1a) Calculate the expansion coefficients of the model projection $\tilde{T}_{l_2(-\infty,\infty)}(n,\Theta)$ according to

$$\gamma_{kl} = (\tilde{T}, T_{kl}) = \sum_n \int_{-1/2}^{1/2} \tilde{T}(n,\Theta)\, u_T^*(n,\Theta;k,l)\, d\Theta\ , \qquad -\infty \le k,l \le \infty\ .$$

1b) Form the matrix $\mathbf{\Gamma} = (\gamma_{kl})$ and take its Hermitian component

$$\mathbf{\Gamma}_H = \tfrac{1}{2}(\mathbf{\Gamma} + \mathbf{\Gamma}^+)\ .$$

2) Calculate the largest eigenvalue λ_1 and the associated (normalized) eigenvector $\mathbf{u}_1$ of $\mathbf{\Gamma}_H$.

3) If $\lambda_1 \ge 0$, the synthesis solution is given by

$$x_{opt}(n) = e^{j\varphi} \sqrt{\lambda_1}\, u_{1,n} \tag{2.23}$$

where $u_{1,n}$ is the n-th component of the vector $\mathbf{u}_1$ and φ is an arbitrary phase constant.

2.3 The Power Algorithm

The general signal synthesis method derived in the previous section requires the computation of the largest (nonnegative) eigenvalue λ_1 and the corresponding (normalized) eigenvector $\mathbf{u}_1$ of a Hermitian matrix $\mathbf{\Gamma}_H$. This can be done by means of a simple and efficient iterative method known as the *power algorithm* [32]. We here briefly state the power algorithm for two reasons: firstly, it is practically useful, and secondly, it will provide the starting-point for developing iterative signal synthesis algorithms for smoothed WD versions and the spectrogram (see Sections 4-6).

In the following, we assume that $\mathbf{\Gamma}_H$ is a Hermitian matrix whose largest eigenvalue λ_1 equals the eigenvalue with maximum magnitude. We further assume that λ_1 has multiplicity 1. These requirements are usually satisfied in the context of signal synthesis. The power algorithm is then given by the following iteration scheme: For i=1,2,...

(a) $\mathbf{v}^{(i)} = \mathbf{\Gamma}_H\, \mathbf{u}^{(i-1)}$

(b) $\lambda^{(i)} = \| \mathbf{v}^{(i)} \|$

(c) $\mathbf{u}^{(i)} = \mathbf{v}^{(i)} / \lambda^{(i)}\ .$

This iteration is initialized by a largely arbitrary start vector $\mathbf{u}^{(0)}$. Indeed, $\lambda^{(i)}$ and $\mathbf{u}^{(i)}$ are guaranteed to converge to the largest eigenvalue λ_1 and the corresponding (normalized) eigenvector $\mathbf{u}_1$, respectively, unless the start vector $\mathbf{u}^{(0)}$ happens to be orthogonal on $\mathbf{u}_1$. The convergence speed of the power algorithm is governed by the ratio $|\lambda'_2|/\lambda_1$; here, λ'_2 denotes the eigenvalue with second-largest *magnitude* (i.e., λ'_2 may be negative). The convergence will be faster for smaller $|\lambda'_2|/\lambda_1$. Of

course, in practice the iteration must be terminated after a finite number of iteration steps. An appropriate termination criterion is $\|\mathbf{u}^{(i)}-\mathbf{u}^{(i-1)}\| < \varepsilon$, where ε is a suitably chosen small constant.

3. WIGNER DISTRIBUTION

This section discusses the signal synthesis problem for the discrete-time WD [18]

$$WD_{x,y}(n,\Theta) = 2 \sum_{m} x(n+m)\, y^*(n-m)\, e^{-j4\pi\Theta m} . \tag{3.1}$$

Since the general algorithm derived in the previous section requires the BTFR to be unitary on the space $\mathcal{E}$ on which signal synthesis is performed, we first investigate the unitarity properties of the WD in Subsection 3.1. It turns out that the WD is not globally unitary, but it is unitary on certain subspaces $\mathcal{H}$ of "halfband signals." Furthermore, the WD is shown to be unitary (up to a factor) on two subspaces $\mathcal{E}$ and $\mathcal{O}$ corresponding to the even-indexed and odd-indexed signal samples, respectively.

In Subsection 3.2, we consider the case of global signal synthesis. Despite the WD's global non-unitarity which prohibits the application of the general algorithm of Subsection 2.2, an algorithm for global signal synthesis exists [10] and is easily derived. However, the synthesized signal is seen to suffer from an awkward "relative" phase ambiguity which generally leads to severe signal distortion. We also give a subspace interpretation of global signal synthesis: global signal synthesis is shown to be equivalent to two separate *subspace* signal synthesis procedures on the subspaces $\mathcal{E}$ and $\mathcal{O}$ of even-indexed and odd-indexed signal samples, respectively.

Arguing that WD aliasing should generally be avoided, we consider signal synthesis on a halfband subspace $\mathcal{H}$ in Subsection 3.3. It is shown that the "halfband constraint" $x(n) \in \mathcal{H}$ prevents WD aliasing, assures the WD's unitarity, and avoids the troublesome relative phase ambiguity of the synthesis result. Since the WD is unitary on $\mathcal{H}$, an algorithm for halfband signal synthesis is easily derived by specialization of the general signal synthesis algorithm of Subsection 2.2.

Other approaches to WD signal synthesis and simulation results are finally presented in Subsections 3.4 and 3.5, respectively.

3.1 Unitarity Properties

From the WD's definition (3.1), it follows that the WD is 1/2-periodic with respect to the normalized-frequency variable Θ,

$$WD_{x,y}(n,\Theta+1/2) = WD_{x,y}(n,\Theta) .$$

This is inconsistent with the 1-periodic Fourier transform $X(\Theta)$ of a signal $x(n)$. In general, the WD will hence be aliased with respect to Θ. More precisely, it is well known [18] that the auto-WD $WD_x(n,\Theta)$ is non-aliased if and only if the signal

x(n) is a *halfband signal*, i.e., if its Fourier transform is restricted to some frequency interval with bandwidth 1/2, which is just one half of the spectral period 1. Halfband signals are thus effectively oversampled by a factor of 2.

Using (1.3), the WD's impulse response is obtained as

$$u_{WD}(n,\Theta;k,l) = \delta(n-\tfrac{k+l}{2})\, e^{-j2\pi(k-l)\Theta}\,[1+(-1)^{k+l}]\ .$$

Checking condition (2.8), it is easily shown that the WD is *not* globally unitary,

$$\sum_n \int_{-1/4}^{1/4} u^*_{WD}(n,\Theta;k_1,l_1)\, u_{WD}(n,\Theta;k_2,l_2)\, d\Theta \neq \delta(k_1-k_2)\,\delta(l_1-l_2)$$

(note that the integration is here over a frequency interval of length 1/2 only, according to the WD's 1/2-periodicity). Indeed, it has been shown in [18] that Moyal's formula (2.3) is not, in general, satisfied by the discrete-time WD. Also, the WD's global non-unitarity has been implicitly noted in [33] where it is shown that an orthonormal signal basis does not induce an orthonormal WD-domain basis. However, we will see in the following that *the WD is unitary for halfband signals.*

Halfband subspaces. We shall call x(n) a *halfband signal* if it is bandlimited in a frequency band $\Theta_0-1/4 \le \Theta < \Theta_0+1/4$ with bandwidth 1/2, i.e., if

$$X(\Theta) = 0 \qquad \text{for} \qquad \Theta_0+\tfrac{1}{4} \le \Theta < \Theta_0+\tfrac{3}{4}\ . \tag{3.2}$$

The parameter Θ_0 is the center frequency of the halfband. Two important special cases are given by $\Theta_0=0$ (this corresponds to the lowpass band $-1/4 \le \Theta < 1/4$ and, in particular, to real-valued signals oversampled by a factor of 2), and $\Theta_0=1/4$ (this corresponds to the one-sided band $0 \le \Theta < 1/2$, i.e., to analytic signals). If Θ_0 is assumed fixed, then the collection of all halfband signals satisfying (3.2) constitute a linear signal subspace which we shall call a *halfband subspace* $\mathcal{H}$. The orthogonal projection [27-29] $x_{\mathcal{H}}(n)$ of a signal x(n) on $\mathcal{H}$ is given by

$$x_{\mathcal{H}}(n) = \sum_{n'} h(n-n')\, x(n')\ , \tag{3.3}$$

and an orthonormal basis $\{h_k(n)\}$ of $\mathcal{H}$ is

$$h_k(n) = \sqrt{2}\, h(n-2k)\ , \qquad -\infty < k < \infty\ . \tag{3.4}$$

Here,

$$h(n) = \tfrac{1}{2}\,\mathrm{sinc}\left(\tfrac{1}{2}n\right) e^{j2\pi\Theta_0 n} \qquad \longleftrightarrow \qquad H(\Theta) = \begin{cases} 1\ , & \Theta_0-1/4 \le \Theta < \Theta_0+1/4 \\ 0\ , & \Theta_0+1/4 \le \Theta < \Theta_0+3/4 \end{cases}, \tag{3.5}$$

with $\mathrm{sinc}(\alpha) = \sin(\pi\alpha)/(\pi\alpha)$, are the impulse response and frequency response, respectively, of an ideal "halfband filter" with passband $\Theta_0-1/4 \le \Theta < \Theta_0+1/4$. Any halfband signal $x(n)\in\mathcal{H}$ can be represented in terms of the basis $h_k(n)$,

$$x(n) = \sum_k \alpha_k h_k(n) , \qquad x(n) \in \mathcal{H} , \tag{3.6}$$

where the coefficients $\alpha_k=(x,h_k)$ can be shown to be (up to a factor) the even-indexed signal samples,

$$\alpha_k = \sqrt{2}\, x(2k) . \tag{3.7}$$

With this, (3.6) becomes the interpolation relation

$$x(n) = 2 \sum_k h(n-2k)\, x(2k) , \qquad x(n) \in \mathcal{H} . \tag{3.8}$$

It has been shown in [18] that Moyal's formula (2.3) is satisfied for halfband signals; we thus conclude that *the WD is unitary on halfband subspaces* $\mathcal{H}$. In particular, for a given halfband subspace $\mathcal{H}$, the *induced WD-domain subspace* $\mathcal{H}_{WD}$ is spanned by an induced orthonormal basis $WD_{kl}(n,\Theta)$ which, using (2.9), is obtained as

$$WD_{kl}(n,\Theta) = WD_{h_k,h_l}(n,\Theta) = 2\, WD_h(n-(k+l),\Theta)\, e^{-j4\pi\Theta(k-l)} \tag{3.9}$$

with

$$WD_h(n,\Theta) = 4\left(\tfrac{1}{4}-|\Theta-\Theta_0|\right) \operatorname{sinc}\left[4\left(\tfrac{1}{4}-|\Theta-\Theta_0|\right) n\right] , \qquad \Theta_0-\tfrac{1}{4} \le \Theta < \Theta_0+\tfrac{1}{4} . \tag{3.10}$$

Summarizing, we can say that the WD is aliased and non-unitary on the entire signal space $l_2(-\infty,\infty)$, but it is non-aliased and unitary on a halfband subspace $\mathcal{H}$.

The subspaces $\mathcal{E}$ and $\mathcal{O}$. Apart from the halfband subspaces $\mathcal{H}$, there exist two other signal subspaces on which the WD is essentially unitary. These subspaces are associated with the subsequences of even-indexed and odd-indexed signal samples, respectively.

By definition, the "even-index" subspace $\mathcal{E}$ consists of all signals in which only the even-indexed signal samples are nonzero,

$$x(n) \in \mathcal{E} \quad \Leftrightarrow \quad x(2k+1) = 0 .$$

The orthogonal projection of an arbitrary signal x(n) on $\mathcal{E}$ is obtained by setting the odd-indexed signal samples equal to zero,

$$x_{\mathcal{E}}(n) = \begin{cases} x(n), & n=2k \\ 0 , & n=2k+1 \end{cases} .$$

We note, for later use, that the Fourier transform of $x_{\mathcal{E}}(n)$ can be shown to be an aliased version of the Fourier transform of x(n),

$$X_{\mathcal{E}}(\Theta) = \tfrac{1}{2}[\, X(\Theta)+X(\Theta-1/2)\,] . \tag{3.11}$$

Any signal $x(n) \in \mathfrak{E}$ can obviously be written as

$$x(n) = \sum_k x(2k)\, \delta(n-2k), \qquad x(n) \in \mathfrak{E};$$

from this, we conclude that the subspace $\mathfrak{E}$ is spanned by the (orthonormal) basis

$$e_k(n) = \delta(n-2k),$$

and that the expansion coefficients α_k are simply the even-indexed signal samples $x(2k)$.

Working out (2.4)-(2.7), it is readily shown that the condition (2.4) for subspace unitarity (with frequency integration length 1/2 according to the WD's 1/2-periodicity) is satisfied apart form a factor of 2 which merely effects a scaling. Thus, the induced WD-domain subspace $\mathfrak{E}_{WD}$ is spanned by the basis

$$WD_{kl}(n,\Theta) = WD_{e_k,e_l}(n,\Theta) = 2\,\delta\big(n-(k+l)\big)\, e^{-j4\pi\Theta(k-l)} \tag{3.12}$$

which is orthonormal apart from a factor of 2, $(WD_{kl}, WD_{k'l'}) = 2\delta(k-k')\delta(l-l')$.

We note that completely analogous results are obtained for the "odd-index" signal subspace $\mathfrak{O}$ defined by $x(n) \in \mathfrak{O} \Leftrightarrow x(2k) = 0$.

3.2 Global Signal Synthesis

In the previous subsection, it has been shown that the discrete-time WD is not globally unitary. Therefore, the global (unconstrained) signal synthesis problem

$$\varepsilon_x = \| \tilde{\Upsilon} - WD_x \| \longrightarrow \min_x \tag{3.13}$$

cannot be solved by direct application of the general method of Subsection 2.2. Nevertheless, the solution is easily derived [10]. In the following, we assume that the model $\tilde{\Upsilon}(n,\Theta)$ is real-valued and 1/2-periodic with respect to the frequency variable Θ just as the auto-WD $WD_x(n,\Theta)$, and that the frequency integration length in the synthesis error (3.13) is 1/2, consistent with the 1/2-periodicity of $WD_x(n,\Theta)$.

We start by writing the WD (3.1) as

$$WD_{x,y}(n,\Theta) = 2 \sum_m c_{x,y}(n,m)\, e^{-j4\pi\Theta m}, \tag{3.14}$$

where

$$c_{x,y}(n,m) \triangleq x(n+m)\, y^*(n-m) = q_{x,y}(n+m,n-m)$$

is derived from the outer signal product $q_{x,y}(k,l)$ by the coordinate transform $k=n+m$, $l=n-m$. Analogously, the model $\tilde{\Upsilon}(n,\Theta)$ can be written as

$$\tilde{\Upsilon}(n,\Theta) = 2 \sum_m \tilde{c}(n,m)\, e^{-j4\pi\Theta m}$$

where $\tilde{c}(n,m)$ is given by

$$\tilde{c}(n,m) = \frac{1}{2} \int_{-1/2}^{1/2} \tilde{T}(n,\frac{\Theta}{2})\, e^{j2\pi m\Theta}\, d\Theta \ . \tag{3.15}$$

Using Parseval's relation, the (squared) synthesis error norm can then be reformulated as

$$\varepsilon_x^2 = \| \tilde{T} - WD_x \|^2 = 2 \| \tilde{c} - c_x \|^2 = 2 \sum_n \sum_m | \tilde{c}(n,m) - x(n+m)\, x^*(n-m) |^2 \ .$$

Separating even-indexed and odd-indexed signal samples

$$x_e(k) \triangleq x(2k) \ , \qquad x_o(k) \triangleq x(2k+1) \ ,$$

the squared synthesis error can finally be expressed as

$$\varepsilon_x^2 = \varepsilon_{e,x_e}^2 + \varepsilon_{o,x_o}^2 \tag{3.16}$$

where

$$\varepsilon_{e,x_e}^2 = 2 \| \tilde{q}_e - q_{x_e} \|^2 = 2 \sum_k \sum_l | \tilde{q}_e(k,l) - x_e(k)\, x_e^*(l) |^2 \tag{3.17}$$

$$\varepsilon_{o,x_o}^2 = 2 \| \tilde{q}_o - q_{x_o} \|^2 = 2 \sum_k \sum_l | \tilde{q}_o(k,l) - x_o(k)\, x_o^*(l) |^2 \tag{3.18}$$

with

$$\tilde{q}_e(k,l) \triangleq \tilde{c}(k+l, k-l) \ , \qquad \tilde{q}_o(k,l) \triangleq \tilde{c}(k+l+1, k-l) \ . \tag{3.19}$$

According to (3.16), the synthesis error is decomposed into two error components involving each only the even-indexed and only the odd-indexed signal samples, respectively. Hence, even-indexed and odd-indexed signal samples can be synthesized separately by solving the two *independent* minimization problems $\varepsilon_{e,x_e} \rightarrow \min$ and $\varepsilon_{o,x_o} \rightarrow \min$ which yield $x_e(n)$ and $x_o(n)$, respectively. The error components (3.17) and (3.18) can be written in obvious matrix-vector notation as

$$\varepsilon_{e,x_e}^2 = 2 \| \mathbf{Q}_e - \mathbf{x}_e \mathbf{x}_e^+ \|^2 \ , \qquad \varepsilon_{o,x_o}^2 = 2 \| \mathbf{Q}_o - \mathbf{x}_o \mathbf{x}_o^+ \|^2 \ .$$

The square matrices $\mathbf{Q}_e$ and $\mathbf{Q}_o$ are Hermitian provided that the model $\tilde{T}(n,\Theta)$ is real-valued. Hence, each minimization is recognized as a rank-1 approximation of a Hermitian matrix, which is analogous to the final approximation problem considered in Subsection 2.2 (cf. (2.16)). Reasoning as in Subsection 2.2, the eigen-equations (cf. (2.17))

$$\mathbf{Q}_e \mathbf{x}_e = \|\mathbf{x}_e\|^2 \mathbf{x}_e \ , \qquad \mathbf{Q}_o \mathbf{x}_o = \|\mathbf{x}_o\|^2 \mathbf{x}_o \tag{3.20}$$

are obtained as necessary conditions for the optimal solutions. The solutions are then derived as (cf. (2.20))

$$\mathbf{x}_{e,opt} = e^{j\varphi_e}\sqrt{\lambda_{e1}}\,\mathbf{u}_{e1}\ , \qquad \mathbf{x}_{o,opt} = e^{j\varphi_o}\sqrt{\lambda_{o1}}\,\mathbf{u}_{o1}\ , \tag{3.21}$$

where λ_{e1} and λ_{o1} are the largest eigenvalues of $\mathbf{Q}_e$ and $\mathbf{Q}_o$, respectively, $\mathbf{u}_{e1}$ and $\mathbf{u}_{o1}$ are the correponding (normalized) eigenvectors, and φ_e and φ_o are two arbitrary (unknown) phase constants.

Summary of the global signal synthesis algorithm. In the following summary of the global signal synthesis algorithm derived above, the power algorithm (see Subsection 2.3) is incorporated for calculating the largest eigenvalues and corresponding eigenvectors of the matrices $\mathbf{Q}_e$ and $\mathbf{Q}_o$. It is assumed that the largest eigenvalues of $\mathbf{Q}_e$ and $\mathbf{Q}_o$ equal the respective eigenvalues with maximal magnitude.

1a) Transform the model $\tilde{T}(n,\Theta)$ as

$$\tilde{c}(n,m) = \frac{1}{2}\int_{-1/2}^{1/2} \tilde{T}\left(n,\frac{\Theta}{2}\right) e^{j2\pi m\Theta}\,d\Theta\ . \tag{3.22}$$

1b) Form the Hermitian matrices $\mathbf{Q}_e$ and $\mathbf{Q}_o$ as

$$Q_{e,kl} \triangleq \tilde{c}(k+l,k-l)\ , \qquad Q_{o,kl} \triangleq \tilde{c}(k+l+1,k-l)\ . \tag{3.23}$$

2) Calculate the largest eigenvalues λ_{e1}, λ_{o1} and the corresponding (normalized) eigenvectors $\mathbf{u}_{e1}$, $\mathbf{u}_{o1}$ of the matrices $\mathbf{Q}_e$ and $\mathbf{Q}_o$ by means of the power-algorithm iteration: for i=1,2,...

(a) $$\mathbf{v}_e^{(i)} = \mathbf{Q}_e\,\mathbf{u}_e^{(i-1)}\ , \qquad \mathbf{v}_o^{(i)} = \mathbf{Q}_o\,\mathbf{u}_o^{(i-1)}\ ; \tag{3.24}$$

(b) $$\lambda_e^{(i)} = \|\mathbf{v}_e^{(i)}\|\ , \qquad \lambda_o^{(i)} = \|\mathbf{v}_o^{(i)}\|\ ; \tag{3.25}$$

(c) $$\mathbf{u}_e^{(i)} = \mathbf{v}_e^{(i)} / \lambda_e^{(i)}\ , \qquad \mathbf{u}_o^{(i)} = \mathbf{v}_o^{(i)} / \lambda_o^{(i)}\ . \tag{3.26}$$

On convergence ($i\to\infty$), the looked-for eigenvalues and eigenvectors are obtained as

$$\lambda_{e1} = \lambda_e^{(\infty)}\ ,\quad \mathbf{u}_{e1} = \mathbf{u}_e^{(\infty)}\ ; \qquad \lambda_{o1} = \lambda_o^{(\infty)}\ ,\quad \mathbf{u}_{o1} = \mathbf{u}_o^{(\infty)}\ . \tag{3.27}$$

3) Form the synthesis solution $x_{opt}(n)$ by interleaving even-indexed and odd-indexed signal samples as

$$x_{opt}(n) = \begin{cases} e^{j\varphi_e}\sqrt{\lambda_{e1}}\,u_{e1,k}\ , & n=2k \\ e^{j\varphi_o}\sqrt{\lambda_{o1}}\,u_{o1,k}\ , & n=2k+1 \end{cases}\ , \tag{3.28}$$

where φ_e and φ_o are two arbitrary (unknown) phase constants.

In contrast to the general solution (2.23) derived in Subsection 2.2 for a globally

unitary BTFR, the solution (3.28) for the WD contains two *independent* unknown phase constants φ_e and φ_o associated with the subsequences of even-indexed and odd-indexed signal samples $x_e(k) = x(2k)$ and $x_o(k) = x(2k+1)$, respectively. Thus, there exists not only a phase ambiguity affecting the entire signal x(n) but, in addition, a "relative" phase ambiguity which produces a phase offset of even-indexed and odd-indexed signal samples *relative to each other*. This relative phase ambiguity, which is a consequence of the WD's global non-unitarity, will be further discussed in Section 7.

We note that an analogous algorithm for global signal synthesis exists for the AF version AF1 (see (1.10)) whose unitarity properties are analogous to those of the WD. For WD', RD, and AF2, on the other hand, the global signal synthesis method of Subsection 2.2 can be applied since these BTFRs are globally unitary.

Subspace interpretation of global signal synthesis. We have seen above that global signal synthesis in the case of the WD splits up into two separate and independent synthesis procedures for the even-indexed and odd-indexed signal samples $x_e(k) = x(2k)$ and $x_o(k) = x(2k+1)$, respectively. We shall now show that these synthesis procedures can be interpreted as *subspace signal synthesis* procedures on the signal subspaces $\mathcal{E}$ and $\mathcal{O}$ introduced in Subsection 3.1.

Let us first consider subspace signal synthesis on the subspace $\mathcal{E}$, i.e., the constrained minimization

$$\varepsilon_x = \| \tilde{T} - WD_x \| \longrightarrow \min_{x \in \mathcal{E}} .$$

Since the WD is unitary on $\mathcal{E}$ (apart from a scaling factor of 2), we can apply the general synthesis method of Subsection 2.2. Step 1a is the calculation of the projection coefficients (including a factor 1/2 to compensate for the scaling factor)

$$\gamma_{kl} = \tfrac{1}{2}\left(\tilde{T}, WD_{kl}\right) = \tfrac{1}{2} \sum_n \int_{-1/4}^{1/4} \tilde{T}(n,\Theta)\; 2\,\delta\big(n-(k+l)\big)\, e^{j4\pi\Theta(k-l)}\, d\Theta =$$

$$= \tfrac{1}{2} \int_{-1/2}^{1/2} \tilde{T}\left(k+l, \tfrac{\Theta}{2}\right) e^{j2\pi\Theta(k-l)}\, d\Theta ,$$

where (3.12) has been used. Comparing with (3.22) and (3.23), we note the identity

$$\gamma_{kl} = Q_{e,kl} .$$

For a real-valued model $\tilde{T}(n,\Theta)$, the matrix $\boldsymbol{\Gamma} = (\gamma_{kl}) = \mathbf{Q}_e$ is Hermitian. Thus, Step 1b (calculation of the Hermitian component $\boldsymbol{\Gamma}_H$) may be omitted, and the expansion coefficients of the synthesized signal are directly obtained according to Step 3 as

$$\boldsymbol{\alpha}_{opt} = e^{j\varphi} \sqrt{\lambda_1}\, \mathbf{u}_1 \qquad (3.29)$$

(cf. (2.22)). But the expansion coefficients α_k were shown in Subsection 3.1 to be

the even-indexed signal samples $x_e(k)=x(2k)$. Furthermore, since $\mathbf{\Gamma} = \mathbf{Q}_e$, the largest eigenvalue λ_1 and the corresponding eigenvector $\mathbf{u}_1$ of $\mathbf{\Gamma}$ equal the largest eigenvalue λ_{e1} and correspending eigenvector $\mathbf{u}_{e1}$ of $\mathbf{Q}_e$, respectively. Thus, the result (3.29) can be rewritten as

$$\mathbf{x}_{e,opt} = e^{j\varphi_e} \sqrt{\lambda_{e1}}\, \mathbf{u}_{e1},$$

which is exactly the result (3.21) of global signal synthesis. It has thus been shown that the global signal synthesis result for the even-indexed signal samples $x_e(k)$ equals the result of subspace signal synthesis on the space $\mathfrak{E}$. A completely analogous derivation shows that the global signal synthesis result for the odd-indexed signal samples $x_o(k)$ equals the result of subspace signal synthesis on the space $\mathfrak{O}$.

3.3 Halfband Signal Synthesis

In the previous subsection, it has been shown that the global WD signal synthesis problem can readily be solved despite the WD's global non-unitarity. However, the occurrence of a relative phase ambiguity of even-indexed and odd-indexed signal samples is an awkward inconvenience of global signal synthesis (see Section 7).

In practice, the application of the WD is usually restricted to halfband signals for which the WD is non-aliased. Thus, it is reasonable to require that the result of WD signal synthesis, too, be a halfband signal. For example, one may be interested in obtaining an analytic signal as the result of signal synthesis. We are thus led to consider the following *halfband signal synthesis* problem incorporating a halfband constraint $x(n) \in \mathfrak{H}$,

$$\varepsilon_x = \| \tilde{T} - WD_x \| \longrightarrow \min_{x \in \mathfrak{H}}, \tag{3.30}$$

where $\mathfrak{H}$ is a halfband signal subspace with given (application-specific) center frequency Θ_0. Since the WD is unitary on $\mathfrak{H}$, the solution to the halfband signal synthesis problem (3.30) is simply a special case of the general synthesis algorithm derived in Subsection 2.2. Moreover, since the result of the general algorithm is unique up to a phase affecting the entire signal, the troublesome relative phase ambiguity of global signal synthesis is automatically avoided.

We now specialize the general subspace signal synthesis algorithm of Subsection 2.2 to the BTFR WD and to the signal subspace $\mathfrak{H}$. With the induced WD-domain basis given by (3.9) as

$$WD_{kl}(n,\Theta) = 2\, WD_h\big(n-(k+l),\Theta\big)\, e^{-j4\pi\Theta(k-l)},$$

the expansion coefficients of the model's projection on the induced WD-domain subspace $\mathfrak{H}_{WD}$ are

$$\gamma_{kl} = (\tilde{T}, WD_{kl}) = \sum_n \int_{-1/4}^{1/4} \tilde{T}(n,\Theta)\ 2\ WD_h(n-(k+l),\Theta)\ e^{j4\pi\Theta(k-l)}\, d\Theta\ .$$

This can be split into a convolution operation with respect to the time index n and an inverse Fourier transform with respect to the normalized-frequency variable Θ (see the algorithm summary given below). After calculation of the expansion coefficients γ_{kl}, the synthesis solution is obtained according to (2.21), (2.22).

Summary of the halfband signal synthesis algorithm. The resulting halfband signal synthesis algorithm is now summarized; again, the power algorithm is used for calculating the largest eigenvalue and the corresponding eigenvector of the coefficient matrix $\mathbf{\Gamma} = (\gamma_{kl})$.

1a) Perform the convolution

$$\tilde{T}_{\mathcal{H}}(n,\Theta) = \sum_{n'} WD_h(n-n',\Theta)\ \tilde{T}(n',\Theta) \qquad (3.31)$$

where $WD_h(n,\Theta)$ is given by (3.10). Due to (3.10), this convolution amounts to an ideal lowpass filtering (with Θ-dependent cutoff frequency) of the model $\tilde{T}(n,\Theta)$ in the time direction. We note that $\tilde{T}_{\mathcal{H}}(n,\Theta)$ can be shown to be the orthogonal projection of the model $\tilde{T}(n,\Theta)$ on the induced WD-domain subspace $\mathcal{H}_{WD}$. The "projected model" $\tilde{T}_{\mathcal{H}}(n,\Theta)$ is thus an element of $\mathcal{H}_{WD}$.

1b) Calculate the inverse Fourier transform

$$c_{\mathcal{H}}(n,m) = \frac{1}{2} \int_{-1/2}^{1/2} \tilde{T}_{\mathcal{H}}\left(n,\frac{\Theta}{2}\right) e^{j2\pi m\Theta}\, d\Theta\ .$$

1c) Form the matrix $\mathbf{\Gamma} = (\gamma_{kl})$ with

$$\gamma_{kl} = 2\, c_{\mathcal{H}}(k+l,k-l)\ . \qquad (3.32)$$

It is easily shown that $\mathbf{\Gamma}$ is a Hermitian matrix due to the real-valuedness of the model $\tilde{T}(n,\Theta)$.

2) Calculate the largest eigenvalue λ_1 and the associated (normalized) eigenvector $\mathbf{u}_1$ of $\mathbf{\Gamma}$ by means of the power-algorithm iteration: for i=1,2,...

(a) $$\mathbf{v}^{(i)} = \mathbf{\Gamma}\,\mathbf{u}^{(i-1)}\ ,$$

(b) $$\lambda^{(i)} = \|\mathbf{v}^{(i)}\|\ ,$$

(c) $$\mathbf{u}^{(i)} = \mathbf{v}^{(i)}/\lambda^{(i)}\ .$$

On convergence ($i \to \infty$), λ_1 and $\mathbf{u}_1$ are obtained as

$$\lambda_1 = \lambda^{(\infty)}, \qquad \mathbf{u}_1 = \mathbf{u}^{(\infty)}\ .$$

3) Form the synthesis solution as

$$x_{opt}(n) = \sum_k \alpha_{opt,k}\, h_k(n)$$

with

$$\boldsymbol{\alpha}_{opt} = e^{j\varphi} \sqrt{\lambda_1}\, \mathbf{u}_1 ,$$

where φ is an arbitrary phase constant.

With $\alpha_{opt,k} = \sqrt{2}\, x_{opt}(2k)$ and $h_k(n) = \sqrt{2}\, h(n-2k)$ according to (3.7) and (3.4), respectively, Step 3 of this algorithm can be reformulated as follows:

3a) Obtain the even-indexed signal samples as

$$x_{opt}(2k) = e^{j\varphi} \sqrt{\lambda_1/2}\, u_{1,k} .$$

3b) Derive the odd-indexed signal samples by means of the interpolation

$$x_{opt}(2k+1) = 2 \sum_{k'} h(2k+1-2k')\, x_{opt}(2k') . \qquad (3.33)$$

Thus, the halfband signal synthesis algorithm first calculates the optimal even-indexed signal samples, from which the odd-indexed signal samples are then derived by means of an interpolation step.

Comparison with global signal synthesis. Comparing the halfband signal synthesis algorithm (HSSA) as stated above with the global signal synthesis algorithm (GSSA) summarized by (3.22)-(3.28), we note two similarities: both algorithms perform an inverse Fourier transform and a coordinate transform and require the calculation of the largest eigenvalue and corresponding eigenvector of one (HSSA) or two (GSSA) Hermitian matrices. On the other hand, there are two main differences. Firstly, the HSSA uses the model's projection $\hat{T}_{\mathfrak{H}}(n,\Theta)$ on $\mathfrak{H}_{WD}$ for inverse Fourier transform (and thus requires an additional projection (convolution) step) while the GSSA takes the model $\hat{T}(n,\Theta)$ itself. Secondly, the HSSA synthesizes only the even-indexed signal samples and derives the odd-indexed samples through a simple interpolation; in contrast, the GSSA performs separate syntheses of even-indexed and odd-indexed signal samples. As a consequence, the result of halfband signal synthesis features only a phase ambiguity affecting the entire signal whereas the result of global signal synthesis contains an additional relative phase ambiguity of even-indexed and odd-indexed signal samples.

A reduced-cost halfband signal synthesis algorithm. In some situations, it is advantageous to use a modified algorithm for halfband signal synthesis which is suboptimal but very economical. Let us call a model $\hat{T}(n,\Theta)$ *halfband-consistent* if it is an element of the induced WD-domain subspace $\mathfrak{H}_{WD}$ and thus equals its projection on $\mathfrak{H}_{WD}$, $\hat{T}(n,\Theta) = \hat{T}_{\mathfrak{H}}(n,\Theta)$. For example, the WD outcome $T_x(n,\Theta)$ of a halfband signal $x(n) \in \mathfrak{H}$ is an element of $\mathfrak{H}_{WD}$ and thus halfband-consistent. When

applying halfband signal synthesis to a halfband-consistent model $\tilde{T}(n,\Theta)$, Step 1a, i.e., calculation of the projection $\tilde{T}_{\mathfrak{H}}(n,\Theta)$ by means of the convolution (3.31), can obviously be omitted, which yields a substantial reduction of computation. In fact, the halfband signal synthesis algorithm here reduces essentially to the "even" part of the *global* signal synthesis algorithm, with subsequent interpolation to obtain the odd-indexed signal samples.

Of course, exact halfband consistency of a model will be the exception rather than the rule. However, there are frequently situations where a model $\tilde{T}(n,\Theta)$ is *nearly halfband-consistent* in the sense that the normalized deviation $\|\tilde{T}-\tilde{T}_{\mathfrak{H}}\| / \|\tilde{T}\|$ is small. In particular, a model will be nearly halfband-consistent if it is derived from a valid WD outcome of a halfband signal through some modification which preserves halfband consistency apart from small errors. If we then perform the halfband signal synthesis algorithm but omit the projection convolution (3.31), we obtain a synthesis result which is a halfband signal (since it is derived by the interpolation (3.33)) but which is suboptimal since it does not minimize the synthesis error. Of course, it may be expected that the deviation from the optimal solution will be small provided that the model's deviation from halfband consistency is itself small.

3.4 Other Approaches

Two approaches to the signal synthesis problem in the case of the WD have been discussed in the previous two sections. The first approach, global or unconstrained signal synthesis, produces signals with a phase ambiguity of even-indexed and odd-indexed samples relative to each other; in general, the WD of these signals will contain some amount of aliasing even if the phases of even-indexed and odd-indexed samples are matched to each other by one of the phase matching algorithms discussed in Section 7. The second approach, halfband signal synthesis, avoids the relative phase ambiguity of even-indexed and odd-indexed signal samples and produces signals whose WD is strictly non-aliased.

Alternative definition of WD. A third approach, proposed in [34], circumvents the problems of the WD's global non-unitarity and relative phase ambiguity by formulating the global signal synthesis problem as

$$\varepsilon_x = \|\tilde{T} - WD'_x\| \longrightarrow \min_x ,$$

where WD' is the alternative definition of the discrete-time WD given by (1.5). In contrast to the WD version defined by (1.4), WD' is globally unitary. Thus, the general algorithm for global signal synthesis derived in Subsection 2.2 can be applied, and the synthesis solution x(n) does not suffer from a relative phase ambiguity of even-indexed and odd-indexed signal samples. However, just as the WD, WD' is aliased in general, the only exception again being the case of halfband signals. A halfband restriction (i.e., halfband signal synthesis) can again be used if one wishes to obtain halfband signals with non-aliased WD'.

Gabor-type basis. An interesting modification of signal synthesis is proposed in [35,36]. The synthesis result x(n) is represented by its Gabor expansion

$$x(n) = \sum_k \sum_l \alpha_{kl}\, g_{kl}(n) \qquad \text{with} \qquad g_{kl}(n) = g(n-kN_o)\, e^{j2\pi l \Theta_o n} . \tag{3.34}$$

The signals $g_{kl}(n)$ are time-frequency shifted versions of a signal g(n) which is assumed to be well concentrated with respect to both time and frequency (e.g., a Gaussian signal). A problem with the expansion (3.34) is that the basis signals $g_{kl}(n)$ are generally not orthogonal. This problem, however, can be overcome by means of a second basis which is bi-orthogonal to $g_{kl}(n)$. Combining the concepts of bi-orthogonal bases and induced WD-domain bases, the signal synthesis problem can again be transferred to the coefficient domain, and the optimal Gabor coefficients $\alpha_{opt,kl}$ can be derived in a way similar to the derivation in Subsection 2.2. The advantage of the Gabor-type basis $g_{kl}(n)$ is the time-frequency localization and time-frequency concentration of the basis signals $g_{kl}(n)$, which permits a "time-frequency-selective" signal synthesis. The problem of WD aliasing remains but can be controlled to some extent using the frequency localization and concentration of the basis signals $g_{kl}(n)$: by including, in the Gabor expansion (3.34), only those basis signals $g_{kl}(n)$ which are well concentrated in a given halfband interval, the synthesis result x(n) is assured to be approximately a halfband signal. Thus, the Gabor scheme can be used to (approximately) realize halfband-constrained signal synthesis.

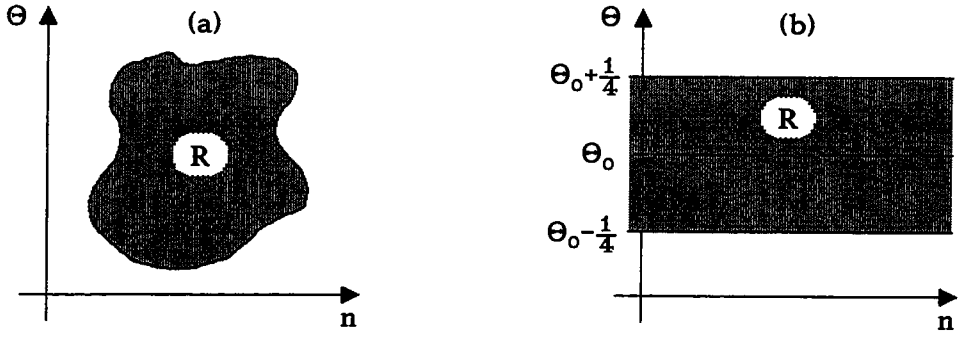

Figure 2. Time-frequency regions: (a) general region, (b) region corresponding to a halfband subspace $\mathfrak{H}$.

Time-frequency subspaces. To a given time-frequency region R (see *Figure 2.a*), we can construct a linear signal subspace $\mathfrak{S}_R$ whose signals fill the region R energetically [30,31]. Signal subspaces with this property of time-frequency energy localization are called *time-frequency subspaces*. In fact, the halfband subspace $\mathfrak{H}$ is a special case of a time-frequency subspace, where the region R is an infinite strip (*Figure 2.b*). When subspace signal synthesis is performed on a time-frequency subspace $\mathfrak{S}_R$, the synthesis result is guaranteed to assume most of its energy

inside the time-frequency region R. Somewhat similar to the Gabor-basis approach discussed above, this results in a time-frequency-selective synthesis. Aliasing and global non-unitarity of the WD are avoided if the time-frequency subspace $\mathcal{S}_R$ is constructed as a subspace of the halfband subspace $\mathcal{H}$, $\mathcal{S}_R \subset \mathcal{H}$; the general subspace signal synthesis algorithm of Subsection 2.2 can then be applied. The systematic construction of time-frequency subspaces is discussed in [30,31].

3.5 Simulation Results

Figure 3 compares computer simulation results obtained with the global signal synthesis algorithm and the halfband signal synthesis algorithm. Both algorithms were applied to the model function $\tilde{T}(n,\Theta)$ shown in Figure 3.a.

The result of global signal synthesis is depicted in Figure 3.b. Although the model is defined for $0 \le \Theta < 1/2$, the synthesized signal assumes the main part of its energy on the complementary band $1/2 \le \Theta < 1$ (see Figure 3.b.2). This is due to a phase mismatch between even-indexed and odd-indexed signal samples - a consequence of the relative phase ambiguity of global signal synthesis. We note that this phase mismatch can be removed to a large extent by means of the phase matching algorithms discussed in Section 7 (cf. Figure 12).

The application of halfband signal synthesis is demonstrated in the right-hand column of Figure 3. To be consistent with the model's frequency band $0 \le \Theta < 1/2$, the center frequency of the halfband subspace $\mathcal{H}$ was chosen as $\Theta_0=1/4$, i.e., $\mathcal{H}$ is the subspace of analytic signals. Figure 3.c shows the result of the first step of halfband signal synthesis, namely, the projection $\tilde{T}_{\mathcal{H}}(n,\Theta)$ of the model $\tilde{T}(n,\Theta)$ on the induced halfband subspace $\mathcal{H}_{WD}$ (cf. Eq. (3.31)). The final result of halfband signal synthesis is given in Figure 3.d. Note that the synthesized signal is now properly bandlimited in the halfband $0 \le \Theta < 1/2$.

4. SMOOTHED VERSIONS OF WIGNER DISTRIBUTION

So far, we have discussed BTFRs possessing a unitarity property on the space $\mathcal{S}$ on which signal synthesis is to be performed. Strictly speaking, global signal synthesis for the WD has been an exception since the WD is not unitary on the entire signal space $l_2(-\infty,\infty)$. However, global WD signal synthesis has been shown to amount to two separate signal synthesis procedures on the subspaces $\mathcal{E}$ and $\mathcal{O}$ on which the WD is essentially unitary.

In the case of unitary BTFRs, the signal synthesis problem has been reduced to the solution of an eigenvalue-eigenvector equation involving either the vector of signal samples (case of global signal synthesis) or a coefficient vector characterizing the signal (case of subspace signal synthesis). In either case, the solution is given by the largest positive eigenvalue and the corresponding eigenvector of a Hermitian matrix. It has been noted that these quantities can be computed iteratively by means of the power algorithm reviewed in Subsection 2.3.

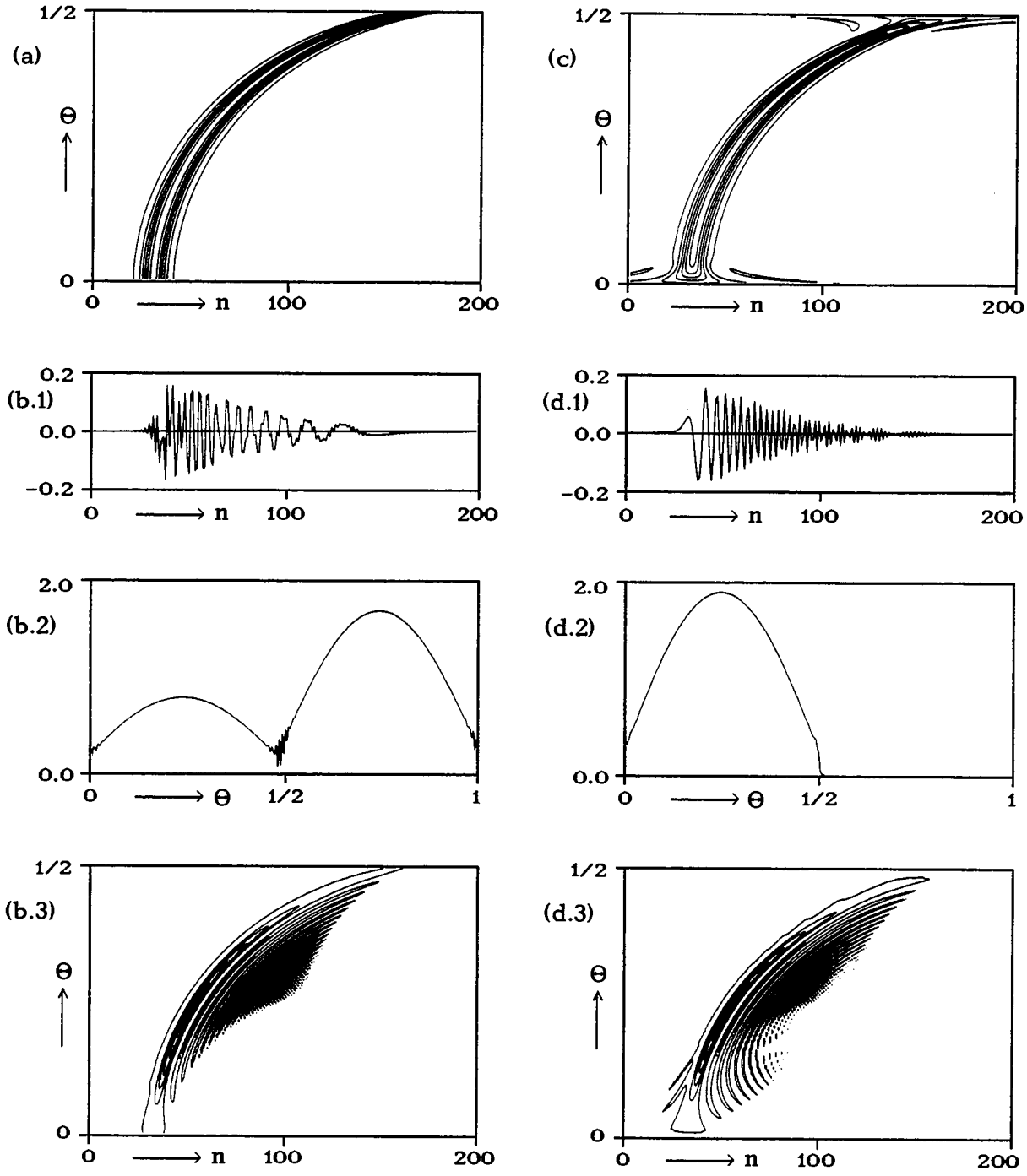

Figure 3. Results of WD global signal synthesis and WD halfband signal synthesis. (a) Model $\Upsilon(n,\Theta)$; (b) result of global signal synthesis: (b.1) real part of synthesized signal, (b.2) signal spectrum, and (b.3) WD; (c) projected model $\Upsilon_{\mathfrak{D}}(n,\Theta)$; (d) result of halfband signal synthesis: (d.1) real part of synthesized signal, (d.2) signal spectrum, and (d.3) WD. (The WD contour-line plots (b.3) and (d.3) show the positive WD parts on a linear scale.)

In this section, we consider the class of *smoothed WDs* (SWDs). The results derived so far cannot be applied to SWDs since SWDs are inherently non-unitary. On the other hand, SWDs are practically important since (i) they can often be computed on an "on-line" or short-time basis, which makes them suited for the time-frequency processing of signals with arbitrary lengths; and (ii) the interference terms of SWDs are attenuated as compared to the WD, so that the interpretation of SWDs is greatly facilitated [1,6,21,22,24] and certain applications of signal synthesis do not suffer from troublesome interference term effects [15] (see Section 8).

Due to the non-unitarity of SWDs, the derivation of a closed-form solution to the signal synthesis problem appears to be impossible. An equation representing a necessary condition for the optimal signal can be found; however, this is a complicated third-order equation which can only be solved by some iterative method. While a number of iterative standard methods are available, we here develop a method which is motivated by the case of unitary BTFRs or, in particular, the WD: first, the equation is reformulated such that it resembles the conventional eigen-equation encountered in the WD case; next, this "quasi eigen-equation" is solved iteratively by means of a natural modification of the power algorithm [15,37].

In the following subsection, we introduce the general class of SWDs, discuss a convenient characterization using kernel functions, and specify the kernels of some important SWDs. Using the strategy sketched above, iterative algorithms for global and halfband signal synthesis are then developed in Subsections 4.2 and 4.3, respectively. Simulation results are presented in Subsection 4.4.

4.1 The Class of Smoothed Wigner Distributions

By definition, the class of *smoothed Wigner distributions* (SWDs) [6] contains BTFRs which are derived from the WD by a time-frequency convolution

$$\mathrm{SWD}_{x,y}(n,\Theta) = \sum_{n'} \int_{-1/4}^{1/4} \Phi(n-n',\Theta-\Theta')\, \mathrm{WD}_{x,y}(n',\Theta')\, d\Theta' , \tag{4.1}$$

where $\Phi(n,\Theta)$ is a smoothing function such that the convolution (4.1) actually amounts to a two-dimensional lowpass filtering of the WD. It is easily shown that the SWD can be written as

$$\mathrm{SWD}_{x,y}(n,\Theta) = 2 \sum_{m} c_{x,y}^{(\varphi)}(n,m)\, e^{-j4\pi\Theta m} \tag{4.2}$$

with

$$c_{x,y}^{(\varphi)}(n,m) = \sum_{n'} \varphi(n-n',m)\, c_{x,y}(n',m) = \sum_{n'} \varphi(n-n',m)\, x(n'+m)\, y^*(n'-m) \tag{4.3}$$

where

$$\varphi(n,m) = \frac{1}{2} \int_{-1/2}^{1/2} \Phi\left(n,\frac{\Theta}{2}\right) e^{j2\pi m\Theta}\, d\Theta .$$

Note that (4.2) is reminiscent of the WD as expressed by (3.14). We require the auto-SWD $SWD_x(n,\Theta)$ to be real-valued just as the auto-WD $WD_x(n,\Theta)$. For this, the smoothing function $\Phi(n,\Theta)$ has itself to be real-valued, so that $\varphi(n,m)$ has to be Hermitian with respect to m,

$$\varphi(n,-m) = \varphi^*(n,m) .$$

Important examples of the SWD are the *pseudo Wigner distribution* (PWD) (1.6), the *smoothed pseudo Wigner distribution* (SPWD) (1.7), the *exponential distribution* (ED) (1.8), the *Born-Jordan distribution* (BJD) [7], and the *cone-kernel representation* (CKR) [38,22]. Formally, also the WD can be considered a special (limiting) case of the SWD. We now give a list of the kernels $\varphi(n,m)$ for the SWDs enumerated above:

WD: $$\varphi_{WD}(n,m) = \delta(n)$$

PWD: $$\varphi_{PWD}(n,m) = \delta(n)\, h^2(m)$$

SPWD: $$\varphi_{SPWD}(n,m) = g(n)\, h^2(m)$$

ED: $$\varphi_{ED}(n,m) = \begin{cases} \delta(n) & , \quad m=0 \\ \sqrt{\frac{\sigma}{4\pi m^2}}\, \exp\left(-\frac{\sigma n^2}{4m^2}\right) & , \quad m\neq 0 \end{cases}$$

BJD: $$\varphi_{BJD}(n,m) = \begin{cases} \delta(n) & , \quad m=0 \\ \frac{1}{2|m|}\, \mathrm{rect}_{|m|}(n) & , \quad m\neq 0 \end{cases}$$

CKR: $$\varphi_{CKR}(n,m) = e^{-2\alpha m^2}\, \mathrm{rect}_{|m|}(n) .$$

Here, g(n) and h(m) are two real-valued and even window functions satisfying $0 \le g(n), h(m) \le 1$, $h(0) = 1$, $\sum_n g(n) = 1$, $\lim_{n\to\infty} g(n) = 0$, and $\lim_{m\to\infty} h(m) = 0$, α and σ are two positive parameters, and $\mathrm{rect}_k(n)$ is the rectangular sequence defined by

$$\mathrm{rect}_k(n) = \begin{cases} 1, & |n| \le k \\ 0, & |n| > k \end{cases} \qquad (k \ge 0) \quad .$$

We observe that, for all SWDs listed above, the kernel $\varphi(n,m)$ satisfies the following properties: $\varphi^*(n,m) = \varphi(n,m)$; $\varphi(-n,m) = \varphi(n,m)$; $\varphi(n,-m) = \varphi(n,m)$; $\varphi(0,0) = 1$; $\lim_{n\to\infty} \varphi(n,m) = \lim_{m\to\infty} \varphi(n,m) = 0$.

4.2 Global Signal Synthesis

We now consider the global signal synthesis problem for the SWD,

$$\varepsilon_x = \| \hat{T} - SWD_x \| \longrightarrow \min_x ,$$

where the SWD's kernel $\varphi(n,m)$ is assumed to be given. Reasoning as in the WD case (cf. the derivation in Subsection 3.2), the squared synthesis error norm can be rewritten as

$$\varepsilon_x^2 = 2\,\|\mathcal{C} - c_x^{(\varphi)}\|^2 = 2\sum_n\sum_m |\mathcal{C}(n,m) - c_x^{(\varphi)}(n,m)|^2 ,$$

where $\mathcal{C}(n,m)$ and $c_x^{(\varphi)}(n,m)$ are given by (3.15) and (4.3), respectively. Setting the derivative of ε_x^2 with respect to $x(n)$ equal to zero, we obtain as a necessary condition for the signal $x(n)$ minimizing ε_x^2 the following system of third-order equations:

$$\sum_m x(n-2m)\, C_x(n-m,m) = 0 \qquad \text{for all } n \tag{4.4}$$

with

$$C_x(n,m) = \sum_{n'} \varphi^*(n'-n,m)\left[\mathcal{C}(n',m) - c_x^{(\varphi)}(n',m)\right] . \tag{4.5}$$

We now split the set of equations (4.4) into a subset for n even ($n=2\upsilon$) and a subset for n odd ($n=2\upsilon+1$). After a change of indices, the two subsets of equations can be written in matrix-vector notation as

$$\mathbf{Q}'_{e,x}\,\mathbf{x}_e = \mathbf{o} , \qquad \mathbf{Q}'_{o,x}\,\mathbf{x}_o = \mathbf{o} \tag{4.6}$$

where, as usual, $\mathbf{x}_e$ and $\mathbf{x}_o$ denote the vectors of even-indexed and odd-indexed signal samples, respectively, and the matrices $\mathbf{Q}'_{e,x}$ and $\mathbf{Q}'_{o,x}$ are derived from $C_x(n,m)$ as

$$(\mathbf{Q}'_{e,x})_{kl} = C_x(k+l,k-l) , \qquad (\mathbf{Q}'_{o,x})_{kl} = C_x(k+l+1,k-l) .$$

We stress that the separate equations (4.6) do not imply that the syntheses of even-indexed and odd-indexed signal samples are decoupled (as in the case of the WD), since each of the matrices $\mathbf{Q}'_{e,x}$ and $\mathbf{Q}'_{o,x}$ generally contains both the even-indexed and the odd-indexed samples. On the other hand, it is easily shown that the overall *phases* of the subsequences of even-indexed and odd-indexed signal samples are again decoupled. This means that the solution of global SWD signal synthesis will again feature a relative phase ambiguity.

The quasi power algorithm. We now develop an iterative technique for solving the third-order equations (4.6) [15,37]. This technique is inspired by the power algorithm used in the WD case, and it is based on a reformulation of (4.6). Indeed, adding $\|\mathbf{x}_e\|^2\mathbf{x}_e$ and $\|\mathbf{x}_o\|^2\mathbf{x}_o$ to the first and second equation of (4.6), respectively, we obtain the equations

$$\mathbf{Q}_{e,x}\,\mathbf{x}_e = \|\mathbf{x}_e\|^2\mathbf{x}_e , \qquad \mathbf{Q}_{o,x}\,\mathbf{x}_o = \|\mathbf{x}_o\|^2\mathbf{x}_o , \tag{4.7}$$

where the new matrices $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$ are defined as

$$\mathbf{Q}_{e,x} = \mathbf{Q}'_{e,x} + \mathbf{x}_e\mathbf{x}_e^+ , \qquad \mathbf{Q}_{o,x} = \mathbf{Q}'_{o,x} + \mathbf{x}_o\mathbf{x}_o^+ . \tag{4.8}$$

The matrices $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$ are easily checked to be Hermitian for a real-valued model $\hat{T}(n,\Theta)$ and an SWD kernel $\varphi(n,m)$ which is Hermitian with respect to m. The equations (4.7) resemble the eigen-equations (3.20) encountered in the WD case; however, the matrices $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$ here are themselves *signal-dependent*, i.e., they are (quadratic) functions of the signal x(n). Nevertheless, we propose to solve the "quasi-eigen-equations" (4.7) by means of a suitable modification of the power algorithm, which accounts for the signal-dependency of the matrices $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$ simply by updating these matrices at each iteration step. Assuming that the results of the (i-1)-th iteration step have already been derived, the i-th iteration step of this "quasi power algorithm" involves the following operations:

First, the matrices $\mathbf{Q}_{e,x(i-1)}$ and $\mathbf{Q}_{o,x(i-1)}$ are formed by inserting the (i-1)-th iteration version $x^{(i-1)}(n)$ of the signal x(n) into the definition (4.8) of $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$. Next, the matrix-vector multiplication of the power algorithm is performed (cf. (3.24)),

$$\mathbf{v}_e^{(i)} = \mathbf{Q}_{e,x(i-1)}\,\mathbf{u}_e^{(i-1)} , \qquad \mathbf{v}_o^{(i)} = \mathbf{Q}_{o,x(i-1)}\,\mathbf{u}_o^{(i-1)} .$$

Then, the usual normalizations (cf. (3.25), (3.26)) are carried out,

$$\lambda_e^{(i)} = \|\mathbf{v}_e^{(i)}\| , \qquad \lambda_o^{(i)} = \|\mathbf{v}_o^{(i)}\| ;$$

$$\mathbf{u}_e^{(i)} = \mathbf{v}_e^{(i)} / \lambda_e^{(i)} , \qquad \mathbf{u}_o^{(i)} = \mathbf{v}_o^{(i)} / \lambda_o^{(i)} .$$

Finally, the i-th iteration version of the signal x(n) is formed by interleaving even-indexed and odd-indexed signal samples (cf. (3.28)),

$$x^{(i)}(n) = \begin{cases} \sqrt{\lambda_e^{(i)}}\; u_{e,k}^{(i)} , & n=2k \\ \sqrt{\lambda_o^{(i)}}\; u_{o,k}^{(i)} , & n=2k+1 \end{cases} .$$

Note that $x^{(i)}(n)$ will be used in the next ((i+1)-th) iteration for constructing the matrices $\mathbf{Q}_{e,x(i)}$ and $\mathbf{Q}_{o,x(i)}$.

It is seen that the quasi power algorithm differs from the power algorithm in that the signal-dependent matrices $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$ are updated at each iteration step using the current version of the signal x(n). When the iteration converges ($i \to \infty$), the synthesis solution x(n) is obtained as

$$x(n) = x^{(\infty)}(n) ,$$

where certain phase factors of even-indexed and odd-indexed signal samples are implicitly included; these depend on the initial vectors $\mathbf{u}_e^{(0)}$ and $\mathbf{u}_o^{(0)}$ with which the iteration is started. Alternatively, the general solution (containing arbitrary phase values φ_e and φ_o) can be written as

$$x(n) = \begin{cases} e^{j\varphi_e}\sqrt{\lambda_e^{(\infty)}}\,u_{e,k}^{(\infty)}, & n=2k \\ e^{j\varphi_o}\sqrt{\lambda_o^{(\infty)}}\,u_{o,k}^{(\infty)}, & n=2k+1 \end{cases} .$$

Summary of the global signal synthesis algorithm. Before further discussing the algorithm derived above, we summarize it as follows (see [37] for a discussion of its implementation):

1) Transform the model $\tilde{T}(n,\Theta)$ as

$$\tilde{c}(n,m) = \frac{1}{2}\int_{-1/2}^{1/2} \tilde{T}(n,\frac{\Theta}{2})\, e^{j2\pi m\Theta}\, d\Theta . \tag{4.9}$$

2) Perform the quasi power algorithm iteration: for i=1,2,...

(a) Construct the matrices $\mathbf{Q}_{e,x(i-1)}$ and $\mathbf{Q}_{o,x(i-1)}$ as follows:

$$c_{x(i-1)}^{(\varphi)}(n,m) = \sum_{n'} \varphi(n-n',m)\, x^{(i-1)}(n'+m)\, x^{(i-1)*}(n'-m) , \tag{4.10}$$

$$C_{x(i-1)}(n,m) = \sum_{n'} \varphi^*(n'-n,m)\left[\tilde{c}(n',m) - c_{x(i-1)}^{(\varphi)}(n',m)\right] ; \tag{4.11}$$

$$\left(\mathbf{Q}_{e,x(i-1)}\right)_{kl} = C_{x(i-1)}(k+l,k-l) + x_e^{(i-1)}(k)\, x_e^{(i-1)*}(l) , \tag{4.12}$$

$$\left(\mathbf{Q}_{o,x(i-1)}\right)_{kl} = C_{x(i-1)}(k+l+1,k-l) + x_o^{(i-1)}(k)\, x_o^{(i-1)*}(l) . \tag{4.13}$$

(b) Calculate the matrix-vector products

$$\mathbf{v}_e^{(i)} = \mathbf{Q}_{e,x(i-1)}\, \mathbf{u}_e^{(i-1)} , \qquad \mathbf{v}_o^{(i)} = \mathbf{Q}_{o,x(i-1)}\, \mathbf{u}_o^{(i-1)} . \tag{4.14}$$

(c) Perform the normalizations

$$\lambda_e^{(i)} = \|\mathbf{v}_e^{(i)}\| , \qquad \lambda_o^{(i)} = \|\mathbf{v}_o^{(i)}\| ; \tag{4.15}$$

$$\mathbf{u}_e^{(i)} = \mathbf{v}_e^{(i)} / \lambda_e^{(i)} , \qquad \mathbf{u}_o^{(i)} = \mathbf{v}_o^{(i)} / \lambda_o^{(i)} . \tag{4.16}$$

(d) Form the signal

$$x^{(i)}(n) = \begin{cases} \sqrt{\lambda_e^{(i)}}\, u_{e,k}^{(i)}, & n=2k \\ \sqrt{\lambda_o^{(i)}}\, u_{o,k}^{(i)}, & n=2k+1 \end{cases} . \tag{4.17}$$

3) After convergence ($i\to\infty$), the synthesis solution $x(n)$ is obtained as $x(n) = x^{(\infty)}(n)$.

The quasi power algorithm summarized above represents one of several possible ways to iteratively solve (or, more precisely, attempt to solve) the necessary-condition equations (4.6). It is justified to some extent by the following facts:

(i) In the case of the WD, which is a special case of the SWD, the quasi power algorithm reduces to the WD algorithm (i.e., the conventional power algorithm) as given by (3.22)-(3.28). Indeed, for $\varphi(n,m) = \varphi_{WD}(n,m) = \delta(n)$, we find that

$$\mathbf{Q}_{e,x(i-1)} = \mathbf{Q}_e \ , \qquad \mathbf{Q}_{o,x(i-1)} = \mathbf{Q}_o \ ,$$

where $\mathbf{Q}_e$ and $\mathbf{Q}_o$ are the signal-*independent* matrices (see (3.23)) of the WD case. Consequently, the matrices $\mathbf{Q}_{e,x(i-1)}$ and $\mathbf{Q}_{o,x(i-1)}$ remain unchanged throughout the iteration, and the iteration thus reduces to the original power algorithm. Therefore, the power algorithm, which is known to converge and produce the correct (optimal) solution, is a special case of the quasi power algorithm.

(ii) When the iteration converges, the resulting signal x(n) is guaranteed to be a solution to the necessary-condition equation (4.4). To show this, we first consider the "even" equation of (4.14) which can be rewritten as

$$\|\mathbf{x}_e^{(i)}\| \, \mathbf{x}_e^{(i)} = \mathbf{Q}_{e,x(i-1)} \, \frac{\mathbf{x}_e^{(i-1)}}{\|\mathbf{x}_e^{(i-1)}\|} \ . \tag{4.18}$$

Now, convergence means $\mathbf{x}_e^{(i)} = \mathbf{x}_e^{(i-1)}$, so that (4.18) reduces to

$$\mathbf{Q}_{e,x(i)} \, \mathbf{x}_e^{(i)} = \|\mathbf{x}_e^{(i)}\|^2 \, \mathbf{x}_e^{(i)} \ ,$$

which shows that $\mathbf{x}_e^{(i)}$ satisfies the first ("even") equation of (4.7). An analogous argument shows that $\mathbf{x}_o^{(i)}$ satisfies the second ("odd") equation of (4.7). But together, the equations (4.7) are equivalent to the necessary-condition equation (4.4), which completes our proof.

Of course, the necessary-condition equation must be expected to have many solutions; thus, the fact that the synthesis result satisfies the necessary-condition equation does not prove that this result is truly optimal in the sense that it provides the global minimum of the synthesis error ε_x. Also, there is no analytic proof that the iteration converges at all. However, both properties are satisfied in the special case given by the WD: here, the iteration is guaranteed to converge, and the result is guaranteed to provide the global minimum of the synthesis error.

We have practically applied the quasi power algorithm to the PWD and SPWD. Our experimental results here suggest that the algorithm converges (and produces satisfactory results) for practically arbitrary models and practically arbitrary start vectors, provided that the SWD smoothing function $\Phi(n,\Theta)$ is normalized as $\sum_n \int_{-1/4}^{1/4} \Phi(n,\Theta) \, d\Theta = \sum_n \varphi(n,0) = 1$. The convergence speed has been observed to depend primarily on the amount of smoothing performed in the SWD: the convergence will be slower in the case of more smoothing. For example, the convergence is slower for the SPWD than for the PWD (with identical window h(n)) since the SPWD features an additional smoothing in the time direction.

The PWD case [39]. The PWD is another special case of the SWD, with kernel

$$\varphi(n,m) = \varphi_{PWD}(n,m) = \delta(n) \, h^2(m) \ .$$

Inserting this into (4.3), it is easily seen that the matrices $\mathbf{Q}_{e,x}$ and $\mathbf{Q}_{o,x}$ are here given by

$$(\mathbf{Q}_{e,x})_{kl} = h^2(k-l)\,\mathcal{C}(k+l,k-l) + \left[1-h^4(k-l)\right] x_e(k)\,x_e^*(l) \tag{4.19}$$

$$(\mathbf{Q}_{o,x})_{kl} = h^2(k-l)\,\mathcal{C}(k+l+1,k-l) + \left[1-h^4(k-l)\right] x_o(k)\,x_o^*(l)\ . \tag{4.20}$$

But this shows that $\mathbf{Q}_{e,x}$ depends only on the even-indexed signal samples $x_e(k)$ and $\mathbf{Q}_{o,x}$ depends only on the odd-indexed signal samples $x_o(k)$. As a consequence, the syntheses of even-indexed and odd-indexed signal samples are completely decoupled just as in the WD case. This is due to the fact that the PWD does not feature a smoothing with respect to the time variable n.

4.3 Halfband Signal Synthesis

Just as the WD itself, $SWD_x(n,\Theta)$ is 1/2-periodic with respect to the frequency variable Θ and thus aliased unless the signal x(n) is a halfband signal. (Strictly speaking, $SWD_x(n,\Theta)$ is generally aliased even for halfband signals; this is a consequence of the smoothing in the frequency direction. However, this aliasing effect is of secondary importance when compared to the aliasing encountered in the case of non-halfband signals.) Also, we have seen that a relative phase ambiguity of even-indexed and odd-indexed signal samples exists in the case of global signal synthesis. These facts again call for the introduction of a halfband constraint $x(n) \in \mathcal{H}$ in the formulation of signal synthesis [37, 40],

$$\varepsilon_x = \left\| \hat{T} - SWD_x \right\| \longrightarrow \min_{x \in \mathcal{H}}\ , \tag{4.21}$$

where $\mathcal{H}$ is a halfband subspace with given center frequency Θ_0. As in the case of the WD, we can take account of the halfband constraint $x(n) \in \mathcal{H}$ by representing x(n) in terms of the orthonormal basis $h_k(n)$ spanning $\mathcal{H}$ (see (3.4)),

$$x(n) = \sum_k \alpha_k\, h_k(n)\ .$$

Inserting into (4.21), the halfband-constrained minimization with respect to x(n) reduces to an unconstrained minimization with respect to the coefficients α_k:

$$\varepsilon_x = \varepsilon_\alpha = \left\| \hat{T} - \sum_k \sum_l \alpha_k \alpha_l^*\, SWD_{kl} \right\| \longrightarrow \min_{\alpha}\ , \tag{4.22}$$

where

$$SWD_{kl}(n,\Theta) \triangleq SWD_{h_k,h_l}(n,\Theta)\ .$$

In contrast to the WD case, the cross-SWDs $SWD_{kl}(n,\Theta)$ do not constitute an orthonormal basis in general; this is due to the fact that the SWD is not unitary on

$\mathfrak{H}$ (the only exception being the WD). Therefore, (4.22) cannot be further simplified. To obtain a necessary condition for the optimal coefficients α_k minimizing the synthesis error (4.22), we set the derivatives of the squared synthesis error ε_α^2 with respect to the coefficients α_k equal to zero; this yields the system of third-order equations

$$\boldsymbol{\Gamma}'_\alpha \, \boldsymbol{\alpha} = \mathbf{o} \tag{4.23}$$

where

$$(\boldsymbol{\Gamma}'_\alpha)_{kl} = \gamma_{kl} - \sum_{k'} \sum_{l'} \alpha_{k'} \alpha_{l'}^{*} \, P_{k,l,k',l'}$$

with the inner products

$$\gamma_{kl} = (\tilde{T}, SWD_{kl}) , \qquad P_{k,l,k',l'} = (SWD_{k',l'}, SWD_{kl}) .$$

To solve the necessary-condition equation (4.23), we again use an iterative scheme which is motivated by the WD case and the power algorithm. Adding $\|\boldsymbol{\alpha}\|^2 \boldsymbol{\alpha}$, (4.23) becomes

$$\boldsymbol{\Gamma}_\alpha \, \boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|^2 \, \boldsymbol{\alpha} \tag{4.24}$$

with

$$\boldsymbol{\Gamma}_\alpha = \boldsymbol{\Gamma}'_\alpha + \boldsymbol{\alpha}\boldsymbol{\alpha}^{+} .$$

Eq. (4.24) is again a "quasi-eigen-equation;" we once more propose to solve it iteratively by means of the quasi-power algorithm described in the previous section, i.e., the original power algorithm is modified to include an update of the matrix $\boldsymbol{\Gamma}_\alpha$ at each iteration using the current version of $\boldsymbol{\alpha}$.

Summary of the halfband signal synthesis algorithm. The resulting algorithm is summarized in the following; an efficient implementation can be found in [37]. We note that Steps 1a through 1d of our summary can be shown to be equivalent to the calculation of the inner product $\gamma_{kl} = (\tilde{T}, SWD_{kl})$.

1a) Smooth the model according to

$$\tilde{T}^{(\Phi)}(n,\Theta) = \sum_{n'} \int_{-1/4}^{1/4} \Phi(n'-n,\Theta'-\Theta) \; \tilde{T}(n',\Theta') \, d\Theta' .$$

1b) Perform a Θ-dependent lowpass filtering in the time direction (cf. (3.31))

$$\tilde{T}_{\mathfrak{H}}^{(\Phi)}(n,\Theta) = \sum_{n'} WD_h(n-n', \Theta) \; \tilde{T}^{(\Phi)}(n',\Theta) .$$

1c) Calculate the inverse Fourier transform

$$\mathfrak{C}_{\mathfrak{H}}^{(\varphi)}(n,m) = \frac{1}{2}\int_{-1/2}^{1/2} \tilde{T}_{\mathfrak{H}}^{(\Phi)}(n,\tfrac{\Theta}{2})\, e^{j2\pi m\Theta}\, d\Theta \ .$$

1d) Calculate the coefficients

$$\gamma_{kl} = 2\,\mathfrak{C}_{\mathfrak{H}}^{(\varphi)}(k+l,k-l) \ .$$

1e) Calculate the inner products (see [37] for an efficient implementation)

$$P_{k,l,k',l'} = \left(SWD_{k',l'}, SWD_{kl}\right) .$$

2) Perform the quasi power algorithm iteration: for i=1,2,...
(a) Construct the matrix $\mathbf{\Gamma}_{\alpha(i-1)}$ as

$$\left(\mathbf{\Gamma}_{\alpha(i-1)}\right)_{kl} = \gamma_{kl} - \sum_{k'}\sum_{l'} \alpha_{k'}^{(i-1)}\, \alpha_{l'}^{(i-1)*}\, P_{k,l,k',l'} + \alpha_{k}^{(i-1)}\, \alpha_{l}^{(i-1)*} \ . \qquad (4.25)$$

(b) Calculate the matrix-vector product

$$\mathbf{v}^{(i)} = \mathbf{\Gamma}_{\alpha(i-1)}\, \mathbf{u}^{(i-1)} \ .$$

(c) Perform the normalization

$$\lambda^{(i)} = \|\mathbf{v}^{(i)}\| \ ,$$

$$\mathbf{u}^{(i)} = \mathbf{v}^{(i)}/\lambda^{(i)} \ .$$

(d) Calculate the current coefficient vector

$$\boldsymbol{\alpha}^{(i)} = \sqrt{\lambda^{(i)}}\, \mathbf{u}^{(i)} \ .$$

3a) Obtain the even-indexed signal samples as

$$x(2k) = e^{j\varphi}\, \frac{1}{\sqrt{2}}\, \alpha_k^{(\infty)} \ .$$

3b) Derive the odd-indexed signal samples by interpolation,

$$x(2k+1) = 2\sum_{k'} h(2k+1-2k')\, x(2k') \ .$$

As in the case of global signal synthesis discussed in the previous section, the following facts can be shown:

(i) For the special case of the WD, the matrix $\mathbf{\Gamma}_{\alpha}$ reduces to the $\boldsymbol{\alpha}$-independent matrix $\mathbf{\Gamma}$ of (3.32); hence, the iterative algorithm does not contain the matrix update (4.25) and thus reduces to the WD algorithm (3.31)-(3.33), i.e., to the power algorithm which is known to converge and produce the correct result for the WD.

(ii) If the iterative algorithm converges, then the resulting coefficient vector $\boldsymbol{\alpha}$ is guaranteed to satisfy the necessary-condition equation (4.23).

A reduced-cost halfband signal synthesis algorithm. Compared to both global signal synthesis for the SWD and halfband signal synthesis for the WD, the above algorithm for halfband signal synthesis in the SWD case is computationally quite expensive. A cheaper but suboptimal alternative is again obtained by a suitable modification of the global signal synthesis algorithm (4.9)-(4.17), similar to the reduced-cost strategy for WD halfband signal synthesis discussed at the end of Subsection 3.3. This modification is obtained by performing only the "even" part of the quasi power algorithm (4.10)-(4.17), and replacing (4.17) by the interpolation

$$x^{(i)}(2k+1) = 2 \sum_{k'} h(2k+1-2k')\, x^{(i)}(2k')$$

where

$$x^{(i)}(2k) = \sqrt{\lambda_e^{(i)}}\, u_{e,k}^{(i)} \ .$$

By this, it is guaranteed that at each iteration step, and thus after convergence as well, the signal $x^{(i)}(n)$ is a halfband signal. However, the result is suboptimal since its coefficients generally do not satisfy the necessary-condition equation (4.23).

4.4 Simulation Results

The application of the iterative SWD signal synthesis methods described in Subsections 4.2 and 4.3 to the PWD is illustrated in Figures 4 and 5, respectively. The PWD window h(n) is a Hamming window of length 63.

Figure 4 demonstrates the iterative process of global signal synthesis by means of the quasi power algorithm. Phase matching was performed by applying the spectral spread algorithm (see Subsection 7.3) at each iteration. The start signal $\mathbf{u}^{(0)}$ for the quasi power iteration was chosen as a noise signal whose energy is irregularly spread over the time-frequency plane (see Figure 4.b). From Figures 4.c through 4.e, it is seen how the iteration signal gradually adapts to the model shown in Figure 4.a. The minimization of the synthesis error corresponding to the convergence of the quasi power algorithm is shown in Figure 4.f.

Figure 5 presents results of both optimal and reduced-cost halfband signal synthesis from the model shown in Figure 5.a. It is seen that these results are quite similar; hence, the reduced-cost algorithm is an interesting alternative to the optimum algorithm.

5. SPECTROGRAM

The auto-*spectrogram* (cf. (1.9)) is given by the squared magnitude of the short-time Fourier transform using an analysis window h(n),

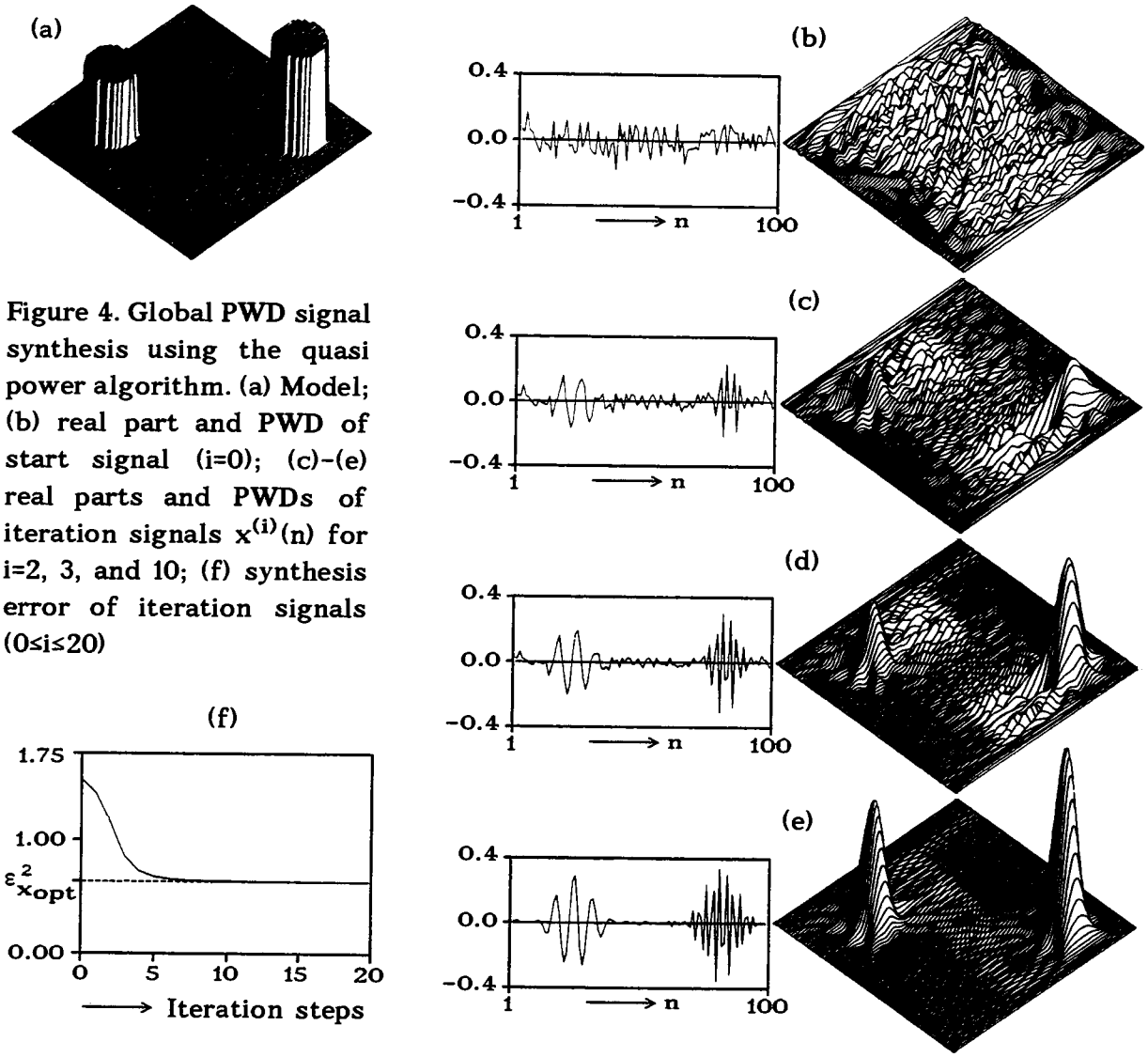

Figure 4. Global PWD signal synthesis using the quasi power algorithm. (a) Model; (b) real part and PWD of start signal (i=0); (c)-(e) real parts and PWDs of iteration signals $x^{(i)}(n)$ for i=2, 3, and 10; (f) synthesis error of iteration signals ($0 \le i \le 20$)

$$S_x(n,\Theta) = \left| \sum_{n'} x(n')\, h(n'-n)\, e^{-j2\pi\Theta n'} \right|^2 . \qquad (5.1)$$

Unlike the continuous-time spectrogram [2], the discrete-time spectrogram (5.1) is not an SWD. Indeed, $S_x(n,\Theta)$ is 1-periodic with respect to Θ just as the Fourier transform $X(\Theta)$, whereas the SWD is 1/2-periodic. As a consequence, the spectrogram is effectively non-aliased not only for halfband signals but quite generally. Also, it is easily shown that a relative phase ambiguity cannot occur in the result of global signal synthesis. Thus, there is no call for a halfband restriction, and we therefore limit our attention to the problem of global signal synthesis

$$\varepsilon_x = \| \tilde{T} - S_x \| \longrightarrow \min_x .$$

We note that this synthesis problem is not equivalent to the problem studied in

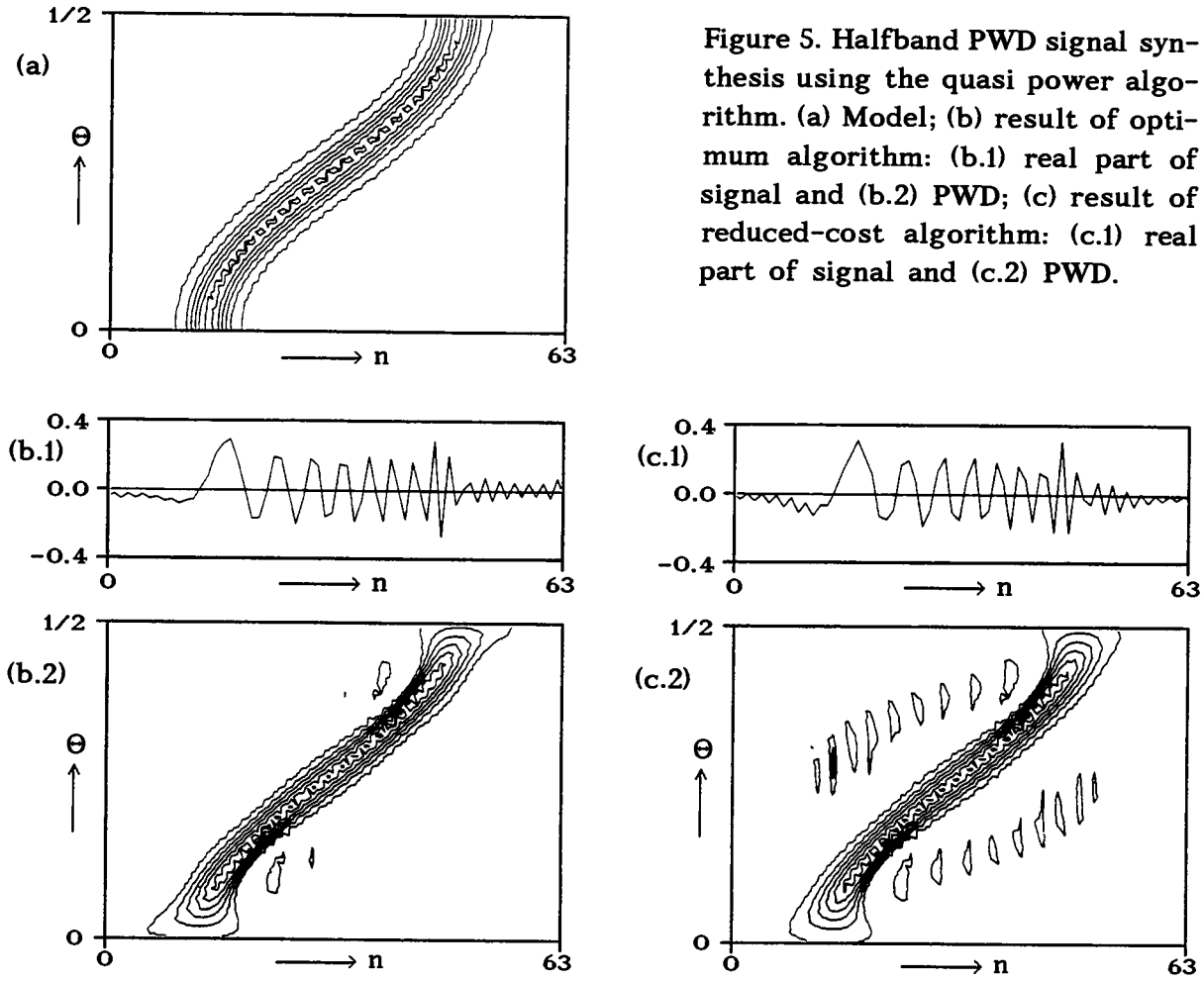

Figure 5. Halfband PWD signal synthesis using the quasi power algorithm. (a) Model; (b) result of optimum algorithm: (b.1) real part of signal and (b.2) PWD; (c) result of reduced-cost algorithm: (c.1) real part of signal and (c.2) PWD.

[41], which involves either the short-time Fourier transform itself or its magnitude (rather than squared magnitude, i.e., the spectrogram).

Similar to the SWD case, the application of the general algorithms of Subsection 2.2 to the spectrogram is impossible since the spectrogram is inherently non-unitary. We here develop an iterative algorithm for global signal synthesis which is similar to the SWD algorithm described in Subsection 4.2.

Defining a modified outer signal product $d_{x,y}(n,m)$ as

$$d_{x,y}(n,m) = q_{x,y}(n, n-m) = x(n)\,y^*(n-m) ,$$

the auto-spectrogram can be written as

$$S_x(n,\Theta) = \sum_m d_x^{(h)}(n,m)\, e^{-j2\pi\Theta m} \qquad (5.2)$$

where

$$d_x^{(h)}(n,m) = \sum_{n'} d_h(n'-n,m)\, d_x(n',m) .$$

With (5.2) and Parseval's relation, the squared synthesis error ε_x^2 becomes

$$\varepsilon_x^2 = \|\tilde{T} - S_x\|^2 = \|\tilde{d} - d_x^{(h)}\|^2 = \sum_n \sum_m |\tilde{d}(n,m) - d_x^{(h)}(n,m)|^2$$

where

$$\tilde{d}(n,m) = \int_{-1/2}^{1/2} \tilde{T}(n,\Theta)\, e^{j2\pi m\Theta}\, d\Theta .$$

Setting the derivative of ε_x^2 with respect to x(n) equal to zero, we obtain as a necessary condition for a minimum the system of third-order equations

$$\mathbf{Q}'_x \mathbf{x} = \mathbf{o} \qquad (5.3)$$

with the matrix $\mathbf{Q}'_x$ given by

$$(\mathbf{Q}'_x)_{kl} = D_x(k,k-l)$$

where

$$D_x(n,m) = \sum_{n'} d_h^*(n-n',m) \left[\tilde{d}(n',m) - d_x^{(h)}(n',m) \right] .$$

By addition of $\|\mathbf{x}\|^2 \mathbf{x}$, the necessary-condition equation (5.3) can again be given the form of a "quasi eigen-equation,"

$$\mathbf{Q}_x \mathbf{x} = \|\mathbf{x}\|^2 \mathbf{x} \qquad \text{with} \qquad \mathbf{Q}_x = \mathbf{Q}'_x + \mathbf{x}\mathbf{x}^+ ,$$

which can be solved iteratively by means of the usual quasi power algorithm.

Summary of the global signal synthesis algorithm. The above derivation yields an iterative algorithm for global signal synthesis which is summarized below.

1) Transform the model $\tilde{T}(n,\Theta)$ as

$$\tilde{d}(n,m) = \int_{-1/2}^{1/2} \tilde{T}(n,\Theta)\, e^{j2\pi m\Theta}\, d\Theta .$$

2) Perform the quasi power algorithm: for i=1,2,...

(a) Construct the matrix $\mathbf{Q}_{x(i-1)}$ as follows:

$$d_{x^{(i-1)}}^{(h)}(n,m) = \sum_{n'} d_h(n'-n,m)\, x^{(i-1)}(n')\, x^{(i-1)*}(n'-m) ,$$

$$D_{x(i-1)}(n,m) = \sum_{n'} d_h^*(n-n',m) \left[\tilde{d}(n',m) - d_{x^{(i-1)}}^{(h)}(n',m) \right] ,$$

$$(\mathbf{Q}_{x(i-1)})_{kl} = D_{x(i-1)}(k,k-l) + x^{(i-1)}(k)\, x^{(i-1)*}(l) .$$

(b) Calculate the matrix-vector product

$$\mathbf{v}^{(i)} = \mathbf{Q}_{x(i-1)}\, \mathbf{u}^{(i-1)} .$$

(c) Perform the normalization

$$\lambda^{(i)} = \|\mathbf{v}^{(i)}\| , \qquad \mathbf{u}^{(i)} = \mathbf{v}^{(i)} / \lambda^{(i)} .$$

(d) Form the signal

$$x^{(i)}(n) = \sqrt{\lambda^{(i)}}\; u_k^{(i)} .$$

3) After convergence ($i \to \infty$), the synthesis solution x(n) is obtained as

$$x(n) = e^{j\varphi}\, x^{(\infty)}(n) ,$$

where φ is an arbitrary phase.

6. ON-LINE SYNTHESIS ALGORITHMS FOR PSEUDO WIGNER DISTRIBUTION

The pseudo Wigner distribution (PWD)

$$PWD_{x,y}(n,\Theta) = 2 \sum_{m=-K}^{K} x(n+m)\, y^*(n-m)\, h^2(m)\, e^{-j4\pi\Theta m}$$

is a special case of the SWD; as such, the general synthesis algorithms developed in Subsections 4.2 and 4.3 can be applied. However, the possibility of calculating the PWD on an *on-line* basis raises the question whether signal synthesis can be performed on an on-line basis as well.

Throughout this section, we assume that the PWD's analysis window h(n) has finite time support $-K \le n \le K$, i.e., the *window length* is L = 2K+1. Then, when calculating the auto-PWD, $PWD_x(n,\Theta)$, at a given time instant n, we need only a local, finite-length record x(n-K), x(n-K+1),..., x(n+K) of the signal x(n). This allows an "on-line" computation of the PWD of signals which may have arbitrary time lengths. Unfortunately, the global signal synthesis algorithm for the PWD, as derived in Subsection 4.2, does not feature a similar on-line characteristic. Rather, the entire signal x(n) is synthesized as a whole, making use of the entire model $\hat{\Upsilon}(n,\Theta)$. Moreover, for longer signals both computation and memory requirements will become prohibitively large. Practical application of the PWD synthesis algorithm of Subsection 4.2 is therefore restricted to the off-line synthesis of signals with moderate time lengths. For the on-line synthesis of signals without any length restriction, on the other hand, we would need a sequential algorithm which synthesizes signals sample-by-sample or synthesizes consecutive signal blocks, using only a local time interval of the model for the calculation of each signal sample or signal block.

An obvious way to perform blockwise signal synthesis is to cut the model $\hat{\Upsilon}(n,\Theta)$ into adjacent blocks and then apply the synthesis scheme of Subsection 4.2 to each

model block. However, this approach is clearly inadequate since adjacent synthesis processes are in no way related and the synthesized signal will thus show discontinuities at the block boundaries. Of course, the scheme could be refined by having consecutive blocks overlap and somehow combining the synthesis results in the overlap intervals. Still, it seems that such an approach is somewhat artificial.

In the following two subsections, we develop sequential global signal synthesis algorithms for the PWD which are *recursive* in the sense that the synthesis of each signal block or signal sample is based on the previously synthesized signal samples and a local model interval [37,39,42]. These PWD signal synthesis algorithms feature an on-line mode of operation and avoid discontinuities at the block boundaries since the previously synthesized signal samples are taken into account. We note that a somewhat similar PWD synthesis algorithm has been proposed in [43]; however, unlike the algorithms discussed in the following, this algorithm is not based on a local optimization and has been observed to produce poor results [42].

6.1 Block-by-Block Synthesis

As in the case of the WD (see Subsection 3.2), the squared synthesis error ε_x^2 in the PWD case can be split into a component ε_{e,x_e}^2 involving the even-indexed signal samples and a component ε_{o,x_o}^2 involving the odd-indexed signal samples,

$$\varepsilon_x^2 = \| \tilde{T} - \mathrm{PWD}_x \|^2 = \varepsilon_{e,x_e}^2 + \varepsilon_{o,x_o}^2 ,$$

where

$$\varepsilon_{e,x_e}^2 = 2 \sum_k \sum_{l=k-K}^{k+K} | \tilde{q}_e(k,l) - h^2(k-l)\, x_e(k)\, x_e^*(l) |^2$$

$$\varepsilon_{o,x_o}^2 = 2 \sum_k \sum_{l=k-K}^{k+K} | \tilde{q}_o(k,l) - h^2(k-l)\, x_o(k)\, x_o^*(l) |^2$$

with $\tilde{q}_e(k,l)$, $\tilde{q}_o(k,l)$ given by (3.19) and (3.15). The error components ε_{e,x_e}^2 and ε_{o,x_o}^2 can again be treated separately and independently. In the following, we only consider the "even" component ε_{e,x_e}^2 since the odd case is analogous.

It is easily shown that the error component ε_{e,x_e}^2 can be decomposed as

$$\varepsilon_{e,x_e}^2 = \sum_k \varepsilon_{e,x_e}^2(k) \tag{6.1}$$

with

$$\varepsilon_{e,x_e}^2(k) \triangleq 4 \sum_{l=k-K}^{k-1} | \tilde{q}_e(k,l) - h^2(k-l)\, x_e(k)\, x_e^*(l) |^2 + 2 \, | \tilde{q}_e(k,k) - x_e(k)\, x_e^*(k) |^2 . \tag{6.2}$$

We note that the k-th error component $\varepsilon_{e,x_e}^2(k)$ is *causal* in that it depends only on the signal samples $x_e(n)$ with $n \le k$; to be more precise, it involves the signal samp-

les $x_e(n)$ in the range $k-K \le n \le k$. In a sense, $\varepsilon^2_{e,x_e}(k)$ is the contribution of the k-th time instant to the overall (squared) error ε^2_{e,x_e}. Suppose, now, that the signal $x_e(n)$ has already been synthesized up to time $\hat{n}$, i.e., the signal $x_e(n)$ is known for $n \le \hat{n}$. Based on this knowledge, we want to synthesize the next block of N samples, i.e., the signal $x_e(n)$ in the range $\hat{n}+1 \le n \le \hat{n}+N$. Let us group these unknown signal samples into a vector $\hat{\mathbf{x}}_{e,U}$ (here, the index "U" stands for "unknown," and the hat "^" is to remind us that the respective quantity is localized around the time instant $\hat{n}$),

$$(\hat{\mathbf{x}}_{e,U})_n \triangleq x_e(\hat{n}+n) , \qquad 1 \le n \le N .$$

For later use, we also define a vector $\hat{\mathbf{x}}_{e,K}$ containing the last K signal samples that have already been synthesized before and are accordingly known, i.e., $x_e(n)$ for $\hat{n}-K+1 \le n \le \hat{n}$ (the index "K" stands for "known"),

$$(\hat{\mathbf{x}}_{e,K})_n \triangleq x_e(\hat{n}-K+n) , \qquad 1 \le n \le K .$$

Finally, the combination of the two vectors will be denoted by $\hat{\mathbf{x}}_e$,

$$\hat{\mathbf{x}}_e \triangleq \begin{pmatrix} \hat{\mathbf{x}}_{e,K} \\ \hat{\mathbf{x}}_{e,U} \end{pmatrix} ; \qquad (\hat{\mathbf{x}}_e)_n \triangleq x_e(\hat{n}-K+n) , \qquad 1 \le n \le K+N .$$

Based on the previously synthesized, known signal samples contained in $\hat{\mathbf{x}}_{e,K}$, we want to synthesize the new, unknown signal block $\hat{\mathbf{x}}_{e,U}$. According to the error decomposition (6.1), the contribution of the unknown signal block $\hat{\mathbf{x}}_{e,U}$ to the overall (squared) error ε^2_{e,x_e} is given by

$$\hat{\varepsilon}^2_{x_{e,U}} = \sum_{k=\hat{n}+1}^{\hat{n}+N} \varepsilon^2_{e,x_e}(k) . \tag{6.3}$$

The result $\hat{\mathbf{x}}_{e,U,opt}$ of our recursive synthesis is now defined as the $\hat{\mathbf{x}}_{e,U}$ minimizing $\hat{\varepsilon}_{x_{e,U}}$,

$$\hat{\varepsilon}_{x_{e,U}} \longrightarrow \min_{\hat{\mathbf{x}}_{e,U}} .$$

Inserting (6.2) into (6.3) and setting the gradient of $\hat{\varepsilon}^2_{x_{e,U}}$ with respect to the unknown signal samples $\hat{\mathbf{x}}_{e,U}$ equal to zero, we obtain the following system of N equations as a necessary condition for the optimal $\hat{\mathbf{x}}_{e,U}$:

$$\sum_{l=k-K}^{\hat{n}+N} h^2(k-l) \left[q_e(k,l) - h^2(k-l)\, x_e(k)\, x_e^*(l) \right] x_e(l) = 0 , \qquad \hat{n}+1 \le k \le \hat{n}+N . \tag{6.4}$$

The lower summation bound k-K may be replaced by $\hat{n}-K+1$ since $h^2(k-l)$ will be zero for $l < k-K$. After adding

$$\left(\sum_{l=\hat{n}-K+1}^{\hat{n}+N} |x_e(l)|^2 \right) x_e(k) = \|\hat{x}_e\|^2 \, x_e(k)$$

to both sides, the necessary-condition equation (6.4) becomes

$$\hat{Q}_{e,x} \, \hat{x}_e = \|\hat{x}_e\|^2 \, \hat{x}_{e,U} \tag{6.5}$$

where the (rectangular) matrix $\hat{Q}_{e,x}$ is defined as

$$\left(\hat{Q}_{e,x}\right)_{kl} = q_{e,x}^{(h)}(\hat{n}+k, \hat{n}-K+1) \quad , \quad 1 \le k \le N \; , \; 1 \le l \le K+N \; , \tag{6.6}$$

with (cf. (4.19))

$$q_{e,x}^{(h)}(k,l) = h^2(k-l) \, \tilde{q}_e(k,l) + \left[1 - h^4(k-l)\right] x_e(k) \, x_e^*(l) \; . \tag{6.7}$$

Note that $\hat{Q}_{e,x}$ contains both the known signal samples (the elements of $\hat{x}_{e,K}$) and the unknown signal samples (the elements of $\hat{x}_{e,U}$).

Again, we propose to solve (6.5) iteratively by a suitable modification of the quasi power algorithm. The i-th iteration consists of the following steps. First, using the signal vector $\hat{x}_e^{(i-1)}$ derived in the (i-1)-th iteration, the matrix $\hat{Q}_{e,x(i-1)}$ is formed by means of (6.6) and (6.7). Then, the matrix-vector product

$$\hat{v}_{e,U}^{(i)} = \hat{Q}_{e,x(i-1)} \, \hat{u}_e^{(i-1)}$$

is computed; this yields the "unknown" signal vector $\hat{x}_{e,U}^{(i)}$ up to a scaling factor. This scaling factor, however, depends on the entire signal vector $\hat{x}_e^{(i)}$. Consistent with the quasi eigen-equation (6.5), we first define the "quasi eigenvalue"

$$\lambda_e^{(i)} = \|\hat{x}_e^{(i)}\|^2 \; . \tag{6.8}$$

We then derive the vectors $\hat{u}_{e,U}^{(i)}$, $\hat{x}_{e,U}^{(i)}$, $\hat{x}_e^{(i)}$, and $\hat{u}_e^{(i)}$ from $\hat{v}_{e,U}^{(i)}$ as follows:

$$\hat{u}_{e,U}^{(i)} = \hat{v}_{e,U}^{(i)} / \lambda_e^{(i)}$$

$$\hat{x}_{e,U}^{(i)} = \sqrt{\lambda_e^{(i)}} \, \hat{u}_{e,U}^{(i)} = \hat{v}_{e,U}^{(i)} / \sqrt{\lambda_e^{(i)}}$$

$$\hat{x}_e^{(i)} = \begin{pmatrix} \hat{x}_{e,K} \\ \hat{x}_{e,U}^{(i)} \end{pmatrix} = \begin{pmatrix} \hat{x}_{e,K} \\ \hat{v}_{e,U}^{(i)} / \sqrt{\lambda_e^{(i)}} \end{pmatrix} \tag{6.9}$$

$$\hat{u}_e^{(i)} = \hat{x}_e^{(i)} / \|\hat{x}_e^{(i)}\| = \hat{x}_e^{(i)} / \sqrt{\lambda_e^{(i)}} \; .$$

Note that $\hat{x}_e^{(i)}$ and $\hat{u}_e^{(i)}$ will be used for the next iteration. The above equations

also show how $\lambda_e^{(i)}$ can be calculated: combining (6.8) and (6.9), we obtain

$$\lambda_e^{(i)} = \|\hat{\mathbf{x}}_{e,K}\|^2 + \|\hat{\mathbf{v}}_{e,U}^{(i)}\|^2 / \lambda_e^{(i)} .$$

This represents a quadratic equation in $\lambda_e^{(i)}$ which is solved by

$$\lambda_e^{(i)} = \frac{1}{2}\left(\|\hat{\mathbf{x}}_{e,K}\|^2 + \sqrt{\|\hat{\mathbf{x}}_{e,K}\|^2 + 4\|\hat{\mathbf{v}}_{e,U}^{(i)}\|^2} \right) .$$

A similar derivation can be given for the odd-indexed signal samples.

Summary of the block-by-block algorithm. Combining the results obtained for the even-indexed and odd-indexed signal samples, we can summarize the "block-by-block" algorithm for global signal synthesis in the case of the PWD as follows.

1a) With $\hat{n}$ being a given time instant, transform the model as

$$\mathcal{C}(n,m) = \frac{1}{2}\int_{-1/2}^{1/2} \tilde{T}(n,\tfrac{\Theta}{2})\, e^{j2\pi m\Theta}\, d\Theta \qquad \text{for} \quad 2(\hat{n}+N-L_S) \le n \le 2(\hat{n}+N)-1 ,$$

where N and L_S are the block length and the step length, respectively, to be explained below.

1b) Define the known vectors $\hat{\mathbf{x}}_{e,K}$, $\hat{\mathbf{x}}_{o,K}$ and unknown vectors $\hat{\mathbf{x}}_{e,U}$, $\hat{\mathbf{x}}_{o,U}$ as

$$(\hat{\mathbf{x}}_{e,K})_n \triangleq x_e(\hat{n}-K+n) , \qquad (\hat{\mathbf{x}}_{o,K})_n \triangleq x_o(\hat{n}-K+n) , \qquad 1 \le n \le K ;$$

$$(\hat{\mathbf{x}}_{e,U})_n \triangleq x_e(\hat{n}+n) , \qquad (\hat{\mathbf{x}}_{o,U})_n \triangleq x_o(\hat{n}+n) , \qquad 1 \le n \le N .$$

2) Perform the quasi power algorithm iteration: for i=1,2,...

(a) Construct the matrices $\hat{\mathbf{Q}}_{e,x(i-1)}$ and $\hat{\mathbf{Q}}_{o,x(i-1)}$ as follows:

$$\left(\hat{\mathbf{Q}}_{e,x(i-1)}\right)_{kl} = q_{e,x(i-1)}^{(h)}(\hat{n}+k, \hat{n}-K+l) , \qquad 1 \le k \le N , \quad 1 \le l \le K+N$$

$$\left(\hat{\mathbf{Q}}_{o,x(i-1)}\right)_{kl} = q_{o,x(i-1)}^{(h)}(\hat{n}+k, \hat{n}-K+l) , \qquad 1 \le k \le N , \quad 1 \le l \le K+N ,$$

where

$$q_{e,x(i-1)}^{(h)}(k,l) = h^2(k-l)\, \mathcal{C}(k+l, k-l) + \left[1 - h^4(k-l)\right] x_e^{(i-1)}(k)\, x_e^{(i-1)*}(l)$$

$$q_{o,x(i-1)}^{(h)}(k,l) = h^2(k-l)\, \mathcal{C}(k+l+1, k-l) + \left[1 - h^4(k-l)\right] x_o^{(i-1)}(k)\, x_o^{(i-1)*}(l) .$$

(b) Calculate the matrix-vector products

$$\hat{\mathbf{v}}_{e,U}^{(i)} = \hat{\mathbf{Q}}_{e,x(i-1)}\, \hat{\mathbf{u}}_e^{(i-1)} , \qquad \hat{\mathbf{v}}_{o,U}^{(i)} = \hat{\mathbf{Q}}_{o,x(i-1)}\, \hat{\mathbf{u}}_o^{(i-1)} .$$

(c) Perform normalizations and scalings:

$$\lambda_e^{(i)} = \frac{1}{2}\left(\|\hat{\mathbf{x}}_{e,K}\|^2 + \sqrt{\|\hat{\mathbf{x}}_{e,K}\|^2 + 4\|\hat{\mathbf{v}}_{e,U}^{(i)}\|^2} \right), \qquad \lambda_o^{(i)} = \frac{1}{2}\left(\|\hat{\mathbf{x}}_{o,K}\|^2 + \sqrt{\|\hat{\mathbf{x}}_{o,K}\|^2 + 4\|\hat{\mathbf{v}}_{o,U}^{(i)}\|^2} \right);$$

$$\hat{\mathbf{x}}_{e,U}^{(i)} = \hat{\mathbf{v}}_{e,U}^{(i)} / \sqrt{\lambda_e^{(i)}}, \qquad \hat{\mathbf{x}}_{o,U}^{(i)} = \hat{\mathbf{v}}_{o,U}^{(i)} / \sqrt{\lambda_o^{(i)}};$$

$$\hat{\mathbf{x}}_e^{(i)} = \begin{pmatrix} \hat{\mathbf{x}}_{e,K} \\ \hat{\mathbf{x}}_{e,U}^{(i)} \end{pmatrix}, \qquad \hat{\mathbf{x}}_o^{(i)} = \begin{pmatrix} \hat{\mathbf{x}}_{o,K} \\ \hat{\mathbf{x}}_{o,U}^{(i)} \end{pmatrix};$$

$$\hat{\mathbf{u}}_e^{(i)} = \hat{\mathbf{x}}_e^{(i)} / \sqrt{\lambda_e^{(i)}}, \qquad \hat{\mathbf{u}}_o^{(i)} = \hat{\mathbf{x}}_o^{(i)} / \sqrt{\lambda_o^{(i)}}.$$

3) After convergence ($i \to \infty$), obtain new blocks of the synthesized signal as

$$\hat{\mathbf{x}}_{e,U} = \hat{\mathbf{x}}_{e,U}^{(\infty)}, \qquad \hat{\mathbf{x}}_{o,U} = \hat{\mathbf{x}}_{o,U}^{(\infty)}.$$

4) For synthesizing the next signal blocks of even-indexed and odd-indexed signal samples, replace $\hat{n}$ by $\hat{n}+L_S$ with step length L_S not greater than the block length N, $L_S \leq N$, and go to Step 1.

Recursive continuation of block-by-block synthesis. For the synthesis of longer signals (in fact, signals with arbitrary length), the block synthesis described above must be continued in a recursive way as stated in Step 4. The synthesis of the (m+1)-th signal block is based on the m-th signal block (and possibly signal blocks lying still farther in the past) and a local time interval of the model. Hence, the block-by-block algorithm is indeed suited for the on-line synthesis of signals without any length restriction.

The recursive continuation of block-by-block synthesis is illustrated in *Figure 6*, where it has been assumed that the block length N is smaller than the one-sided PWD window length K (note, however, that $N > K$ is allowed, too). In the following, we only discuss the even-indexed signal samples since the odd case is analogous. Suppose that we have just synthesized the m-th signal block $\hat{\mathbf{x}}_{e,U,m}$ comprising signal samples $x_e(n)$ with $\hat{n}_m+1 \leq n \leq \hat{n}_m+N$; these samples are thus known. We now synthesize the (m+1)-th signal block $\hat{\mathbf{x}}_{e,U,m+1}$ with signal samples $x_e(n)$ for $\hat{n}_{m+1}+1 \leq n \leq \hat{n}_{m+1}+N$, where

$$\hat{n}_{m+1} \triangleq \hat{n}_m + L_S$$

with the *step length* L_S. The step length L_S can be freely chosen in the range $1 \leq L_S \leq N$ (according to Figure 6, a choice $L_S > N$ would cause $\hat{\mathbf{x}}_{e,K,m+1}$ to contain unknown samples). For $L_S=N$, there does not occur any overlap of successive signal blocks $\hat{\mathbf{x}}_{e,U,m}$, $\hat{\mathbf{x}}_{e,U,m+1}$. For $L_S < N$, however, successive signal blocks overlap with $L_O = N-L_S$ being the length of overlap. In the overlap case $L_S < N$, we note that the overlap interval $\hat{n}_{m+1}+1 \leq n \leq \hat{n}_m+N$ is synthesized (at least) twice since it is part of both $\hat{\mathbf{x}}_{e,U,m}$ and $\hat{\mathbf{x}}_{e,U,m+1}$. We propose to incorporate the result contained in $\hat{\mathbf{x}}_{e,U,m+1}$ in the final synthesized signal; the result contained in

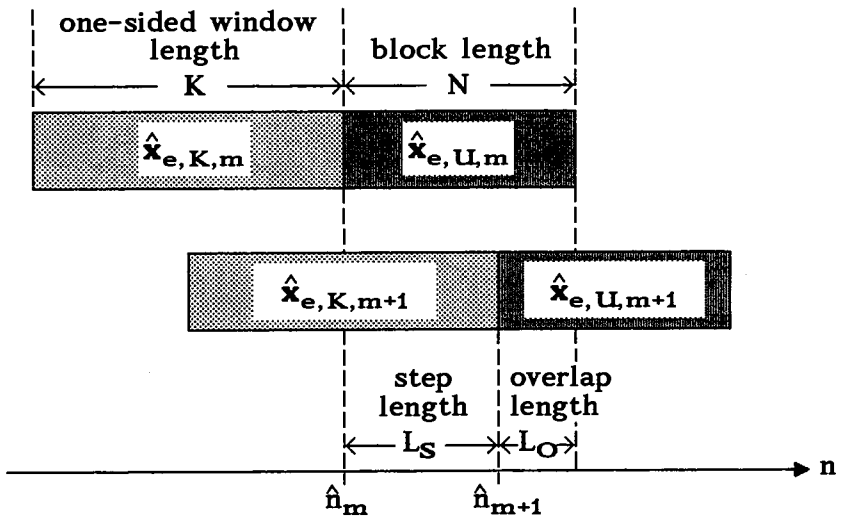

Figure 6. Recursive continuation of block-by-block synthesis.

$\hat{x}_{e,U,m}$ is then not used for the synthesized signal but may be used as part of the initial signal vector $\hat{x}^{(o)}_{e,U,m+1}$ to speed up the convergence of the quasi power iteration of the (m+1)-th signal block.

To start the block-by-block synthesis algorithm, i.e., to synthesize the very first signal block $\hat{x}_{e,U,1}$, the first "known" signal vector $\hat{x}_{e,K,1}$ (comprising signal samples $x_e(n)$ for $1 \le n \le K$) has to be initialized. Fortunately, it seems that the block-by-block algorithm is quite insensitive to initialization errors: according to our experiments, the error introduced by an even extremely faulty initialization does not propagate over more than a small number of blocks.

Some comments. (i) We stress that the block-by-block synthesis algorithm will generally produce signals which are suboptimal. In contrast to the algorithm of Subsection 4.2, the block-by-block algorithm does not attempt to minimize the total synthesis error $\varepsilon_x = \| \tilde{T} - PWD_x \|$ of the entire signal; instead, the syntheses of individual signal blocks are based on a minimization of individual *local* synthesis errors $\hat{\varepsilon}_{x_{e,U}}$, $\hat{\varepsilon}_{x_{o,U}}$ as given by (6.3). The block-by-block algorithm can thus be considered to be *locally optimal*.

(ii) It is interesting to compare the necessary-condition equations of the PWD synthesis scheme of Subsection 4.2 with the necessary-condition equations of the block-by-block synthesis scheme. In the first case, insertion of (4.19) into the "even" equation of (4.6) yields

$$\sum_{l=k-K}^{k+K} h^2(k-l) \left[\mathcal{C}(k+1,k-l) - h^2(k-l)\, x_e(k)\, x_e^*(l) \right] x_e(l) = 0 , \qquad -\infty < k < \infty ; \qquad (6.10)$$

for the block-by-block case, on the other hand, (6.4) gives

$$\sum_{l=k-K}^{\min\{k+K,\, \hat{n}+N\}} h^2(k-l) \left[\mathcal{C}(k+1,k-l) - h^2(k-l)\, x_e(k)\, x_e^*(l) \right] x_e(l) = 0 , \quad \hat{n}+1 \le k \le \hat{n}+N . \qquad (6.11)$$

Analogous equations hold for the odd-indexed samples. We see that the two equations have identical forms. However, Eq. (6.10) is valid for all k, and the summation on the left-hand side involves the even-indexed signal samples $x_e(l)$ in the range $[k-K, k+K]$. In contrast, Eq. (6.11) is valid only in the local interval $\hat{n}+1 \le k \le \hat{n}+N$, and the summation involves the even-indexed signal samples $x_e(l)$ only in the range $[k-K, \min\{k+K, \hat{n}+N\}]$ [42].

(iii) If the quasi power algorithm converges, then the resulting signal block is guaranteed to be a solution to the necessary-condition equation (6.4).

(iv) An important difference between block-by-block synthesis and the synthesis of the entire signal as discussed in Subsection 4.2 is that the block-by-block algorithm's memory and computation requirements per signal block and iteration step are independent of the overall signal length; they only depend on the block length N (for given window length $L = 2K+1$). The convergence speed for each individual block (assuming fixed block length N) again depends primarily on the amount of smoothing employed in the PWD: shorter PWD windows h(n) (corresponding to more smoothing) yield slower convergence.

(v) For a given PWD window h(n), the parameters of the block-by-block algorithm are the *block length* N and the *step length* L_S (or, equivalently, the *overlap length* $L_O = N-L_S$). These parameters control the quality of the synthesis result as well as the computational expense. Better synthesis results (i.e., results with smaller residual error, corresponding to a smaller deviation from the optimal result) will be obtained for larger block length N and/or smaller step length L_S (i.e., larger overlap length L_O). This, of course, entails more expensive computation. While the recursive block-by-block algorithm avoids discontinuities at the block boundaries, the step length is often visible in the synthesis result. This effect can be avoided by choosing the minimal step length $L_S = 1$ (see also Subsection 6.2).

(vi) Since the block-by-block algorithm is an algorithm for global signal synthesis, its result again contains arbitrary phases φ_e, φ_o of the subsequences of even-indexed and odd-indexed signal samples. These phases are determined by the initial vectors $\hat{\mathbf{x}}_{e,K,1}$ and $\hat{\mathbf{x}}_{o,K,1}$ used for starting the recursive synthesis process. They propagate from the m-th block synthesis to the (m+1)-th block and thus determine the phases of the entire synthesis result. On-line algorithms for resolving this phase ambiguity are presented in Subsection 7.4.

(vii) It seems to be difficult to develop a locally optimal on-line synthesis algorithm for *halfband* signal synthesis. However, non-optimal halfband synthesis with on-line characteristic can easily be performed by synthesizing e.g. the even-indexed samples using the block-by-block algorithm and deriving the odd-indexed samples by means of interpolation (this is analogous to the "reduced-cost" halfband signal synthesis algorithm discussed in Subsection 4.3). Since the exact interpolation is not consistent with an on-line operation, some recursive or short-time interpolation algorithm (e.g., using an FIR filter) has to be employed.

(viii) Apart from the PWD, there exist many other SWDs which can be computed on-line. Indeed, any SWD allows for on-line computation if its kernel $\varphi(n,m)$ (see Subsection 4.1) is finite-support with respect to both variables n and m (if neces-

sary, this can be enforced by a truncation or windowing). Unfortunately, the development of locally optimal on-line synthesis algorithms seems to be difficult in the general SWD case. The reason for this is the time-smoothing in the SWD which causes a coupling of signal blocks. Here, the PWD is an exception since it does not contain a time smoothing.

6.2 Sample-by-Sample Synthesis

The computational expense and memory requirements of the block-by-block algorithm depend on the block length N and the step length L_S. A minimum-cost algorithm is obtained by choosing block length N=1. This obviously entails step length L_S=1 and zero overlap, L_O=0. The signal blocks $\hat{\mathbf{x}}_{e,U}$, $\hat{\mathbf{x}}_{o,U}$ here reduce to single signal samples, and the signal is thus synthesized *sample by sample* [37,39].

The case N=1 is particularly simple since it allows a closed-form solution and thus avoids an iterative scheme like the quasi power algorithm. To show this, we consider the synthesis of even-indexed samples; the odd case is again analogous. Suppose that the even-indexed signal samples $x_e(n)$ have already been synthesized up to the time instant $\hat{n}$. The next signal sample $x_e(\hat{n}+1)$ is then synthesized such that the $(\hat{n}+1)$-th error contribution $\varepsilon_{e,x_e}(\hat{n}+1)$ (see (6.2)) is minimized,

$$\varepsilon_{e,x_e}(\hat{n}+1) \longrightarrow \min_{x_e(\hat{n}+1)} .$$

Setting the derivative of $\varepsilon^2_{e,x_e}(\hat{n}+1)$ with respect to the single variable $x_e(\hat{n}+1)$ equal to zero, we obtain as a necessary condition the single equation

$$\sum_{l=\hat{n}-K+1}^{\hat{n}+1} h^2(\hat{n}+1-l) \left[\tilde{q}_e(\hat{n}+1,l) - h^2(\hat{n}+1-l)\, x_e(\hat{n}+1)\, x_e^*(l) \right] x_e(l) = 0 \quad . \tag{6.12}$$

This is a third-order equation in the single variable $x_e(\hat{n}+1)$. According to (6.12), the magnitude and phase of $x_e(\hat{n}+1)$ are determined by

$$|x_e(\hat{n}+1)|^3 + p\,|x_e(\hat{n}+1)| - |q| = 0 \quad , \tag{6.13}$$

$$\arg\{x_e(\hat{n}+1)\} = \arg\{q\} - \arg\{p+|x_e(\hat{n}+1)|^2\} \quad , \tag{6.14}$$

where we have defined the known quantities

$$p \triangleq \sum_{l=\hat{n}-K+1}^{\hat{n}} |x_e(l)\, h^2(\hat{n}+1-l)|^2 - \tilde{q}_e(\hat{n}+1,\hat{n}+1)$$

$$q \triangleq \sum_{l=\hat{n}-K+1}^{\hat{n}} \tilde{q}_e(\hat{n}+1,l)\, h^2(\hat{n}+1-l)\, x_e(l) \; .$$

It follows from $p \in \mathbb{R}$ and $|q| > 0$ that (6.13) has a unique real-valued and non-negative

solution

$$|x_e(\hat{n}+1)| = \sqrt[3]{a+b} + \sqrt[3]{a-b} \qquad \text{with} \qquad a \triangleq \frac{|q|}{2} \;,\; b \triangleq \sqrt{\frac{|q|^2}{4}+\frac{p^3}{27}} \;. \tag{6.15}$$

The synthesis solution $x_e(\hat{n}+1)$ is finally given by (6.15) and (6.14).

6.3 Simulation Results

Figure 7 demonstrates the performance of the block-by-block synthesis algorithm discussed in Subsection 6.1. The length of the PWD window h(n) is $L = 2K+1 = 63$. The block length was chosen as N=8. Two step lengths are compared: maximal step length $L_S = N = 8$ (Figure 7.b) and minimal step length $L_S=1$ (Figure 7.c). The recursive synthesis process was initialized with an estimate of the first 16 signal samples which was obtained by applying off-line PWD synthesis (cf. Subsection 4.2) to the corresponding initial model block. Note that, in the $L_S=8$ case, the step length is visible in the synthesized signal via a parasitic amplitude modulation (see Figure 7.b.2).

A synthesis experiment using the sample-by-sample synthesis algorithm discussed in Subsection 6.2 is shown in *Figure 8*. The PWD window length is again $L = 2K+1 = 63$. Each of the recursive synthesis processes of even-indexed and odd-indexed samples was initialized with $K = 31$ zero samples. It is seen that, in spite of this extremely faulty initialization, the synthesized signal rapidly adapts to the model.

In both of the experiments shown, even-indexed and odd-indexed signal samples were synthesized separately, and phase matching was performed by means of the on-line "spectral spread algorithm" discussed in Subsection 7.4, with phase matching performed simultaneously with the synthesis of each signal block or signal sample (in the case of iterative block-by-block synthesis, phase matching was performed for each iteration step separately) [37].

7. PHASE MATCHING ALGORITHMS

It has been shown in the Introduction that, for any BTFR, the result of signal synthesis invariably contains a phase ambiguity: the synthesized signal is unique only up to a constant phase factor. An additional, more troublesome phase ambiguity exists in the case of WD, SWD, and AF1 if *global* signal synthesis is performed. Here, the subsequences of even-indexed and odd-indexed signal samples contain separate phase ambiguities; these can be represented in terms of an "absolute" phase ambiguity affecting the entire signal and a "relative" phase ambiguity of even-indexed and odd-indexed signal samples *relative to each other*. The relative phase ambiguity is much more awkward than the absolute phase ambiguity since it generally produces severe signal distortion. We have seen, however, that the relative phase ambiguity can be avoided by performing halfband signal synthesis instead of global signal synthesis.

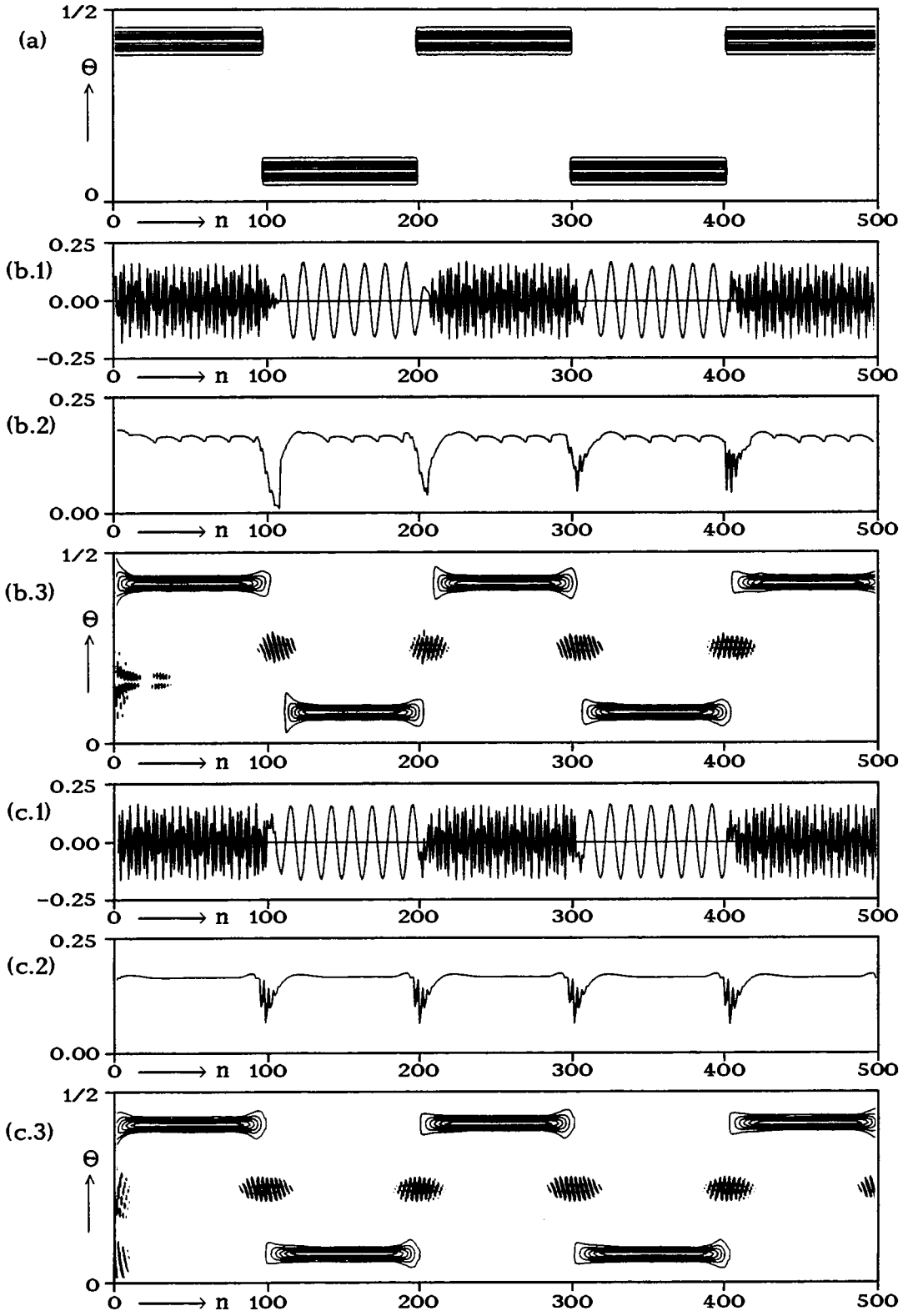

Figure 7. PWD on-line signal synthesis using the block-by-block algorithm. (a) Model; (b) synthesized signal for L_S=8: (b.1) real part, (b.2) envelope, and (b.3) PWD; (c) synthesized signal for L_S=1: (c.1) real part, (c.2) envelope, and (c.3) PWD.

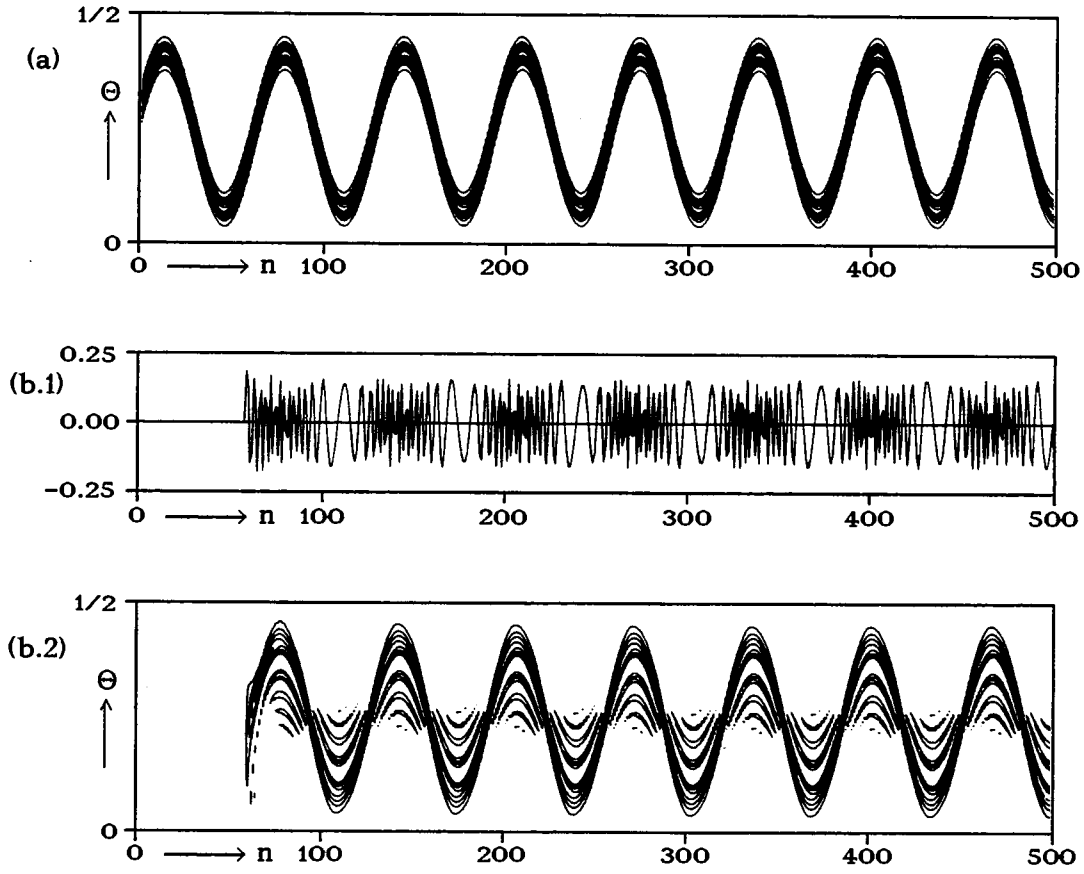

Figure 8. PWD on-line signal synthesis using the sample-by-sample algorithm. (a) Model; (b) synthesized signal: (b.1) real part and (b.2) PWD.

The present section discusses various methods for *phase matching*, i.e., resolving the phase ambiguities of the synthesized signal [44,37,10]. These methods can be grouped into two categories: firstly, *reference-based* phase matching where the phases are matched to a reference signal, and secondly, *autonomous* phase matching which does not require a reference signal. Reference-based phase matching is capable of resolving both the relative and the absolute phase ambiguity while autonomous phase matching only resolves the relative phase ambiguity. Autonomous phase matching is useful in the case of global signal synthesis if a meaningful reference signal is not available and, hence, reference-based phase matching cannot be performed.

This section is organized as follows. After a discussion and analysis of absolute and relative phase ambiguities in Subsection 7.1, Subsection 7.2 considers two algorithms for reference-based phase matching. Subsection 7.3 then presents two approaches to autonomous phase matching and shows that these approaches can be generalized, resulting in a unified mathematical framework for autonomous phase matching. "On-line" versions of the reference-based and autonomous phase matching

algorithms are discussed in Subsection 7.4. These on-line algorithms feature a short-time or causal mode of operation and thus permit the on-line processing of signals with arbitrary length. Simulation results are presented in Subsection 7.5.

7.1 Absolute and Relative Phase Ambiguities

For any BTFR, and for both global signal synthesis and subspace signal synthesis on arbitrary signal subspaces, the result of signal synthesis is always ambiguous with respect to a constant phase. Indeed, if x(n) is a solution of signal synthesis, then $e^{j\varphi}\,x(n)$, where φ is an arbitrary constant phase, is a solution as well. This phase ambiguity of signal synthesis is caused by the invariance of BTFRs to constant phase factors (see (1.12), (1.13)).

We now consider global signal synthesis in the case of the WD. Let x(n) be some signal, and let us derive from x(n) a new, "phase-rotated" signal $\tilde{x}(n)$ by applying constant phase factors $e^{j\varphi_e}$, $e^{j\varphi_o}$ to the subsequences of even-indexed and odd-indexed signal samples, respectively,

$$\tilde{x}(n) = \tilde{x}^{(\varphi_e,\varphi_o)}(n) \triangleq e^{j\varphi_e}\,x_{\mathcal{E}}(n) + e^{j\varphi_o}\,x_{\mathcal{O}}(n) = \begin{cases} e^{j\varphi_e}\,x(n), & n=2k \\ e^{j\varphi_o}\,x(n), & n=2k+1 \end{cases}, \tag{7.1}$$

where the signals $x_{\mathcal{E}}(n)$ and $x_{\mathcal{O}}(n)$ have been defined in Subsection 3.1. Alternatively, the phase-rotated signal $\tilde{x}(n)$ may also be written as

$$\tilde{x}(n) = \tilde{x}^{(\varphi,\psi)}(n) = e^{j\varphi}\,\tilde{x}^{(\psi)}(n)$$

with

$$\tilde{x}^{(\psi)}(n) = x_{\mathcal{E}}(n) + e^{j\psi}x_{\mathcal{O}}(n) = \begin{cases} x(n), & n=2k \\ e^{j\psi}x(n), & n=2k+1 \end{cases}. \tag{7.2}$$

The "absolute phase" $\varphi=\varphi_e$ affects the entire signal whereas the "relative phase" $\psi=\varphi_o-\varphi_e$ describes a phase rotation of even-indexed and odd-indexed samples *relative to each other*. Now, inserting into the WD definition (1.4), it can easily be shown that the WD is invariant to arbitrary phase rotation of the subsequences of even-indexed and odd-indexed signal samples,

$$WD_{\tilde{x}}(n,\Theta) \equiv WD_x(n,\Theta)\ . \tag{7.3}$$

Then, combining (7.3) and the definition (3.13) of the synthesis error ε_x, it follows that the original signal x(n) and any "phase-rotated" version $\tilde{x}(n)$ achieve the same synthesis error,

$$\varepsilon_{\tilde{x}} = \|\tilde{T} - WD_{\tilde{x}}\| = \|\tilde{T} - WD_x\| = \varepsilon_x\ .$$

Therefore, if x(n) is a solution to the global WD signal synthesis problem (3.13), then $\tilde{x}(n)$ is a solution as well. We conclude that the result of global WD signal

synthesis is *ambiguous* with respect to the phases φ_e and φ_o or, equivalently, φ ("absolute phase ambiguity") and ψ ("relative phase ambiguity"). On practical application of WD signal synthesis, these phases will assume arbitrary values. An analogous argument shows that the same phase ambiguities exist in the case of SWD and AF1.

We have thus seen that (i) for any BTFR, the result of global or subspace signal synthesis contains an absolute phase ambiguity and (ii) for WD, SWD, and AF1, the result of global signal synthesis contains both an absolute and a relative phase ambiguity. Now, while the absolute phase ambiguity merely results in a phase factor affecting the entire signal, the relative phase ambiguity obviously produces severe signal distortion. In the frequency domain, this distortion can be interpreted as an aliasing effect. Indeed, transforming $\tilde{x}^{(\psi)}(n)$ into the frequency domain and inserting the relations

$$X_{\mathfrak{E}}(\Theta) = \tfrac{1}{2}[\,X(\Theta)+X(\Theta-1/2)\,] \quad , \qquad X_{\mathfrak{O}}(\Theta) = \tfrac{1}{2}[\,X(\Theta)-X(\Theta-1/2)\,] \tag{7.4}$$

(cf. (3.11)), the Fourier transform of $\tilde{x}^{(\psi)}(n)$ is obtained as

$$\tilde{X}^{(\psi)}(\Theta) = c_+ X(\Theta) + c_- X(\Theta-1/2) \qquad \text{with} \qquad c_\pm = \tfrac{1}{2}\left(1 \pm e^{j\psi}\right) . \tag{7.5}$$

The troublesome relative phase ambiguity can be avoided by performing halfband signal synthesis instead of global signal synthesis. Indeed, if the result of signal synthesis is constrained to be a halfband signal, then it is unique apart from an absolute phase. This directly follows from the aliasing relation (7.5): if x(n) is a halfband signal, then $\tilde{x}^{(\psi)}(n)$ can only be a halfband signal (on the *same* halfband) if $\psi = 0$. Alternatively, it follows from the interpolation formula (3.8) of halfband signals that even-indexed and odd-indexed signal samples are strictly coupled, and a relative phase rotation is hence incompatible with a signal's halfband property. We stress, however, that the absolute phase ambiguity is not removed by a halfband constraint.

7.2 Reference-Based Phase Matching

Matching of absolute phase. We first consider the problem of matching a signal's absolute phase to a reference signal y(n) which we suppose to be given. Let x(n) be the result of signal synthesis. To match the absolute phase, we form the phase-rotated version $e^{j\varphi}x(n)$ and choose the absolute phase φ such that the resulting signal is as close to the reference signal y(n) as possible, which amounts to the minimization

$$\varepsilon(\varphi) \;\triangleq\; \|y - e^{j\varphi}x\| \;\longrightarrow\; \min_{\varphi} . \tag{7.6}$$

Developing the squared norm $\varepsilon^2(\varphi)$ as

$$\varepsilon^2(\varphi) = \|y - e^{j\varphi}x\|^2 = \|y\|^2 + \|x\|^2 - 2\,\mathrm{Re}\left\{e^{-j\varphi}(y,x)\right\} ,$$

it follows that the minimum of $\varepsilon^2(\varphi)$ is obtained by maximizing

$$\mathrm{Re}\left\{e^{-j\varphi}(y,x)\right\} = |(y,x)|\cos\left[\arg\{(y,x)\} - \varphi\right];$$

thus, the optimal phase is given by

$$\varphi_{opt} = \arg C_R \qquad \text{with} \qquad C_R = (y,x) = \sum_n y(n)\,x^*(n)\ . \tag{7.7}$$

Matching of absolute and relative phase [10]. For matching both the absolute and the relative phase or, equivalently, the phases of even-indexed and odd-indexed signal samples, we form the phase-rotated signal $\mathfrak{x}^{(\varphi_e,\varphi_o)}(n)$ according to (7.1) and choose the phases φ_e, φ_o such that

$$\varepsilon(\varphi_e,\varphi_o) \triangleq \|y - \mathfrak{x}^{(\varphi_e,\varphi_o)}\| \longrightarrow \min_{\varphi_e,\varphi_o}\ ,$$

where y(n) is again a reference signal. Inserting (7.1) and the decomposition $y(n)=y_{\mathfrak{E}}(n)+y_{\mathfrak{O}}(n)$, and separating even-indexed and odd-indexed signal samples, it is easily seen that the squared error $\varepsilon^2(\varphi_e,\varphi_o)$ can be decomposed as

$$\varepsilon^2(\varphi_e,\varphi_o) = \varepsilon^2_{\mathfrak{E}}(\varphi_e) + \varepsilon^2_{\mathfrak{O}}(\varphi_o)$$

with

$$\varepsilon_{\mathfrak{E}}(\varphi_e) = \|y_{\mathfrak{E}} - e^{j\varphi_e}x_{\mathfrak{E}}\|\ , \qquad \varepsilon_{\mathfrak{O}}(\varphi_o) = \|y_{\mathfrak{O}} - e^{j\varphi_o}x_{\mathfrak{O}}\|\ .$$

The error components $\varepsilon_{\mathfrak{E}}(\varphi_e)$ and $\varepsilon_{\mathfrak{O}}(\varphi_o)$ can thus be minimized separately. Reasoning as above, the optimal phases are then obtained as [10]

$$\varphi_{e,opt} = \arg C_{R,e} \qquad \text{with} \qquad C_{R,e} = (y_{\mathfrak{E}}, x_{\mathfrak{E}}) = \sum_k y(2k)\,x^*(2k)\ , \tag{7.8}$$

$$\varphi_{o,opt} = \arg C_{R,o} \qquad \text{with} \qquad C_{R,o} = (y_{\mathfrak{O}}, x_{\mathfrak{O}}) = \sum_k y(2k+1)\,x^*(2k+1)\ . \tag{7.9}$$

7.3 Autonomous Phase Matching

We now turn to the problem of resolving the *relative* phase ambiguity without the help of a reference signal. Indeed, when no reference signal is available, there generally does not exist a reasonable criterion for determining the *absolute* phase; the absolute phase ambiguity must therefore remain unresolved. We take account of this fact by forming the phase-rotated signal $\mathfrak{x}^{(\psi)}(n)$ as in (7.2). The relative phase ψ has now to be adjusted according to some optimality criterion. Two different approaches [44,37] are discussed below. In the following, we assume that x(n) is the result of global signal synthesis in the case of the WD. Note, however, that the methods derived also apply to the SWD and AF1 for which similar arguments can be given.

The halfband approximation criterion [44]. In practice, the WD is applied only to signals which are halfband (or at least nearly halfband) so that no substantial aliasing of the WD occurs. Thus, it is natural to require that the signal synthesis result, too, should be halfband or at least *nearly* halfband. The latter extension is important in the context of global signal synthesis since in general (if the model is not halfband-consistent as explained in Subsection 3.3) no choice of the phase ψ can be found such that the resulting (phase-matched) signal $\underline{x}^{(\psi)}(n)$ is exactly halfband. We therefore adjust the relative phase ψ such that the resulting signal $\underline{x}^{(\psi)}(n)$ is *as nearly halfband as possible*; this will be termed the *halfband approximation criterion* (HAC). Phrased mathematically, the HAC requires that the distance of $\underline{x}^{(\psi)}(n)$ from a halfband subspace $\mathfrak{H}$ (with given center frequency Θ_0) be minimal. As shown in *Figure 9*, this distance is given by $d(\psi) = \|\underline{x}^{(\psi)} - \underline{x}_{\mathfrak{H}}^{(\psi)}\|$, where $\underline{x}_{\mathfrak{H}}^{(\psi)}(n)$ is the orthogonal projection of $\underline{x}^{(\psi)}(n)$ on $\mathfrak{H}$. Thus, the HAC reads

$$d(\psi) \triangleq \|\underline{x}^{(\psi)} - \underline{x}_{\mathfrak{H}}^{(\psi)}\| \longrightarrow \min_{\psi} .$$

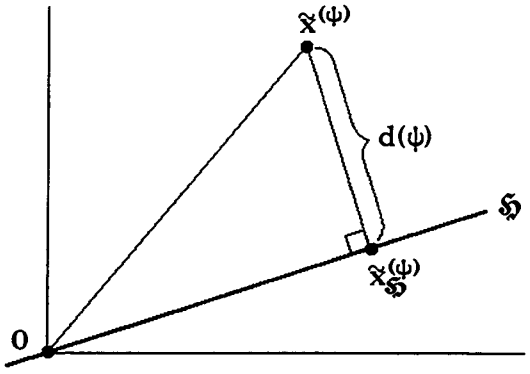

Figure 9. Distance of signal $\underline{x}^{(\psi)}(n)$ from halfband subspace $\mathfrak{H}$.

Using (3.3), Parseval's relation, and (3.5), the squared norm $d^2(\psi)$ can be expressed in the frequency domain as

$$d^2(\psi) = \|\underline{x}^{(\psi)} - \underline{x}_{\mathfrak{H}}^{(\psi)}\|^2 = \int_{-1/2}^{1/2} \left|[1-H(\Theta)]\,\underline{\tilde{X}}^{(\psi)}(\Theta)\right|^2 d\Theta = \int_{-1/2}^{1/2} [1-H(\Theta)]\,|\underline{\tilde{X}}^{(\psi)}(\Theta)|^2 d\Theta =$$

$$= \int_{-1/2}^{1/2} |\underline{\tilde{X}}^{(\psi)}(\Theta)|^2 d\Theta - \int_{-1/2}^{1/2} H(\Theta)\,|\underline{\tilde{X}}^{(\psi)}(\Theta)|^2 d\Theta \quad . \tag{7.10}$$

The first term in (7.10) is the energy of $\underline{x}^{(\psi)}(n)$ which is easily shown to be independent of ψ. Thus it remains to maximize the second term,

$$m_H(\psi) \triangleq \int_{-1/2}^{1/2} H(\Theta)\,|\underline{\tilde{X}}^{(\psi)}(\Theta)|^2 d\Theta = \int_{\Theta_0-1/4}^{\Theta_0+1/4} |\underline{\tilde{X}}^{(\psi)}(\Theta)|^2 d\Theta \longrightarrow \max_{\psi} . \tag{7.11}$$

This amounts to a maximization of the signal's energy inside the halfband interval $\Theta_0-1/4 < \Theta < \Theta_0+1/4$.

The spectral spread criterion [44]. The HAC has just been shown to maximize the energy of $\tilde{x}^{(\psi)}(n)$ inside the halfband $\Theta_0-1/4 < \Theta < \Theta_0+1/4$. This can be approximated by maximizing the concentration of the Fourier transform $\tilde{X}^{(\psi)}(\Theta)$ relative to the halfband's center frequency Θ_0, which is done by minimizing the *spread* of $\tilde{X}^{(\psi)}(\Theta)$ about Θ_0. This leads to the *spectral spread criterion* (SSC)

$$\sigma^2(\psi) \triangleq \frac{\int_{\Theta_0-1/2}^{\Theta_0+1/2} \rho(\Theta)\,|\tilde{X}^{(\psi)}(\Theta)|^2\,d\Theta}{\int_{\Theta_0-1/2}^{\Theta_0+1/2} |\tilde{X}^{(\psi)}(\Theta)|^2\,d\Theta} \longrightarrow \min_{\psi} , \tag{7.12}$$

where the weighting function $\rho(\Theta)$ has yet to be specified. While the conventional definition of spread would use $\rho(\Theta)=(\Theta-\Theta_0)^2$, we here have to remain within the framework of discrete-time signals whose Fourier transforms are 1-periodic functions. Hence, we choose the 1-periodic weighting function

$$\rho(\Theta) \triangleq [\sin \pi(\Theta-\Theta_0)]^2 = \tfrac{1}{2}[1 - \cos 2\pi(\Theta-\Theta_0)] = 1 - S(\Theta)$$

with

$$S(\Theta) = \tfrac{1}{2}[1 + \cos 2\pi(\Theta-\Theta_0)] , \tag{7.13}$$

which is shown in *Figure 10.*

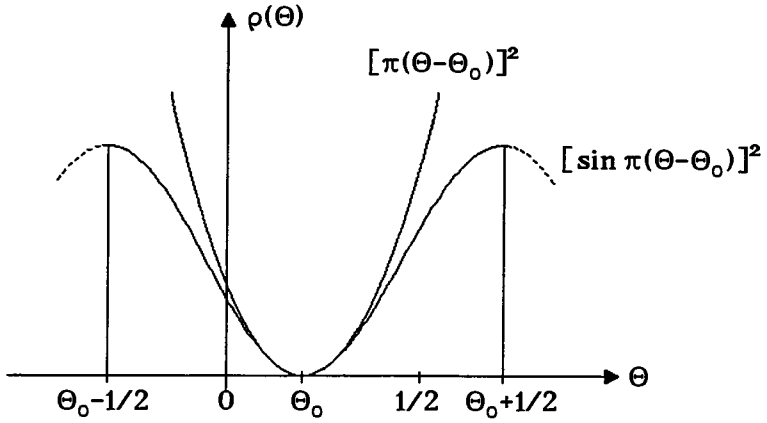

Figure 10. Weighting function for the definition of spectral spread.

In order to minimize $\sigma^2(\psi)$, we first note that the denominator of (7.12) is the energy of $\tilde{x}^{(\psi)}(n)$ which is independent of the phase ψ. Thus, there remains to

minimize the numerator of (7.12),

$$\int_{\Theta_0-1/2}^{\Theta_0+1/2} \rho(\Theta)\,|\hat{X}^{(\Psi)}(\Theta)|^2\,d\Theta = \int_{-1/2}^{1/2} [1-S(\Theta)]\,|\hat{X}^{(\Psi)}(\Theta)|^2\,d\Theta =$$

$$= \int_{-1/2}^{1/2} |\hat{X}^{(\Psi)}(\Theta)|^2\,d\Theta - \int_{-1/2}^{1/2} S(\Theta)\,|\hat{X}^{(\Psi)}(\Theta)|^2\,d\Theta \ . \qquad (7.14)$$

Since the first term of (7.14) is again independent of ψ, there finally remains to maximize the second term,

$$m_S(\psi) \triangleq \int_{-1/2}^{1/2} S(\Theta)\,|\hat{X}^{(\Psi)}(\Theta)|^2\,d\Theta \longrightarrow \max_{\Psi} \ . \qquad (7.15)$$

A generalized criterion for autonomous phase matching. Comparing (7.11) and (7.15), we notice that both the HAC and the SSC require the maximization of a frequency-domain moment,

$$m_F(\psi) \triangleq \int_{-1/2}^{1/2} F(\Theta)\,|\hat{X}^{(\Psi)}(\Theta)|^2\,d\Theta \longrightarrow \max_{\Psi} \ , \qquad (7.16)$$

where the weighting function $F(\Theta)$ depends on the criterion used: in the HAC case, $F(\Theta)=H(\Theta)$ has rectangular shape; in the SSC case, $F(\Theta)=S(\Theta)$ is sinusoidal. The weighting functions are similar, though, since they both tend to suppress the signal's Fourier transform outside the halfband $\Theta_0-1/4 < \Theta < \Theta_0+1/4$ while emphasizing it inside this halfband (see *Figure 11*). Indeed, Eq. (7.16) can be viewed as a generalized criterion for autonomous phase matching; the shape of the weighting function $F(\Theta)$ is here arbitrary apart from the requirement that $F(\Theta)$ attenuates (emphasizes) the signal's Fourier transform outside (inside) the specified halfband.

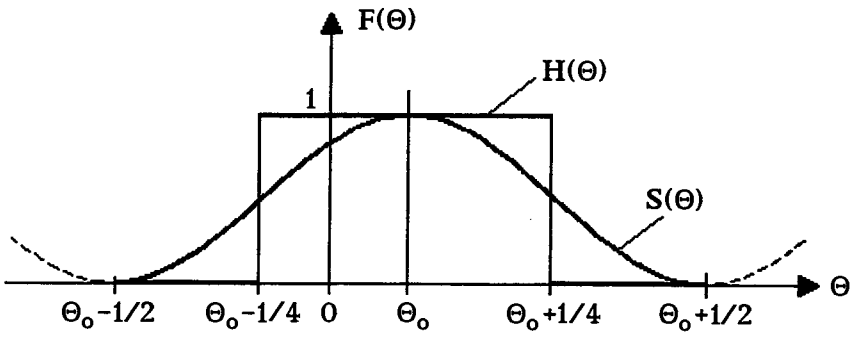

Figure 11. Weighting functions of HAC and SSC.

We now derive the solution to the general maximization problem (7.16). Inserting the frequency-domain version of (7.2), $\hat{X}^{(\Psi)}(\Theta) = X_{\mathcal{C}}(\Theta) + e^{j\Psi}X_{\mathcal{D}}(\Theta)$, we obtain

$$m_F(\psi) = \int_{-1/2}^{1/2} F(\Theta)\,|X_{\mathcal{C}}(\Theta)|^2\,d\Theta + \int_{-1/2}^{1/2} F(\Theta)\,|X_{\mathcal{D}}(\Theta)|^2\,d\Theta + \\ + \int_{-1/2}^{1/2} F(\Theta)\,2\,\mathrm{Re}\left\{ X_{\mathcal{C}}(\Theta)\left[e^{j\psi} X_{\mathcal{D}}(\Theta)\right]^*\right\} d\Theta \,. \tag{7.17}$$

The first two components of (7.17) do not depend on ψ; thus, there remains to maximize the last component which, assuming $F(\Theta)$ to be real-valued, becomes

$$2\,\mathrm{Re}\left\{ e^{-j\psi}\, C_F \right\} = 2\,|C_F|\cos(-\psi + \arg C_F) \tag{7.18}$$

with

$$C_F \triangleq \int_{-1/2}^{1/2} F(\Theta)\, X_{\mathcal{C}}(\Theta)\, X_{\mathcal{D}}^*(\Theta)\, d\Theta \,. \tag{7.19}$$

The phase maximizing (7.18) is thus obtained as

$$\psi_{opt} = \arg C_F \,. \tag{7.20}$$

We note that a time-domain expression for C_F can be derived by means of Parseval's relation,

$$C_F = (x_{\mathcal{C}F}, x_{\mathcal{D}}) = \sum_k x_{\mathcal{C}F}(2k+1)\, x^*(2k+1) = (x_{\mathcal{C}F\mathcal{D}}, x_{\mathcal{D}}) \,, \tag{7.21}$$

where $x_{\mathcal{C}F}(n)$ denotes the result of filtering $x_{\mathcal{C}}(n)$ with a filter with frequency response $F(\Theta)$.

The practical computation of C_F naturally depends on the specific weighting function $F(\Theta)$. Specializing to the weighting functions $H(\Theta)$ and $S(\Theta)$ defined by the HAC and the SSC, respectively, we obtain the following results.

The halfband approximation algorithm. For $F(\Theta)=H(\Theta)$, $C_F=C_H$ can be expressed as

$$C_H = \int_{\Theta_0-1/4}^{\Theta_0+1/4} Y(\Theta)\,d\Theta \qquad \text{with} \qquad Y(\Theta) \triangleq \tfrac{1}{4}\left[X(\Theta)+X(\Theta-\tfrac{1}{2})\right]\left[X(\Theta)-X(\Theta-\tfrac{1}{2})\right]^* .$$

This follows upon insertion of (7.4) and (3.5) into (7.19).

The spectral spread algorithm. For $F(\Theta)=S(\Theta)$, $C_F=C_S$ is given by

$$C_S = \tfrac{1}{4}\sum_k \left[e^{-j2\pi\Theta_0}\, x(2k+2) + e^{j2\pi\Theta_0}\, x(2k) \right] x^*(2k+1) \,.$$

This is obtained from (7.21) and (7.13) after straightforward manipulation.

Some comments. In a sense, the halfband approximation algorithm (HAA) with $F(\Theta)=H(\Theta)$ is optimal since it produces the signal which is "as nearly halfband as possible" and thus causes minimal aliasing in the WD. The spectral spread algorithm (SSA) is only an approximation but has the practical advantage of being considerably

less expensive than the HAA since it does not require the computation of a Fourier transform. Anyway, the results of the HAA and SSA may be expected to be very similar in many cases since the weighting functions of the HAA and SSA have similar overall characteristics. This property has been confirmed by experiments (see Subsection 7.5).

The results of the HAA and SSA are *identical* in the case of a *halfband-consistent* model (cf. Subsection 3.3). In this case, there exists a solution $x^{(\mathfrak{H})}(n)$ of *global* signal synthesis which is a halfband signal and equals the result of *halfband* signal synthesis. However, the global signal synthesis algorithm will generally not yield this specific solution $x^{(\mathfrak{H})}(n)$ but some other solution with relative phase mismatch. It can then be shown [44] that the generalized algorithm (7.19)-(7.21) for autonomous phase matching (with mild assumptions regarding the weighting function $F(\Theta)$) always recovers the desired halfband solution $x^{(\mathfrak{H})}(n)$, i.e., it removes the relative phase mismatch contained in the solution of global signal synthesis. Of course, in practice a model will rarely be exactly halfband-consistent. Still, the fact that both the HAA and SSA are consistent with halfband signal synthesis in the above sense seems to indicate that, in the practically important case of *nearly* halfband consistent models, the solutions obtained by the HAA and SSA will be very similar and, in particular, close to the solution of halfband signal synthesis.

There exists an interesting formal relation between reference-based phase matching on the one hand and autonomous phase matching on the other. According to (7.9), reference-based phase matching calculates the phase of odd-indexed signal samples as $\varphi_{o,opt} = \arg(y_{\mathfrak{D}}, x_{\mathfrak{D}})$ while autonomous phase matching calculates the relative phase ψ (which, up to the absolute phase, can be interpreted as the phase of odd-indexed signal samples φ_o) as $\psi_{opt} = \arg(x_{\mathfrak{C}F\mathfrak{D}}, x_{\mathfrak{D}})$ (see (7.20), (7.21)). As far as the relative phase is concerned, autonomous phase matching can thus be interpreted as reference-based phase matching with reference signal $y(n) = x_{\mathfrak{C}F}(n)$.

7.4 On-Line Algorithms

All phase matching agorithms discussed so far use the entire synthesis result x(n) to derive the optimal phase or phases for phase matching. This mode of operation conforms to signal synthesis algorithms which synthesize all signal samples simultaneously. However, it is not suited for the on-line synthesis algorithms presented for the PWD in Section 6. These on-line algorithms synthesize successive signal blocks or signal samples one after the other; this calls for phase matching algorithms which are compatible with on-line processing in the sense that they operate only on local or causal segments of the synthesis result. In the following, we develop on-line algorithms for both reference-based and autonomous phase matching [44,37].

Reference-based phase matching - absolute phase. We first reconsider the problem of matching the absolute phase φ of the signal synthesis result x(n) to a reference signal y(n). At time n, we want to calculate a *local estimate* $\hat{\varphi}(n)$ of the

optimal phase φ_{opt} defined by (7.7). With this local estimate, the n-th sample of the phase-matched signal is then formed according to

$$\hat{x}(n) \triangleq e^{j\hat{\varphi}(n)} x(n) \ .$$

Note that, in general, different phases $\hat{\varphi}(n)$ are used for different samples $\hat{x}(n)$.

To be compatible with on-line processing, we assume that, at time n, the signals x(k) and y(k) are known only for $k \le n+N$, where $N \ge 0$ is a fixed parameter. It is then natural to define the local phase estimate $\hat{\varphi}(n)$ at a given time instant n as the solution to the local minimization problem

$$\varepsilon_n^2(\varphi) = \| y - e^{j\varphi} x \|_{w_n}^2 \triangleq \sum_k w(k-n) \, |y(k) - e^{j\varphi} x(k)|^2 \longrightarrow \min_{\varphi} \ , \qquad (7.22)$$

where w(k) is some non-negative window satisfying $w(k) = 0$ for $k > N$. We note that (7.22) is simply a local or windowed version of the error norm (7.6). Due to the windowing, only samples of x(k) and y(k) with $k \le n+N$ are contained in this local error. It is easily shown that the solution to the minimization problem (7.22) is

$$\hat{\varphi}(n) = \arg \hat{C}_R(n) \qquad \text{with} \qquad \hat{C}_R(n) = (y, x)_{w_n} = \sum_k w(k-n) \, y(k) \, x^*(k) \ .$$

Again, $\hat{C}_R(n)$ contains only samples of x(k) and y(k) with $k \le n+N$. Note that the above result for on-line phase matching is analogous to the result for off-line phase matching as given by (7.7); the only difference is the local windowing contained in the inner product.

Reference-based phase matching - absolute and relative phase. For matching both the absolute and the relative phase to a reference signal, we proceed analogously by calculating local estimates $\hat{\varphi}_e(n)$ and $\hat{\varphi}_o(n)$ of the optimal phases $\varphi_{e,opt}$ and $\varphi_{o,opt}$ defined by (7.8) and (7.9), respectively. We then form the phase-matched signal according to

$$\hat{x}(n) \triangleq e^{j\hat{\varphi}_e(n)} x_{\mathcal{E}}(n) + e^{j\hat{\varphi}_o(n)} x_{\mathcal{O}}(n) \ . \qquad (7.23)$$

The local phase estimates $\hat{\varphi}_e(n)$, $\hat{\varphi}_o(n)$ at a given time instant n are again defined as the solution to the local minimization problem

$$\varepsilon_n^2(\varphi_e, \varphi_o) = \| y - \hat{x}^{(\varphi_e, \varphi_o)} \|_{w_n}^2 \triangleq \sum_k w(k-n) \, |y(k) - \hat{x}^{(\varphi_e, \varphi_o)}(k)|^2 \longrightarrow \min_{\varphi_e, \varphi_o} \ .$$

Here, $\hat{x}^{(\varphi_e, \varphi_o)}$ is defined by (7.1), and the window w(k) is as before. The resulting phase estimates are given by

$$\hat{\varphi}_e(n) = \arg \hat{C}_{R,e}(n) \ , \quad \hat{C}_{R,e}(n) = (y_{\mathcal{E}}, x_{\mathcal{E}})_{w_n} = \sum_k w(2k-n) \, y(2k) \, x^*(2k) \ , \qquad (7.24)$$

$$\hat{\varphi}_o(n) = \arg \hat{C}_{R,o}(n) \ , \quad \hat{C}_{R,o}(n) = (y_{\mathcal{O}}, x_{\mathcal{O}})_{w_n} = \sum_k w(2k+1-n) \, y(2k+1) \, x^*(2k+1) \ . \qquad (7.25)$$

Note that the phase estimates $\hat{\varphi}_e(n)$ and $\hat{\varphi}_o(n)$ are used in (7.23) only for n even and n odd, respectively.

Autonomous phase matching. We next develop an on-line version of the generalized algorithm for autonomous phase matching described in Subsection 7.3. For this, we have to calculate a time-varying estimate $\hat{\psi}(n)$ of the optimal relative phase ψ_{opt} as given by (7.20) and (7.21). With this estimate $\hat{\psi}(n)$, the phase-rotated signal is then formed as (a modified approach is proposed in [37])

$$\hat{x}(n) \;\triangleq\; x_{\mathcal{C}}(n) + e^{j\hat{\psi}(n)}\, x_{\mathcal{D}}(n) \;.$$

The phase estimate $\hat{\psi}(n)$ (which is relevant only for odd n) is not based on the minimization of a local error; rather, it is obtained heuristically by replacing the inner product (7.21) by a locally windowed version,

$$\hat{\psi}(n) = \arg \hat{C}_F(n) \;, \quad \hat{C}_F(n) = \left(x_{\mathcal{C}F}\,, x_{\mathcal{D}}\right)_{w_n} = \sum_k w(2k+1-n)\; x_{\mathcal{C}F}(2k+1)\; x^*(2k+1) \;. \qquad (7.26)$$

Here, $x_{\mathcal{C}F}(n)$ is the result of filtering $x_{\mathcal{C}}(n)$ with a filter with frequency response $F(\Theta)$; we note that this filter has to be approximated by a recursive or short-time version (e.g., an FIR filter) in order to be compatible with on-line processing. Comparing (7.26) with (7.25), we see that on-line autonomous phase matching can again be interpreted as on-line reference-based phase matching with reference signal $y(n)=x_{\mathcal{C}F}(n)$. Specializing to $F(\Theta)=S(\Theta)$, there results a computationally attractive on-line version of the SSA, where $\hat{C}_F(n)=\hat{C}_S(n)$ is given by

$$\hat{C}_S(n) = \left(x_{\mathcal{C}S}\,, x_{\mathcal{D}}\right)_{w_n} = \tfrac{1}{4} \sum_k w(2k+1-n) \left[e^{-j2\pi\Theta_0}\, x(2k+2) + e^{j2\pi\Theta_0}\, x(2k) \right] x^*(2k+1) \;.$$

Some comments. The window function w(k) may either have finite length (e.g., $w(k) = 0$ for $|k| > N$), or it may extend to $-\infty$. In the latter case, the exponential window $w(k) = e^{\alpha k} u(-k)$ (where u(k) is the unit step and $\alpha \geq 0$) is particularly efficient since the phase estimates $\hat{\varphi}(n)$ etc. may then be calculated recursively [44,37]. In the limiting case $\alpha=0$, i.e., $w(k) = u(-k)$, the time-varying phase estimates formally converge to the corresponding optimal phases of the off-line case for $n \to \infty$ [44]; however, this window may cause an overflow when computing $\hat{C}(n)$ on a digital computer and thus results in a potentially unstable algorithm.

In most applications, the on-line algorithm for autonomous phase matching will be combined with the on-line algorithm for PWD signal synthesis discussed in Section 6. Here, experiments suggest that best results are obtained when the phase matching is incorporated in the quasi power algorithm, i.e., when a separate phase matching step is performed in each iteration step of the quasi power algorithm. This combination of on-line signal synthesis and on-line phase matching is discussed in more detail in [37].

7.5 Simulation Results

Figure 12 reconsiders the WD signal synthesis experiment presented in Figure 3, this time concentrating on the aspect of (off-line) phase matching. Global WD signal synthesis was performed from the model shown in Figure 3.a. The synthesis result without phase matching is depicted in Figure 12.a (cf. Figure 3.b). The model is defined for $0 \leq \Theta < 1/2$; however, due to a relative phase mismatch, the greater part of signal energy is located in the complementary band $1/2 \leq \Theta < 1$.

The signals shown in Figures 12.b and 12.c were obtained by applying the off-line versions of the halfband approximation algorithm and spectral spread algorithm, respectively, to the result of global signal synthesis described above. The resulting signals are well concentrated in the desired band $0 \leq \Theta < 1/2$. The results of both phase matching algorithms are seen to be practically identical although the model is not halfband consistent. Also, there exists a close similarity to the result of halfband signal synthesis shown in Figure 3.d. We note that the WDs of the signals depicted in Figure 12 are identical (due to the WD's phase invariance) and are shown in Figure 3.b.3.

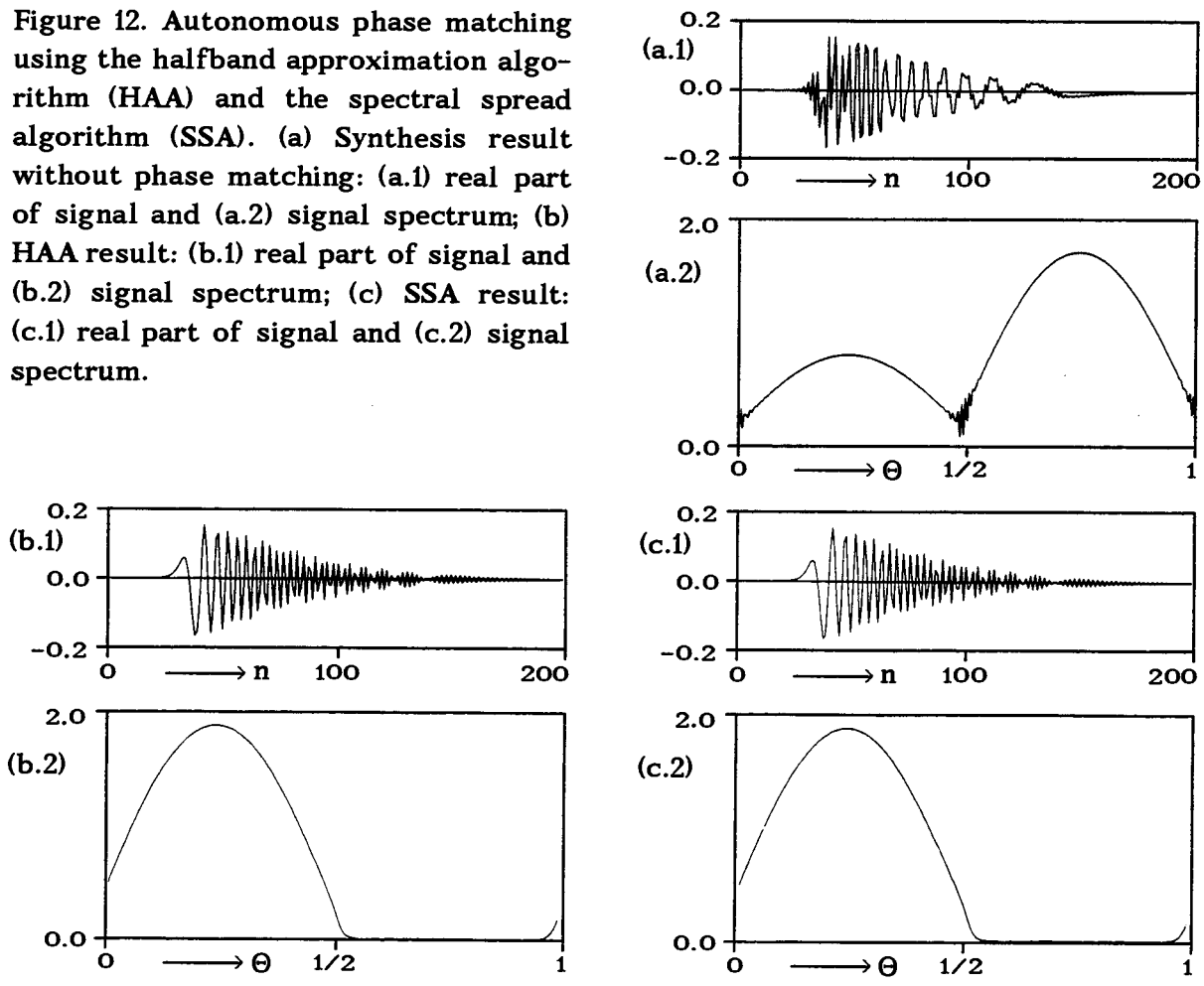

Figure 12. Autonomous phase matching using the halfband approximation algorithm (HAA) and the spectral spread algorithm (SSA). (a) Synthesis result without phase matching: (a.1) real part of signal and (a.2) signal spectrum; (b) HAA result: (b.1) real part of signal and (b.2) signal spectrum; (c) SSA result: (c.1) real part of signal and (c.2) signal spectrum.

The performance of the on-line version of the spectral spread algorithm (SSA) is illustrated in *Figure 13.* On-line PWD signal synthesis using the sample-by-sample algorithm described in Subsection 6.2 was performed from the model shown in Figure 13.a. The PWD window was chosen as a Hamming window of length 63. The synthesis result without phase matching is given in Figure 13.b. Again, a large part of signal energy is seen to exist in the band $1/2 \le \Theta < 1$ which is complementary to the model band $0 \le \Theta < 1/2$. Figure 13.c shows the synthesized signal after phase matching by means of the on-line version of the SSA using the exponential window $w(k) = e^{\alpha k}u(-k)$ with $\alpha = 0.01$. The signal is well concentrated in the model band $0 \le \Theta < 1/2$. The time evolution of the local relative-phase estimate $\hat{\psi}(n)$ is depicted in Figure 13.d; it is seen that $\hat{\psi}(n)$ is very close to the relative phase ψ_{opt} obtained by the *off-line* version of the SSA. (Indeed, the results of the off-line and on-line SSA obtained in this experiment were so close to each other that the respective plots did not show any differences; therefore, no plots illustrating the off-line results are included in Figure 13.)

8. APPLICATION TO TIME-FREQUENCY FILTERING

An important application of the signal synthesis methods discussed in previous sections is the problem of time-frequency filtering. In the simplest case, we would like to design a filter that passes all signals located in a given time-frequency region R and suppresses all signals located outside this region. (A signal is "located in a time-frequency region" if the effective support of the signal's WD is located in this region.)

A conceptually simple approach to performing such a time-frequency filtering is an analysis-masking-synthesis scheme based on the WD [10,14,16,45,46]. Here, the WD of the signal to be processed is calculated (analysis step), the signal's WD is multiplied by a mask (masking step), and finally the output signal is synthesized from the masked WD (synthesis step). The mask is the indicator function of the "time-frequency pass region" R, i.e., it is one inside R and zero outside R. The synthesis step makes use of the WD synthesis methods of Section 3 and the phase matching methods of Section 7.

Interference term effects in WD-based signal synthesis. The overall method is highly nonlinear, and it suffers from the quadratic nature of the WD, specifically, from the occurrence of cross or interference terms (ITs) [6]. An example is shown in *Figure 14.* We are given a three-component signal consisting of two chirp components $c_1(n)$, $c_2(n)$ and a Gaussian component $g(n)$. We wish to isolate the second chirp component $c_2(n)$ using the WD analysis-masking-synthesis method. Unfortunately, the oscillatory IT of the other two signal components $c_1(n)$ and $g(n)$ is located on top of the WD of $c_2(n)$. Hence, this IT is still present after masking the WD, i.e., in the synthesis model. This parasitic IT has a dramatic influence on the result of signal synthesis. In fact, it is shown in Figure 14.c that the synthesized signal is not the signal component desired but a linear combination of the other two signal components $c_1(n)$ and $g(n)$, i.e., the signal components to be suppressed.

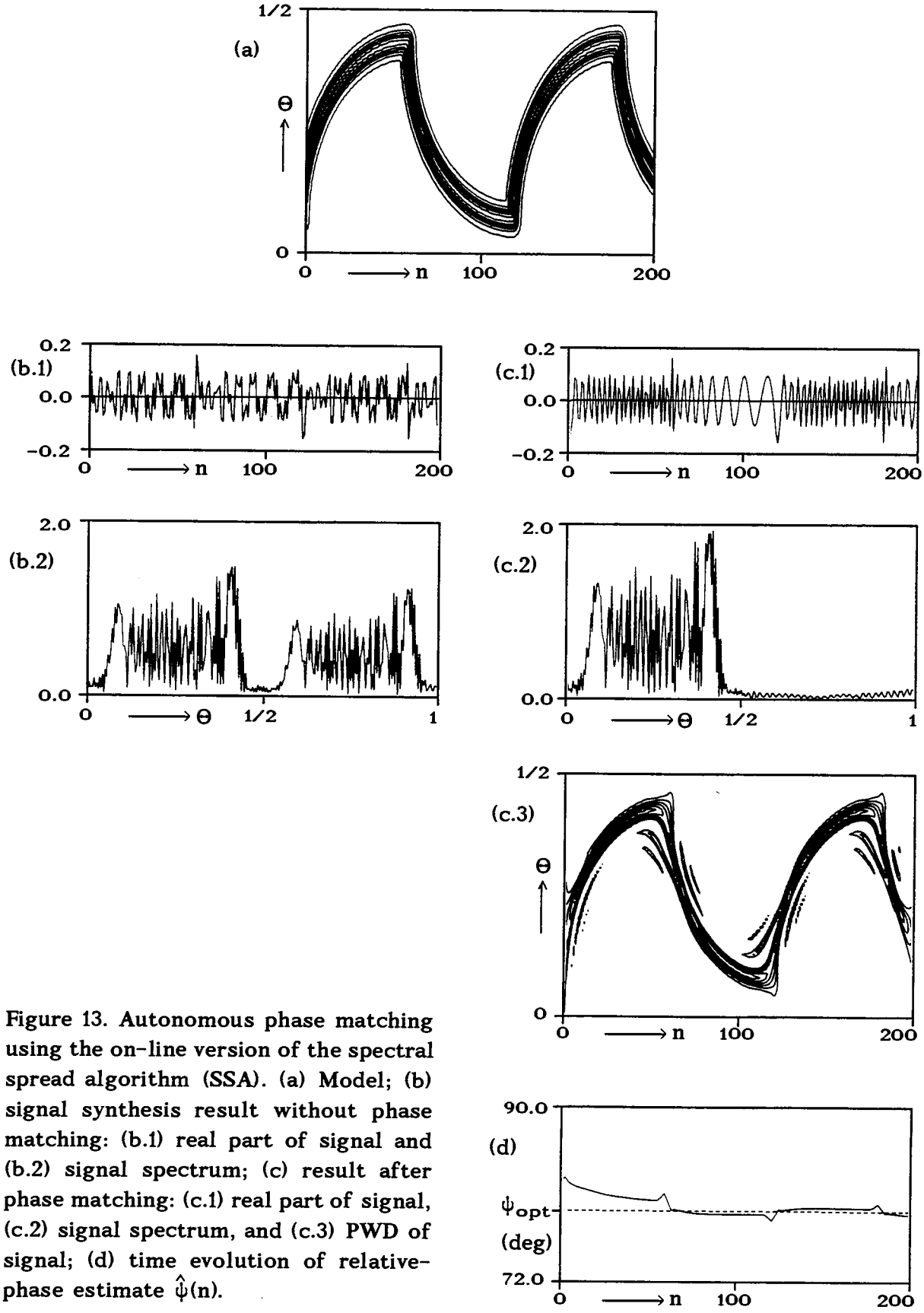

Figure 13. Autonomous phase matching using the on-line version of the spectral spread algorithm (SSA). (a) Model; (b) signal synthesis result without phase matching: (b.1) real part of signal and (b.2) signal spectrum; (c) result after phase matching: (c.1) real part of signal, (c.2) signal spectrum, and (c.3) PWD of signal; (d) time evolution of relative-phase estimate $\hat{\psi}(n)$.

A theoretical analysis of IT effects in WD signal synthesis [15] has shown that this specific behavior will always occur whenever the WD of the desired signal component is overlaid by the IT of two other components, provided that the geometric mean of the energies of these two interfering signal components is larger than the energy of the desired component. (In the opposite case, the desired component will be obtained.)

Avoiding IT effects. There are several ways to combat IT effects. One approach, proposed in [15], is the use of a smoothed WD (SWD) instead of the WD. Here, IT

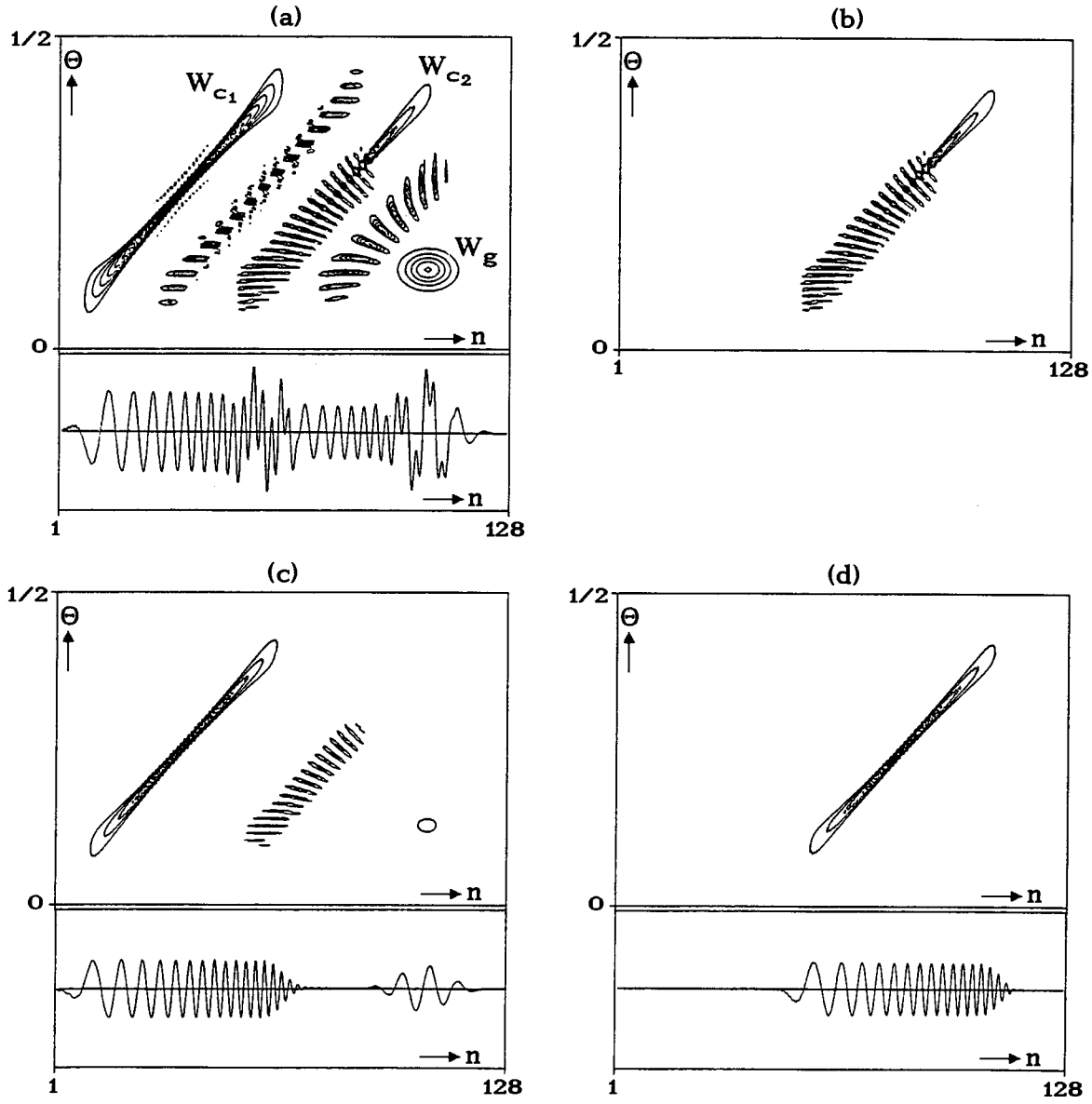

Figure 14. Interference effects in WD-based signal synthesis [15]. (a) Real part and WD of three-component signal; (b) result of masking the WD of the three-component signal; (c) result of WD-based signal synthesis from the masked WD; (d) desired signal component $c_2(n)$.

effects are reduced since the SWD's ITs are attenuated as compared to the WD's ITs [6]. Note that this advantage of the SWD provides a motivation for our detailed treatment of SWD signal synthesis algorithms in Sections 4 and 6. The application of SWD-based signal synthesis to the example of Figure 14 is shown in *Figure 15*. It is seen that the IT of the signal components to be suppressed is essentially smoothed out and hence does not affect the result of SWD signal synthesis, which is very similar to the desired signal component $c_2(n)$.

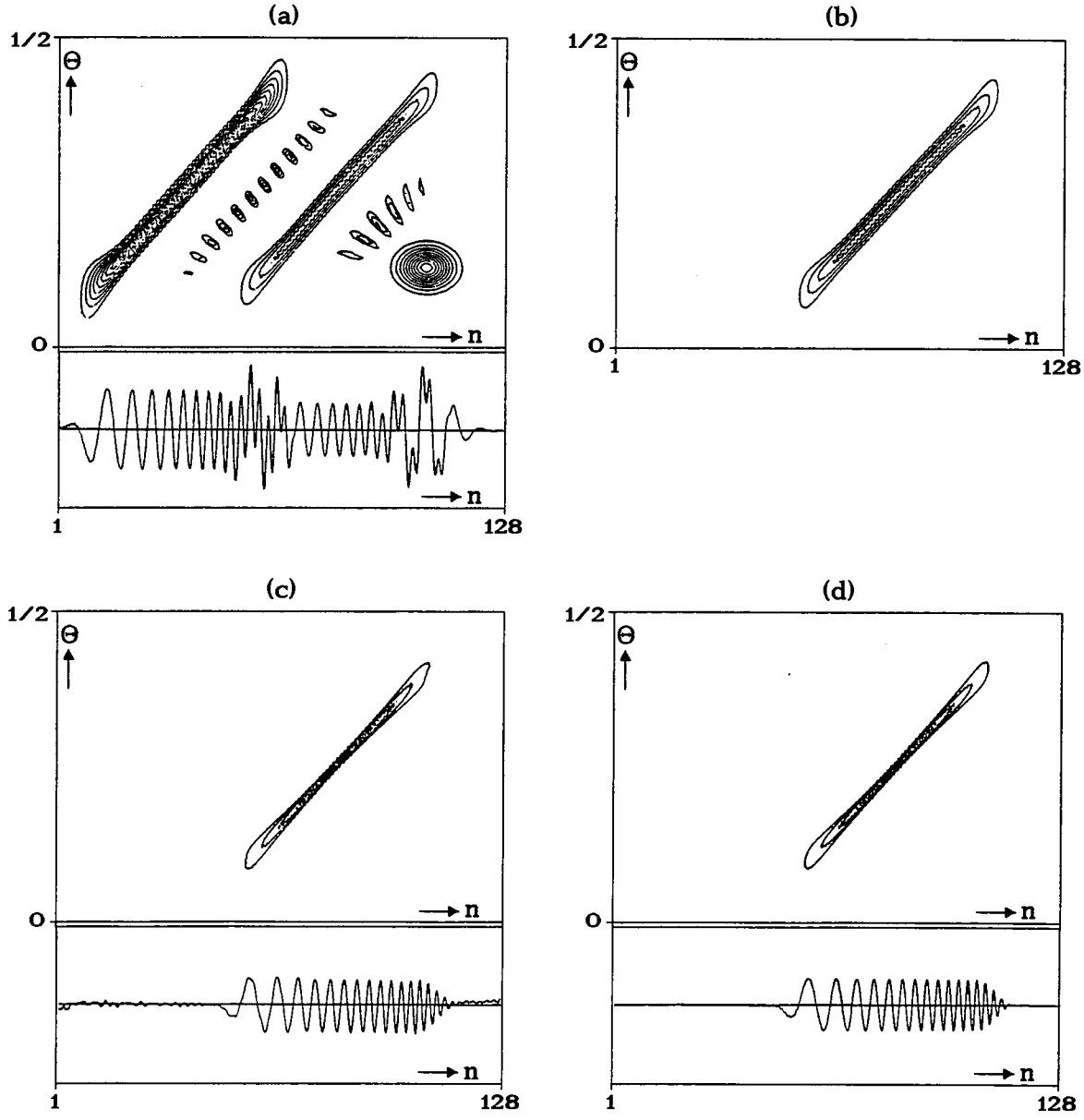

Figure 15. Avoiding interference effects by using SWD-based signal synthesis [15]. (a) Real part and SWD of three-component signal; (b) result of masking the SWD of the three-component signal; (c) real part and WD of result of SWD-based signal synthesis from the masked SWD; (d) desired signal component $c_2(n)$.

A further approach to reducing IT effects is the use of a *masked* WD [47]. This is analogous to SWD-based signal synthesis as discussed in Section 4, with the difference that a masked WD (MWD), i.e., the product of the WD and a pre-defined masking function, is used instead of an SWD. The masking function may be designed to ignore parts of the model where parasitic ITs are located; it is *not* identical with the mask used in the overall analysis-masking-synthesis process. Similar to the SWD case, the MWD synthesis problem leads to a quasi-eigen-equation that can be solved iteratively by a suitable version of the quasi power algorithm (cf. Section 4) [47].

An example illustrating the application of MWD-based signal synthesis as compared with WD-based synthesis is shown in *Figure 16*. Again, the filtering task considered is the isolation of a signal component whose WD is overlaid by the IT of two other components. The synthesis model is a masked version of the WD of the three-component signal, which still contains the parasitic IT. Consequently, the result of WD-based signal synthesis is again a linear combination of the interfering signal components (see Figure 16.c). The result of MWD-based synthesis from the same model, with the MWD mask designed to ignore the parasitic IT, is shown in Figure 16.f; it is seen to be very similar to the desired signal component.

The use of a masked *cross*-WD (instead of a masked auto-WD as discussed above) has been proposed in [48,16]. This method involves a masked cross-WD of the signal to be synthesized and a reference signal which is assumed fixed. The synthesis problem here leads to a linear equation which may be ill-conditioned. The cross-WD signal synthesis is part of a recursive overall scheme where the reference signal is improved after each synthesis step [48,16].

The examples shown in Figures 14-16 were somewhat pessimistic in that the mask was applied to a WD overlaid by an IT. A better procedure would be to first isolate/synthesize a signal component whose WD is *not* overlaid by an IT, subtract it from the overall multicomponent signal, calculate the WD of this reduced signal, isolate/synthesize the next component, etc. The resulting recursive signal-synthesis/signal-subtraction scheme has been proposed and applied to seismic signal analysis in [45]. An alternative method, which performs an approximate signal decomposition without a masking step, has been described in [49].

Linear time-frequency filters. Troublesome IT effects are avoided altogether if the analysis-masking-synthesis scheme is performed using a *linear* time-frequency representation (e.g., the short-time Fourier transform or the wavelet transform) instead of the quadratic WD [50-53]. Signal synthesis based on a linear time-frequency representation can always be performed by solving a linear equation. The overall analysis-masking-synthesis scheme then results in a linear, time-varying filter. Alternative "direct" designs of such a linear, time-varying "time-frequency filter" (which are not based on an analysis-masking-synthesis scheme) are the *Zadeh filter*, the *Weyl filter*, and the *time-frequency projection filter* [30,31,53,54].

A comparison of various linear and nonlinear schemes for time-frequency filtering is shown in *Figure 17* [53]. The input signal consists of an FM component (to be suppressed) and a (desired) Gaussian component. The filtering task is a

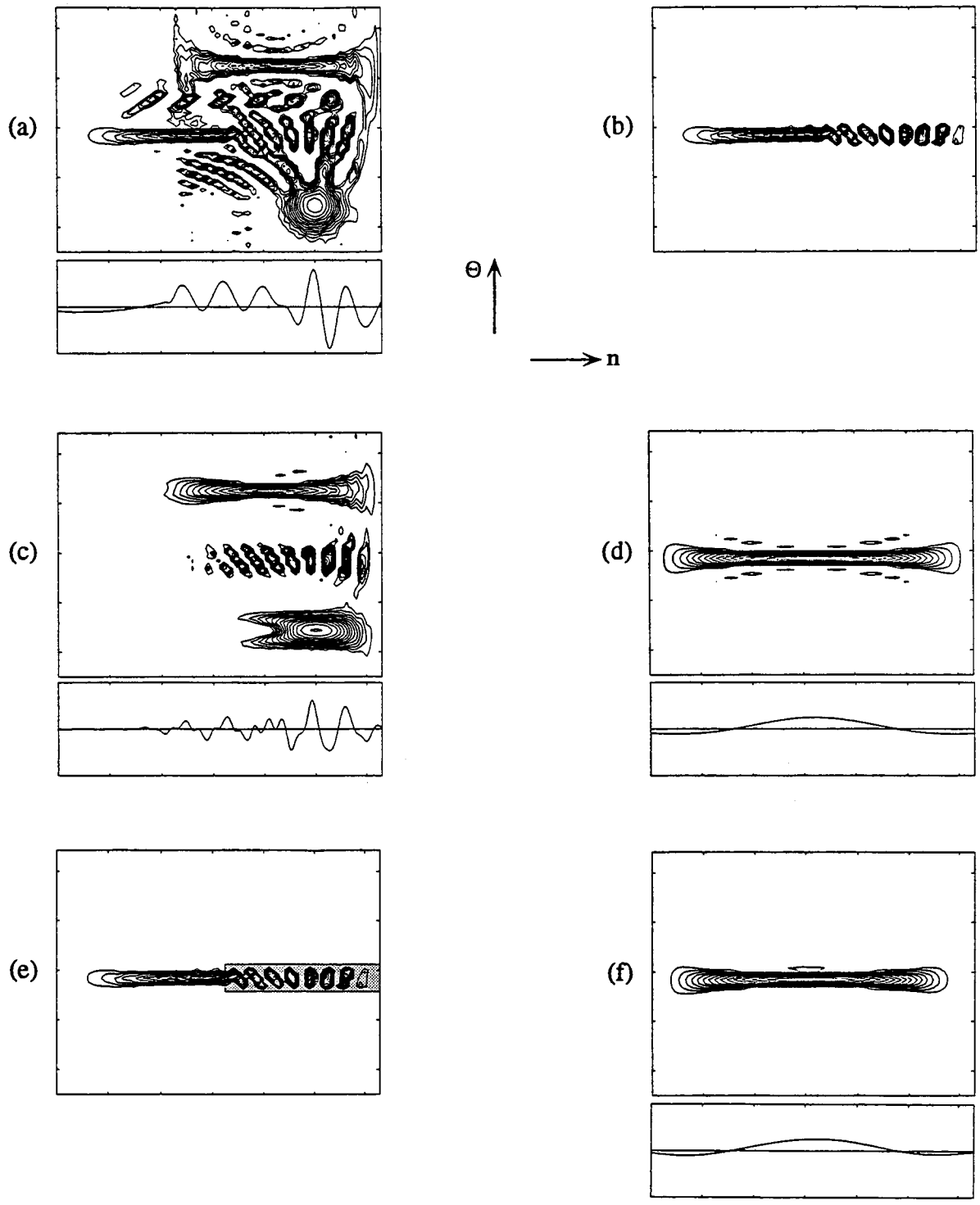

Figure 16. Avoiding interference effects by using MWD-based signal synthesis [47]. (a) Real part and WD of three-component signal; (b) result of masking the WD of the three-component signal; (c) real part and WD of result of WD-based signal synthesis from the model in (b); (d) desired signal component; (e) model as in (b) and MWD mask (the MWD mask is such that the shaded region is ignored); (f) result of MWD-based synthesis from the model in (b).

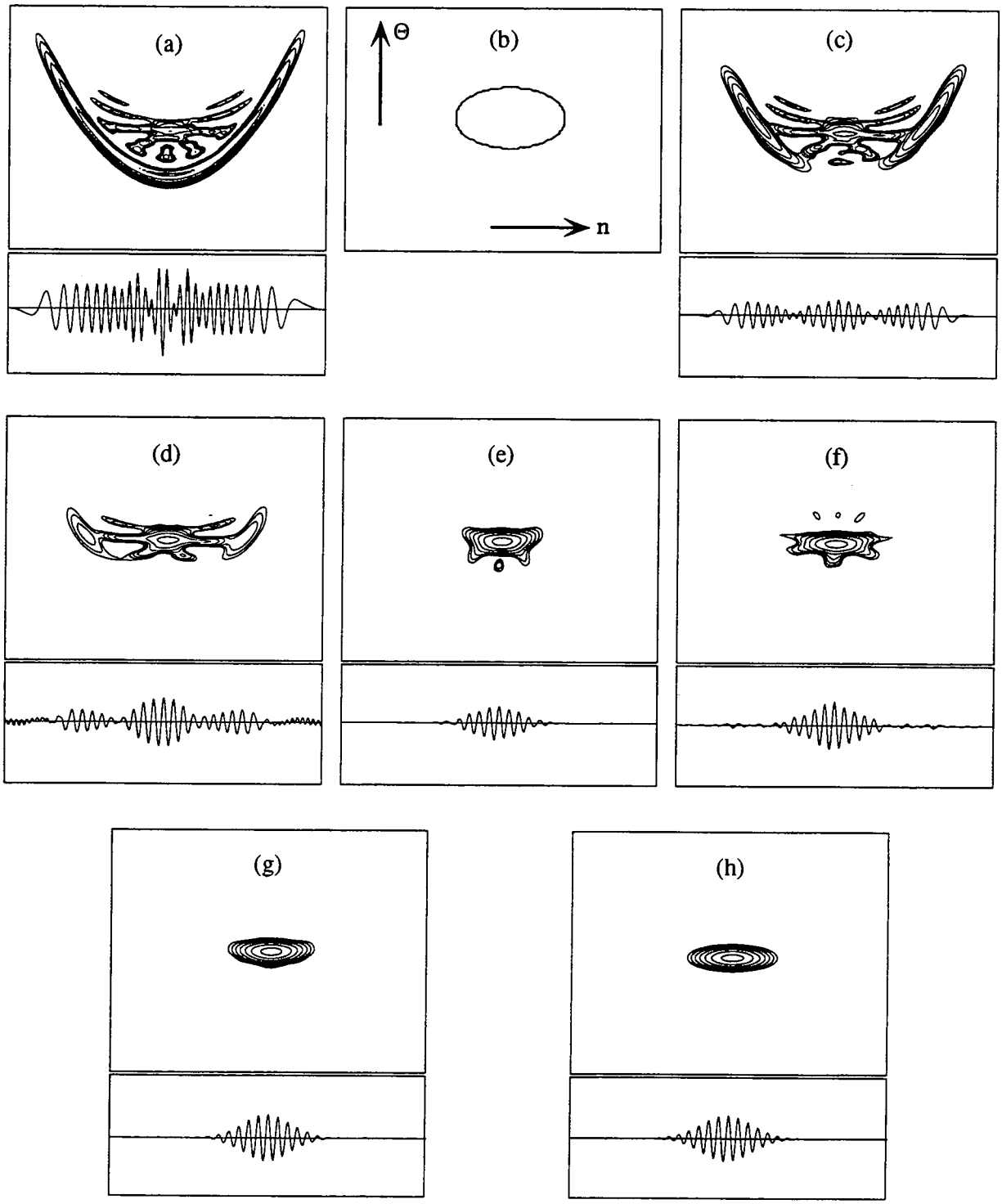

Figure 17. Performance comparison of various nonlinear and linear time-frequency filters [53]. (a) Real part and WD of two-component signal; (b) mask used for masking the WD, SWD or STFT, or used for direct filter design; (c) result of WD-based analysis-masking-synthesis (AMS); (d) result of SWD-based AMS; (e) result of linear STFT-based AMS; (f) result of linear Weyl filter; (g) result of linear time-frequency projection filter; (h) desired signal component.

difficult one since the signal components are very close in the time-frequency plane. The first three methods shown are based on the analysis-masking-synthesis scheme involving the WD, SWD, and short-time Fourier transform (STFT), respectively. Hence, the first two time-frequency filters are nonlinear while the third one is linear. The result of the WD-based filter again suffers from the presence of parasitic ITs inside the mask; this effect is somewhat reduced in the result of the SWD-based filter, and altogether avoided by the (linear) STFT-based filter. The remaining two methods (Weyl filter and time-frequency projection filter) are linear filters which are based on a direct filter design [53,54]. This comparison shows the superiority of linear time-frequency filters as compared to nonlinear analysis-masking-synthesis methods.

9. CONCLUSION

We conclude this chapter with some general remarks on the signal synthesis algorithms presented, and a brief survey of related work.

9.1 Summary and Discussion

The algorithms discussed in this chapter show strong similarities but also some differences. To avoid confusion, it is helpful to summarize and distinguish the various categories of signal synthesis algorithms. Three possible classifications are the following:

1. Algorithms for *global* signal synthesis (without a constraint on the synthesis result), and for *subspace* signal synthesis (where the synthesis result is constrained to be an element of a prescribed linear signal subspace). In the context of the WD, SWD, and AF1, an important special case of subspace signal synthesis is *halfband* signal synthesis where the synthesis result is constrained to be a halfband signal.
2. Algorithms for BTFRs which are *unitary* on the signal space on which signal synthesis is performed (like WD', RD, AF2 and, with certain restrictions, WD and AF1), and for BTFRs which are inherently *non-unitary* (like the SWD or spectrogram).
3. Algorithms for *off-line* signal synthesis (where the entire signal is synthesized as a whole, making use of the entire model), and for *on-line* signal synthesis (here, individual signal blocks or samples are synthesized sequentially, using local blocks or slices of the model). On-line signal synthesis is important in the context of SWDs allowing an on-line calculation (e.g., PWD with a finite-length window).

All signal synthesis algorithms discussed consist of two main steps. Step 1 is a transformation which maps the original time-frequency model into a "signal product domain" (case of global signal synthesis) or a coefficient domain (case of subspace signal synthesis). Step 2 depends on the unitarity property of the BTFR for which the signal synthesis problem is formulated. If the BTFR is (subspace-) unitary, then Step 2 is the calculation of the dominant eigenvalue and eigenvector of a Hermitian matrix; this may be done iteratively by means of the power algorithm. In the case of a non-unitary BTFR, Step 2 is the solution of a third-order equation

which no longer is a conventional eigenvalue-eigenvector equation. For this, we propose the iterative "quasi power algorithm" which is a natural extension of the power algorithm.

A special situation is encountered in the case of the WD, SWD, and AF1 where the result of *global* signal synthesis contains a troublesome "relative" phase ambiguity in addition to the "absolute" phase ambiguity which is always present in the synthesis result. Various phase matching algorithms can be used for resolving the relative and/or absolute phase ambiguity of the synthesized signal. Resolving the absolute phase ambiguity requires a reference signal. The relative phase ambiguity, on the other hand, can also be resolved without a reference signal by application of the "autonomous" phase matching algorithms discussed in Section 7. If *halfband* signal synthesis is used instead of global signal synthesis, then the problem of relative phase ambiguity is avoided altogether. For halfband signal synthesis, either an optimal algorithm or a reduced-cost (suboptimal) algorithm may be used.

In the case of the WD, SWD or AF1, the various methods for signal synthesis (global, optimal halfband or reduced-cost halfband) and for phase matching (absolute and/or relative phase; reference-based or autonomous) can be combined into the following synthesis strategies:

1. Global signal synthesis and
    1.1 reference-based matching of absolute and relative phase, or
    1.2 autonomous matching of relative phase and
        1.2.1 reference-based matching of absolute phase, or
        1.2.2 no matching of absolute phase;

2. Optimal halfband signal synthesis and
    2.1 reference-based matching of absolute phase, or
    2.2 no matching of absolute phase;

3. Reduced-cost halfband signal synthesis and
    3.1 reference-based matching of absolute phase, or
    3.2 no matching of absolute phase.

The best strategy naturally depends on the specific application. In this context, the following questions have to be answered: Is the absolute phase of importance? Is a meaningful reference signal for phase matching available? Is the strict bandlimitation of the result of halfband signal synthesis desirable?

Of course, the synthesis strategies sketched above are also different with respect to computational requirements. While it is not clear in general whether global signal synthesis or halfband signal synthesis is more expensive (this depends on the size of the model and the convergence speed of the power algorithm or quasi power algorithm iteration), it is certain that reduced-cost halfband signal synthesis is cheaper than both global signal synthesis and optimal halfband signal synthesis.

9.2 Extensions and Related Work

The signal synthesis algorithms discussed in this chapter can be extended in several directions. Signal synthesis algorithms for unitary affine and hyperbolic time-frequency representations are discussed in [55,56]. The extension of the basic signal synthesis scheme to the optimum time-frequency synthesis of nonstationary random processes is considered in [57]. A similar extension is the optimum synthesis of linear signal spaces or, equivalently, the optimum design of linear "time-frequency projection filters" [30,31,54]. The optimum time-frequency design of general linear, time-varying systems has been studied in [58]. The application of extended signal synthesis to automatic signal decomposition is considered in [49].

We finally note that an alternative approach to signal synthesis, based on alternating convex projections, has been proposed in [56,59]. This method allows the inclusion of rather general side constraints but appears to be quite expensive.

REFERENCES

[1] F. Hlawatsch and G.F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Sig. Processing Mag.*, Vol. 9, April 1992, pp. 21-67.

[2] T.A.C.M. Claasen and W.F.G. Mecklenbräuker, "The Wigner distribution - A tool for time-frequency signal analysis, Part III: Relations with other time-frequency signal transformations," *Philips J. Res.*, Vol. 35, pp. 372-389, 1980.

[3] F. Hlawatsch, "Duality and classification of bilinear time-frequency signal representations," *IEEE Trans. Signal Processing*, Vol. 39, No. 7, July 1991, pp. 1564-1574.

[4] W. Mecklenbräuker, "A tutorial on non-parametric bilinear time-frequency signal representations," in *Signal Processing*, J.L. Lacoume, T.S. Durrani and R. Stora, eds., North-Holland, Amsterdam, pp. 278-336, 1987.

[5] P. Flandrin, "Representations temps-frequence des signaux non-stationnaires," These d'Etat, Institut National Polytechnique de Grenoble, France, 1987.

[6] F. Hlawatsch and P. Flandrin, "The interference structure of the Wigner distribution and related time-frequency signal representations," in *The Wigner Distribution - Theory and Applications in Signal Processing*, Eds. W. Mecklenbräuker and F. Hlawatsch, Amsterdam: Elsevier, 1997.

[7] L. Cohen, "Time-frequency distributions - A review," *Proc. IEEE*, Vol. 77, No. 7, pp. 941-981, July 1989.

[8] B. Boashash, "Time-frequency signal analysis," chapter in *Advances in Spectrum Estimation*, Ed. S. Haykin, Prentice-Hall, 1990.

[9] F. Hlawatsch, "Regularity and unitarity of bilinear time-frequency signal representations," *IEEE Trans. Information Theory*, Vol. 38, No. 1, Jan. 1992, pp. 82-94.

[10] G.F. Boudreaux-Bartels and T.W. Parks, "Time-varying filtering and signal estimation using Wigner distribution synthesis techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-34, pp. 442-451, June 1986.

[11] C. Wilcox, "The synthesis problem for radar ambiguity functions," Tech. Rep. No. 157, Math. Res. Center, Univ. of Wisconsin, Madison, Wis., April 1960.

[12] S.M. Sussman, "Least-square synthesis of radar ambiguity functions," *IRE Trans. Inform. Theory*, pp. 246-254, April 1962.

[13] F. Hlawatsch and W. Krattenthaler, "Bilinear signal synthesis," *IEEE Trans. Signal Processing*, Vol. 40, No. 2, Feb. 1992, pp. 352-363.

[14] B.E.A. Saleh and N.S. Subotic, "Time-variant filtering of signals in the mixed time-frequency domain," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-33, No. 6, Dec. 1985, pp. 1479-1485.

[15] W. Krattenthaler and F. Hlawatsch, "Time-frequency design and processing of signals via smoothed Wigner distributions," *IEEE Trans. Signal Processing*, Vol. 41, No. 1, Jan. 1993, pp. 278-287.

[16] G.F. Boudreaux-Bartels, "Time-varying signal processing using Wigner distribution synthesis techniques", in *The Wigner Distribution - Theory and Applications in Signal Processing*, Eds. W. Mecklenbräuker and F. Hlawatsch, Elsevier, 1997.

[17] E.P. Wigner, "Quantum-mechanical distribution functions revisited," in *Perspectives in Quantum Theory*, W. Yourgrau and A. van der Merwe, eds., Dover, 1971.

[18] T.A.C.M. Claasen and W.F.G. Mecklenbräuker, "The Wigner distribution - A tool for time-frequency signal analysis, Part II: Discrete-time signals," *Philips J. Res.*, Vol. 35, pp. 276-300, 1980.

[19] F. Peyrin and R. Prost, "A unified definition for the discrete-time, discrete-frequency, and discrete time/frequency Wigner distributions," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-34, pp. 858-867, Aug. 1986.

[20] P. Flandrin and W. Martin, "Pseudo-Wigner estimators for the analysis of nonstationary processes," in *Proc. IEEE Spectr. Est. Workshop II*, Tampa, FL, pp. 181-185, Nov. 1983.

[21] W. Wokurek, F. Hlawatsch, and G. Kubin, "Wigner distribution analysis of speech signals," in *Proc. 1987 Int. Conf. on Digital Signal Processing*, pp. 294-298, Florence, Italy, Sept. 1987.

[22] F. Hlawatsch, T.G. Manickam, R. Urbanke, and W. Jones, "Smoothed pseudo Wigner distribution, Choi-Williams distribution, and cone-kernel representation: Ambiguity-domain analysis and experimental comparison," *Signal Processing*, vol. 43, no. 2, May 1995, pp. 149-168.

[23] H.-I. Choi and W.J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-37, pp. 862-871. June 1989.

[24] J. Jeong and W.J. Williams, "Kernel design for reduced interference distributions," *IEEE Trans. Signal Processing*, Vol. 40, No. 2, Feb. 1992, pp. 402-412.

[25] R.A. Altes, "Detection, estimation, and classification with spectrograms," *J. Acoust. Soc. Am.*, Vol. 67, No. 4, Apr. 1980, pp. 1232-1246.

[26] P.M. Woodward, *Probability and Information Theory with Application to Radar*. Pergamon Press, London, 1953.

[27] L.E. Franks, *Signal Theory*. Prentice Hall, 1969.

[28] A.W. Naylor and G.R. Sell, *Linear Operator Theory in Engineering and Science*. Springer Verlag, 1982.

[29] D.G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley

& Sons, 1969.

[30] F. Hlawatsch and W. Kozek, "Time-frequency projection filters and time-frequency signal expansions," *IEEE Trans. Signal Proc.*, vol. 42, pp. 3321-3334, Dec. 1994.

[31] F. Hlawatsch and W. Kozek, "The Wigner distribution of a linear signal space," *IEEE Trans. Signal Processing*, vol. 41, no. 3, pp. 1248-1258, March 1993.

[32] G.H. Golub and C.F. van Loan, *Matrix Computations*. John Hopkins University Press, Baltimore, MD, 1984.

[33] B.V.K. Kumar, C.P. Neuman, and K.J. De Vos, "Discrete Wigner synthesis," *Signal Processing*, Vol. 11, pp. 277-304, 1986.

[34] J. Wexler and S. Raz, "Synthesis of discrete-time signals from distributions," *Electronics Letters*, Vol. 25, pp. 93-95, Jan. 1989.

[35] S. Raz, "Synthesis of signals from Wigner distributions: representation on biorthogonal bases," *Signal Processing*, Vol. 20, pp. 303-314, 1990.

[36] J. Wexler and S. Raz, "Wigner-space synthesis of discrete-time periodic signals," *IEEE Trans. Signal Processing*, Vol. 40, No. 8, Aug. 1992, pp. 1997-2006.

[37] W. Krattenthaler, "Signal synthesis algorithms for non-smoothed and smoothed Wigner distributions," Doctoral dissertation, Technische Universität Wien, Vienna, Austria, Feb. 1990.

[38] Y. Zhao, L.E. Atlas, and R.J. Marks, II, "The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-38, pp. 1084-1091, July 1990.

[39] W. Krattenthaler and F. Hlawatsch, "Two signal synthesis algorithms for pseudo Wigner distribution," *Proc. IEEE ICASSP-88*, New York, April 1988, pp. 1550-1553.

[40] W. Krattenthaler and F. Hlawatsch, "General signal synthesis algorithms for smoothed versions of Wigner distribution," *Proc. IEEE ICASSP-90*, Albuquerque, NM, pp. 1611-1614, Apr. 1990.

[41] D.W. Griffin and J.S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-32, No. 2, pp. 236-242, Apr. 1984.

[42] W. Krattenthaler and F. Hlawatsch, "Improved signal synthesis from pseudo Wigner distribution," *IEEE Trans. Signal Processing,* Vol. 39, pp. 506-509, Feb. 1991.

[43] K.-B. Yu and S. Cheng, "Signal synthesis from pseudo Wigner distribution and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-35, pp. 1289-1302, Sept. 1987.

[44] F. Hlawatsch and W. Krattenthaler, "Phase matching algorithms for Wigner-distribution signal synthesis," *IEEE Trans. Signal Processing*, Vol. 39, No. 3, March 1991, pp. 612-619.

[45] G.F. Boudreaux-Bartels and P.J. Wiseman, "Wigner distribution analysis of acoustic well logs," *Proc. IEEE ICASSP-87*, Dallas, TX, Apr. 1987, pp. 2237-2240.

[46] J. Jeong and W.J. Williams, "Time-varying filtering and signal synthesis," in *Time-Frequency Signal Analysis — Methods and Applications*, Ed. B. Boashash, Melbourne, Australia: Longman and Cheshire, 1991.

[47] F. Hlawatsch, A.H. Costa, and W. Krattenthaler, "Time-frequency signal

synthesis with time-frequency extrapolation and don't-care regions," *IEEE Trans. Signal Processing*, vol. 42, no. 9, Sept. 1994, pp. 2513-2520.

[48] T.J. McHale and G.F. Boudreaux-Bartels, "A weighted cross Wigner distribution synthesis algorithm for use with partial model information," *IEEE 1990 DSP Workshop*, New Paltz, NY, pp. 3.11.1-2, Sept. 1990.

[49] F. Hlawatsch and W. Krattenthaler, "A new approach to time-frequency signal decomposition," *Proc. IEEE ISCAS-89*, Portland, Oregon, May 1989, pp. 1248-1251.

[50] M.R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 28, Feb. 1980, pp. 55-69.

[51] I. Daubechies, "Time-frequency localization operators: A geometric phase space approach," *IEEE Trans. Inform. Theory*, Vol. 34, No. 4, July 1988, pp. 605-612.

[52] I. Daubechies and T. Paul, "Time-frequency localization operators: A geometric phase space approach: II. The use of dilations," *Inverse Problems*, No. 4, 1988, pp. 661-680.

[53] W. Kozek and F. Hlawatsch, "A comparative study of linear and nonlinear time-frequency filters," *Proc. IEEE-SP Int. Sympos. Time-Frequency and Time-Scale Analysis*, Oct. 1992, Victoria, BC, Canada, pp. 163-166.

[54] W. Kozek and F. Hlawatsch, "Time-frequency filter banks with perfect reconstruction," *Proc. IEEE ICASSP-91*, Toronto, Canada, May 1991, pp. 2049-2052.

[55] F. Hlawatsch, A. Papandreou, and G.F. Boudreaux-Bartels, "Regularity and unitarity of affine and hyperbolic time-frequency representations," *Proc. IEEE ICASSP-93*, Minneapolis, MN, Apr. 1993, pp. 245-248.

[56] L.B. White, "The wide-band ambiguity function and Altes' Q-distribution: Constrained synthesis and time-scale filtering," *IEEE Trans. Information Theory*, Vol. 38, No. 2, March 1992, pp. 886-892.

[57] F. Hlawatsch and W. Kozek, "Second-order time-frequency synthesis of nonstationary random processes," *IEEE Trans. Inf. Theory*, vol. 41, Jan. 1995, pp. 255-267.

[58] F. Hlawatsch, "Wigner distribution analysis of linear, time-varying systems," *Proc. IEEE ISCAS-92*, San Diego, CA, May 1992, pp. 1459-1462.

[59] L.B. White, "Time-frequency filtering and synthesis using convex projections," *SPIE Adv. Algor. for Signal Processing*, Vol. 1348, San Diego, CA, pp. 158-169.