

Technical Memorandum

Contract: HELMAS, F29601-01-C-0001



9 Jan 2004

Revised 1 Feb 2007

To: (Memorandum for record)

From: Boris Venet, MZA Associates (venet@mza.com, 505-245-9970)

Re: WaveTrain and superposition of polychromatic fields – Memo Version 3

Abstract

The WaveTrain optical propagation code does not contain “explicit” support for the propagation and superposition of polychromatic optical fields. Nevertheless, if the user makes a relatively simple redefinition of the input fields, the operations carried out by a key WaveTrain sensor system will correctly compute the superposition (interference) of narrow-band polychromatic fields. This allows the treatment of a variety of important interference problems in which non-zero bandwidth is critical. Specific applications may include remote laser vibrometry and other ladar/lidar problems. We discuss the required WaveTrain input manipulations, and the range of problems that can be modeled. This memo applies to all versions of Wavetrain up to and including version G and version 2007A (the latest).

WaveTrain and superposition of polychromatic fields (Ver. 3)

*Boris Venet, MZA Associates Corp.
9 Jan 2004 (Revised 1 Feb 2007)*

1 Introduction

WaveTrain (WT) does not contain “explicit” support for the propagation and complex superposition of polychromatic optical fields. Nevertheless, if the user makes a relatively simple redefinition of the input fields, the operations carried out by a key WT detector system will correctly compute the superposition (interference) of nearby discrete frequencies, or more generally narrow-band polychromatic fields. This allows WT to simulate the basic aspects of problems in which non-zero bandwidth is critical, and allows application of the better-developed WT capabilities to these types of problems. Examples of such problems are laser vibrometry and a variety of other ladar/lidar problems where frequency mixing of one sort or another plays an important role. This memo applies to all versions of Wavetrain up to and including version G and version 2007A (the latest).

The key mathematical concept that we use is the representation of polychromatic fields by the slowly-varying envelope (SVE) formalism, and an associated complex-variable formula for computing optical irradiance.

In brief, the relevant issue in WT is as follows. For paraxial propagation along z , consider a scalar, complex-format optical field of the form

$$\Psi(\vec{\rho}, z, t) = U(\vec{\rho}, z, t) e^{i(k_0 z - \omega_0 t)} \quad (1)$$

where $\vec{\rho} = (x, y)$, k_0 and ω_0 refer to the nominal optical wavelength λ_0 , and U is a slowly-varying complex envelope factor. In the applications for which WT was originally designed, the time dependence of U arises from two sources: (a) dynamic atmospheric turbulence, i.e., “frozen-flow” turbulence in conjunction with the mean, *transverse* motion of sources, atmosphere, and sensors, and (b) *transverse* source or sensor jitters. Now the key problem is that the internal machinery of WT does *not* keep track of the phasor $e^{i(k_0 z - \omega_0 t)}$. The Fresnel propagation machinery only operates on the U factor to propagate fields across z increments. Of course the Fresnel propagator formula contains the nominal optical wavelength $2\pi/k_0$ as a scale factor, but it omits the $e^{i(k_0 z - \omega_0 t)}$ phasor. This procedure was perfectly reasonable for the original design applications of WT. However, if we want to model sensing applications where the sensor can temporally follow the interference pattern of two nearby optical frequencies, then a bit of customization is required in the assembly of the WT system. In the present memorandum, we show how to use existing WT

library components plus the "m-system" feature to give WT the required information about the $e^{i(kz - \omega t)}$ factors of the interfering waves.

In Sections 2 and 3, we begin by reviewing some aspects of the slowly-varying envelope (SVE) formalism, with a view to setting up equations that relate directly to the WaveTrain modeling example in Section 4.

2 Review of SVE form and optical irradiance

Let $E_R(t)$ be a scalar component of the real electric field of an electromagnetic wave. The slowly-varying envelope (SVE) representation is

$$E_R(t) = A(t) \cos(\omega_0 t + \phi(t)) \quad (2)$$

where ω_0 is the central frequency (or, more generally, some characteristic frequency), and where $A(t)$ and $\phi(t)$ are real modulation functions. To make the notation somewhat more compact, we omit writing the spatial dependences, but everything below is consistent with their presence. E_R and A have units of *Volt/m* (electric field).

The SVE form is particularly useful when the modulations $A(t)$ and $\phi(t)$ vary slowly with respect to ω_0 , and *also* with respect to an averaging time to be discussed below. Fields given by Equation (2) are polychromatic. When the modulations are slow, the fields are sometimes called “quasi-monochromatic”. A special case that will be of interest below is a set of (two or more) discrete, closely-spaced optical frequencies.

Optical sensors measure a quantity called “optical irradiance” (alternate term: “optical intensity”), which signifies a time-average of the (power/area) that is carried by the EM wave. The time averaging window T_{avg} is assumed very long compared to the optical period $2\pi/\omega_0$. Most fundamentally, T_{avg} represents a temporal smoothing associated with the field-matter interactions in the sensor, but external sensor electronics can also increase T_{avg} . The optical irradiance {power/area} carried by the wave is

$$I(t) = b \cdot \frac{1}{T_{avg}} \cdot \int_{t-T_{avg}}^t dt' E_R^2(t') \quad (3)$$

The constant b is a conversion factor that takes the electric field units to W/m^2 units. Now the more precise meaning of "slowly-varying" envelope is that the variation is slow compared to T_{avg} . If $A(t)$ and $\phi(t)$ in Equation (2) have the slowly-varying property, then the integral (3) reduces to

$$I(t) \approx \frac{1}{2} b \cdot A^2(t) \quad (4)$$

to high accuracy. Note that $I(t)$ is only affected by the magnitude modulation $A(t)$, and not at all by the phase modulation $\phi(t)$.

3 Complex representation of SVE fields and irradiance

Analytically, it is very convenient to rewrite Equation (2) using the complex representation

$$\begin{aligned} E(t) &= A(t) e^{-i(\omega_0 t + \phi(t))} \\ &= U(t) e^{-i\omega_0 t} \end{aligned} \quad (5)$$

where the complex SVE factor, $U(t) = A(t) \exp(-i\phi(t))$, now contains both the magnitude and phase modulations. For consistency of notation, observe that $\text{Real}\{E(t)\} = E_R(t)$. Now we note a simple but highly useful trick: the optical irradiance $I(t)$, as given by Equation (4), can be computed from the complex field representation by simply writing

$$\begin{aligned} I(t) &\approx \frac{1}{2} b \cdot A^2(t) \\ &= \frac{1}{2} b \cdot U(t) U^*(t) \\ &= \frac{1}{2} b \cdot E(t) E^*(t) \end{aligned} \quad (6)$$

The point is this: the irradiance time-averaging operation required by the fundamental definition (3) is effected merely by computing the magnitude-squared of the complex SVE (second line of Equation (6)), or the magnitude-squared of the full complex $E(t)$ (third line of Equation (6)). This result is strictly a mathematical accident, but is a very convenient trick for computational purposes. All basic optics textbooks state a version of this trick, but not all references make it clear that the procedure applies to the SVE case as well to the purely monochromatic case.

The usefulness of the complex form of the irradiance equation is most evident when we deal with a superposition of fields. As an example that is directly relevant to the WaveTrain implementation to be discussed in Section 4, consider a superposition of several closely-spaced monochromatic fields:

$$\begin{aligned} E(t) &= E_1(t) + E_2(t) + E_3(t) \\ &= A_1(t) e^{-i(\omega_1 t + \phi_1(t))} + A_2(t) e^{-i(\omega_2 t + \phi_2(t))} + A_3(t) e^{-i(\omega_3 t + \phi_3(t))} \\ &= \left[A_1(t) e^{-i(\Omega_1 t + \phi_1(t))} + A_2(t) e^{-i(\Omega_2 t + \phi_2(t))} + A_3(t) e^{-i(\Omega_3 t + \phi_3(t))} \right] e^{-i\omega_0 t} \\ &= U(t) e^{-i\omega_0 t} \end{aligned} \quad (7)$$

where $\Omega_k = \omega_k - \omega_0$.

Now we assume that Ω_k is slowly varying compared to T_{avg} . In that case, the entire complex envelope term inside [...], on the third line of Equation (7), constitutes a single complex SVE factor $U(t)$. Therefore, it follows from the basic result (6) that the net irradiance of the superposition must be given by

$$\begin{aligned} I(t) &= \frac{1}{2} b \cdot U(t) U^*(t) \\ &= \frac{1}{2} b \cdot \left| A_1(t) e^{-i(\Omega_1 t + \phi_1(t))} + A_2(t) e^{-i(\Omega_2 t + \phi_2(t))} + A_3(t) e^{-i(\Omega_3 t + \phi_3(t))} \right|^2 \end{aligned} \quad (8)$$

The cross terms could be easily computed, if they were explicitly desired; for purely numerical computation, it would be better to *not* expand out the square. We see, among other things, that the relative SVE phases do

affect the irradiance, although an overall SVE phase cancels out, as observed trivially in connection with Formulas (4)-(6).

To summarize in terms of a procedural prescription: the elementary “trick” formula, $I = (\frac{1}{2})b \cdot UU^*$, applies to any combination of SVE fields, providing we carry out the following prescription:

- (1) Factor off a common “characteristic” optical frequency phasor, $e^{-i\omega_0 t}$, from each field, so that the residual is SVE (as in Equation (7)).
- (2) Sum the residual complex SVE terms, and compute the magnitude-squared (as in Equation (8)).
- (3) In practice, one usually ignores the b conversion factor. This simply amounts to assigning units of $\sqrt{W/m^2}$ to $U(t)$, instead of *Volt/m*.
- (4) In practice, it is not even necessary to carry out step (1) explicitly, since the $e^{-i\omega_0 t}$ factor would disappear as well upon applying the magnitude-squared operation to the entire $E(t)$. However, depending on the context, it may be useful to explicitly display the complex SVE factors by factoring off the common $e^{-i\omega_0 t}$.

4 Modeling of SVE fields in WaveTrain

First, we recall that Equation (8) applies to fields superposed at any given space point, so that the net irradiance formula applies generally to the computation of $I(\vec{r}, t)$, where the SVE factors A_i and ϕ_i are now functions of spatial coordinate as well as time.

The WaveTrain (WT) sensor system called **SimpleFieldSensor** produces an output variable called **fld**. If one wavetrain impinges on **SimpleFieldSensor**, then **fld** contains the slowly-varying complex optical field U , at the sequence of times specified by **SimpleFieldSensor**’s temporal sampling parameters. Note two key points:

- (1) With WT’s monochromatic source modules, the usual source of time variation in U is moving turbulence screens, or possibly platform jitters impressed on the beam via dynamic beam-tilt modules.
- (2) The characteristic optical frequency phasor, $e^{-i\omega_0 t}$, is completely absent from **fld**, as discussed in Section 1 (although WaveTrain does use the specified source wavelength as a scaling factor when computing the Fresnel propagation).

Now, if two or more wavetrains from different monochromatic source modules, but with identical wavelengths, impinge on **SimpleFieldSensor**, then this sensor returns a **fld** output that is proportional to the complex sum of the individual slowly-varying U ’s. Each field separately may have an arbitrary spatial dependence, and the correct spatial interference pattern will be generated.

Now, let us consider the polychromatic (but SVE) case; in particular, for conceptual simplicity consider two closely-spaced monochromatic source fields. If the WaveTrain user simply inserted two WT source modules with slightly different wavelength specifications λ_1 and λ_2 , then **SimpleFieldSensor** would *not* report the correct time-varying fringe pattern. The part of the time variation due to transverse motion in the system interacting with turbulence would be present. But, the part of the time variation (“walking fringes”) due to the slight wavelength difference would *not* be present, because the U ’s of the two wavetrains contain no

explicit representation of the difference between $e^{-i\omega_1 t}$ and $e^{-i\omega_2 t}$. In other words, in the absence of turbulence, WT would report a completely static interference pattern.

Despite the absence of the optical frequency factors in WaveTrain, SimpleFieldSensor can still produce the correct superposition, *if* the user explicitly introduces the desired SVE time dependence into the WT system. This is best explained via an example. Figure 1 shows the simple physical problem that will be modeled, and Figure 2 shows the corresponding WaveTrain model.

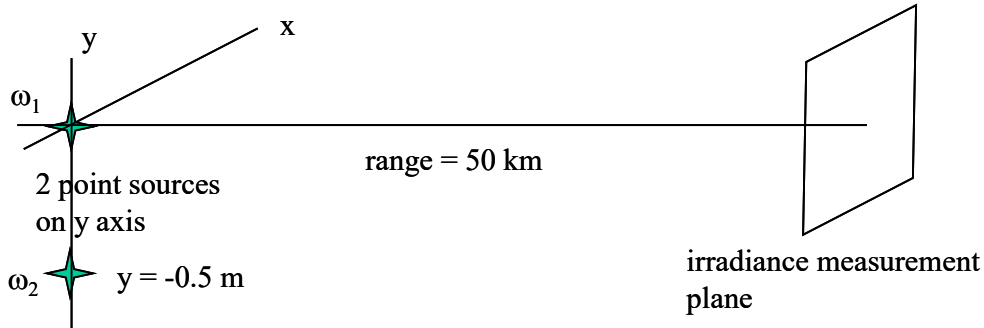


Figure 1: Polychromatic (but SVE) problem to be modeled using WaveTrain.

In Figure 1 we see two spatially separated point sources, located at $y = 0$ and $y = -0.5m$, which radiate at slightly different frequencies ω_1 and ω_2 . The light from each source propagates a distance of 50 km, where it is received by an optical detector that can measure the space-time dependence of the net irradiance.

The mathematical representation of the two optical fields, in the sensor plane, is as follows. Let the complex electric fields at the sensor plane be:

$$\text{Source 1: } E_1(\vec{\rho}, t) = [A_1 f_1(\vec{\rho})] \cdot e^{-i\omega_1 t}$$

$$\text{Source 2: } E_2(\vec{\rho}, t) = A_2 f_2(\vec{\rho}) e^{-i\omega_2 t} = [A_2 f_2(\vec{\rho}) e^{-i\Omega_2 t}] \cdot e^{-i\omega_1 t}, \quad \text{where } \Omega_2 = \omega_2 - \omega_1$$

The optical SVE fields are:

$$\text{Source 1: } U_1(\vec{\rho}, t) = A_1 f_1(\vec{\rho}) \tag{9}$$

$$\text{Source 2: } U_2(\vec{\rho}, t) = A_2 f_2(\vec{\rho}) e^{-i\Omega_2 t} \tag{10}$$

In the above formulas, the A_k are constants, and $\vec{\rho} = (x, y)$ is the transverse spatial coordinate in the sensor plane. The factors $f_1(\vec{\rho})$ and $f_2(\vec{\rho})$ are quadratic phases whose exact formulas are inessential for present purposes (though of course they affect the spatial fringe pattern in the sensor plane). The $f_1(\vec{\rho})$ and $f_2(\vec{\rho})$ factors are automatically modeled by the WaveTrain machinery. In this example, we are omitting turbulence, so the $f_1(\vec{\rho})$ and $f_2(\vec{\rho})$ factors are time-independent; if we inserted turbulence into the propagation path, then the WT machinery would generate the corresponding time variations in $f_1(\vec{\rho})$ and $f_2(\vec{\rho})$.

Now we consider Figure 2, the WT model, starting with the sources. The first main point with respect to WT is that the SVE quantity $U_1(\vec{\rho}, t)$, defined by Equation (9), is precisely the complex grid that is produced at the sensor plane by the WT library system **PointSource**.

However, to produce $U_2(\vec{\rho}, t)$, the user must add a customized WT system to explicitly introduce the SVE factor $\exp(-i\Omega_2 t)$. This could be implemented in various ways, one of which is illustrated in Figure 2. In Figure 2, we have introduced a user-created Matlab m-system called **OpdTRampU**. This system works in conjunction with the WT library system **OpdMap**, to create the SVE phase $\exp(-i\Omega_2 t)$ at every space grid point, and to apply that time-varying phase factor to the wavetrain associated with source 2. (In Section 6.1 we explain a few more details, but right now we proceed with the main concepts). Note that **PointSource #2** is assigned the same WaveTrain wavelength (labeled “wvln0”) as **PointSource #1**, because that parameter corresponds to the characteristic frequency factor for both sources, denoted $\exp(-i\omega_1 t)$ in the above equations.

The remaining WaveTrain systems are again existing library systems. The **Combiner** is a system that simply channels both incident wavetrains into the WT propagation system **AtmoPath**. (The **Combiner** is not a physical beamsplitter: it is simply a substitute for having multiple input connections to the **AtmoPath** system). The **AtmoPath** system performs physical (Fresnel) propagation of the two beams to the sensor plane. We can set the atmospheric turbulence in **AtmoPath** to zero strength, because for the present demonstration purposes we want to see walking fringes unperturbed by turbulence effects. Finally, **SimpleFieldSensor** reports the net propagated field (complex sum of the individual U s as discussed above).

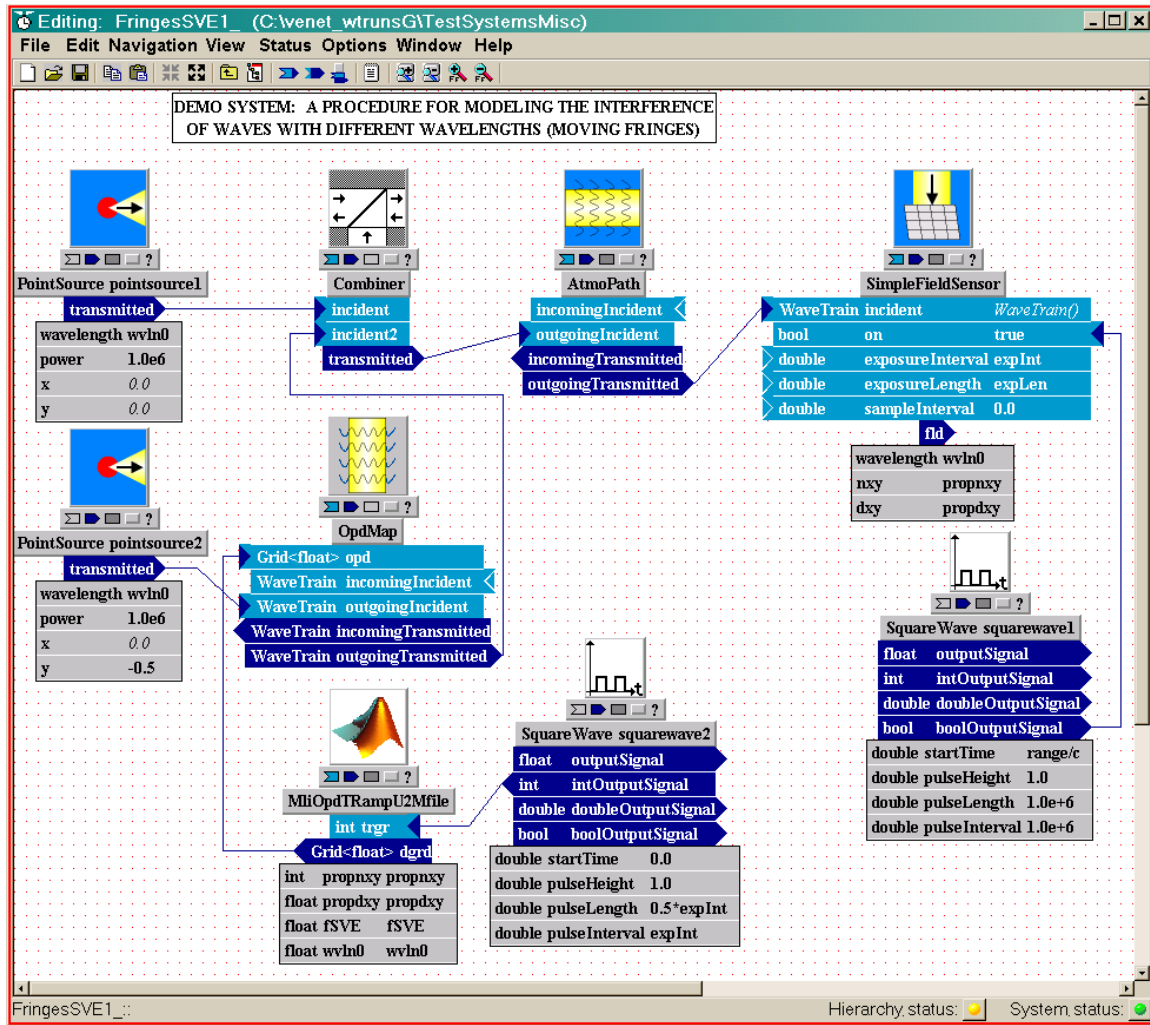


Figure 2: The WaveTrain system corresponding to the physical problem in Figure 1.

4.1 The “bottom line”

Given the modulated source that we have constructed, we can now see why SimpleFieldSensor works even though the field sum is not monochromatic. At every simulation virtual time instant, SimpleFieldSensor, by construction, computes the complex sum of the field grids that it receives, namely the sum $U(\vec{\rho}, t) = U_1(\vec{\rho}, t) + U_2(\vec{\rho}, t)$. But, this is exactly the SVE sum that is needed to compute the ultimately desired net optical irradiance, $I(\vec{\rho}, t) = U(\vec{\rho}, t) U^*(\vec{\rho}, t)$, in the sensor plane. The net irradiance will have spatial variation controlled by the source geometry (the usual “fringes”), but also temporal variation, due to the polychromatic (SVE) nature of the problem. In the above problem, the temporal variation consists of “walking fringes”, moving in the y direction. (As a brief addendum on WT usage, note that the above WT system only outputs the net U , and the final magnitude-squaring operation to get the net I must be done in post processing, after reading the WaveTrain trf-file output).

The main constraint in this general procedure, as far as simulation practicality is concerned, is that the simulation time step must be chosen so that it resolves all SVE time factors introduced by the user: in the preceding example, we must temporally resolve $\exp(-i\Omega_2 t)$. As a numerical example, suppose that the angular frequency spread Ω_2 corresponds to 100kHz. Then the simulation time step would need to be on the order of one microsecond to resolve the time-varying fringes. Note that, if the characteristic wavelength is 1000 nm ($1\mu\text{m}$), then a 100 kHz spread corresponds to a wavelength spread of $3.3\text{E-}7$ nm.

Another constraint in this procedure is that the turbulence effects on the two (or more) different frequencies will have the following (usually very small) error. Since the WaveTrain library modules think that the fields all have the same wavelength (e.g., wvln0 in Figure 2), the field perturbations induced by the turbulent propagation module will not be completely correct. However, it is a fact that the correlation between turbulent irradiance and phase of two beams of different wavelengths is rather high even for frequency differences much greater than typical SVE bandwidths. For any specific application, we would need to keep this potential error in mind, and carry out further estimates if necessary.

5 Procedural summary

The particular SVE dependence that we modeled in the above example represents a special physical case. However, the example clearly illustrates the procedure one must follow to model any polychromatic (but SVE) problem. To recap, the overall procedure is:

(1) Express the source fields in SVE mathematical form.

We say “source fields” as a shorthand: in some problems, the SVE character may be intrinsic to the source, whereas in other problems the SVE character may be imposed by moving or vibrating reflectors, etc.

(2) Construct WaveTrain systems that impose the SVE modulations.

The degree of difficulty of this step may vary greatly depending on the polychromatic nature of the problem. Once the user constructs a few basic systems, they can of course be reused in a variety of problems.

(3) Use SimpleFieldSensor in the standard WT manner, to compute the SVE field superposition.

6 Miscellaneous details and auxiliary remarks

6.1 Some details of the WaveTrain implementation

In section 4, we stated that the combination of two WT systems (OpdTRampU and OpdMap) accomplished the modulation $\exp(-i\Omega_2 t)$. To explain in slightly more detail, the division of labor is as follows. The home-brewed m-system OpdTRampU outputs, at every discrete time instant of the simulation, a 2D spatial array whose value everywhere is an optical path difference $d(t) = -(\text{wvln0}/2\pi) \cdot \Omega_2 t$, in meters. This 2D array is passed by WT to the WT-library system OpdMap, whose function is to grab wavetrains and multiply their complex field grids by a phasor corresponding to the opd spatial array that is input to OpdMap.

The main reason we used this implementation is that it provides a relatively simple way to add the required user customization to WT. A more elegant approach might be to create a single modified OpdMap

system with the required properties. However, this would require C⁺⁺ programming and more understanding of WT at the source code level. The customization of WT using m-systems is considerably simpler, and quite powerful. The user need only know Matlab m-file programming, plus the basic rules of m-system usage as documented in the WT documentation set.

Other implementations than the one diagrammed above, also using a combination of user-defined m-systems and existing WT-library systems, could be used. Some of these others may be superior in efficiency or generality to the approach used above. However, the basic concept is always that the user must in one way or another insert the t-dependent SVE factors that distinguish the different wavelengths.

6.2 Degree of freedom in defining the SVE fields

There is a degree of freedom available to us, when factoring off the characteristic frequency factor to form the SVE factors. In the 2-source example discussed in Section 4, we could have factored out the average frequency

$$\exp\left(-i\frac{\omega_1 + \omega_2}{2}t\right) \quad (11)$$

instead of factoring out $\exp(-i\omega_1 t)$ as we did in our implementation. Although theoretically more elegant, the drawback of the symmetric factorization (11) is that it requires the addition of modulating systems to both sources. On the other hand, a numerical advantage is that it allows a simulation time step twice as long, because the SVE frequency is only half as big as when the asymmetric factorization is used.