



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Subspace Detectors: Efficient Implementation

D. B. Harris, T. Paik

July 27, 2006

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Subspace Detectors: Efficient Implementation

Dave Harris and Tim Paik
Lawrence Livermore National Laboratory
July 26, 2006

Introduction

The optimum detector for a known signal in white Gaussian background noise is the matched filter, also known as a correlation detector [Van Trees, 1968]. Correlation detectors offer exquisite sensitivity (high probability of detection at a fixed false alarm rate), but require perfect knowledge of the signal. The sensitivity of correlation detectors is increased by the availability of multichannel data, something common in seismic applications due to the prevalence of three-component stations and arrays.

When the signal is imperfectly known, an extension of the correlation detector, the subspace detector, may be able to capture much of the performance of a matched filter [Harris, 2006]. In order to apply a subspace detector, the signal to be detected must be known to lie in a signal subspace of dimension $d \geq 1$, which is defined by a set of d linearly-independent basis waveforms. The basis is constructed to span the range of signals anticipated to be emitted by a source of interest.

Correlation detectors operate by computing a running correlation coefficient between a template waveform (the signal to be detected) and the data from a window sliding continuously along a data stream. The template waveform and the continuous data stream may be multichannel, as would be true for a three-component seismic station or an array. In such cases, the appropriate correlation operation computes the individual correlations channel-for-channel and sums the result (Figure 1). Both the waveform matching that occurs when a target signal is present and the cross-channel stacking provide processing gain. For a three-component station processing gain occurs from matching the time-history of the signals and their polarization structure.

The projection operation that is at the heart of the subspace detector can be expensive to compute if implemented in a straightforward manner, i.e. with direct-form convolutions. The purpose of this report is to indicate how the projection can be computed efficiently for continuous multichannel seismic data.

The speed of the calculation is significant as it may become desirable to deploy subspace detectors numbering in the thousands. One application contemplated for these detectors is as screens against signals from repeating sources such as mines or aftershocks of large earthquakes. With many tens of stations and potentially hundreds of sources to screen, efficient implementations are desirable. Speed, of course, can be achieved by procuring faster computers or special-purpose hardware. The approach we examine here is the development of two efficient algorithms that can make the calculations run faster on any machine.

In the first section, we describe the subspace detector as we use it for the detection of repeating seismic events, defining terms and the parameterization used in succeeding sections. This section also reviews how the correlation computations central to the matched filter and subspace detectors can be implemented as a collection of convolution operations. Convolution algorithms using fast Fourier transforms, such as the overlap-add and overlap-save methods, have long been known as efficient implementations of discrete-time finite-impulse-response filters [e.g. Oppenheim and Schaffer, 1975]. These may be extended in a straightforward manner to implement multichannel correlation detectors.

In the second section, we describe how multichannel data can be multiplexed to compute the required convolutions with a single pair of FFT operations instead of a pair for each channel. This approach increases speed approximately twofold.

Seismic data, almost invariably, are oversampled. This characteristic provides an opportunity for increased efficiency by decimating the data prior to performing the correlation calculations. In the third section, we describe a bandpass transformation of the data that allows a more aggressive decimation of the data without significant loss of fidelity in the correlation calculation. The transformation computes a complex-analytic representation for the template waveforms and the multichannel data, followed by a demodulation for both to base-band (i.e. a single band around zero frequency). This approach provides a factor of two to four increase in speed depending on the details of data sampling rate and the desired pass band of the correlation calculation.

The approaches described in the last two sections can be used simultaneously to compound efficiencies.

Formulation of the Detection Problem and Convolution Implementation

Our objective is to detect the occurrence of signals of a particular class in a noisy (multichannel) data stream. The data may be observations of a single seismic trace, a multichannel waveform from a three-component station or a multichannel waveform from an array or network. Data pre-processing may consist of filtering the data into a preferred detection passband followed by decimation to remove dead bandwidth (a high frequency band with zero energy). We consider discrete-time data streams $x_i[n] = x_i(n\Delta t)$, with n an integer time index. The streams are the digitized representations of bandpass analog signals $x_i(t)$ sampled every Δt seconds. The channel index $i = 1, 2, \dots, N_C$ with N_C the total number of data channel streams.

For what follows, it frequently will be convenient to pack the N_C individual streams into a single channel-sequential multiplexed stream:

$$\begin{aligned} x[n \cdot N_C + i - 1] &= x_i[n] \\ n &= -\infty, \dots, \infty \\ i &= 1, \dots, N_C \end{aligned} \tag{1}$$

By convention, when referring to single-channel signals we will use the symbol x_i and when referring to the corresponding multiplexed signals, we will use the symbol x .

Detectors usually are conceived to implement a binary hypothesis test on the presence or absence of a signal in a data observation (detection) window [Van Trees, 1968]. The test chooses between the null hypothesis H_0 , that noise only is present, and the alternative hypothesis H_1 , that both signal and noise are present (Figure 1). Under several possible criteria, the detection rule is a likelihood ratio test that compares the probability that the observed data $\underline{x}[n]$ are due to signal and noise to the probability that they are due to noise alone:

$$\Lambda(\underline{x}[n]) = \frac{p(\underline{x}[n]|H_1)}{p(\underline{x}[n]|H_0)} < \gamma \tag{2}$$

The data vector $\underline{x}[n]$ is a finite segment of the continuous multiplexed stream of equation 1 (Figure 3) beginning at sample n in the individual channel signals:

$$\begin{aligned} \underline{x}[n] &= \\ &\left[x_1[n] \ x_2[n] \ \dots \ x_{N_C}[n] \ \dots \ x_1[n + N_T - 1] \ x_2[n + N_T - 1] \ \dots \ x_{N_C}[n + N_T - 1] \right]^T \end{aligned} \tag{3}$$

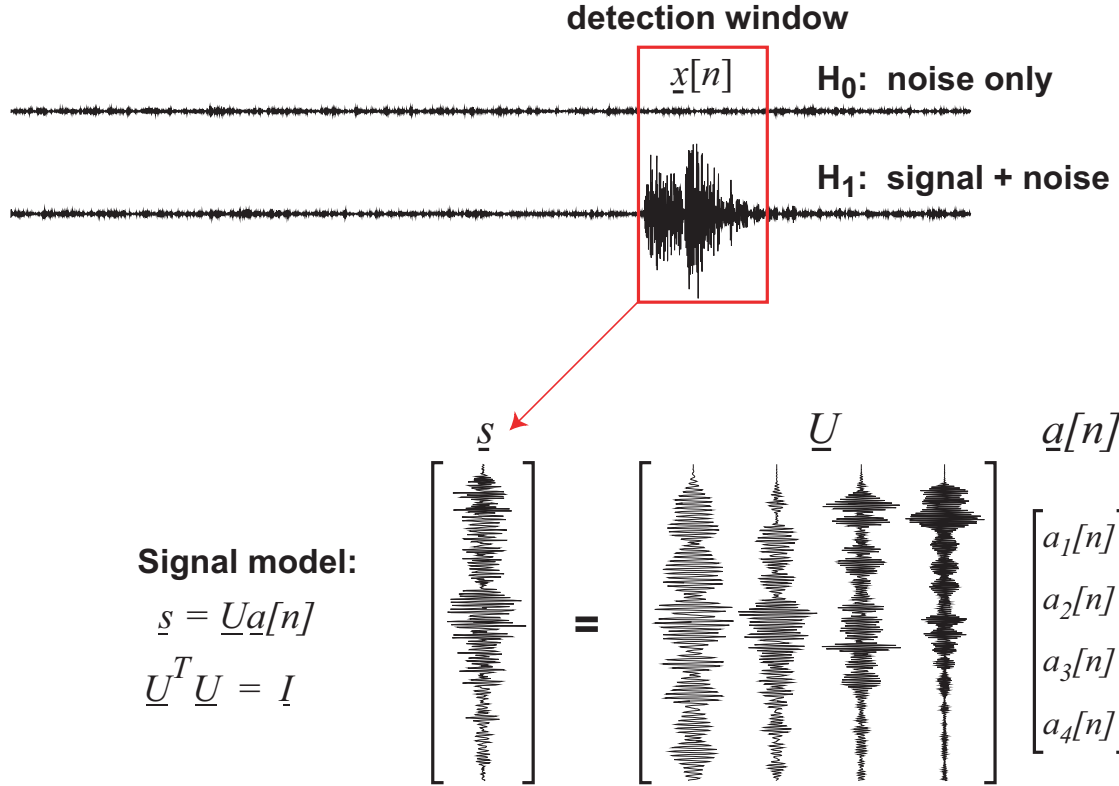


Figure 1 The detection problem is usually formulated for a window sliding along a continuous data stream. At each window position, a binary hypothesis test is conducted, testing whether the data in the window consist of noise alone or signal and noise superimposed. In a subspace detector, the signal is modeled as the sum of weighted basis functions. The weights are considered deterministic, but unknown and must be estimated for each position of the window.

The superscript T denotes the transpose operation. Corresponding to the multiplexed data vector $\underline{x}[n]$, we also define the collection of data vectors for individual channels $\underline{x}_i[n]$:

$$\underline{x}_i[n] = \begin{bmatrix} x_i[n] & x_i[n+1] & \dots & x_i[n+N_T-1] \end{bmatrix}^T \quad (4)$$

Defining the duration of the observation window as N_T samples for each of the individual channels $\underline{x}_i[n]$, the total number of samples comprising $\underline{x}[n]$ is $N = N_T \cdot N_C$.

In this discussion, the probability model for the data is multivariate Gaussian. The data consist of noise alone or signal with additive noise:

$$\begin{aligned} \underline{x}_i[n] &= \underline{\eta}_i[n] && \text{under hypothesis } H_0 \\ \underline{x}_i[n] &= \underline{s}_i[n] + \underline{\eta}_i[n] && \text{under hypothesis } H_1 \end{aligned} \quad (5)$$

The noise $\underline{\eta}_i[n]$ is assumed to be zero-mean, and temporally and spatially uncorrelated (white) with unknown power level (i.e. unknown variance σ^2). In this discussion, we make the simplifying assumption that the noise power is the same on all channels. The signal is assumed to be deterministic, but dependent upon a vector of unknown parameters $\underline{a}[n]$, and is specified as an unknown linear combination of basis waveforms, collectively called the representation:

$$\underline{s}_i[n] = \underline{U}_i \underline{a}[n] \quad (6)$$

The d columns of the $N_T \times d$ representation matrices \underline{U}_i constitute the waveform basis for channel i in sampled form.

This signal model assumes that the source can produce d independently scalable waveforms, as might be the case, for example, with the six independent components of a source moment tensor. Each of the independent components of the source would produce a collection of waveforms over the set of observing channels that would scale up or down in unison. Hence, the coefficients $\underline{a}[n]$ are not a function of the channel, but common to all channels. The columns of the representation matrices \underline{U}_i however do represent the particular waveforms that each independent source component (e.g. Green's function component) will express on the individual observing channels.

Under these assumptions, the probability model for the data under the alternative hypothesis (signal present) is:

$$p(\underline{x}[n]|H_1) = \prod_{i=1}^{N_c} \left(\frac{1}{2\pi\sigma^2} \right)^{N_T/2} \exp \left[-\frac{(\underline{x}_i[n] - \underline{s}_i[n])^T (\underline{x}_i[n] - \underline{s}_i[n])}{2\sigma^2} \right] \quad (7)$$

and the model under the null hypothesis (no signal present) is:

$$p(\underline{x}|H_0) = \prod_{i=1}^{N_c} \left(\frac{1}{2\pi\sigma^2} \right)^{N_T/2} \exp \left[-\frac{\underline{x}_i^T[n] \underline{x}_i[n]}{2\sigma^2} \right] \quad (8)$$

Because there are unknown parameters in the probability densities under each hypothesis, the likelihood ratio test of equation (2) must be modified to use definite parameter values. A reasonable choice for the parameter values is their maximum likelihood estimates given the available data. The likelihood ratio test that results is referred to as the Generalized Likelihood Ratio Test (GLRT) [Van Trees, 1968]:

$$\Lambda(\underline{x}[n]) = \frac{\max_{\{\underline{a}[n], \sigma\}} p(\underline{x}[n]|H_1)}{\max_{\{\sigma\}} p(\underline{x}[n]|H_0)} \quad (9)$$

$$\Lambda(\underline{x}[n]) < \gamma \text{ declare } H_0$$

$$\Lambda(\underline{x}[n]) > \gamma \text{ declare } H_1$$

The GLRT usually gives good performance in the detectors that implement it.

It is simpler to evaluate the log likelihood ratio:

$$l(\underline{x}[n]) = \ln(\Lambda(\underline{x}[n])) = \max_{\{\underline{a}[n], \sigma\}} \ln[p(\underline{x}[n]|H_1)] - \max_{\{\sigma\}} \ln[p(\underline{x}[n]|H_0)] \quad (10)$$

Performing the indicated maximizations (see Appendix A), the result is:

$$l(\underline{x}[n]) = -\frac{N}{2} \ln \left(1 - \underline{b}^T[n] \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{U}_i \right)^{-1} \underline{b}[n] \right) \quad (11)$$

$$\underline{b}[n] = \left(\sum_{i=1}^{N_c} \underline{x}_i^T[n] \underline{x}_i[n] \right)^{-1/2} \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{x}_i[n] \right)$$

Interpretation

Equation (11) has a simple interpretation in terms of a multichannel correlation operation. Considering first the simple case of a single template waveform (i.e. $d = 1$), the representation matrices each have a single column \underline{u}_i . The key calculation of equation (11), then, is the summation:

$$\sum_{i=1}^{N_c} \underline{u}_i^T \underline{x}_i[n] \quad (12)$$

which is a sum of correlation calculations between the observed waveform on the i^{th} channel and the single basis waveform of the representation for that channel:

$$\sum_{i=1}^{N_c} \left(\sum_{j=0}^{N_T-1} u_i[j] x_i[n+j] \right) \quad (13)$$

Here we have referred to the definition of $\underline{x}_i[n]$ found in equation (4). Similarly, the representation vectors \underline{u}_i are defined by:

$$\underline{u}_i = \left[u_i[0] \ u_i[1] \ \dots \ u_i[N_T-1] \right]^T \quad (14)$$

The operation expressed in equation (13) is displayed graphically in Figure 2. As a function of the time index n , the quantity in parentheses in (13) is a correlation calculation between signals defined by u_i and x_i . The objective of a correlation detector is to use a previously-observed template waveform u_i to detect later occurrences of the same or similar waveforms in x_i . When more than one channel of data are available, equation (13) tells us that the appropriate detection operation is to sum the correlations across all channels of data. As shown in the figure, process-

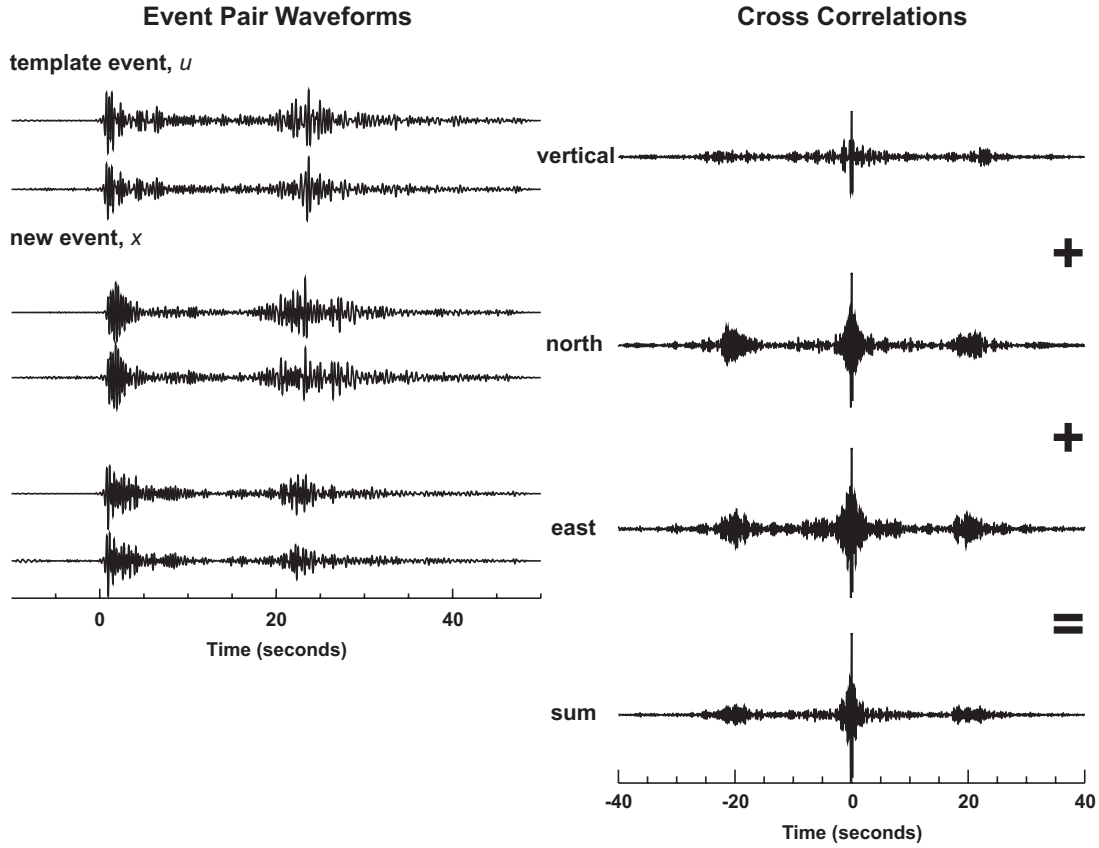


Figure 2 Detection of a known multichannel signal in uncorrelated noise is optimized by an operation that stacks correlations across channels. This operation is demonstrated here for three-component observations of two nearly-identical mining events by the ELKO broadband station. At left are the paired waveforms for the two events for the vertical, north and east channels, respectively. Note that the signals for the two events are highly similar on any one channel, but the signals for either event on the three channels are dissimilar. The corresponding cross-correlations of waveforms for the event pairs on each channel are shown at right. They, too, are dissimilar, but have aligned maxima. Consequently, the sum of the channel-wise correlations develops processing gain as the correlation maxima add coherently.

ing gain is obtained because the correlation peaks are aligned across channels. Alignment occurs because the details of the template signals across channels match the corresponding details of a similar signal (if one is present). In an array signal, for example, the time delays among channels induced by wave propagation across the array are exactly matched by two signals from the same source. Similarly, in the three-component case, the polarization structure among the vertical, north and east channels is matched for similar signals.

Efficient (Convolution) Implementation

Continuing our discussion of the simple ($d = 1$) correlation case, the key to efficient evaluation of the correlation calculation is to realize that the correlation is required continuously, and that it is efficient to calculate it for blocks of data (length $N_B \gg N_T$ samples) for a large number of values of n simultaneously. With the change of variables $j = N_T - 1 - l$, the correlation of equation (13) can be written as a convolution:

$$\sum_{j=0}^{N_T-1} u_i[j]x_i[n+j] = \sum_{l=0}^{N_T-1} u_i[N_T-1-l]x_i[n+N_T-1-l] = \sum_{l=0}^{N_T-1} v_i[l]y_i[n-l] \quad (15)$$

where $v_i[l] = u_i[N_T-1-l]$ is the time-reversed version of the template corresponding to the i^{th} data channel, and $y_i[n] = x_i[n+N_T-1]$ is a shifted version of the i^{th} channel of the input data stream.

Equation (15) can be implemented readily with discrete-time Fourier transforms and the Fast Fourier Transform (FFT) algorithm [Oppenheim and Schaffer, 1975] using the overlap-save or overlap-add methods (see Appendix B for a brief summary of the overlap-add approach). The procedure involves:

1. breaking the data streams $y_i[n]$ into successive, non-overlapping blocks of length N_B ,
2. computing discrete Fourier transforms (DFTs) of the blocks,
3. multiplying the block transforms by the DFT of $v_i[l]$,
4. computing the inverse DFTs of the transform products, and
5. piecing the convolved blocks back into a continuous filtered stream.

Since the DFT of $v_i[l]$ is used over and over, it need be computed only once and stored. The incremental cost (per block of input data) of the convolution of equation (15) consists of one forward DFT, the product and one inverse DFT per channel of data. The cost of computing equation (15) per block of input data over all channels, in a straightforward fashion, is, therefore, $2N_C$ DFTs of length at least $N_T + N_B - 1$.

Multiple Templates: the Subspace Detector

In the event that multiple (d) templates define the detector representation, each of the representation matrices has d columns:

$$\underline{U}_i = \begin{bmatrix} \underline{u}_i^1 & \underline{u}_i^2 & \dots & \underline{u}_i^d \end{bmatrix} \quad (16)$$

and each column \underline{u}_i^j is defined by individual elements $u_i^j[n]$, $n = 0, \dots, N_T - 1$ as in equation (14) by:

$$\underline{u}_i^j = \begin{bmatrix} u_i^j[0] & u_i^j[1] & \dots & u_i^j[N_T - 1] \end{bmatrix}^T \quad (17)$$

The log likelihood function (equation 11) is complicated in this case, requiring a set of d convolutions of the type expressed in equation (15) to be calculated. This complexity motivates the change in representation described in the next section.

Efficient Multichannel Convolution

The log likelihood ratio of equation (11) can be substantially simplified if the structure of the data stream and detection templates are recast from collections of individual channels to single multiplexed sequences in channel-sequential form (equations 1 and 3). In addition, an efficient algorithm can be devised for implementing the correlations indicated in equations (11) and (13) when multiplexed forms of the data streams and templates are used. The basic idea is to correlate the multiplexed template directly with the multiplexed data stream, instead of computing the sum of the individual channel correlations as in equation (13). With the appropriate decimation of the multiplexed correlation (implemented as an aliasing operation in the frequency domain), roughly a factor of two savings in computation is possible.

The signal model of equation (6) can be recast more simply as:

$$\underline{s}[n] = \underline{U}\underline{a}[n] \quad (18)$$

The representation matrix \underline{U} consists of d column vectors:

$$\underline{U} = [\underline{u}^1 \ \underline{u}^2 \ \dots \ \underline{u}^d] \quad (19)$$

which are multiplexed column vectors from the corresponding single-channel representation matrices \underline{U}_i . For example, the j^{th} column of \underline{U} is:

$$\underline{u}^j = \left[u_1^j[0] \ u_2^j[0] \ \dots \ u_{N_C}^j[0] \ \dots \ u_1^j[N_T-1] \ u_2^j[N_T-1] \ \dots \ u_{N_C}^j[N_T-1] \right]^T \quad (20)$$

The elements $u^j[\cdot]$ of the multiplexed template \underline{u}^j can be represented in terms of the elements $u_i^j[\cdot]$ of the single-channel templates \underline{u}_i^j by:

$$\begin{aligned} u^j[k \cdot N_C + i - 1] &= u_i^j[k] \\ k &= 0, \dots, N_T - 1 \\ i &= 1, \dots, N_C \end{aligned} \quad (21)$$

Without loss of generality, \underline{U} can be made orthonormal:

$$\underline{U}^T \underline{U} = \underline{I} \quad (22)$$

where \underline{I} is the $d \times d$ identity matrix. Figure 3 depicts the organization of the data and the template in channel-sequential multiplexed form, and, in particular, demonstrates that the single-channel elements of the template and data can be made to correspond in the multiplexed forms.

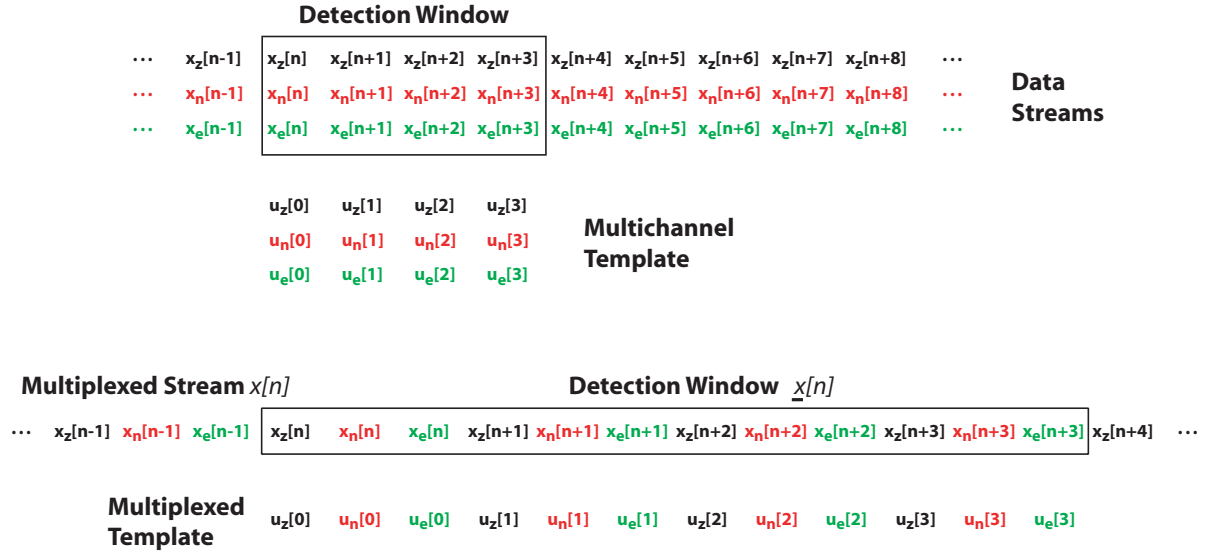


Figure 3 To simplify mathematical expressions and to support an efficient correlation algorithm, it is convenient to reformat multichannel data and signal-matching templates into channel-sequential multiplexed form. The example above shows a portion of a three-component data stream with vertical, north and east (z,n,e) channels and a corresponding three-component template in the upper part of the diagram. In a correlation-matching operation, the template effectively overlays the data in the detection window. The product of data and template samples from corresponding channels (z with z, n with n and e with e) is formed and the resulting products are summed to form the correlation value, as in Figure 1. In the bottom part of the diagram, the same sum of products can be formed by a simple dot product of data and template when reordered into single channel-sequential multiplexed vectors or streams.

Under these assumptions, the probability density for the observed data is

$$p(\underline{x}[n]|H_0) = \left[\frac{1}{2\pi\sigma^2} \right]^{N/2} \exp\left(-\frac{1}{2\sigma^2} \underline{x}^T[n] \underline{x}[n]\right) \quad (23)$$

under the null hypothesis (no signal present) and

$$p(\underline{x}[n]|H_1) = \left[\frac{1}{2\pi\sigma^2} \right]^{N/2} \exp\left(-\frac{1}{2\sigma^2} (\underline{x}[n] - \underline{U}\underline{a}[n])^T (\underline{x}[n] - \underline{U}\underline{a}[n])\right) \quad (24)$$

under the alternative hypothesis (signal present). The data vector $\underline{x}[n]$ is as defined in equation (3).

Using the multiplexed representation, the log generalized likelihood ratio of equation (10) assumes the form [Harris, 2006]:

$$l(\underline{x}[n]) = -\frac{N}{2} \left(\frac{\underline{x}^T[n] \underline{x}[n] - \underline{x}_p^T[n] \underline{x}_p[n]}{\underline{x}^T[n] \underline{x}[n]} \right) = -\frac{N}{2} (1 - c[n]) \quad (25)$$

where $\underline{x}_p[n] = \underline{U}\underline{U}^T \underline{x}[n]$, and $c[n] = \frac{\underline{x}_p^T[n] \underline{x}_p[n]}{\underline{x}^T[n] \underline{x}[n]}$

which is considerably simpler than equation (11). The vector $\underline{x}_p[n]$ is the least-squares estimate of the signal in the detection window, and is obtained by projecting the data in the detection window into the subspace defined by \underline{U} (see Figure 4). The quantity $c[n]$ is the ratio of the energy in the projected data to the energy in the original data. It is a positive quantity with values ranging between 0 and 1. It closely resembles the square of a correlation coefficient (it is exactly that for the case $d = 1$, i.e. the correlation detector). This quantity $c[n]$ is the sufficient statistic for the subspace detector, and is the quantity that we seek to calculate efficiently.

The key quantities requiring efficient evaluation are the numerator $\underline{x}_p^T[n] \underline{x}_p[n]$ and the denominator $\underline{x}^T[n] \underline{x}[n]$ of the sufficient statistic. Because the representation matrix \underline{U} is orthonormal, the numerator can be written as $\underline{x}_p^T[n] \underline{x}_p[n] = \hat{\underline{a}}^T[n] \hat{\underline{a}}[n]$ where $\hat{\underline{a}}[n] = \underline{U}^T \underline{x}[n]$ is the least-squares estimate of the signal coefficient vector $\underline{a}[n]$ defined in equations (6) and (16) under the assumption of uncorrelated noise. We examine the calculation of $\hat{\underline{a}}[n]$ in detail first.

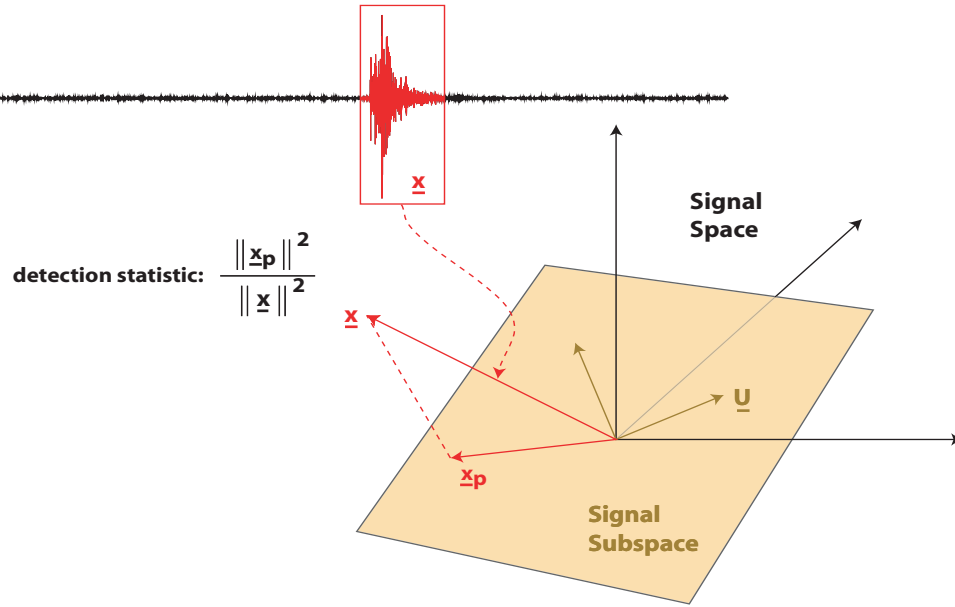


Figure 4 The subspace detector operates by projecting the (possibly multichannel) data in a detection window (treated as a vector \underline{x}) into a subspace spanned by the columns of the subspace representation matrix \underline{U} . The ratio of the squared norm of the projected vector to the squared norm of the original data vector is the detection statistic. This statistic resembles a correlation coefficient in that it ranges between 0 and 1 and measures the linear dependence between the data and the set of vectors comprising the subspace representation. This statistic is computed continuously as the data window slides down the continuous data stream one sample at a time. When the statistic exceeds a predefined threshold value, a detection is declared.

Vector $\hat{\underline{q}}[n]$ is a $d \times 1$ vector with individual elements $\hat{a}^i[n]$ that are the inner products between the columns of \underline{U} and the multiplexed data vector $\underline{x}[n]$:

$$\hat{a}^j[n] = (\underline{u}^j)^T \underline{x}[n] \quad (26)$$

By reference to equations (3) and (20), this expression can be rewritten as the double sum:

$$\hat{a}^j[n] = \sum_{k=0}^{N_T-1} \sum_{i=1}^{N_C} u_i^j[k] x_i[n+k] \quad (27)$$

Similarly, by reference to equations (1) and (21), (27) can be expressed in terms of the multiplexed template and data stream:

$$\hat{a}^j[n] = \sum_{i=1}^{N_C} \sum_{k=0}^{N_T-1} u^j[k \cdot N_C + i - 1] x[(n+k) \cdot N_C + i - 1] \quad (28)$$

With the change of variables $l = k \cdot N_C + i - 1$, this expression can be simplified to:

$$\hat{a}^j[n] = \sum_{l=0}^{N_T \cdot N_C - 1} u^j[l] x[n \cdot N_C + l] \quad (29)$$

and interpreted as the correlation between the multiplexed template and multiplexed data stream sampled at integer multiples of N_C . Here, the multiplexed template and data stream are being treated as normal scalar time series. Only every N_C^{th} sample of the correlation function between these two time series produces a legitimate value equal to the sum of correlations expressed in equation (13).

The basic concept is illustrated in Figure 5, which shows that the convolution between multiplexed data streams produces legitimate summed correlation samples at offsets corresponding to integer multiples of N_C . The trick is to compute just those samples and not the others corresponding to products between mismatched channels. This outcome is achieved by recognizing that a decimated version (decimated by a factor of N_C) of the correlation between multiplexed template and multiplexed data stream is required.

Decimation in time corresponds to an aliasing operation in the frequency domain (see Appendix C). Consequently, the overlap-add algorithm must be modified to incorporate an aliasing operation in the frequency domain. The computational savings arises from the fact that although a length $(N_T + N_B - 1) \cdot N_C$ forward FFT of the multiplexed data must be computed to calculate

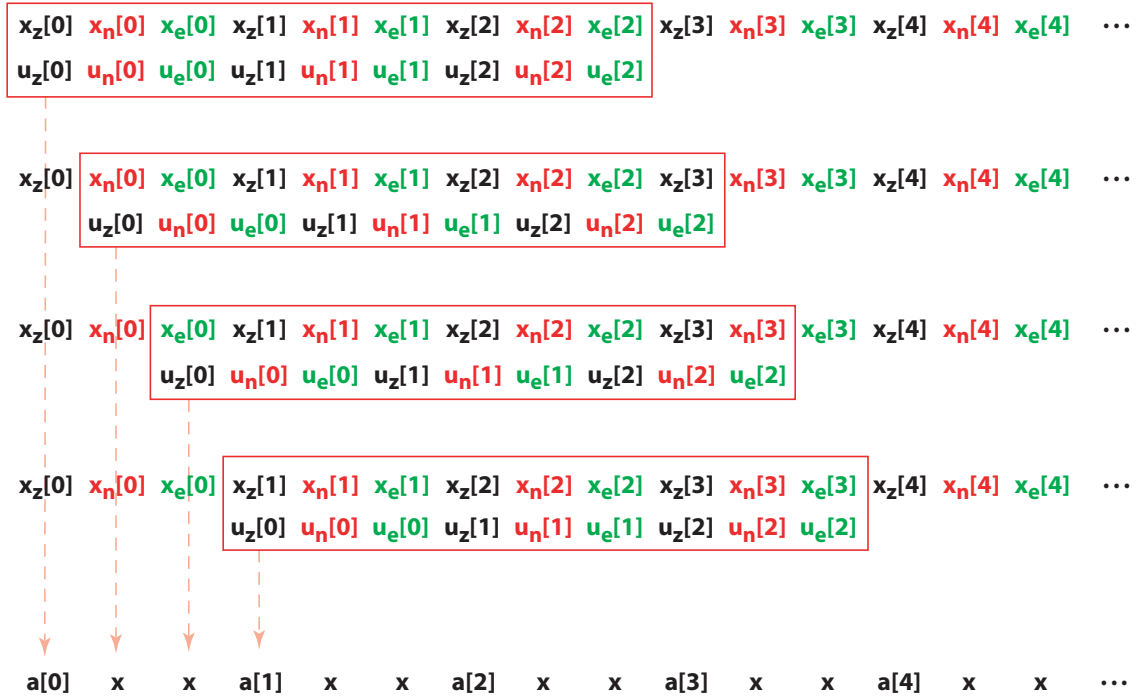


Figure 5 Direct correlation of a multiplexed data stream with a multiplexed template results in a sequence that has valid correlations every N_C samples. The example above depicts correlation between a channel-sequential multiplexed three-component data stream and a corresponding multiplexed three-sample ($N_T = 3$ multiplexed template. In the cross-correlation sequence at the bottom of the figure, a valid multichannel correlation appears every third sample. The trick behind efficient evaluation of the valid samples is to realize that a decimated version of the multichannel correlation sequence is desired. This objective is accomplished by aliasing in the frequency domain.

the frequency domain product of data and template, only a length $N_T + N_B - 1$ inverse FFT is required to compute the time-domain samples of the correlation function.

The steps in the overlap-add algorithm for a single data block consist of:

1. Filtering and decimating the data as a preprocessing step.
2. Reading a buffer of N_B samples for each of the N_C channels of the data stream, and multiplexing them into a single length $N_B \cdot N_C$ channel-sequential multiplexed sequence.
3. Padding the sequence with zeros to $(N_T + N_B - 1) \cdot N_C$ points and computing it's $(N_T + N_B - 1) \cdot N_C$ length DFT. The DFT actually can be larger than this length with suitable zero-padding, but must be an integer multiple of N_C to support the aliasing operation.
4. Multiplying the DFT of the multiplexed data by the DFT of the multiplexed template. The template DFT need be computed only once and stored, since it is used repeatedly.
5. Aliasing the product DFT by a factor of N_C , resulting in an aliased DFT of length $N_T + N_B - 1$.
6. Computing the inverse of the aliased product DFT, overlapping with the previous block calculation and adding the results.

The complete calculation of the numerator of $c[n]$ in equation (25) sums the power in the correlation stacks over all templates:

$$\hat{\underline{a}}^T[n] \hat{\underline{a}}[n] = \sum_{i=1}^d (\hat{a}^i[n])^2 \quad (30)$$

The structure for performing this computation for all d of the templates is shown in Figure 6. The algorithm decreases the computational cost by a factor of approximately

$$\frac{N_C + 1}{2N_C} + \frac{\log N_C}{2\log(N_T + N_B)} \quad (31)$$

The logarithm is taken with respect to base 2 if radix-2 FFTs are used in the calculation. As the number of channels increases, this factor tends to $1/2$, assuming $N_T + N_B \gg N_C$, as is usually the case.

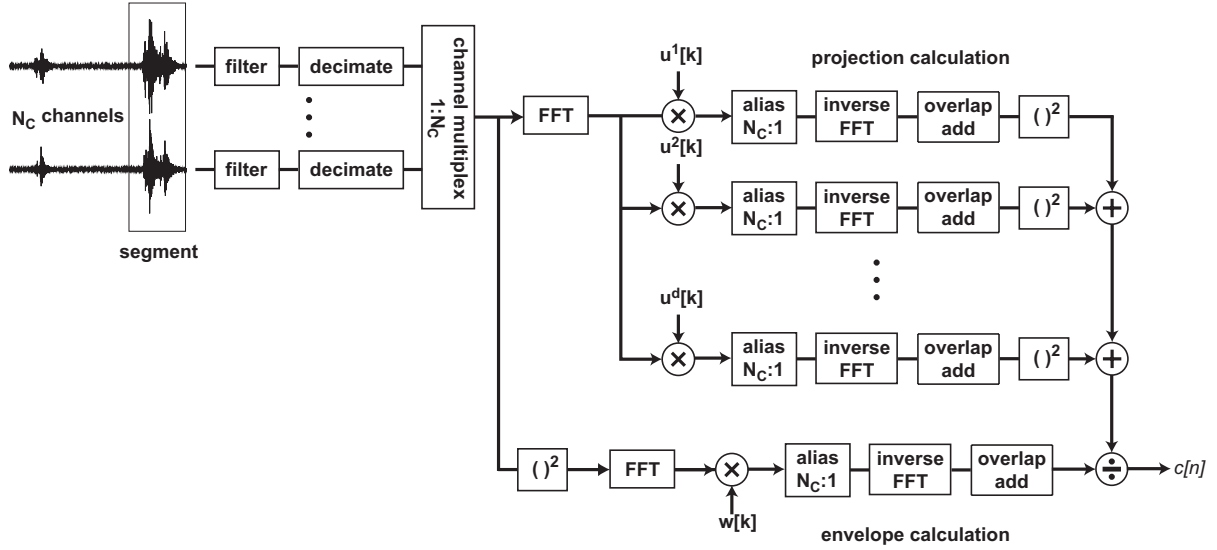


Figure 6 The multichannel correlation algorithm doubles the speed of subspace detection statistic calculations. The data from N_C channels are filtered into the detection passband, decimated to eliminate dead bandwidth, then multiplexed into a single channel-sequential multiplexed data stream. A block of the multiplexed data is transformed with an FFT and the resulting DFT is multiplied by the DFTs of each of the d similarly multiplexed templates. These frequency-domain products are each aliased by a factor equal to the number of channels and a smaller inverse FFT is used to calculate the resulting correlations between the templates and the data. The appropriate overlap-add calculations are performed to synthesize the continuous correlations from the constituent block calculations, then squared and added to form the numerator of the detection statistic. A similar calculation on the square of the multiplexed signal is performed to generate the denominator of the detection statistic (labeled envelope calculation).

Denominator calculation

The denominator $\underline{x}^T[n] \underline{x}[n]$ of the sufficient statistic (equation 25) also can be evaluated efficiently using a multiplexed overlap-add type algorithm as indicated in the bottom of Figure 6 (labeled envelope calculation). The denominator is a running sum of squares of the filtered data stream. The sum is performed over the time interval defined by the detection window:

$$\underline{x}^T[n] \underline{x}[n] = \sum_{k=1}^{N_C} \sum_{j=0}^{N_T-1} x_k^2[n+j] = \sum_{k=1}^{N_C} \sum_{j=0}^{N_T-1} x^2[(n+j) \cdot N_C + k - 1] \quad (32)$$

Simplifying:

$$\underline{x}^T[n] \underline{x}[n] = \sum_{l=0}^{N_T \cdot N_C - 1} w[l] x^2[n \cdot N_C + l] \quad (33)$$

where we have made the change of variables $l = j \cdot N_C + k - 1$. The essential insight for an efficient implementation is that the running sum of squares on the multiplexed data sequence can be implemented as a convolution with a “boxcar” function $w[l]$, i.e. a constant sequence consisting of $N_T \cdot N_C$ successive ones. In equation (23), the sum describes a correlation operation, but with a change of variables similar to that used in equation (15), it can be implemented as a convolution.

As in the case of the multiplexed convolution to evaluate the numerator term, this calculation is evaluated only at integer multiples of N_C . This calculation also corresponds to a decimation which can be implemented efficiently (as in Figure 6) with aliasing in the frequency domain.

Bandpass Transformation to Maximize Decimation

Another technique for improving computational efficiency, usually more effective than the multichannel convolution method of the last section, is aggressive waveform decimation following a bandpass transformation. This technique can be used in conjunction with the multichannel convolution technique to compound efficiencies.

As remarked earlier, it usually is the case that seismic data are oversampled. Oversampling is pronounced when the data are filtered with a bandpass filter prior to correlation detection. Bandpass filtering is a preprocessing step used to exclude portions of the signal spectrum where the signal-to-noise ratio is low, or to limit the signal to a (typically low-frequency) band where the correlations among signals from proximate events are high.

As depicted in Figure 7 for a single channel of data, there can be significant dead bandwidth (i.e. frequency bands with no signal energy) following bandpass filtering. One simple approach to removing some of the dead bandwidth is to decimate the data, i.e. to construct a new data sequence from the original sequence by retaining only every R^{th} sample of the original:

$$\tilde{x}_l[n] = x_l[nR] \quad (34)$$

The top line of Figure 7 depicts the result of decimation by a factor of two, which eliminates the upper half of the available bandwidth. In this depiction, the decimation operation still leaves a significant amount of dead bandwidth, particularly around d.c. (zero Hertz).

A more sophisticated approach, allowing greater elimination of dead bandwidth, is to transform the data to retain only a single sideband of the signal's Fourier transform then to demodulate this sideband to baseband (d.c.) prior to decimation. This alternative is shown in the lower part of Figure 7. The resulting data sequence is complex, in fact a demodulated complex analytic representation of the original sequence. The process consists of the following steps for an original signal $x_l[n]$ which has been bandpass filtered around center frequency f_c :

1. Demodulate the signal (shift it downward in frequency by f_c):

$$y_l[n] = x_l[n]e^{i2\pi f_c n} \quad (35)$$

2. Filter the demodulated signal with a lowpass filter to eliminate the lower (negative-frequency) sideband:

$$z_l[n] = \sum_k y_l[k]h_{lp}[n-k] \quad (36)$$

3. Decimate the data:

$$\tilde{z}_l[n] = z_l[nR] \quad (37)$$

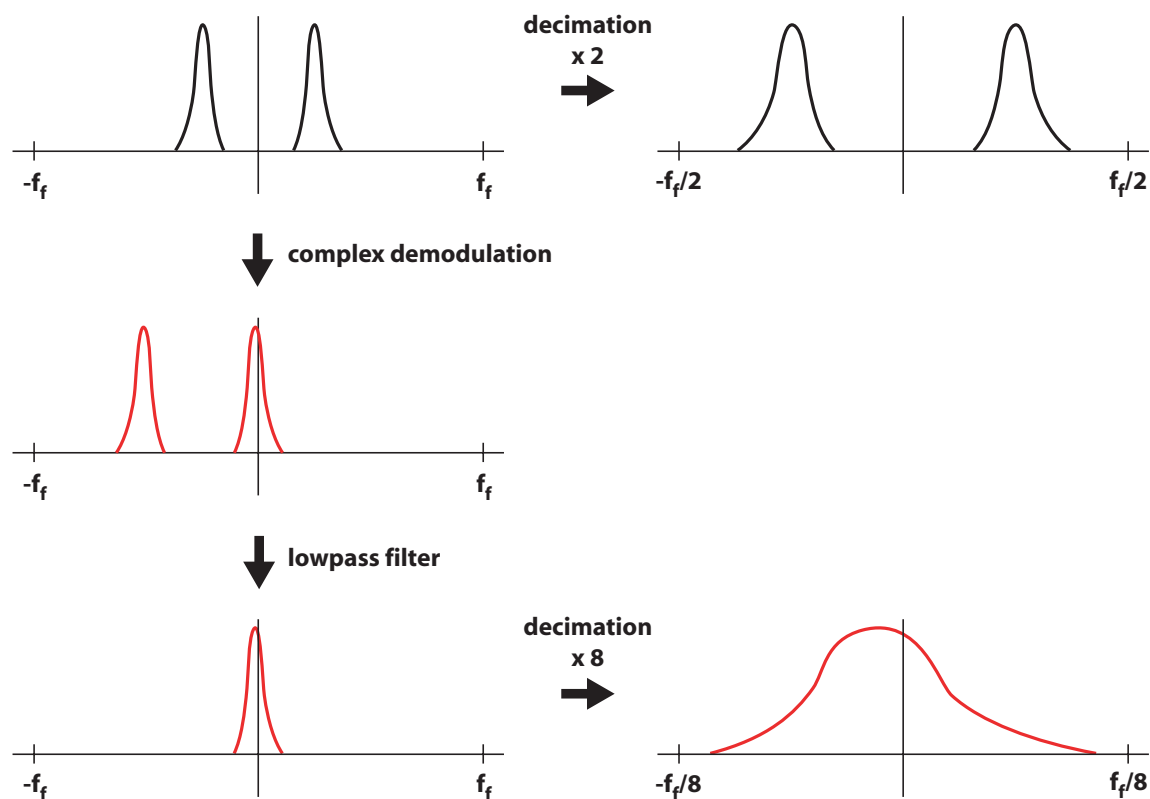


Figure 7 The most significant efficiency improvement is achieved by decimating the data prior to performing the correlation calculation. Often decimation at a modest rate is possible with the simple decimation as depicted in the top of this figure. The figure at upper left shows a bandpass spectrum for a signal which does not occupy all of the available data bandwidth, a common occurrence with seismic data. The available digital band ranges from $-f_f$ to f_f (the folding frequencies) Hertz. The spectrum at upper right results when the data are simply decimated by a factor of 2. A more complex data transformation, shown in the middle and lower parts of the figure, allows more aggressive decimation. The transformation consists of three steps: a complex demodulation to move the positive-frequency sideband to baseband, a lowpass filter operation to eliminate the negative-frequency sideband, and a decimation operation to fill the available frequency band with positive-frequency sideband. The transformation results in complex sequences; the spectra that correspond to complex sequences are shown in red and those that correspond to real sequences are shown in black.

Here $h_{lp}[n - k]$ is the impulse response of a lowpass filter, which, in practice, we implement with a Butterworth IIR filter.

Of course, both the template and the continuous data stream must be transformed in identical preprocessing steps for the correlation calculation to be unaltered by the transformation. That the correlation calculation carried out upon the baseband complex analytic versions of the signals is equivalent to the calculation on the original real signals is demonstrated in Appendix D.

The virtue of this more complicated approach is that it is possible to choose a much larger decimation factor R than would be the case with simple decimation. How large a factor can be determined approximately by the bandwidth $2f_b$ of the data and the center frequency (assuming the positive sideband has support only in the frequency range $f_c \pm f_b$). In the simple decimation case, the largest feasible decimation factor is $\lfloor f_c / (f_c + f_b) \rfloor$, where the symbol $\lfloor \cdot \rfloor$ means “largest integer less than”. For the bandpass transformation case, the largest possible decimation factor is $\lfloor f_c / f_b \rfloor$. The ratio between them is roughly $(f_c + f_b) / f_b$. In the case of data filtered into the 2 to 4 Hertz band, this ratio is approximately 4. Because the signals that result from the bandpass transformation are complex instead of real, the number of calculations required for all operations (FFTs, etc.) upon them double. Consequently, the gain in processing speed for our example is more like a factor of 2. The processing speed gain becomes more favorable to the bandpass transformation the larger is the center frequency of the data with respect to its bandwidth.

In practice we find that very aggressive decimation can result in the peak of the detection statistic being poorly sampled when a signal is present. The net result can be a significant attenuation of the peak correlation value. Nonetheless, it is possible to achieve significant increases in speed by implementing this strategy and making judicious choices in the decimation rate.

The processing structure depicted in Figure 6 must be modified to account for the complex demodulation operation and the fact that many of the partial correlation results have to be carried as complex sequences. The resulting modifications to the multichannel convolution operation are shown in Figure 8, which demonstrates how both ideas described in this report can be combined to increase efficiency.

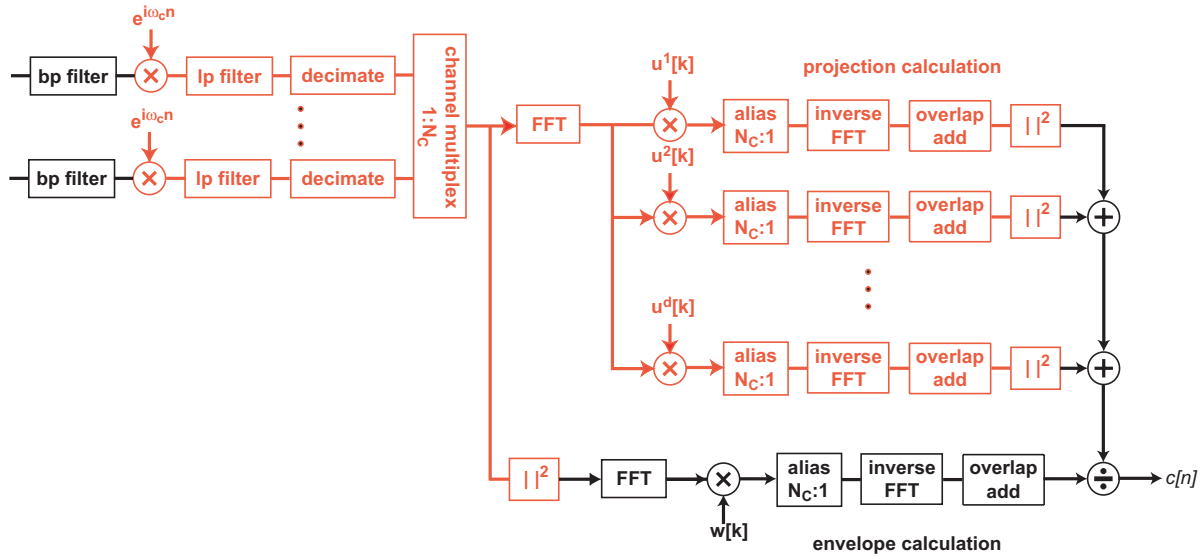


Figure 8 Structure for computing the subspace detection statistic using the bandpass transformation and multichannel convolution techniques. This processing structure often is a factor of 4 or more efficient than straightforward decimation and convolution with the overlap-save algorithm. The portion of the calculation that is carried out with complex sequences is shown in red; that carried out with real sequences is shown in black.

References

Harris, D. B. (2006), “Subspace Detectors: Theory”, Lawrence Livermore National Laboratory Technical Report UCRL-TR-222758, 46 pages, Livermore, CA, July 12, 2006.

Oppenheim, A. V. and R. W. Schaffer (1975), *Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

Van Trees, H. L. (1968), *Detection, Estimation and Modulation Theory*, vol. 1, John Wiley and Sons, New York.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

Appendix A Evaluation of the Log Likelihood Ratio

Evaluation of the log likelihood ratio of equation (10):

$$l(\underline{x}[n]) = \ln(\Lambda(\underline{x}[n])) = \max_{\{\underline{a}[n], \sigma\}} \ln[p(\underline{x}[n]|H_1)] - \max_{\{\sigma\}} \ln[p(\underline{x}[n]|H_0)] \quad (\text{A.1})$$

can be done independently for the numerator and denominator separately. Taking the numerator portion first (see equation 7):

$$\ln(p(\underline{x}[n]|H_1)) = - \sum_{i=1}^{N_c} \left(\frac{N_T}{2} \ln(2\pi\sigma^2) + \frac{(\underline{x}_i[n] - \underline{U}_i \underline{a}[n])^T (\underline{x}_i[n] - \underline{U}_i \underline{a}[n])}{2\sigma^2} \right) \quad (\text{A.2})$$

The partial derivatives of this expression with respect to the unknown parameters (i.e. the partial derivative with respect to σ and the gradient with respect to $\underline{a}[n]$) are set to zero:

$$\begin{aligned} \frac{-N_c N_T}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{N_c} (\underline{x}_i[n] - \underline{U}_i \underline{a}[n])^T (\underline{x}_i[n] - \underline{U}_i \underline{a}[n]) &= 0 \\ \sum_{i=1}^{N_c} \underline{U}_i^T (\underline{x}_i[n] - \underline{U}_i \underline{a}[n]) &= 0 \end{aligned} \quad (\text{A.3})$$

Simplifying these equations leads to:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N_c} (\underline{x}_i[n] - \underline{U}_i \underline{a}[n])^T (\underline{x}_i[n] - \underline{U}_i \underline{a}[n]) \quad (\text{A.4})$$

$$\underline{a}[n] = \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{U}_i \right)^{-1} \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{x}_i[n] \right)$$

Substituting the expression for σ^2 in (A.4) into (A.2) and simplifying:

$$\ln(p(\underline{x}[n]|H_1)) = -\frac{N}{2} \left(\ln \left(2\pi \frac{1}{N} \sum_{i=1}^{N_c} (\underline{x}_i[n] - \underline{U}_i \underline{a}[n])^T (\underline{x}_i[n] - \underline{U}_i \underline{a}[n]) \right) + 1 \right) \quad (\text{A.5})$$

The denominator portion of the log likelihood ratio is:

$$\ln(p(\underline{x}[n]|H_0)) = - \sum_{i=1}^{N_c} \left(\frac{N_T}{2} \ln(2\pi\sigma^2) + \frac{\underline{x}_i^T[n]\underline{x}_i[n]}{2\sigma^2} \right) \quad (\text{A.6})$$

This portion is maximized by setting the derivative of this expression with respect to σ to zero:

$$\frac{-N_c N_T}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{N_c} \underline{x}_i^T[n]\underline{x}_i[n] = 0 \quad (\text{A.7})$$

The maximizing value of σ is:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N_c} \underline{x}_i^T[n]\underline{x}_i[n] \quad (\text{A.8})$$

Inserting (A.8) into (A.6) and simplifying:

$$\ln(p(\underline{x}[n]|H_0)) = -\frac{N}{2} \left(\ln \left(2\pi \frac{1}{N} \sum_{i=1}^{N_c} \underline{x}_i^T[n]\underline{x}_i[n] \right) + 1 \right) \quad (\text{A.9})$$

Assembling all the pieces into (A.1):

$$l(\underline{x}[n]) = -\frac{N}{2} \ln \left(\frac{\sum_{i=1}^{N_c} (\underline{x}_i[n] - \underline{U}_i \underline{a}[n])^T (\underline{x}_i[n] - \underline{U}_i \underline{a}[n])}{\sum_{i=1}^{N_c} \underline{x}_i^T[n]\underline{x}_i[n]} \right) \quad (\text{A.10})$$

Inserting the expression for $\underline{q}[n]$ in (A.4) into (A.10) and simplifying, we obtain the desired result:

$$l(\underline{x}[n]) = -\frac{N}{2} \ln \left(\frac{\sum_{i=1}^{N_c} \underline{x}_i^T[n] \underline{x}_i[n] - \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{x}_i[n] \right)^T \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{U}_i \right)^{-1} \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{x}_i[n] \right)}{\sum_{i=1}^{N_c} \underline{x}_i^T[n] \underline{x}_i[n]} \right) \quad (\text{A.10})$$

which can be recast in a normalized form:

$$l(\underline{x}[n]) = -\frac{N}{2} \ln \left(1 - \underline{b}^T[n] \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{U}_i \right)^{-1} \underline{b}[n] \right) \quad (\text{A.10})$$

$$\underline{b}[n] = \left(\sum_{i=1}^{N_c} \underline{x}_i^T[n] \underline{x}_i[n] \right)^{-1/2} \left(\sum_{i=1}^{N_c} \underline{U}_i^T \underline{x}_i[n] \right)$$

Appendix B Overlap-Add Implementation of Convolutions

The convolution operation of equation (15) describes the essential quantities to be evaluated in correlation and subspace detectors:

$$\sum_{l=0}^{N_T-1} v_i[l] y_i[n-l] \quad (\text{B.1})$$

For simplicity in this discussion, we drop the superscripts and subscripts and examine the evaluation of the basic convolution operation:

$$a[n] = \sum_{l=0}^{N_T-1} v[l] y[n-l] \quad (\text{B.2})$$

It is efficient to calculate $a[n]$ in blocks of N_B values simultaneously using discrete Fourier transforms implemented with Fast Fourier Transform (FFT) algorithms. To do so, the data $y[n]$ are acquired in blocks of length N_B :

$$y[kN_B] \quad y[kN_B + 1] \quad \dots \quad y[kN_B + N_B - 1] \quad (\text{B.3})$$

where $k = \dots, -1, 0, 1, \dots$.

It is conceptually useful to represent the blocks with a gating operation:

$$y^k[n] = w[n - kN_B] y[n] \quad (\text{B.4})$$

where

$$w[n] = 0 \quad n < 0; \quad n \geq N_B$$

$$w[n] = 1 \quad 0 \leq n < N_B$$

Each signal $y^k[n]$ representing a block is non-zero only in the finite interval defining the block and zero elsewhere. The continuous data stream can be reconstructed from the blocks with the infinite sum:

$$y[n] = \sum_{k=-\infty}^{\infty} y^k[n] \quad (\text{B.5})$$

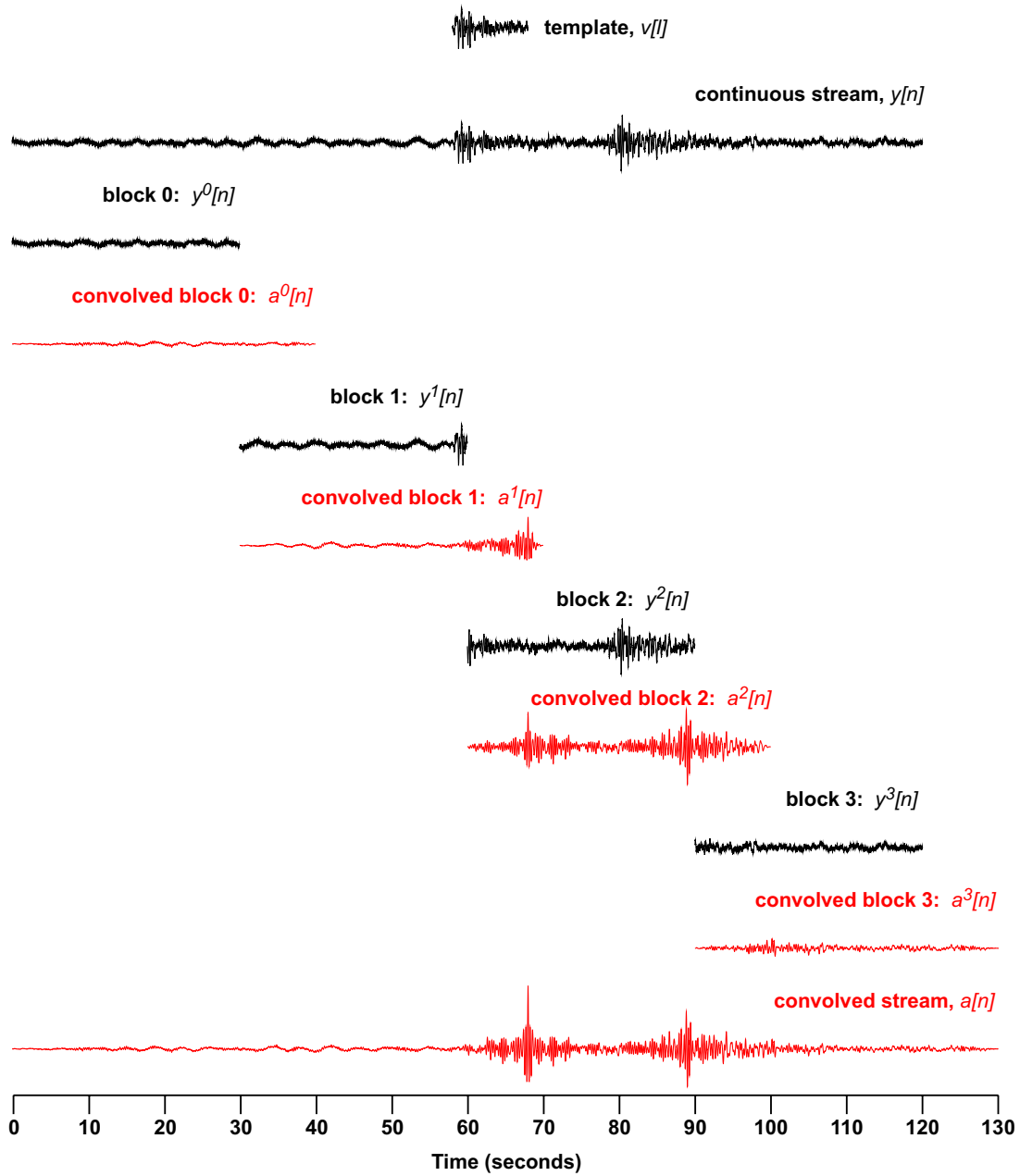


Figure B1 Example of the overlap-add algorithm used to convolve a template waveform with a continuous data stream. The stream is broken into successive non-overlapping blocks and each of these is convolved with the template. The resulting convolved blocks are overlapped and summed to form the desired continuous convolution.

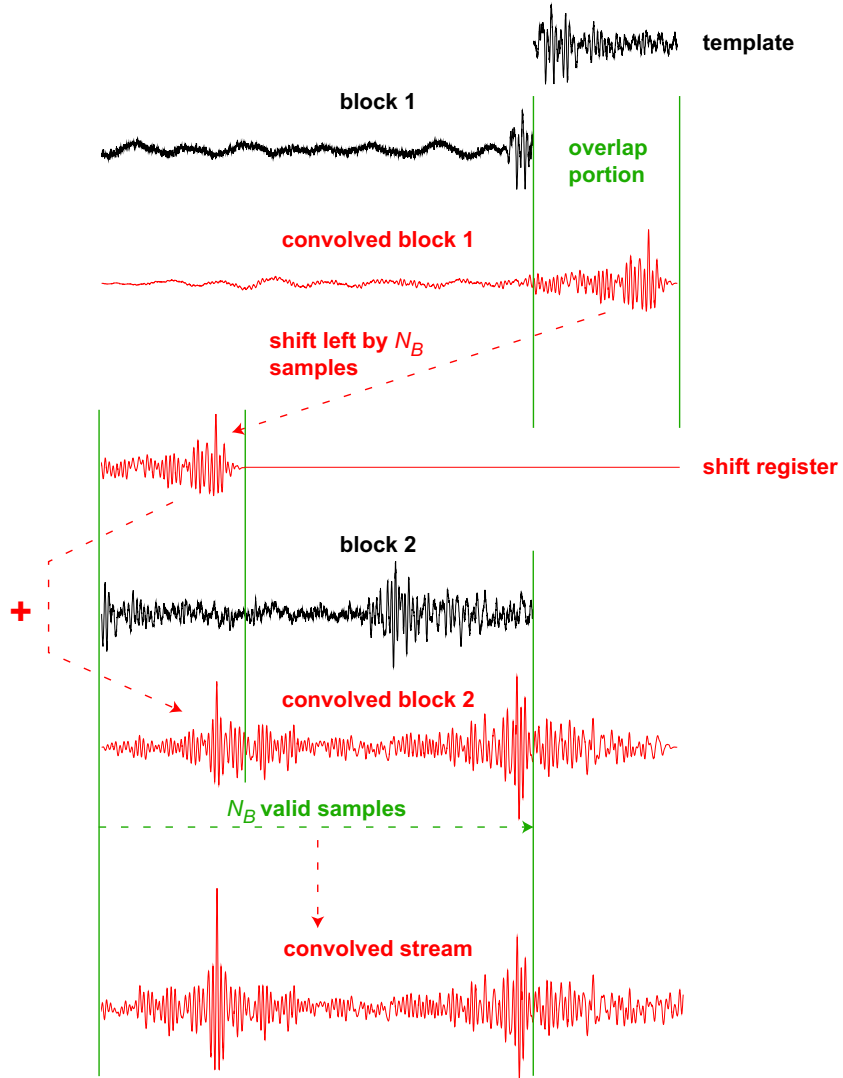


Figure B2 A simple shift-register simplifies the overlap-add operation. The data from one block are loaded into an array and convolved with the template to form an $N_B + N_T - 1$ sample partial result. When the last $N_T - 1$ points of the previous calculation are added to the first $N_T - 1$ points of the current calculation, N_B valid samples result, which can be loaded into a buffer containing the continuous output (convolved) stream. It is convenient to maintain the current resulting calculation in a shift register and shift the contents N_B samples to the left (zero-padding) to set up the calculation of the next block convolution.

By inserting (B.5) into (B.2), the convolution of the continuous stream can be written in terms of the sum of convolutions of the individual blocks:

$$a[n] = \sum_{k=-\infty}^{\infty} a^k[n] \quad (\text{B.6})$$

$$a^k[n] = \sum_{l=0}^{N_T-1} v[l]y^k[n-l]$$

Figure B1 illustrates the process of convolving the length- N_B data blocks with the length- N_T template ($v[l]$). The resulting convolved blocks are sequences of length $N_B + N_T - 1$. The convolutions may be performed by computing DFTs of the data blocks, multiplying these by the DFT of the template and computing inverse DFTs of the products. To avoid aliasing the resulting convolutions, the DFTs must be at least $N_B + N_T - 1$ samples long [Oppenheim and Shafer, 1975].

Successive convolved blocks overlap by $N_T - 1$ points, and are summed to form $a[n]$ continuously. In each convolved block, the first and last $N_T - 1$ points are not correct samples of the final convolved stream, but rather partial results that need to be added to the last $N_T - 1$ points of the prior convolved block and the first $N_T - 1$ points of the next convolved block respectively. A conceptually simple approach to implementing this algorithm maintains a shift register that saves the final $N_T - 1$ points of each block convolution and shifts them N_B to the left to be added to the first $N_T - 1$ of the next block convolution, as shown in figure B2.

Appendix C Decimation in Time by Aliasing in the Frequency Domain

The multichannel convolution algorithm depends upon the following simple result, a proof of which is presented here for convenience.

The decimated sequence $x[pN_C]$ $p = 0, 1, \dots, M-1$ can be obtained as the inverse length M Discrete Fourier Transform (DFT) of an aliased version $Y[k]$ of the DFT $X[k]$ of the original length $N = M \cdot N_C$ sequence $x[n]$.

$$Y[k] = \sum_{r=0}^{N_C-1} X[rM+k] \quad (\text{C.1})$$

In the application of this result to the multichannel convolution algorithm $M = N_T + N_B - 1$.

Proof

The Discrete Fourier Transform (DFT) $X[k]$ of length N of a sequence $x[n]$ is defined by:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{i2\pi kn}{N}} \quad (\text{C.2})$$

and the corresponding inverse DFT by:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{i2\pi kn}{N}} \quad (\text{C.3})$$

The decimated sequence can be represented in terms of $X[k]$ by substituting $n = pN_C$ into equation B.3:

$$x[pN_C] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{i2\pi kp}{M}} \quad p = 0, 1, \dots, M-1 \quad (\text{C.4})$$

Define $k = rM + q$ where $r = 0, 1, \dots, N_C-1$ and $q = 0, 1, \dots, M-1$. Then:

$$x[pN_C] = \frac{1}{N} \sum_{q=0}^{M-1} \sum_{r=0}^{N_C-1} X[rM+q] e^{\frac{i2\pi(rM+q)p}{M}} \quad p = 0, 1, \dots, M-1 \quad (\text{C.5})$$

Since $e^{\frac{i2\pi rMp}{M}} = e^{i2\pi rp} = 1$ with r and p both integers,

$$x[pN_C] = \frac{1}{N} \sum_{q=0}^{M-1} \sum_{r=0}^{N_C-1} X[rM+q] e^{\frac{i2\pi qp}{M}} \quad p = 0, 1, \dots, M-1 \quad (\text{C.6})$$

Simplifying:

$$x[pN_C] = \frac{1}{N} \sum_{q=0}^{M-1} Y[q] e^{\frac{i2\pi qp}{M}} \quad p = 0, 1, \dots, M-1 \quad (\text{C.7})$$

where

$$Y[q] = \sum_{r=0}^{N_C-1} X[rM+q]$$

which is the desired result.

Appendix D Equivalence of Correlation Calculations Under Bandpass Transformation

The key calculations in correlation and subspace detectors involve inner products between templates and data in either unmultiplexed (equation 13) or multiplexed (equation 29) versions of the algorithms. These inner products are normalized by calculations of the signal energy in the detection window, which themselves are inner products of the data with itself. In this appendix we establish the equivalence the inner products on the original templates and data and their complex analytic counterparts obtained from the bandpass transformation. This assertion follows from a relation between inner products of two signals in the time and frequency domains:

$$\sum_n u'[n]x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} U'(\Omega)X(\Omega)d\Omega \quad (\text{D.1})$$

Here, the symbol $'$ refers to the conjugate operation. In this relation, the frequency domain quantities are the Discrete-Time Fourier Transform (DTFT) of the respective signals. The DTFT is defined by the pair:

$$X(\Omega) = \sum_n x[n]e^{-i\Omega n} \quad (\text{D.2})$$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\Omega)e^{i\Omega n} d\Omega$$

A real signal has a conjugate-symmetric DTFT:

$$X(\Omega) = X'(-\Omega) \quad (\text{D.3})$$

Consequently, a real bandpass signal centered around frequency Ω_c can be written as a sum of two sidebands (see Figure 7):

$$X(\Omega) = X_b(\Omega - \Omega_c) + X_b'(-\Omega - \Omega_c) \quad (\text{D.4})$$

related to a complex baseband signal $X_b(\Omega)$ centered on zero frequency.

Since the sidebands are disjoint, the inner product D.1 of two bandpass signals occupying the same bands can be written as:

$$\begin{aligned} \frac{1}{2\pi} \int_{-\pi}^{\pi} U'(\Omega)X(\Omega)d\Omega &= \frac{1}{2\pi} \int_{-\pi}^{\pi} U_b'(\Omega - \Omega_c)X_b(\Omega - \Omega_c)d\Omega + \\ &\quad \frac{1}{2\pi} \int_{-\pi}^{\pi} U_b(-\Omega - \Omega_c)X_b'(-\Omega - \Omega_c)d\Omega \end{aligned} \quad (D.5)$$

With the change of variables $\Psi = \Omega - \Omega_c$ in the first integral on the right and $\Psi = -\Omega - \Omega_c$ in the second integral,

$$\begin{aligned} \frac{1}{2\pi} \int_{-\pi}^{\pi} U'(\Omega)X(\Omega)d\Omega &= \frac{1}{2\pi} \int_{-\pi}^{\pi} U_b'(\Psi)X_b(\Psi)d\Psi + \frac{1}{2\pi} \int_{-\pi}^{\pi} U_b(\Psi)X_b'(\Psi)d\Psi \\ &= 2Re \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} U_b'(\Psi)X_b(\Psi)d\Psi \right\} \end{aligned} \quad (D.6)$$

By applying the bandpass transformation to the original real template $u[n]$ and data $x[n]$, we are constructing complex baseband signals $u_b[n]$ (from the template) and $x_b[n]$ (from the data stream) that correspond to $U_b(\Omega)$ and $X_b(\Omega)$: respectively

$$\begin{aligned} u_b[n] &= H_{lp}\{e^{i\Omega_c n} u[n]\} \\ x_b[n] &= H_{lp}\{e^{i\Omega_c n} x[n]\} \end{aligned} \quad (D.7)$$

Here, $H_{lp}\{\}$ refers to the lowpass filtering operation used to eliminate the negative-frequency sideband of the original signals (see Figure 7). The complex versions of the detection algorithms compute the inner products

$$\sum_n u_b'[n]x_b[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} U_b'(\Omega)X_b(\Omega)d\Omega \quad (D.8)$$

From equation (D.6) it is apparent that the inner products are equivalent:

$$2Re \left\{ \sum_n u_b'[n]x_b[n] \right\} = \sum_n u[n]x[n] \quad (D.9)$$