LLNL-TR-463535

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Computational Complexity of Subspace Detectors and Matched Field Processing

D. B. Harris

December 2, 2010

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Computational Complexity of Subspace Detectors and Matched Field Processing

D. B. Harris

Lawrence Livermore National Laboratory

Subspace Detectors

Subspace detectors implement a correlation-type calculation on a continuous (network or array) data stream [Harris, 2006]. The difference between subspace detectors and correlators is that the former projects the data in a sliding observation window onto a basis of template waveforms that may have a dimension (d) greater than one, and the latter projects the data onto a single waveform template. A standard correlation detector can be considered to be a degenerate (d=1) form of a subspace detector. Figure 1 below shows a block diagram for the standard formulation of a subspace detector. The detector consists of multiple multichannel correlators operating on a continuous data stream. The correlation operations are performed with FFTs in an overlap-add approach that allows the stream to be processed in uniform, consecutive, contiguous blocks.

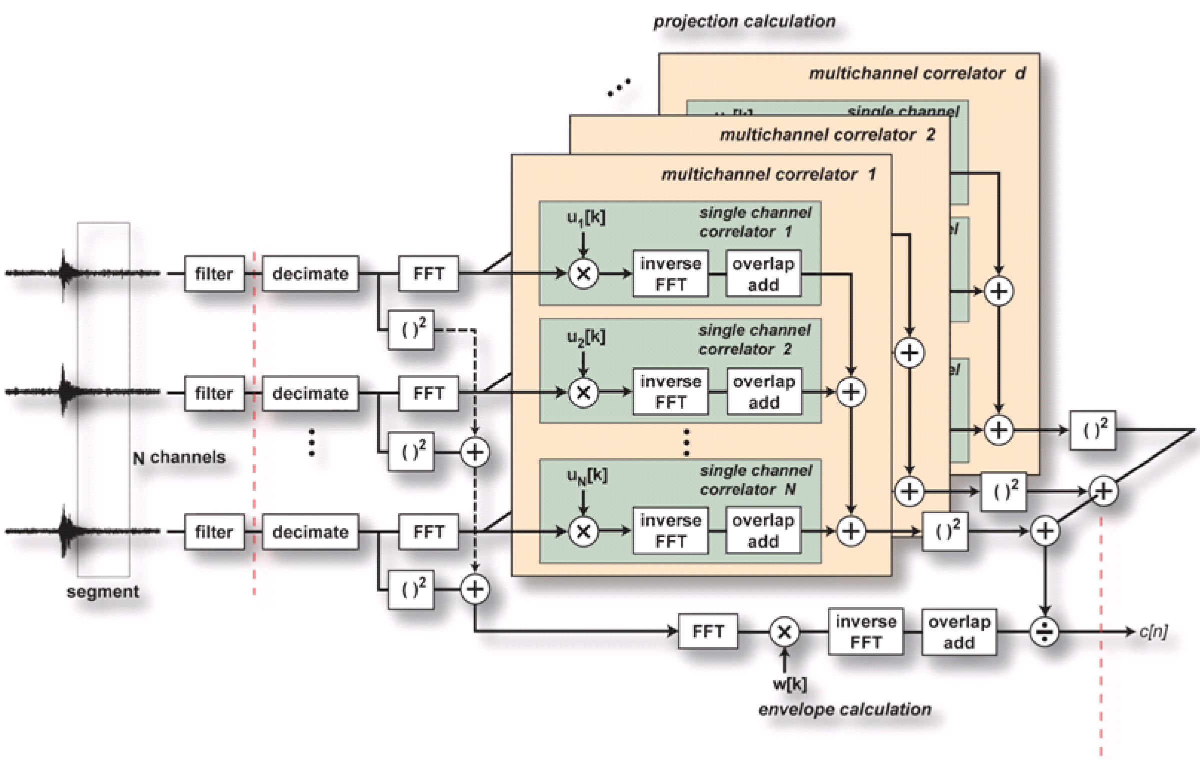

Figure 1 Subspace Detector Schematic

Figure 1 is slightly misleading for a calculation of computational complexity, as it is possible, when treating all channels with the same weighting (as shown in the figure), to perform the indicated summations in the multichannel correlators before the inverse FFTs and to get by with a single inverse FFT and overlap add calculation per multichannel correlator. In what follows, we make this simplification.

Note that decimation may be performed to reduce the number of calculations if the data are substantially oversampled.

Subspace detectors calculate the running detection statistic $c[n]$:

$$c[n] = \frac{\sum_{d=1}^{N_d}\left(\sum_{i=1}^{N_c}\sum_{k=1}^{N_T} u_i^d[k]\, x_i\,[n-k]\,\right)^2}{\sum_{k=1}^{N_T}\, w[k]\,\left(\sum_{i=1}^{N_c} x_i^2\,[n-k]\right)} \tag{1}$$

with $N_c \cdot N_d$ convolutions of single-channel data streams with length N_T correlation templates, normalized by an STA-like calculation of running energy in a window also implemented with a convolution operation (denominator above). The entities and parameters of the calculation are defined in Table I.

Table I: Nomenclature for Subspace Detector complexity calculation

Entity	Definition
$\boldsymbol{x}_i[n]$	Data stream from single channel of observing network or array
$\boldsymbol{N_c}$	Number of observing channels
$\boldsymbol{u_i^d}[n]$	Subspace template basis function, channel i, dimension d
$\boldsymbol{N_d}$	Number of subspace dimensions
$\boldsymbol{N}_T$	Duration of the templates in samples
R	Decimation rate
$w[n]$	Window of length N_T/R consisting of all ones – used to sum energy
P	Number of poles in anti-alias/bandpass filter

The convolutions are implemented with an overlap-add algorithm, using DFT calculations. For this purpose, the data streams are broken into consecutive, contiguous blocks of length N_B samples. The data streams may be filtered by a bandpass or lowpass filter intended to select the processing frequency band, then decimated by a factor of $\boldsymbol{R}$, which reduces the block length to ${}^{N_B}/_{R}$ (for no decimation, $\boldsymbol{R} = 1$), and, correspondingly, the waveform template length from N_T samples to N_T/R samples. In each block, each convolution is implemented with length-N DFTs, where N is the least power of two greater than or equal to $N_B/R + N_T/R - 1$. The number of computation components per block is summarized in Table II.

Table II: Number of computation components per data block.

Computation Component	Number per Block
Anti-alias/bandpass IIR filtering operations	N_c
Forward DFTs	$N_c + 1$
Block-Template DFT products	$N_c \cdot N_d$
Stack of DFT products	N_d
Block Power Calculations (squaring signals)	$N_c + N_d$
Block Power Stack over dimensions	1
Block Power Stack over channels	1
Inverse DFTs	$N_d + 1$
Overlap-Add block sums	2
Block Ratio	1

This count of computation components assumes that the DFTs of the subspace templates $u_c^d[\boldsymbol{n}]$ and the running-energy calculation kernel $\boldsymbol{w}[\boldsymbol{n}]$ are pre-computed and stored.

The numbers of multiplications and additions per block required for each computation component are listed in Table III. A *P*-pole bandpass pre-processing filter has been assumed.

Table III: Number of operations per block for each computation component

Computation Component	Real Multiplications	Real Additions
Anti-alias/bandpass IIR filter	$(2P+1)N_B$	$2PN_B$
DFT	$\frac{1}{2}Nlog_2N - \frac{3}{2}N + 2$	$\frac{3}{2}Nlog_2N - \frac{5}{2}N + 4$
Block-Template DFT product	$\frac{3N}{2} - 1$	$3\left(\frac{N}{2} - 1\right)$
Stack of DFT products	0	$(N_c - 1)N$
Block Power Calculation (squaring signals)	$\frac{N_B}{R}$	0
Block Power Stack over dimensions	0	$\frac{N_B}{R}(N_d - 1)$
Block Power Stack over channels	0	$\frac{N_B}{R}(N_c - 1)$
Inverse DFT	$\frac{1}{2}Nlog_2N - \frac{1}{2}N + 2$	$\frac{3}{2}Nlog_2N - \frac{5}{2}N + 4$
Overlap-Add block sum	0	N
Block Ratio	$\frac{N_B}{R}$	0

Here we have assumed that real split-radix FFTs are used to implement the DFTs [Sorensen et al., 1987]. We caution that FFT execution times may not be dominated by multiplications and additions, but are substantially affected by indexing calculations and data moves. The ultimate performance of FFT algorithms is highly dependent upon implementation. Here we have also assumed that sine and cosine tables for the FFT algorithms have been pre-computed and stored (a standard practice). In addition, we have assumed that complex multiplications are implemented with 3 real multiplications and three real additions (alternatively they may be implemented with 4 real multiplications and two real additions). We also are assuming that subtractions can be counted as additions and divisions as multiplications.

The total number of operations required per block of data is obtained by combining the elements of tables II and III. The total number of multiplications per block is:

$$(4P+1)N_B N_c \;+\; (N_c+N_d+2)\left(\frac{1}{2}N log_2 N - \frac{3}{2}N + 2\right) \;+\; (N_d+1)N \;+\; N_c N_d\left(\frac{3N}{2} - 1\right) \;+\; (N_c+\;N_d+1)\frac{N_B}{R} \tag{2a}$$

and the total number of additions per block is:

$$4PN_B N_c \;+\; (N_c+N_d+\;2)\left(\frac{3}{2}N log_2 N - \frac{5}{2}N + 4\right) \;+\; \frac{N_B}{R}\cdot(N_d+N_c-2) \;+\; N_c\cdot N_d\left(\frac{5}{2}N - 3\right) \;-\; (N_d-2)\cdot N \tag{2b}$$

For a representative example, we use values for a 9-channel network or array, sampled at 100 samples per second, with a template 10 seconds (1000 samples) long and a processing block size of 150 seconds (15000 samples). We assume that the processing band is 5-20 Hz, which permits a decimation rate of 2.

Table IV: Operation count for an example problem

Quantity	Value
$\boldsymbol{N_c}$	9
$\boldsymbol{N_d}$	3
$\boldsymbol{N}_T$	1,000
$\boldsymbol{N}_B$	15,000
R	2
N	8,192
P	10
Total # multiplies	3,870,485
Total # adds	5,269,439
#multiples/sample	258 (rounded)
#adds/sample	351 (rounded)

Matched Field Processing

Matched Field Processing has been developed for seismic verification problems as a classification technique, but also may be adapted as a calibrated beamforming technique for use in a detector [Harris and Kvaerna, 2010]. The application considered here is event classification for which the technique is well developed. An estimate of operation counts for a detector would be premature, since the form of a detector is still a subject of research. The wideband form of the classification algorithm operates on a collection of spectral covariance matrices estimated from a data window in a manner directly comparable to FK methods. The basic calculation consists of a ratio of sums of quadratic forms:

$$c = \frac{\sum_{k=k_{min}}^{k_{max}} \boldsymbol{e}_k^H \boldsymbol{C}_k \boldsymbol{e}_k}{\sum_{k=k_{min}}^{k_{max}} tr\{\boldsymbol{R}_k\}} \qquad (3)$$

The vectors $\boldsymbol{e}_k$, called steering vectors, have N_c elements and are obtained as the principal eigenvectors of sample covariance matrices averaged over a number of data windows recording calibration events. There are N_H collections of steering vectors, corresponding to the number of classification hypotheses. The covariance matrices $\boldsymbol{C}_k$ are $N_c \times N_c$ matrices formed from waveforms in the data window filtered into extremely narrow frequency bands. Each quadratic form $\boldsymbol{e}_k^H \boldsymbol{C}_k \boldsymbol{e}_k$ represents an estimate of the power in an observed array or network waveform in a particular frequency band k delivered by a signal originating at a particular target location. The total number of bands is $N_b = k_{max} - k_{min} + 1$.

If we treat the collection of waveforms recorded by a network or an array as a vector $\boldsymbol{x}[n]$, then the signal filtered into narrow frequency band k is

$$\boldsymbol{x}_k[n] = \sum_{\boldsymbol{r}} \boldsymbol{x}[r]\ h_k[n-r] \tag{4}$$

A particularly efficient means (called a phase vocoder, see [Portnoff, 1980]) of calculating signals filtered into many regular, narrow frequency bands is available if the filters are related systematically to a prototype baseband filter:

$$h_k[n] = h_0[n]e^{i2\pi kn/M} \tag{5}$$

where M is a number of frequency bands. The $N_b \leq M$ bands used to calculate the matched field statistic in equation (3), are a subset of the bands constructed in equations (4) and (5).

The covariance matrices of equation (3) are calculated not from the $\boldsymbol{x}_k[n]$ directly, but from closely related quantities known as complex envelopes. The complex envelopes are obtained by substituting (5) into (4):

$$\boldsymbol{x}_k[n] = e^{i2\pi kn/M} \sum_{r} (\boldsymbol{x}[r]\ h_0[n-r])\ e^{-i2\pi kr/M} \tag{6}$$

The complex envelope functions $\tilde{\boldsymbol{x}}_k[n]$ are defined as:

$$\tilde{\boldsymbol{x}}_k[n] = \sum_{r} (\boldsymbol{x}[r]\ h_0[n-r])\ e^{-i2\pi kr/M} \tag{7}$$

we see that $\boldsymbol{x}_k[n] \;=\; e^{i2\pi kn/M}\,\tilde{\boldsymbol{x}}_k[n]$.

The complex envelope functions can be evaluated efficiently with a length-M FFT. Evaluation consists of a windowing operation $\boldsymbol{x}[r]\,h_0[n-r]$, followed by an aliasing operation [Portnoff, 1980] to reduce the length of this product to M samples. It is common to design the baseband filter impulse response $h_0[n]$ to be symmetric and of length $2Mf+1$, where f is an integer design factor typically 2 or 3. The length of the product is reduced to M samples via a $2f$-fold aliasing operation.

The covariance matrices of equation (3) are calculated from samples of the complex envelopes as follows:

$$\boldsymbol{C}_k \;=\; \frac{1}{N_s}\sum_n \tilde{\boldsymbol{x}}_k[n](\tilde{\boldsymbol{x}}_k[n])^H \qquad (8)$$

where N_s is the number of time samples used to form the estimate. The time samples need not be consecutive. In fact, because the complex envelope functions are so narrowband, it is possible to decimate them heavily in forming the covariance matrix estimate.

The relevant parameters controlling the computational complexity of the matched field calculation are summarized in Table V:

Table V: Parameters controlling Matched Field computation complexity

Parameter	Definition
P	Number of poles in preprocessing (highpass) filter
$2L$	Data window size
$\boldsymbol{N_c}$	Number of observing channels
$\boldsymbol{N_b}$	Number of frequency bands for MF statistic evaluation
M	FFT size
f	Baseband filter design factor
$\boldsymbol{N_s}$	Number of complex envelope samples for covariance matrix
R	Decimation rate
$\boldsymbol{N_H}$	Number of hypotheses to test

A certain data window size is required for the calculation of complex envelope samples. The minimum size is: $L \;=\; (N_s-1)\cdot R + 2Mf + 1$, but a longer window is desirable to eliminate edge effects in prefiltering operations. In the operation counts to follow, we will assume that twice this amount of data is used. The mean of these data will be removed and the data will be filtered with a 4-pole IIR highpass filter.

The number of computation components in the Matched Field calculation is summarized in Table VI.

Table VI: Number of computation components in Matched Field calculation.

Computation Component	Number
Preprocessing mean removal and IIR filtering operations	N_c
Window/Alias operations	$N_s \cdot N_c$
Real DFTs	$N_s \cdot N_c$
Covariance matrix calculations	N_b
Trace calculations	N_b
Quadratic form evaluations	$N_b \cdot N_H$

The number of operations required for each component of computation is summarized in Table VII.

Table VII: Number of operations for each computation component

Computation Component	Real Multiplications	Real Additions
Preprocessing demean & IIR filtering operation	$2L(2P+1)+1$	$2L(P+1)-1$
Window/Alias operation	$2Mf+1$	$2Mf$
Real DFT	$\frac{1}{2}Mlog_2M-\frac{3}{2}M+2$	$\frac{3}{2}Mlog_2M-\frac{5}{2}M+4$
Covariance matrix calculation	$N_c^2(3N_s+1)$	$N_c^2(5N_s-1)$
Trace calculation	0	N_c
Quadratic form evaluation	$3N_c(N_c+1)$	$5N_c^2+N_c-2$

Combining (multiplying) the counts of tables VI and VII, the total number of multiplications is:

$$(2L(2P+1)+1)N_c \; + \; (2Mf+1)N_sN_c \; + \; \left(\frac{1}{2}Mlog_2M-\frac{3}{2}M+2\right)N_sN_c \\ + \; N_c^2(3N_s+1)N_b \; + \; 3N_c(N_c+1)N_b \cdot N_H \; + \; N_H \qquad \text{(9a)}$$

and the total number of additions is:

$(2L(P+1)-1)N_c \; + \; 2MfN_sN_c \; + \; \left(\frac{3}{2}Mlog_2M - \frac{5}{2}M + 4\right)N_sN_c$ $+ \; N_c^2(5N_s-1)N_b \; + \; N_cN_b \; + \; (5N_c^2 + N_c\,)N_bN_H \; - \; 2N_H$	(9b)

Recall that $L \; = \; (N_s - 1)\cdot R + 2Mf + 1$.

Table VIII summarizes the number of calculations for an example problem similar in size to that described in [Harris and Kvaerna, 2010].

Table VIII: Operation count for an example problem

Quantity	Value
P	4
L	913
$\boldsymbol{N_c}$	17
$\boldsymbol{N_b}$	33
M	128
f	3
$\boldsymbol{N_s}$	10
R	16
$\boldsymbol{N_H}$	10
Total # multiplies	1,052,582
Total # adds	1,410,827
#multiples/sample(2L)	576 (rounded)
#adds/sample(2L)	773 (rounded)

References

Harris, D. (2006), Subspace detectors: Theory, Lawrence Livermore Natl. Lab. Rep. UCRL-TR-222758, 46 pp., Lawrence Livermore Natl. Lab., Livermore, Calif. (Available at http://www.llnl.gov/tid/lof/documents/pdf/335299.pdf)

Harris, D. B. and T. Kvaerna (2010), Superresolution with seismic arrays using empirical matched field processing, Geophysical Journal International, **182**, 1455-1477.

Portnoff, M. R. (1980), Time-frequency representation of digital signals and systems based on short-time Fourier analysis, IEEE Trans. Acoust. Speech and Sig. Proc., **ASSP-28**(1), 55-69.

Sorensen, H. V., M. T. Heideman, and C. S. Burrus (1986), On Computing the Split-Radix FFT, IEEE Trans. On Acoustics, Speech and Signal Processing, Vol. ASSP-34, No. 1, pp. 152-156.

Sorensen, H. V., D. L. Jones, M. T. Heideman, and C. S. Burrus (1987), Real-Valued Fast Fourier Transform Algorithms, IEEE Trans. On Acoustics, Speech and Signal Processing, Vol. ASSP-35, No. 6, pp. 849-863.