LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T060221-00-Z**	2007/06/28

KleineWelle Technical Document

L. Blackburn[†]
[†]*Massachusetts Institute of Technology*

Draft

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW17-161
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

WWW: http://www.ligo.caltech.edu/

CVS Id: kwdoc.tex,v 1.1 2007/06/28 00:00:00 lindy Exp — Processed with LaTeX on 2007/06/28

Contents

1 Introduction

KleineWelle is a multiresolution method which attempts to find and characterize transients in an input timeseries. The method may be applied to the interferometric gravitational wave channel (LSC-DARM_ERR) or to other interferometer and environmental channels to search for statistical artifacts. KleineWelle makes use of the dyadic wavelet transform [2] to search for regions of excess energy in the time-scale decomposition. The strength of this approach lies in the computational efficiency of the transform, as well as its multiresolution properties.

2 Algorithm

2.1 Wavelet Transform

The Wavelet transform[1, 2] for timeseries $f(t)$ is defined by the integral,

$$W_f(u,s) = \int_{-\infty}^{+\infty} f(t)\,\frac{1}{\sqrt{s}}\,\psi^{\star}\left(\frac{t-u}{s}\right)\,dt, \tag{1}$$

where the wavelet, ψ, is a time-localized function of zero average.

The coefficients $W_f(u,s)$ are evaluated continuously over times, u, and scales, s. Our ability to resolve in time and frequency is then determined by the properties ψ assumes at each scale. At large scale, ψ is highly dilated yielding improved frequency resolution at the expense of time resolution. At small scale, we achieve good time resolution with large uncertainty in frequency.

For the case of discrete data, a computationally efficient algorithm exists for calculating wavelet coefficients over scales that vary as powers of two: $s \in \{2^{j-1} \,|\, j \in \mathbb{Z}^{+}\}$. This is the dyadic wavelet transform, which can be implemented for a limited family of wavelets using conjugate mirror filters. The filters consist of a high pass filter, $\hat{H}$, and low pass filter, $\hat{L}$, which can be applied in a cascade to obtain the wavelet coefficients. Beginning with the original time series, A_0, of length N, two sequences of length $N/2$ are obtained by application of the high pass and low pass filters followed by down-sampling. The sequence of detail coefficients, D_j, and approximation coefficients, A_j, are defined at each level, j, of the decomposition by

$$\begin{aligned} D_j &= \hat{H}(A_{j-1}) \quad \text{and} \\ A_j &= \hat{L}(A_{j-1}). \end{aligned} \tag{2}$$

The detail coefficients for scale s, where $s = 2^{j-1}$, calculated in this manner are the same as the wavelet coefficients obtained from Equation 1. If N is a power of two, so that $N = 2^m$, the final approximation sequence will be A_m.

The simplest dyadic wavelet is the Haar function:

$$\psi^{\mathrm{Haar}}(t) = \begin{cases} 1 & 0 \le t < 1/2 \\ -1 & 1/2 \le t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The corresponding high pass and low pass filters are

$$\begin{aligned} \hat{H}^{\mathrm{Haar}} &= [+1, -1]/\sqrt{2} \quad \text{and} \\ \hat{L}^{\mathrm{Haar}} &= [+1, +1]/\sqrt{2}, \end{aligned} \tag{4}$$

from which we see that the detail coefficients are related to the differences of each pair of points in the parent series, while the approximation coefficients are related to the averages of each pair.

2.2 Data Conditioning and Linear Predictive Filtering

KleineWelle supports the use of linear predictive filtering to whiten the data prior to the wavelet decomposition. This simplifies the statistics of the analysis, and removes the otherwise dominating effects of strong lines and other coherent noise sources from the calculation [3]. A user enables linear predictive whitening by setting the *filter* option to *lpef*. A Butterworth high-pass prefilter is automatically applied and the linear preditive filter order is set automatically depending on the frequency range of the search.

2.3 Event Selection

As the whitened data are passed to the dyadic wavelet transform, for a sufficiently large scale, j, the wavelet coefficients, D_j, within the scale will approach a zero-mean Gaussian distribution with standard deviation σ_j by the central limit theorem. We therefore define the sequence of squared normalized coefficients, or normalized pixel energies at scale 2^j,

$$E_j = D_j^2/\sigma_j^2, \tag{5}$$

which is chi-squared distributed with one degree of freedom

It is then simply a matter of thresholding on the energy of individual pixels, $\epsilon_{ij} \in E_j$, to identify statistical outliers. We may also choose to cluster nearby pixels to better detect bursts which deviate from the dyadic wavelet tiling of the time-frequency plane. For a randomly selected cluster C of N pixels, we define the total normalized cluster energy,

$$E_C = \sum_{(i,j)\in C} \epsilon_{ij}, \tag{6}$$

which is also chi-square distributed, but with N degrees of freedom.

The *significance* of this cluster of pixels is defined by,

$$S = -\ln \int_{E_C}^{\infty} \chi^2_N(E) dE, \tag{7}$$

So that the significance is a function of cluster energy E_C and the number of pixels in the cluster N. It is therefore possible to select a threshold significance to achieve a desired white noise false event rate.

3 DMT Implementation

KleineWelle currently runs as a DMT Monitor [4]. This allows for the possibility of running online using streaming realtime data, however the current focus of the pipeline is offline analysis. The most current version uses the DMT multistream offline build which does not support online data access but allows for two input file streams to be processed at the same time (useful for adding software injections).

3.1 Dependencies

KleineWelle depends on a DMT implementation of linear filtering, LPEFilter, by S. Chatterji, and a dyadic wavelet library, mywv2lib, by E. Katsavounidis and G. Martin. LPEFiler itself inherets from the FIRFilter class in DMT, and is used in the same way. The wavelet library provides a simple interface via arrays of floats to both set the input series, and to get the decomposition coefficients.

3.2 Data Access

KleineWelle uses DMT's DatEnv for data access. In this framework, a certain amount of data called the *stride* is available for processing at any one time. The data processing is done in a loop ProcessData() which obtains the next continuous stride of data each time it is called. Since there is no guarantee that the data are continuous from call to call, some amount of data handling is required in kleineWelle to both account for filter transients and maintain a power-of-two length of data for the dyadic wavelet transform. When kleineWelle has detected a discontinuity in the current stride of data from the last block, it temporarily sets the stride to one second and loops once more to include *stride*+4 seconds of data before filtering. The first four seconds of data are then discarded leaving exactly *stride* seconds of filtered data. The FIRFilter and LPEFilter classes automatically store history to handle continuous data.

KleineWelle can also be passed a segment list of times to be analyzed using the *segmentFile*

parameter. No data will be analyzed which falls outside of these segments. If data is not available for times within a segment, it is skipped and no error is produced.

Besides the four second loss due to filter transients, kleineWelle also loses some amount of data at the end of each analysis segment due to the fact that the amount of continuous data anlayzed must always be a multiple of the stride.

3.3 Simulations

KleineWelle supports loading a separate stream of MDC data sets through DMT's multistream class. While the *channel* parameter defines the gravitational wave channel, the *injections* parameter defines the channel containing the injection timeseries. The injections can be scaled by a multiplicative *factor* which is 1.0 by default. If there is only one data stream, kleineWelle will assume the both the data *channel* and *injections* channel are in the available data stream. If there are two data streams, the injections are read from the second data stream. To use the multistream class to read in two data streams simultaneously, one must first create the two ASCII lists of frame files which define each stream. Next, the user must create a third two-line ASCII file which containes the filenames of these lists. This 'list of lists' is passed to DMT using the "–inlists" option.

3.4 Dyadic Wavelet Transform

The stride of filtered data are passed to the wavelet library for decomposition. KleineWelle currently makes use of the Haar wavelet (equation 3) though other wavelets could be added in the future without major changes to the code. The wavelet library returns an array of floats the same length of the input array that contains all the decomposition coefficients. For an input series length N, the coefficients for scale j will be available in indicies $N/2^j$ to $N/2^{j-1}$ (counting from 1). The standard deviation of coefficients in each scale is stored in a separate array and used for the subsequent statistical analysis.

3.5 Clustering

The wavelet coefficients which pass a certain amplitude threshold derived from the standard deviation of coefficients in a given scale are stored as pixels in a vector of Element structures. Each structure contains information about the pixel's time and scale, as well as its unnormalized and normalized energy (Equation 5).

The pixels are run through a recursive clustering algorithm which takes nearby pixels and clusters them into a single cluster represented by an Event structure. The clustering method depends on a single integer input parameter *maxDist*. The distance between two pixels is

defined as the sum of the difference in scales and the difference in time, with time measured in units of the wider pixel. A pixel's distance with itself is zero, and two pixels with an adjacent side will have distance 1. Two pixels with an adjacent corner will have distance 1 + 1 = 2. When clustering, any two pixels with distance less than or equal to maxDist will be clustered. A pixel can only belong to one cluster, and pixels with distance greater than maxDist will only be clustered if there are intermediate pixels to connect them.

3.6 Event Generation

Each cluster is represented by an Event structure, and contains derived quantities such as the absolute start and end times of the event, the energy weighted central time, the energy weighted central scale, and the cluster significance (Equation 7).

Clusters which pass the significance threshold are output in multicolumn ASCII format to standard output. The current list of columns is,

- start time
- end time
- normalized energy-weighted central time
- normalized energy-weighted central scale translated into an approximate frequency
- summed unnormalized energy
- summed normalized energy
- number of pixels in the cluster
- cluster significance

3.7 Syntax

Options to kleineWelle are passed using an options file. Besides the DatEnv arguments, this is the only argument sent to the program, thus kleineWelle can be run with,

- ./kleineWelle *optionsfile* (online)
- ./kleineWelle *optionsfile* -infile '*gwffiles*' (currently unsupported in multistream DMT)
- ./kleineWelle *optionsfile* -inlist *gwffilelist.txt*
- ./kleineWelle *optionsfile* -inlists *listofgwffilelists.txt*

The current set of available options and their defaults are included and described with the sample options file in the CVS distribution.

References

[1] Daubechies I, *Ten Lectures on Wavelets*, (1992) SIAM.

[2] Mallat S, *A Wavelet Tour of Signal Processing*, (1999) Academic Press.

[3] Chatterji S, Blackburn L, Martin G, Katsavounidis E, *Multiresolution techniques for the detection of gravitational-wave bursts*, Class. Quantum Grav. 2004

[4] http://www.ligo.caltech.edu/ jzweizig/dmt/DMTProject