Coherent Line Removal

—in LAL C code—

Alicia M. Sintes
sintes@aei-potsdam.mpg.de
Max-Planck-Institut für Gravitationsphysik
Albert-Einstein-Institut
Golm, Germany
http://www.aei-potsdam.mpg.de

Hanford LSC meeting. August 15-17, 2000
Coherent Line Removal (CLR)

- CLR is an algorithm able to remove interference present in the data while preserving the stochastic detector noise.
- CLR works when the interference is present in many harmonics, as long as they remain coherent with one another.
- CLR can remove the external interference without removing any ‘single line’ signal buried by the harmonics.
- CLR can be used to remove all harmonics of periodic or broadband signals (e.g., those which change frequency in time).
- CLR requires little a priori knowledge of the signals we want to remove (e.g., no external reference).

The principal of CLR is the following:

We assume that the interference has the form

$$y(t) = \sum_{n} a_n m(t)^n + (a_n m(t)^n)^*$$

where $a_n$ are complex amplitudes and $m(t)$ is a nearly monochromatic function near a frequency $f_0$.

The idea is to use the information in the different harmonics of the interference to construct a function $M(t)$ that is as close a replica as possible of $m(t)$ and then construct a function close to $y(t)$ which is subtracted from the output of the system cancelling the interference.
• CLR can be used to remove lines corresponding to the electricity supply frequency and its harmonics.

Figure 1: Decimal logarithm of the periodogram of $2^{19}$ points of the prototype data. (a) One of the harmonics near 754 Hz. (b) The same data after the removal of the interference using CLR. (c) The same experimental data with an artificial signal added at 452 Hz. (d) The data in (c) after the removal of the interference, revealing that the signal remains detectable.
Figure 2: Comparison of a zoom of the spectrogram. (a) is obtained from the prototype data. We can observe the wandering of the incoming electrical signal. The other two remaining lines at constant frequency correspond to violin modes. (b) The same spectrogram as in (a) after applying CLR, showing how the electrical interference is completely removed.
Package cohlineremoval

This package provides routines for finding line harmonics, generating a reference interference signal, and removing all the interference harmonics.

The package is organized under the header CLR.h and the modules HarmonicFinder.c, RefInterference.c and CleanAll.c.

The code is written following the last LAL specifications (LIGO-T990030-08, from 07.18.2000) and also includes the automatic documentation system.

The code was compiled and tested using a previous version (function names without the prefix LAL) i.e., linked with the last release lal-0.4-beta and with the fftw library.
Structures

struct REAL4TVectorCLR

It contains time domain data and other information. The fields are:

UINT4 length  The number of elements in data.
REAL4 *data   The (real) time domain data.
REAL4 deltaT  The sample spacing in time (in seconds).
REAL4 fLine   The interference fundamental frequency $f_0$ (in Hz), e.g., 60 Hz.

struct REAL4FVectorCLR

It stores the spectrum, $|\tilde{x}(\nu)|^2$, and other information needed to find the location of the line harmonics. The fields are:

UINT4 length  The number of elements in data.
REAL4 *data   The (real) frequency domain data, e.g., $|\tilde{x}(\nu)|^2$.
REAL4 deltaF  The $\Delta F$ offset between samples (in Hz).
REAL4 fLine   The interference fundamental frequency $f_0$ (in Hz), e.g., 60 Hz.
HarmonicFinder.c

void LALHarmonicFinder(
    LALStatus *status,
    INT4Vector *out,
    REAL4FVectorCLR *in2,
    INT4Vector *in1);

Given certain harmonic indices \{k\} finds the frequency interval location (in bins) of the interference (around \(k \times f_0\)) from the power spectrum.

*in1*: The harmonic indices

\[\text{in1->length}\]  Number of harmonics.
\[\text{in1->data}\]  List of harmonics to consider, e.g., \(\{k\} = \{3, 5, 9, 11 \ldots\}\)

*in2*: The power spectrum, \(|\tilde{x}(\nu)|^2\), \(f_0\) and \(\Delta f\)

\[\text{in2->length}\]  The number of elements in in2->data.
\[\text{in2->data}\]  The spectrum, \(|\tilde{x}(\nu)|^2\).
\[\text{in2->deltaF}\]  The \(\Delta F\) offset between samples (in Hz).
\[\text{in2->fLine}\]  The interference fundamental frequency \(f_0\) (in Hz), e.g., 60 Hz.

*out*: a vector whose length is \(\text{out->length} = 3 \times \text{in1->length}\), and contains for each considered harmonic, in the following order, its index \(k\) and the bin location of \(\nu_{ik}\) and \(\nu_{fk}\).

\[\text{out->length}\]  The number of elements in out->data.
\[\text{out->data}\]  \(\{k, \nu_{ik}, \nu_{fk}\}\), e.g., \(\{3, 9868, 9894, 5, 16449, 16487, 9, 29607, 29675. \ldots\}\).
RefInterference.c

```c
void LALRefInterference(
    LALStatus  *status,
    COMPLEX8Vector  *out,
    COMPLEX8Vector  *in1,
    INT4Vector      *par );
```

Generates a reference interference signal.

Given the complex vector `*in1` of length \(n/2 + 1\), containing the Fourier transform of the data \(\tilde{x}(\nu)\),

\[
\text{in1->length} \quad \text{The number of elements in in1->data} = n/2 + 1. \\
\text{in1->data} \quad \text{The data} \ \tilde{x}(\nu),
\]

and given another vector `*par` containing the information related to the harmonics from which we want to construct the reference signal (i.e., indices, and initial and final frequency bin locations),

\[
\text{par->length} \quad \text{The number of elements in par->data.} \ \text{This is equal to} \\
\quad \text{three times the number of harmonics that will be used to construct the} \\
\quad \text{reference signal} \ M(t). \\
\text{par->data} \quad \{k, \nu_{ik}, \nu_{fk}\}, \ e.g., \ \{3, 9868, 9894, 5, 16449, 16487, 9, \\
\quad 29607, 29675. \ldots \},
\]

it generates the time domain \(M(t)\) reference interference signal, `*out`. This is a complex vector of length \(n\).

\[
\text{out->length} \quad \text{The number of elements in out->data} = n. \\
\text{out->data} \quad M(t) \text{ complex data.}
\]

\(M(t)\) corresponds to a nearly monochromatic function near the frequency \(f_0\), that is implicit in the information given in `*par`. 
CleanAll.c

void LALCleanAll (  
    LALStatus        *status,  
    REAL4Vector      *out,  
    COMPLEX8Vector   *in2,  
    REAL4TVectorCLR  *in1 );

Gets data cleaned from line harmonic interference given a time  
domain reference signal.

*in1*: the time domain data of type REAL4TVectorCLR, containing \( f_0 \) and the sampling spacing.

\[ \text{in1->length} \quad \text{The number of elements in in1->data} = n. \]
\[ \text{in1->data} \quad \text{The (real) time domain data, } x(t). \]
\[ \text{in1->deltaT} \quad \text{The sample spacing in seconds.} \]
\[ \text{in1->fLine} \quad \text{The interference fundamental frequency } f_0, \text{ e.g., } 60 \text{ Hz.} \]

*in2*: the time domain reference signal (a complex vector).

\[ \text{in2->length} \quad \text{The number of elements in in2->data} = n. \]
\[ \text{in2->data} \quad \text{The } M(t) \text{ complex data.} \]

*out*: a real vector containing the clean data.

\[ \text{out->length} \quad \text{The number of elements in out->data} = n. \]
\[ \text{out->data} \quad \text{The clean (real) time domain data.} \]