

# PlaneMon: Airplane Detection Monitor

Evan Goetz and Keith Riles

## 1 Introduction

The Laser Interferometer Gravitational-wave Observatory (LIGO) is a ground-based gravitational-wave detector. Since it is built on earth, LIGO is sensitive to numerous disturbances, from both environmental and anthropogenic sources. Airplanes are an example of anthropogenic noise that has been found in the gravitational-wave data generated by LIGO.

During the S2 science run, the burst analysis group found a single candidate event that survived all of the cuts made by the analysis. This event was later identified as elevated acoustic levels at the LIGO–Hanford site corresponding to an airplane flying nearby. The elevated acoustic levels occurred in coincidence with an unknown event at the LIGO–Livingston site. This event was then cut due to the elevated acoustic levels at Hanford.

It is clear, then, that the LIGO instruments can be sensitive to elevated acoustic noise from airplanes. It is necessary to flag data that could be contaminated by airplane acoustic noise. In addition to flagging data, properties of the airplane’s trajectory can be determined from the acoustic signals. We have developed a Data Monitor Tool (DMT) program to analyze microphones that are placed throughout the buildings of the LIGO sites.

In Section 2 of the paper, we discuss what airplane signals look like and how we go about analyzing these signals. Section 3 of the paper discusses the methods for collecting and analyzing this data through the DMT framework. Then, in Section 4, we discuss how to fit the data to find the airplane parameters, and in Section 5 we discuss how to run and use PlaneMon.

## 2 Airplane Signals

Suppose we have an airplane traveling past a detector (e.g. a microphone) as Figure 1 indicates, with  $x$  as the horizontal distance away from the distance of closest approach,  $s$  as the distance of closest approach,  $d$  as the line of sight distance from the detector to the airplane and  $v_p$  as the speed of the airplane. Note in this convention that  $x$  is initially large and positive, decreasing to large negative values as  $t \rightarrow \infty$ .

The equation for a doppler shifted signal from a sub-sonic source is given as

$$f' = f \frac{v_0}{v_0 - v_p \cos \theta} \quad (1)$$

where  $f'$  is the detected frequency,  $f$  is the frequency of the source in its rest frame,  $v_0$  is the speed of sound and  $\theta$  is the angle between the airplane velocity and the line of sight to the detector.

In order to visualize the connection between airplane trajectory and signal we will see at a detector, let us rewrite Equation (1) as,

$$f' = f \frac{v_0}{v_0 - v_p \cos(\tan^{-1}(s/x))} \quad (2)$$

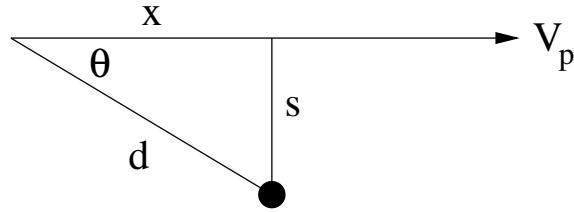


Figure 1: Airplane passing a detector.

Now, given a trajectory through 3-dimensional space, we can deduce the airplane signal. Here, we have used Matlab to plot these trajectories and theoretical signals. We have assumed constant velocity ( $v_p = 300$  mi/hr), constant altitude ( $z = 1000$  m), no acceleration ( $\dot{v}_p = 0$  mi/hr<sup>2</sup>) and a trajectory parallel to the x-y plane (taking each microphone to have zero altitude). This plot includes the time delay for the finite speed of sound, and for each plot, the plane travels from the lower left to the upper right.

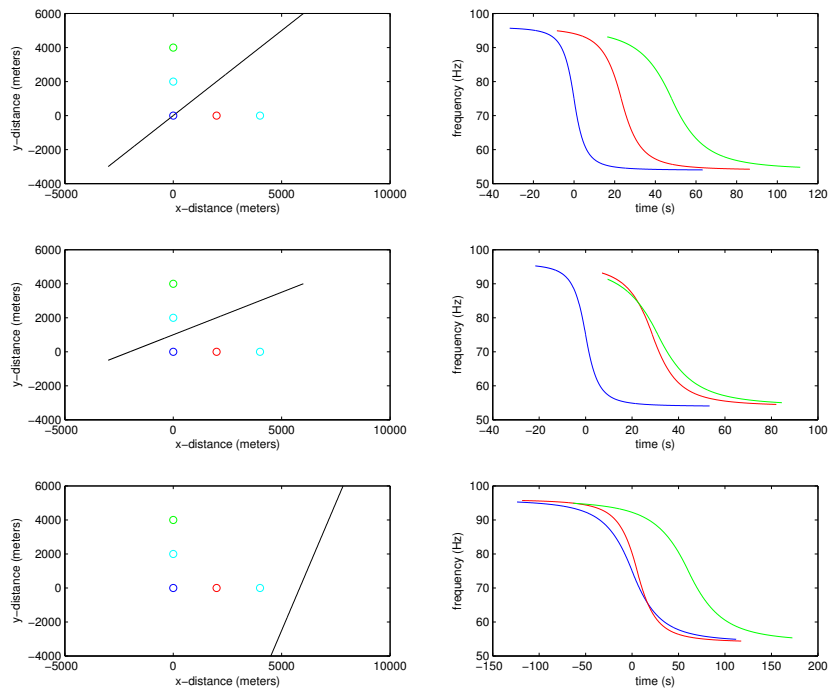


Figure 2: Airplane trajectories and the associated signals a microphone would detect. Each airplane path is from the lower left to the upper right.

### 3 Methods

Data used in this analysis is collected from microphone channels placed near vacuum chambers in the various buildings of the LIGO sites. The data that comes through the data acquisition system comes in the form of a time series. We give this frame data to the DMT program, which we call PlaneMon, to analyze.

The monitor takes this time series and Hanning windows the data with 50 percent overlap. The Fast Fourier Transform (FFT) is then performed on the windowed data. The power in 1 Hz bins is then calculated. When PlaneMon is running nominally, the data is whitened by dividing the current data by an exponentially averaged power spectral density estimate with a time constant of 2 minutes. When PlaneMon is started, there is a 2 minute period in which PlaneMon “learns” the shape of the data. Here, we assume there are no airplanes flying overhead. If an airplane is flying during this period, it may not be flagged by PlaneMon.

Data is collected and held for 2 minutes. We have chosen 2 minutes as the nominal length of time to hold data to be sure that the airplane signal has left the data stream. Once the data has been whitened, any bins that are above a set threshold are marked as potential “bumps”. The cluster of adjacent pixels with the largest amount of power above the threshold is selected as the airplane nucleus. The weighted central frequency and standard deviation for this cluster is then computed. We then plot these data points on a time-frequency plot.

When enough data points have been collected in a set period of time, PlaneMon is triggered the channel, and data from that channel is then saved for 2 minutes. Not all data points collected will be airplane data points. To eliminate excess data points, we perform cuts on the data. If we plot the data on a time versus inverse-frequency plot, we require the slope to be positive between all data points. Any data point that leads to negative slope is cut.

Next, if there are still enough data points, we use the ROOT program to fit the data to a fifth-order polynomial. We choose to fit to a polynomial to ensure convergence. This polynomial is then forced to follow an odd function, as discussed in Section 4. For any single data point, we require the  $\chi^2$  to be less than a set threshold. Any data point that has a  $\chi^2$  that is larger than the threshold is pruned. We can either re-fit the data at this point or not. We save computational cycles if we choose to ignore the re-fitting. This procedure is repeated until we can no longer continue pruning data points.

Finally, if there are still enough surviving data points, we cut any data point if the  $1/f_i$  for the data point is greater than  $1/f$  (for  $t_i < t_0$ ) or any point if  $1/f_i$  is less than  $1/f$  (for  $t_i > t_0$ ). See Section 4 for methods to find  $f$  and  $t_0$ . After this final cut, we then attempt to fit the remaining data points to the Doppler equation. This gives the meaningful parameters  $t_0$ ,  $v$ ,  $s$  and  $v_p$ .

### 4 Finding Airplane Parameters

Let us assume that we have collected data points from an airplane signal from microphone data channels. Using the ROOT program, we then fit this data to a fifth-order polynomial of the form:

$$\frac{1}{f_i} = A_0 + A_1 t_i + A_2 t_i^2 + A_3 t_i^3 + A_4 t_i^4 + A_5 t_i^5 \quad (3)$$

Here,  $f_i$  is the frequency of the bump and  $t_i$  is the time of the bump. The coefficients,  $A_i$  are solved by the ROOT polynomial fitting program.

We need a good first guess and also some constraints so that ROOT will not spend a long time trying to find the appropriate parameters. In order to do this, we observed many airplane events and chose initial values for each  $A_i$ . Then we plotted the time derivatives (see below) for cases in which the airplane flies very near or very far, very fast or very slow, and either high or low frequency. See Figures 3 and 4.

Next, we assume that the airplane signal in the spectrograms is an odd function about an inflection point. This inflection point is at the time of closest approach (neglecting corrections due to the finite speed of sound) and the measured natural frequency of the airplane. Therefore, we map the fifth order polynomial to an odd function of the form:

$$\frac{1}{f_i} = B_0 + B_1(t_i - t_0) + B_3(t_i - t_0)^3 + B_5(t_i - t_0)^5 \quad (4)$$

In this case,  $t_0$  is the inflection point (i.e. the time of closest approach) and  $B_0$  is the natural frequency.

Now expand Equation (4) (dropping subscripts),

$$\begin{aligned} \frac{1}{f} = & B_0 + t(B_1 + 3B_3t_0^2 + 5B_5t_0^4) + t^2(-3B_3t_0 - 10B_5t_0^3) + \\ & t^3(B_3 + 10B_5t_0^2) - 5B_5t_0t^4 + B_5t^5 - t_0(B_1 + B_3t_0^2 + B_5t_0^4) \end{aligned} \quad (5)$$

Then we match similar factors of  $t$  and solve for the unknowns in Equation (4):

$$B_5 = A_5 \quad (6)$$

$$t_0 = \frac{-A_4}{5B_5} \quad (7)$$

$$B_3 = A_3 - 10B_5t_0^2 \quad (8)$$

$$B_1 = A_1 - 3B_3t_0^2 - 5B_5t_0^4 \quad (9)$$

$$B_0 = A_0 + t_0(B_1 + B_3t_0^2 + B_5t_0^4) \quad (10)$$

Now that we have the B coefficients, we can plot the odd function, and we know a good first guess for the time of closest approach,  $t_0$ , and the natural frequency of the airplane,  $B_0$ .

Let us return to the Doppler equation,

$$f' = f \frac{v_0}{v_0 - v_p \cos(\tan^{-1}(s/x))} \quad (11)$$

We want to find  $s$  and  $v_p$  from the data so say we take the inverse of the Doppler equation and its time derivatives, where the superscript (1) is the first time derivative, superscript (2) is the second time derivative and so on (note that  $\dot{x} = -v_p$ ):

$$\frac{1}{f'} = \frac{1}{f} \left( 1 - \frac{v_p}{v_0} \frac{x}{\sqrt{x^2 + s^2}} \right) \quad (12)$$

$$\left( \frac{1}{f'} \right)^{(1)} = \frac{1}{f} \frac{v_p^2}{v_0} \frac{s^2}{(x^2 + s^2)^{3/2}} \quad (13)$$

$$\left( \frac{1}{f'} \right)^{(2)} = \frac{1}{f} \frac{v_p^3}{v_0} \left( \frac{3s^2x}{(x^2 + s^2)^{5/2}} \right) \quad (14)$$

$$\left( \frac{1}{f'} \right)^{(3)} = \frac{1}{f} \frac{v_p^4}{v_0} \left( \frac{12s^2x^2 - 3s^4}{(x^2 + s^2)^{7/2}} \right) \quad (15)$$

$$\left( \frac{1}{f'} \right)^{(4)} = \frac{15s^2}{f} \frac{v_p^5}{v_0} \left( \frac{4x^3 - 3s^2x}{(x^2 + s^2)^{9/2}} \right) \quad (16)$$

$$\left( \frac{1}{f'} \right)^{(5)} = \frac{45s^2}{f} \frac{v_p^6}{v_0} \left( \frac{s^4 + 8x^4 - 12s^2x^2}{(x^2 + s^2)^{11/2}} \right) \quad (17)$$

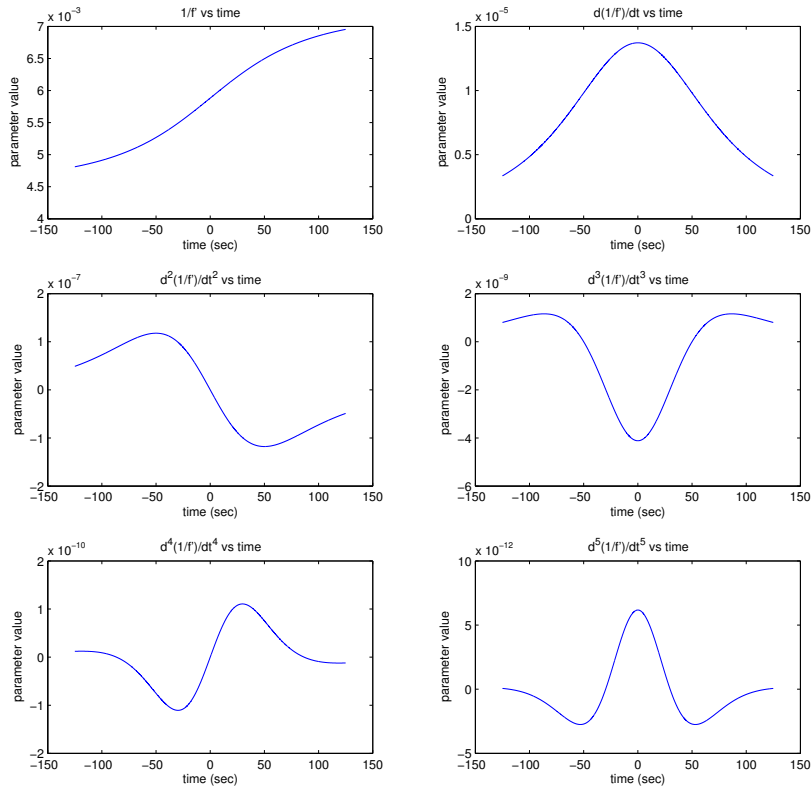


Figure 3: Time derivatives of  $1/f'$  versus time for an “average” event. Here  $v_p = 180$  m/s,  $f = 80$  Hz and  $s = 8000$  m.

When an airplane is at its distance of closest approach,  $d = s$ , we have  $x = 0$ . Equations (12) through

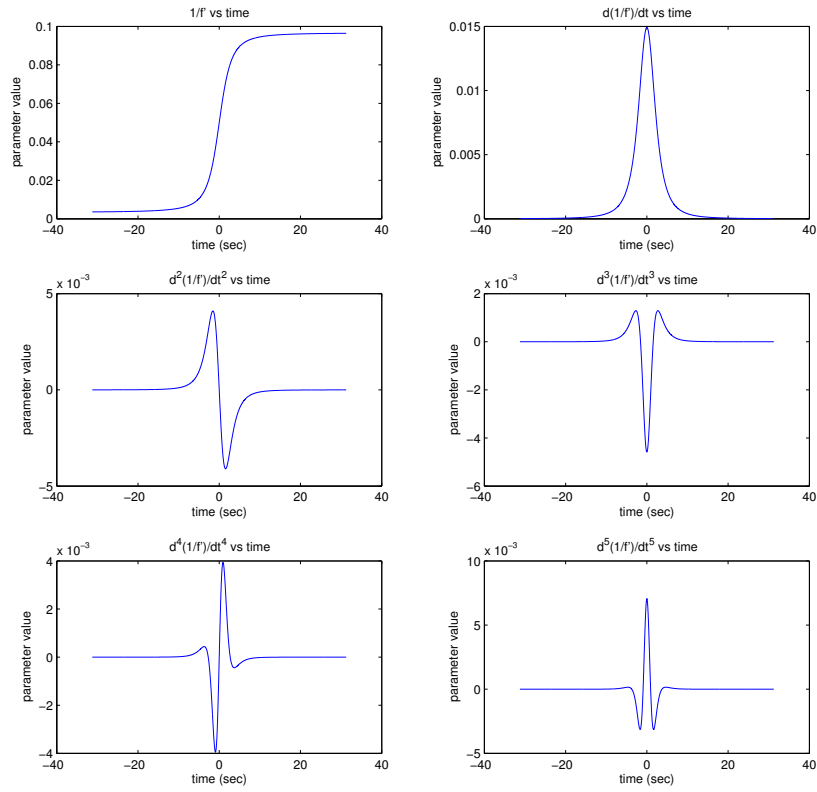


Figure 4: Time derivatives of  $1/f'$  versus time for a “maximal” event. An airplane signal may appear along any part of these curves, so the maximum and minimum values of each plot constrain each  $A_i$ . Here  $v_p = 320$  m/s,  $f = 20$  Hz and  $s = 1000$  m.

(17) then reduce to:

$$\frac{1}{f'} = \frac{1}{f} \quad (18)$$

$$\left(\frac{1}{f'}\right)^{(1)} = \frac{v_p^2}{fv_0s} \quad (19)$$

$$\left(\frac{1}{f'}\right)^{(2)} = 0 \quad (20)$$

$$\left(\frac{1}{f'}\right)^{(3)} = \frac{-3v_p^4}{fv_0s^3} \quad (21)$$

$$\left(\frac{1}{f'}\right)^{(4)} = 0 \quad (22)$$

$$\left(\frac{1}{f'}\right)^{(5)} = \frac{45v_p^6}{fv_0s^5} \quad (23)$$

which we then solve using  $B_0$ ,  $B_1$ , etc.

Now we can use Equations (19) and (21) to solve for  $v_p$  and  $s$ :

$$s = \frac{v_p^2}{f v_0} \left[ \left( \frac{1}{f'} \right)^{(1)} \right]^{-1} \quad (24)$$

$$\Rightarrow v_p^2 = -3f^2 v_0^2 \left[ \left( \frac{1}{f'} \right)^{(1)} \right]^3 \left[ \left( \frac{1}{f'} \right)^{(3)} \right]^{-1} \quad (25)$$

$$\Rightarrow s = -3f v_0 \left[ \left( \frac{1}{f'} \right)^{(1)} \right]^2 \left[ \left( \frac{1}{f'} \right)^{(3)} \right]^{-1} \quad (26)$$

Thus, by determining  $(1/f)^{(1)}$  and  $(1/f)^{(3)}$  at the time of closest approach (using  $B_1$  and  $B_3$  at  $t_0$  from equations (9), (8) and (7), respectively), we can find to a good approximation the values  $v_p$  and  $s$  for an airplane event. This is then used as a starting guess to solve more accurately, the Doppler equation of the form

$$\frac{1}{f'} = \frac{1}{f} \left( 1 - \frac{v_p^2(t-t_0)}{v_0 \sqrt{v_p^2(t-t_0)^2 + s^2}} \right) \quad (27)$$

where the initial values of  $s$  and  $v_p$  come from equations (26) and (25), respectively. The ROOT equation fitting program is used to find approximate values for  $f$ ,  $t_0$ ,  $s$  and  $v_p$ . These values are printed out on any plot that has been fitted using the preceding method.

The next step in the process (which has not yet been implemented) will use parameters from multiple microphone channels to derive an initial guess of the 3-D trajectory of the airplane. A refined trajectory will then be found from fitting simultaneously all the microphone bumps to an expected set, taking into account sound propagation delays.

## 5 Running PlaneMon

PlaneMon has a status summary webpage located through the SPI DMT webpages. This webpage gives a global status of the monitor as well as localized status indicators for each analyzed channel. Through the homepage, one can access real-time data plots of elevated acoustic levels (updated every 15 seconds), archived events listed by GPS time (updated for each new event) and a log file containing useful information for debugging purposes.

During periods when there are no airplanes flying nearby, the global status will read “quiet”. If there is some noise, but not enough to believe it is an airplane, then the global status will read “elevated”. If, however there is enough evidence to believe there is an airplane flying nearby, the global status will read “airplane candidate”. The global status is determined by how many channels are triggered.

A channel is triggered when there are six or more bumps within 15 seconds, with none of those being more than four seconds apart. Once a channel is triggered, it remains triggered for two minutes. Triggered channels are labeled with the local status “activity”. If there are not enough data points in the 15 seconds to trigger the channel, but there are data points, then the channel is labeled as “elevated”. Otherwise, each channel is nominally “quiet”.

### 5.1 Configuration file

PlaneMon must have a configuration file whose name is hard-coded into the program located in the same folder which the PlaneMon program resides. Currently, this file must be named “PlaneMon.conf” and must

have the following information in the following order:

Item	Description
Stride	Time series length to be analyzed
Maximum Strides	Total number of strides to process
Lower frequency	Lower bound on the frequencies to be analyzed
Upper frequency	Upper bound on the frequencies to be analyzed
Number of channels	Total number of channels to be analyzed
List of channels	Names of the channels to be analyzed

## 5.2 Verbosity options

Depending on the amount of acoustic noise one wishes to catalog in the PlaneMon archives, there is an option when PlaneMon is started to indicate the verbosity level. This option has three levels:

Verbosity level	Description
1	<b>Verbose:</b> All channels archived when 1 or more channels are triggered
2	<b>Semi-verbose:</b> All channels archived when 2 or more channels are triggered
3	<b>Quiet:</b> All channels archived when 3 or more channels are triggered

Depending on the user setting of the verbosity, we archive all channel plots depending on the number of channels triggered. For example, if the user has specified a “Verbose” level, then anytime a single channel is triggered, all the channels’ data will be saved over the next two minutes. In addition, anytime a channel is triggered we log this into the database of triggers. When channels begin to show activity we generate alarms to notify interferometer operators and Scimons that there may be airplanes flying nearby.

This is an optional flag and if is not defined it defaults to the “Quiet” mode.

## 5.3 Location

The monitor must be told which location it is operating from. Therefore the “-loc” argument is not optional. There are two options for the two LIGO sites.

Argument	Location
H	Hanford Site
L	Livingston Site

## 5.4 Graphics output

The monitor must be told how to output graphics with no argument (uses netpbm and a perl script run by cronjob) to create gif files from eps files, the -makegif argument which uses netpbm to create the gif files from eps files at the time of creation or the -makejpg argument which creates jpeg files.



## 5.5 Example

As an example, say we wish to run PlaneMon on live data at the Hanford site with the “Semi-verbose” verbosity option and output jpeg graphics. The input would then be: “PlaneMon -verbosity 2 -loc H -makejpg”.