

AN0125 Application Note: Sample Flame Detection Ratio Based Algorithm

1 INTRODUCTION

This document covers a possible flame detection algorithm which has been implemented in Pyreos' ezPyro flame detection demo kits.

2 A DESCRIPTION OF A FLAME DETECTION ALGORITHM

The flame algorithm that is in use on the ezPyro flame detection demo kit uses the magnitude of the FFT plots of each of the sensors. Each sensor has the same data processing applied to it prior to the ratio based comparison.

Key equations for understanding the usage of FFT calculations.

$$1. \text{ Highest Detectable Frequency} = \text{Nyquist Frequency} = \frac{\text{Sample Frequency}}{2}$$

This is important to ensure that the sample frequency is set high enough to observe the highest frequency that you are interested in.

$$2. \text{ Number of FFT Bins} = \frac{\text{FFT Window Width}}{2}$$

This is used to determine the resolution of the FFT in equation 3.

$$3. \text{ Resolution of Bins} = \frac{\text{Nyquist Frequency}}{\text{Number of FFT Bins}}$$

The resolution is important for the situation where you want to ignore the bins that have leakage from the DC bin. If you want to accurately observe the frequency at 1 Hz then you must ensure that the resolution allows for this frequency to be monitored without incorporating the bins that DC components have affected due to spectral leakage of the FFT. This is easily visualized by changing the window width in the flame demo software whilst viewing the FFT plots of each channel. Larger window widths allow for better resolution.

An FFT bin is the name given to grouping a range of frequencies into a single value centred at a particular frequency. Increasing the number of FFT bins means that the number of frequencies being grouped together is smaller and a larger number of bins will comprise the same frequency range. Please see appendix Section 1 for a visual aid to understanding FFT bin resolution.

$$4. \text{ Time Period of Window} = \text{FFT Window Width} \times \text{Sample Period}$$

The time period of the window being analysed is important when considering the response time of the algorithm. The longer the window the longer the algorithm takes to respond. It is a trade-off between response time and stability.

The data processing prior to the comparison is comprised of 5 steps described in section 3.

3 INITIAL SENSOR DATA PROCESSING

3.1 FFT Window Width

This is the window of data that is being used by the FFT. It can be visualized by a plot of data containing exactly the number of data points set by the parameter FFT Window width.

The image below shows a window of data 1024 points long.

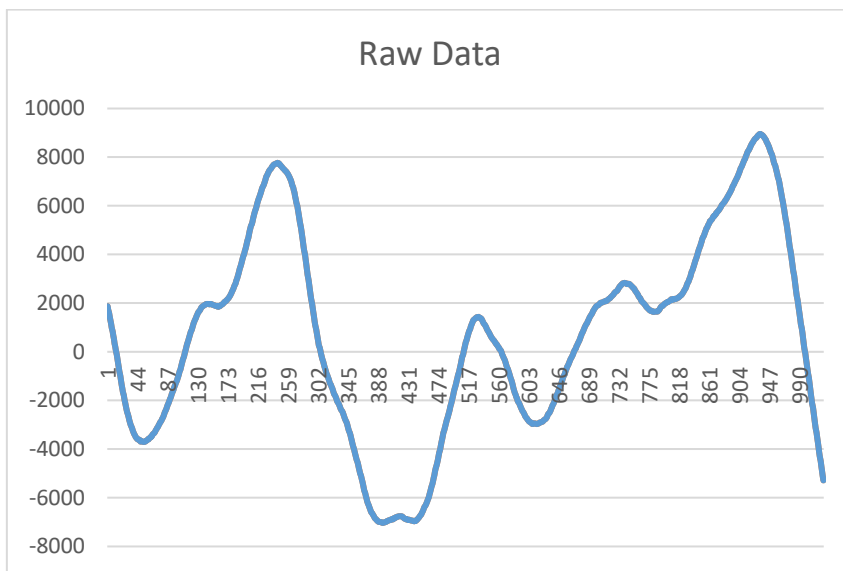


Figure 1: Raw Fire Data

3.2 Hanning Window

The Hanning window is used as it reduces an aspect of the FFT calculation known as spectral leakage. The Hanning window is a data set of the same length as raw data set.

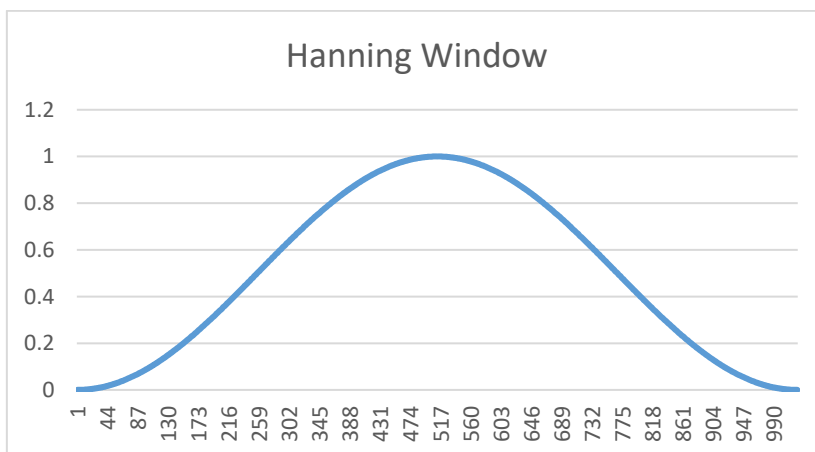


Figure 2: 1024 Point Hanning Window

The equation for producing each element, ‘n’, of a Hanning window for a data set ‘N’ elements long is

$$w(n) = \sin^2\left(\frac{\pi n}{N - 1}\right)$$

The Hanning window data set is then multiplied by the FFT window raw data element by element

$$Hanned\ Data\ Window[i] = Hanning\ Window[i] \times Raw\ Data\ Window[i]$$

This forces the ends of the data set to zero in a smooth manner.

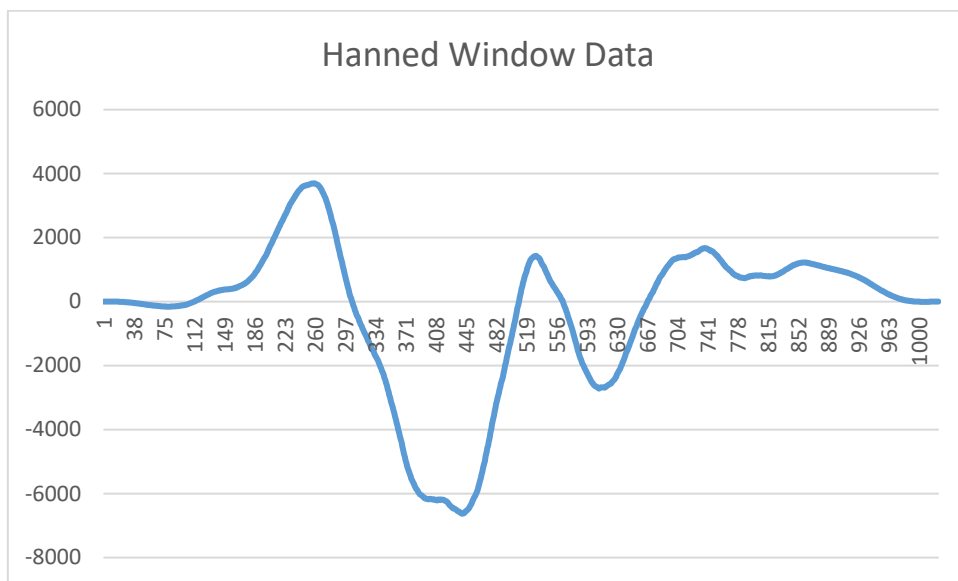


Figure 3: Hanning Window Modified Data Set

3.3 FFT

The data at this stage has been prepared for the FFT function to be performed. The output of an FFT can be visualised well using a bar plot. An example of FFT bar plot is shown below.

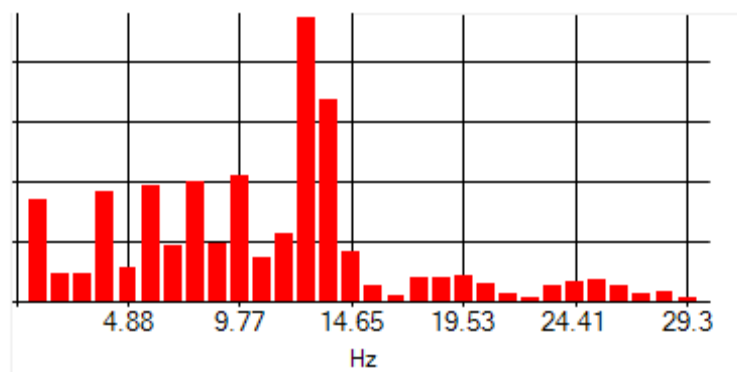


Figure 4: FFT plot using 1024 Window Width

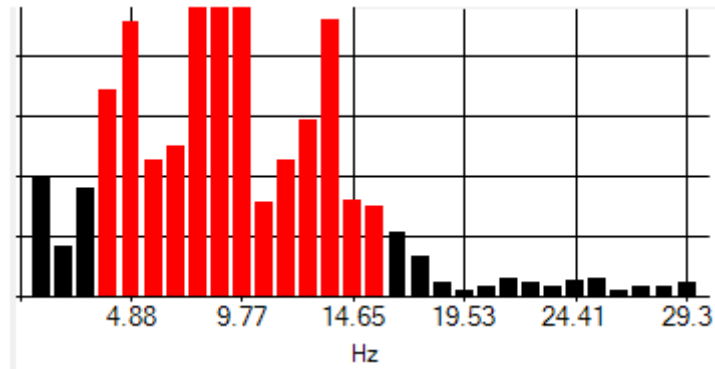
This provides a data set corresponding to the signal strength for each frequency able to be determined based on the set up of the algorithm.

3.4 FFT Summation

Now that the frequencies are available in the FFT dataset this stage involves looking at the desired frequency range. This is done simply by summing the values of the bins between the desired low and high frequencies.

$$Channel_x \text{ Signal Strength} = \sum_{Low \text{ Freq}}^{High \text{ Freq}} FFT \text{ Dataset}$$

For example using the range of 4 and 16 Hz would add all the values of the following graph that are in red and ignore anything outside this range.



By doing the above summation we now have a single value that represents the signal strength of a particular sensor in the range of frequencies set by the algorithm parameters.

3.5 Signal Multiplier

This next stage allows for further calibration of the system. It is simply a means to reduce or increase the value produced by the summation performed previously. This allows for compensation of differences in responsivity and noise levels of the sensors used in the system.

$$\mu_x(\text{Channel}_x \text{ Signal Strength}) = \mu_x\left(\sum_{Low \text{ Freq}}^{High \text{ Freq}} FFT \text{ Dataset}\right)$$

The effect of μ_x is shown below, it is simply multiplication.

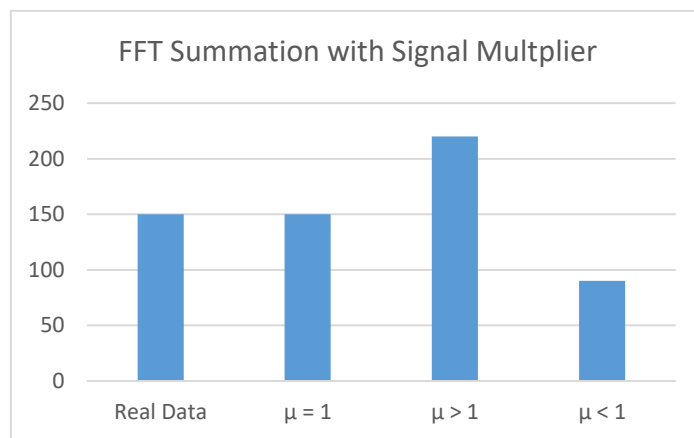


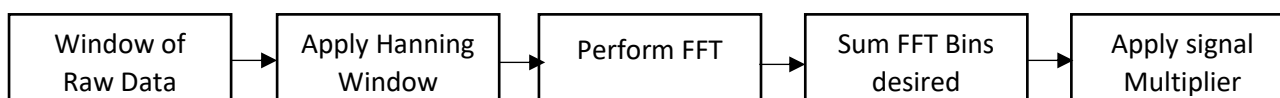
Figure 5: Signal Multiplier Effect

3.6 Summary of Data Processing

The previous sections describe the stages required to process the data when doing an FFT based flame detection algorithm.

1. Get a full window of data corresponding to the FFT window width that you have selected.
2. Multiply element by element to produce a data set that converges to zero at either end.
3. Perform the FFT on the data set produced from raw data and Hanning window.
4. Sum the value of the FFT bins in the desired frequency range.
5. Apply a single multiplier if needed to compensate for any system level differences.

The flow of this is as shown below.



The outcome of the above stages results in a data point for each sensor which for the next section will be labelled according to the ezPyro SMD flame demo kit channels. The labels specific to this user case are listed below. For the ezPyro TO demo kit the 4.64um flame sensor is omitted (LP = Long pass, BP = Band pass)

$$1. \quad \varepsilon_1 = \text{Ch1 Human Motion Rejection (5.0 um LP)} = \mu_1 \times \sum_{\text{Low Freq}}^{\text{High Freq}} (\text{FFT HM})$$

$$2. \quad \varepsilon_2 = \text{Ch2 Sunlight Rejection (3.91 um BP)} = \mu_2 \times \sum_{\text{Low Freq}}^{\text{High Freq}} (\text{FFT SR})$$

$$3. \quad \sigma_1 = \text{Ch3 Flame Channel 1 (4.64 um BP)} = \mu_3 \times \sum_{\text{Low Freq}}^{\text{High Freq}} (\text{FFT } F_1)$$

$$4. \quad \sigma_2 = \text{Ch3 Flame Channel 1 (4.64 um BP)} = \mu_4 \times \sum_{\text{Low Freq}}^{\text{High Freq}} (\text{FFT } F_1)$$

Epsilon 1 and 2 ($\varepsilon_1, \varepsilon_2$) represent the rejection channels after the data processing listed in this section and sigma 1 and 2 (σ_1, σ_2) represent the flame channels output for the data processing listed above.

With these it is now possible to check the ratio of the post processed flame channels against the rejection channels. This is described in the next two sections.

4 RATIO BASED ALGORITHM

Since the algorithm is based on the ratio of the flame channels to rejection channels post processed values we need to select our ratio which if met will produce a fire detection result. This comes down to setting up the values. The values will differ from system to system but can be adjusted to take system differences into account.

$$\text{Human Motion Rejection Ratio Threshold} = 1.5 = \frac{\sigma_1}{\varepsilon_1}$$

Or

$$\text{Human Motion Rejection Ratio Threshold} = 1.5 = \frac{\sigma_2}{\varepsilon_1}$$

This means that at least one of the flame channels must produce an output signal after the processing described in section 3 that is at least 1.5 times the rejection channel. It is the same for the sunlight rejection channel

$$\text{Sunlight Rejection Ratio Threshold} = 1.2 = \frac{\sigma_1}{\varepsilon_2}$$

Or

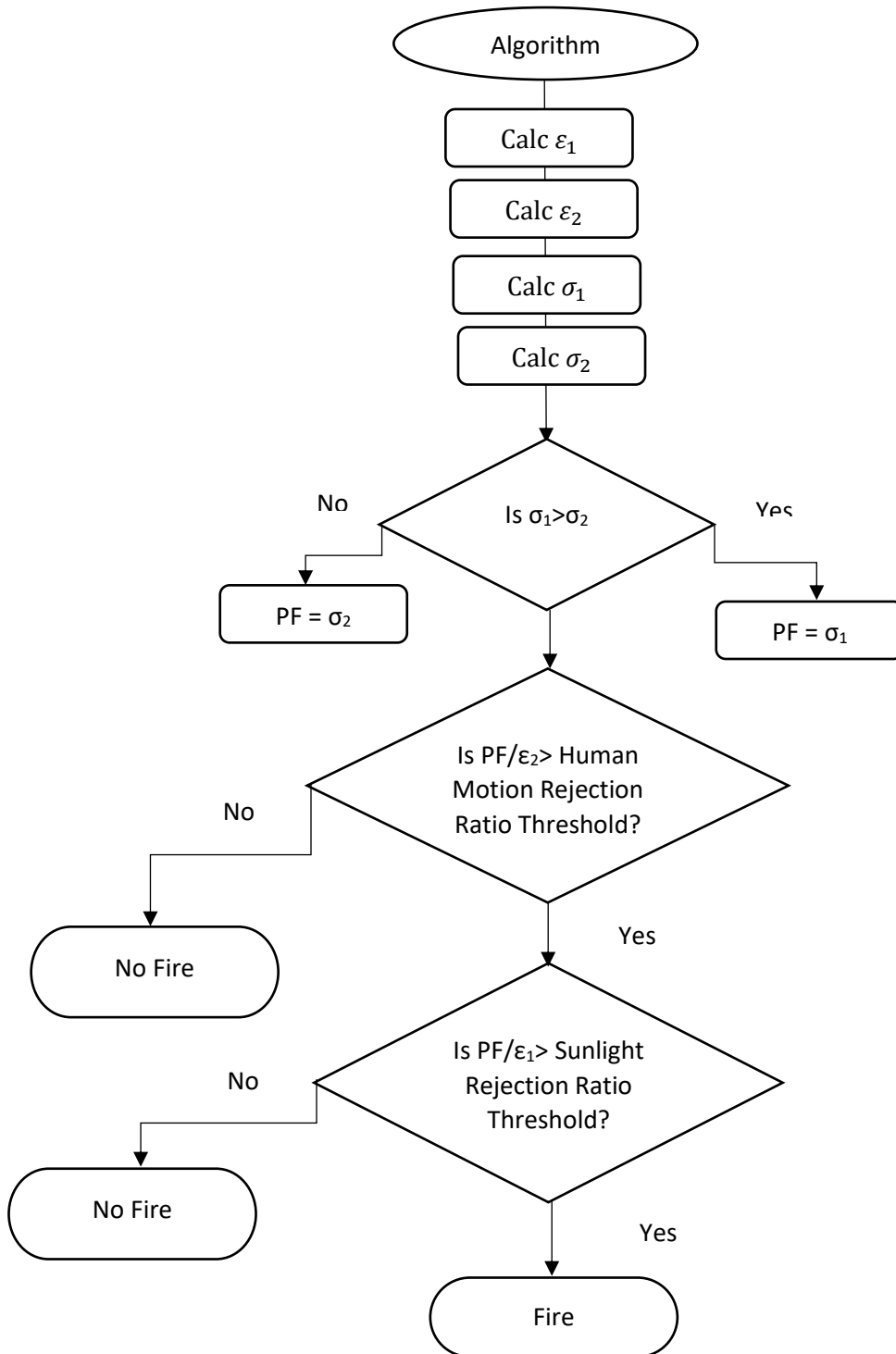
$$\text{Sunlight Rejection Ratio Threshold} = 1.2 = \frac{\sigma_2}{\varepsilon_2}$$

For a flame event to be considered real one of the flame sensors data must be greater than the ratios of both of the two rejection channels.

Since one of the flame channels will be larger than the other it is only required to look at the ratios of the largest flame channel signal against the rejections.

This is shown in the flowchart in the next section.

5 FLOWCHART



6 SUMMARY

The key points are listed below.

1. Perform post processing as described in section 3.
2. Determine largest flame channel signal.
3. Divide the Flame signal by rejection channel 1. If greater than that channels threshold check other channel.
4. Divide the Flame signal by rejection channel 2. If greater than that channels threshold a flame event has been registered.

7 ADDITIONAL FEATURES

The final part of the flow diagram can be replaced with a counter. The counter would store some number of fire or no fire events. When the ratio of fire to no fire events reaches some proportion of the total number of recorded events then a flame event has occurred.

The final part of the flow diagram can be replaced with a counter. The counter would store some number of fire or no fire events. When the ratio of fire to no fire events reaches some proportion of the total number of recorded events then a fire can be said to be in the FoV of the sensors. This makes the system less responsive but more robust against short time duration events that could cause a false alarm 4 ~ 5 cm. The main flame flicker frequency of this test condition is normally between 2 ~ 4 Hz. This implies a 2~4 Hz amplifier is required for IR flame detection.

8 MCU BASED ALGORITHM

Due to the computational expense of the FFT, especially on larger window sizes, no manufacturers use the FFT calculation on a microcontroller.

It is extremely useful to view the data in such a way as it shows you the actual frequencies that you should be interested in. For initial testing with our demo kit this is very desirable.

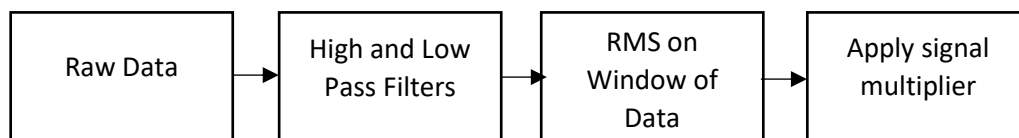
Most system designers use a computationally less expensive form of the same type of processing explained in this document.

1. High and low pass filter, implemented in op-amps when using an analogue sensor. When using a digital product such as ezPyro the same effect can be accomplished with digital signal processing.

At this stage the signal still contains the frequencies you are interested in as set by the filters being used. However if the demo kit used this method you might not see other frequencies that could use in your algorithm.

2. Perform RMS on a window of data to determine the signal strength of the combination of the frequencies in the bandpass of the filters used.

This is equivalent to being at the beginning of stage 5 of the data processing described in this document prior to applying the signal multipliers.



The window of data in the 3rd box has the same effect as increasing the FFT window width in terms of the time response to an input signal. Longer windows are more stable but slower to respond. This can be set to any value desired as it does not have the restriction of being value of the form 2^x like the FFT.

After the signal multipliers the same comparisons can be performed as written in sections 4 and 5.

9 APPENDIX

9.1 FFT Bins

The easiest way to visualise the effect of increased resolution of the FFT is to show the plot of the FFTs produced from our sensors. Increased resolution of the FFT means that there are a larger number of FFT bins within the same range of frequencies.

For a sample frequency of 1KHz (highest ezPyro sampling frequency) and plotting the FFTs within 0 to 30 Hz using the ezPyro flame demo software shows the increased number of bins as the window width is increased. The frequency range being included by the algorithm has been left as 4 to 16 Hz for all the diagrams. As can be seen in figure 6 the range of 4 to 16 Hz includes the first bin that will contain leakage from the DC bin. This is generally undesirable.

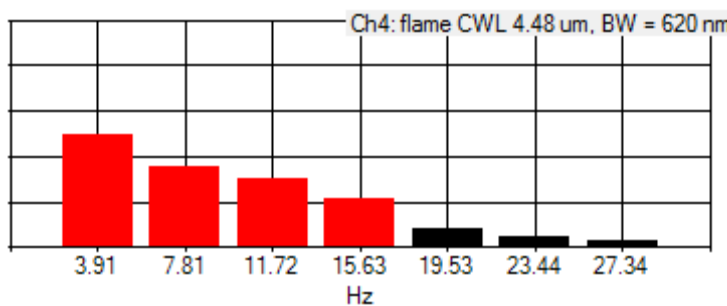


Figure 6: FFT Window Width = 256

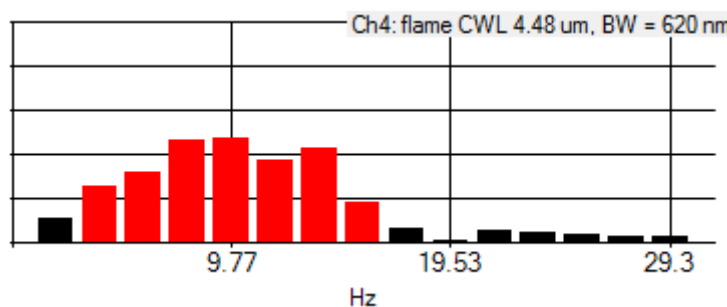


Figure 7: FFT Window Width = 512

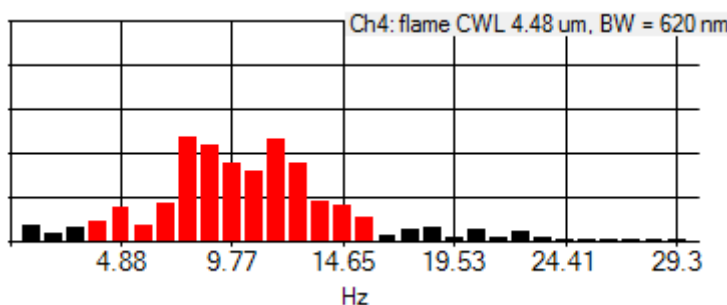


Figure 8: FFT Window Width = 1024

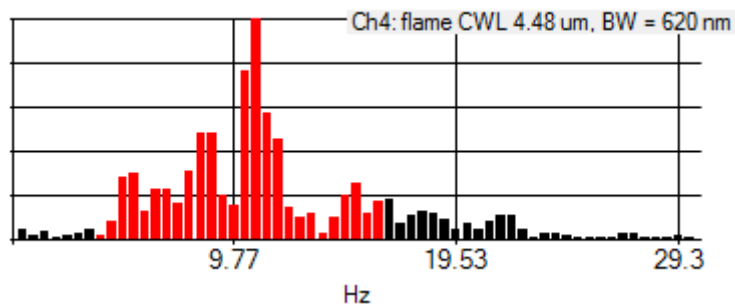


Figure 9: FFT Window Width = 2048

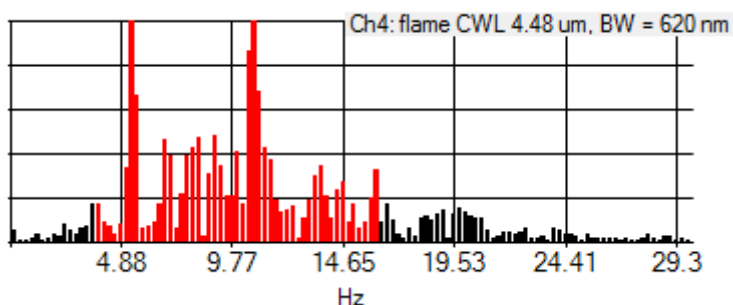


Figure 10: FFT Window Width = 4096

As can be seen the number of bins within the range 0 to 30 Hz increases as the window of data being analysed is increased.

Increasing the resolution in this way allows the user to be more selective of the cut-off frequencies but longer windows of data are slower to respond to new events. This is a compromise that needs to be determined through testing for a specific application.

Summary:

1. FFT window width should be large enough to allow the user to select the frequency range desired accurately.
2. FFT window width should be large enough to give a resolution that allows the bins susceptible to DC bin leakage to be ignored by the algorithm.
3. FFT window width should not be too large to introduce undesired time delays associated with large windows of data to show significant response to input signals.

10 REVISION HISTORY

Revision Number	Revision Date	Description of Changes
1.0	20/07/2018	First Version
1.1	29/08/2018	Ratio Based Algorithm Version – First Version
1.2	12/10/2018	Small edits (Presentation)