

Real-time spectral analysis using Prism Signal Processing

Henry, M.

Published PDF deposited in Coventry University's Repository

Original citation:

Henry, M 2019, 'Real-time spectral analysis using Prism Signal Processing', Journal of Physics: Conference Series, vol. 1379, no. 1, 012030.

<https://dx.doi.org/10.1088/1742-6596/1379/1/012030>

DOI 10.1088/1742-6596/1379/1/012030

ISSN 1742-6588

ESSN 1742-6596

Publisher: IOP Publishing

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

PAPER • OPEN ACCESS

Real-time spectral analysis using Prism Signal Processing

To cite this article: Manus Henry 2019 *J. Phys.: Conf. Ser.* **1379** 012030

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Real-time spectral analysis using Prism Signal Processing

Manus Henry^{1,2}

¹Department of Engineering Science, University of Oxford.

²School of Electrical Engineering and Computer Science, South Ural State University, Chelyabinsk, Russia

E-mail: manus.henry@eng.ox.ac.uk

Abstract. The Prism is a new signal processing object which provides fully recursive FIR filtering i.e. the computational load is small and fixed irrespective of filter length. The Prism also has negligible design cost. These features are attractive to the next generation of Industry 4.0 sensors requiring low cost and flexible signal processing. Recent work has demonstrated how, using a few simple design rules, a chain of six Prisms can create narrowband filters with arbitrary central frequency and bandwidth, also with low design and computational cost. In this paper this technique is extended to provide spectral analysis of data, whereby a sequence of filters is applied to a data set to provide selective frequency analysis.

1. Prism signal processing

The Prism (precise, repeat integral signal monitor), is a recently developed signal processing object. It is based upon a Fourier-style double integration applied recursively via a sliding windows calculation. Either one or two outputs are generated. If there are two, these form a sine/cosine pair from which the properties of the input signal, such as amplitude, phase and/or frequency, can be calculated. Unlike conventional, convolution-based finite impulse response (FIR) filters, the Prism calculation is fully recursive, having a low computational cost which is independent of window length; its benefits include a linear phase response, and a low design cost, as the filter ‘coefficients’ are simply linearly spaced sine and cosine values. Although the frequency response design options for a single Prism are relatively limited, serial and/or parallel networks of Prisms can be created to provide a wider range of responses suited to a range of tasks including notch and bandpass filtering, or tracking multiple frequency components in a signal.

In [1], an introduction to Prism Signal Processing (PSP) is provided as with potential applications in the Internet of Things/Industry 4.0. Current applications include sensor validation [2] generally, pressure sensors [3], the condition monitoring of rotating machinery [4, 5], and the first application of Coriolis mass flow metering to fuel injection monitoring in internal combustion engines [6].

Consider a sinusoidal input signal (or indeed a signal component at a specified frequency)

$$s(t) = A \sin(2\pi ft + \phi), \quad (1)$$

where t (or τ) is time, A is the amplitude, f is the frequency, and ϕ is the initial phase offset at $t = 0$. The basic mathematical operation used in the Prism is to calculate double integrals of the form:

$$I_{[s|c][s|c]}^h = m^2 \int_{-\frac{1}{m}}^0 [\sin | \cos](2\pi hmt) \left(\int_{t-\frac{1}{m}}^t [\sin | \cos](2\pi hm\tau) \cdot s(\tau) d\tau \right) dt \quad (2)$$



where $[s | c]$ indicates choosing one of s (sin) or c (cosine), h is the harmonic number, a positive integer, and m is the characteristic frequency of the Prism. Equation (2) describes a family of Fourier-style double integrals whereby at each level the input signal is multiplied by a modulating sine or cosine function and integrated over the period of the characteristic frequency m . The harmonic number h determines how many whole cycles of the modulation function occur over the period of integration $1/m$. In the Prism itself, either one or two pairs of such integrals are evaluated each sample. The output of each integral pair is combined to provide a filtered output of the input signal:

$$G_s^h = I_{ss}^h + I_{cc}^h = A \operatorname{sinc}^2(r) \frac{r^2}{r^2 - h^2} \sin(\phi - 2\pi r) \quad (3)$$

$$G_c^h = I_{cs}^h - I_{sc}^h = A \operatorname{sinc}^2(r) \frac{hr}{r^2 - h^2} \cos(\phi - 2\pi r) \quad (4)$$

2. PSP narrowband filtering

As described in [7, 8], simple design rules have been developed for building narrowband FIR filters using a signal chain consisting of three pairs of Prisms. Figure 1 shows the frequency response of the generic Prism filter for an arbitrary central frequency c and bandwidth b .

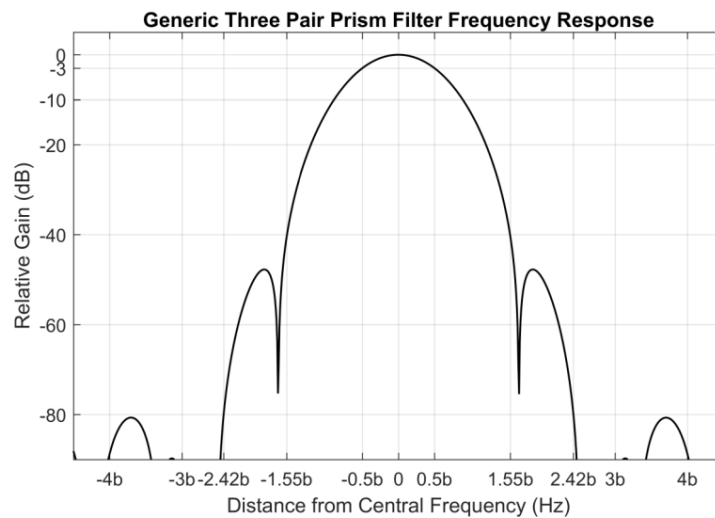


Figure 1. Frequency response of bandpass filter formed from three Prism pairs.

Prism narrowband filtering offers a significant speed-up compared with conventional FIR filtering. In [8], an FPGA implementation of real-time Prism narrowband filtering is described, in which a signal sampled at 2 MHz can be filtered around an arbitrary central frequency with a bandwidth as narrow as 0.1 Hz. This requires a total Prism filter length of 192 million samples, but because the calculation is recursive, it is able to run in real time on a single FPGA, as long as sufficient memory is available to store the sample data. A conventional, convolution-based, non-recursive FIR filter would require 384 TMAC/s (i.e. 3.84×10^{14} multiply-and-accumulate operations per second i.e. supercomputer performance) to achieve this throughput, assuming such long filters could be designed.

The combination of instant filter design and fast filter throughput provides the opportunity for developing Prism spectral analysis, as described in the next section.

3. Prism spectral analysis

Prism spectral analysis is carried out by passing the data set to be analyzed through a series of frequency-specific detectors, one for each frequency at which amplitude information is required. Each

detector consists of a narrowband filter followed by a Prism-based tracker ([1, 6]) to calculate the amplitude of the resulting filtered signal. Within the tracker, the two sine/cosine outputs of a Prism (given in equations (3) and (4)) are combined in a weighted root-sum square calculation to provide an estimate of the amplitude of the input signal on a sample-by-sample basis. Such amplitude estimates can only be provided once the narrowband filter and the tracker filter are filled with samples, and so one design consideration is the tradeoff between the narrowband filter length and the data set to be examined. To generate a single amplitude value to represent the corresponding frequency over the data set, the average value can be used, as discussed in the simulation example below.

The principle benefits of the Prism spectral analysis technique are flexibility and numerical accuracy, the latter arising from the suppression of frequency leakage via the narrowband filtering. Flexibility is provided because the range, spacing and filter bandwidths to be analyzed are entirely arbitrary. While it is certainly possible, as illustrated below, to perform an analysis across a large contiguous range of spectral frequencies, it is also possible to select only individual and/or sub-ranges of frequencies for detailed, narrow bandwidth analysis, thus avoiding the outlay of computational effort on spectral regions which are not of interest.

The computational requirements of Prism spectral analysis, based on recursive FIR techniques, are low enough to make the technique viable, but remain significantly higher than for the FFT itself. The computational load for an FFT is $O(N \log_2 N)$ for a data set of size N . By contrast, for Prism signal processing, the computational cost for *each frequency detector* is approximately $300 N$. Thus only rarely would the technique be suitable for narrowband analysis across a wide frequency range. More typically, a mixture of coarse and narrow frequency detectors would be used, as discussed at the end of the paper.

4. Simulation Example

A comparison between conventional FFT and Prism-based spectral analysis techniques is given via a simulation example, based upon typical sensor signal characteristics from a Coriolis mass flow meter. Table 1 gives the frequencies and amplitudes of five components in the simulated sensor signal. Data sets were generated using 50 kHz sampling, either noise-free or with white noise (0.01 V standard deviation) added. Each simulation data set consisted of 2^{20} or approximately 1 million samples, corresponding to a time series of approximately 20 s duration.

Table 1. Frequencies and amplitudes of components in simulated sensor signal.

Frequency (Hz)	Amplitude (V)
80	0.2
160	0.02
190	0.01
240	0.015
320	0.001

Results for the FFT calculation were derived using the standard MATLAB call *fft()*. Note that a wide variety of FFT algorithms exist, with various advantages and disadvantages; an optimal choice will depend upon the characteristics of the data set and the analytic requirements. The intention here is simply to use a straightforward and widely accessible algorithm as a basis for comparison with the Prism technique.

The Prism spectral analysis was carried out over the range 12 Hz – 500 Hz, at intervals of 0.1 Hz. Note that all Prisms have a gain of zero at DC and so a non-zero lowest frequency must be selected. The filter bandwidth was also set to 0.1 Hz over the entire range, but these choices are arbitrary and

independent for each detector. As discussed in [7], the sample length of the narrowband filter design shown in Figure 1 is approximately $1.92fs/b$ where fs is the sampling rate and b is the filter bandwidth. Each frequency detector also requires a tracker, but this is relatively short compared to the bandpass filter. For a detector at frequency $f \in [12, 500]$ Hz, the optimal tracker Prism length is fs/f samples; for the lowest frequency, 12 Hz, this amounts to approximately 4000 samples. Overall, therefore, the detector length is approximately 5×10^5 samples, compared with the data set length of approximately 10^6 samples. Accordingly, the detector ‘warm-up’ requires approximately half the data set, and a time series of approximately 5×10^5 amplitude values are generated for averaging. While a conventional convolutional FIR filter of order 5×10^5 , applied to a data set of length N samples, would require $10^6 N$ arithmetic operations (i.e. 5×10^5 each of multiply and accumulate per sample), the Prism bandpass filter requirement is only $300 N$. With different design choices, for example selecting a bandwidth of around 0.05 Hz, a longer bandpass filter would result in fewer amplitude values generated from the same data set.

Figures 2 - 5 show typical results obtained by applying FFT and Prism spectral analyses to noise-free and noisy data sets. In the noise free case the Prism analysis generates a significantly lower noise floor compared to the FFT, and in both cases the Prism narrow-band filtering results in a reduction in frequency leakage, resulting in sharp spectral peaks.

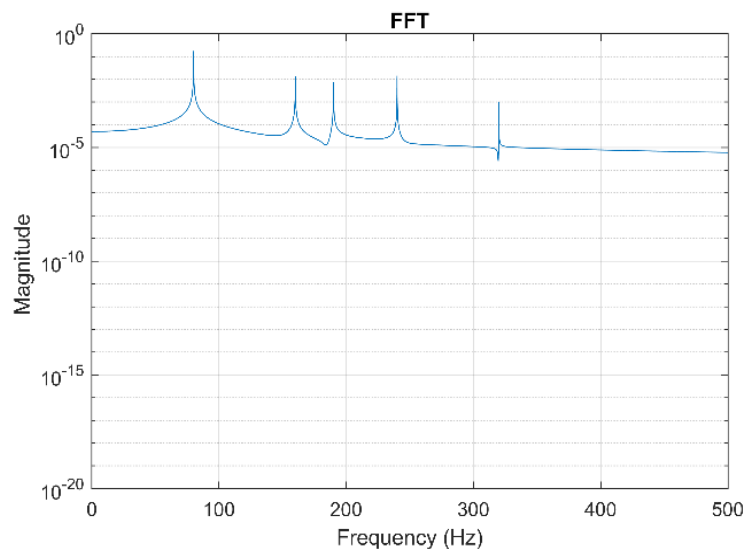


Figure 2. FFT analysis of simulated data set – noise free.

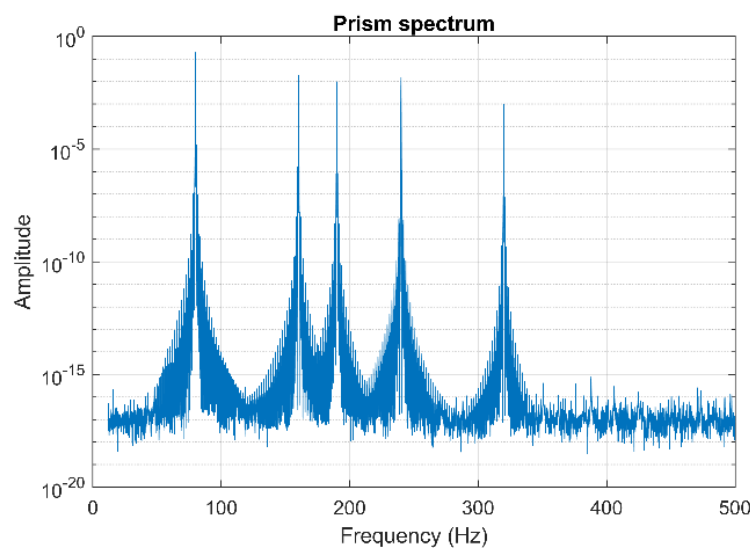


Figure 3. Prism analysis of simulated data set – noise free.

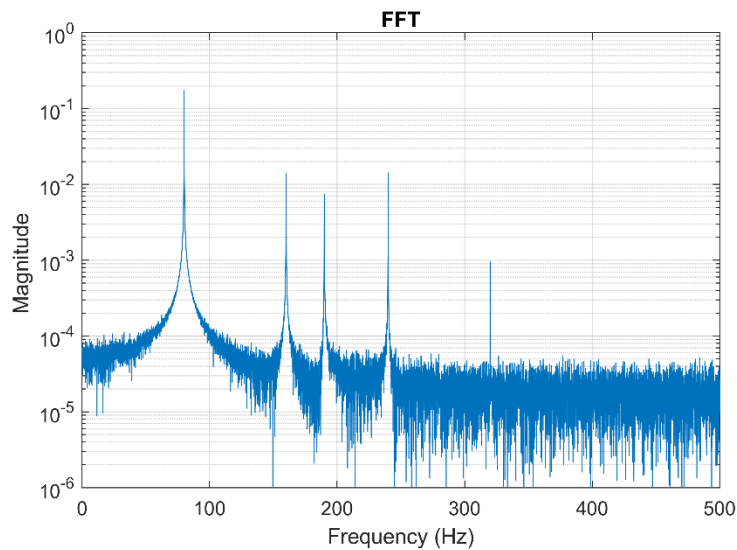


Figure 4. FFT analysis of simulated data set – with added noise.

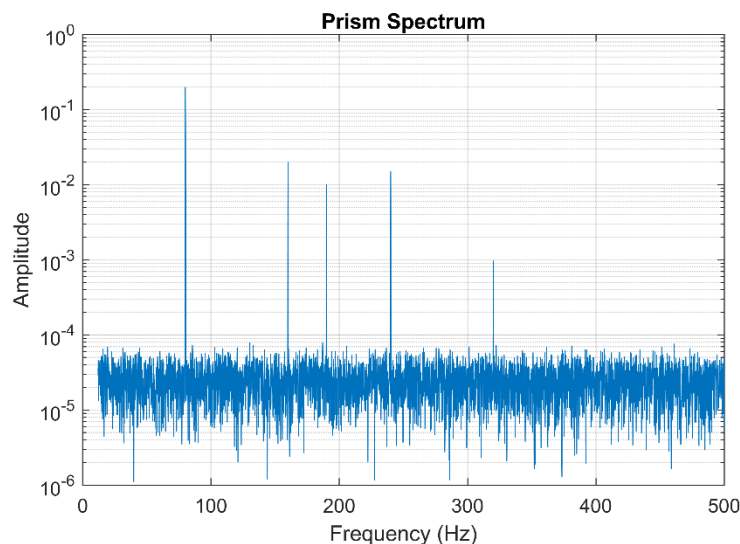


Figure 5. Prism analysis of simulated data set – with added noise.

A quantitative assessment of the Prism spectral analysis has been carried out as follows. For each case (noise-free and noisy), 100 data sets were generated. For both the noise-free and the noisy data sets, the initial phase of each frequency component (Table 1) was generated randomly, so that each of the 100 data sets is distinct. In addition, for the noisy data, different random noise sequences were added to each data set. Prism spectral analysis was applied to each of the two sets of 100 trials to obtain the calculated amplitude of each of the frequency components listed in table 1. Based on these results, the mean, mean error, and standard deviation were collated for the calculated amplitude for each component. The mean error and the standard deviation are scaled as a percentage of the true amplitude of the corresponding frequency component, so that the relative accuracy and variability can be assessed. The results are shown in tables 2 and 3.

In the noise-free case (Table 2), high amplitude accuracy is achieved, irrespective of the initial phase of the frequency components, so that the mean errors and standard deviations are small. In the noisy case (Table 3) the average errors remain small, while the standard deviation of the error increases as the amplitude of the signal component decreases relative to the noise level. For example, for the 80 Hz component, which has a true amplitude of 0.2 V, the calculated value had a standard deviation of approximately 0.1 % of the mean value, while for the 320 Hz component, with a true amplitude of 1 mV, the calculated value had a standard deviation of approximately 2 %.

Table 2. Results of Prism tracking of spectral component amplitudes in simulated sensor signal – no noise (100 trials).

Frequency (Hz)	True Amplitude (V)	Calculated Amplitude Mean (V)	Calculated Amplitude Mean Error (%)	Calculated Amplitude Standard Deviation (% of mean)
80	0.2	2.0000e-01	-1.1793e-08	1.4557e-10
160	0.02	2.0000e-02	-1.8900e-07	3.3356e-10
190	0.01	1.0000e-02	-3.3471e-07	1.0579e-09
240	0.015	1.5000e-02	-9.5902e-07	3.2710e-09
320	0.001	1.0000e-03	-3.0413e-06	4.9119e-09

Table 3. Results of Prism tracking of spectral component amplitudes in simulated sensor signal – with added white noise 0.01 V standard deviation (100 trials).

Frequency (Hz)	True Amplitude (V)	Calculated Amplitude Mean (V)	Calculated Amplitude Mean Error (%)	Calculated Amplitude Standard Deviation (% of mean)
80	0.2	2.0000e-01	-1.7747e-03	1.0151e-02
160	0.02	1.9999e-02	-3.1187e-03	1.0999e-01
190	0.01	9.9998e-03	-1.6958e-03	1.7003e-01
240	0.015	1.5001e-02	3.9012e-03	1.4967e-01
320	0.001	1.0006e-03	6.3456e-02	1.9849e+00

5. Conclusions

Prism signal processing is a recursive FIR technique offering low design and computational cost compared with conventional convolutional forms of FIR filtering. This paper has demonstrated its potential as an alternative approach to spectral analysis. Frequency leakage is minimized by the use of narrowband filtering, leading to accurate estimation of amplitude at each selected frequency component, and the technique affords complete flexibility with regard to the frequencies and bandwidths to be analysed. While its computational cost is unlikely to be competitive with the FFT over the full spectral range of a data set, Prism spectral analysis may prove useful for examining key regions of a data spectrum in detail (perhaps where these key regions are identified using conventional FFT or widely spaced, broader-band Prism spectral analysis), or for data sets where its high accuracy warrants the higher computational requirement.

References

- [1] Henry M, Leach F, Davy M, Bushuev O, Tombs M S, Zhou F and Karout S 2017 *IEEE Ind. Electron. M.* **11**. URL 10.1109/MIE.2017.2760108
- [2] Henry M P 2018 *Meas. Tech.* URL 10.1007/s11018-018-1345-1
- [3] Henry M P, Bushuev O and Ibryaeva O 2017 *26th IEEE Int. Symp. on Industrial Electronics (19-21 June 2017, Edinburgh, Scotland, UK)* URL 10.1109/ISIE.2017.8001451
- [4] Henry M and Sinitsin V V 2018 *Proc. IEEE Ind Cyber-Physical Syst ICPS (May 15-18, 2018, St.Petersburg, Russia)* pp 452-457. URL 10.1109/ICPHYS.2018.8390747
- [5] Henry M and Sinitsin V V 2018 *Proc. IEEE Ind Cyber-Physical Syst ICPS (May 15-18, 2018, St.Petersburg, Russia)* pp 458-463 URL 10.1109/ICPHYS.2018.8390748

- [6] Leach F, Karout S, Zhou F, Tombs M, Davy M and Henry M 2017 *Flow Meas Instrum.* URL 10.1016/j.flowmeasinst.2017.09.009
- [7] Henry M P 2018 *44th Annual Conf. of the IEEE Industrial Electronics Society IECON 2018 (October 21-23, 2018, Washington D.C., USA)*
- [8] Owen J and Henry M 2018 *44th Annual Conf. of the IEEE Industrial Electronics Society IECON 2018 (October 21-23, 2018, Washington D.C., USA)* URL 10.1109/iecon.2018.8591619