Nonlinear filtering in digital phase sensitive demodulation

Ziqiang Cui[1,2]‡, Huaxiang Wang[1] and Wuliang Yin[2]

1. Tianjin key laboratory of process measurement and control, School of electrical engineering and automation, Tianjin University, Tianjin, 300072, China
2. School of Electrical and Electronic Engineering, The University of Manchester, Manchester M13 9PL, UK

E-mail: cuiziqiang@tju.edu.cn

July 2014
To be submitted to MST.

Abstract. The phase sensitive demodulation has been widely used in many applications, *e.g.* impedance measurement, communication, sonar and radar systems. In most cases, white noise is assumed for system analysis and improvement. However, impulsive noise is frequently encountered in many applications, which imposes great challenges in these systems. The paper presents a novel nonlinear filter method intended for the impulsive noise removal. Unlike its linear counterparts, the proposed method tries to figure out and remove the impulsive noise in time domain rather than in frequency domain. This ensures simpler implementation in digital signal processing. We compare the performance of the proposed method with the standard PSD and the PSD with low-pass filter in suppressing both white noise and impulsive noise and analyse the theoretical limits of the signal-to-noise ratio (SNR). A FPGA based hardware implementation of the proposed method is presented for reference. Our numerical simulations and experimental results validate the theoretical predications, which show that the proposed method can provide SNR of $10dB$ better than the standard PSD method.

1. Introduction

The phase sensitive demodulation (PSD) is a critical problem encountered in many communication and measurement systems. Numerous methods for demodulation of the carrier amplitude and phase, including analog multiplier demodulation technique has become standard. By using PSD, it is possible to eliminate the influence of low-frequency noise sources such as $1/f$ noise, mechanical vibrations, atmospheric fluctuations in remote sensing optoelectronic instrumentation, etc. The basic principle of the PSD is to use a carrier wave of certain frequency, i.e. $f : z(t) = cos(2\pi ft)$, to modulate the physical parameter that to be measured. Thus, instead of directly measurement, we get

‡ To whom correspondence should be addressed.

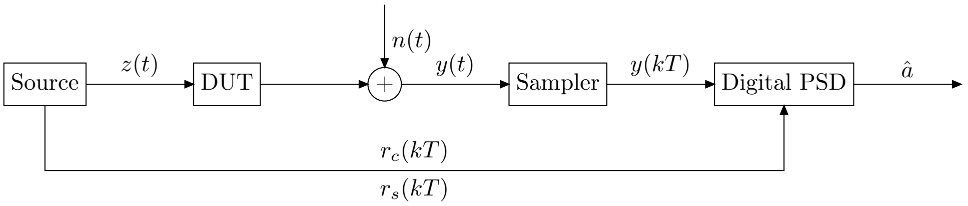


Figure 1. The digital PSD model. a denotes the amplitude change caused by the DUT. The signal $az(t)$ is then corrupted by noise $n(t)$.

the modulated signal, i.e. $y(t)$, which is also corrupted by the additive noise. The job of digital PSD is to estimate the signal changes in amplitude and phase, *i.e.* A and φ, that caused by the device under test (DUT). By taking advantage the correlation between $y(t)$ and the reference signals. The block diagram of digital PSD is schematically illustrated in Figure.1.

Thus, the received signal at the sampler is

$$y(t) = A\cos(2\pi f t + \varphi) + n(t) \tag{1}$$

which after sampling at rate $1/T$ corresponds to the sequence

$$y(kT) = A\cos(\frac{2\pi k}{N} + \varphi) + n(kT) \tag{2}$$

where $N = f_s/f$ is the sample number in a sine-wave signal period and f_s is the sampling frequency. For convenience, equation(2) can be formulated in a simple form

$$y(k) = A\cos(\frac{2\pi k}{N} + \varphi) + n(k) \tag{3}$$

The digital PSD is to find an estimate for A and φ by using a series of successive samples of $y(k)$ and the following equations,

$$U_c(k) = \frac{1}{N}\sum_{n=k-N}^{k} y(n)r_c(n) = A\cos\varphi \tag{4}$$

$$U_s(k) = \frac{1}{N}\sum_{n=k-N}^{k} y(n)r_s(n) = A\sin\varphi \tag{5}$$

$$\tag{6}$$

$$\begin{cases} A &= \sqrt{U_c^2 + U_s^2} \\ \varphi &= arctg(U_s/U_c) \end{cases} \tag{7}$$

It is usually assumed that the distribution of the measurement errors is Gaussian, or at least very close to Gaussian. If this assumption holds, the PSD is not only the best linear estimator, but also is the most efficient. In this paper, however, we want to investigate the case where the assumption fails. As non-Gaussian noise arises, this approach can place severe demands on the linear filters. This is especially true when impulse noises caused by electrostatic discharge . We here report nonlinear filtering method for digital PSD that is better in the sense that not only the Gaussian noise but also the non-Gaussian noise can be filtered.The basic concept of a linear filter is the separation of signals based on their non-overlapping frequency content. For nonlinear filters it is more convenient to consider separating signals based on whether they can be considered smooth or rough (noise-like).

2. Methodology

Median filter is a classical approach to remove impulsive noise. The standard median filter for digital signal processing was first suggested by Tukey [19]. Let $x(n)$ be the signal to be smoothed. Then the output of the standard median filter of length $2k+1$ with the center of the window at point n, is defined by

$$y(n) = MED[x(n-k), x(n-k+1), ..., x(n+k)] \tag{8}$$

where MED denotes the median.

Median filter can be used for removing impulses in an image without smearing the edge information. This is of significant importance in image processing. The success of median filters is based on two intrinsic properties: edge preservation and efficient noise attenuation with robustness against impulsive type noise. Neither property can be achieved by traditional linear filtering techniques without resorting to time-consuming manipulations. Median filter will not smear out sharp discontinuities in the data, as long as the duration of the discontinuity exceeds some critical duration. Thus the size of moving median filter which can be used is strictly dependent on the minimum duration of discontinuity which the user wishes to preserve. The feature is especially of importance for digital signal processing applications where sharp discontinuities caused by impulsive noise exist. In some cases, however, the durations of the samples contaminated by the impulsive noise are usually not identical. Thereby, the length of a moving median filter needs to be selected with care. In most cases, the general trend is that the longer the moving median, the more it smooth out. Unfortunately, the sine-wave signal does not belong to the class.

As illustrated in Fig.2, a median filter with window length of 3 slides sequentially over the signals and the mid-sample within the window is replaced by the median of all samples inside the filter window. Two samples contaminated by the impulsive noise are presented in the signal sequence. Instead of smoothing the impulsive noise, the median filter replaces the impulsive noise with its neighboring samples, as indicated in Fig.2(b). In addition, median filters also introduce a great deal of distortion by modifying the genuine signal samples that are mistaken for impulsive noise. For instances, some genuine samples, i.e. the maximum and minimum valued samples of the sine-wave signal and the samples that are next to impulsive noise, may be distorted by the median filter. It can be readily deduced that the longer of the median filter window, the more samples will be contaminated.

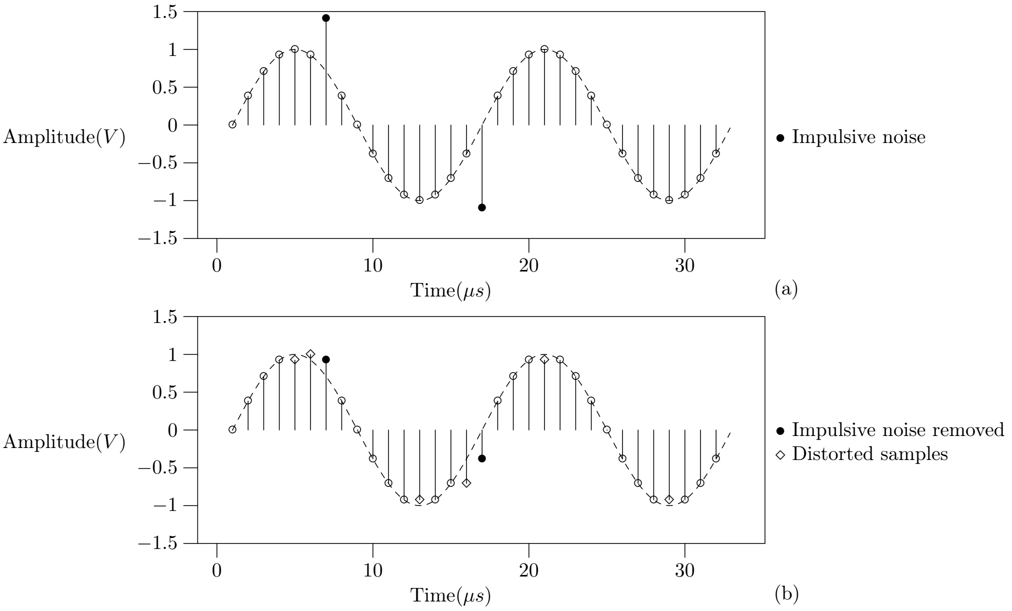


Figure 2. Input and output of a median filter. Note that in addition to suppressing the impulsive outlier, the filter also distorts some genuine signal components.

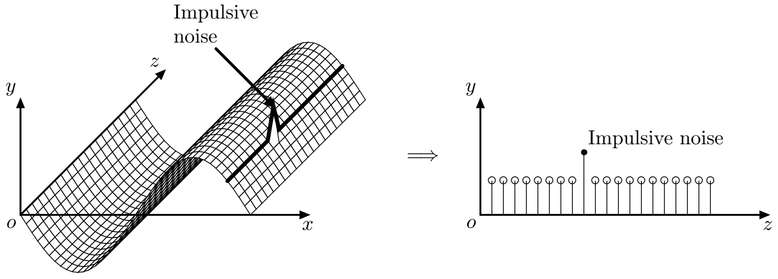


Figure 3. The sine-wave signals

In consideration of the aforementioned disadvantages of median filter, directly applying median filter in PSD, which takes sine-wave carrier as a necessity, is unable to produce high quality signal restoration. Even so, we managed to design an alternative method to apply a median filter to restore the sine-wave signals that are contaminated by impulsive noise. This method is based on a simple fact that the sine-wave signal exhibits significant periodicity even if it were seriously contaminated. By taking advantage of its periodicity, a series of repeated samples that occurred at an identical phase of each sine-wave period can be extracted from a sine-wave signal sequence. As illustrated in Fig.3, a number of consecutively periods of the sine-wave are arranged along a new axis in the new coordinate system, i.e. z-axis. The advantage is that the samples along z-axis are of identical phase in the sine-wave period, which results in a relatively flat signal, as shown in Fig.3. If the duration of the corrupted samples in the original signal does not exceed N samples, i.e. the duration of a complete sine-wave, only one corrupted signal will appear in the rearranged signal.

Assuming that

$$\mathbf{y} = [y(0), y(1), y(2), ..., y(n), ...], \quad n \in \mathbf{N}$$

are successive samples of the corrupted sine-wave signal.

The method is not intended to directly applying median filter to $\mathbf{y}$. The vector $\mathbf{y}$ is first formulated into matrix form,

$$\mathbf{Y} = \begin{bmatrix} y(0) & y(1) & \dots & y(N-1) \\ y(N) & y(N+1) & \dots & y(2N-1) \\ \vdots & \vdots & \vdots & \vdots \\ y(jN) & y(jN+1) & \dots & y((j+1)N-1) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} y_{j,k} \end{bmatrix}_{M\times N}$$
$$= \begin{bmatrix} \mathbf{Y}_0 & \mathbf{Y}_1 & \dots & \mathbf{Y}_{(N-1)} \end{bmatrix}, \quad j \in \mathbf{N} \tag{9}$$

where $y_{j,k} = y(jN + k)$ represents the element of matrix $\mathbf{Y}$, M denotes the number of row in the matrix and $\mathbf{Y}_k$ is the k^{th} column of the matrix $\mathbf{Y}$, $k = 0, 1, \dots, N-1$.

$$\mathbf{Y}_k = \begin{bmatrix} y(k) & y(N+k) & y(2N+k) & \dots & y(jN+k) & \dots \end{bmatrix}^T \tag{10}$$

By substituting equation(3), the $\mathbf{Y}_k$ can be formulated to

$$\mathbf{Y}_k = \begin{bmatrix} y_{0,k} \\ y_{1,k} \\ \vdots \\ y_{j,k} \\ \vdots \end{bmatrix} = A\cos(\frac{2\pi k}{N} + \varphi) \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} + \begin{bmatrix} n_{0,k} \\ n_{1,k} \\ \vdots \\ n_{j,k} \\ \vdots \end{bmatrix}, \quad j \in \mathbf{N} \tag{11}$$

The signal represented by $\mathbf{Y}_k$ is a relatively flat signal with offset of $A\cos(\frac{2\pi k}{N} + \varphi)$, which is contaminated by $n(k)$. There are two facts should be emphasised,

(i) The noise, either Gaussian or impulsive noise, is of independent and identical distribution (*i.i.d.*), and

(ii) the possibility of occurrence of N successive samples caused by impulsive noise is much less than that of 1 samples.

Therefore, it is more convenient to use a shorter median filter for impulsive noise removal along the columns. The filter output for $\mathbf{Y}_k$ can be described by

$$\hat{\mathbf{Y}}_k = MED(\mathbf{Y}_k) \tag{12}$$

Correspondingly,

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{\mathbf{Y}}_0 & \hat{\mathbf{Y}}_1 & \dots & \hat{\mathbf{Y}}_{(N-1)} \end{bmatrix} = \begin{bmatrix} \hat{y}_{j,k} \end{bmatrix}_{M\times N}, \quad j \in \mathbf{N} \tag{13}$$

The digital PSD can be described by the following matrix equation,

$$\begin{aligned}\mathbf{U} &= \hat{\mathbf{Y}} \cdot \mathbf{R} \\ &= \left[\ \hat{y}_{j,k}\ \right]_{M\times N} \cdot \left[\ r_k\ \right]_{N\times 1} = \left[\ u_j\ \right]_{M\times 1}\end{aligned} \tag{14}$$

where

$$r_k = \frac{1}{N} e^{i\frac{2\pi k}{N}}, \tag{15}$$

$$u_j = \sum_{k=0}^{N-1} \hat{y}_{j,k} \cdot r_k = \frac{1}{N} \sum_{k=0}^{N-1} \hat{y}_{j,k} \cdot e^{i\frac{2\pi k}{N}} \tag{16}$$

Finally, the demodulation results, *i.e.* A and φ, can be calculated by

$$\begin{cases} A &= |u_j| \\ \varphi &= arg(u_j) \end{cases} \tag{17}$$

3. Hardware implementation

The goal of digital PSD is to realize synchronous modulation and demodulation with the maximum usage of digital signal processing techniques, which has already been reported by many researches[xxx]. It should be noted that the proposed method requires much less logic resources than its linear counterparts, *i.e.* FIR filters. And most of the mainstream DSP and FPGA chips from the major producers can fulfill the tasks required by a digital PSD.

The FPGA based digital PSD itself is not a novel design and has already been employed for electrical capacitance tomography system[cite: a high performance digital for ect]. The system is implemented with a low cost FPGA, *i.e.* Xilinx Spartan3-400. The FPGA works at 50MHz and contains 16 embedded multipliers, which provide computational units for implementing the digital PSD and the filter operations. Since the FPGA has no on-chip nonvolatile memory, the program is placed in an external flash memory. The selected ADCs are of 14-bit, powered at 5.0V, support a maximum conversion rate of 10MS/s. The programmable gain amplifier (PGA) is used to extend the input signal range and acts as voltage buffer for the ADCs. The PGAs provide three control bits that are encoded to provide 8 gains/attenuations, i.e. -22, -16, -10, -4, 2, 8 and $14dB$. A direct digital synthesizer (DDS) IP core in FPGA generates sine-wave digital signal, which is then passed to a DAC (Analog Devices AD9754, 14-bit @ 105MS/s) to generate the excitation signal.

To implement the proposed method, a few functional blocks are added to the signal path prior to the PSD block, as illustrated in Figure.4. The acquired signal by ADC, *i.e.* $y(n)$, is multiplexed to three FIFO (first in, first out) units for data buffering. The timing sequence related to Figure.4 is illustrated in Figure.5. Each FIFO can store N samples, which correspond to a period of sine-wave signal, *i.e.* $T = N \cdot T_s$. The wr and rd signals are used to control the FIFO data in and out. Actually, the proposed method is implemented in a pipeline mode, which can be divided into three steps, *i.e.*

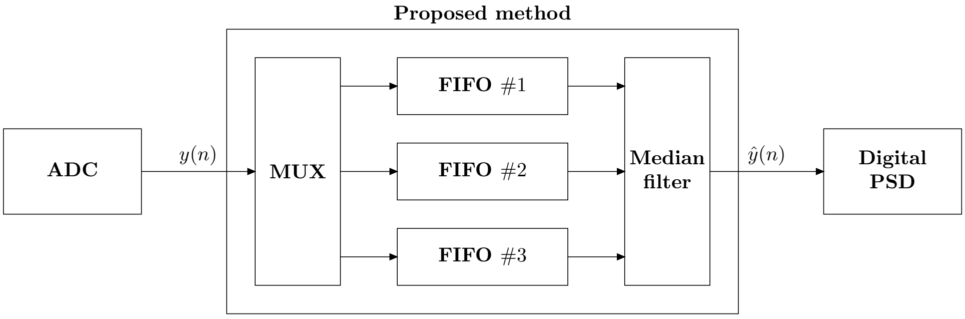


Figure 4. The diagram of modified median filter in digital PSD.

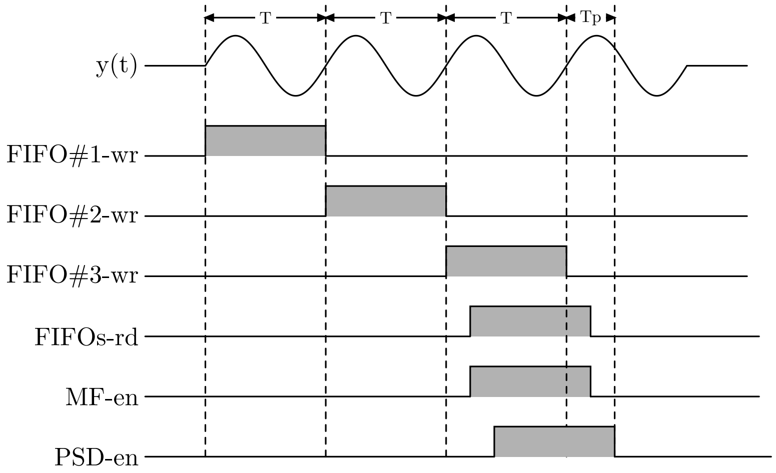


Figure 5. The timing sequence of the FPGA based implementation.

(i) Assert the FIFO *wr* signal one after another in a numerical order in order to fill the FIFOs with the ADC samples.

(ii) Once FIFO #3 begins to stream in, assert the FIFOs-*rd* and MF-*en* to start the median filter, which will enable the median filter to read one samples from each FIFO at a time and produces one median, *i.e.* $\hat{y}(n)$.

(iii) Assert the PSD-*en* signal to start the demodulation process once the filter output is ready.

Note that it is not necessarily to wait for all FIFOs are filled with samples to start the median filter. Also the PSD block can begin the demodulation before the median filter completes all the processes. One can even start the step (i) once FIFO #1 is no longer filled with samples. Actually, T_p is as short as $5T_{clk}$, where $T_{clk} = 1/50MHz = 20ns$ is the FPGA clock period. In other words, the FPGA will produce the final demodulation result within $100ns$ once the ADC sampling process completes. As compared with the sine-wave period $T = 10\mu s$, T_p is much smaller and can be readily neglected in calculating the process time. This is benefited from the pipeline design, in which one can take advantage of the parallel processing capabilities of the FPGA to increase the efficiency of sequential code.

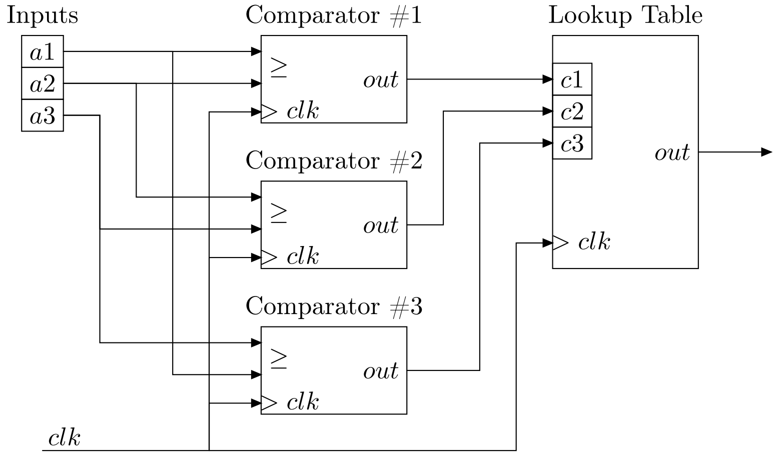


Figure 6. The median filter.

$c1$	$c2$	$c3$	out
0	0	0	$error$
0	0	1	$a2$
0	1	0	$a1$
0	1	1	$a3$
1	0	0	$a3$
1	0	1	$a1$
1	1	0	$a2$
1	1	1	$a1 = a2 = a3$

Table 1. The truth table of the lookup table.

The median filter is implemented with FPGA logics, as schematically shown in Figure.6. The comparators are to find whether one input is greater than the other one. Taking the comparator #1 for example, it will output

$$c1 = \begin{cases} 1 & \text{if } (a1 \geq a2) \\ 0 & \text{if } (a1 < a2) \end{cases} \tag{18}$$

By pairwise comparisons, the comparators produce three signals, *i.e.* $c1$, $c2$ and $c3$. And these signals are used as input to a lookup table, whose outputs are determined by Table.1. It spends $2T_{clk}$ to generate a median filter results.

4. Results and Discussions

To assess the performance of the proposed method in removing impulsive noise, we compared the effect of the modified median filter against its linear counterpart, *i.e.* FIR filter, by evaluating the signal-to-noise ratio (SNR) and equivalent noise bandwidth (ENBW). In particular, we got the original signal with additive white gaussian noise and impulsive noise of different powers to test the method by numerical simulation. Also the proposed hardware system based on FPGA is also tested.

4.1. Numerical simulations

In the numerical simulation, the signal model described by equation(1) is employed. We use power in dBW to measure the noise powers, *i.e.*

$$P_{wn}(dBW) \;= 10\lg\frac{\bar{P}_{wn}}{1W}(dBW) \tag{19}$$

$$P_{imn}(dBW) = 10\lg\frac{\bar{P}_{imn}}{1W}(dBW) \tag{20}$$

where, $\bar{P}_{imn}$ and $\bar{P}_{wn}$ denote the average power of the impulsive noise and the white noise, respectively.

The signal power is also measured in the same way,

$$P_{sig}(dBW) = 10\lg\frac{\bar{P}_{sig}}{1W}(dBW) \tag{21}$$

For simplicity, the amplitude of the sine-wave signal in equation(1) is set to $A = 1$, which will be taken as the true value in the following analysis.

The SNR of the PSD output is used to evaluated the data quality,

$$SNR = 10\lg\frac{\sum_{i=1}^{N}[d(i)]^2}{\sum_{i=1}^{N}[d(i)-\hat{d}]^2} \tag{22}$$

where $d(i)$ is the i^{th} data in a length N data set that obtained consecutively from the PSD, and $\hat{d}$ is the true value of the measured parameter.

In this research, the white noise is treated as a zero-mean normal distribution model. And the impulsive noise is simulated by a Bernoulli-Gaussian model, in which the random time of occurrence of the impulsive noise is modelled by a binary Bernoulli process $n_b(m)$ and the amplitude of the impulses is modelled by a Gaussian process $n_g(m)$. The probability mass function is given by

$$P_B[n_b(m)] = \begin{cases} \alpha & \text{for } n_b(m) = 1 \\ 1-\alpha & \text{for } n_b(m) = 0 \end{cases} \tag{23}$$

where α is the probability when $n_b(m) = 1$. A zero-mean Gaussian probability density function (pdf) of the amplitude of impulsive noise is given by

$$f_N[n_g(m)] = \frac{1}{\sqrt{2\pi\sigma^2}}\exp[-\frac{n_g^2(m)}{2\sigma^2}] \tag{24}$$

where σ^2 is the variance of the noise amplitude. In a Bernoulli-Gaussian model the pdf of an impulsive noise is given by

$$f_N^{BG}[n_i(m)] = (1-\alpha)\delta[n_i(m)] + \alpha f_N[n_i(m)] \tag{25}$$

where $\delta[n_i(m)]$ is the Kronecker delta function.

To assess the performance of the proposed method (PSD+MF), we compared it against other two methods, *i.e.*

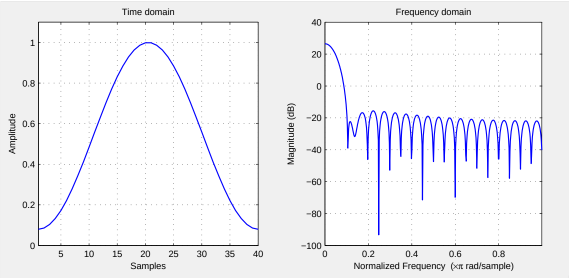


Figure 7. The impulse response of the FIR filter.

- PSD: the standard PSD method that is designed with no pre-filter ahead.
- PSD+LPF: the PSD with low-pass filter, in which a 40^{th} order FIR low-pass filter with a cut-off frequency of $300kHz$ is employed. Its characteristics are illustrated in Figure.7.

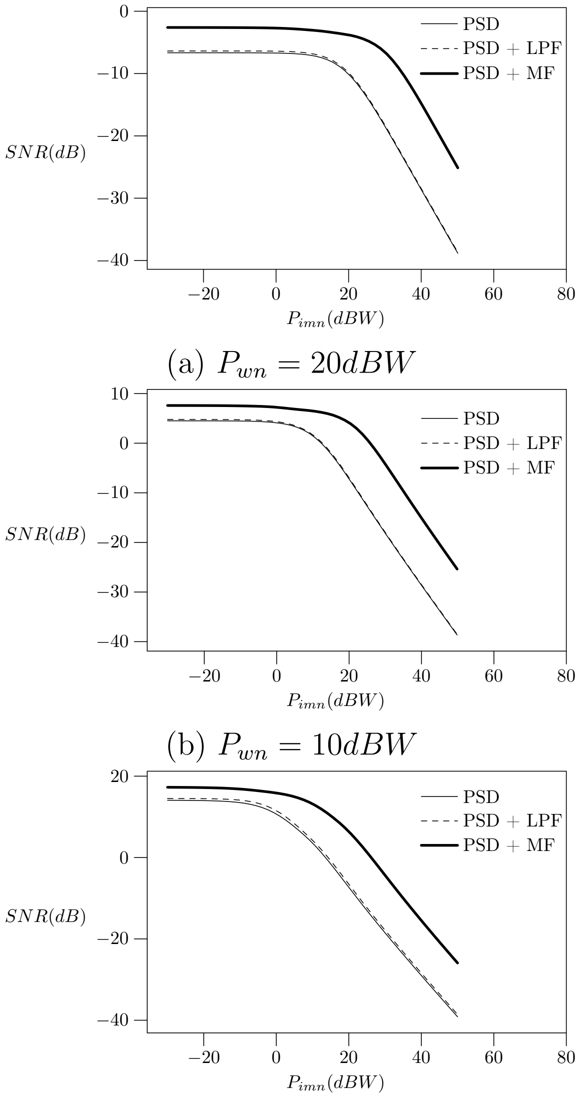
The SNR results that calculated using equation(22) are illustrated in Figure.8, in which P_{imn} varies between -30 and $50dBW$ and the P_{wn} is set to 0, 10 and $20dB$, respectively.It can be found that the trends of the curves in the three figures are similar. The curves roughly keep flat at beginning until they start to decrease at a fixed slope. The curves can be divided into two parts by the turning point, which is actually the balancing point of the white noise and impulsive noise, *i.e.* $P_{imn} = P_{wn}$. The white noise plays a dominant role in the first part. The curve of the proposed method is usually $\sim 3dB$ higher than that of the other methods, while it is difficult to tell the differences between the two lower curves. In consideration of the minor SNR improvement of the PSD+LPF method, it can be said that it is not worthy of the system complexity that added by the FIR filter. In the latter part, the P_{imn} is greater than P_{wn}, which means that the impulsive noise becomes the dominant part in the noise. As P_{imn} increases, the SNR curves begin to fall accordingly at a slope of approximately $-1dB/dBW$. At the same time, however, the advantage of the proposed method in improving SNR becomes significant, *i.e.* from $\sim 3dB$ to $\sim 13dB$.

It is interested that the low-pass filter has few contribution to enhancing the SNR of the PSD. It may be argued that the employed low-pass filter is not an optimal one. The equivalent noise bandwidth (ENBW) of the PSD and its low-pass filter should be discussed to fully understand the role of the low-pass filter in the PSD algorithm. The ENBW of a linear system, *i.e.* $H(f)$, is the width of a rectangle whose area contains the same total white noise power as $H(f)$, which can be described by

$$\int_0^\infty |H(f)|^2 df = |H_m(f)|^2 \cdot \Delta f \tag{26}$$

where, $H_m(f)$ is the maximum value of $H(f)$ and

$$\Delta f = \int_0^\infty \frac{|H(f)|^2}{|H_m(f)|^2} df \tag{27}$$



(a) $P_{wn} = 20dBW$

(b) $P_{wn} = 10dBW$

(c) $P_{wn} = 0dBW$

Figure 8. SNR_{psd} v.s. P_{imn}.

is the ENBW of the linear system. The ENBW of a digital PSD is given by

$$\Delta f_{PSD} = \frac{1}{2T_a} = \frac{1}{2N_aT_s} \tag{28}$$

where N_a is the number of the samples used for generating one demodulation result and $T_s = 1/f_s$ is the sampling time. Theoretically, Δf_{PSD} achieves its the maximum value of $f/2$ when only one sine-wave period is used for demodulation, *i.e.* $T_a = 1/f$ and it can be further reduced by extending T_a. On the other hand, the ENBW of a low-pass filter is greater than its $-3dB$ cut-off frequency, *i.e.* $\Delta f_{LPF} > f_c$, while $f_c > f$. Finally, we get a concise result

$$\Delta f_{PSD} < \Delta f_{LPF} \tag{29}$$

It can be concluded that the PSD always has a narrower ENBW than its pre-filter. And this is valid for any low-pass filter in a PSD system. Therefore, once the white noise

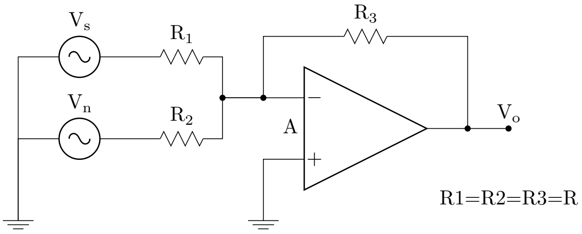


Figure 9. The experimental circuit.

falls into the ENBW of the PSD, it is necessarily within the ENBW of the pre-filter and will not be attenuated by the filter. And this explains why the low-pass filter cannot help enhance the SNR of the PSD.

Theoretically, if the white noise was assumed, the SNR improvement of the digital PSD can be obtained by

$$\Delta SNR = \frac{SNR_{out}}{SNR_{in}} = 10\lg\frac{f_s/2}{\Delta f_{PSD}} = 10\lg(N_a) \tag{30}$$

where $f_s/2$ is the Nyquist frequency. Ideally, the system SNR can be enhanced by increasing the number of the samples N_a as long as the measured parameter remains unchanged.

4.2. Experiments

In the experiments, the amplitude and frequency of the sine-wave signal are $2V_{p-p}$ and $100kHz$, respectively. The ADC input range and sampling frequency are $5V$ and $10MHz$, respectively. Some signal generators now provide such function as arbitrary waveform generator, which can store user-defined arbitrary waveform in its nonvolatile memory. Thereby, we may take advantage of it to generate the impulsive noise with adjustable amplitude. The signal and the impulsive noise are mixed together with the electronic circuit in Figure.9.

The output signal V_o can be obtained by

$$V_o = -(V_s + V_n) \tag{31}$$

In addition, it is assumed that the white noise is already presented in the circuit. In the experiments, each demodulation result is calculated using 100 samples, which is same as that of the numerical simulation. The SNRs are obtained for noise amplitudes ($|V_n|$) are 0.25, 0.5, 1, 2 and 4V, respectively. According to the results listed in Table.2, the proposed method can achieve better SNR in the experiments as well as in the numerical simulation. The improvement on SNR can be as high as $10dB$.

In addition, it is also found that the SNR data are close when $|V_n| = 2$ and 4. Note that the input range of ADC is 5V. To avoid saturation characteristic and possible damage to the ADC, the ADC input incorporate a voltage clamp to ensure the ADC input will always be within a range between -2.5V and +2.5V. Even though the additive

$\|V_n\|$(V)	SNR(dB)		
	PSD	PSD+LPF	PSD+MF
0.25	34.4	34.0	41.2
0.5	29.5	29.8	38.9
1	24.6	25.2	35.7
2	21.5	22.1	32.2
4	21.4	22.2	31.4

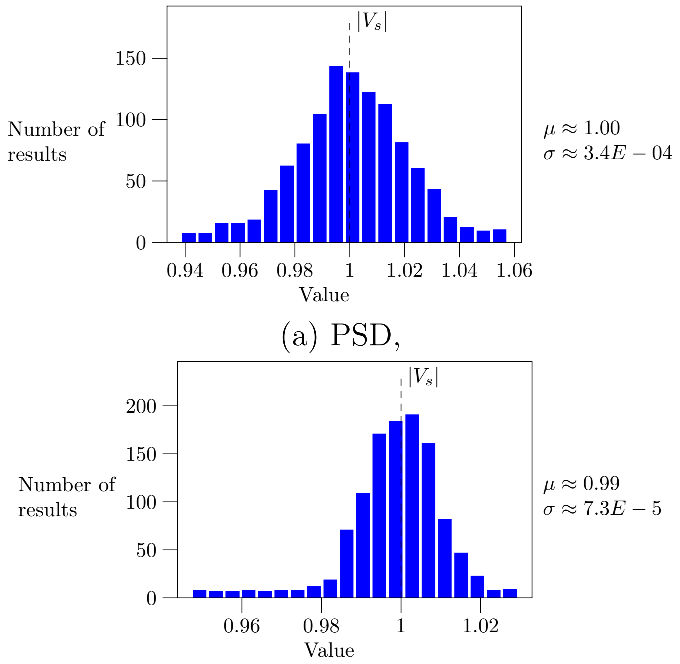
Table 2. The SNRs obtained in the experiments.

noise is large, it will be saturated by the clamping diodes and its power will be limited, which will affect the experimental results to some extent.

It may be argued that the impulsive noise might be further reduced by limiting the ADC input range since the voltage saturation could reduce the noise power. In fact, this is not a good idea. The distribution of the results for $|V_n| = 0.25$ and 4 are illustrated in Figure.10 and 11, respectively. It can be deduced that $|V_o|$ are much less likely to be greater than 2.5V when $|V_n|$ is smaller. As can be seen in Figure.10, $|V_n| = 0.25V$. The distributions for both methods are normal distributions with mean values very close to $|V_s|$. In contrast, as the datasets are not symmetrical shape in Figure.11 in which $|V_n| = 4V$, we can clearly rule out that a normal distribution model would be a suitable choice. In addition, we can find that the possibility of $V > |V_s|$ is much less than that of $V \leq |V_s|$, which results in the lower mean values. This can be explained by the voltage saturation. Thereby, it can be concluded that the voltage saturation will make the mean value of the result a bias estimation to the measured parameter, which will affect the measurement accuracy.

5. Conclusions

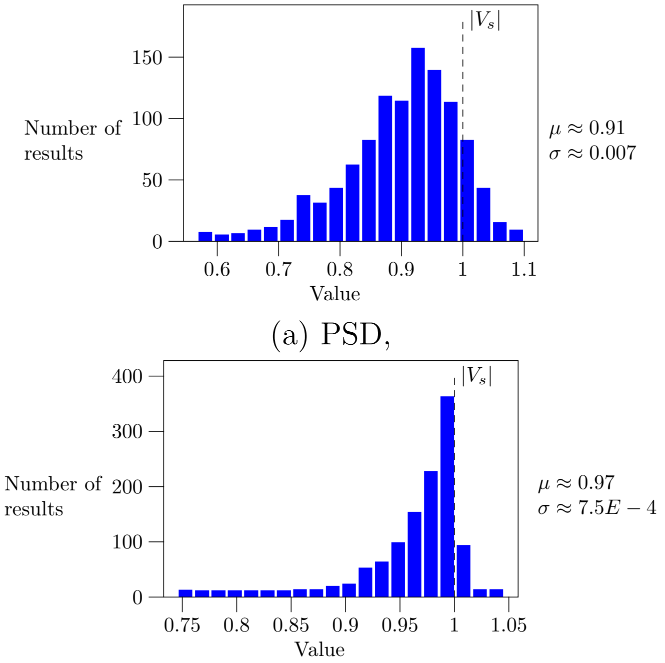
By carefully examining how the impulsive noise affects the PSD performance we were able to devise a more efficient filtering scheme, capable of increasing both the PSD stability and accuracy. In particular, we achieved a SNR improvement of $\sim 10dB$ as compared with that of the standard PSD method. This method was eventually verified in an electrical impedance measurement system, where the FPGA based hardware implementation is simple, requiring only a modified median filter to be added ahead of the digital PDS block. However, this scheme is not limited to technique, and can be used whenever a periodical signal is measured with a digital PSD. Moreover, the method we provided can be readily used as a reference where periodical signal is presented. Our results are particularly suited for situations where a better SNR would be desirable while the impulsive noise dominates the noise.



(a) PSD,

(b) PSD+MF

Figure 10. The result distributions, $|V_n| = 0.25V$.



(a) PSD,

(b) PSD+MF

Figure 11. The result distributions, $|V_n| = 4V$.

Acknowledgments

The authors would like to thank the financial support from Natural Science Foundation of China (Grant Nos. XXXXXX). Financial supports from China Scholarship Council are gratefully acknowledged. This work was completed while Dr. Ziqiang Cui was a Visiting Academic at SISP laboratory under Professor A. J. Peyton's supervision in the University of Manchester.

References