Software Defined Radio and its application in Schottky analysis

Qian Wang
Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, China,
wangqian2016@impcas.ac.cn

Applying software-defined radio (SDR) instead of analog components brings more convenience for radio communication and it is of widespread use nowadays. Based on this methods, two Schottky noise-like signals with different initial time were produced and measurement of their phase difference was done. The information revealed in phase difference will be applied to the future storage rings with several Schottky detectors.

1 Introduction

1.1 Why to use software-defined radio?

Software defined radio (SDR) is radio system which is implemented as a code on a computer or embedded system instead of hardware components for processing the signal in radio communication.

Thanks to the minimal use of analog elements, it becomes more flexible to meet the requirement of different situations, since changing only parameters for the software is needed. With capabilities of digital electronics, many processing steps, which can only be possible in theory, can be provided for mathematical operation on the digitalized signal. As an example, the spectrum analyser in Fig. 1, the hardware analog elements are all fixed with unadjustable parameters, while the SDR has flexible blocks to deal with different math problems.

1.2 The previous work

In 2013 a feasibility test about the application of the software-defined radio in a storage ring has been done [2]. To prove this concept, a signal has been transferred into the computer, then analysed by GNU radio. It showed that the spectrum analyser in DAQ system could be in principle replaced by a HQ soundcard for Schottky detecting system at the storage ring ESR in GSI. A high frequency dedicated board and high sensitivity of soundcard would allow for an alternative high-precision spectrum analyser.

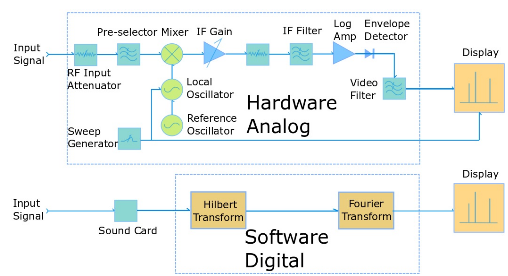

Fig. 1: *Spectrum analyser: Hardware vs SDR [1]*

1.3 Tool we use

Like in the previous work, we continue using GNU Radio, which serves as the software part to realize all needed functions.

GNU Radio — an open source toolkit accompanied with a GUI companion, which works in OSX, Windows and Linux. It supports signal source from SDR preprocessing devices, signal data from the files and even itself. It is user-friendly by just connecting the blocks for signal processing in the GUI.

In the previous work, the test was successfully done on the dedicated inexpensive hardware. But there are also more professional and advanced commercial SDR devices of high frequency and precision, such as the following SDR, which is planned for the FAIR control system.

ETTUS research B210 — a fully integrated, two-channel Universal Software Radio Peripheral (USRP) board with continuous frequency coverage from 70 MHz to 6 GHz. It combines full duplex, MIMO (2 Tx & 2 Rx) operation with up to 56 MHz of real-time bandwidth, fast and convenient USB 3.0 connectivity and open and reconfigurable Spartan 6 XC6SLX160 FPGA. Full support for USRP Hardware Driver (UHD) software insures to easily work with GNU Radio and OpenBTS.

1.4 Prospect for future Schottky detectors

There are many Schottky detectors in both of the two new storage rings HESR and CR, which will be built at FAIR. It is necessary to compare results from different detectors in one ring. Using software-defined radio to get real-time spectrum information is of viability and versatility.

2 Simulation

2.1 Signal from one particle in Schottky detection

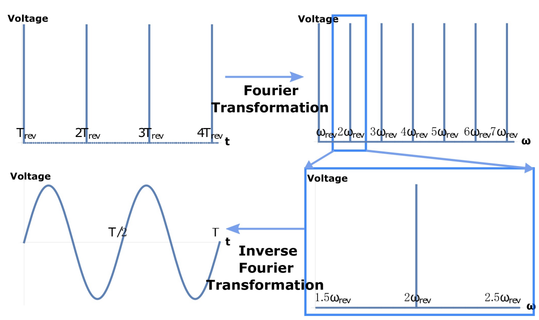

Fig. 2: *Impulse train: filtering one special harmonic from the resulting comb spectrum in frequency domain then transforming it to get a simple sine wave for one particle*

The Schottky detectors, both pick-up and resonator, are sensitive to charged particles. For one charged particle circulating in the storage ring, it will be recorded by detection system as a pulse of voltage when it approaches the detector. Finally the signal will form a infinite impulse train (Eq. 1) and a corresponding comb spectrum in the frequency domain, if the particle in ring exists for long enough time [3].

$$p(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT_{rev}) \quad (1)$$

In the frequency domain, the frequencies distribute only at the points of $\omega = n\omega_{rev}$ ($\omega_{rev} = \frac{2\pi}{T_{rev}}$, n is integer) with equal value (Eq. 2).

$$p(\omega) = F[p(t)] = \omega_{rev} \sum_{n=-\infty}^{+\infty} \delta(\omega - n\omega_{rev}) \quad (2)$$

After filtering one harmonic from the repeated frequency ranges, a resulting sine wave of this particle is formed (Fig. 2).

2.2 The pre-analysis of the two signal experiment

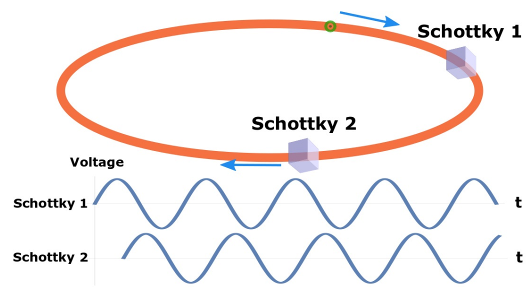

Fig. 3: *Two Schottky detectors in the storage ring*

Focusing on one particle's circulatory movement in the storage ring, if we install another Schottky detector in the ring (Fig. 3), the only difference between the signal from two detectors of this particle will be the time it needs to reach the detectors, in other words, the time delay. Meanwhile, it leads a phase rotation in frequency domain [4], as can be seen in Table. 1.

	Time domain	Freq. domain
Original	$f(t)$	$F(\omega)$
Time delay	$f(t - t_0)$	$F(\omega)e^{i\omega t_0}$

Tab. 1: *Time delay in time domain equals phase swift in frequency domain*

2.3 Simulation for the phase shift

To get a better understanding of the effect of time delay on phase shift ϕ, we need to simulate the single frequency sine wave phase shift spectrum at first.

The Fourier transformation of a sine wave is

$$F(\omega, \phi) = (i\pi\delta(\omega + \omega_0) - i\pi\delta(\omega - \omega_0))e^{i\phi} \quad (3)$$

The phase spectrum for that is

$$Arg(\omega, \phi) = (\frac{\pi}{2} - \phi)(\delta(\omega + \omega_0) - \delta(\omega - \omega_0)) \quad (4)$$

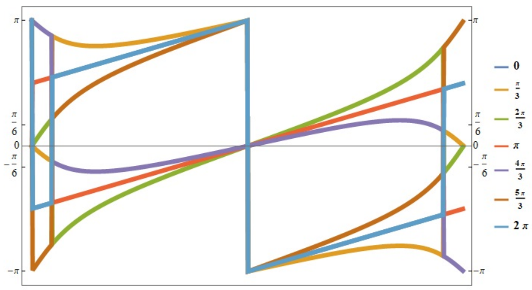

Fig. 4: *Phase spectrum: a simple sine wave with $\frac{\pi}{3}$ phase shift at each revolution*

The result in Fig. 4 revealed significant variations in one revolution of the phase rotation.

3 Measurement

3.1 Signal in real world and communication devices

While the input signal to any communication system is by nature real, many devices for communication today prefer complex signals (IQ signal, I for In-phase, Q for Quadrature) to get clearer information of both amplitude and phase. Therefore, we chose the IQ signals for the Schottky detection.

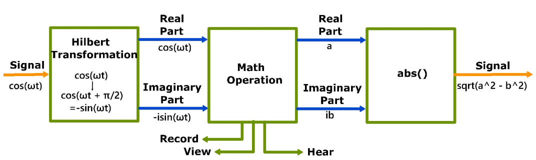

Fig. 5: *Signal processing: receiving a signal from real world into the communication device, doing some math operations and sending back*

Considering the definition of the IQ signal [5], the natural way to complete the full complex information of the original real signal without changing its spectrum is to get the signal phase shift by 90° in the imaginary part. This approach is known as Hilbert transformation.

If you want to get the IQ signal back to the real world, to get the power of the signal by absolute operation is one simple way (Fig. 5).

3.2 The phase rotator

We constructed a signal analysis flow (Fig. 6 top) to measure the phase shift between two sine waves, one was the reference while the other was modified with a phase rotator. The incoming

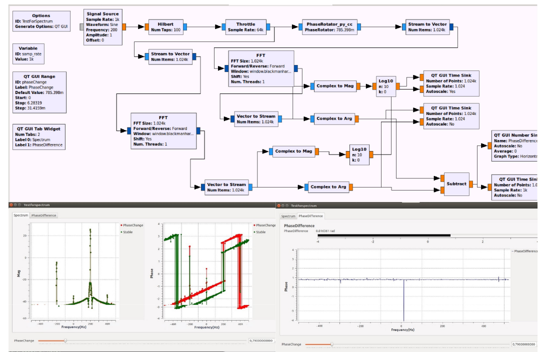

Fig. 6: *Sine wave rotation set-up & result*

signal was real to imitate the signal from Schottky. Then it was transformed with a Hilbert filter to a complex signal to get ready for the math operation in the IQ domain. Next, applying the block of a phase rotator, implemented in python within the scope of this work, to one channel signal was needed. This rotator has a fixed complex input and output, multiplying $e^{i\phi}$ to get ϕ phase shift. It also has getter and setter functions for the dynamic action in GUI. All results, both the amplitude and phase spectrum (Fig. 6 bottom), were presented in GUI, after the final FFT processing. Dragging the slider for phase rotator, the phase spectrum went up or down with phase increasing or decreasing. The value for phase changing was the same as the phase difference presented on the other tab page.

3.3 Test for the two impulse trains

For Schottky detector, the impulse train comes first as mentioned above. First, we measured the phase rotation for one impulse train to get a general idea of its phase shift spectrum. The set-up blocks (Fig. 7 top) was similar to the sine wave test with only the source changing. Since no ready-made impulse train signal source was available in GNU Radio, we applied two equal square waves with one sample delay between them. By taking only the non-negative part of the subtracted result, the impulse train with the same frequency of the original square wave was formed. The results (Fig. 7 bottom) was in accordance with expectation. Only the phase spectrum changed while the amplitude spectrum was unaffected as phase changed. The output of the phase shift was the same as rotation input.

Next, we consider the delay situation. For the set-up (Fig. 8 top), only change was a sam-

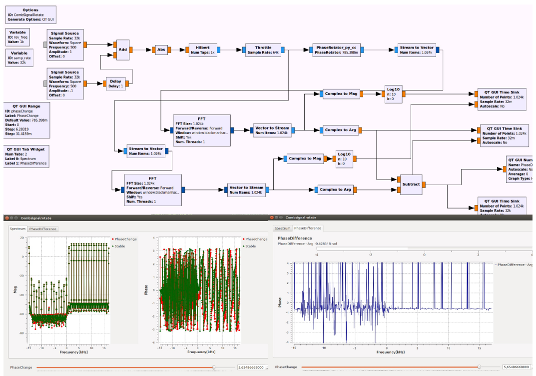

Fig. 7: *Impulse train rotation set-up & result*

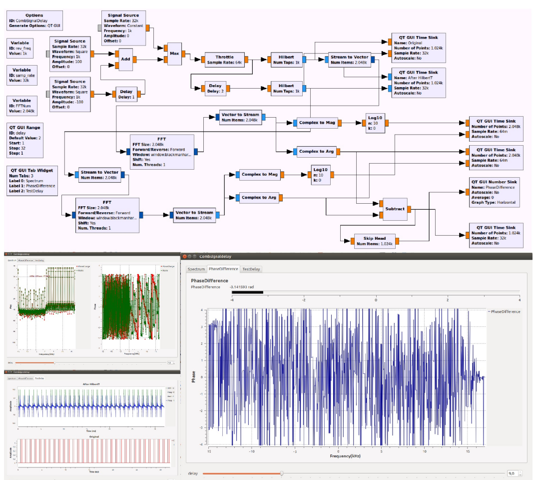

Fig. 8: *Impulse train Delay Set-Up & Result*

ple delay block was added before the Hilbert filter. To get a more distinct look to the results (Fig. 8 bottom), another page for the time delay monitor in GUI added. Though the value of the phase shift changed as the delay sample number changed. The output value displayed were only $-\pi$, π, or 0. Since Mathematica uses different algorithm libraries with C++/python (the GNU Radio uses) and it is good at fast symbolic computation in GUI. We rebuilt the same processing using Mathematica to investigate this result (Fig. 9). Unfortunately, considering the effect arising from noise, the results were exactly the same as the former ones. There existed some choppy phase shift points along with the smooth change of sample delay, which may be a special character of the impulse train in my case (sample rate $32kHz$, signal frequency $1kHz$). A further research on this change pattern needs to be done.

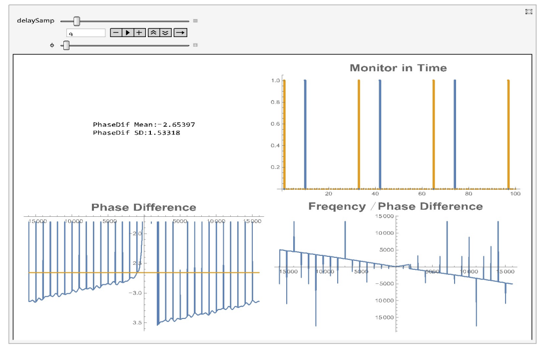

Fig. 9: *Investigating strange phase difference output*

4 Summary

Using GNU Radio, the measurement of phase rotation for both single sine wave and impulse trains has been done. The sample delay for impulse train test contributed to changing the phase. But the pattern for this changing was unclear. Further investigation is needed. Using the phase information from the multiple Schottky detectors in the same storage ring may enable deeper insight and more precise in-ring measurements.

Acknowledgments

First of all, I would like to thank the organisers, especially Jörn, to make this memorable time at GSI. Though I had a bad bike accident, comfort and relief from GSI members made me feel at home. For my project, my appreciation goes to Shahab for his supervision. My further gratitude is to Yuri and the rest of the SPARC Detectors group for warmly accepting me.

References

[1] *Spectrum Analysis Basics*, **Application Note 150** (Agilent Technologies, 2006).

[2] M. Gluza *International Summer Student Program Book of Reports*, **Part IV** 107 (GSI, Darmstadt, 2013).

[3] F. Caspers, J. Tan. *Schottky Tutorial* (CERN, Geneva, 2008).

[4] A. V.Oppenheim, et al. *Discrete Time Signal Processing* (PHEI, Beijing, 2015).

[5] NI - http://www.ni.com/tutorial/4805/en/