CW 19/20 summary

Alexander Pastor

08.05.2017

Contents

1 Understanding RF Specifications

1.1 Basic Terminology

digitizer amplitude error
The following formula provides the dampening or attenuation factor E of the digitizer (where 1 on the E-axis means 0 amplitude):

$$E = 1 - \frac{R}{\sqrt{1+R^2}} \tag{1}$$

A X-Hz digitizer is defined to have $E = \frac{1}{\sqrt{2}}$ at the frequency X, which implies $R = 1$ for the frequency X. X is called bandwidth of the digitizer in this context, and R is the ratio of the digitizer bandwidth and the maximum frequency of interest $\frac{f_d}{f_i}$.

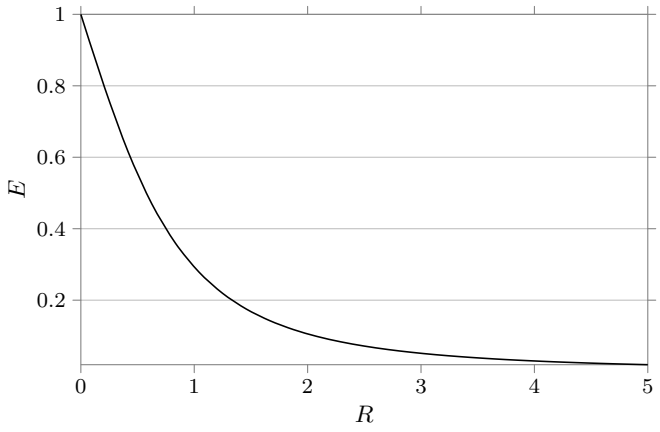

Figure 1: Digitization error verus bandwidth ratio

It is recommended by NI to have X be 3 to 5 times higher than the frequency of interest. This corresponds to errors or dampening between 1.94% and 5.13%

rise time
Rise time is defined as the time a signal needs to rise from 10% to 90% of its steady-state or periodic maximum.

- The rise time of a simple RC-circuit is about $\frac{0.35}{RC}$.
- The formula to calculate the total rise time of a digitized signal is:

$$T_{r_t} = \sqrt{{T_{r_s}}^2 + {T_{r_d}}^2}$$

- In order to minimize rise time errors NI recommends to have T_{r_d} be around $\frac{1}{3}$ and $\frac{1}{5}$ of T_{r_s}.

Nyquist Theorem
The bandwidth of the digitizer must be at least 2 times the maximum frequency of the signal to avoid aliasing.

$\Leftrightarrow$ To extinguish aliasing in the passband one either has to make sure the Nyquist Theorem is matched or apply a lowpass filter to limit the signal's bandwidth.

phase noise
Phase noise is the frequency domain representation of time domain jitter. Jitter in turn is the first derivative of a delay. Phase noise is usually close to the carrier and thus measured in dBc (decibels relative to the carrier).

spectral density
Measures the power per frequency in a frequency domain [W/Hz].

resolution bandwidth
The resolution bandwidth equals the FFT bin size and is smallest resolvable frequency. It is inversely proportional to the number of samples.

noise density
Noise level below which signal cannot be detected. One has to distinguish between broadband noise density of a sinusoidal and random (perhaps AWGN) noise. In the latter case one can use a narrow band filter to improve our signal.

dynamic range
Ratio between the highest and lowest signal level a circuit can take. Spurious dynamic range is the ratio between the highest and lowest signal level, without any unwanted sinusoidal frequencies. It is usually expressed in dB.

voltage standing wave ratio (VSWR)
The VSWR is the reflected-to-transmitted wave ratio. Self-evidently, the smaller, the better.

$$VSWR = \frac{1+p}{1-p}$$

where p is the reflection coefficient.

Note that is also possible to represent reflections as return loss (in dB):

$$
\begin{aligned}
RL &= 10 \log \left(\frac{P_{\text{in}}}{P_{\text{reflected}}} \right) \\
&= 20 \log \left(\frac{V_{\text{in}}}{V_{\text{reflected}}} \right) \\
&= -20 \log \left(\frac{V_{\text{reflected}}}{V_{\text{in}}} \right) \\
&= -20 \log \left(|p| \right)
\end{aligned}
$$

frequency response
Just as a reminder: the frequency response is the fourier transform of the impulse response $h(t)$.

$$
H(f) = \int_{-\infty}^{\infty} h(\tau) \cdot \exp(-j2\pi ft)\, d\tau
$$

modulation error ratio (MER)
A measure of SNR for digitally modulated signals, also expressed in dB.

$$
MER = \frac{\sum_{j=1}^{N} \tilde{I}_j^2 + \tilde{Q}_j^2}{\sum_{j=1}^{N} (\tilde{I}_j - I_j)^2 + (\tilde{Q}_j - Q_j)^2}
$$

Where the components with $\sim$ are the ideal and those without are the actually received components of the N symbols of the signal.

IQ procedure
In-phase and quadrature demodulation procedure.

error vector magnitude (EVM)
Quantifies demodulation performance under the influence of noise. The EVM is the normalized root-mean-square of the difference of ideal and measured location over all symbols. It is normalized by the maximum length of an ideal symbol to make the EVM's value comparable under different system gain conditions.

$$
EVM = \frac{\sqrt{\frac{1}{N} \sum_{j=1}^{N} (\tilde{I}_j - I_j)^2 + (\tilde{Q}_j - Q_j)^2}}{|v_{\max}|}
$$

third-order intercept (TOI)
Theoretical number. TOI is the power level when the third-order harmonic distortion has the same power level as the fundamental tone. The bigger the TOI, the better since it is a measure for the maximum RF-circuit power.

2 Physical Layer Challenge

2.1 Basic Terminology

2.1.1 FIR Filters

passband, transitionband, stopband
The passband is the frequency band that is not attenuated band of a filter, i.e. that band of allowed frequencies. The stopband is the band that a filter stops, or attenuates strong enough, so that the signal amplitude for those frequencies is below the stopband threshold. The transitionband is an attenuated band between passband and stopband.

baseband
The baseband, which has to be distinguished from the passband is a band ranging from 0 Hz to some higher cut-off frequency.

filter tap
A filter tap is a (coefficient, delay) pair. The number of taps is often denoted as N. This number is a good mesaure for the amount of filtering, the required space and the amount of calculation required by the filter.

2.1.2 Frame Synchronization

frame preamble and start frame delimiter (SFD)
Aliases: syncword or sync sequence framing.
Repetitive pattern of all the same nibbles (and thus bytes). Marks an Ethernet frame beginning. After the preamble the one-byte *start frame delimiter* is located to mark the beginning of real data. This mechanism is employed to achieve a self-clocking protocol. The preamble is needed to resynchronize the receiver once a bit slip occurs.

CRC-based framing
[deferred] CRC-based framing is used in ATM protocols by reusing the header CRC. How exactly does this work? Also, I wondered since 802.11 and Ethernet both also have a FCS (CRC) at the end of their frames, do they use some kind of CRC-based / syncword-based framing at the same time? As of now I think it is as follows: 802.11 and Ethernet frame synchronization is syncword-based and CRC is employed for error correction only.

self-clocking signal
Signal that is synchronized without a separate clock signal. Receiver synchronization information is contained in the signal (preamble+SFD).

bit slip
A lost bit or an extra bit in the signal.

[question] Why aren't Ethernet (and 802.11) using byte- or bitstuffing methods for frame synchronization? I thought maybe bit errors could corrupt the whole "synchronization complex", but one lost frame shouldn't be too bad? I guess that a big 56-bit preamble and 8-bit SFD might just be more robust against bit errors?

2.1.3 Flowchart Components

Attack and Decay Time
The attack time and decay times are similar concepts to the rise and fall time of a signal. The attack time is a measure of how fast the AGC reacts to rising signal amplitude, while decay time is the adaption time when the signal amplitude is falling.

Source: Electronics Stackexchange thread

2.2 FIR Filters vs. IIR Filters

A short list of advantages and disadvantages of FIR filters compared to IIR filters.

- FIR filter are less susceptible to quantization errors than IIR filters.
- Linear-phase filtering can be implemented with FIR filters.
- FIR filters are always stable.
- FIR filters are slower than IIR filters.
- FIR filters require more memory and computation time on the hardware, because they tend to need more taps to implement the same functionality as IIR filters.

2.3 iNets PHY Layer

I received GRC flowcharts of a wireless transceiver and a test application.

The transceiver consists of a transmitter and a receiver. The transceiver has a UDP server as input for the transmitter and a USRP source hardware device as input for the receiver.

Let me now describe how the two components work (let's think inside the box):

2.3.1 The Transmitter

- The data from the UDP source is first packeted (optionally with CRC checksum).

- Then, the PDU is converted into a tagged stream, because GR does not have a "natural" packet concept, instead packets are delimited by tags as described later.
- The bits are repacked and then converted into symbols.
- The symbols are then outputted to a constellation diagram and passed to a root raised cosine filter.
- The tagged stream multiply length tag block specifies a multiplicative constant for the burst shaper to add zero-padding to the stream (why?).
- The signal then is multiplied with 0.6 (why?)
- ... and then serves as the input for a burst shaper.

[confirmed How exactly does the repack bits block work? I mean what does output byte mean? I presume the following: example - bits per input byte 8, bits per output byte 2 - all bits of an input byte are relevant so they are split up into four bytes where only the first two bits are relevant and the rest is nonsense. The relevant bits are fed to a byte to symbol converter.

2.3.2 The Receiver

- The received signal is adaptively amplified by the AGC (automatic gain control) to a certain level no matter what input signal magnitude (only constrained by a maximum amplifying factor).
- The polyphase clock synchronizer filters (e.g. RRC filter) and then minimizes the derivative of the filtered signal to maximize SNR and minimize ISI (by removing the influence of timing offsets of receiver and transmitter clocks). This is actually achieved by having multiple filters in a filterbank and compare which filter has the lowest difference to zero at the optimal sampling point.
- I didn't really get the implementation details of the above filter.
- The next step is frame synchronization (preamble synchronization).
- derotator seems to be a block for debugging purposes only.
- The baseband derotator then performs derotation by ϕ simply by multiplying with $\exp(-j\phi)$ in order to compensate for carrier frequency offset (CFO) and sampling clock offset (SCO) caused by oscillator mismatch and the Doppler effect. What does the variable μ do?
- The data then is passed to a demultiplexer whose job it is to separate the header from the payload. The payload is subsequently demodulated, repacked, parsed and fed beck to a dedicated port of the demux block. Probably to decide what to do with the payload? The payload is handed

to a constellation plot, a demodulator and a MPSK SNR estimator. The demodulated payload bitstream is repacked and converted back from a tagged stream to a protocol data unit.

3 GNU Radio Techniques and Concepts

3.1 Creating a hierarchical block directly in GRC

In the options block, generate options must be set to "Hier Block". Then press Ctrl+A and right click on any block. Under "More" choose "Create Hier". After a reload the block should be usable.

3.2 The use of Virtual Sinks and Virtual Sources

Virtual sinks and sources are a way to keep the flowgraph tidy. Due to the fact, that the horizontal space is limited several collected "rows" of blocks look overcrowded. The solution is elegant. Simply feed the output of a block to a virtual sink and let in the next row a virtual source with the same name output what the sink has devoured.

3.3 Read/Write Stream Tags

A straight copy from the Blocks Coding Guide.

Reading:

```
int work(int noutput_items,
         gr_vector_const_void_star &input_items,
         gr_vector_void_star &output_items)
{
    std::vector tags;
    //number of items read on port
    const uint64_t nread = this->nitems_read(0);
    //assumption for sync block, this can change
    const size_t ninput_items = noutput_items;

    //read all tags associated with port 0 for items in this work function
    this->get_tags_in_range(tags, 0, nread, nread+ninput_items);

    //work stuff here...
}
```

Writing:

```
int work(int noutput_items,
        gr_vector_const_void_star &input_items,
```

```
        gr_vector_void_star &output_items)
{
    //which output item gets the tag?
    const size_t item_index = ?
    const uint64_t offset = this->nitems_written(0) + item_index;
    pmt::pmt_t key = pmt::pmt_string_to_symbol("example_key");
    pmt::pmt_t value = pmt::pmt_string_to_symbol("example_value");

    //write at tag to output port 0 with given absolute item offset
    this->add_item_tag(0, offset, key, value);

    //work stuff here...
}
```

3.4 Tagged Stream Blocks

For our networking stuff we need packets, but GNR only has streams. The solution to this dilemma are tagged stream blocks. They as all blocks operate on streams, but delimit packets with tags. These tags must be the first item of a PDU and contain a PMT integer indicating the PDU's length, otherwise the flowgraph crashes.

4 Good to Know...

4.1 Git

4.1.1 .gitignore

Create a `.gitignore` file in a directory. Files in this directory and all subdirectories specified in the `.gitignore` regex' will be ignored when committing.

4.2 Cleaning Up the Mess

Removing files from a remote repository is easy - executive summary:

```
#Add files to ignore into .gitignore or remove files from disk
# touch .gitignore
# OR
# rm file

#In the root directory
git rm --cached -r ./
# git commit -am string is a shortcut for
# git add ./
# git commit -m string
git commit -am "removed files"
git push
```

4.3 C++

4.3.1 C++ Classes, Structs, Namespaces

The most important points summarized:

- We're talking about C++ here. In C++ there's no difference between structs and classes, except that members and base classes are public in structs and private in classes by default.
- The whole namespace and class fuss is about data encapsulation versus logical relation of functions.
- Some argue that functions directly realted to a class should always be members of the functions.
- Scott Meyers however suggests (where f is a function and C a class):

```
if (f needs to be virtual)
    make f a member function of C;
else if (f is operator>> or operator<<)
{
    make f a non-member function;
    if (f needs access to non-public members of C)
        make f a friend of C;
}
else if (f needs type conversions on its left-most argument)
{
    make f a non-member function;
    if (f needs access to non-public members of C)
        make f a friend of C;
}
else if (f can be implemented via C's public interface)
    make f a non-member function;
else
    make f a member function of C;
```

- For then avoiding conflicts in big projects wrapping namespaces are advocated by Meyers.

Source: What Scott Meyers says.

4.3.2 (Boost) Bind

The boost bind method is a generalization of the `std::bind1st( const F& f, const T& x )` and `std::bind2nd( const F& f, const T& x )` functions.
There is an excellent article on the topic here.

The most important points can be summarized as follows:

- The Boost library generalizes the STL's binding methods by allowing the bind of an arbitrary amount of arguments to a function.
- A *functor* is a class in which the `operator()()` is implemented. One can think of the class as a function with an inner state.
- The binding functions are *adaptors* which effectively allow us to e.g. use binary functions when unary functions are expected.
- As of C++11 this whole binding concept is deprecated. In C++17 this whole concept will be removed from the STL.
- Instead one should use λ-expressions (since C++11), inspired from implicit mathematical function declaration like this (mapping) $x \mapsto 2x^2$.

4.4 LaTeX

I learned...

- ... when to use input or include
- ... some tikz basics
- ... using the `\newcommand{}` command.