Implementation of Digital Signal Processing: Some Background on GFSK Modulation

Sabih H. Gerez
University of Twente, Department of Electrical Engineering
s.h.gerez@utwente.nl

Version 5 (March 9, 2016)

This document provides some background information on GFSK modulation on behalf of student exercises for the course *Implementation of Digital Signal Processing* as taught at the University of Twente[1].

1 Principle

Gaussian frequency shift keying (GFSK) is a modulation method for digital communication found in many standards such as Bluetooth, DECT and Wavenis. Digital communication amounts to translating symbols from a discrete alphabet into a signal that the transmitting side can send into a transmission medium and from which the receiving side can recover the original symbols.

In the context of this example, the alphabet has only two symbols `0` and `1`. When the alphabet consists of just two symbols, the symbols are called *bits*. The modulation method is a variant of *frequency modulation* (FM) of some carrier frequency ω_c.[2] *Frequency shift keying* (FSK) conveys information by decreasing the carrier frequency for the duration of a `0` symbol and increasing the frequency for the duration of a `1` symbol. If one applies Gaussian filtering to the square-wave signal that would shift the carrier frequency, one gets GFSK.

The models presented here are restricted to the digital part of the entire communication system using an *intermediate frequency* ω_{IF} as carrier frequency. In a real-life system, the signals traveling between antennas have a (much) higher carrier frequency, the so-called *radio frequency* ω_{RF}. Analog circuits are normally used for upconversion to RF at the transmitter and downconversion to IF at the receiver. An *analog-to-digital converter* (ADC) at the receiver side, brings the signal back to the digital domain. The discussion below leaves out the RF part of the signal processing chain and pretends that the communication takes place at IF.

2 Transmitter

The transmitted signal $s(t)$ can be described by a cosine at ω_{IF} with a time-dependent phase:

$$s(t) = A\cos(\omega_{\mathrm{IF}}t + \phi(t)) \tag{1}$$

[1] `http://wwwhome.ewi.utwente.nl/~gerezsh/vlsidsp/index.html`

[2] In this document the term *frequency* is sloppily used for what is actually the *angular frequency* ω: $\omega = 2\pi f$.

In this formula, A is the signal's amplitude which is constant as the modulation only affects phase.

$\phi(t)$ is derived from the bits that are transmitted:

$$\phi(t) = h\pi \int_{-\infty}^{t} \sum_i a_i \gamma(\tau - iT) d\tau$$

h is the *modulation index*: the larger the value, the wider the bandwidth occupied around the carrier. A frequently encountered case is: $h = 0.5$. Note that the case $h = 0$ represents an unmodulated carrier.

a_i is a sequence of numbers: $+1$ if the ith bit is `1` and -1 if that bit is `0`.

$\gamma(t)$ is the frequency pulse. If no Gaussian filter would be applied (FSK instead of GFSK), the frequency pulse would be rectangular: $\frac{1}{T_s}$ in the interval $[0, T_s]$ and 0 outside this interval, where T_s is the duration of one symbol. So, ignoring the sign of a_i, the phase contribution of one symbol would be $h\pi$. Continuing the reasoning, transmitting a continuous series of `1`'s during 1 second would amount to a total phase shift of $\frac{h\pi}{T_s}$. Note that the total phase shift in one second is equal to the angular frequency shift. The real frequency shift is then $\frac{h}{2T_s}$. This means that the instantaneous frequency of an FSK signal is either $f_{\text{IF}} - \frac{h}{2T_s}$ or $f_{\text{IF}} + \frac{h}{2T_s}$ ignoring the effects of switching between the two frequencies ($f_{\text{IF}} = \frac{\omega_{\text{IF}}}{2\pi}$).

The Gaussian filter smoothens the shape of the frequency pulse and makes it wider than one symbol period (this causes *intersymbol interference*). The goal is to avoid the high frequencies caused by switching. When the sequence to be transmitted contains multiple equal bits, the effect of filtering dies out and the extreme instantaneous frequencies mentioned for FSK are reached. Otherwise, the frequency swing around ω_{IF} is smaller.

The Gaussian filter is given by:

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{t}{\sigma})^2}$$

where σ is related to the filter's 3-dB bandwidth B:

$$\sigma = \frac{\sqrt{\ln 2}}{2\pi B}$$

Note that the Gaussian filter's impulse response spans from $-\infty$ to ∞. For practical implementations, the span has to be limited.

3 Noise, SNR, and BER

The *communication channel* is the connection between transmitter and receiver. Distortion of the signal by the channel affects the quality of the received signal. The considered design models so-called *additive white Gaussian noise* (AWGN) as the only source of distortion. If $n(t)$ is the noise signal, the noisy signal $s_n(t)$ can simply be expressed as:

$$s_n(t) = s(t) + n(t)$$

Note that the model does not introduce any attenuation to the signal. The noise is parameterized by the *signal-to-noise ratio* (SNR) which is the quotient of the signal power and the noise power (often expressed in dB). In the model, the signal strength is kept constant and the noise power is chosen that corresponds to the given SNR.

When designing the GFSK receiver, the *bit-error rate* (BER), viz. the number of wrongly detected bits divided by the total number of transmitted bits, is the measure of quality. The design goal is to make the receiver as cheap as possible, for example in terms of logic gates in an ASIC realization, while satisfying the BER requirements.

4 Receiver

A common method to extract the transmitted bits from the modulated signal is to shift the signal to baseband (to reduce the carrier frequency to zero), to filter the signal and then apply a so-called *delay and multiply* transformation. These steps will be explained in short below.

First, the signal as described in Equation 1 will be rewritten to:

$$s(t) = A\cos((\omega_{\text{IF}} + \omega_d)t)$$

Here, ω_d is the instantaneous frequency offset due to modulation (remember that frequency is the time derivative of phase, so $\omega_d = \frac{d}{dt}\phi(t)$). So, if one would be able to know ω_d for a specific symbol period, one would be able to know the value of the bit from the sign of ω_d: $\omega_d > 0$ would mean that a `1` was received and $\omega_d < 0$ that a `0` was received.

In order to eliminate ω_{IF}, one can multiply $s(t)$ with an unmodulated sine and cosine:

$$\begin{aligned} i_m(t) &= s(t) \cdot \cos(\omega_{\text{IF}} t) \\ q_m(t) &= s(t) \cdot -\sin(\omega_{\text{IF}} t) \end{aligned}$$

This is called *mixing*. The two signals $i_m(t)$ and $q_m(t)$ are called the *in-phase* and *quadrature* components of the new signal.

Following the product rules of trigonometry, one then gets:

$$\begin{aligned} i_m(t) &= \frac{A}{2}\left(\cos((2\omega_{\text{IF}} + \omega_d)t) + \cos(\omega_d t)\right) \\ q_m(t) &= \frac{A}{2}\left(-\sin((2\omega_{\text{IF}} + \omega_d)t) + \sin(\omega_d t)\right) \end{aligned}$$

The interpretation of these formulae is that both $i_m(t)$ and $q_m(t)$ will now contain the original signal twice: once around center frequency $2\omega_{\text{IF}}$ and once around center frequency 0. The signal component around $2\omega_{\text{IF}}$ can be removed by low-pass filtering to obtain:

$$\begin{aligned} i_l(t) &= \frac{A}{2}\cos(\omega_d t) \\ q_l(t) &= \frac{A}{2}\sin(\omega_d t) \end{aligned}$$

The delay-and-multiply operation is a classical technique for FM demodulation [1]. It amounts to computing:

$$d(t) = q_l(t) \cdot i_l(t - \Delta T) - i_l(t) \cdot q_l(t - \Delta T)$$

Applying once again the product rules for sines and cosines, one finds:

$$d(t) = \frac{A^2}{4}\sin(\omega_d \Delta T)$$

Parameter	Value
symbol rate	500 kHz
modulation index h	0.5
input sample frequency (ADC output)	8 MHz
hardware clock frequency	8 MHz
f_{IF}	1 MHz
bandwidth low-pass filter	1 MHz

Figure 1: *Problem specification.*

If $\Delta T = T_s$ (one could also choose other values) and $h = 0.5$, remembering that the maximal frequency deviation for GFSK is $\frac{h\pi}{T_s}$, the value of $d(t)$ will be at most $\frac{A^2}{4} \sin \frac{\pi}{2} = \frac{A^2}{4}$. Similarly, the minimum value of $d(t)$ will be $-\frac{A^2}{4}$. Actually, $d(t) > 0$ for $\omega_d > 0$ and $d(t) < 0$ for $\omega_d < 0$. This means that the sequence of mixing, low-pass filtering and delay and multiply has resulted in a signal from which the original sequence of bits can be extracted by sampling at the right moment.

5 Facts on the Reference Designs

This section presents some numeric information about the reference GFSK designs and discusses its implications.

Some specifications for the GFSK project are given in Figure 1. Modulation uses 16 samples for each symbol. For this reason, a symbol rate of 0.5 MHz results in a sample rate of 8 MHz (so, using the Nyquist criterion, the maximum frequency representable in the system is 4 MHz). A hardware clock of 8 MHz implies a *one-to-one* implementation.

It was mentioned in Section 2, that the Gaussian filter has an infinite span in theory, but that the span needs to be limited in practice. In the reference design, the span has been limited to 4 symbol periods. It is therefore implemented as a 64-tap FIR filter.

The low-pass filter has a bandwidth of 1 MHz, meaning that all signals with a higher frequency are suppressed. This has consequences for the noise added in the channel. As the bandwidth of the simulated system is 4 times as large as the bandwidth considered after filtering, the SNR is corrected by a factor of 4. This can be seen as follows: the added noise is white, meaning that all frequencies are equally present; the filter removes three quarters of the noise; so, to have the correct noise energy in the frequency band of interest, a correction with 4 is necessary.

The low-pass filter is an FIR filter with coefficients optimized for a multiplierless implementation as published in [2].

As the maximum frequency after filtering is 1 MHz, one can reduce the sample rate to 2 MHz, using a downsampling factor of 4. This is actually the case in the example receiver designs that are provided. In the 2 MHz domain, this means that there are 4 samples for each symbol instead of 16.

Delay and multiply consumes and produces 4 samples per symbol. The problem left is to detect the symbol boundaries in this data stream and take a decision about the bits received. Finding the symbol boundary is called *synchronization*. The optimal decision about the bits received should involve the samples of the symbol itself as well as those of its neighbors (remember that the Gaussian filter in the transmitter spreads a symbol across multiple symbol periods).

The reference design takes a pragmatic approach on synchronization and decision. It has a block called the *slicer* for this purpose. Only the four samples of the symbol itself are considered. For the sake of simplicity, no samples of neighboring symbols are taken into account in spite of the presence of intersymbol interference. The four samples are added and then the sign of the sum is checked. Adding the four samples instead of inspecting just one, makes the decision more robust in the presence of noise. The hardware does not perform synchronization. Synchronization is supposed to be performed by an external hardware unit that provides a parameter called `slicer_offset` which is an integer in the range from 0 to 3. The slicer continually adds the last four samples that it has received, but only updates its output when an internal modulo-4 counter has a value equal to `slicer_offset`.

As the output of the slicer only changes once in 4 samples, its output can be downsampled once again resulting in an output stream consisting of bits: one bit is delivered for each transmitted bit.

References

[1] J.H. Park. An FM detector for low S/N. *IEEE Transactions on Communication Technology*, COM-18(2):110–118, April 1970.

[2] J.M.P. Langlois, D. Al-Khalili, and R.J. Inkol. Polyphase filter approach for high performance, FPGA-based quadrature demodulation. *Journal of VLSI Signal Processing*, 32:237–254, 2002.