Formulae and Algorithms for the GMSK Modulation

Laszlo Hars

Panasonic Information and Networking Technologies Laboratory
E-mail: Laszlo.Hars@Research.Panasonic.com

Abstract Exact formulae are presented for the instantaneous phase and frequency of GMSK modulated signals used at telecommunication systems GSM, DECT, TETRA. Polynomial and rational approximations are derived for them with different accuracy. The spectrum of the modulated signal and the zero positions of the equivalent FM are also investigated. Algorithms for (de)modulation are described.

INTRODUCTION

In telecommunication systems the information is transmitted by means of some kind of modulation of high frequency electromagnetic waves. If the data to be transmitted is digital, digital modulation schemes are used. There have been many of them proposed and used, like Phase Shift Keying, Frequency Shift Keying, and Quadrature Amplitude Modulation. They present different tradeoffs between cost and tolerance to noise and other disturbances, spectral efficiency etc. One of the most widely used digital modulation techniques is GMSK: Gaussian Minimum (phase) Shift Keying.

A general modulated signal is expressed as

$$A(t)\cos\left(\omega_c t + \varphi(t)\right)$$

where t is the time, $A(t)$ is the amplitude envelope of the signal, $\omega_c = 2\pi \cdot f_c$ is the (angular) frequency of the carrier, $\varphi(t)$ is the phase offset. The instantaneous frequency of the modulated signal is the derivative of the total phase with respect to the time

$$2\pi \cdot f(t) = \omega_c + \varphi'(t),$$

therefore it is determined by $\varphi(t)$ up to the constant ω_c.

In order to minimize the interference between different sources of radio frequency signals, their bandwidth must be restricted. In general, limiting the bandwidth means band-pass filtering the signal. The more sharp edges we have in the frequency domain, the longer is the impulse response of the filter. It means the bits of the modulating digital signal have more and more affects on each other, the demodulation gets more difficult.

At GMSK the amplitude $A(t)$ is kept constant and the bits of the digital modulation data are used to increase or decrease the phase by a fixed angle ($\pi/2$). The speed of the change of the phase is limited by the means of a Gauss (low-pass) filter. It

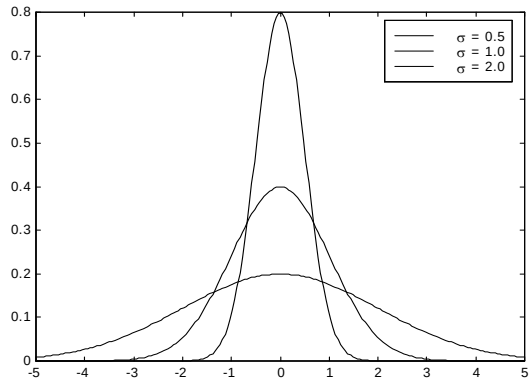
effectively restricts the bandwidth of the transmitted signal in cost of introducing a small interference of the modulating bits. In theory, each bit influences the phase at any time in the past and in the future. However, this affect is small a few bits away in the past. A few bits later the phase gets increased or decreased by a nearly constant angle. It means, in a long period the bits have a cumulative (summed) affect on the phase, and the influence of each bit is steady outside a short transient time interval.

The Gauss Function

The classical Gauss function plays an important role in the definition of the GMSK modulation:

$$h(t) = \frac{1}{\sqrt{2\pi}\cdot\sigma T} e^{-\frac{1}{2}\left(\frac{t}{\sigma T}\right)^2}.$$

With appropriate scaling of the variable t we can take $T = 1$. Often it is convenient to do so. Here are the plots of $h(t)$ with $\sigma = 0.5, 1, 2$.



The Gauss Function

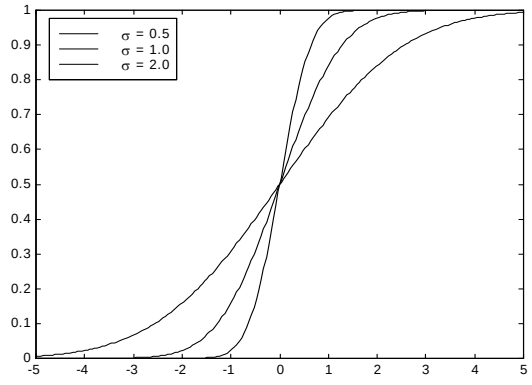
Note, that $1 = \int_{-\infty}^{\infty} h(t)\,dt$. The integral of $h(t)$ cannot be expressed with elementary functions, but they are needed often, so a special function has been defined:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

With the erf function

$$\int_{-\infty}^{x} h(t)\,dt = \frac{\operatorname{erf}\left(\frac{x}{\sigma T\sqrt{2}}\right)+1}{2\,T}.$$

There have been good piece-wise polynomial approximations published for the erf. With a few arithmetic operations high accuracy is achieved, as it is implemented in MATLAB, a numerical calculations software packet.



Integral of the Gauss Function

Sometimes we need even the integral of the erf function, too:

$$\int \operatorname{erf}(x)\,dx = \frac{1}{\sqrt{\pi}} e^{-x^2} + x\cdot\operatorname{erf}(x).$$

Convolution Integral

The convolution g of two functions h and r is defined as

$$g(t) = \int_{-\infty}^{\infty} h(u)\cdot r(t-u)\,du.$$

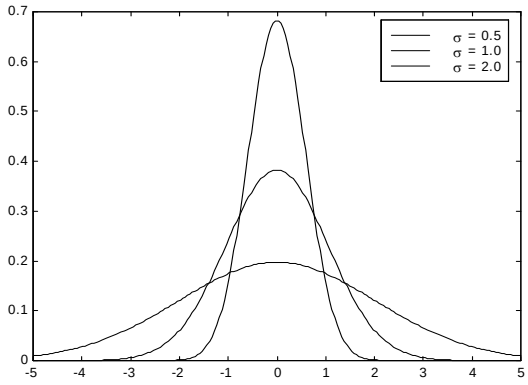
As an example, let us convolute the Gauss function with the rectangular pulse:

$$\mathrm{rect}(t) = \begin{cases} 1 & \text{if } |t| < \frac{T}{2} \\ 0 & \text{otherwise} \end{cases}$$

The convolution is

$$g(t) = \int_{-\infty}^{\infty} h(u) \cdot \mathrm{rect}(t-u)\, du = \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} h(u)\, du = \frac{1}{2T}\left[\mathrm{erf}\left(\frac{t+\frac{T}{2}}{\sigma T\sqrt{2}} \right) - \mathrm{erf}\left(\frac{t-\frac{T}{2}}{\sigma T\sqrt{2}} \right) \right].$$

This function is the Gauss filtered rectangular pulse. It has very similar shape to the Gauss function:



Gauss-Filtered Pulse

The match between the filtered pulse ($\sigma = 1$) and a little wider Gauss function ($\sigma = 1.04158$) is very good. The maximum difference is less than 1.6×10^{-4}.

THE GMSK MODULATION

Define the constant $\sigma = \dfrac{\sqrt{\log 2}}{2\pi BT}$, with BT = 0.3 in case of the GSM system. Other common values are BT = 0.5 at DECT and BT = 0.25 at Tetrapol. The corresponding σ values are 0.44168, 0.265010, 0.530021 respectively. Here B corresponds to the bandwidth of the filter represented by the Gauss function. The smaller it is, (the larger is σ), a filtered pulse has less sharp edges, the fast changes become more gradual.

The function $g(t)$ is the convolution of the Gauss function with a rectangular pulse as above. Let α_k = ±1 be the sequence of the (signed) bits to modulate with. The instantaneous phase as a function of time is defined as:

$$\varphi(t) = \pi c \cdot \sum_k \alpha_k \int_{-\infty}^{t-kT} g(u)\, du$$

with the constant c, the modulation index. (It is ½ at all the telecom systems in use.) With the earlier formulae we can write the phase in closed form which can be evaluated to any precision. The integral is easy to derive from that of the erf:

$$\overline{G}(t) \overset{def}{=} \int g(t)\, dt = \tfrac{1}{2T}\int \mathrm{erf}\left(\frac{t+\frac{T}{2}}{\sigma T\sqrt{2}} \right) dt - \tfrac{1}{2T}\int \mathrm{erf}\left(\frac{t-\frac{T}{2}}{\sigma T\sqrt{2}} \right) dt =$$

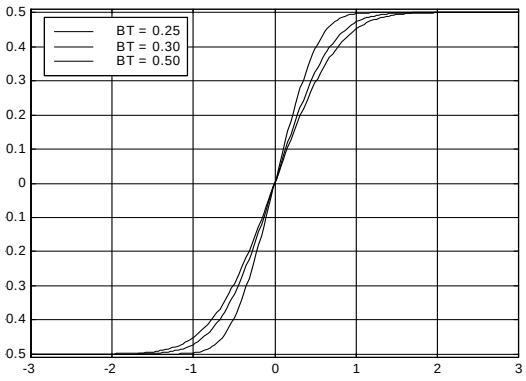
$$\frac{\sigma}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t+\frac{T}{2}}{\sigma T}\right)^2} + \frac{t+\frac{T}{2}}{2T}\mathrm{erf}\left(\frac{t+\frac{T}{2}}{\sigma T\sqrt{2}} \right) - \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\frac{T}{2}}{\sigma T}\right)^2} - \frac{t-\frac{T}{2}}{2T}\mathrm{erf}\left(\frac{t-\frac{T}{2}}{\sigma T\sqrt{2}} \right).$$

At $-\infty$ the exponential terms approach 0, the erf converges rapidly to –1. (We know that $\mathrm{erf}(t) = -1 + \mathrm{o}\left(\frac{1}{t}\right)$ there). What remains is

$$\frac{t+\frac{T}{2}}{2T}\left(-1+\mathrm{o}\left(\frac{1}{t}\right)\right) - \frac{t-\frac{T}{2}}{2T}\left(-1+\mathrm{o}\left(\frac{1}{t}\right)\right) = -\frac{1}{2} + \mathrm{o}(1)$$

giving $-\frac{1}{2}$ for the limit at $-\infty$. Similarly,

$$\lim_{t\to\infty} \overline{G}(t) = \frac{1}{2}.$$



Integral of a Gauss filtered impulse

The bit weight function, being an integral from $-\infty$, must have a limit 0 there. Accordingly, we define **bitwgt(t)**$= G(t) = \overline{G}(t) + \frac{1}{2}$. In MATLAB it looks:

```
function u = bitwgt(t, BT, T)
t1  = t/T - 0.5;
t2  = t/T + 0.5;
s   = sqrt(log(2)) / (2*pi*BT);
s2  = s * sqrt(2);
s2p = s / sqrt(2*pi);
u   = s2p*exp(-(t2/s).^2/2) + t2/2.*erf(t2/s2)...
    - s2p*exp(-(t1/s).^2/2) - t1/2.*erf(t1/s2) + 0.5;
```

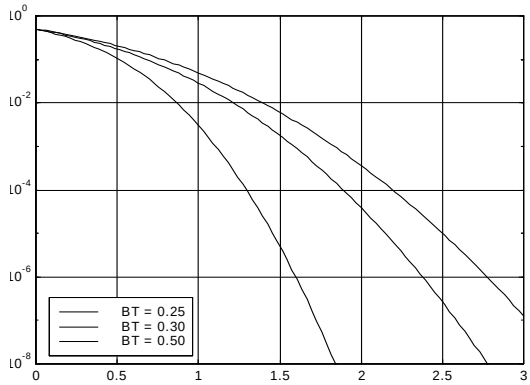
To perform the phase modulation of a carrier of frequency ω_c, we need:

$$\cos\left(\omega_c t + \varphi(t)\right)$$

with the instantaneous phase depending on the modulation bits and the time:

$$\varphi(t) = \pi c \cdot \sum_k \alpha_k \cdot G(t - kT).$$

In theory, each bit influences the phase at any time in the past and in the future. However, the function G is very close to 0 if $t \ll k - 2T$, so the affect on the far past is negligible. If $t \gg k + 2T$, G is very close to 1. So the bits have a cumulative affect in the future, and the affect of each bit remains constant after a short transient period. The graph above shows that in the GSM case ±2 bits time is enough to use function G and 0/1 elsewhere, with less than -88 dB error. (±1.75 bits give -70 dB, about 12 ADC bit accuracy, ±1.5 bits give -55 dB, 0.18% accuracy.) In the DECT case 1.375 bit time gives already almost 90 dB accuracy, 1.25 bits give almost -75 dB, more than 12 ADC bit accuracy.



Deviation from 1 of the bit-weight

RATIONAL APPROXIMATIONS

For simulation purposes the above MATLAB function **bitwgt** is perfect, but in a real-time environment, like in an instrument sending GMSK modulated signals a faster

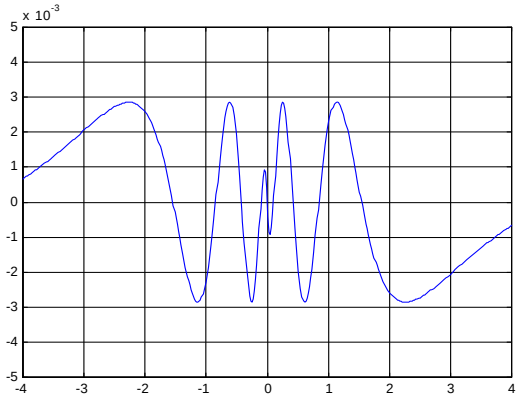
way is needed to calculate the phase. A look-up table is fast, but if memory is restricted we must use polynomial or rational approximations, which are only a little slower. The advantage of rational approximations is that the argument need not be restricted to a specific interval, the result can be accurate everywhere. If divisions are slow, polynomial approximations can be faster, though they need a couple of compare-branch instructions to provide 0 or 1 outside of the approximation interval. We present several approximation functions of different accuracy. C functions are listed generating accurate phase samples with the approximations. They are used in phase error measurements and in modulators.

The function G is smooth, so in finite intervals low order rational functions can be found to approximate it. Because of space limitations we restrict ourselves to GSM, the most important case. Similar techniques lead to formulae for cases where $BT \neq 0.3$.

Considering the shape of the graph of G, an approximation of the form $\frac{1}{2}+\frac{x\cdot\left(a+b\cdot x^2\right)}{c+|x|\cdot\left(d+e\cdot x^2\right)}$ looks promising. It can be calculated with about 10 CPU cycles using DSP chips, which have division instructions. With an optimization procedure one can find the best set of coefficients, as follows. The asymptotic approximation must be as good as everywhere else, so during the optimization we keep the ratio of the coefficients of the highest x-powers in the numerator and denominator constant (near 1). This constant gives ideally the same asymptotic error as the largest approximation error at smaller arguments, which is not known until the optimization finishes. Therefore one needs an extra level of iteration to bring these values reasonably close to each other. The other coefficients are determined by a weighted least-squares optimization or a mini-max approximation with the Remez exchange algorithm. (See [5].) The result is

$$G(x)\approx\frac{1}{2}+\frac{x\cdot\left(0.80161+1.93462\cdot x^2\right)}{1.01427+|x|\left(0.92459+3.89053x^2\right)}$$

It gives an error at most ±0.0029. This particular case is difficult to handle with the Remez exchange algorithm, because there are more extreme values of the error function than the number of free parameters would normally produce. A weighted least-squares optimization works well, though.



Error of a rational approximation of G(x)

An advantage of this formula is that we need not restrict the argument to a specific interval, the result is always accurate.

If we replace x^2 with $|x|$ a similar good approximation is expected. However, there is no extra extreme value of the error function now, and the best approximation polynomial gives a much larger (±0.0095) error, with only one less arithmetic operation:

$$G(x) \approx \frac{1}{2} + \frac{x \cdot (0.2743 + 0.8314 \cdot |x|)}{0.5206 + |x| \cdot (0.17554 + 1.6952 \cdot |x|)}.$$

We can use higher order rational approximations, too. Different techniques help making them as fast as possible to evaluate. E.g. if we see a coefficient to be small at an optimum approximation, we fix it to 0, and redo the calculations. Usually we loose only a little accuracy but save 1 or 2 arithmetic operations. An example is the following:

$$G(x) \approx \frac{1}{2} + x \cdot \frac{1.114609 - 0.1662698 \cdot |x| + 1.817079 \cdot x^2}{1.393870 + |x| \cdot (0.862554 + 3.622348 \cdot x^2)}.$$

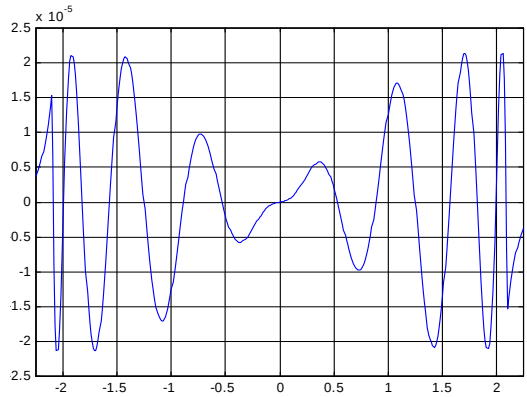
It gives an error at most ±0.00166, about half of the error of the first rational approximation above, with two more arithmetic operations.

POLYNOMIAL APPROXIMATIONS

If our computational platform has no fast division, polynomial approximations can be faster to evaluate. Because $G(x)–½$ is an odd function, in symmetric intervals the best approximating polynomials are of the form $½+x \cdot p(x^2)$. If the argument is outside of the approximation interval, we use 0 or 1. One gets less than ±0.0022% absolute error (0.0012°), ±0.0017% relative error in the interval [–2.095, 2.095] with the following $p(x)$:

$$0.00005737350850 \cdot x^6 – 0.00118403525473 \cdot x^5 + 0.01062748051621 \cdot x^4$$
$$– 0.05487241021198 \cdot x^3 + 0.18149294715579 \cdot x^2 – 0.40690840660122 \cdot x + 0.74237558203693$$

This gives a 13th degree approximation polynomial, which is good enough for moderate precision simulations. To determine this polynomial we used first the least squares optimization algorithm of MATLAB and extracted every other coefficients (the non-zero ones). This is already very good, but the maximum error can be improved further with a coefficient optimization or a special weighting schema for a second least-squares optimization. These result in the polynomial we listed above.



Error of the p_{13} polynomial approximation

Its peak error is 2.135e–5. If higher accuracy is desired, one can repeat the same procedure with longer polynomials. For example the best 19th degree approximation polynomial has an absolute error smaller than 3.604e–6. The following table contains the coefficients of a few best mini-max approximation polynomials along with the maximum errors and the limits beyond which the trivial 0 or 1 value is used:

Coefficients	Approximation Error	Approximation Interval
9.39827957127403e-9 -3.59336850815957e-7 6.04854418895465e-6 -5.85741073236036e-5 0.000354178044482604 -0.00130981700348589 0.00215429267022587 0.00566478296345006 -0.0507306804098209 0.179857667538025 -0.406722927913324 0.742386044098122	7.92e-7	2.425
2.34099199874687e-7 -5.97110568850486e-6 5.96066535905209e-5 -0.000245918977800911 -0.000343578515094397 0.00940718761489751 -0.0541398530566104 0.181577115183351 -0.407122283650813 0.74241341718403	3.604e–6	2.257
5.73735085036847e-5 -0.00118403525472754 0.0106274805162063 -0.054872410211976 0.18149294715579 -0.406908406601224 0.742375582036931	2.135e–5	2.095
-0.000536574128250518 0.00782589786918152 -0.0490708250525103 0.175668079274966 -0.404407893574327 0.742070795527179	0.0000979	1.89
0.00350411939854986 -0.0364605066426604 0.159644742767781 -0.396147057641104 0.740887899611663	0.000351	1.73
-0.0136463081061985 0.110545625947229 -0.35744115683194 0.732774337850225	0.001787	1.55
0.0653215840982424 -0.315019192202077 0.722911344964959	0.004175	1.4

For example, the approximation based on the polynomial in the second last row:
$p_3(x) = -0.0136463 \cdot x^3 + 0.110546 \cdot x^2 - 0.357441 \cdot x + 0.732774$ of error ±0.0018, in full form:

$$\varphi(t) \approx \begin{cases} 0 & if \quad t < -1.55 \\ \frac{1}{2} + t \cdot p_3\left(t^2\right) & if \quad t \in [-1.55, 1.55] \\ 1 & if \quad t > 1.55 \end{cases}$$

One can try approximations in the form $½+x \cdot p(|x|)$, too. If polynomials of the same degree are used as at the approximations in the form $½+x \cdot p(x^2)$, the same number of operations is needed. We tried these too, but the approximations were generally inferior to the previous ones (for the same error we needed a polynomial of one higher degree).

FAST GENERATION OF IDEAL GMSK MODULATION PHASE

Here we give C functions, which calculate very precisely the modulation GSM phase. (For other GMSK systems, like DECT, one only needs to change the constants.)

The phase calculation will have an error at most around the number of samples-per-bit times the error of the weight function. They are optimized for the requirements of phase error measurements. The signal has to be digitized with an integer number of samples per modulation bit. Ideally one of the sampling time points falls to the middle of a bit. At a signal received from a mobile phone it is not normally the case. For the calculation of the phase error of a signal we need to generate a perfectly modulated GSM signal at the same time offset relative to the middle of the bits. (The RMS error between the 2 signals has to be minimized, thus the phase samples are calculated at many offsets.)

Because the bit weight function is only needed at some raster points (which repeat themselves later, sifted by a number of bit periods), we pre-calculate the necessary weights and re-use them later. These values occupy only about 5 times the samples per bit (which is chosen in a measurement instrument between 2 and 16, normally 4).

We can only deal with a finite number of bits, therefore make an assumption, that the tail bits of the modulation sequence are infinitely many zeros at both ends. The actual effects of these zeros are only perceived at a finite distance of less than 2.5 bits. To be safe, do always simulations with 3 or 4 zero modulation bits at the tails.

```
#define  BITWGT_LIM  2.425

/*************************************************************************\
* BitWeight     23rd order antisymmetric polynomial approximation
*   (12 coefficients) 0.5+x*p11(x^2) evaluated by the Horner's scheme, error +/- 7.92e-7
* Parameter:    t  Normalized Time point: +/-0.5 are the bit-boundaries
* Return value: The weight a bit affects the phase at time t
\*************************************************************************/
double BitWeight( double t ) {
    double t2;
    if ( t <-BITWGT_LIM ) return 0.0;
    if ( t > BITWGT_LIM ) return 1.0;
    t2 = t * t;
    return ((((((((((9.39827957127403e-9  * t2 - 3.59336850815957e-7) * t2
                   + 6.04854418895465e-6) * t2 - 5.85741073236036e-5) * t2
                   + 0.000354178044482604)* t2 - 0.0013098170034858 9) * t2
                   + 0.00215429267022587) * t2 + 0.00566478296345006) * t2
                   - 0.0507306804098209 ) * t2 + 0.179857667538025  ) * t2
                   - 0.406722927913324  ) * t2 + 0.742386044098122  ) * t + 0.5;}

/*************************************************************************\
* GSMPhase      The GSM modulation phase in radians calculated at time points
*               t = TimeOffs + (0 : Samps_Bit*BitsLen - 1) / Samps_Bit
*               (uses n extra modulation bits at both tails)
*               The algorithm is optimized for short bit sequences.
*               Otherwise it is faster to save the phase sums in a look-up table
*               Modulation phase error < Samps_Bit * Error(BitWeight)
* Parameters:
*   ModBits     Pointer to the array of modulation bits (+1,-1)
*               - MUST HAVE (3) EXTRA ENTRIES BEFORE AND AFTER THE BITS -
*                 before = (int)(BitWgtLim - TimeOffs)
*                 after  = (int)(BitWgtLim + TimeOffs)
*   BitsLen     Length of the modulation bit sequence
*   TimeOffs    Time offset in the interval (-1, 1)
*   Samps_Bit   Samples per bit
*   BitWgtLim   The limit beyond the GMSK bit weight function has 0/1 values
*   BitWgt      The name of the bit weight function
*   wgt         Pointer to a buffer for the bit weights
*               size >= FLOOR( 2 * BITWGT_LIM * Samps_Bit) + 1
*   Phase       Buffer for the calculated phase samples
* Return value: The length of the output data
\*************************************************************************/
int GSMPhase( int   *ModBits,   int      BitsLen,         double  TimeOffs, int     Samps_Bit,
              double BitWgtLim, double (*BitWgt)(double), double *wgt,      double *Phase) {
```

```
int    i, j, k,
       past = 0,     /* accumulated past phase in the non-trivial interval*/
       m1 = Samps_Bit * (BitWgtLim + TimeOffs),  /* #weights on the left  */
       m2 = Samps_Bit * (BitWgtLim - TimeOffs),  /* #weights on the right */
       k1 = TimeOffs  -  BitWgtLim,              /* -#bits on the right   */
       iw = m1 - Samps_Bit * k1,                 /* wgt index of last bit */
       WgtLen = m1 + m2 + 1;
double t, ph, Bit_Samps = 1.0 / Samps_Bit;
for ( i = 0, t = TimeOffs - m1 * Bit_Samps;
      i < WgtLen;
      ++i,   t+= Bit_Samps)
    wgt[i] = (*BitWgt)(t);           /* The bit weights needed */
past = 0;
t    = TimeOffs;
for ( i = 0; i < BitsLen*Samps_Bit; ++i) {
    if ( iw  >= WgtLen) {
         iw  -= Samps_Bit;
         past+= ModBits[k1];
         k1  += 1;     }
    ph = past;
    for ( j   = iw,   k = k1;
          j  >= 0;
          j  -= Samps_Bit, ++k)
        ph   += ModBits[k] * wgt[j]; /* Loops about 2 * BitWgtLim times */
    *Phase++  = M_PI_2 * ph;
    t  += Bit_Samps;
    iw += 1;     }
return BitsLen*Samps_Bit; }
```

Remark: the approximation errors of the phase generation accumulate and can be as large as the error of the bit weight function multiplied by the sampling rate. It means, higher sampling rate requires proportionally more accurate bit weight approximation.

INSTANTANEOUS FREQUENCY

The derivative of the phase is the frequency (c is the modulation index):

$$2\pi f(t) = \frac{d\varphi}{dt} = \pi c \cdot \sum_k \alpha_k G'(t - kT) = \pi c \cdot \sum_k \alpha_k g(t - kT).$$

Maximum Frequency Deviation

The maximum of the frequency occurs when each α_k equals to 1, the minimum when each α_k equals to –1. Substituting the formula for $g(t)$ we get the exact expression

$$\max f = \frac{c}{4T} \sum_k \left[\operatorname{erf}\left(\frac{t - T\cdot\left(k + \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) - \operatorname{erf}\left(\frac{t - T\cdot\left(k - \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) \right] = \frac{c}{4T}\left[\operatorname{erf}(\infty) - \operatorname{erf}(-\infty)\right] = \frac{c}{2T}$$

That is: $\boxed{\max f = \frac{c}{2T} = \frac{c}{2} f_{bit}}$. In case of the GSM system $f_{bit} = \frac{13\,\text{MHz}}{48} = 270.833\ldots\text{KHz}$ and $c = \frac{1}{2}$ are defined. They give $\max f = \frac{13\,\text{MHz}}{192} = 67.708\ldots\text{KHz}$. In case of DECT not c, but f_{bit} = 1152 KHz and $\max f$ = 288 KHz = $f_{bit}/4$ are defined. They give also $c = \frac{1}{2}$.

Frequency Deviation with Alternating Modulation Bits

Let us look at the frequency deviation in case of a modulation bit sequence of ...,+1,–1,+1,–1,... . The frequency cannot reach its maximum of $\frac{c}{2T}$ because of the fast

changing bits. Therefore this sequence provides a good test of a GMSK modulation device. The exact value of the instantaneous frequency at time *t* is given by the function

$$f(t) = \frac{c}{2} \cdot \sum_k (-1)^k \, g(t - kT)$$

Substituting the definition of *g* in it gives the frequency in form of the infinite sum

$$f(t) = \frac{c}{4T} \cdot \sum_k (-1)^k \left[\operatorname{erf}\left(\frac{t - T\cdot\left(k - \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) - \operatorname{erf}\left(\frac{t - T\cdot\left(k + \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) \right]$$

This function has a period of 2*T*, changes sign when shifted by *T*, and because of the symmetry its extreme values are at integer multiples of *T*. Choosing a large enough finite sub-range of the sum we get an approximation of the frequency in the middle of this range. We know, that the expression in the square bracket approaches 0 very fast, so discarding both the unused infinite sums causes only very little errors in the middle.

$$f(t) \approx \frac{c}{2T} \sum_{k=-n+1}^{n-1} (-1)^k \operatorname{erf}\left(\frac{t - T\cdot\left(k - \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) + \frac{(-1)^n c}{4T} \left[\operatorname{erf}\left(\frac{t - T\cdot\left(-n - \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) - \operatorname{erf}\left(\frac{t - T\cdot\left(n + \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) \right]$$

Here the expression in the square brackets is almost 2. In fact,

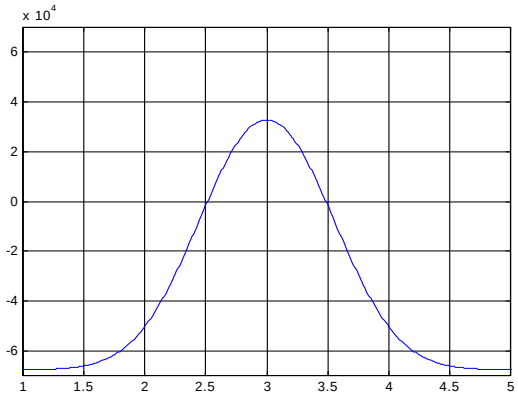
$$f(t) = \frac{c}{2T} \cdot \lim_{n\to\infty} \left((-1)^n + \sum_{k=-n+1}^{n-1} (-1)^k \operatorname{erf}\left(\frac{t - T\cdot\left(k - \frac{1}{2}\right)}{\sigma T\sqrt{2}} \right) \right)$$

Numerical calculations give, in case of the GSM (*BT* = 0.3), a peak frequency deviation of **32,914.7 Hz,** 48.6124% of the possible maximum. It is less than the half of the peak FM deviation. In case of the DECT (*BT* = 0.5), the peak frequency deviation is **253,902 Hz**, 88.1604% of the maximum. It is close to the peak FM deviation.

Zeros of the Frequency Deviation

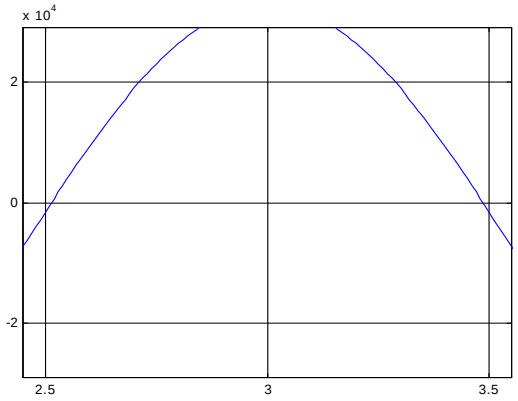
For demodulation purposes it is also helpful to know where the frequency deviation curve crosses the zero line. It would be nice, if these happened at exactly halfway between the centers of the modulation bits. Unfortunately, this is not the case: we have interference between the modulation bits (ISI). It is relatively small at DECT, but large enough at GSM to cause problems with high accuracy measurements.

The exact formulae are so complex, that we cannot solve the corresponding equations exactly. The only possibility is to use accurate approximations and numerical solutions of the resulting equations.



FM with modulation bits 00100

To see the effect more clearly we can zoom in the zero crossings (see the figure).



FM zoomed

We can put the zero-crossing position offsets in a table. Simple linear interpolation is enough for the calculations. Bits further than 3 positions away do not have significant affects, so the table is only 16 entries long: because of the symmetry we only consider 01 bit changes, and the 2 previous and the 2 following modulation bits.

The calculations are performed with 100 samples per bit. Larger values are more accurate, but for our purposes 100 is more than enough. The offset values are accurate up to at least 3 decimal places after the point ($1.4\% \approx 1.4183\%$). The table shows that only one more bit on both sides of a 01 modulation bit pair affects noticeably the position of the corresponding zero of the instantaneous frequency curve. It can be summarized with the following

Modulation Bits		0-crossing offset [%]
0 0 0	1 0 0	1.4
0 0 0	1 0 1	1.4
0 0 0	1 1 0	0.0
0 0 0	1 1 1	0.0
0 1 0	1 0 0	0.0
0 1 0	1 0 1	0.0
0 1 0	1 1 0	-1.4
0 1 0	1 1 1	-1.4
1 0 0	1 0 0	1.4
1 0 0	1 0 1	1.4
1 0 0	1 1 0	0.0
1 0 0	1 1 1	0.0
1 1 0	1 0 0	0.0
1 1 0	1 0 1	0.0
1 1 0	1 1 0	-1.4
1 1 0	1 1 1	-1.4

Rule: *The position of a zero of the instantaneous frequency between modulation bits 01 is moved*

- *1.4% earlier by a leading 0 bit*
- *1.4% later by a trailing 1 bit.*

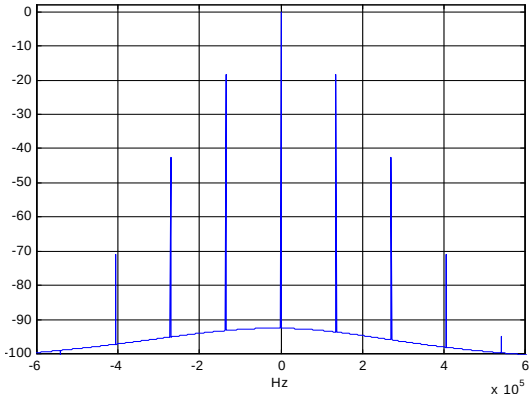
Their affects are additive.

These mean, that there are 3 possible zero crossing offsets: 0 and ±1.4%. It follows, that a positive or negative frequency period can be longer or shorter by 0, ±1.4% or ±2.8% than an integer number of modulation bit period.

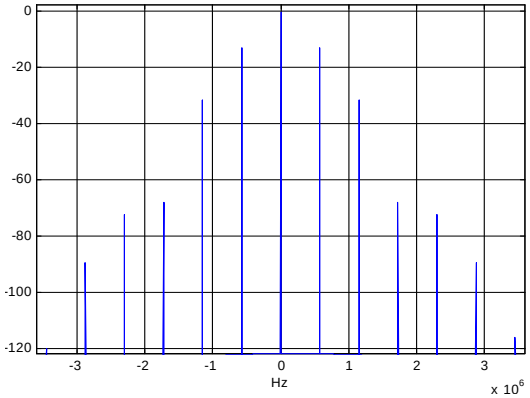
They are so small that don't disturb a demodulation algorithm based on the position of the zeros of the frequency curve. Only high precision measurements need compensation for it. Of course, at DECT this jitter of the zero crossing is even smaller.

GMSK FREQUENCY SPECTRUM

Modulating with all 0's or with all 1's yield a constant frequency shift of the carrier, that is, the spectrum (power spectral density) consists of only one nonzero point. The other extreme is when modulating with alternating ±1 bits. This pattern makes the frequency of the modulated signal to change the fastest, therefore the spectrum will decay the most slowly off the carrier. Also, this bit pattern occurs the most frequently, because differential coding is used at most of the telecom systems, and it converts the silence (all 0 pattern) to alternating bits.



GSM spectrum of modulation with 1010…bits



DECT spectrum of modulation with 1010…bits

Using the expressions derived earlier one could find a closed form formula for the spectrum, but it looks very complicated. Instead, we use numerical approximations and discrete Fourier transforms for the spectrum. The peaks are at integer (k) multiples of $\frac{1}{2} f_{bit}$ ($135416\frac{2}{3}$ Hz at GSM, 576 KHz at DECT). They are listed in the following table.

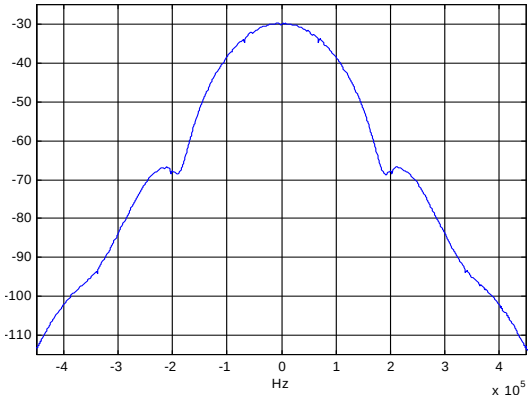
k	0	1	2	3	4	5
GSM	-0.1288	-18.37	-42.67	-71.08		
DECT	-0.4457	-13.176	-31.955	-68.156	-72.483	-89.680

Levels of GMSK spectral components at modulation with 1010…bits [dB]

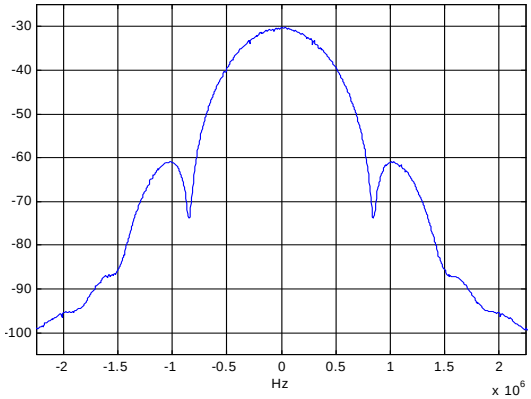
It is also important to look at the spectrum with random modulating bits. Compressed data, speech all look like random bit sequences. To see the corresponding spectrum, the simplest way is the simulation. Let us generate many pseudo-random bit sequences (1000 for the following figures) and average the corresponding spectral data.

The graph shows, that at 200 KHz frequency offset (the GSM channel spacing) the modulated signal is still –36 dBc strong. Even the second adjacent channel is disturbed.

(There are local peaks at the integer multiples of $\frac{1}{2} f_{bit} = 576$ KHz, due to the



Random bits modulated GSM spectrum



Random bits modulated DECT spectrum

subsequences of alternating bits.) We see, that at 1.728 MHz frequency offset (the DECT channel spacing) the modulated signal is under –63 dBc. With some degradation of signal quality the first adjacent channels might be usable. No channels further away are disturbed.

DEMODULATION METHODS

Since the information is coded in phase, we have to retrieve the phase of the carrier sine wave. The change of the phase caries the information, so we need the phase difference between consecutive samples, or the derivative of the phase, the modulation frequency (FM). Next check the sign of the instantaneous frequency around the middle of the bit period. (Finding the optimum sampling point is discussed in a subsequent paper.) If it is positive, the modulation bit is 1, if it is negative, the modulation bit is 0 (or –1).

The FM conversion can be done also in the digital domain. Normally the signal is mixed down to an intermediate frequency band and converted to digital there. The digital sample sequence has to be processed, a typical DSP task. One could look at the instantaneous frequency of the signal by a short, sliding FFT. This is not very fast, because it calculates all the frequency components and we only need the strongest one. Similarly, wavelet transforms are also not practical. A better method is to split the signal into quadrature (complex) components by a Hilbert transform or by multiplying with $\cos(\omega_0 t)$ and $\sin(\omega_0 t)$. The component $\cos(\omega t)$ changes to ½ times of

$$\cos((\omega_0 - \omega)t) + \cos((\omega_0 + \omega)t) \text{ and } \sin((\omega_0 - \omega)t) - \sin((\omega_0 + \omega)t).$$

The multiplier frequency ω_0 is to be chosen equal to the intermediate frequency, so we get a base-band conversion. The unwanted high-frequency components at $\omega + \omega_0$ are to be filtered out by a low-pass filter. The resulting two sequences are called the in-phase (*I*) and the quadrature (*Q*) components. The inverse tangent of their ratios gives the phase of the received signal at the corresponding time points.

FM bandwidth for frequency discriminator demodulation

One possibility for GMSK demodulation is applying a frequency (phase) discriminator to the modulated signal. (It can be analog hardware or after an analog-to-digital conversion digital quadrature converter with phase difference calculation.) The result is the instantaneous (FM) frequency. In real-life environments there are disturbing foreign signals, like noise. Their effects can be reduced by restricting the bandwidth of the signal to the necessary minimum before the frequency discriminator, and also after it.

The necessary bandwidth before the FM discriminator can be derived from our spectrum investigations. At least the bit-frequency is needed: at GSM 270.8 KHz (±135.4 KHz), at DECT 1152 KHz, otherwise strong spectral components get lost.

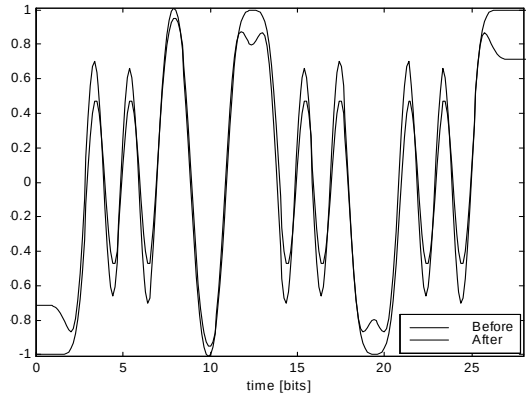
Inter-Symbol Interference

We have seen that there is a relatively large interference between the affects of individual modulation bits. A single 1 between 0 bits causes only half as much frequency

deviation as it would between 1 bits. It is usually considered disadvantageous, (and it really is at memory-less demodulation), but with some signal processing, the ISI can be reduced or it even can be used to improve the noise tolerance of the demodulation.

Shaping the FM frequency response

If $BT \leq 0.3$ the peak FM at 0101… modulation bit pattern is less than half of the maximum which occurs at a modulation with bits 111… or 000…, so a frequency offset between the transmitter and the receiver can move the threshold of the comparator too close to the FM peak of 0101… case. For an improvement (relaxing the frequency offset requirements, which is in the range of 100 Hz at GSM) the band limiting filter of the FM data could have a larger gain at half the bit frequency than at DC. It increases the level of FM components when needed, undoing the low-pass effects of the Gauss filter.



FM enhancement of a GSM signal

If the gain at even higher frequencies is also increased in the hope to get closer to the original rectangular pulse shape of the bit-stream, the noise gets amplified, too. In each practical situation one has to extend the bandwidth until the frequency where the spectral components of the signal are still larger than the noise or other disturbances. With 40 dB S/N ratio the cut off frequency must be somewhere around 250 KHz at GSM.

Demodulation based on correlation

(We use here the GSM case as an example.) Having the FM data one could cut out a sequence of 5-bit period, centered at the middle of a bit. We keep the ideal modulated FM sequences of all $2^5 = 32$ bit-quintets in a table. Leaving out the first and last half bit period, do a correlation of the received sequence with all the stored ones. The sequence with the largest correlation is used to determine the middle bit. (Only one bit out of 5.) Then shift the correlation window a bit-period further on the FM data of the received signal, and repeat the same procedure for the next demodulated bit. This way the 2 neighbors, who affected the modulation phase of the middle bit at modulation time are used to help at the demodulation, too. The ISI actually works for us.

MODULATION METHODS

In the previous section we discussed the demodulation process. It is equally important to do the modulation also fast and accurate. The modulation information is always in the digital domain, at some point it has to be converted to analog, because the transmitted signal is normally of such a high frequency that cannot be digitally generated.

We discuss two alternatives here using digital signal processing. One method generates FM samples, which are used to tune a voltage-controlled oscillator. This is the simplest, but has some drawbacks:

- the modulation distortion of the signal is determined by the analog hardware and it is much higher than what we can achieve at digitally generated signals
- the method is not applicable for other modulation schemes, like QPSK, QAM, therefore no dual mode transmitter is possible
- there is no way to manipulate the signal amplitude for compensation of known hardware frequency response, distortion.

The second method generates digital amplitude samples. They get converted to an analog signal, which is up-converted (mixed) to the RF to be transmitted. This method does not have the drawbacks of the previous one, but needs a good (expensive) analog mixer. It also has two variants as discussed below.

Intermediate Frequency Carrier

A DSP generates samples of a GMSK modulated intermediate frequency carrier signal. How to chose the intermediate frequency? It has to be mixed up to the transmitter frequency, normally in the 900 or 1800 MHz range. A mixer of real signals produces the sum and differences of two frequencies, therefore the intermediate frequency must be as high as possible to allow a cheap filter to get rid of the unwanted side-band. With inexpensive components we can achieve a few tens of MHz. To separate the two side-bands very good quality filters are needed. It is sometimes better for the up-conversion to use 2 or more steps. First mix to a second intermediate frequency, where the side-bands are not too close relative to the center frequency. The image rejection filter is cheap, but we need at least a second mixer - filter pair, too, to reach the transmitter band.

The second problem is finding the correct sampling rate. The digital to analog conversion process introduces alias frequencies, which have to be removed by an analog interpolation filter. In order to keep this filter simple, the sampling rate must be so much higher than the intermediate frequency, as possible (to get the aliases far away from the signal). The intermediate frequency itself has to be as high as possible, so to find a compromise is not easy. Also, a high sampling rate requires expensive components and more power.

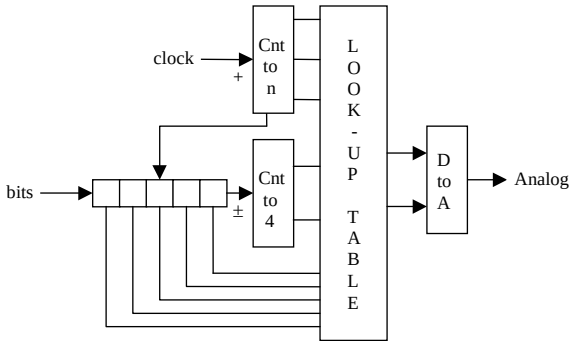
Base-Band Samples

At this method a DSP generates base-band samples. We need a complex (quadrature) signal, because the frequency spectrum is not symmetric. Using quadrature mixers there are no side-bands. (At least in theory. In praxis a -30 to -40 dB side-band signal is still there, but it causes now a distortion, which cannot be removed later.) Here the side-band separation filter is traded in the quadrature mixer, which is more complex than a real mode mixer. The generated signal is of low frequency, therefore the sampling rate can be smaller, but we need a complex signal (I and Q components), which makes the two solutions comparable in cost and complexity.

DSP of the sample generation

The frequency-samples method is simpler, because the past and future bits only have effects in a limited time period (at most 2 bits on both sides are significant). Therefore, attaching the 2 leading and 2 trailing bits to the actual one can be used to select a table. In this table there are the FM samples at different time offsets from the middle of the bit. Normally we have from 1 to 16 samples per bit, so the combined look-up table is from 32 to 512 entries long.

When direct amplitude samples are generated, one needs the phase. In a table the cosine and the sine of it is stored, maybe adjusted to compensate the distortion and frequency response effects of the hardware. Unfortunately, the phase depends on all the past bits and on a few bits in the future. But the dependence on the far past is simple. A 1-bit causes a $\pi/2$ phase shift added, a –1-bit causes it subtracted. Therefore we only need to keep the sum of the past bits modulo 4, since the trigonometric functions are periodic with 2π. The effects of the near-bits are handled similarly as in the case of frequency-samples, but we need 4 tables according to the past bits. Having some work done in hardware the table-size can be reduced, because two pairs of tables contain the same numbers, one negated. If quadrature data is stored the cosine and sine tables need to be swapped at odd bit sum values. That is, if we have a 2-bit up/down counter for the past bits and some sign handling and table selection logic we can get away with only double table-size as at the frequency-samples method.



LITERATURE

[1] Abramowitz & Stegun, *Handbook of Mathematical Functions*, Dover Publications, 1965, sec. 7.1.

[2] W. J. Cody, *Rational Chebyshev Approximations for the Error Function*, Math. Comp., 1969, pp. 631-638.

[3] W. J. Cody, *An Overview of Software Development for Special Functions*, Lecture Notes in Mathematics, 506, Numerical Analysis Dundee, 1975, G. A. Watson (ed.), Springer Verlag, Berlin, 1976.

[4] Hart, et. Al., *Computer Approximations*, John Wiley & Sons, New York, 1968.

[5] A. Ralston, *A First Course in Numerical Analysis*, McGraw-Hill, 1965.

[6] N. Kalouptsidis, *Signal Processing Systems: Theory and Design*, John Wiley & Sons, New York, 1997.

[7] Y. Akaiwa, *Introduction to Digital Mobile Communication*, J. Wiley & Sons, New York, 1997.

[8] M. K. Simon, S. M. Hinedi, W. C. Lindsey, *Digital Communication Techniques*, Prentice Hall, 1995.

[9] J. G. Proakis, *Digital Communications*, McGraw-Hill, 1989.