GFSK modulation

June 3rd, 2016

Oltan Doci

History

Date	Draft	Author	Comment
June 3rd, 2016	0	O.D	Creation
June 14th, 2016	1	O.D	Add arctan method
June 29th, 2016	2	O.D	Review + add Goertzel algorithm
October 12th, 2017	3	O.D	Add orthogonality and MSK example

Definitions: 1st part

Number of bits per symbol (M is the coding states):

$$\log_2(M)$$

For BFSK, M=2 so there is only **one bit per symbol**

Frequency f_1 is representing a "0" bit:

$$s_0(t) = e^{j(2\pi f_0 t + \varphi_0)}$$

Frequency f_2 is representing a "1" bit:

$$s_1(t) = e^{j(2\pi f_1 t + \varphi_1)}$$

Data Rate in bits/sec, where T_{sym} is the symbol duration in seconds

$$DR = \frac{\log_2(M)}{T_{sym}} = \frac{1}{T_{sym}}$$

Eg. DR = 50 kbps:

$$T_{sym} = \frac{1}{DR} = 20\mu s$$

Bandwidth (spectral) Efficiency in bits/sec/Hz, where BW is the occupied band around the carrier:

$$\eta = \frac{DR}{BW}$$

Orthogonality: inner product

- Symbols inner product can be written as:

$$
\begin{aligned}
\langle s_0 | s_1 \rangle &= \int_0^{T_{sym}} s_0^*(t) \cdot s_1(t) \cdot dt \\
&= \int_0^{T_{sym}} e^{j(2\pi f_1 t + \varphi_1)} \cdot e^{-j(2\pi f_0 t + \varphi_0)} \cdot dt \\
&= \int_0^{T_{sym}} e^{j(2\pi (f_1 - f_0) t + \varphi_1 - \varphi_0)} \cdot dt \\
&= \frac{1}{j2\pi(f_1 - f_0)} \cdot \left[e^{j(2\pi (f_1 - f_0) t + \varphi_1 - \varphi_0)} \right]_0^{T_{sym}} \\
&= \frac{e^{j(2\pi (f_1 - f_0) T_{sym} + \varphi_1 - \varphi_0)} - e^{j(\varphi_1 - \varphi_0)}}{j2\pi(f_1 - f_0)} \\
&= \frac{e^{j(\pi (f_1 - f_0) T_{sym} + \varphi_1 - \varphi_0)} \cdot \left(e^{j\pi (f_1 - f_0) T_{sym}} - e^{-j\pi (f_1 - f_0) T_{sym}} \right)}{j2\pi(f_1 - f_0)} \\
&= \frac{e^{j(\pi (f_1 - f_0) T_{sym} + \varphi_1 - \varphi_0)} \cdot 2j \sin\left(\pi (f_1 - f_0) T_{sym} \right)}{j2\pi(f_1 - f_0)} \\
&= e^{j(\pi (f_1 - f_0) T_{sym} + \varphi_1 - \varphi_0)} \cdot T_{sym} \cdot \mathrm{sinc}(\pi (f_1 - f_0) T_{sym})
\end{aligned}
$$

Orthogonality: non coherent case

- Non coherent conditions: phases are unknown => need to take the magnitude

$$\left|\left\langle s_0 \mid s_1 \right\rangle\right| = \left| e^{j(\pi(f_1 - f_0)T_{sym} + \varphi_1 - \varphi_0)} \cdot T_{sym} \cdot \mathrm{sinc}(\pi(f_1 - f_0)T_{sym}) \right|$$

$$= T_{sym} \cdot \left| \mathrm{sinc}(\pi(f_1 - f_0)T_{sym}) \right|$$

$$= \begin{cases} T_{sym} \ if \ \ f_0 = f_1 \\ 0 \ if \ \ f_0 \neq f_1 \ and \ \pi(f_1 - f_0)T_{sym} = k\pi, k \in \mathbb{Z}^* \end{cases}$$

$$\Rightarrow \frac{f_1 - f_0}{DR} = k$$

Orthogonality: coherent case

- Coherent conditions: phases are known
  => need to take only the real part

$$\mathrm{Re}\left(\left\langle s_0 \middle| s_1 \right\rangle\right) = \mathrm{Re}\left(\frac{e^{j(2\pi(f_1-f_0)T_{sym}+\varphi_1-\varphi_0)} - e^{j(\varphi_1-\varphi_0)}}{j2\pi(f_1-f_0)}\right)$$

$$= \mathrm{Re}\left(\frac{-j\left(\cos\left(2\pi(f_1-f_0)T_{sym}+\varphi_1-\varphi_0\right) + j\sin\left(2\pi(f_1-f_0)T_{sym}+\varphi_1-\varphi_0\right) - \cos\left(\varphi_1-\varphi_0\right) - j\sin\left(\varphi_1-\varphi_0\right)\right)}{2\pi(f_1-f_0)}\right)$$

$$= \frac{\sin\left(2\pi(f_1-f_0)T_{sym}+\varphi_1-\varphi_0\right) - \sin\left(\varphi_1-\varphi_0\right)}{2\pi(f_1-f_0)}$$

- Because of known phases, with same phases at origin, we get:

$$\mathrm{Re}\left(\left\langle s_0 \middle| s_1 \right\rangle\right) = \frac{\sin\left(2\pi(f_1-f_0)T_{sym}\right)}{2\pi(f_1-f_0)}$$

$$= T_{sym} \cdot \mathrm{sinc}\left(2\pi(f_1-f_0)T_{sym}\right)$$

$$= \begin{cases} T_{sym} \ if \ \ f_0 = f_1 \\ 0 \ if \ \ f_0 \neq f_1 \ and \ 2\pi(f_1-f_0)T_{sym} = k\pi, k \in \mathbb{Z}^* \end{cases}$$

$$\Rightarrow \frac{f_1-f_0}{DR} = \frac{k}{2}$$

- This allows an additional orthogonality case k/2 = ½: this is called MSK (ex GSM)

Definitions: 2nd part

Higher carrier frequency: MARK, logic “1”, so $f_1 = +\Delta f = f_{MARK}$

Lower carrier frequency: SPACE, logic “0”, so $f_0 = -\Delta f = f_{SPACE}$

Frequency **deviation** in Hz:

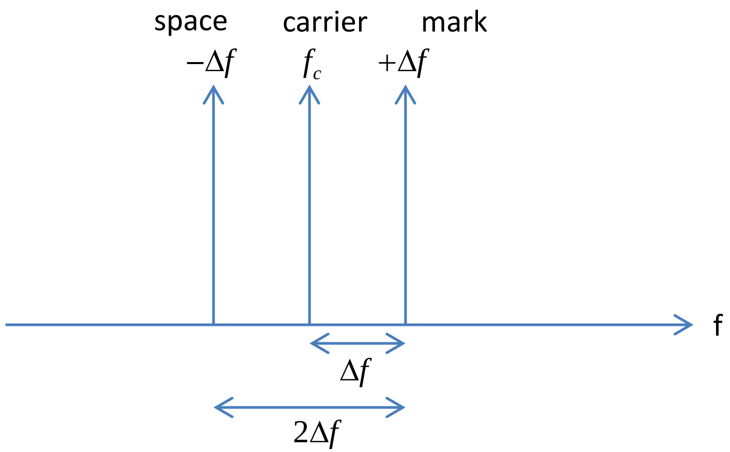
Δf : it is the deviation from the carrier

Frequency **separation** in Hz:

$$\left| f_{MARK} - f_{SPACE} \right| = \Delta f - (-\Delta f)$$
$$= 2 \cdot \Delta f$$

Modulation index:

$$m = \frac{\left| f_{MARK} - f_{SPACE} \right|}{DR}$$
$$= \frac{2 \cdot \Delta f}{DR}$$



Modulation

- Complex signal:

$$s_n = e^{j2\pi f_n + \varphi_n} \quad 0 \le n \le M-1$$

- Real signal (modulated)

$$\begin{aligned} s_{BFSK}(t) &= \mathrm{Re}(s_n \bullet e^{j(2\pi f_c t + \theta)}) \\ &= \mathrm{Re}(e^{j(2\pi f_c t + 2\pi f_n t + \varphi_n + \theta)}) \\ &= \cos(2\pi f_c t + 2\pi f_n t + \varphi_n + \theta) \end{aligned}$$

With $\theta_n = \varphi_n + \theta$:

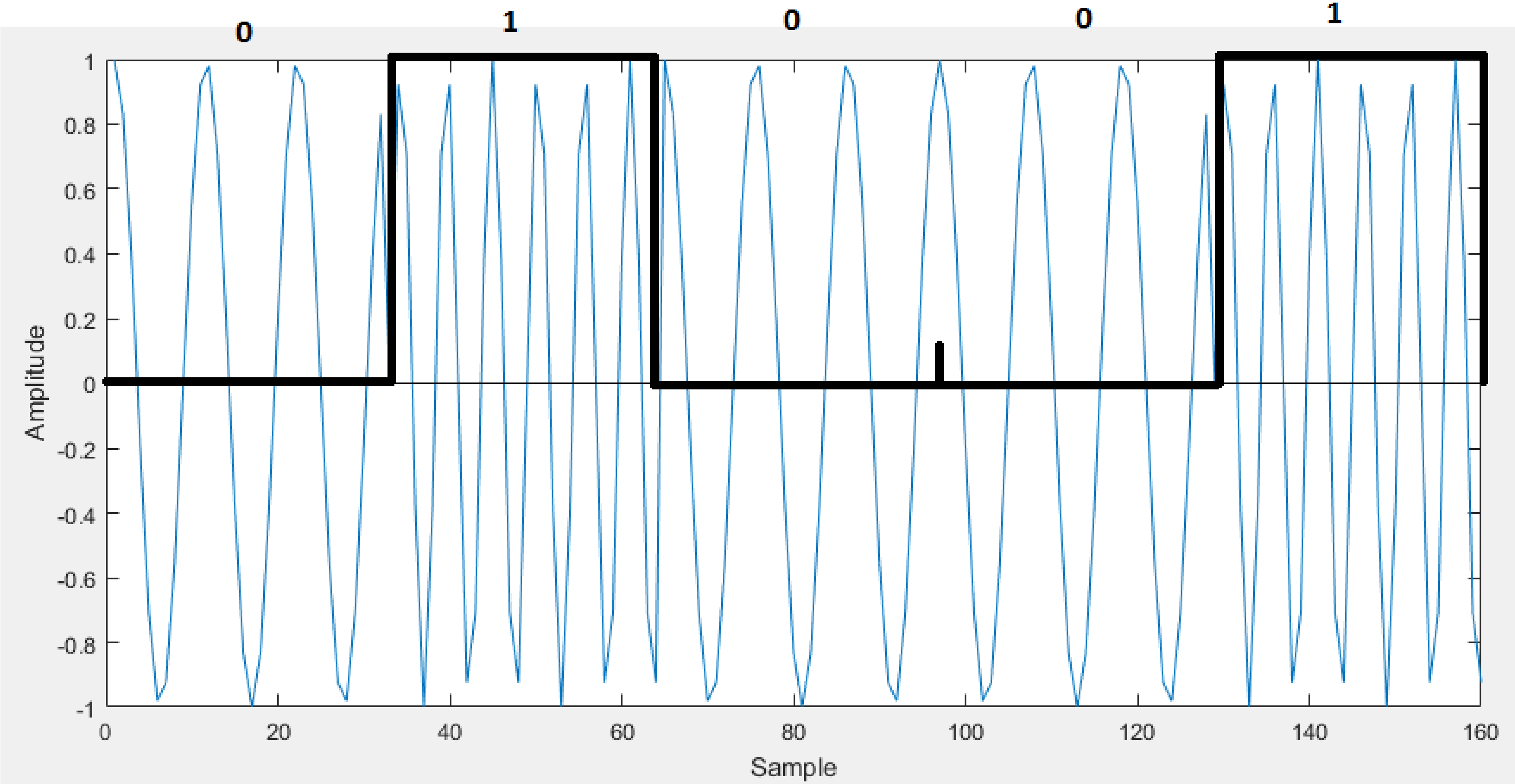
$$\begin{aligned} s_{BFSK}(t) &= \cos(2\pi f_c t) \bullet \cos(2\pi f_n t + \theta_n) - \sin(2\pi f_c t) \bullet \sin(2\pi f_n t + \theta_n) \\ &= I(t) \bullet \cos(2\pi f_c t) + Q(t) \bullet \left[-\sin(2\pi f_c t) \right] \end{aligned}$$

Where:

$$I(t) = \cos(2\pi f_n t + \theta_n)$$

$$Q(t) = \sin(2\pi f_n t + \theta_n)$$

BFSK example



Modulation

- To each bit level corresponds a frequency
- To avoid side band power which occurs during the transition between 0/1 bits, the bistream signal is filtered by a Gaussian lowpass filter: this is called pulse shaping
- The phase trajectory of the FSK signal which is smoothed by the Gaussian filtering is continuous: this technique is called CPFSK
- The I/Q samples can be generated by a CORDIC algorithm, or by using a phase accumulator with a LUT containing SIN/COS values, …
- In this document we will use the LUT SIN/COS principle

Gaussian filter

The gaussian filtering will “round” the corners of NRZ signal in order to smooth phase trajectory. The input parameter of the filter is BT.
It is defined below:

$$\sigma = \frac{\sqrt{l\,\mathrm{og}(2)}}{2\pi \frac{BT}{T_b}}$$

$$delay : default = 3$$

$$samples = round\left(\frac{delay \bullet \sigma}{T_s}\right)$$

$$t_{delay} = samples \bullet T_s$$

$$t = \left[0, T_s, 2T_s, 3T_s, \ldots, 2 \bullet samples \bullet T_s\right]$$

$$h = \frac{1}{\sigma \bullet \sqrt{2\pi}} \bullet e^{-\frac{(t - t_{delay})^2}{2\sigma^2}}$$

$$x : input$$

$$y_{filtered} = conv(x, h) \bullet T_s$$

Derivative function

- The derivative of x function respect to time t is written:

$$\frac{dx}{dt}(t_0) = \lim_{t->t_0} \frac{x(t)-x(t_0)}{t-t_0}$$

- In digital we will define the derivative discrete function by the following 2 approximations:

$$\frac{dx_k}{dt_k} \approx \frac{\Delta x}{\Delta t}$$

$$= \frac{x(k)-x(k-1)}{t_k - t_{k-1}} \qquad \textbf{(1)}$$

$$\frac{dx_k}{dt_k} \approx \frac{\Delta x}{\Delta t}$$

$$= \frac{x(k+1)-x(k-1)}{t_{k+1} - t_{k-1}} \qquad \textbf{(2)}$$

- In this presentation we will use the definition **(1)**, i.e. the difference taken between 2 consecutive points

Phase accumulator (floating)

- The instantaneous frequency can be found via the phase:

$$\omega = 2\pi f = \frac{d\phi}{dt}$$

$$\Rightarrow f = \frac{1}{2\pi} \cdot \frac{d\phi}{dt}$$

- One gets:

$$d\phi = 2\pi f dt$$

So: $\Delta\phi = 2\pi f \Delta t$

$$\phi_k - \phi_{k-1} = 2\pi f (t_k - t_{k-1})$$

With: $t_{sampling} = \frac{1}{f_{sampling}} = t_k - t_{k-1}$

So: $\frac{\phi_k}{2\pi} = \frac{\phi_{k-1}}{2\pi} + \frac{f}{f_{sampling}}$

Phase accumulator (floating)

- Simplified expression:

$$\phi \leftrightarrow \frac{\phi}{2\pi}$$
$$\Rightarrow \phi_k = \phi_{k-1} + \frac{f}{f_{sampling}}$$
$$\phi_k \in [0;1]$$

- For each increment one gets sin/cos values :

$$\phi \leftrightarrow \frac{\phi}{2\pi}$$
$$\Rightarrow \{\sin(2\pi\phi_k), \cos(2\pi\phi_k)\}$$

- Frequency resolution of the phase accumulator:

$$resolution = \frac{f_{sampling}}{2^n}$$

Phase accumulator (quantified)

- Digital quantification with n bits:

$$\overline{\phi_k} = \overline{\phi_{k-1}} + \frac{f}{f_{sampling}} \bullet 2^n$$

$$\overline{\phi_k} \in \left[0; 2^n \right]$$

- The issue is the big size of the LUT of 2^n because:

$$\cos(2\pi\phi_k) = \cos\left(\frac{2\pi\overline{\phi_k}}{2^n} \right)$$

- So we need a LUT of 2^n for SIN and a LUT of 2^n for COS values. To resolve this issue we can reduce a lot the LUT size. Let's rewrite LUT SIN/COS formulas:

$$\cos(2\pi\phi_k) = \cos\left(\frac{2\pi\overline{\phi_k}}{2^n} \right) = \cos\left(\frac{\frac{2\pi\overline{\phi_k}}{2^p}}{\frac{2^n}{2^p}} \right) = \cos\left(\frac{2\pi\frac{\overline{\phi_k}}{2^p}}{2^{n-p}} \right)$$

With: $W = n - p$

$$\Rightarrow \cos(2\pi\phi_k) = \cos\left(\frac{2\pi\frac{\overline{\phi_k}}{2^p}}{2^W} \right)$$

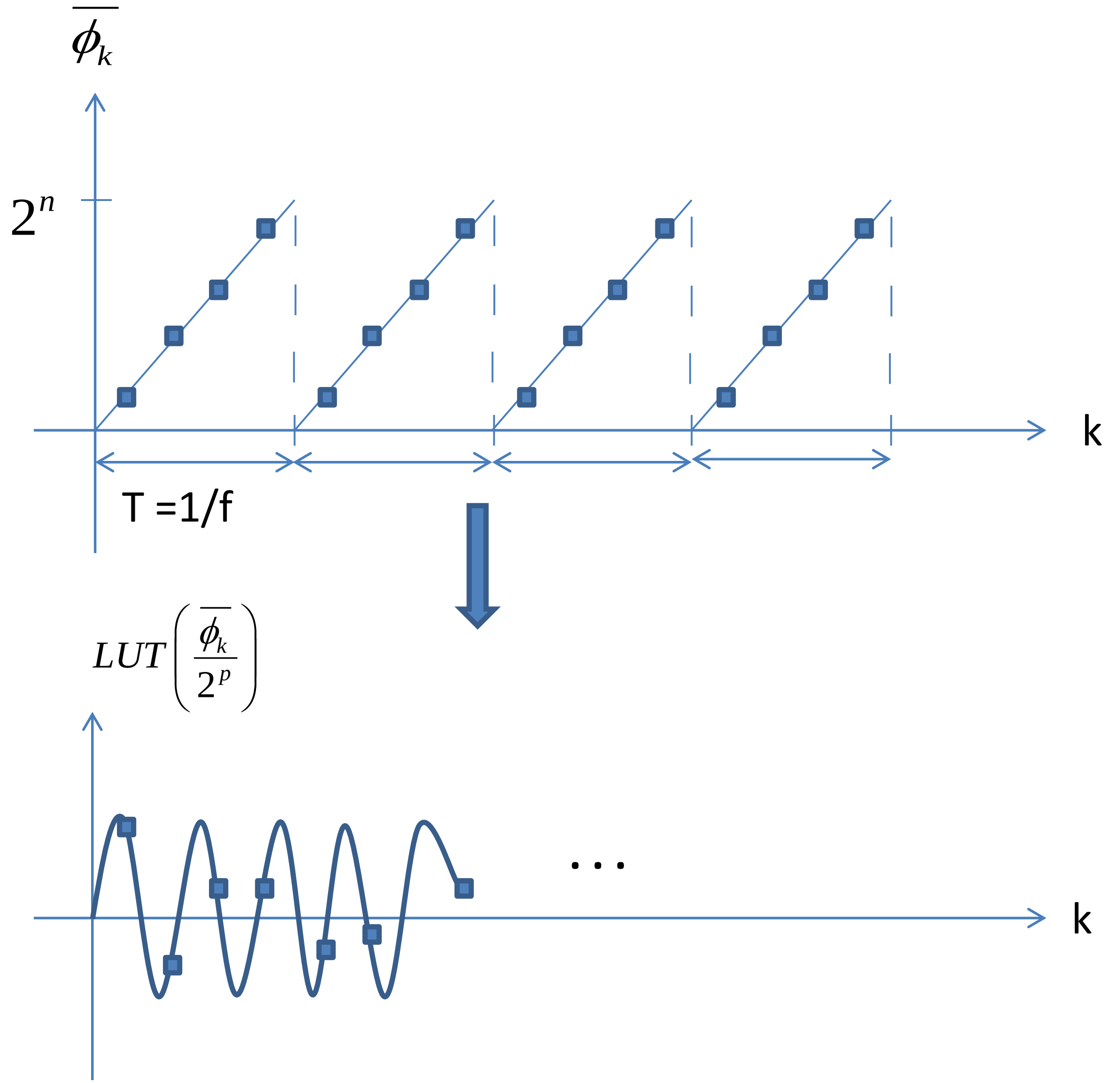
$$\Rightarrow \sin(2\pi\phi_k) = \sin\left(\frac{2\pi\frac{\overline{\phi_k}}{2^p}}{2^W} \right)$$

- The LUT size is divided by 2^p

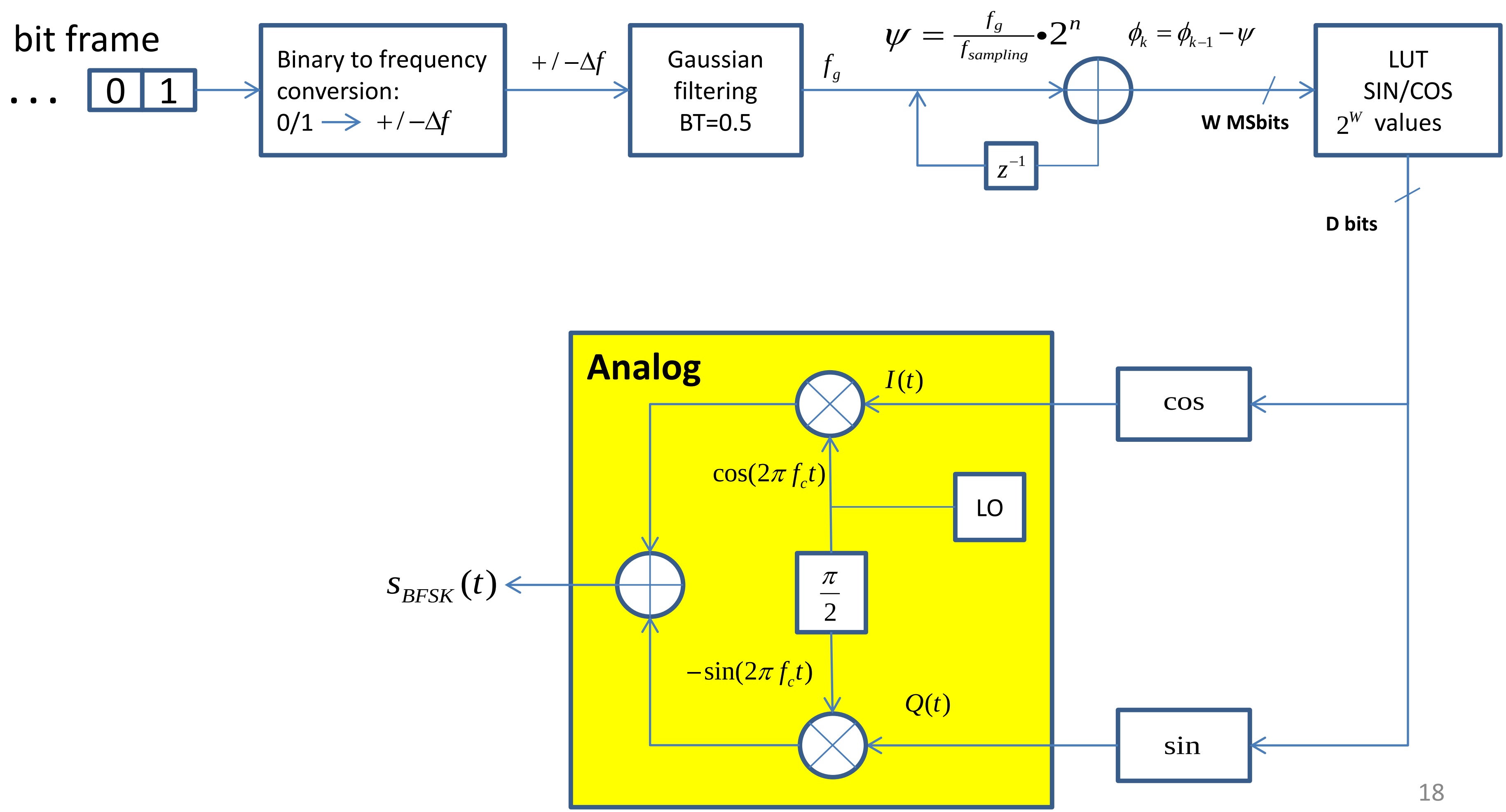
Phase accumulator (quantified)

- For each phase increment, take only the W MSbits of the phase
- This is exactly the index in the LUT for sin/cos values
- LUT length is 2^W
- LUT sin/cos values can be quantified by D bits (signed)

Phase accumulator with SIN/COS LUT



BFSK modulator proposal



LUT creation

- SIN/COS:

$$LUT_{\cos}(i) = round((2^{D-1}-1) \bullet \cos\left(\frac{2\pi i}{2^W}\right)) \quad i \in \left\{0; 2^W - 1\right\}$$

$$LUT_{\sin}(i) = round((2^{D-1}-1) \bullet \sin\left(\frac{2\pi i}{2^W}\right)) \quad i \in \left\{0; 2^W - 1\right\}$$

- Eg 1: W = 12 bits, D = 16 bits: LUT size = 2^W = 4096

$$LUT_{\cos}(i) = round(32767 \bullet \cos\left(\frac{2\pi i}{4096}\right)) \quad i = 0,1,\ldots,4095$$

$$LUT_{\sin}(i) = round(32767 \bullet \sin\left(\frac{2\pi i}{4096}\right)) \quad i = 0,1,\ldots,4095$$

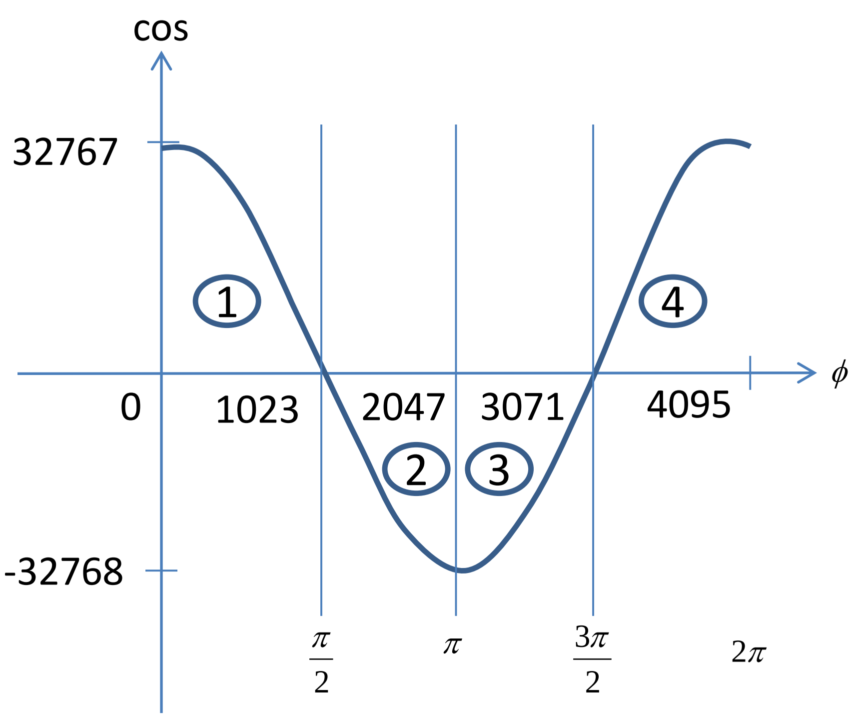
- Eg 2: W = 12 bits, D = 16 bits: LUT size = $\frac{2^W}{4}$ = 1024 => optimized size

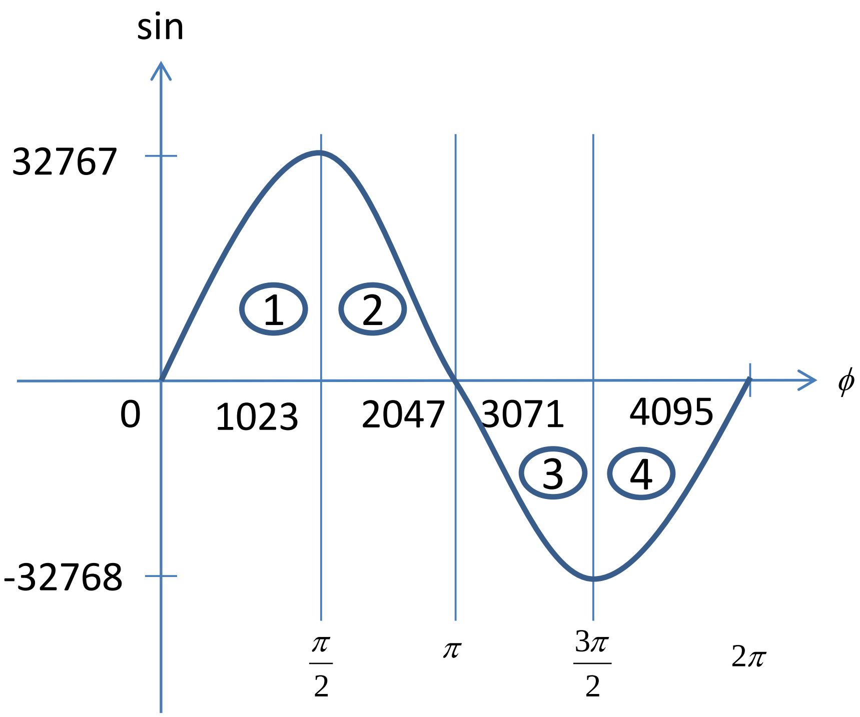
$$LUT_{\cos}(i) = round(32767 \bullet \cos\left(\frac{2\pi i}{4096}\right)) \quad i = 0,1,\ldots,1023$$

$$LUT_{\sin}(i) = round(32767 \bullet \sin\left(\frac{2\pi i}{4096}\right)) \quad i = 0,1,\ldots,1023$$

LUT SIN/COS with optimized size

- LUT is composed only by quadrant 1 (1024 values):





LUT SIN/COS with optimized size

LUT cos:

$$0 \le \phi < 1024 \Rightarrow \cos(\phi)$$
$$1024 \le \phi < 2048 \Rightarrow -\cos(2048 - \phi)$$
$$2048 \le \phi < 3072 \Rightarrow -\cos(\phi - 2048)$$
$$3072 \le \phi < 4096 \Rightarrow \cos(4096 - \phi)$$

LUT sin:

$$0 \le \phi < 1024 \Rightarrow \sin(\phi)$$
$$1024 \le \phi < 2048 \Rightarrow \sin(2048 - \phi)$$
$$2048 \le \phi < 3072 \Rightarrow -\sin(\phi - 2048)$$
$$3072 \le \phi < 4096 \Rightarrow -\sin(4096 - \phi)$$

If we want we can optimize again, by creating only one LUT (COS for example) and deduce the second LUT via translations :

$$\sin(a) = \cos\left(a - \frac{\pi}{2}\right)$$

It is not done in this presentation.

Phase accumulator implementation

- The phase accumulator builds sin/cos signals based on the current phase
- Frequency resolution:

$$\frac{f_{sampling}}{2^n}$$

- Simple implementation of a countdown phase accumulator:

Global initialization:

$$\phi_{curr} = 0$$

$$\phi_{prev} = 0$$

For each frequency sample (Gaussian filtering output):

$$\phi_{curr} = \phi_{prev} - \psi$$

if $\phi_{curr} < 0$

then $\phi_{prev} = \phi_{curr} + 2^n$

else if $\phi_{curr} > 2^n$

then $\phi_{prev} = \phi_{curr} - 2^n$

else $\phi_{prev} = \phi_{curr}$

Phase accumulator example

Take W MSbits of the phase:

$$\phi = \phi_{prev} >> p \Leftrightarrow \phi = floor\left(\frac{\phi_{prev}}{2^p}\right)$$

Where p = n – W

E.g:

$$n = 24$$
$$W = 12$$
$$f_{sampling} = 2MHz$$
$$\phi \in \left[0; 2^{24} - 1\right]$$
$$resolution = \frac{2MHz}{2^{24}} = 119mHz$$
$$p = n - W = 12$$
$$LUT_{idx} \in \left[0; 2^{24-12} - 1\right] = [0; 4095]$$

Trigonometric formulas

Some useful formulas:

$$\cos^2 a + \sin^2 a = 1$$

$$\tan a = \frac{\sin a}{\cos a}$$

$$\cos(a+b) = \cos a \bullet \cos b - \sin a \bullet \sin b$$

$$\cos(a-b) = \cos a \bullet \cos b + \sin a \bullet \sin b$$

$$\sin(a+b) = \sin a \bullet \cos b + \cos a \bullet \sin b$$

$$\sin(a-b) = \sin a \bullet \cos b - \cos a \bullet \sin b$$

$$\tan(a+b) = \frac{\tan a + \tan b}{1 - \tan a \tan b}$$

$$\tan(a-b) = \frac{\tan a - \tan b}{1 + \tan a \tan b}$$

$$\cos 2a = \cos^2 a - \sin^2 a = 2\cos^2 a - 1 = 1 - 2\sin^2 a$$

$$\sin 2a = 2\sin a \bullet \cos a$$

$$\tan 2a = \frac{2\tan a}{1 - \tan^2 a}$$

$$\cos^2 a = \frac{1 + \cos 2a}{2}$$

$$\sin^2 a = \frac{1 - \cos 2a}{2}$$

$$\tan^2 a = \frac{1 - \cos 2a}{1 + \cos 2a}$$

Non-coherent demodulation

- On the receiver side the signal passes trough SIN/COS oscillator:

$$s_{RX}(t) = s_{BFSK}(t)$$
$$= I(t) \bullet \cos(2\pi f_c t) + Q(t) \bullet \left[-\sin(2\pi f_c t)\right]$$

I channel:

$$s_I(t) = s_{RX}(t) \bullet \cos(2\pi f_c t)$$
$$= (I(t) \bullet \cos(2\pi f_c t) + Q(t) \bullet \left[-\sin(2\pi f_c t)\right]) \bullet \cos(2\pi f_c t)$$
$$= I(t) \bullet \cos^2(2\pi f_c t) - Q(t) \bullet \sin(2\pi f_c t) \bullet \cos(2\pi f_c t)$$
$$= \frac{I(t) \bullet (1 + \cos(2\pi \bullet 2 f_c t))}{2} - \frac{Q(t) \bullet \sin(2\pi \bullet 2 f_c t)}{2}$$

Q channel:

$$s_Q(t) = s_{RX}(t) \bullet \sin(2\pi f_c t)$$
$$= (I(t) \bullet \cos(2\pi f_c t) + Q(t) \bullet \left[-\sin(2\pi f_c t)\right]) \bullet \sin(2\pi f_c t)$$
$$= I(t) \bullet \cos(2\pi f_c t) \bullet \sin(2\pi f_c t) - Q(t) \bullet \sin^2(2\pi f_c t)$$
$$= \frac{I(t) \bullet \sin(2\pi \bullet 2 f_c t)}{2} - \frac{Q(t) \bullet (1 - \cos(2\pi \bullet 2 f_c t))}{2}$$

Non-coherent demodulation

- After filtering:

$$s_I(t) = \frac{I(t)}{2}$$
$$s_Q(t) = -\frac{Q(t)}{2}$$

- In presence of frequency offset, noise and channel the received signal becomes:

$$s_{RX}(t) = a \bullet s_{BFSK}(t - t_0) \bullet e^{j(2\pi f_{offset} t + \gamma)} + w(t)$$

The signal is attenuated by a and shifted in time by t_0 due to the channel
There is also a frequency shift by f_{offset} due to the drift
We consider that white Gaussian noise w is added to the signal

With: $f_{RX} = f_n + f_{offset}$

And: $\theta_{RX} = \theta_n + \gamma$

One gets: $I(t) = a \bullet \cos(2\pi f_{RX}(t - t_0) + \theta_{RX}) + w(t)$

$$Q(t) = a \bullet \sin(2\pi f_{RX}(t - t_0) + \theta_{RX}) + w(t)$$

Non-coherent demodulation

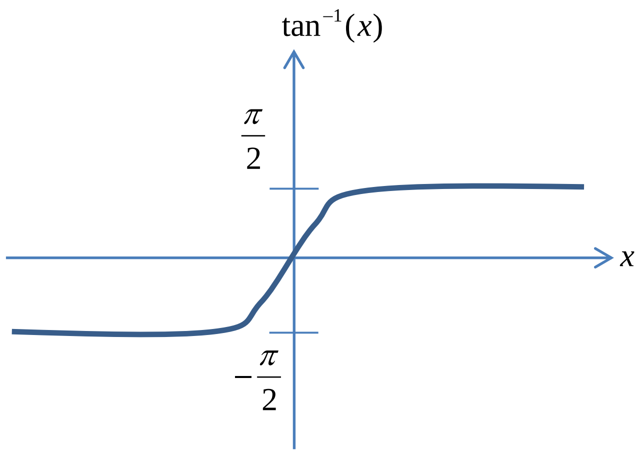
- Because of the shift due to the frequency offset the demodulator can’t decode bits correctly
- In order to remove the frequency offset on the receiver side, I/Q samples must be corrected by this frequency offset so in the time domain I/Q samples are multiplied by a complex oscillator running at frequency offset.
- This frequency offset estimation is handled on the preamble. The idea is to compute the frequency offset using the demodulated preamble signal, so it is the mean value of frequency samples.
- The time offset is also handled on the preamble. The idea is to compute a cross-correlation with the reference preamble NRZ. The presence of a peak indicates the right timing index.

Arctan method

We have: $\omega = 2\pi f = \frac{d\phi}{dt}$

So: $f = \frac{1}{2\pi} \bullet \frac{\Delta\phi}{\Delta t}$

Where: $\Delta t = t_k - t_{k-1} = T_{sampling} = \frac{1}{f_{sampling}}$



So the frequency deviation is deduced by:

$$\Delta f = \frac{f_{sampling}}{2\pi} \bullet \Delta\phi$$

Where: $\Delta\phi = \phi_k - \phi_{k-1}$

And: $\phi = \arg\left(s_I + js_Q\right) = \tan^{-1}\left(\frac{s_Q}{s_I}\right) \in \left]-\frac{\pi}{2};\frac{\pi}{2}\right[$

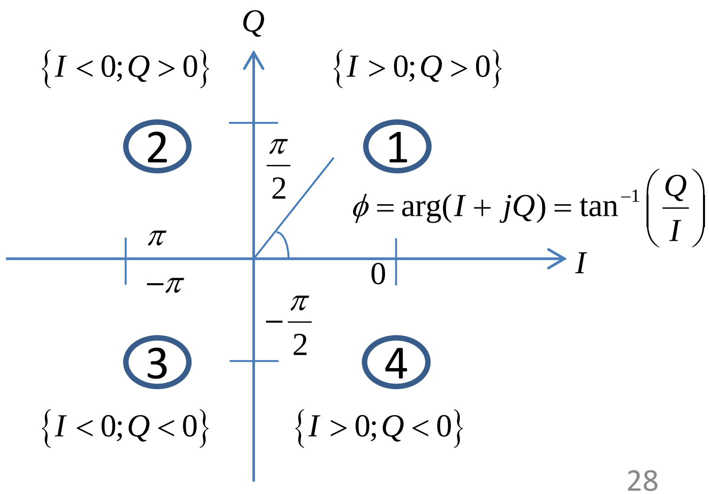
The arctan function returns values within $\left]-\frac{\pi}{2};\frac{\pi}{2}\right[$ so the principal argument must be managed by the following way in order to be within $]-\pi;\pi[$:

$q1 \Rightarrow \phi$

$q2 \Rightarrow \phi = \phi + \pi$

$q3 \Rightarrow \phi = \phi - \pi$

$q4 \Rightarrow \phi$



Arctan method

- Because of phase jump, this formula will produce glitches when computing phase derivate.

- The implementation must handle the phase jump by using an “unwrap” function in order to smooth the phase trajectory.

Differential method

- Based on the previous phase formula one gets:

$$f = \frac{1}{2\pi} \bullet \frac{d\phi}{dt}$$

$$= \frac{1}{2\pi} \bullet \frac{d\left(\tan^{-1}\left(\frac{Q}{I}\right)\right)}{dt}$$

$$= \frac{1}{2\pi} \bullet \frac{1}{1+\left(\frac{Q}{I}\right)^2} \frac{d}{dt}\left(\frac{Q}{I}\right)$$

$$= \frac{1}{2\pi} \bullet \frac{\dfrac{\frac{\Delta Q}{\Delta t} \bullet I - Q \bullet \frac{\Delta I}{\Delta t}}{I^2}}{1+\dfrac{Q^2}{I^2}}$$

$$= \frac{1}{2\pi} \bullet \frac{1}{\Delta t} \bullet \frac{\Delta Q \bullet I - Q \bullet \Delta I}{I^2 + Q^2}$$

Where: $\Delta t = t_k - t_{k-1} = T_{sampling} = \dfrac{1}{f_{sampling}}$

$$\Delta I = I_k - I_{k-1}$$

$$\Delta Q = Q_k - Q_{k-1}$$

Differential method

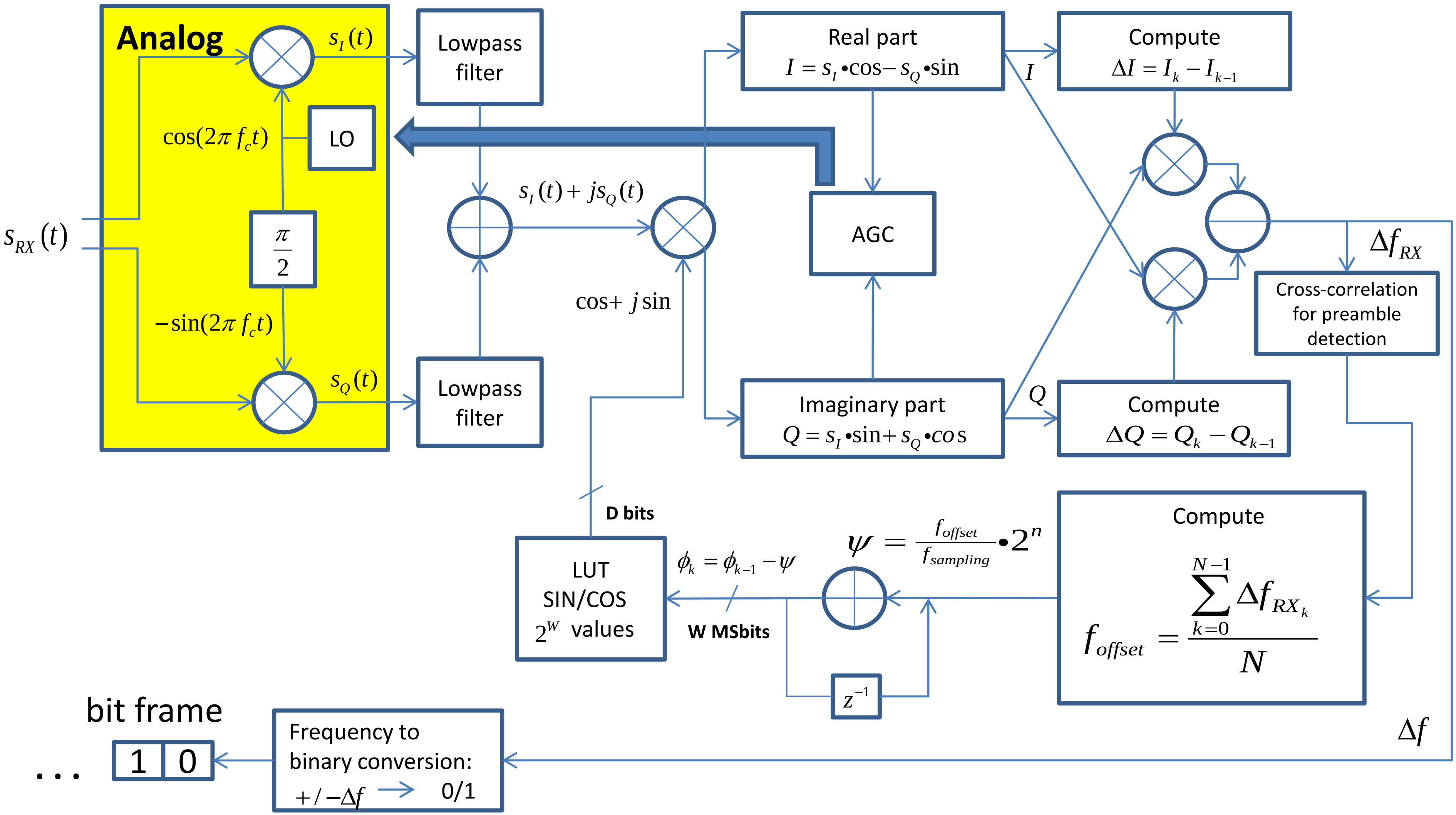
- So the frequency deviation is deduced by:

$$\Delta f = \frac{f_{sampling}}{2\pi} \bullet \frac{\Delta Q \bullet I - Q \bullet \Delta I}{I^2 + Q^2}$$

- This method is more simple to implement than the arctan method (CORDIC algorithm for arctan and phase jump handling), even if mathematically these 2 formulas are the same

- Physically we will run simulations for these both methods and compare the received processing gain vs input SNR

BFSK demodulator proposal



… other FSK demodulators

- In the literature we can find several FSK demodulators which are different than the two demodulators detailed previously

- They can be based on correlation, Kalman filtering, Goertzel algorithm, …

- Let’s study the demodulator using the Goertzel algorithm

Goertzel algorithm

DFT

Starting with DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}}$$

$$k = 0, 1, \ldots, N-1$$

With: $e^{j2\pi k \frac{N}{N}} = e^{j2\pi k} = 1$

One gets:

$$X(k) = e^{j2\pi k \frac{N}{N}} \bullet \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}}$$

$$= \sum_{n=0}^{N-1} x(n) e^{j2\pi \frac{k}{N} \bullet (N-n)}$$

Convolution

- X(k) can be interpreted as a discrete convolution of the finite-duration sequence x(n) with the sequence $e^{j2\pi k\frac{n}{N}}$ where $n = 0,1,\ldots,N-1$

- Consequently, the response to a system with impulse response $h(n) = e^{j2\pi k\frac{n}{N}}$ is:

$$\begin{aligned} y(N) &= x(N) * h(N) \\ &= \sum_{n=0}^{N-1} x(n) e^{j2\pi\frac{k}{N}\bullet(N-n)} \\ &= X(k) \end{aligned}$$

- So X(k) is the DFT value of output when **n = N**

Physical interpretation

- The Z transform is:

$$X(z) = Y(z) \bullet H(z)$$

With: $$H(Z) = \sum_{n=-\infty}^{+\infty} e^{j2\pi k\frac{n}{N}} \bullet Z^{-n}$$

$$= \sum_{n=-\infty}^{+\infty} \left(e^{j2\pi\frac{k}{N}} \bullet Z^{-1} \right)^{n}$$

$$= \frac{1}{1 - e^{j2\pi\frac{k}{N}} \bullet Z^{-1}}$$

Where:

$$\left| e^{j2\pi\frac{k}{N}} \bullet Z^{-1} \right| < 1 \Leftrightarrow |Z| > 1$$

Physical interpretation

- Difference equation:

$$Y(Z) = \frac{X(Z)}{1 - e^{j2\pi\frac{k}{N}} \bullet Z^{-1}}$$

$$Y(Z) - e^{j2\pi\frac{k}{N}} \bullet Z^{-1} Y(Z) = X(Z)$$

$$\Rightarrow y(n) = e^{j2\pi\frac{k}{N}} y(n-1) + x(n)$$

- The system can be viewed as a IIR filter
- Since the input x(n) is complex, the computation of each new y(n) requires 4 real multiplications.
- It is possible to optimize the system function.

Physical interpretation

- Let's simplify the IIR expression:

$$H(Z) = \frac{1}{1 - e^{j2\pi\frac{k}{N}} \bullet Z^{-1}}$$

$$= \frac{1 - e^{-j2\pi\frac{k}{N}} \bullet Z^{-1}}{1 - e^{-j2\pi\frac{k}{N}} \bullet Z^{-1}} \bullet \frac{1}{1 - e^{j2\pi\frac{k}{N}} \bullet Z^{-1}}$$

$$= \frac{1 - e^{-j2\pi\frac{k}{N}} \bullet Z^{-1}}{1 - 2\cos\left(2\pi\frac{k}{N}\right) \bullet Z^{-1} + Z^{-2}}$$

$$= H_1(Z) \bullet H_2(Z)$$

Where: $$H_1(Z) = \frac{1}{1 - 2\cos\left(2\pi\frac{k}{N}\right) \bullet Z^{-1} + Z^{-2}}$$

$$H_2(Z) = 1 - e^{-j2\pi\frac{k}{N}} \bullet Z^{-1}$$

Physical interpretation

- The 1st stage can be observed to be a 2nd-order IIR filter:

$$H_1(Z) = \frac{1}{1 - 2\cos\left(2\pi \frac{k}{N}\right) \bullet Z^{-1} + Z^{-2}}$$

$$\Rightarrow y_1(n) = 2\cos\left(2\pi \frac{k}{N}\right) \bullet y_1(n-1) - y_1(n-2) + x_1(n)$$ Where: $n = 0, 1, \ldots, N-1$ **: Eq 1**

- We know that only the output value $y_1(N)$ evaluated on n = N is used for computing DFT, so one gets:

$$y_1(N) = 2\cos\left(2\pi \frac{k}{N}\right) \bullet y_1(N-1) - y_1(N-2) + x(N)$$

With:

$$x(N) = X(N) = 0$$

So: $$y_1(N) = 2\cos\left(2\pi \frac{k}{N}\right) \bullet y_1(N-1) - y_1(N-2)$$ **: Eq 2**

- The 2nd stage can be observed to be a FIR filter:

$$H_2(Z) = 1 - e^{-j2\pi \frac{k}{N}} \bullet Z^{-1}$$

$$\Rightarrow y_2(n) = x_2(n) - e^{-j2\pi \frac{k}{N}} \bullet x_2(n-1)$$

So: $$y_2(N) = y_1(N) - e^{-j2\pi \frac{k}{N}} \bullet y_1(N-1)$$ **: Eq 3**

Power spectral density

Tone frequency: $X(k) = y_2(N)$

$$= y_1(N) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1)$$

Power: $P = X(k) \bullet \overset{*}{X}(k)$

$$= y_2(N) \bullet \overset{*}{y_2}(N)$$

$$= (y_1(N) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1)) \bullet (\overset{*}{y_1}(N) - e^{+j2\pi\frac{k}{N}} \bullet \overset{*}{y_1}(N-1))$$

$$= y_1(N) \bullet \overset{*}{y_1}(N) + y_1(N-1) \bullet \overset{*}{y_1}(N-1) - e^{+j2\pi\frac{k}{N}} \bullet \overset{*}{y_1}(N-1) \bullet y_1(N) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y_1}(N)$$

$$= \left|y_1(N)\right|^2 + \left|y_1(N-1)\right|^2 - e^{+j2\pi\frac{k}{N}} \bullet \overset{*}{y_1}(N-1) \bullet y_1(N) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y_1}(N)$$

Let’s write: $z = e^{+j2\pi\frac{k}{N}} \bullet \overset{*}{y_1}(N-1) \bullet y_1(N)$

So: $\overset{*}{z} = e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y_1}(N)$

Finally: $P = \left|y_1(N)\right|^2 + \left|y_1(N-1)\right|^2 - \left(z + \overset{*}{z}\right)$

$$= \left|y_1(N)\right|^2 + \left|y_1(N-1)\right|^2 - 2 \bullet \Re(z)$$ **: Eq 4**

Power spectral density

- So to detect the 2 desired frequencies at $+\Delta f$ and $-\Delta f$, we need 2 Goertzel filters:

  - one for computing of $+\Delta f$, i.e. index k
  - one for computing of $-\Delta f$, i.e. index N – k

- Finally we take the maximum of the 2 power values:

  - if the maximum corresponds to the 1st filter, then it's a $+\Delta f$
  - or the maximum is the value of the 2nd filter, and in this case it's a $-\Delta f$

Algorithm study

- The algorithm can be completed as follows:

  - after preamble synchronization, acquire the 1st burst composed by N points time domain of I/Q samples

  - ***step 1***: set initial conditions: $y_1(-1) = y_1(-2) = 0$

  - ***step 2:*** terminate **(Eq 1)** IIR filter after N iterations from 0 to N-1

  - ***step 3:*** apply **(Eq 2)** to construct $y_1(N)$ from the prior outputs $y_1(N-1)$ and $y_1(N-2)$

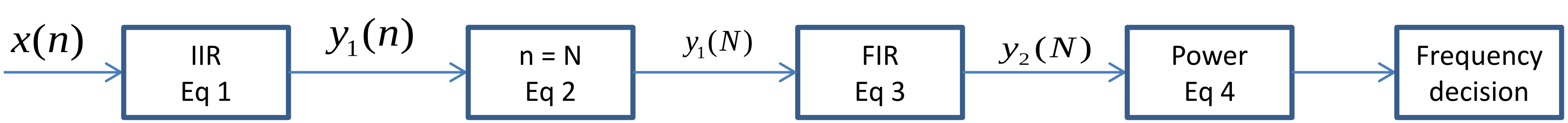
  - ***step 4:*** apply **(Eq 3)** with the calculated $y_1(N)$ value and with $y_1(N-1)$ produced by the final direct calculation of the IIR filter I order to get $y_2(N)$

  - ***step 5:*** compute the signal power with **(Eq 4)** and make frequency decision compared to a threshold

  - for the next burst of N points, set again the same initial conditions and apply all 5 steps above

  - … repeat until the end in order to decode the whole bitstream

- Here is the receiver chain:



Implementation

- We can optimize the current algorithm by merging steps 3 and 4 in only one computing as explained below:

Starting from (Eq 3):

$$
\begin{aligned}
y_2(N) &= x_2(N) - e^{-j2\pi\frac{k}{N}} \bullet x_2(N-1) \\
&= y_1(N) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1) \\
&= 2\cos\left(2\pi\frac{k}{N}\right) \bullet y_1(N-1) - y_1(N-2) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1) \\
&= \left(e^{+j2\pi\frac{k}{N}} + e^{-j2\pi\frac{k}{N}}\right) \bullet y_1(N-1) - y_1(N-2) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1)
\end{aligned}
$$

So: $y_2(N) = e^{j2\pi\frac{k}{N}} \bullet y_1(N-1) - y_1(N-2)$ **: Eq 5**

- Finally, there is no need to evaluate $y_1(N)$ so the FIR is performed only on the 2 last bins of the IIR output i.e. for $y_1(N-1)$ and $y_1(N-2)$

Implementation

Let's rewrite the power spectral density:

$$P = y_2(N) \bullet \overset{*}{y}_2(N)$$

$$= (e^{j2\pi\frac{k}{N}} \bullet y_1(N-1) - y_1(N-2)) \bullet (e^{-j2\pi\frac{k}{N}} \bullet \overset{*}{y}_1(N-1) - \overset{*}{y}_1(N-2))$$

$$= y_1(N-1) \bullet \overset{*}{y}_1(N-1) + y_1(N-2) \bullet \overset{*}{y}_1(N-2) - e^{+j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y}_1(N-2) - e^{-j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y}_1(N-2)$$

$$= \left|y_1(N-1)\right|^2 + \left|y_1(N-2)\right|^2 - e^{+j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y}_1(N-2) - e^{-j2\pi\frac{k}{N}} \bullet \overset{*}{y}_1(N-1) \bullet y_1(N-2)$$

Case 1: $x \in \mathbb{R} \Rightarrow y_1 \in \mathbb{R}$

So: $P = y_1^2(N-1) + y_1^2(N-2) - 2\cos\left(2\pi\frac{k}{N}\right) \bullet y_1(N-1) \bullet y_1(N-2)$ **: Eq 6**

Case 2: $x \in \mathbb{C} \Rightarrow y_1 \in \mathbb{C}$

Let's write: $z = e^{+j2\pi\frac{k}{N}} \bullet y_1(N-1) \bullet \overset{*}{y}_1(N-2)$

So: $\overset{*}{z} = e^{-j2\pi\frac{k}{N}} \bullet \overset{*}{y}_1(N-1) \bullet y_1(N-2)$

Finally: $P = \left|y_1(N-1)\right|^2 + \left|y_1(N-2)\right|^2 - \left(z + \overset{*}{z}\right)$

$= \left|y_1(N-1)\right|^2 + \left|y_1(N-2)\right|^2 - 2 \bullet \Re(z)$ **: Eq 7**

Implementation

- Proposal:

  - after preamble synchronization, acquire the 1^{st} burst composed by N points time domain of I/Q samples

  - ***step 1***: set initial conditions: $y_1(-1) = y_1(-2) = 0$

  - ***step 2:*** terminate **(Eq 1)** IIR filter after N iterations from 0 to N-1

  - ***step 3:*** compute the signal power with **(Eq 6)** or **(Eq 7)** depending if input signal is real or complex, then make frequency decision

  - for the next burst of N points, set again the same initial conditions and apply all 3 steps above

  - … repeat until the end in order to decode the whole bitstream

Example

- Here is a simple example with N = 6 points and x a real signal:

$$
\begin{aligned}
y_p(0) &= x(0) \\
y_p(1) &= c_{re\_p} \bullet y(0) + x(1) \\
y_p(2) &= c_{re\_p} \bullet y(1) - y_p(0) + x(2) \\
y_p(3) &= c_{re\_p} \bullet y(2) - y_p(1) + x(3) \\
y_p(4) &= c_{re\_p} \bullet y(3) - y_p(2) + x(4) \\
y_p(5) &= c_{re\_p} \bullet y(4) - y_p(3) + x(5) \\
P_p &= y_p{}^2(5) + y_p{}^2(4) - c_{re\_p} \bullet y(4)
\end{aligned}
$$

$$
\begin{aligned}
y_m(0) &= x(0) \\
y_m(1) &= c_{re\_m} \bullet y(0) + x(1) \\
y_m(2) &= c_{re\_m} \bullet y(1) - y_m(0) + x(2) \\
y_m(3) &= c_{re\_m} \bullet y(2) - y_m(1) + x(3) \\
y_m(4) &= c_{re\_m} \bullet y(3) - y_m(2) + x(4) \\
y_m(5) &= c_{re\_m} \bullet y(4) - y_m(3) + x(5) \\
P_m &= y_m{}^2(5) + y_m{}^2(4) - c_{re\_m} \bullet y(4)
\end{aligned}
$$

Where: $c_{re\_p} = 2\cos\left(2\pi \frac{k}{6}\right) = 2\cos\left(\frac{k\pi}{3}\right)$ for $+\Delta f$

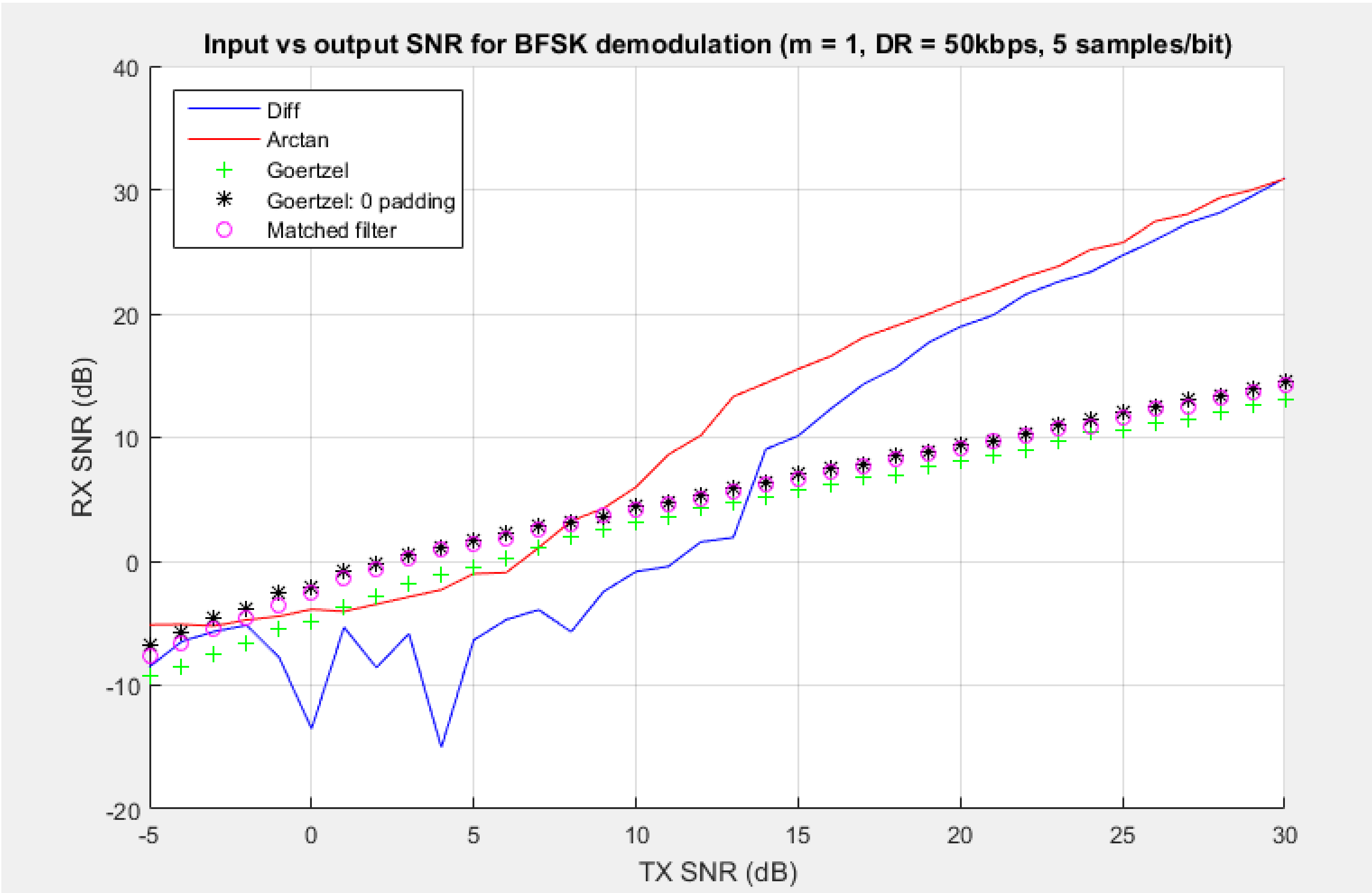
And: $c_{re\_m} = 2\cos\left(2\pi \frac{(6-k)}{6}\right) = 2\cos\left(\frac{(6-k)\pi}{3}\right)$ for $-\Delta f$

So: $P = \max(P_p, P_m)$

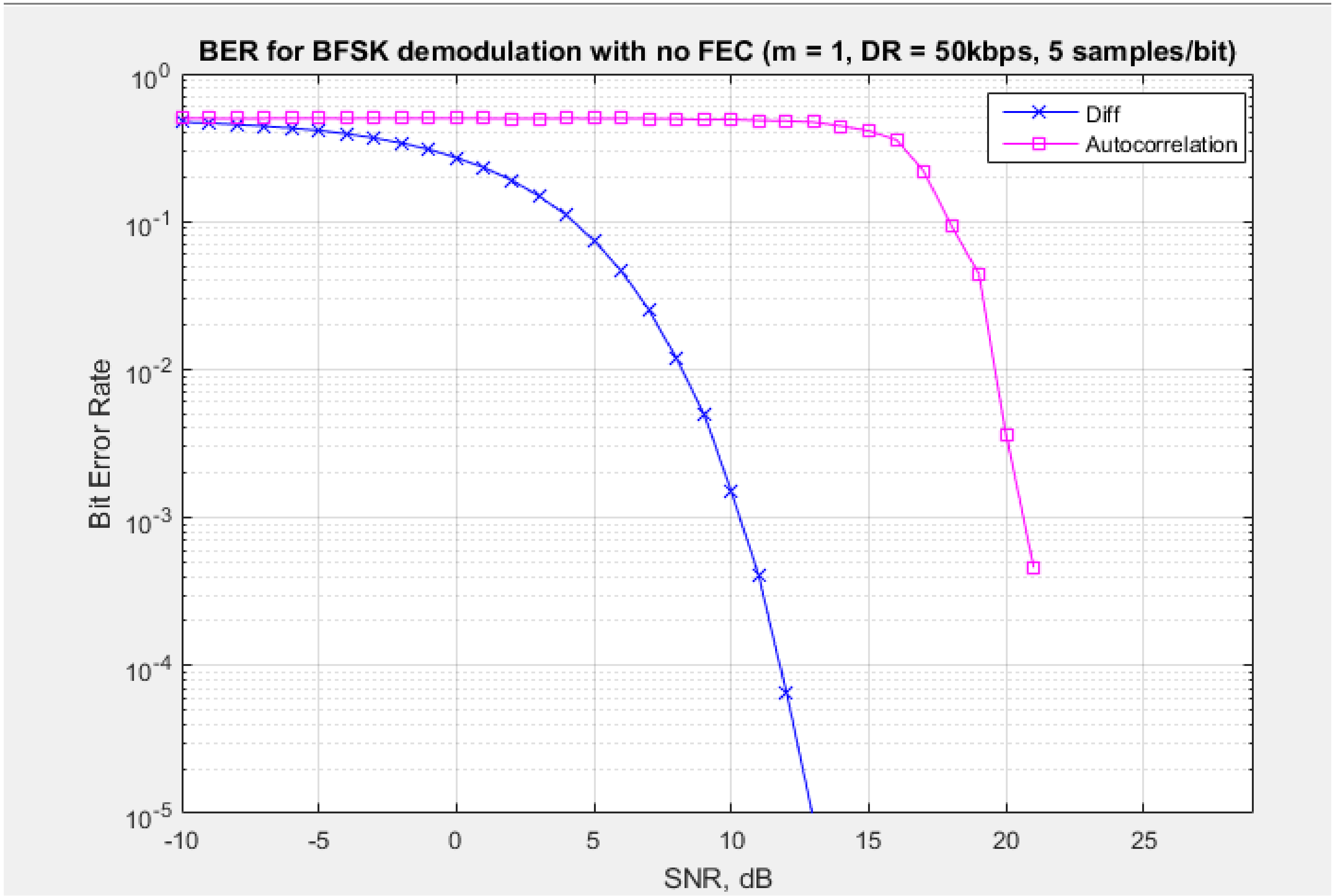
$if\ \ P = P_p \Rightarrow +\Delta f$

$if\ \ P = P_m \Rightarrow -\Delta f$

Simulations



Simulations



Simulations