Comparison of different frequency estimation algorithms*

J.A.

E-mail: contact AT tsdconseil.fr

June 5, 2015

Foreword

This small draft aims at presenting and comparing different frequency estimation techniques. More formally, given N successive observations of a pure sinusoidal or exponential tone, sampled at a fixed frequency $f_s = 1/T_s$, and contaminated by AWGN noise:

$$x_n = A\cos(2\pi f n T_s + \phi) + w_n \quad \text{or} \quad x_n = Ae^{2\pi \mathbf{i} f n T_s + \phi} + w_n \tag{1}$$

with unknown amplitude A, frequency f, and phase ϕ, we seek to estimate the value of f as accurately as possible. We will see that the two problems (real and complex cases) are not equivalent.

Notes: I made this short, informal and incomplete evaluation following an interesting debate on the dsp related forum concerning the efficiency of a new frequency estimator. I want to thank here Eric Jacobsen for his advice, reading tips, and very helpful review of this note, the anonymous forumer Cedron for raising this debate which raised a lot of interesting questions for me, and Rick Lyons for his reading tips.

*Most recent version of this document is available at the following address: http://www.tsdconseil.fr/log/scriptscilab/festim/index-en.html

Rev.	**Date**	**Description**
0.5	06/05/2015	Added Cedron's forumula for complex tone detection
0.4	06/03/2015	Added new Candan (2013 version) estimator to the testbench, changed N from 512 to 128 to better show the difference between Jacobsen, Candan, and new Candan methods, added an explanation from E. Jacobsen for the gaussian 3 points VS 2 points interpretation.
0.3	06/02/2015	• Corrections as advised by Randy Yates: using the conventional terms for normalized frequency and normalized pulsation. • Corrections as advised by Cedron: Cedron's formula index correction, corrected a minus sign in the DFT definition • Added McEachern method in the testbench • Added a comparison of behavior of the gaussian method, with different values for the window width (σ).
0.2	06/01/2015	Corrections as advised by Eric Jacobsen: errors in the graphs labels, definition of the Fourier transform and the first common steps of interpolation algorithms, precisions on the test conditions, bibliography update, terms coherency, english corrections.
0.1	05/31/2015	Initial version

Table 1: *Revision history*

Contents

1 Terms and abbreviations

Normalized frequency ν To simplify the notations, we will in this document always refer to frequencies relative to the sampling frequency f_s: $\nu = f/f_s$. For example, Nyquist frequency ($f_s/2$) will be refered as $\nu = 0.5$. Some refered articles express their estimators as a normalized pulsation Ω, expressed in radians; the relation is straigthforward: $\Omega = 2\pi\nu$.

DFT Discrete Fourier Transform

FFT Fast Fourier Transform

Cramer-Rao Lower Bound (CRLB) This theoretical bound gives the minimum variance of any unbiased estimator. See reference [1].

SNR efficiency An estimator $\hat{\nu}$ is said to be energy efficient iff, when SNR $\to \infty$, $\mathbf{var}(\hat{\nu}) - \text{CRLB}(\text{SNR}) \to 0$. In plain english, it means that when the SNR is high, this estimator has the lowest possible variance (and thus it is efficient at high SNR).

2 Cramer-Rao bounds

These bounds tell the minimum variance of any unbiased estimator ; In other words, they show the best that we could possibly do for our frequency detection problem.

2.1 CRLB for real tones detection

The author did not find a closed form expression or approximation of the CRLB valid for the whole frequency range $[0, 0.5[$. However, in [1], 2 useful approximations are given:

- One approximation of the frequency CRLB, when the phase and amplitude of the signal are supposed to be known :

$$\text{CRLB}(\nu) = \frac{\sigma^2}{A^2 \cdot \sum_{n=0}^{N-1} \left(2\pi n \sin\left(2\pi\nu n + \phi\right)\right)^2} \tag{2}$$

- Another simpler approximation, when the phase and amplitude are unknown, but not valid near DC ($\nu = 0$) and Nyquist ($\nu = 0.5$) frequencies:

$$\text{CRLB}(\nu) = \frac{12\sigma^2}{4\pi^2 A^2 N(N^2 - 1)} \tag{3}$$

  Note that this approximation is independant of the frequency.

On the figure 1 page 5, we have tried to plot these bounds for different parameters: sample size (from 8 to 2048 samples), with known phase (0 or $\pi/2$) or not, and different frequencies from 0 (DC) to 0.5 (Nyquist). Note that the CRLB for unknown phase are always higher than for known phase, as can be expected since there is less information.

The (intuitive) conclusions that can be drawn from these plots are:

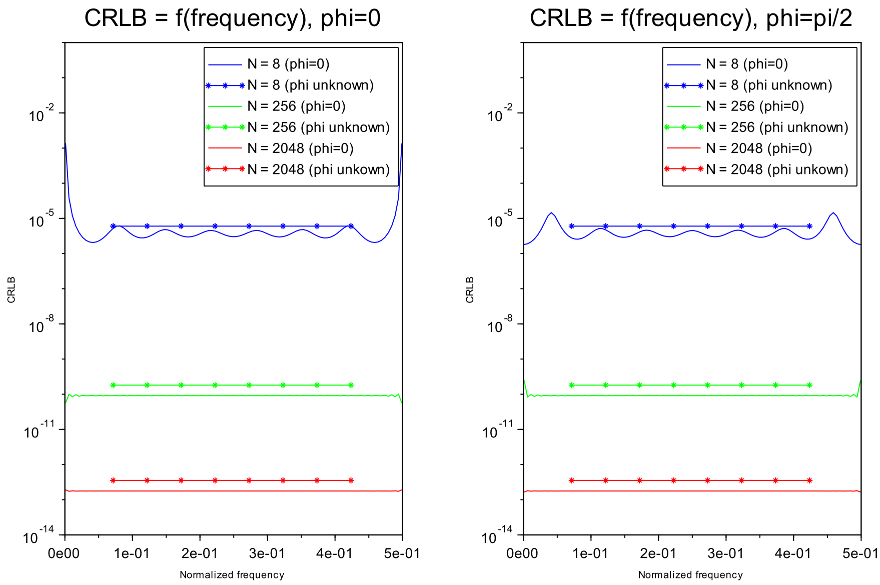

Figure 1: CRLB approximations for real tones, with various window length ($N = 8, 256, 2048$), at fixed SNR (20 dB)

1. We note that knowing the phase of the signal therorically gives additionnal accuracy on the frequency localisation, depending on the frequency (oscillations in the CRLB, especially for small N). We will not use this fact in this evaluation (we will always suppose the phase to be unknown).

2. **If the input is a general (phase unknown or zero) pure sinusoidal tone, it not possible to identify a near-DC or near-Nyquist frequency accurately and repetitively with an unbiased estimator.** As an exception to this rule, if the phase of the input tone is not zero (for a cosinus), then the frequency can be estimated accurately (depending on the input SNR) even near the DC and Nyquist points.

3. The more the number samples available (N) is large, the more the flat region of the CRLB is wide, and so we can expect to be able to identify frequency closer and closer to DC (or Nyquist).

2.2 CRLB for complex tones detection

In this case, due to the complementarity of the I and Q part of the signal ($\pi/2$ phase offset), there is no dependance of the CRLB on the frequency or on the phase:

$$\mathrm{CRLB(f)} = \frac{6\sigma^2}{4\pi^2 A^2 N(N^2-1)} \tag{4}$$

So, we don't expect to have problems for near-DC or near-Nyquist frequencies.

3 Frequency estimation algorithms

3.1 Methods based on 3 DFT bins

3.1.1 Common preprocessing

These methods are based on the same first following steps:

1. Computation of the (unwindowed) DFT of the input signal:

$$X_k = \sum_{j=0}^{N-1} x_j e^{-\frac{2\pi \mathbf{i} jk}{N}} \tag{5}$$

2. Selection of the highest magnitude bin:

$$k^\star = \underset{k=0\dots N-1}{\arg\max} |X_k| \tag{6}$$

   Note that for real tone detection, we can restrict ourself to the positive frequency bins $(0\dots N/2-1)$, since the negative frequency bins have the same magnitude.

3. Computation of the approximate frequency $\hat{\nu}_0 = \frac{k^\star}{N}$ if $k^\star < N/2$, and $\hat{\nu}_0 = -\frac{N-k^\star}{N}$ if $k^\star \geq N/2$ and for complex tones only. We can also express this estimate in radians: $\hat{\Omega}_0 = 2\pi\hat{\nu}_0$. If we stop the processing here (without any interpolation), the resolution is limited $\frac{2\pi}{N}$ radians. This is what we will refer as the **"basic method"** in the following SCILAB simulation.

We can observe that, for a pure tone, most of the signal energy is, in the Fourier domain, focused around the $k^\star$ bin. Although, due to the fact that the real frequency is not necessarily exactly centered at one bin (except if it is a multiple of $1/N$), the signal energy is spread around the highest bin ("spectral leakage", see figure 2 page 7).

This is why we can expect to obtain a more accurate frequency estimation by using the information contained not only in the highest DFT bin $X_{k^\star}$, but also in his two neighbor $X_{k^\star-1}$ and $X_{k^\star+1}$.

3.1.2 Interpolation technics

Quadratic: Quadratic interpolation of the 3 DFT bins magnitudes, as described in [6].

$$\hat{\Omega} = \hat{\Omega}_0 + \hat{\delta}_Q, \quad \delta_Q = \frac{2\pi}{2} \cdot \frac{|X_{k^\star+1}| - |X_{k^\star-1}|}{2|X_{k^\star}| - |X_{k^\star+1}| - |X_{k^\star-1}|} \tag{7}$$

As we will see, this estimator performs quite poorly.

Gaussian: This method (see reference [9]) consists in applying a gaussian window the to the signal before the DFT, and then doing a quadratic interpolation of the log-magnitude of the 3 highest bins. Since the DFT of a gaussian is also a gaussian, after the log-magnitude operation, the signal should be quadratic, hence the interpolation should be without bias. However, since it necessary to truncate the ideal gaussian window, there is some bias, very dependent on the number of samples N of the DFT.

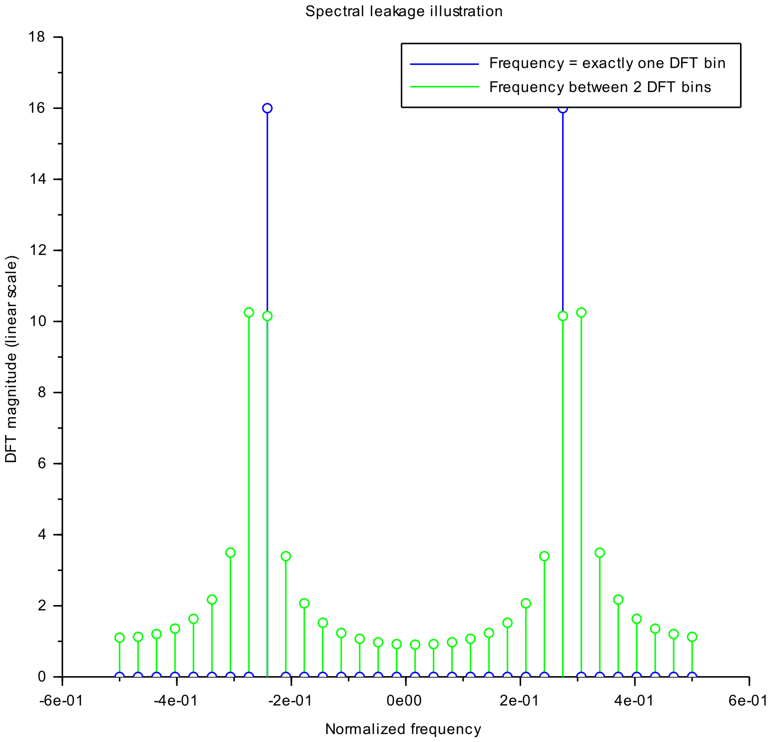

Figure 2: Spectral leakage

McEachern This variant of the Gaussian method is described in refrence [11]. It is very similar to the gaussian method, except that it uses only 2 DFT bins to do the interpolation.

Jacobsen: E. JACOBSEN interpolation formula, as described in [3] and [8]:

$$\hat{\Omega} = \hat{\Omega}_0 + \hat{\delta}_J, \quad \hat{\delta}_J = 2\pi \cdot \mathbf{Re}\left(\frac{X_{k^\star-1} - X_{k^\star+1}}{2X_{k^\star} - X_{k^\star+1} - X_{k^\star-1}}\right) \tag{8}$$

Candan: Bias correction factor upon Jacobsen formula, as described in [4]:

$$\hat{\delta}_C = \hat{\delta}_J \cdot \frac{\tan \pi/N}{\pi/N} \tag{9}$$

Candan2: CANDAN has published a new algorithm in 2013 (see [5]). It is a correction upon the previous estimate:

$$\hat{\delta}'_C = \frac{\tan^{-1}\left(\hat{\delta}_C \cdot \pi/N\right)}{\pi/N} \tag{10}$$

Cedron: Anonymous post by user Cedron on the DSP Related website (see [7]). The formula is based on the direct resolution of the 3 unknowns (frequency, phase and amplitude) / 3 equations problems, using the 3 bins values, and expecting a pure real tone. As such, this is a zero-bias estimator. The formula is a little complicated:

$$\hat{\Omega} = \cos^{-1}\left(\frac{-\cos(2\pi(k^\star-1)/n)X_{k^\star-1} + (1+r)\cos(2\pi k^\star/n)X_{k^\star} - r\cos(2\pi(k^\star+1)/n)X_{k^\star+1}}{-X_{k^\star-1} + (1+r)X_{k^\star} - rX_{k^\star+1}}\right) \tag{11}$$

with $r = e^{-2\pi \mathbf{i}/N}$

Cedron Cplx: Cedron has proposed another 3-bins exact formula for the specific case of pure complex tones. This formula is less complicated than the real tone one:

$$\hat{\Omega} = \hat{\Omega}_0 - \mathbf{i}\log\left(\frac{-rX_{k^\star-1} + (1+r)X_{k^\star} - X_{k^\star+1}}{-X_{k^\star-1} + (1+r)X_{k^\star} - rX_{k^\star+1}}\right) \tag{12}$$

3.2 Direct methods

These estimators are described in reference [2].

Lank-Reed-Pollon (LRP) This is a biased estimator, but it has very little complexity.

$$\hat{\Omega} = \tan^{-1}\left(\mathbb{E}\left[x_n x^\star_{n-1}\right]\right) \tag{13}$$

Note the similarity with a polar discrimator that can be found in an incoherent FSK demodulator, except that here the arctangent operation is done *after taking the average of the signal frequency in the I/Q domain* (better noise immunity than if we averaged the sequence of instantaneous frequencies).

Kay's Circular estimator (KC) Weighted version of LRP to remove the bias. It is unbiased, and SNR efficient (CRLB reached at about 50 dB, for $N = 512$).

$$\hat{\Omega} = \tan^{-1}\left(\mathbb{E}\left[w_n x_n x^{\star}_{n-1}\right]\right) \tag{14}$$

with $w_n = \frac{6n(N-n)}{N(N^2-1)}$

Parabolic Smoothed Central Finite Difference Estimator (PSCFD) It is unbiased, and SNR efficient (CRLB reached at about 20 dB, for $N = 512$).

$$\hat{\Omega} = \tan^{-1}\left(\mathbb{E}\left[w_n \cdot \frac{x_n}{|x_n|} \cdot \frac{x^{\star}_{n-1}}{|x_{n-1}|}\right]\right) \tag{15}$$

with the same weights w_n. From the normalization operation, we can infer this estimator will performs better at high SNR (at low SNR, the normalization can amplify the noise). As will be see later, this will be confirmed by experimentation.

4 SCILAB simulation

The estimators were simulated using SCILAB, an open-source scientific computing software, with similar origins as MATLAB. SCILAB is available for download at the following address: `http://www.scilab.org`

4.1 Tests conditions

Two kinds of test have been done: mean square error (MSE) VS SNR, and MSE VS frequency, in each case both for real tones and complex tones. In all cases, we have fixed **the number of samples** N to 128. Note that this is an arbitrary choice, and that the results are dependant on this value[1].

The **number of trials** for each test condition has been fixed to 100 (an upper value would probably enable more accurate results, but the simulation being done with the interpreted language SCILAB, I had to find a compromise).

The **phase** of the real and complex tones has been fixed to zero, so as to demonstrate the rise of the differente estimators variance when the frequency is near-DC or near-Nyquist (for real tones). Note that this has no impact for complex tones detection. In this last case, the results seem to be identical if a random phase is selected at each trial.

The **frequencies (for MSE VS SNR plots)** were selected randomly (uniform distribution) in the range 0.1 - 0.4 (hence excluding near-DC and near-Nyquist areas). As such, the tested frequencies can fall anywhere between the DFT bins (uniform distribution).

The **frequencies (for MSE VS frequency plots)** were selected uniformly in the range 0.005 - 0.495, so as to demonstrate the variance increase at near-DC and near-Nyquist and for real tones, but not including pure DC and pure Nyquist frequencies, for which the variance of the estimators being infinite, the estimated MSE would not be accurate.

[1] If the reader is interested in reproducing these simulations or changing the paremeters, the simulation scripts described here are available at the following address: `http://www.tsdconseil.fr/log/scriptscilab/festim/index-en.html`

4.2 Adaptation of the direct methods for real tone detection

The detectors based on 3 DFT bins can work indifferently on real or complex tones (because the only difference is the presence or absence of negative frequencies, which do not impact these estimators), so the same algorithms were used in both cases.

On the contrary, the detectors based on direct time series analysis (LRP, KC, PSCFD) were designed for detecting complex tones only. For the detection of real tones, the solution I choose to adapt these algorithms was to build an analytical signal with the Hilbert transform.

Yet this solution is not ideal, since the Hilbert transform is difficult to do exactly (up to my limited knowledge). I experimented first with the FFT method, based on the direct zeroing of the negative frequencies in the frequency space. However this method introduces lot of spurius artifacts due to the circular behavior of the DFT. In a second time, I used a FIR approximation of the Hilbert transform to avoir the circular artefacts, yet, as we will see later, the results are still not optimal. If the reader has any idea or suggestions to improve on this, I would be very grateful!

4.3 Interpretation of the results

4.3.1 Frequency dependance

As confirmed by the experimentations (see figure 5 page 13), the complex tone detection is not impacted by the exact frequency localization. The exception is of course for the basic method (selection of the highest magnitude DFT bins, without inerpolation), which sees its MSE drop when the frequency is exactly a multiple of $1/N$.

On the contrary, the detection of a real tone can be very difficult if (1) the initial phase of the tone is 0, and (2) the exact frequency is near DC or near Nyquist. This is confirmed in figure 6 page 13, where we can see the MSE of the different estimators rising dramatically near DC or Nyquist frequencies.

4.3.2 SNR efficiency

An estimator is SNR efficient if, not only it is unbiased, but also reach the CRLB when SNR is ∞.

The simulations show that (see figure 3 page 12), *for an exponential tone detection*, only the KC and PSCFD estimators are SNR efficient. The Candan 2, Cedron and Cedron cplx estimators have in this case the same performance but are less efficient than the time domain methods (constant gap to CRLB).

For real tone detection, none of the estimators I tested is SNR efficient[2] (see figure 4 page 12). That means that there exists probably at least one other estimator better than the ones tested. If the reader has any suggestion, I would be very interested. Among the tested estimators, the ones that perform best in this case are the Jacobsen, Candan and Cedron estimators, which perform about equaly well up to 70 dB SNR. Above 70 dB, Jacobsen and Candan estimators

[2]Note that the poor performances of the direct method at high SNR for real tones detection is probably due to my inability to extract accurately an analytic signal. As previously said, I would be obliged if the reader has any suggestion about this.

reach their noise floor, and Cedron estimator performs better (logically since it is an exact formula from 3 DFT bins), yet with a constant gap to the CRLB[3].

4.3.3 Behavior at low SNR

At low SNR (< 20 dB), and both in the real and complex cases, the best performing methods are the DFT based: Jacobsen, Candan and Cedron. They perform equally well, so the simplest of the three (Jacobsen) seems to be the most appropriate.

Note that below a SNR of, about, 5 dB for real tones, and -5 dB for complex tones, all the tested methods seem to diverge a lot from the CRLB. This is an indication that there may exist much more accurate technics for operation at low SNR. If the reader has any suggestion about it, I would be very interested.

4.3.4 Gaussian VS McEachern methods

Apparently, the gaussian method (interpolation using 3 DFT points) and the McEachern method (interpolation with only 2 DFT points) behave the same way. This is a little surprising, since we would expect that using another point would bring additionnal information.

This point has been explained by Eric Jacobsen on the dsprelated forum (02-06-2015):

> *"Of the myriad two-term estimators, the gaussian performs well specifically because it eliminates the nulls between sidelobes that are present in most windows. Many of the two-term estimators have large variance when the frequency is near the bin center as one of the terms approaches zero."*

[3] This gap can be interpreted as the consequence of using only 3 DFT bins, hence a part of the signal information is discarded.

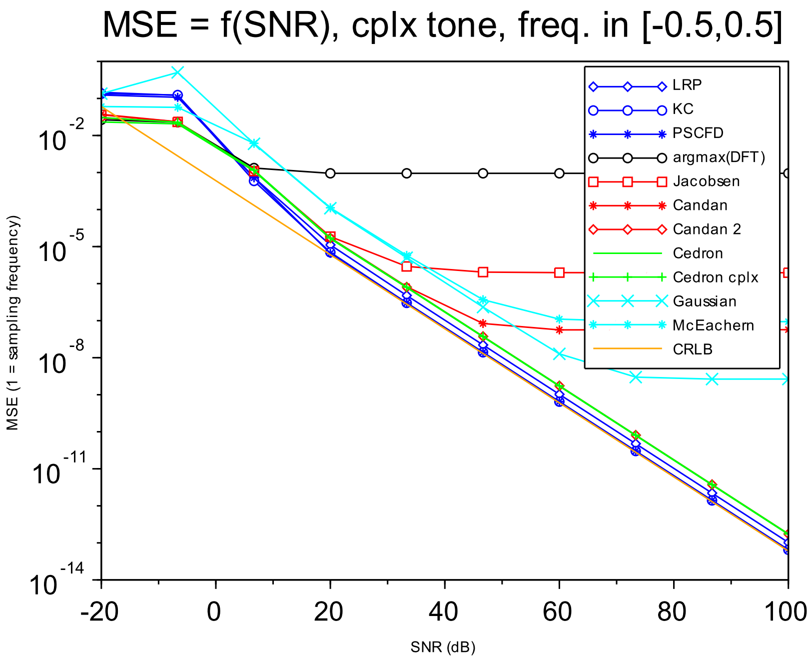

Figure 3: Mean square error simulation, for exponential tone detection

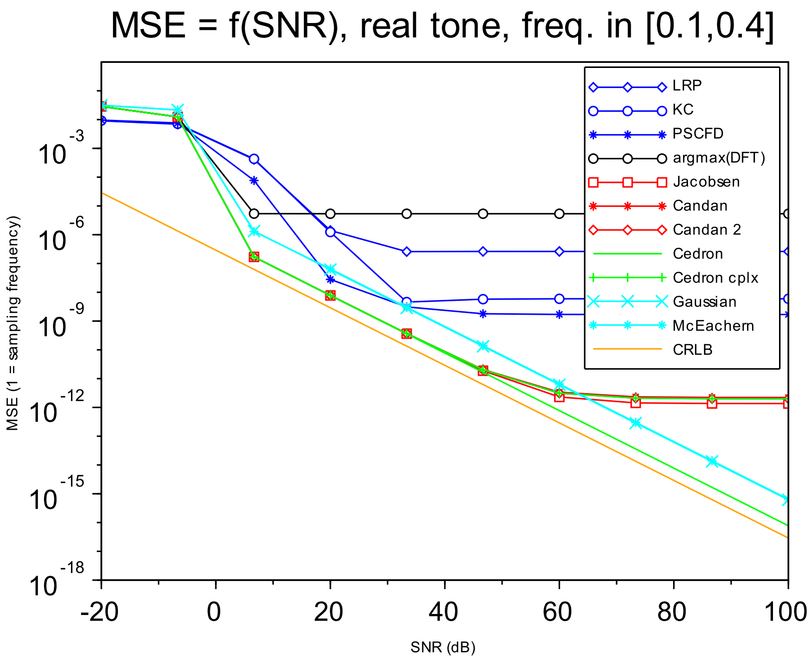

Figure 4: Mean square error simulation, for sinusoidal tone detection

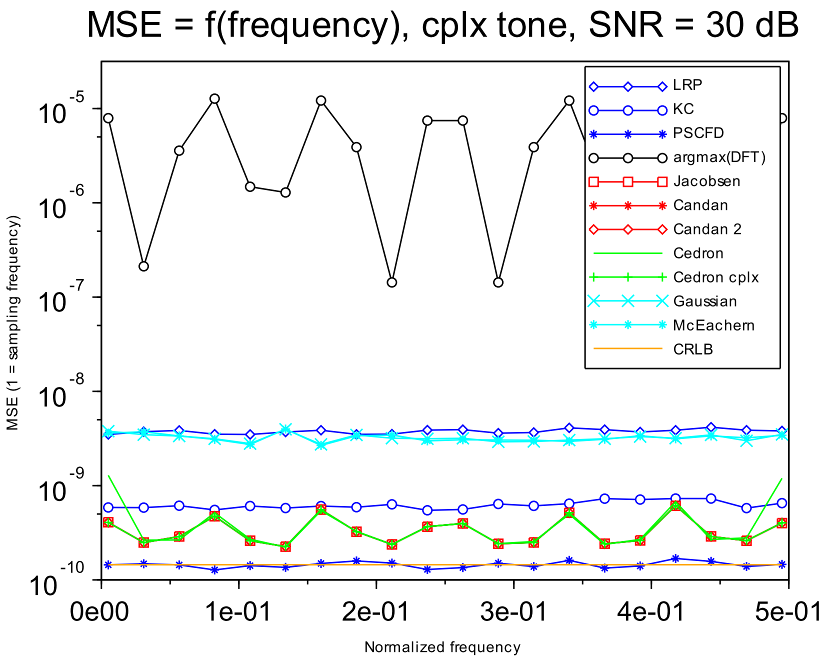

Figure 5: Mean square error simulation, for exponential tone detection

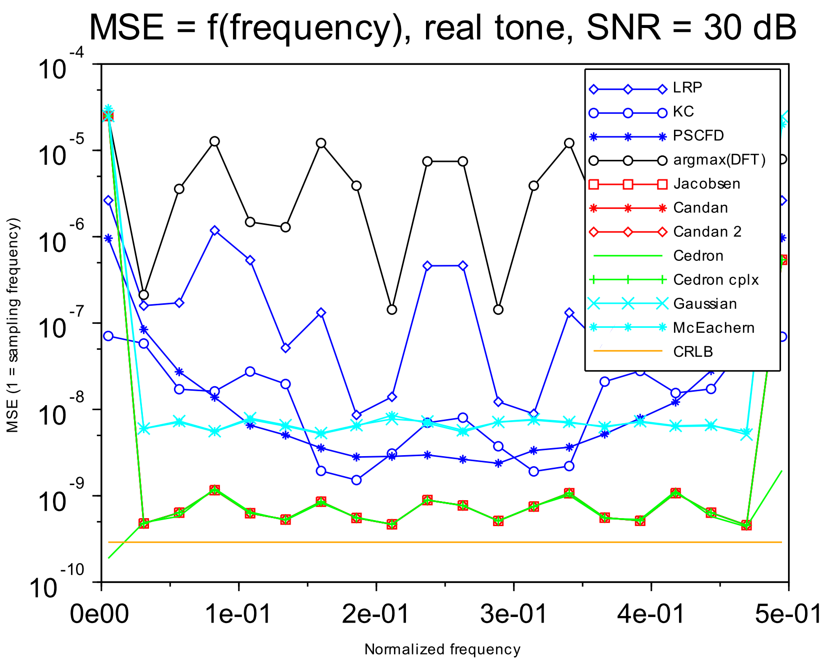

Figure 6: Mean square error simulation, for sinusoidal tone detection

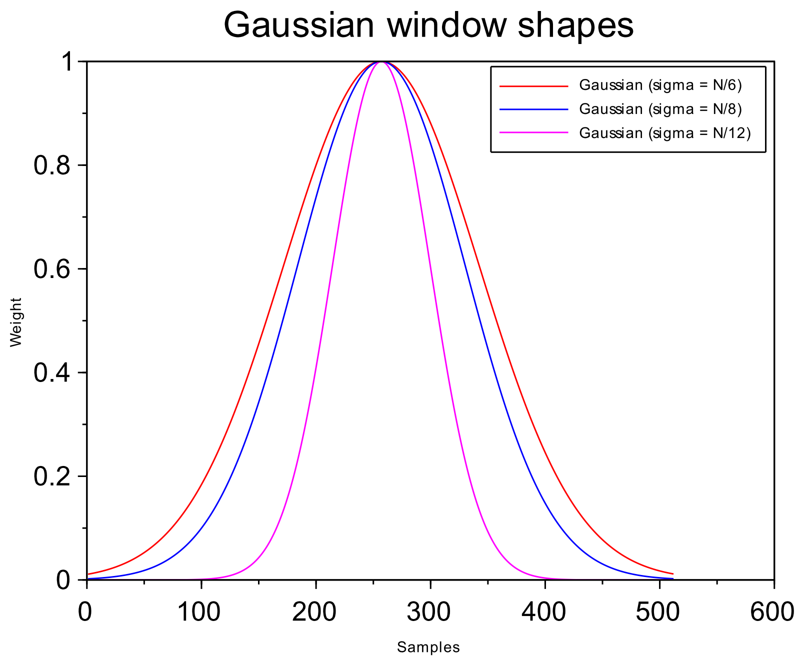

Figure 7: Different gaussian window shapes (N = 512)

4.3.5 Choice of the gaussian window shape factor

For the gaussian method (and likewise for the McEachern method), there is a free paremeter: the standard deviation of the gaussian window applied before the DFT. Figure 7 page 14 show 3 examples of possible windows, with $\sigma = N/6$ (widest window), $N/8$ and $N/12$ (narrowest window).

There is no absolute ideal choice for σ, since if σ is small, then the spectral leaks are well removed (no discontinuity between the begin and the end of the signal due to the circular behavior of the DFT), but most of the signal energy is lost! On the other hand, if σ is too large, too much spectral leaks appear.

This is confirmed by the experimentation (see figure 8 page 15), which shows that for a large window ($\sigma = N/6$), the behavior is good at low SNR, because we do not supress to much the input signal, but at high SNR, the self-noise becomes dominant (noise floor on the red curve).

On the other side, for a narrow window ($\sigma = N/12$), the behavior at high SNR is much better, because much less self-noise (spectral leakages) is injected. On the other hand, at low SNR, the performances are much degraded, due to the fact that we use only a small part of the signal energy.

Note also that, the narrowest is the window, the wider is the gap to the CRLB, which is logical since there is less information after windowing.

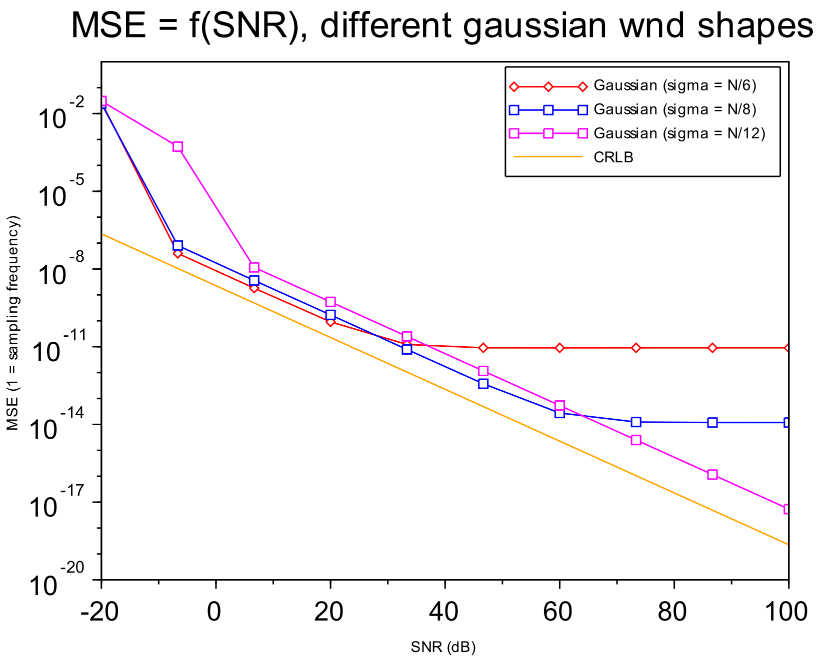

Figure 8: Comparison of different paremeters for the gaussian window (mean square error simulation, for complex tone detection)

5 References

[1] S. KAY, *Fundamental of statistical signal processing: estimation theory*, 1993

[2] V. CLARKSON, *Efficient single frequency estimators*, 2003, link

[3] E. JACOBSEN, P. KOOTSOOKOS, *Fast, Accurate Frequency Estimators*, IEEE Signal Processing Magazine, Vol. 24, Issue 3, pp. 123-125, May, 2007

[4] C. CANDAN, *A method for fine resolution frequency estimation from three DFT samples*, 2011

[5] C. CANDAN, *Analysis and further improvement of fine resolution frequency estimation method from three DFT samples*, 2013, link

[6] M. DONADIO, *How to interpolate the peak location of a DFT or FFT if the frequency of interest is between bins*, link

[7] C. DAWG, *Exact frequency formula for a pure real tone in a DFT*, 2015, link

[8] B. BISCHL, U. LIGGES, C. WEIHS, *Frequency estimation by DFT interpolation: a comparison of methods*, 2009, link

[9] J.O. SMITH III, *Quadratic interpolation of spectral peaks* (chapter of spectral audio signal processing ebook), 2010, link

[10] S. CETINKAYA, *DFT-Based High Resolution Frequency Estimation Using Three Samples*, 2013

[11] K. MCGEE, The Gaussian Ratio Technique for Frequency Estimation, An introduction to signal processing and FFT, link