ANALOG DEVICES

TNJ-081

アナログ電子回路技術ノート

窓関数という傍流の裏方こそ深い趣きがある（前編）

窓関数を用いた FFT 結果の「窓関数補正係数」はどう求めるか

著者: 石井 聡

はじめに

前々回の TNJ-079 と前回の TNJ-080 で「窓関数」というものをお話ししました。窓関数を用いる必要性は[1]でも、また以降でも示すとおり、FFT におけるスペクトル・リークを低減させるためです。FFT では窓関数は必須の裏方といえるものです。

今回と次回は、窓関数を用いた FFT におけるふたつの補正係数の算出方法について検討します。前回（TNJ-080）では Hann 窓でのふたつの補正係数を 0.5, 1.5 として、決め打ちで説明しましたが、今回と次回はこの大きさを実際に求めてみます。

これらの補正係数はネットを探しても、数値自体や、補正係数算出手順の詳しい説明をあまりみることはありません。しかしこのような傍流のネタこそ、それこそ窓関数という裏方のトピックこそ、『趣き』があるもんだなぁと深く感心するものでありました（FFT の考え方の理解を見直す機会にもなりました）。

窓関数や古典文学の『趣き』を知る価値

高校のころは古典文学が苦手だったが

高校生のころは成績をどん底に落としてまでも、アマチュア無線にのめりこんだ私。

今でもよく覚えています。1 年生のころ古文の授業の最後に定期テスト採点結果が配布されました。「はい、今回の最高は□□点、最低は◎◎点でした。では今日はおしまい！」と先生は言うものの、私の点数はそれより低いのです。教室から去る先生に、（今思えば、脅迫直大バカですが）「先生、私は◎◎点より低かったんですが」。閻魔帳（えんまちょう）と当時呼ばれていた成績記入帳面を開いて確認した先生。

「ああ、君が最低だ」。閻魔帳がぱたりと閉じる音を聞きながら、「聞いたオレが馬鹿だった」と思いました。なお 2 年生の途中から「これではいけない」と思ってがんばりはしましたが…。

当時の私には、苦手な古典文学は試験対策のツールでしかありませんでした。しかし年齢を重ねるにしたがい、その『趣き』にふと気がつくようになったのでした。

枕草子『春はあけぼの』を改めて想うと、その『趣き』に気がつく

『春はあけぼの。やうやう白くなりゆく山際、すこし明かりて、紫だちたる雲の、細くたなびきたる。』

梅の蕾が開くころ、ひなびた温泉旅館に家族で泊まった。

未だうすら寒い晩頃に到着した。部屋食でリクエストしてあった家族との宴は、なかなか出来ない団欒で日ごろの補填をしつつ、美味しい鍋と、美味しい日本酒をいただいた。

翌朝は　慣れない枕のためか、朝早くに目がさめた。家族はまだすやすやと寝ている。静かな、穏やかな時間だ。

図 1. 春はあけぼの。やうやう白くなりゆく山際、すこし明かりて、紫だちたる雲の、細くたなびきたる

まだうす暗い窓の外に目をやる。まだ新芽色にもなっていない山の稜線のうえの空は、まだ日の出前だ。しかし凛とした気配のまま、少しずつ、少しずつ明るくなってくる。

そして朝焼けで紫がかっていた細い雲は、色合いの変化が目にみえて速くなる。「朝焼けは紫色に見えるんだ」とその『趣き』に、家族の寝息しか聞こえない旅館の和室で、その美しさにひとり、息をのむ。

生きた、本質的な古文の世界は、現代風にいえば、こんな「趣きを嗜む」世界なのかもしれない（妄想ストーリーでした^o^）。こんな『趣き』に高校生のころ気がつけば、古典文学も苦手ではなかっただろう（で、今頃は銀行員？？^o^）。

FFT で生じるスペクトル・リークと窓関数

スペクトル・リークの生じない条件（窓関数はなし）

図 2 はピーク電圧 2V の 2 周期ぶんの波形を 16 サンプル AD 変換したものです（前回の TNJ-080 の図 4 を再掲）。サンプリング・レート 1ksps（sample per sec）、サンプル全長時間は（1 サンプル 1ms の中央でサンプリング動作が行われているとして）16msec、波形自体の周期は 8msec になり、信号周波数は 125Hz です。

これを FFT したスペクトルが図 3（前回の TNJ-080 の図 5 を再掲）です。時間波形 16 サンプル（16 ポイント）を FFT すると、同じ数、16 の周波数ポイント（これを bin といいます）が得られます。スペクトルが 125Hz と、折り返しの 875Hz の 2 つの bin のみに現れていることが分かります。

これは図 2 において、サンプル番号 15 の次（図には表記されていないサンプル番号 16。17 番目）でゼロに戻り、次の 1 周期のスタート・ポイントになっているからです。

Rev. 0

アナログ・デバイセズ株式会社

本　　社／〒105-6891　東京都港区海岸 1-16-1　ニューピア竹芝サウスタワービル
電話 03（5402）8200
大阪営業所／〒532-0003　大阪府大阪市淀川区宮原 3-5-36　新大阪トラストタワー
電話 06（6350）6868

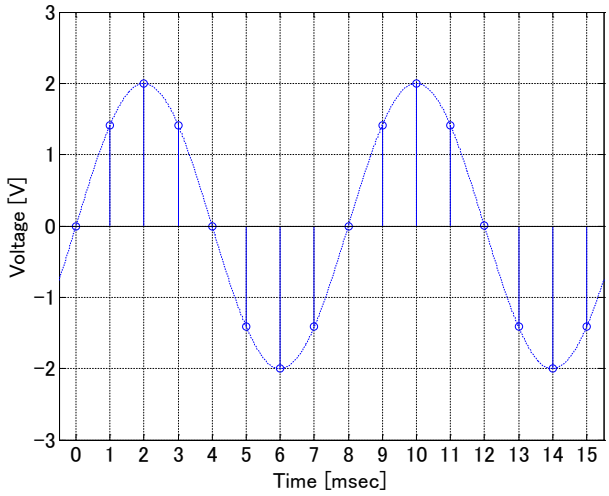


図 2. 2 周期の波形を 16 サンプル AD 変換した時間軸サンプル
（前回の TNJ-080 の図 4 を再掲）

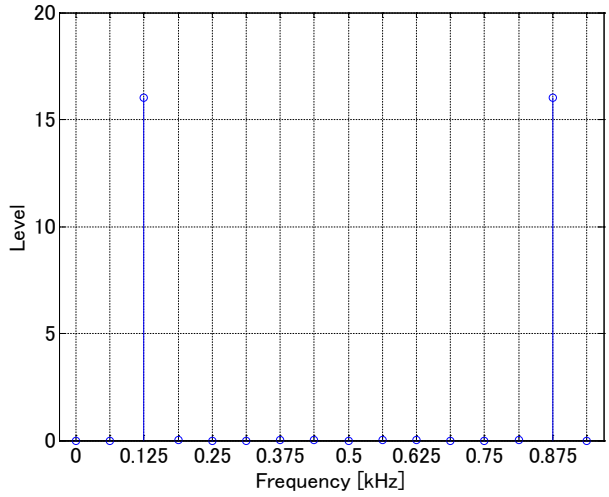


図 3. 図 2 を FFT した周波数スペクトル
（前回の TNJ-080 の図 5 を再掲。最大周波数は 937.5Hz、
bin の周波数ステップは 62.5Hz）

そういう意味ではこの図 2 を複数横につなげていけば、全体の波形形状が連続したかたちで表記できることになります。

実はこれが図 3 において 2 つの bin のみにスペクトルが現れる理由です。「完全な周期性」と呼ぶようです。サンプリング動作という点では、この条件を「コヒーレント・サンプリング」とも呼びます。

また、ピーク 2V の正弦波を 16 ポイントで FFT していますが、2 本のスペクトルが 16 で得られています。16/16 ポイント = 1V で本来の半分の振幅になっていると計算することができます。

スペクトル・リークの生じる条件（窓関数はなし）

つづいて信号の周期を 8msec から 7msec に変更して（周波数 142.86Hz）、サンプリング・レートはさきと同様の 1ksps、サンプル全長時間も 16msec として時間サンプリングとしたものを図 4 に示します。ピーク電圧は 2V で図 2 と同じです。

これを FFT したものを図 5 に示しますが、スペクトルが 2 本だけでなく、その左右にも広がっています。これが「スペクトル・リーク」です。

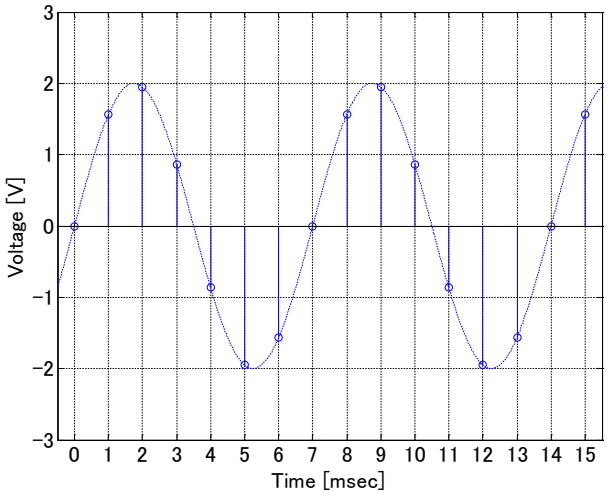


図 4. 図 2 の信号の 1 周期を 7msec に変更した時間軸サンプル

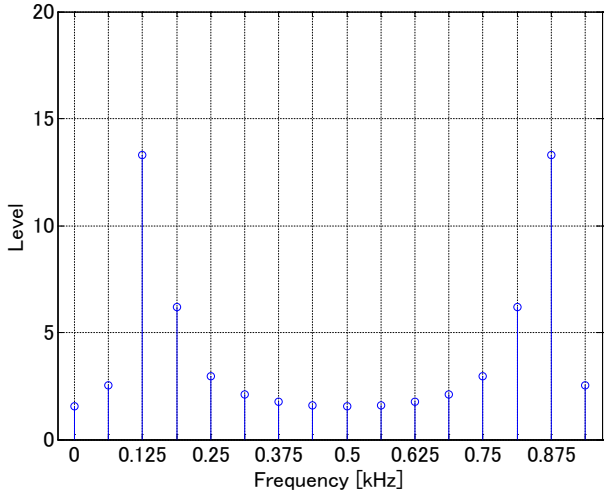


図 5. 図 4 を FFT した周波数スペクトル。大きな
スペクトル・リークが見える

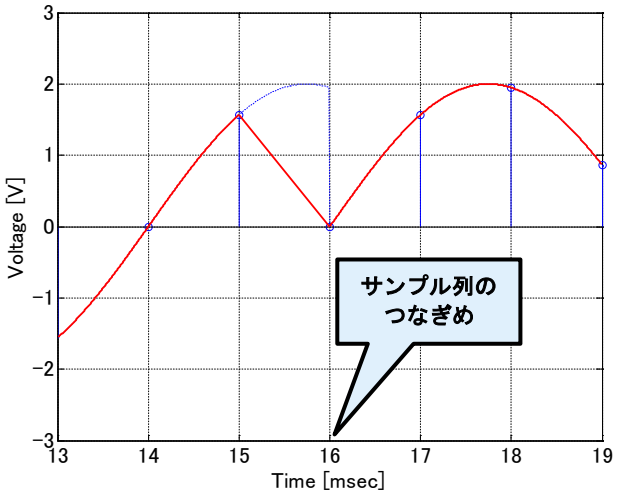


図 6. スペクトル・リークの生じる理由はサンプル最後と次の回
の一連のサンプル列の最初との間が不連続になるから

スペクトル・リークの生じる理由は、図 6 のようにサンプル番号 15 と 16（次の回の一連のサンプル列の最初）の間が不連続だからです。点線はサンプル番号 0 から 15 の波形を外挿したもの、赤の実線はサンプル番号 16 と直結したものです。本来であればサンプル番号 15 と 16 の間は連続であるべきですが、波形位置がずれていることで、「無理やりその間を（数字的に）繋げてしまえば、想定外のスペクトルが出るのでは？」と直観的にも思うのではないでしょうか。

スペクトル・リークの生じる条件に窓関数を適用してみる（以降につながるお話し）

つづいて図 4 の信号に窓関数 Hann 窓をかけます。図 7 の上側の波形が図 4 の信号と Hann 窓、下側の波形が Hann 窓をかけた信号波形です。なおこの Hann 窓の設定は次の TNJ-082 で説明するように、窓関数自体も「完全な周期性」を持たせたかたちにしてあります。この話題は次回で詳しく説明しますので、どうぞお楽しみにお待ちください。

さて、これを FFT したものを図 8 に示します。さきほどは多数のスペクトルがありましたが、2 本のスペクトルのみが大きく現れ、他のスペクトル、「スペクトル・リーク」が低減していることが分かります。

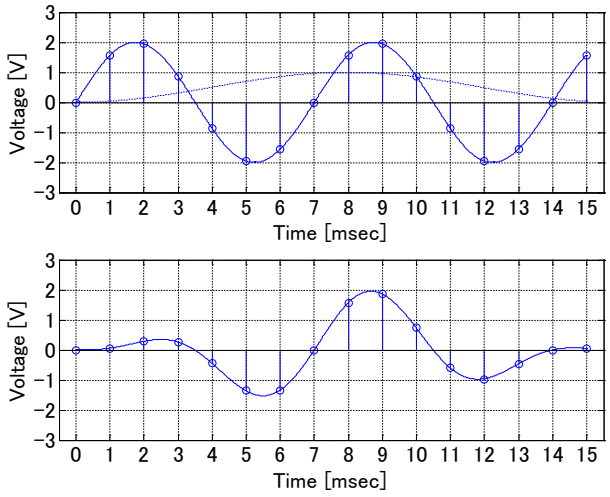


図 7. 図 4 の信号に Hann 窓をかけた時間軸サンプル（上は図 4 の信号と Hann 窓、下は Hann 窓をかけた信号波形）

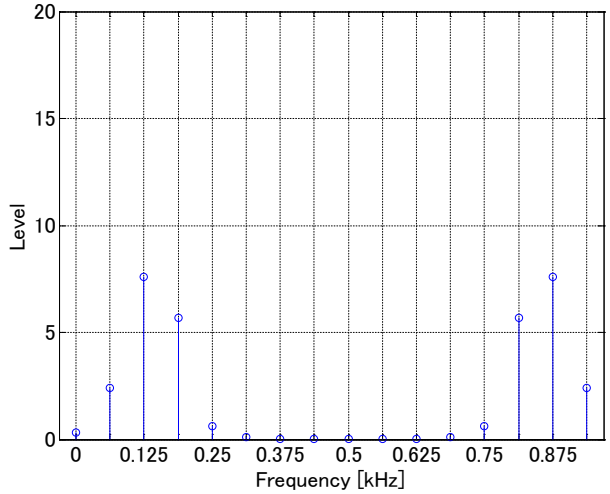


図 8. 図 7 を FFT した周波数スペクトル。スペクトル・リークが低減している。またスペクトル・ピークの大きさも 7.582V になっている

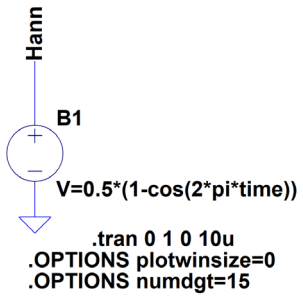


図 9. Hann 窓の減衰補正係数が 0.5 ということを確認してみる LTspice シミュレーション回路

FFT ポイント数を増やせば、窓関数をかけたことにより低減するスペクトル・リークはより急しゅんに減衰することとなり、狭い周波数範囲にスペクトルが集中することになります。

このように窓関数を適用することで、スペクトル・リークを低減できるのです（といっても「完全な周期性」の時間サンプリングに窓関数を適用すると、逆にスペクトル・リークが生じます…。ホント『趣き深い』窓関数です）。

窓関数減衰補正係数のなりたち

窓関数をかけることで、信号の大きさが元々の信号に対して減衰します。窓関数をかけた波形を FFT すれば、減衰した信号の大きさがその結果として出てきます。前々回の TNJ-079 と前回の TNJ-080 で示した Hann 窓だと 0.5 倍になります。

Hann 窓の減衰率「0.5 倍」を検証してみましょう。図 9 のような LTspice モデルで、Hann 窓を連続時間関数

$$w_{Hann}(t) = 0.5[1 - \cos(2\pi t)] \quad (1)$$

ただし

$$0 \le t \le 1$$

として、窓関数全体を LTspice でプロットしてみます（次回に示しますが、範囲、$t \le 1$が『趣き』なのです）。プロットを図 10 に示します。さらにその全体量（面積）を得るために、グラフのラベルを ctrl ＋左クリックして、Average 量を表示してみると、図 11 のように 500mV = 0.5 という答えが得られます。積分時間が 1sec なので、これがそのまま全体量の減衰「率」となり、その逆数が窓関数減衰補正係数になります。

ちなみにこの算出の考え方は、次の TNJ-082 で示す、ノイズ等価帯域幅補正係数を数式的に計算する手順の一部と近い（実際は同じ）ものとなります。

なおこの減衰補正係数は、図 12 に転記した[2]（前回の TNJ-089 の図 7 の再掲）にもいろいろな窓関数減衰補正係数の数値が記載されています。ちなみに図 9 の B1 の関数に

```
V=0.42-0.5*cos(2*pi*time)+0.08*cos(4*pi*time)
```

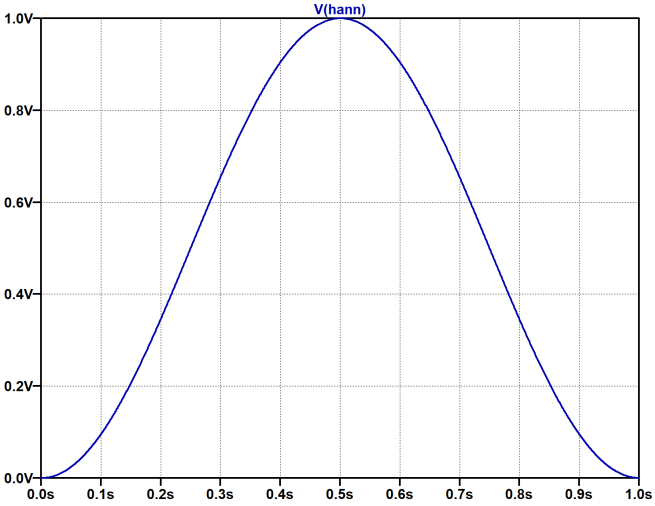
として Blackman 窓 [3]を入れてみると、図 11 に相当する答えは 420mV という値が出て、ちゃんと図 12 と同じになります。

この図 12 の右側は、もうひとつの補正係数、Noise Power Bandwidth です。これは以降で示します。

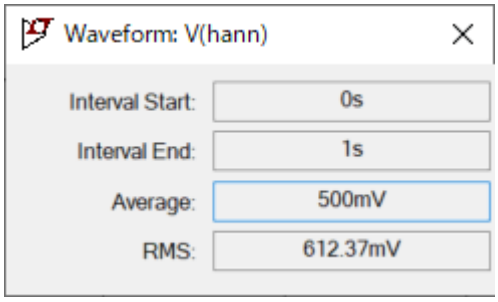
図 4 から図 8 に至るスペクトル・リークの生じる条件に窓関数を適用した例を検証してみると話しは『趣き』深くなりすぎる？！

ここまで Hann 窓の窓関数減衰補正係数 1/0.5 のなりたちについて考えてきました。ここでは図 4 から図 6 に至る、スペクトル・

リークの生じる条件に窓関数を適用した例での補正係数がどうなるかを検証してみましょう。



10. 図 9 のシミュレーション結果



11. 図 10 の結果の全体量を Average として得てみる

Window	Scaling Factor (Coherent Gain)	Noise Power Bandwidth
Uniform (none)	1.00	1.00
Hann	0.50	1.50
Hamming	0.54	1.36
Blackman-Harris	0.42	1.71
Exact Blackman	0.43	1.69
Blackman	0.42	1.73
Flat Top	0.22	3.77

12. いろいろな窓関数の補正係数（[2]の Table 3 より転記）

図 7 の時間軸サンプルに Hann 窓をかけ FFT した図 8 において、信号周波数が 142.86Hz（周期 7msec）、bin の周波数ステップが 62.5Hz ですから、信号周波数はぴったり bin の中心周波数の上には乗っていません。

スペクトル・リークは低減してはいますが、この条件では「窓関数減衰補正係数 = 1/0.5 になるのか」という疑問が生じるかと思います。

実際の数値を得てみましょう。図 5 の窓関数を用いないままで FFT した 125Hz の bin におけるレベルは 13.3152（繰り返しますが実際の信号周波数は 142.86Hz です）、図 8 の Hann 窓をかけた時間軸サンプルを FFT した 125Hz の bin におけるレベルは 7.582 になっています。1 : 0.5 ではありませんね…。同じ条件で信号周波数を 125Hz にしてコヒーレント・サンプリングとした場合は 16.000（窓関数ナシ）、8.000（窓関数あり）で比率は 1 : 0.5（窓関数減衰補正係数 = 1/ 0.5）でしたから、信号周波数が bin の中心からズレていることで、窓関数減衰補正係数の関係から若干低下していることが分かります…。

この辺りの話し（周波数ズレによる振幅レベルの低下）もだいぶ『趣き』があり、そしてだいぶ奥深そうですね！しかし踏み込むと出てこられないような気がしましたので（笑）、これ以上は検討しないことにしておきます…。

ふと疑問に思うこと「源信号の位相が変わっても同じ 2 = 1/0.5 という窓関数減衰補正係数が使えるのか」

執筆していて、さらに、ふと疑問に思いました。「窓関数は中央が大きいな」、「たとえばサンプル時間の全長で 1 周期となるサイン波、コサイン波それぞれに（90° 位相が違うそれぞれに）窓関数をかけた場合に、同じ 2 = 1/0.5 という減衰補正係数が使えるのか」と思ったわけです。

これは結局「オマエはフーリエ変換や FFT の本質を分かっていないな」ということの裏返しなわけですが（汗）。その後の夕刻に 60W の LED 電球を買いにいった帰りに答えに気がつきました（笑）。FFT（実際は離散フーリエ変換）とは

$$\Omega(m) = \sum_{n=0}^{15} x(n)\, w(n) \exp\left(-j\frac{2\pi mn}{16}\right)$$

$$= \sum_{n=0}^{15} x(n)\, w(n) \left[\cos\left(\frac{2\pi mn}{16}\right) - j\sin\left(\frac{2\pi mn}{16}\right)\right] \qquad (2)$$

ここで$x(n)$は時間軸でのサンプル（AD 変換結果）、全長は 16 サンプルとしてあり、$w(n)$は窓関数です。$\exp(jx)$という複素関数でその周波数 bin に関わる信号の全ての位相状態を捉えることができるわけですから、またxによらず$|\exp(jx)| = 1$ですから、結局、信号自体の位相は窓関数処理に影響を与えるものではないことが分かります。$w(n)$による減衰率だけを考えればよいことになります。

具体的にシミュレーションで見てみましょう。図 13 は上記の疑問を確認する LTspice のシミュレーション回路です。窓関数は Hann 窓に設定しています。源信号の位相を.step コマンドで、0, 45, 90, 135, 180, 225, 270, 315° と変えています。

この LTspice のシミュレーションによる計算では、上記の式(2)の離散フーリエ変換ではなく、

$$\Omega(1) = \int_0^1 x(t)w(t)\exp(-j2\pi t)\, dt$$

$$= \int_0^1 x(t)w(t)[\cos(2\pi t) - j\sin(2\pi t)]\, dt \qquad (3)$$

という式（観測する bin の周波数$f = 1$ Hzとしています）で 1sec 連続積分し、フーリエ級数の係数を得ているものです。式(2)のステップを微細にしていけば、フーリエ級数の係数に相当するものが得られるという考え方にたっています。

シミュレーションした結果を図 14 に示します。とはいえこれを見ただけでは、何なのかよく分かりません。そこで LTspice にある.MEASURE ディレクティブ（指示）、そのうち平均値を計算する AVG（平均値）パラメータを使って、積分に相当する大きさを得てみたいと思います。「平均値」というのは全体の大きさをその長さ（ここでは 1）で割るわけですから、積分動作をしていることと等価なわけです。実際の記述は.

```
.MEAS TRAN Freal AVG V(FOUR_R_OUT)
.MEAS TRAN Fimag AVG V(FOUR_I_OUT)
```

とします。ここで Freal は cos 部（実数部）を積分したもの、また Fimag は sin 部（虚数部）を積分したもので、それをさらに手計算で、

```
SQRT(Freal^2 + Fimag^2)
```

とすることで FFT 結果の、周波数$f = 1$ Hzの bin における「大きさ（絶対値）」を得ることができます。

実際に計算してみると、位相を 0, 45, 90, 135, 180, 225, 270, 315 と変えても、どこもこの計算の答えは 0.25 となり（Hann 窓をかけない場合の答えは 0.5 になります）、「源信号の位相が変わっても、同じ 2 = 1/0.5 という Hann 窓関数減衰補正係数が使える」ということが分かりました。

窓関数のノイズ等価帯域幅補正係数のなりたちを連続信号で考える

つづいて窓関数のノイズ等価帯域幅補正係数γ（本技術ノートでは記号γを用います）のなりたちを考えます。

ノイズ等価帯域幅補正係数γの計算の考え方は、図 15 のように、周波数によらずレベルが一定な（一様に分布する）ホワイト・ノイズが入力信号となり、

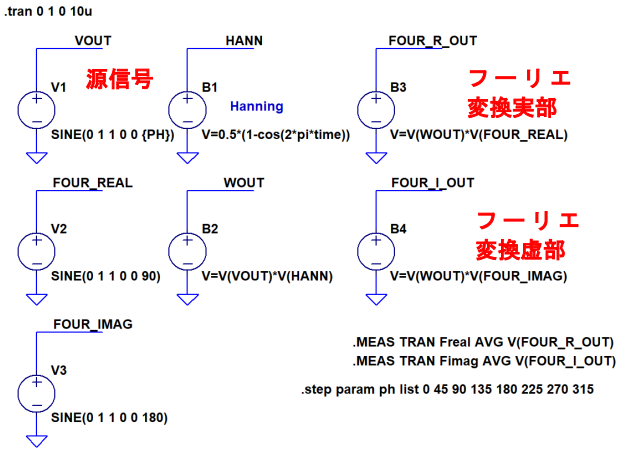


図 13. 源信号の位相が変わっても、同じ減衰補正係数が使えるかを確認する LTspice シミュレーション回路（左上が源信号、右の 2 つがフーリエ変換の実部、虚部の計算）

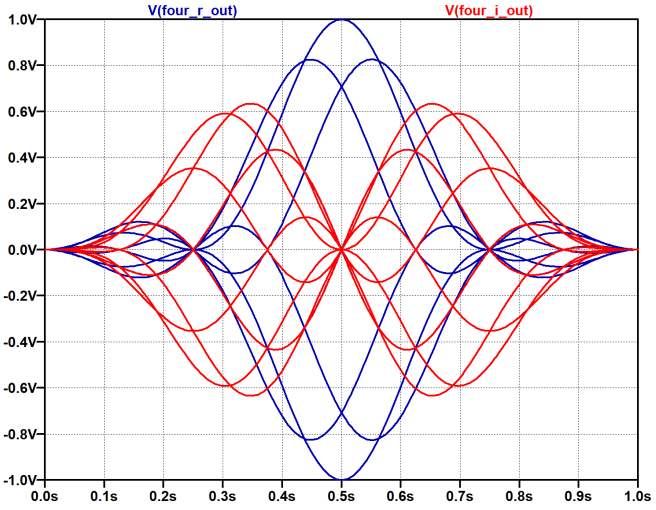


図 14. 図 13 のシミュレーション結果。でもこれだけでは何を見ているのかは分からない（青が実部、赤が虚部）

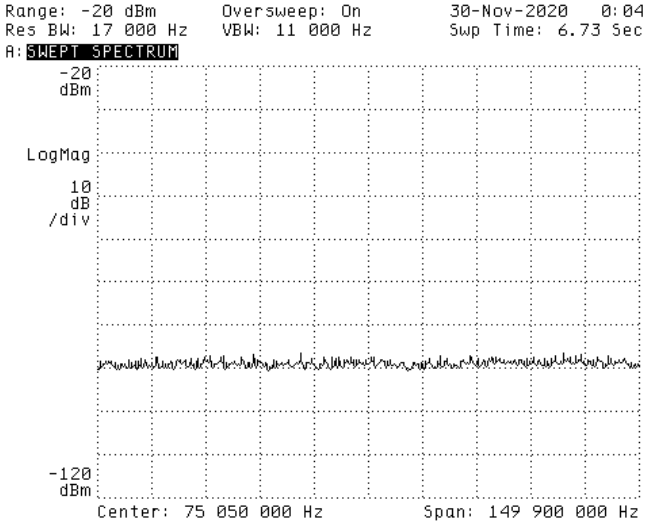


図 15. 周波数によらずレベルが一定な（一様に分布する）ホワイト・ノイズ

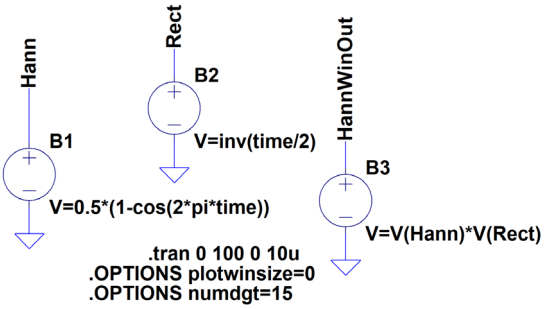


図 16. Hann 窓のノイズ等価帯域幅補正係数を計算する LTspice シミュレーション

① これに窓関数をかけたもの（つまり「窓関数によりフィルタリングしたもの」）から得られた 1 bin の電力Ω_P（この 1 bin の帯域幅をf_{BIN}でとします）と

窓関数フィルタの周波数特性の中央における通過電力を基準値とした 1Hz の矩形フィルタ内に存在する電力量Ω_{PSD}

との比がノイズ等価帯域幅f_{BIN_ADJ}になります。前回の TNJ-080 の式(14)を再掲すると、

$$\Omega_{PSD}(m) = \frac{\Omega_P(m)}{f_{BIN\_ADJ}} = \frac{\Omega_P(m)}{\gamma \times f_{BIN}} \ [\mathrm{W/Hz}] \qquad (4)$$

bin の周波数帯域幅f_{BIN}を 1Hz として、この式を変形すると

$$\gamma = f_{BIN\_ADJ} = \frac{\Omega_{P@1}}{\Omega_{PSD}} \qquad (5)$$

となり等価帯域幅補正係数γが得られることになります。繰り返しますが、$f_{BIN} = 1$Hzとしていますので、ここで得られたノイズ等価帯域幅f_{BIN_ADJ}がそのまま補正係数γになります。

前回のように「この補正係数は Hann 窓では 1.5」といっても、この算出方法を具体的に知りたい、どうすれば計算できるのかと思うのではないでしょうか。説明は[5]にも記載がありますが、いまひとつクリアではありません…。今回と次回の技術ノート・シリーズでは、これを紐解いてみましょう。

今回の技術ノートではまず連続信号で考えてみます。

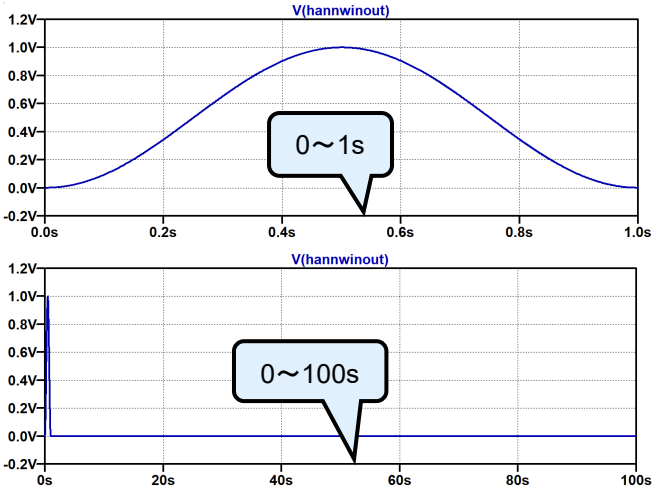


図 17. 図 16 のシミュレーション結果（上：0～1sec、下：0～100sec）

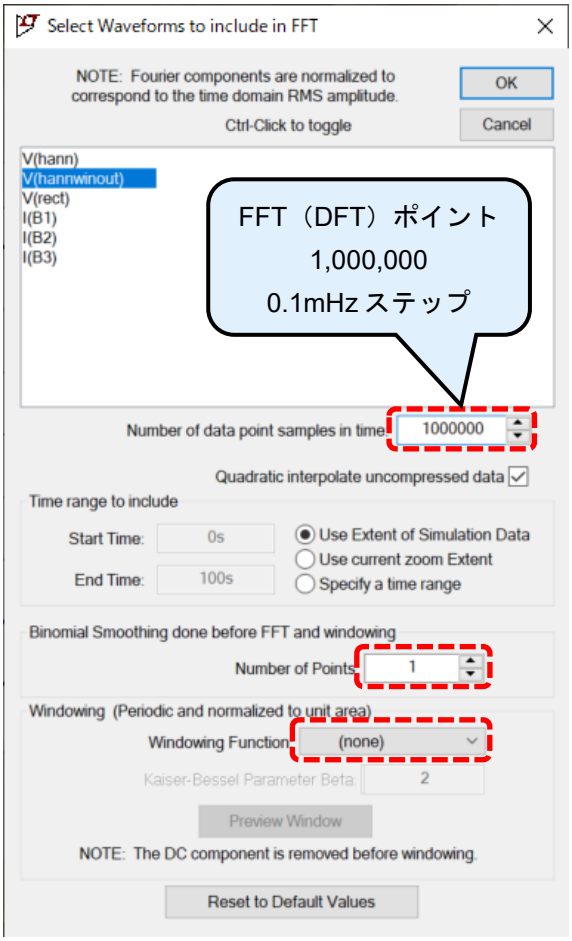


図 18. 図 17 の FFT 計算の条件を設定する

連続信号として LTspice でやってみる

Hann 窓を用いて計算してみます。Hann 窓関数は式(1)と同じです。計算する LTspice シミュレーションを図 16 に示します。これは窓関数全長を 1sec、FFT 結果としてスペクトルを正しく見せるためにシミュレーション全長を 100sec にしてあります。.OPTION numdgt=15 は計算精度を高めるためのディレクティブです。

時間軸のシミュレーション結果（計算結果）を図 17 に示します。これを FFT します。グラフ上で右クリックして View → FFT で図 18 のように設定します。Number of data point samples in time は 1,000,000 にして、シミュレーション全長 100s でサンプル周波数（最大周波数）10kHz、bin 分解能 10mHz を得るようにします。Number of Points は 1 にしてスムージングなし、Windowing Function は(none)にして、窓関数の設定もナシにします（窓関数の特性を求めるわけですから）。OK を押します。

これで計算した結果の軸などを調整したものを図 19 に示します。ラベルを CTRL＋左クリックすると帯域値を計算した結果が図 20 のように表示されます。ここで Power BW というのが、ここまで説明してきたノイズ等価帯域幅に相当します。

あまり知られていないようですが（そういう私も最近発見したのですが）、LTspice ではこのような操作によって、ノイズ等価帯域幅を表示してくれるのです。

図 20 のように、数値として 747.32mHz と表示されますが、これは bin の本来の帯域幅の半分です。これを折り返したものが本来の全帯域幅です。また窓関数自体の全長が 1sec ですから、1Hz の帯域幅に相当します。図 20 の 747.32mHz の倍の 1.49464Hz というのが、ここまで説明した「Hann 窓でのノイズ等価帯域幅補正係数 1.5」になるわけです。

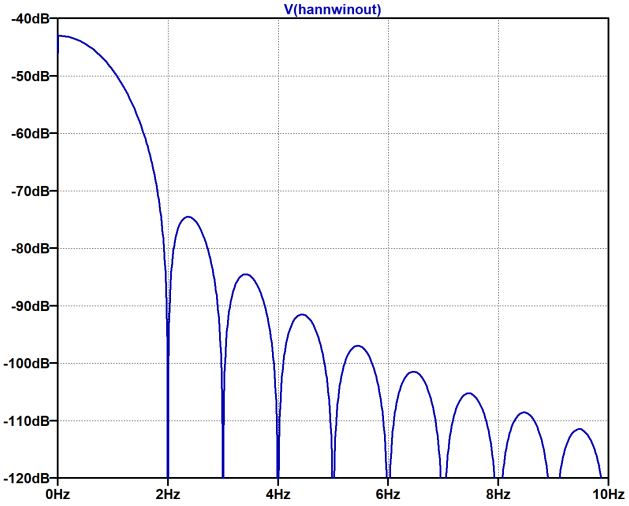


図 19. 図 18 の FFT 計算結果（軸などを調整）

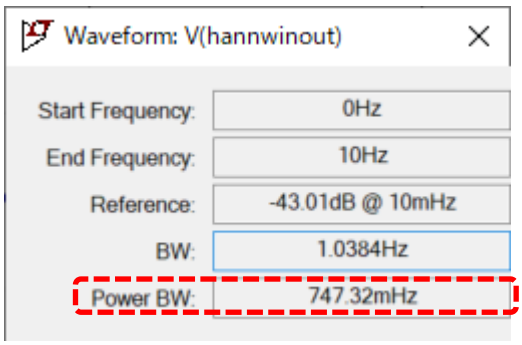


図 20. 図 19 で CTRL＋左クリックすると出てくる帯域計算結果

ノイズ等価帯域幅補正係数について疑問に思うこと「ノイズ等価帯域幅補正係数が 1 より大きいのはなぜか」

「Hann 窓でのノイズ等価帯域幅補正係数は 1.5」となります。これは 1 より大きいですね…。私は直観的にですが、ふと不思議に思いました。「なぜ窓関数をかけた、簡単にイメージしてみるとフィルタをかけたほうの FFT 結果で得られたノイズ等価帯域幅が、窓関数のない、フィルタをかけない FFT 結果のノイズ等価帯域幅より広くなるのか？」という疑問でした。

これは「窓関数をかける」という本質的な操作が、その直観的なイメージと異なるためです。「窓関数のない」というのは、別の言い方をすると「サンプル全体に矩形の窓関数をかけている」ということなのです。「Hann 窓などの窓関数をかけない（窓関数のない）＝矩形の窓関数」というのは、サンプル全体量をより多く取り込むということですから、より時間長が長いフィルタだと、つまりより帯域の狭いフィルタだと考えることができるということです。

まとめ

窓関数の減衰補正係数とノイズ等価帯域幅補正係数について、その基本的な考え方を探究してみました。LTspice を使って連続関数（連続時間系）では、きちんとつじつまが合うことが分かりました。

これまで窓関数について、こんなことを考えたことはありませんでした。深く突っ込んだこともありませんでした。TNJ-080 で、FFT した結果から 1Hz ノイズ電力密度を求めるために、初めて触れたこれらの窓関数補正係数でしたが、探究してみると『ホント趣き深いなぁ』と、窓関数というその世界の広がりに感心した今回の技術ノートでした。

次回はいよいよ、離散信号での減衰補正係数とノイズ等価帯域幅補正係数について検討してみます。その検討の道のりはまた「山あり谷あり」で、自分の知識の無さも露呈してしまったことから、かなり時間がかかり、そしてその『趣き深さ』（と自分の無能さ）に、またもやため息をついたものでした。

ということで、どうぞお楽しみにお待ちください！

参考文献

[1] 石井 聡; 正弦波を A/D 変換し窓関数なしに打ち切って FFT してみると, 回路設計 WEB ラボ, TNJ-008, アナログ・デバイセズ

[2] Michael Cerna, Audrey F. Harvey; The Fundamentals of FFT-Based Signal Analysis and Measurement, Application Note 041, National Instruments

[3] 窓関数, Wikipedia, https://ja.wikipedia.org/wiki 窓関数

[4] Window function, Wikipedia, https://en.wikipedia.org/wiki/Window_function

[5] Keysight Technologies, Inc., Window Shapefactor and Equivalent Noise Bandwidth, http://rfmw.em.keysight.com/wireless/helpfiles/89600B/WebHelp/Subsystems/gui/content/windows_shapefactor_and_equiv_noisebw.htm

[6] Mathworks, enbw 等価ノイズ帯域幅, https://jp.mathworks.com/help/signal/ref/enbw.html

[7] Mathworks; hann (Hanning) window, https://www.mathworks.com/help/signal/ref/hann.html

[8] Alan V. Oppenheim, Ronald W. Schafer; Discrete-Time Signal Processing, Prentice Hall

[9] Harris, Frederic J.; Windows, Harmonic Analysis, and the Discrete Fourier Transform, Sep. 1976. Naval Undersea Center, https://apps.dtic.mil/dtic/tr/fulltext/u2/a034956.pdf