

Convolution: A visual Digital Signal Processing (DSP) tutorial

R.C. Kim (02-21-2014; updated 01-02-2015)

Introduction:

Fourier theory says that *any* periodic signal can be created by adding together different sinusoids (of varying frequency, amplitude and phase). In many applications, an unknown analog signal is sampled with an A/D converter and a Fast Fourier Transform (FFT) is performed on the sampled data to determine the underlying sinusoids.

In this 7-step tutorial, a visual approach based on *convolution* is used to explain basic Digital Signal Processing (DSP) up to the Discrete Fourier Transform (DFT). The DFT is explained instead of the more commonly used FFT because the DFT is much easier to understand. (The DFT is equivalent to the FFT except the DFT is far less computationally efficient.)

In particular, convolution is shown to be the key to understanding basic DSP. Also, some of the concepts are far more intuitive in the frequency-domain vs. the more familiar time-domain.

Note: The distinction between continuous and discrete systems is blurred in this tutorial since the concepts are similar for both. Most of the following figures show continuous signals (except where as noted) for clarity because sampled signals are literally discrete (i.e. "dots") and have the added complication of spectral replicas as will be explained in Step 6.

Step 1: Convolution review

Any *linear system's* output, $y(t)$, can be determined by the equation: $y(t) = h(t) * x(t)$ where $x(t)$ is the input; $h(t)$ is the system's *impulse response* and "*" represents *convolution*. A system's response to an *impulse input* tells us the *complete* frequency response of that system. It does so because an impulse, which is a mathematical abstraction, consists of an infinite number of sinusoids of all frequencies, i.e. it excites a system equally at all frequencies.

An ideal impulse is infinite at $t=0$ and 0 elsewhere in the time-domain (**Figure 1(a)**). **1(b-c)** shows an ideal impulse is approximated by adding up an increasing number of sinusoids. (The sinusoids add up to create an infinite spike at $t=0$ and cancel to 0 everywhere else.)

Figure 1(d) shows the ideal impulse in the frequency-domain (it's a *flat, infinite* horizontal line since all frequencies are equally represented with the same magnitude).

1(a) and **1(d)** are an example of *duality* between the time and frequency domains: *narrow signals in one domain are wide in the other domain.*

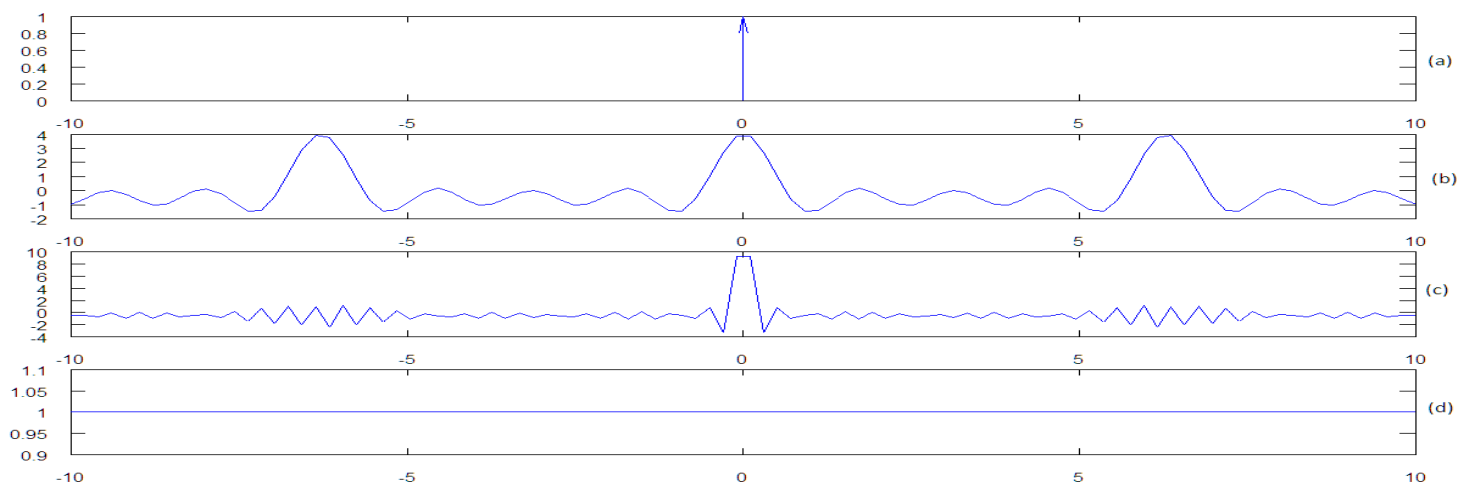


FIGURE-1: Impulse Function: (a) Ideal impulse in time-domain; (b) Sum of 10 sinusoids; (c) Sum of 75 sinusoids; (d) Ideal impulse in frequency-domain

For *discrete systems*, an impulse is 1 (not infinite) at $n=0$ where n is the sample number, and the discrete convolution equation is $y[n] = h[n] * x[n]$.

The *key idea* of discrete convolution is that *any* digital input, $x[n]$, can be broken up into a *series of scaled impulses*. For discrete linear systems, the output, $y[n]$, therefore consists of the sum of scaled and shifted impulse responses, i.e. convolution of $x[n]$ with $h[n]$.

Figure 2(a-f) is an example of discrete convolution. (Don't worry if you don't understand it yet- it should be clearer after Step 4.)

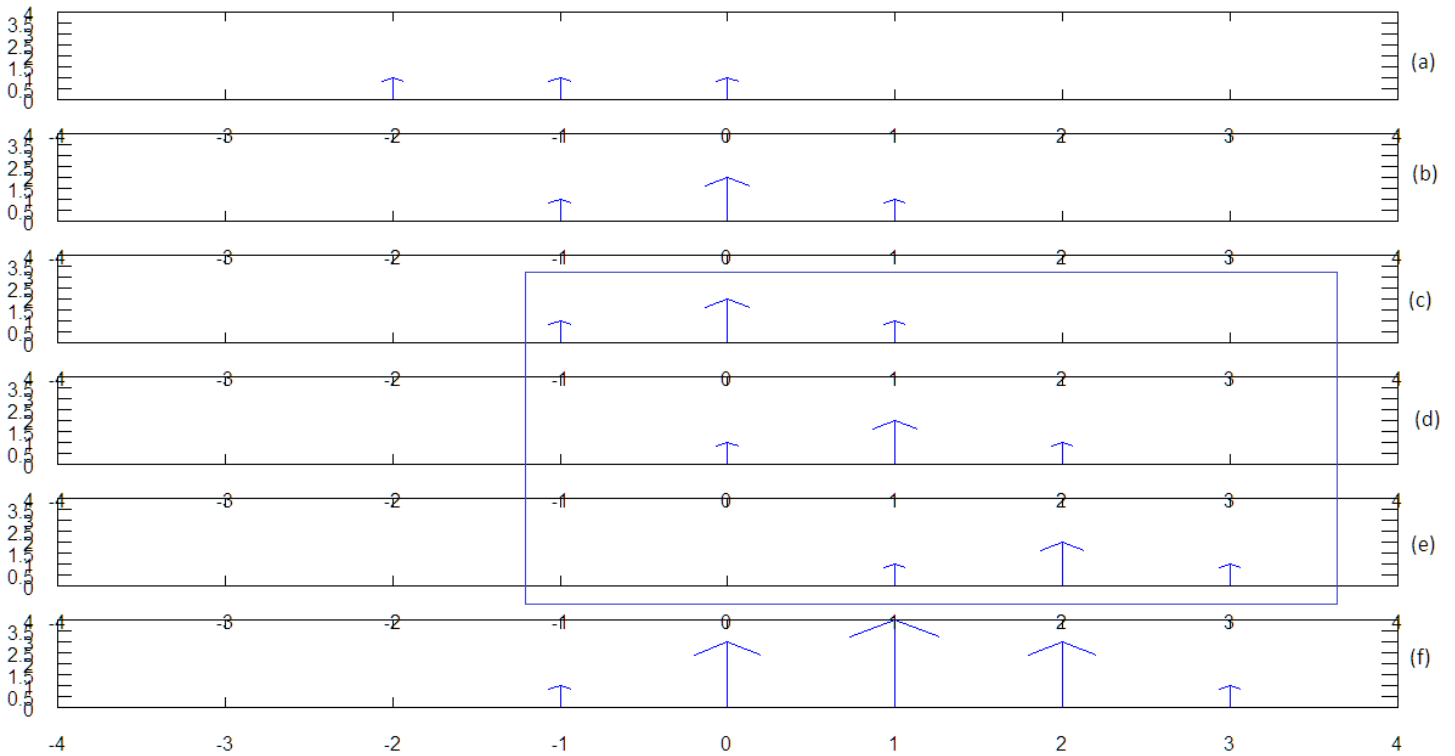


FIGURE-2: Discrete Convolution example: (a) $x[n]$ input; (b) $h[n]$ impulse response (c-e) shifted impulse responses; (f) $y[n]$ =sum of c-e

It can be proven (but it won't here) that convolution in the time-domain is equivalent to multiplication in the frequency-domain and vice-versa: $x[n] * h[n] \leftrightarrow X[f] \cdot H[f]$ ¹.

Likewise, convolution in the frequency-domain is equivalent to multiplication in the time-domain and vice-versa: $x[n] \cdot h[n] \leftrightarrow X[f] * H[f]$.

Step 2: Sinusoids

Sinusoids are important because any periodic signal can be written as the sum of sinusoids. If you understand how DSP works with sinusoids, then you can easily extend that understanding to any linear signal.

¹ $X[f]$ and $H[f]$ are the frequency-domain versions of $x[n]$ and $h[n]$.

We start by looking at the Sine and Cosine functions in the familiar time-domain (**Figure 3(a)**). They're the same except that Cosine is $+90^\circ$ ($+\pi/2$ radians) ahead of Sine.

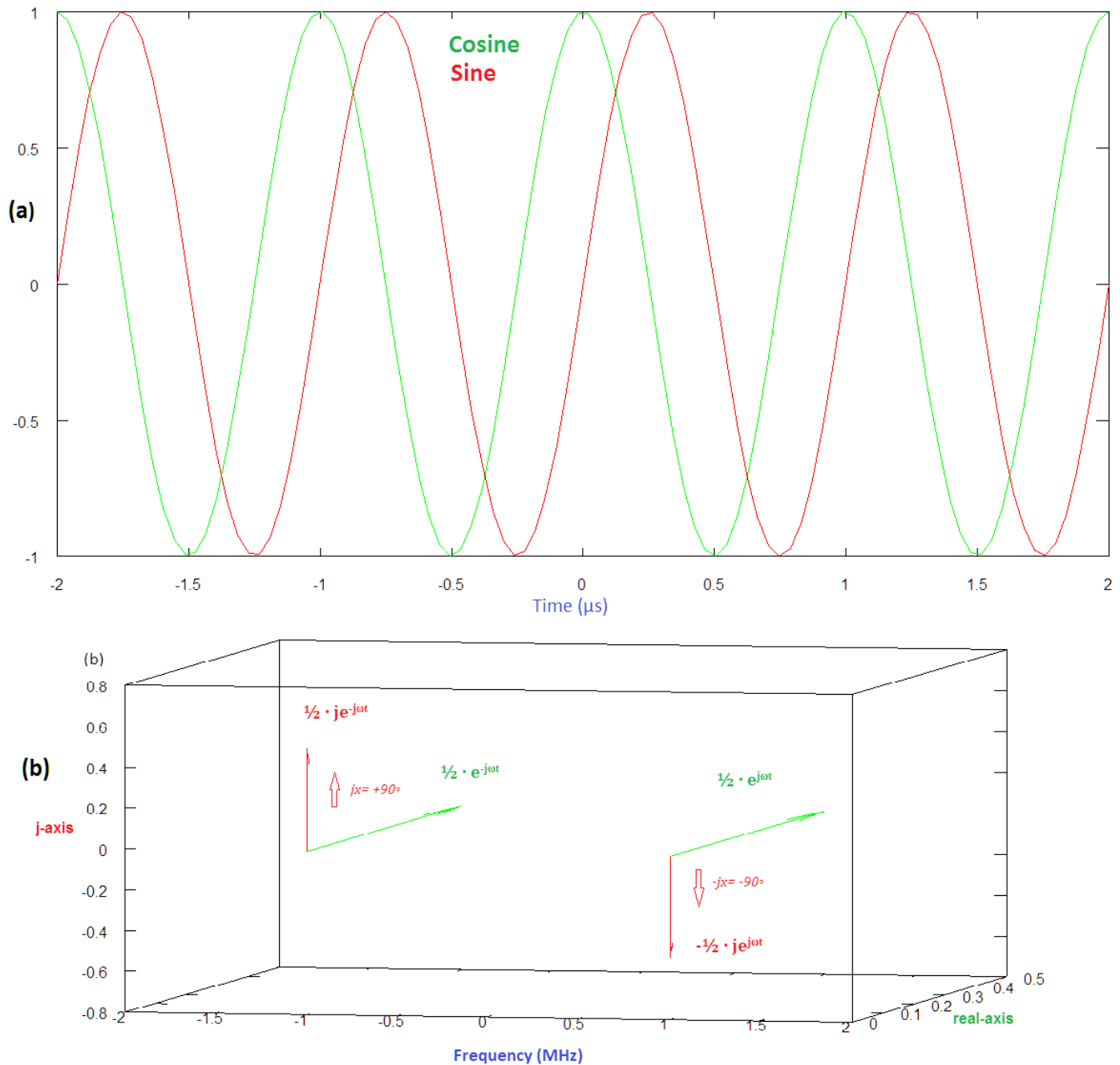


FIGURE-3: Cosine (green) vs. +Sine (red) @ 1 MHz: (a) time-domain; (b) frequency-domain

Sine is an *odd* function because it's not symmetric at $t=0$ ($\sin t = -\sin(-t)$). Cosine is an *even* function since it's symmetric at $t=0$ ($\cos t = \cos(-t)$). Cosine is non-zero at 0- this means that any *DC value* (constant offset) in a signal is due solely to its cosine component(s) since sine is 0.

Now look at **Figure 3(b)**, which shows the frequency-domain representation of sine and cosine. The derivation of these *Fourier transforms (FT)* won't be shown- however, it's still possible to develop an intuitive feel for the frequency-domain representation.

A sinusoid consists of one frequency, so it should be a *single* line in the frequency-domain. Two lines are shown (one positive and one negative) because *two* directions are possible. We saw that cosine is an even function in the time-domain and the same symmetry is seen in the frequency-domain. Sine is an odd function and the asymmetry is also seen in the frequency-domain, i.e. $\sin f = -\sin(-f)$. In the time-domain, cosine is 90° ahead and we see the 90° phase difference in the frequency-domain as well.

Negative frequency will become clearer in Step 3.

Step 3: Complex numbers

Now we introduce the famous **Euler equation**: $e^{\pm j\omega t} = \cos \omega t \pm j \sin \omega t$ where $\omega = 2\pi f$.

By rearranging terms, the following equations also shown in **Figure 3(b)** can be derived:

$$\cos \omega t = \frac{1}{2} \cdot (e^{-j\omega t} + e^{j\omega t})$$

$$\sin \omega t = \frac{1}{2} \cdot (e^{-j(\omega t - \pi/2)} + e^{j(\omega t - \pi/2)}) = \frac{1}{2} \cdot (e^{-j\omega t} \cdot e^{+j\pi/2} + e^{j\omega t} \cdot e^{-j\pi/2}) = \frac{1}{2} \cdot (je^{-j\omega t} - je^{j\omega t})$$

Again, the full derivations won't be shown here- however, *using* these equations is actually straightforward.

First of all, *j* simply represents a +90° phase-shift (rotation) without any change in frequency or magnitude. ($j = \sqrt{-1}$, so if multiplication by -1 causes 180° rotation, then half of that is +90°.)

A more formal definition is to expand the Euler equation to include a phase offset ϕ : $e^{j(\omega t + \phi)} = \cos(\omega t + \phi) + j \sin(\omega t + \phi)$. If $\omega = 0$ and $\phi = \pi/2$ (90°), then $e^{j\pi/2} = \cos \pi/2 + j \sin \pi/2 = 0 + j(1) = +j$.

In **Figure 4**, Sine has been rotated by 90°, and is plotted next to cosine. The 3-d sum is a rotating spiral in the time-domain. If Sine is positive, the spiral rotates in the counter clockwise (CCW) direction. If Sine is negative, the spiral rotates in the clockwise (CW) direction (that's all there is to negative frequency.) The speed of rotation (number of complete circles per second) is simply the frequency.

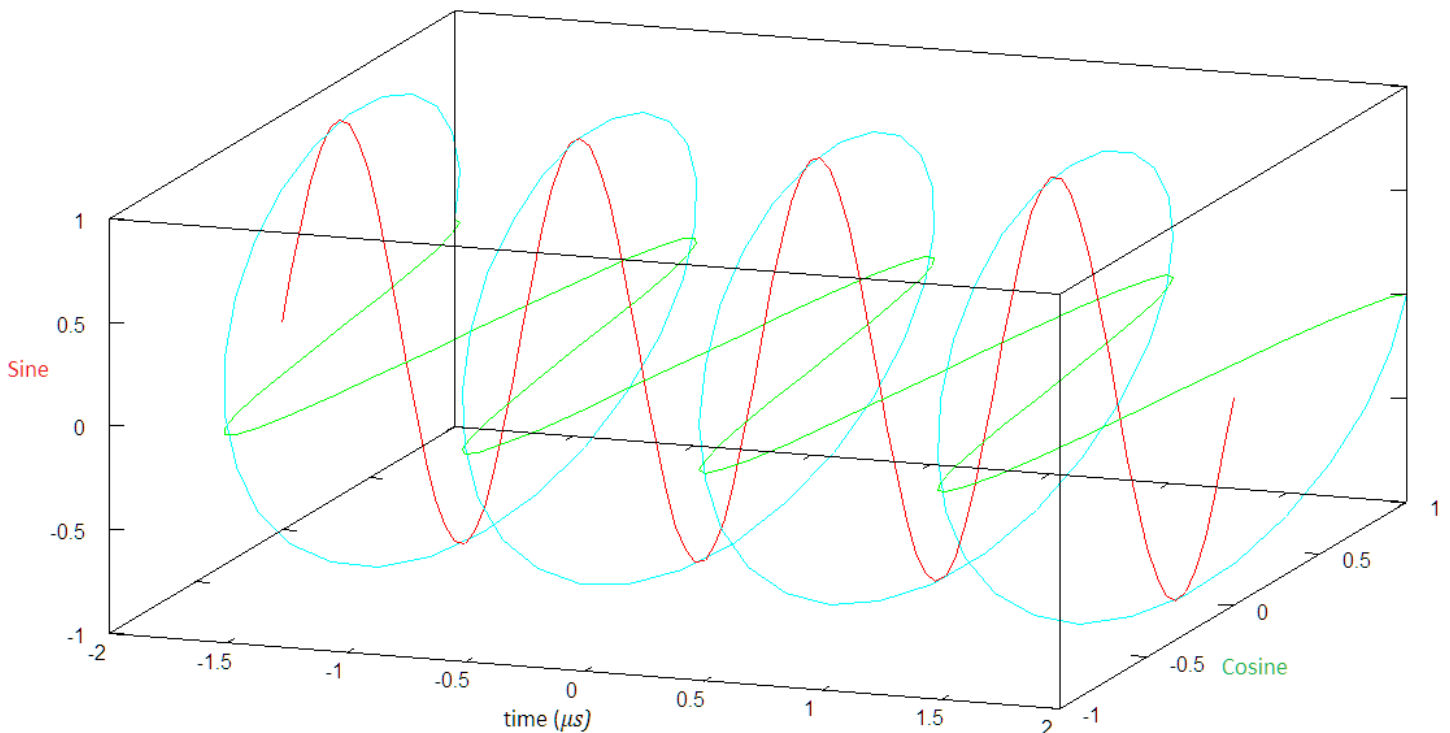


FIGURE-4: Complex-number in the time-domain

In **Figure 5(a-c)**, we look at the frequency-domain version of a complex number. If we rotate Sine by $+90^\circ$, then it looks like **5(b)**. If we then add $\cos A + j\sin A$, one of the lines cancels out, and we're left with a **single side-band (SSB)** vector shown in **5(c)**. This single vector is static, i.e. it does not rotate or move at all.

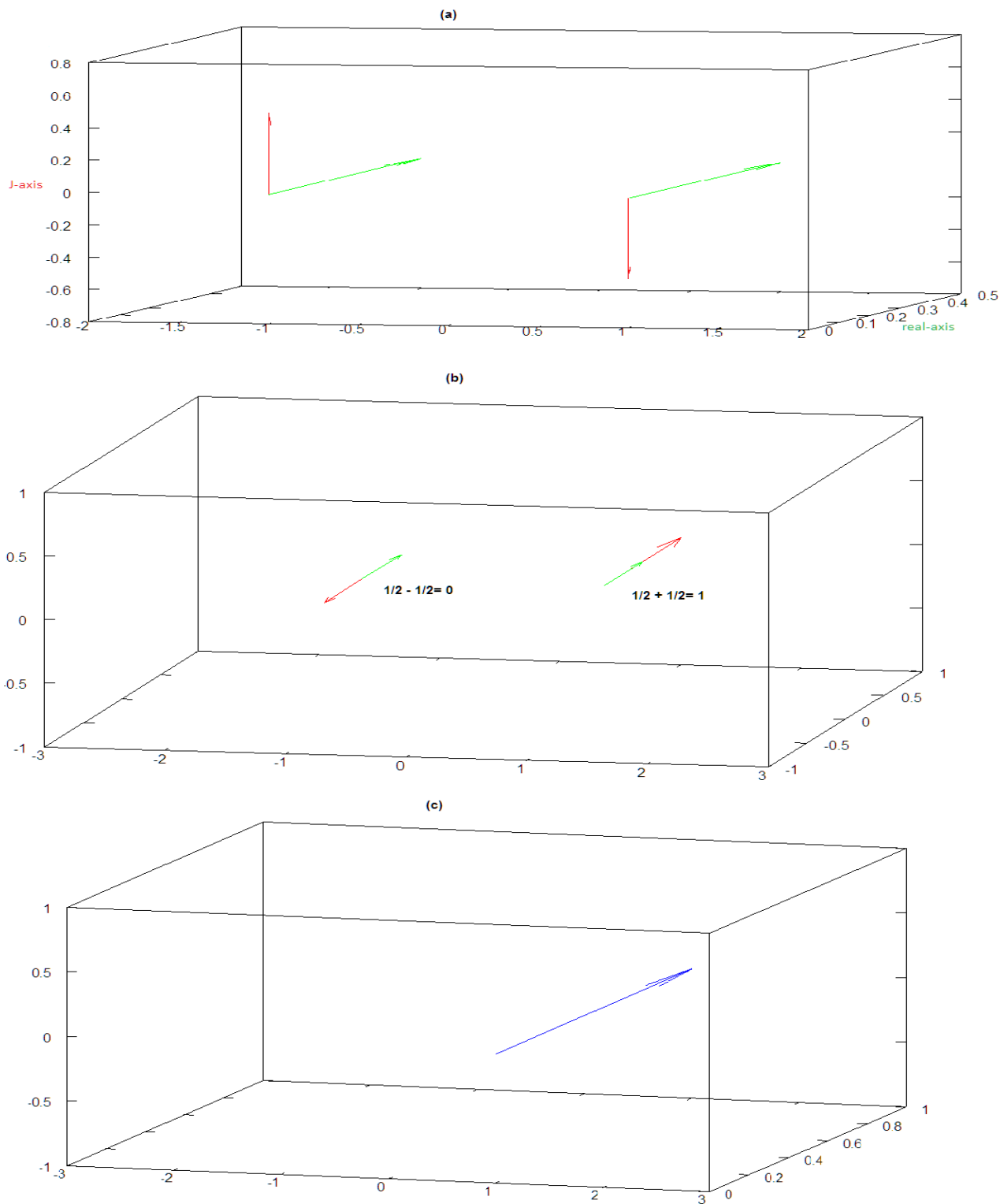


Figure-5: Complex number in the frequency-domain (MHz): (a) $\cos A$ and $\sin A$; (b-c) $\cos A + j\sin A$

Note: For $\cos A - j\sin A$, the SSB vector would be at $f = -A$.

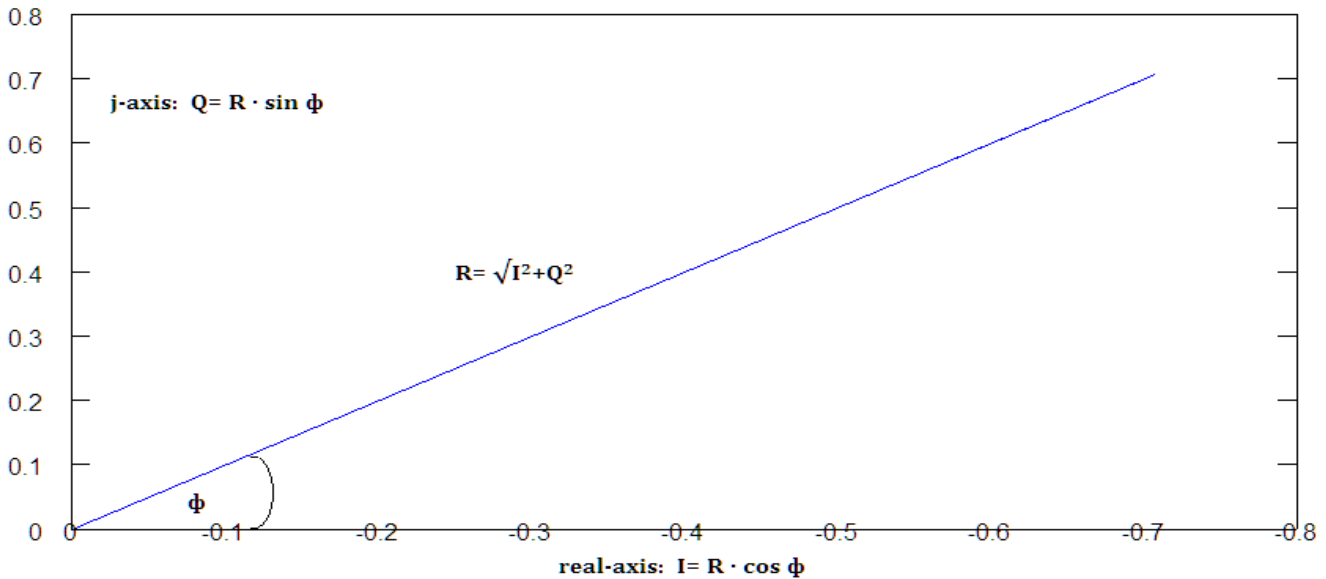


Figure-6: Polar form

Figure 6 shows the *polar form*, which is obtained from a side-view of either Figure 4 or 5(c).

For the polar form, it's common to use the notation: $R \angle \phi$ where $R = \sqrt{I^2 + Q^2}$ and $\phi =$ phase. I is the *real (in-phase)* component and Q is the *imaginary (quadrature-phase)* component as shown in Figure 6².

In the time-domain, the polar form is like a *rotating* clock hand ($\phi = \omega t +$ a phase-constant).

In the frequency-domain, the polar form is like a broken clock- the clock hand does **not** move ($\phi =$ a phase-constant).

Step 4: Convolution with Sinusoids

Most of the time, convolution is thought of only in terms of a system's output as shown in Step 1. However, it's actually a mathematical operation in its own right as shown next.

Convolution is complicated and requires calculus when both operands are continuous waveforms. But when one of the operands is an impulse (delta) function, then it can be easily done by inspection.

The rules of discrete convolution are (not necessarily performed in this order):

- 1) Shift either signal by the other (convolution is commutative).
- 2) Multiply the magnitudes (and add any overlapping lines).
- 3) Add the phases.

² I/Q are always perpendicular to each other regardless of ϕ . For a single complex number, any phase-offset can be set to 0 since the axis-orientation is arbitrary. Phase really only matters when plotting multiple complex numbers, which can have any phase-offset with respect to each other.

Note: I/Q have the form: $i[n] + jq[n]$ in the time-domain; and $I[f] + jQ[f]$ in the frequency-domain.

Cosine-Cosine example:

A simple example is the well-known trig identity: $\cos A \cdot \cos B = \frac{1}{2} \cdot \cos (A+B) + \frac{1}{2} \cdot \cos (A-B)$.

Figure 7(a-c) shows the equivalent operation in the frequency-domain. As explained in Step 1, multiplication in the time-domain is convolution in the frequency-domain. Convolution is performed by “lifting” the function in **7(a)**, which is centered at $f=0$, and shifting the center to both $f=-2$ and $f=+2$ (**7(b)**). After multiplying the magnitudes ($\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$), the result in **7(c)** agrees with the trig identity. In this case, the phases are both 0 so rule #3 is N/A.

Figure 7(d-f) shows the case when $A=B$. The two “overlapping” lines at $f=0$ add up to create a DC value ($\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$).

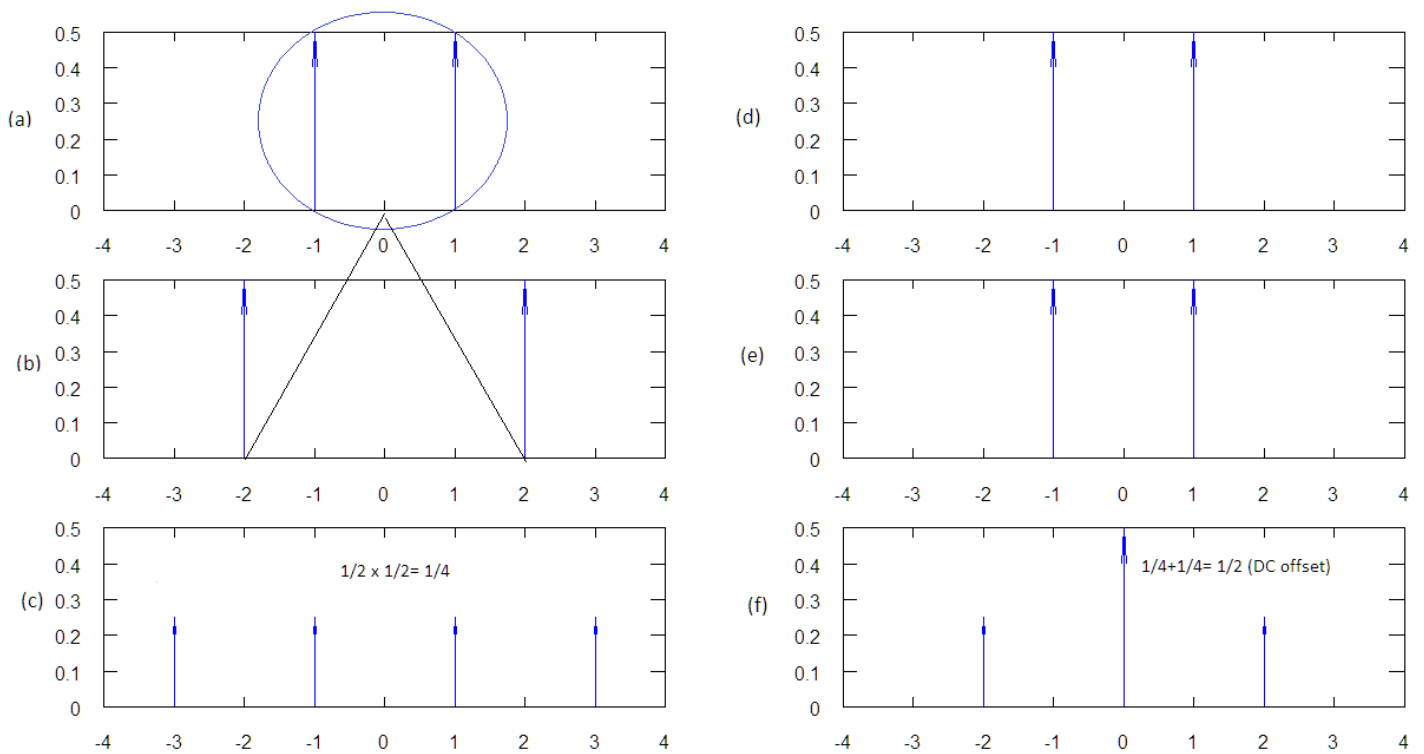


FIGURE-7: Cosine-Cosine convolution in the frequency-domain (real-axis only): (a-c) for different f ; (d-f) for the same f

j-example:

The 3rd rule of convolution is that phases add. Multiplying by j in the time-domain is convolution in the frequency-domain. Since j 's magnitude is 1 and $f=0$, the only effect is rotation by $+90^\circ$ (or -90° with $-j$) without any frequency-shift.

Figure 8(a) show a complex number at $f=1$ ($1 \angle 0^\circ$ in *polar notation*) and j ($1 \angle 90^\circ$) at $f=0$. After multiplication with j , the complex number is rotated to $1 \angle 90^\circ$ in **8(b)**. **8(c)** shows a 2nd shift back to the real (negative) axis ($1 \angle 90^\circ \cdot 1 \angle 90^\circ = 1 \angle 180^\circ$) after multiplying by j again.

Multiplying by j twice more (not shown) would result in $1 \angle 180^\circ \cdot 1 \angle 90^\circ = 1 \angle 270^\circ$; and then $1 \angle 270^\circ \cdot 1 \angle 90^\circ = 1 \angle 360^\circ = 1 \angle 0^\circ$.

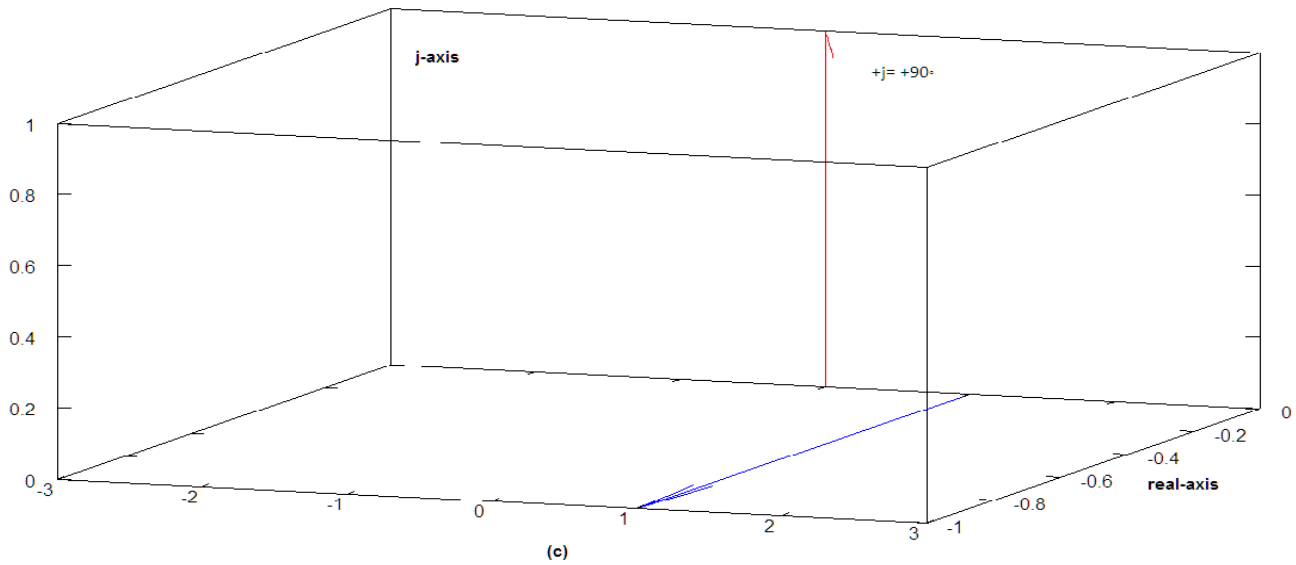
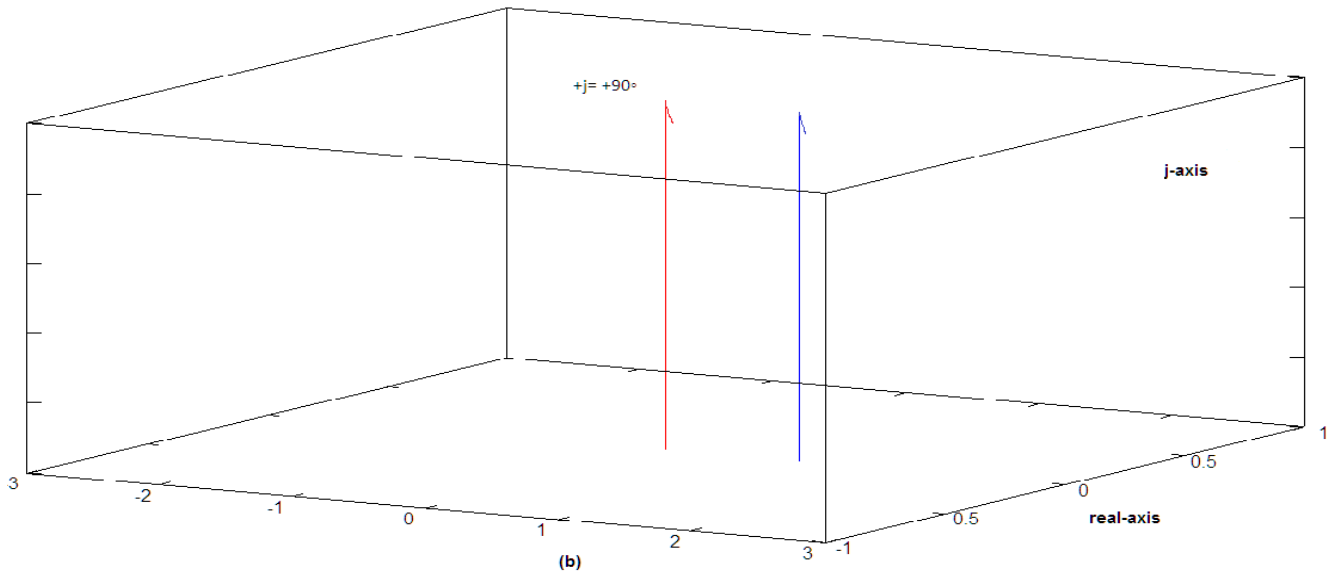
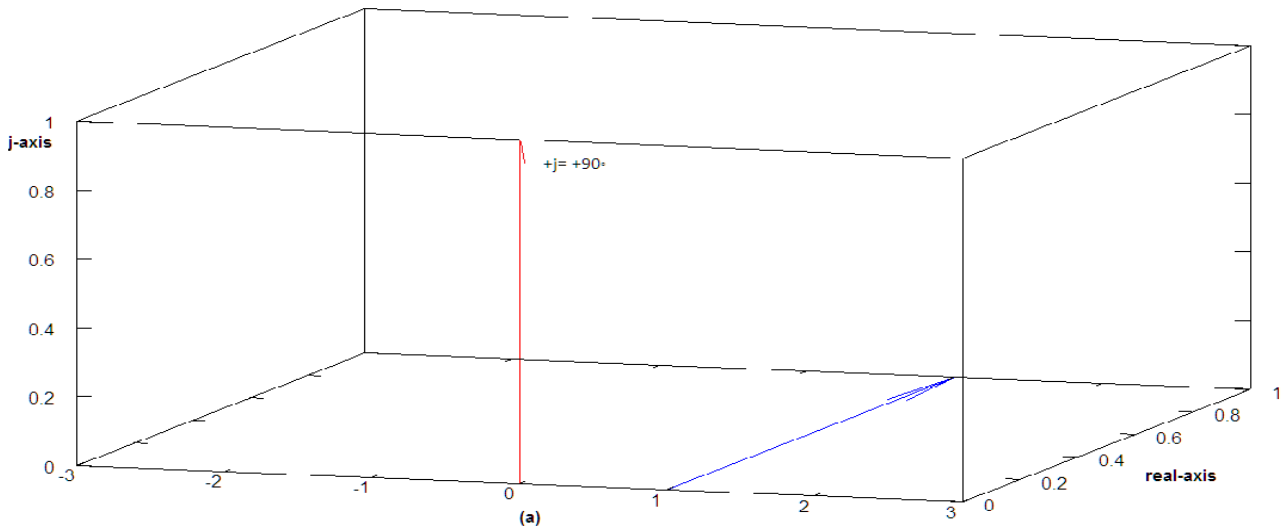


Figure-8: J-Convolution in the frequency-domain: (a) complex number at $1 < 0^\circ$; (b) complex number at $1 < 90^\circ$; (c) complex number at $1 < 180^\circ$

Sine-Sine example:

Figure 9(a-d) shows the case for $\sin A \cdot \sin B = \frac{1}{2} \cdot \cos(A-B) - \frac{1}{2} \cdot \cos(A+B)$.

9(e-h) shows the case for $\sin A \cdot \sin A$. Like **7(f)**, the final result includes a non-zero value at $f=0$ (DC).

In **9(d)** and **9(h)**, **9(c)** and **9(g)** are rotated respectively to the real-axis per rule #3 ($90^\circ + 90^\circ = 180^\circ$ like **8(c)**).

Note: Other trig identities can be shown as well. E.g. adding **Figures 7(f)** and **9(h)** shows that $\sin^2 A + \cos^2 A = 1$.

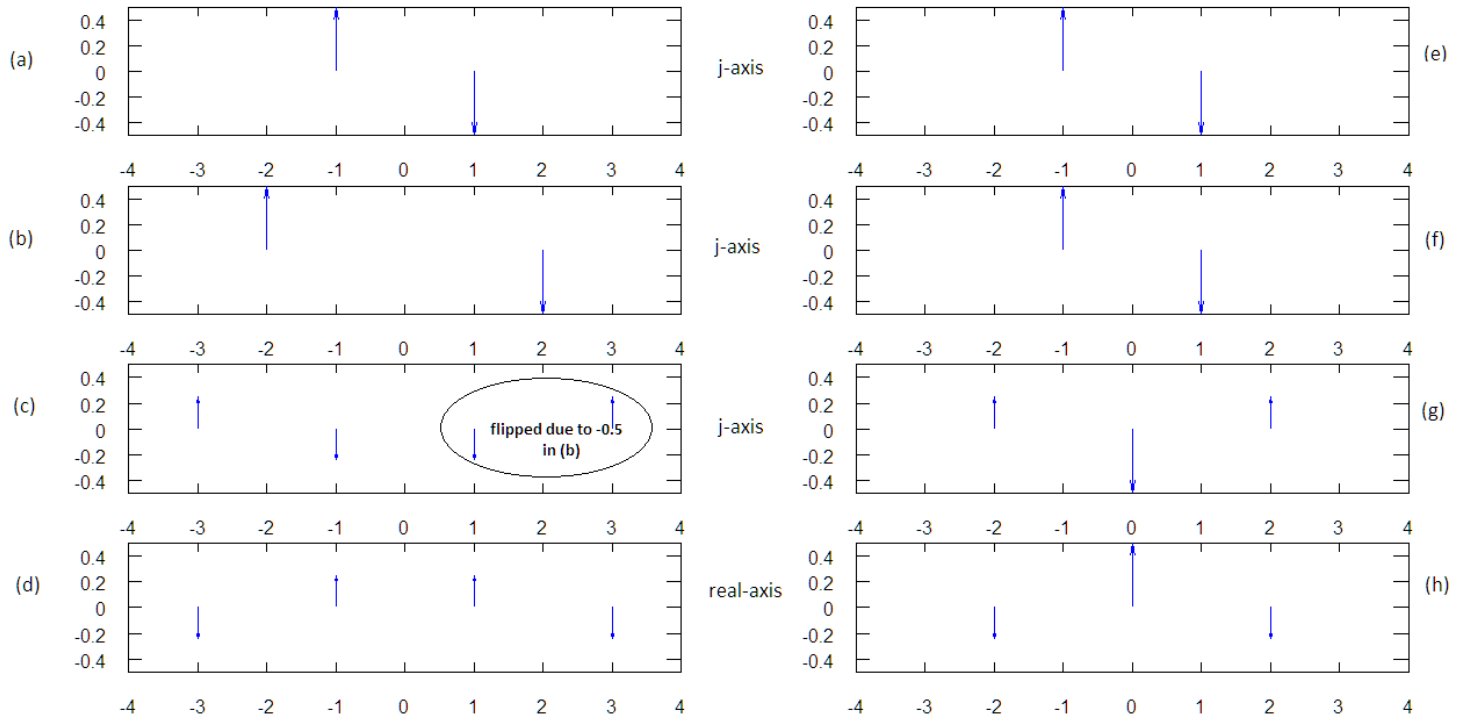


Figure-9: Sine-Sine convolution in the frequency-domain: (a-d) for different f; (e-h) for the same f

Cosine-Sine example:

Figure 10(a-d) shows the case for $\cos A \cdot \sin A = \frac{1}{2} \cdot \sin(A+A) + \frac{1}{2} \cdot \sin(A-A)$. **10(e-h)** shows the case for $\cos A \cdot (-\sin A)$.

In **10(c-d)** and **10(g-h)**, the result has been rotated to the j-axis per rule #3 ($0^\circ + 90^\circ = 90^\circ$ like **8(b)**).

In **10(d)** and **10(h)**, there's no DC value at $f=0$ even though both sinusoids have the same frequency because it cancels out to 0.

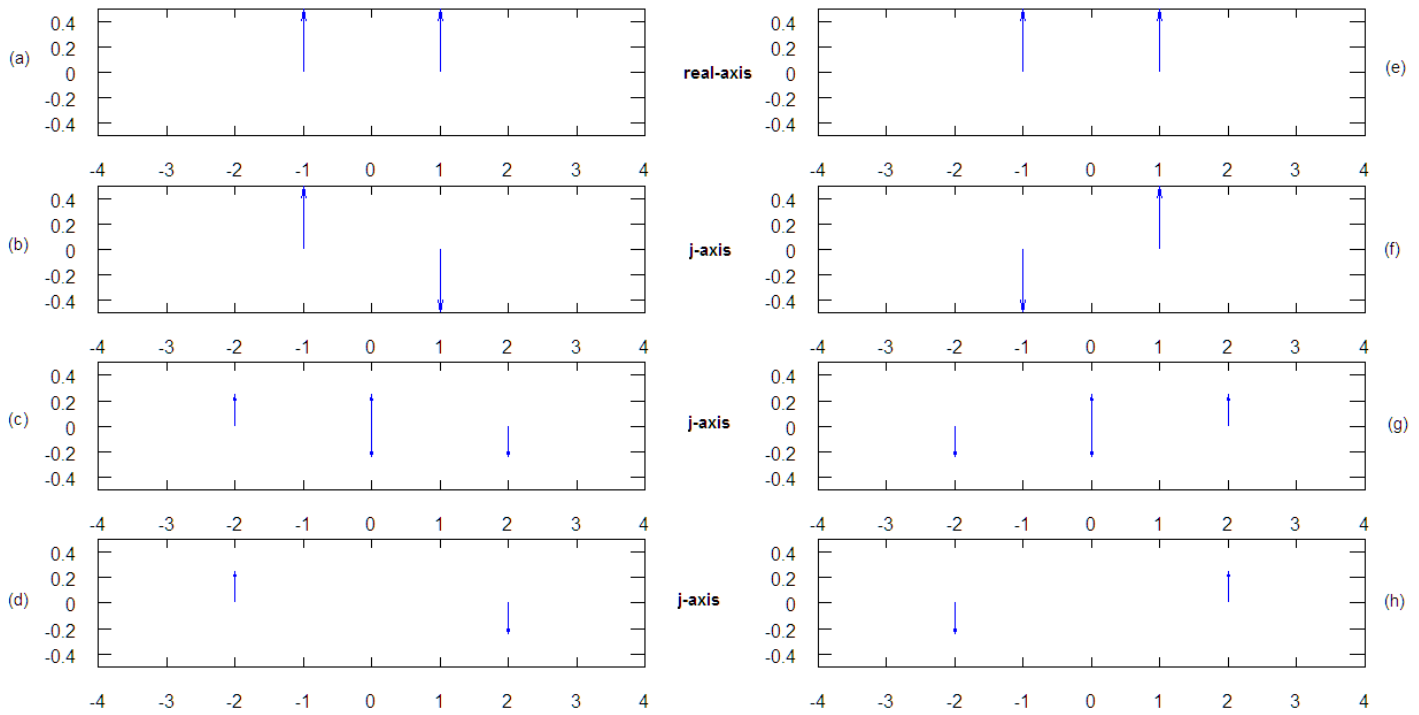


Figure-10: Cosine-Sine convolution in the frequency-domain: (a-d) same f with + Sine; (e-h) same f with -Sine

Key point: If two sinusoids are multiplied in the time-domain (convolution in the frequency-domain), a non-zero value at DC ($f=0$) only occurs when the sinusoids have the **same** frequency and their phase offset is **not** 90° . If the frequencies match, the DC value is max when the phase offset is 0° ($\cos A \cdot \cos A$) and min (0) when the phase offset is 90° ($\cos A \cdot \sin A$). The DC value due to a frequency match is the basis for the DFT.

The time-domain equivalent is that the average of a sinusoid over integer multiples of a *full period* is 0 because it has equal positive and negative area. When two sinusoids are multiplied, the time-domain average is still 0 unless the aforementioned conditions are met.

Step 5: Convolution with Complex Numbers

In Step 4, it was shown that multiplying/convolving two Cosines (with the same frequency) or two Sines (with the same frequency) results in a DC value. For simplicity, **Figures 7-10** showed $0/90^\circ$ phase offsets. For other phases, the DC value is divided between the real and imaginary axes per the I/Q equations in **Figure 6**.

Now, we show how to recover the full magnitude regardless of the phase-offset.

Up front, the answer is simply $e^{j\omega t} \cdot e^{-j\omega t} = e^0 = 1$. This equation is the key to understanding the DFT (as well as frequency translation for up/down-conversion).

Since $e^{j\omega t} = \cos \omega t + j \sin \omega t$, the above equation can be written as $(I + jQ) \cdot (\cos A - j \cdot \sin A)$, if we substitute $A = \omega t$ and $I + jQ = \cos A + j \sin A$. This works out to $I_{DC} + jQ_{DC} = (I \cdot \cos A + Q \cdot \sin A) + j(Q \cdot \cos A - I \cdot \sin A)$. Remember that $j^2 = -1$ and $-j^2 = +1$.

Figure 11(a-f) shows the equivalent convolution in the frequency-domain. When we sum up the lines we get a DC value (with full magnitude=1, not just $1/2$) at $f=0$ and *nothing else*.

We already showed these cases in **Figures 7-10**, so **Figure 11** just sums up the results: **11(a-c)** = $I \cdot \cos A + Q \cdot \sin A$; **11(d-f)** = $Q \cdot \cos A - I \cdot \sin A$. **11(f)**, which is 0 in this case, is flipped to the real-axis due to multiplication by j .

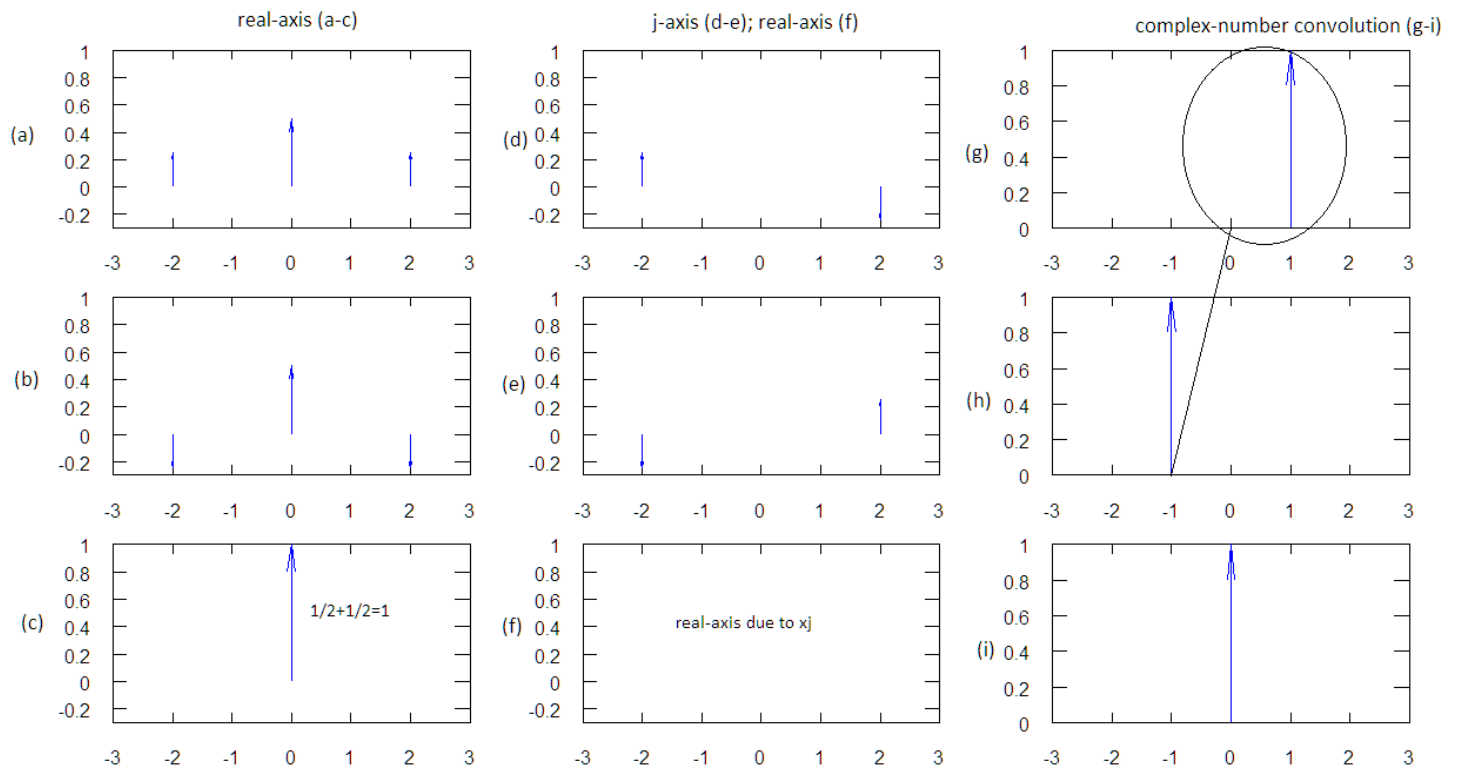


Figure-11: Complex-number convolution: (a) from 7(f) + (b) from 9(h)= (c); (d) from 10(d) + (e) from 10(h)= (f); (g) * (h)= (i)=11(c/f)

This is a lot of work, which is unnecessary, if you just convolve the SSB complex numbers instead as shown in **11(g-i)**.

So far, we've been looking at cases with 0/90° phase offsets. But this works regardless of phase. The rules of convolution are the same, of course, as the rules to multiply exponentials such as Euler's eq.

Figure 12 shows two examples of convolution between complex numbers which have a phase-offset (3d-view).

12(a-c) gives a SSB result with a phase offset: $e^{(j\omega t + \pi/4)} \cdot e^{-j\omega t} = e^{j\pi/4}$

In this example, the phase is non-zero (so the magnitude is divided between the real and j-axis), but the full magnitude=1 can be easily calculated from $\sqrt{(I^2+Q^2)}$, which is all we care about.

12(d-f) gives a SSB result with a 0° offset (I= 1 and Q=0): $e^{(j\omega t + \pi/4)} \cdot e^{-(j\omega t + \pi/4)} = e^0=1$

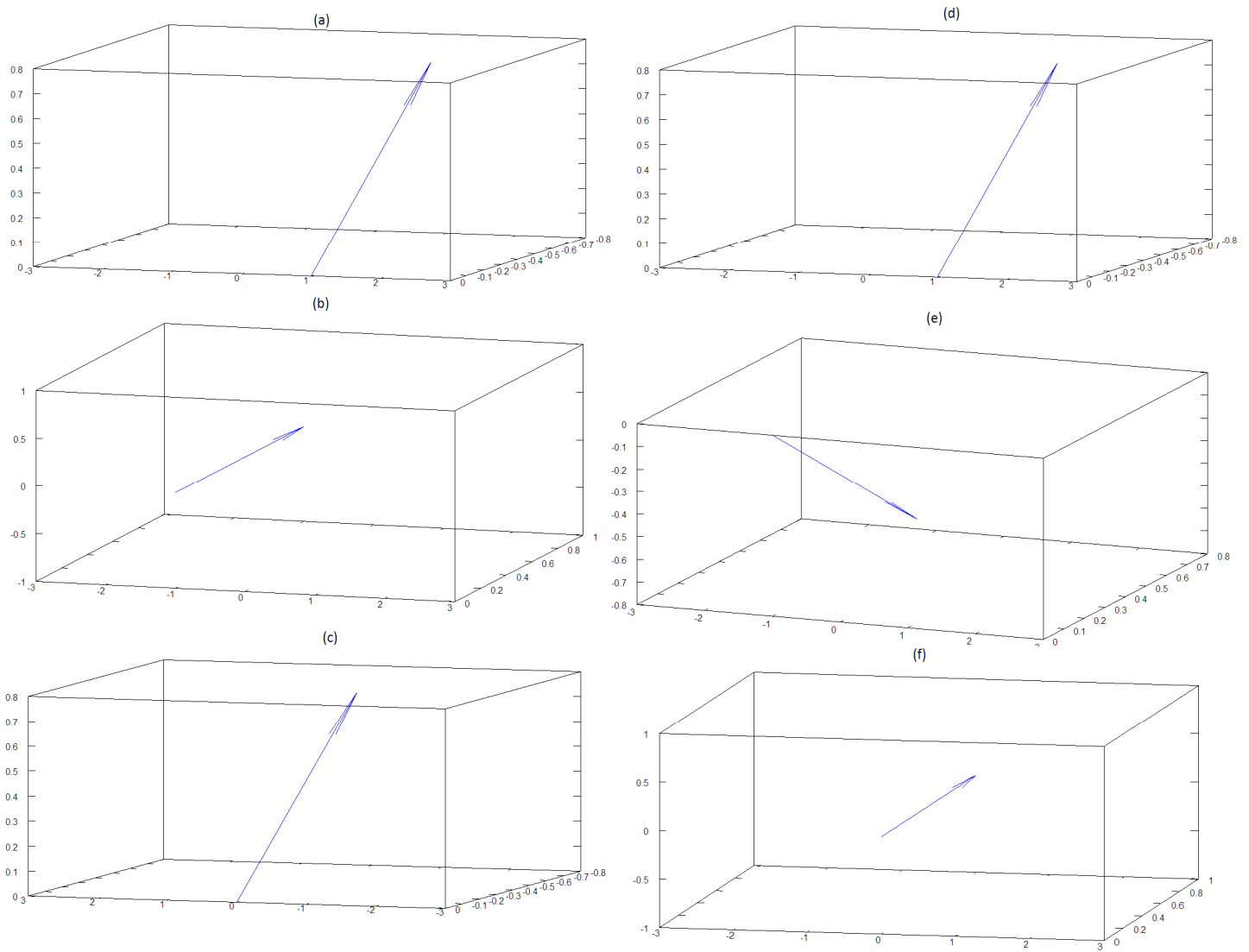


FIGURE-12: 3-D complex-number convolution: (a) * (b) = (c) with non-zero phase; (d) * (e) = (f) with 0 phase

We conclude Step 5 with the complex version of the DFT equation.

$$X[k] = \sum_n x[n] \cdot e^{-j2\pi kn/N} = x[n] \cdot (\cos(2\pi kn/N) - j\sin(2\pi kn/N)) \text{ with } n=0 \dots N-1$$

The DFT consists of two loops: $n=0$ to $N-1$ and $k=0$ to $N-1$ where N is the number of samples. The inner loop ($n=0$ to $n=N-1$) steps through the discrete values of the input data, $x[n]$, multiplied by the discrete values of a complex number of a particular frequency ($n \cdot F_s/N$).

The outer loop ($k=0$ to $k=N-1$) advances the frequency of the complex number to the next harmonic and the inner loop (n) is repeated again. If the complex number's frequency matches a frequency in the input data, then that particular $X[k]$ has a non-zero value.

E.g., if a 1.6 MHz sinusoid is sampled at 16 MHz and 10 samples are taken ($n=0$ to 9). Then the DFT checks for $N=10$ frequency matches from 0, 1.6MHz, 3.2 MHz up to $9 \cdot F_s/10 = 14.4$ MHz. In this case, an exact match would be found at $X[k=1]$ and the rest of the $X[k]$ values would be 0. A key *caveat* is the resolution of the frequency "match"- this is covered in Step 7.

In the complex DFT, $x[n] = i[n] \pm jq[n]$ ³. If only real data is available, the $q[n]$ values can be set to 0. The complex DFT still “works”- however, any match will have 1/2 the magnitude and an “unwanted” SSB at $-2 \cdot f$.

Note: $X[k=0]$ is the DC value and may be a very large spike that dwarfs the frequency matches. A DC value results when $x[n]$ itself has a DC offset. Or when a full period of the underlying sinusoids is not captured (remember a sinusoid only has 0 average area over a multiple of a full period- more on this in Step 7).

Key point: the DFT works by multiplying an unknown signal (with an unknown phase offset) with a series of “test” sinusoids (of varying frequency, which are all integer harmonics of each other). If there’s a frequency match, the result is a non-zero value, which indicates the magnitude. Multiplication is done with both cosine and sine in order to recover the full magnitude, and to avoid missing a frequency match entirely due to a 90° phase offset.

Step 6: Convolution and Sampling

Step 6 shows how convolution explains sampling effects.

In the time-domain, sampling an analog signal with an A/D is equivalent to multiplying the signal by an *infinite* series of impulses every $1/F_s$. In the frequency-domain, sampling has the *same* form as the time-domain as shown in **13(b)** and **13(e)**.

Figure 13(a-c) shows an example of sampling a *baseband* (centered at DC) signal. **13(d-f)** is an example of *undersampling* a *band-pass* (not centered at DC) signal. The results look very different but they’re both explained by convolution: as before, start by lifting the signal and shifting it over to the first impulse, then repeat in both \pm directions. (For the band-pass example, this is equivalent to duplicating each side-band at deltas of ± 20 MHz, which is the sampling frequency in **13(e)**.)

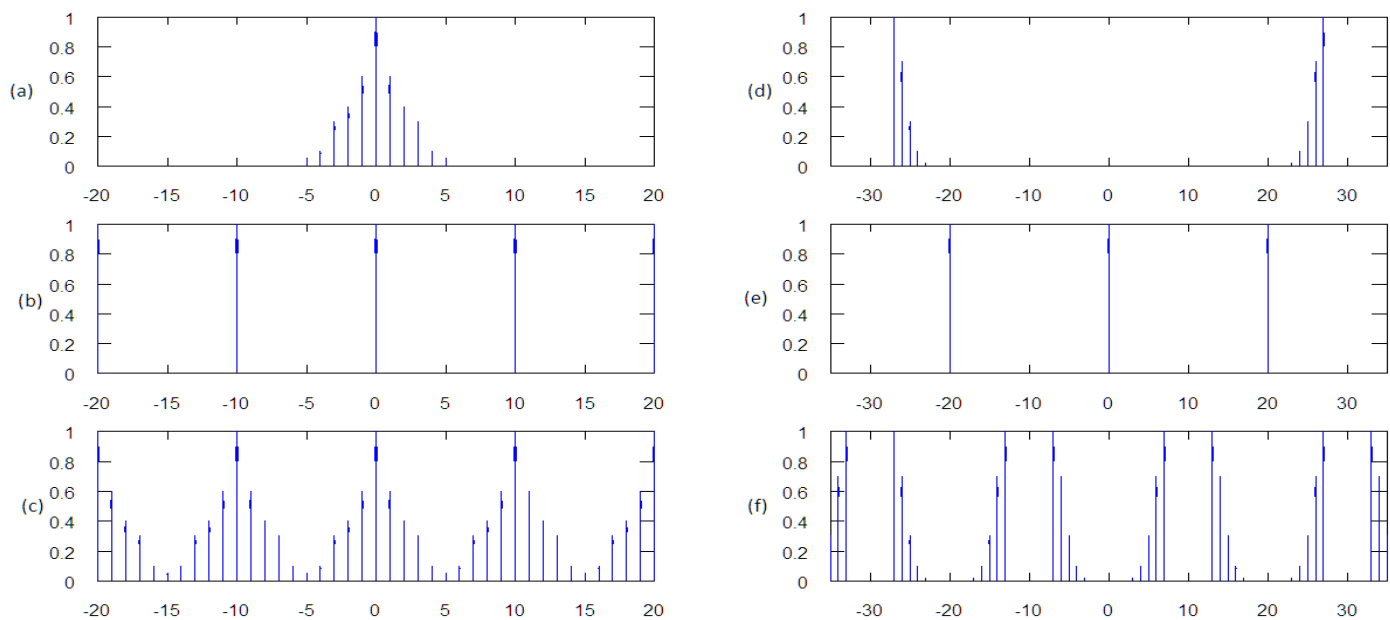


FIGURE-13: Sampling in frequency-domain: (a-c) baseband sampling with $F_s=10$ MHz; (d-f) bandpass undersampling with $F_s=20$ MHz

Up to now, we’ve been ignoring this property of sampled signals- but keep in mind, the DFT really applies only for sampled signals, which have an infinite number of spectral replicas @ $\pm F_s$ deltas in the frequency-domain.

³ **11-12** show a *positive* complex number ($I[f]+jQ[f]$) being shifted down in frequency to $f=0$. But the DFT also works with *negative* frequencies ($I[f]-jQ[f]$) because of *spectral replicas* due to sampling- see **14(d-f)** in Step 6.

You can see that as long as $F_s \geq 2 \cdot BW$ (bandwidth), the replicas don't overlap. This is the *Nyquist bandwidth* requirement. If $F_s < 2 \cdot BW$, then the spectral replicas will overlap- at that point, the original signal **can't** be recovered- this condition is called **aliasing**.

For a sinusoid, BW is the delta from 0 to the frequency. E.g., if a 6 MHz cosine is sampled @ $F_s = 10$ MHz, it'll be aliased to 4 MHz as shown in **Figure 14(a-c)**. (F_s must be $\geq 2 \cdot 6$ MHz = 12 MHz to avoid aliasing.)

Figure 14(d-f) shows a complex number at $f = -5$ MHz with *negative* frequency, which is also sampled at $F_s = 10$ MHz. In this case, the Nyquist bandwidth requirement is met, so there's no aliasing. Notice that a *positive* image at $f = +5$ MHz is created by sampling- this is why the DFT still works for negative frequencies.

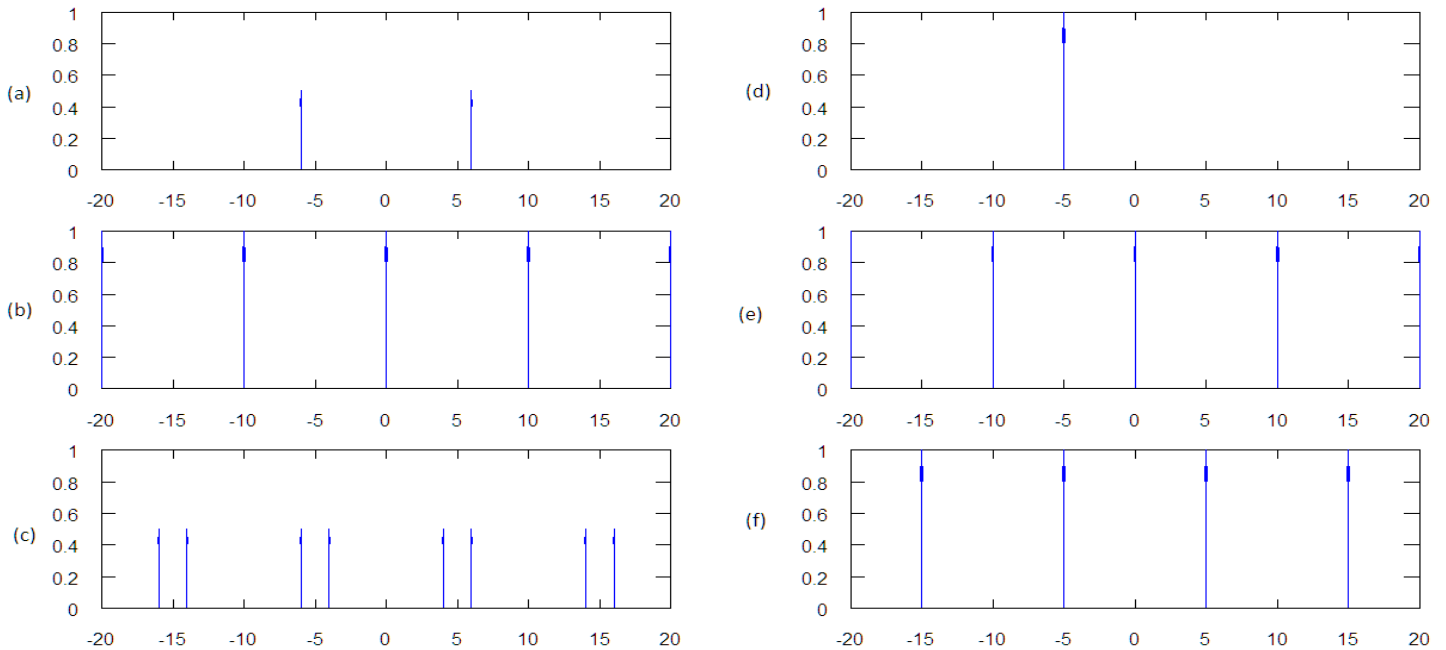


FIGURE-14: More frequency-domain sampling: (a-c) 6 MHz cosine sampled at 10MHz is aliased to 4 MHz; (d-f) SSB at $f = -5$ MHz is not aliased at 10 MHz

This Nyquist criterion can cause a lot of confusion because the "bandwidth" definition isn't consistent ($-f$ to $+f$ vs. 0 to $+f$, etc).

To add to the confusion, **13(d-f)** is an example of undersampling, which technically violates the Nyquist requirement since $F_s = 10$ MHz is $\ll 2f = 2 \cdot 27$ MHz. However, in terms of preventing replica overlap, the Nyquist requirement is really about the *relative* bandwidth of the signal of interest, which is only $\Delta f = 27 - 22$ MHz = 5 MHz in **13(d)**.

Step 7: Convolution and Windowing

The final step of this tutorial examines what happens when F_s (sampling clock) does **not** exactly capture a full period of the signal.

Capturing a complete period is only possible when F_s is a multiple of the signal's fundamental frequency ($F_s = n \cdot f$). In that case, the DFT results in a precise line because the signal appears continuous (truly periodic) to the DFT.

When this is not the case (which is the norm in the real world), then in the time-domain, it's equivalent to multiplying the signal by a *rectangular window*⁴ shown in **Figure 15(a)**. The Fourier transform of the rectangular function is the *Sinc* shape ($\text{sinc}(x)$)

⁴ The rectangular/Sinc function is the "default" case (no window). Other windows (Blackman, Hamming, etc.) are possible with various design trade-offs. This also applies to the LPF shown in **Figure 16**.

shown in **15(b)**. **15(f)** shows what happens when you convolve the Sinc shape with a Cosine function (**15(d)**)- now you get two broad spikes (“spectral leakage”) due to the time-domain discontinuity in **15(e)**.

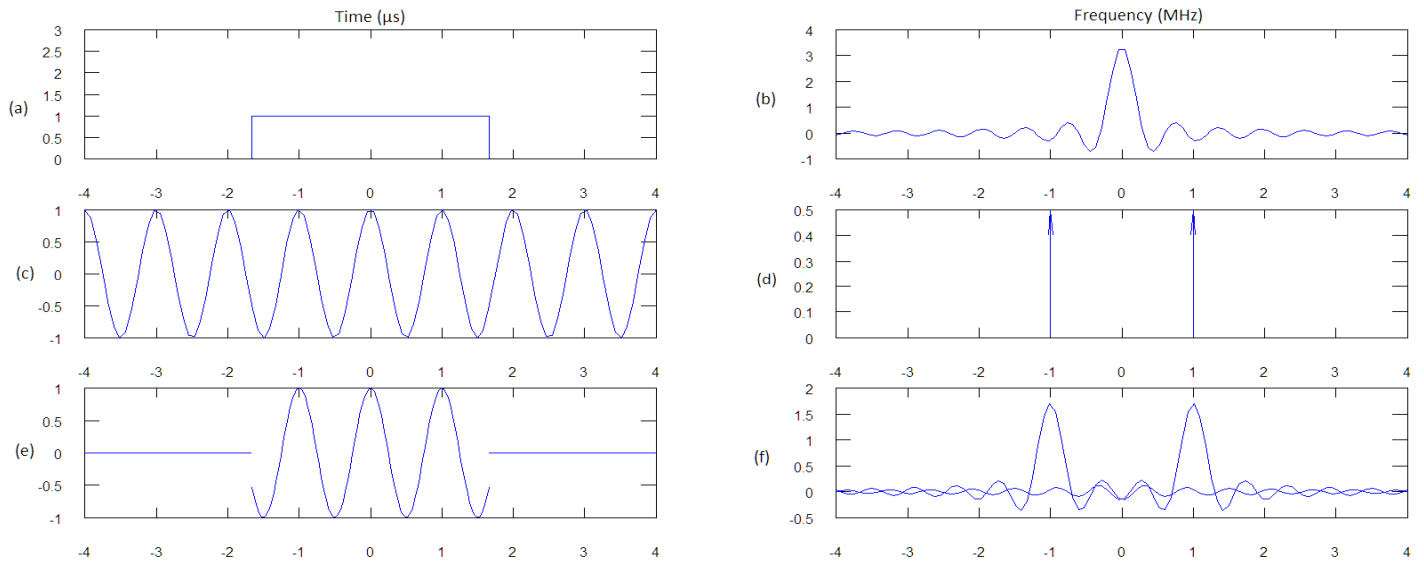


FIGURE-15: Rectangular windowing: (a-b)= rectangle in time/freq-domains; (c-d)= Cosine in time/freq-domains; (e-f)= effect of multiplication/convolution

But this is actually a good thing because it allows the DFT to detect an approximate frequency match (when F_s is unrelated to the signal) provided the Nyquist requirement is met.

We conclude this tutorial by looking at the rectangular function in the frequency-domain (**Figure 16(a-b)**)-which is an ideal Low pass Filter (LPF). In the frequency-domain, it's obvious how the LPF filter works (it's just multiplication of the rectangle with $X[f]$ and frequencies outside of the rectangle (pass region) are attenuated to 0). In the time-domain, an ideal LPF works by convolving Sinc with $x(t)$ - not so obvious even though both forms are equivalent.

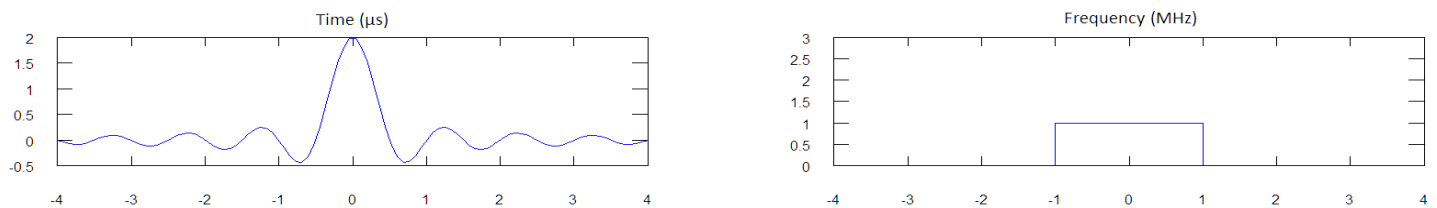


FIGURE-16: Ideal Low-pass Filter: (a)= LPF (Sinc) in time-domain; (b)= LPF (rectangle) in frequency-domain

References:

- 1) “Understanding Digital Signal Processing”, 2nd Edition” (2004), Richard Lyons.
- 2) “Quadrature Signals: Complex but not complicated”, Richard Lyons,
<http://www.dspguru.com/sites/dspguru/files/QuadSignals.pdf>
- 3) “The Scientist and Engineer’s Guide to Digital Signal Processing” (1997), Steven Smith.
<http://www.dspguide.com/>
- 4) “Signals and Systems made Ridiculously Simple” (1995), Zohar Karu.