ECGR 6114

Computer Project: Sampling, Aliasing,Quant. Error, DTFT

Student Name:___________

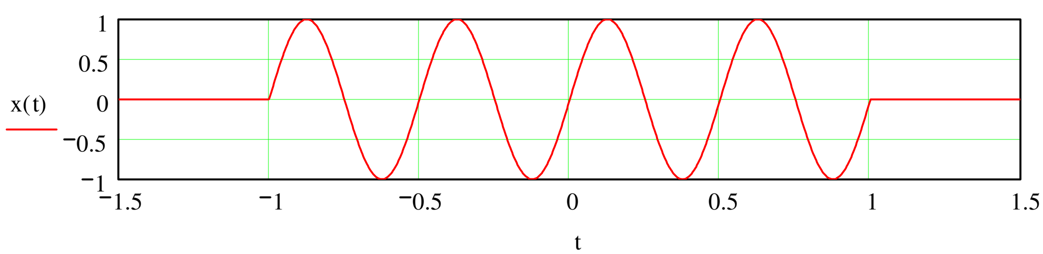
Part 1
Sampling

$\text{rect}(t) := \Phi(t + 0.5) - \Phi(t - 0.5)$ Define rect().

Consider a sinusoid with fequency fc Hz,
windowed by the rectangular gate function rect(t/τ).

$fc := 2$ $\omega c := 2\cdot\pi\cdot fc$ $\tau := 2$

$$x(t) := \sin(\omega c\, t)\cdot \text{rect}\left(\frac{t}{\tau}\right)$$



Consider the sinusoid sampled at a rate of fs samples/second,
over a time period tmax of twice the rect(t/τ) width, or 4 seconds.
By Nyquist, the signal should be sampled at greater than the highest signal frequency.

$fs := 8\cdot fc$. Sampling frequency fs samples/second.

$tmax := \tau$ $tmax = 2$ set the sampling window width relative to the length of the rect()

$Ts := \frac{1}{fs}$ $Ts = 0.063$ Sampling period Ts seconds

$\Omega s := 2\cdot\frac{\pi}{Ts}$ $\Omega s = 100.531$ Sampling frequency fs radians/second

$npts := 2^{\left(\text{floor}\left(\log\left(\text{floor}\left(\frac{tmax}{Ts}\right), 2\right)\right) + 1\right)}$ Set number of points to a power of two for the FFT.
(The equation is a bit cumbersome.)

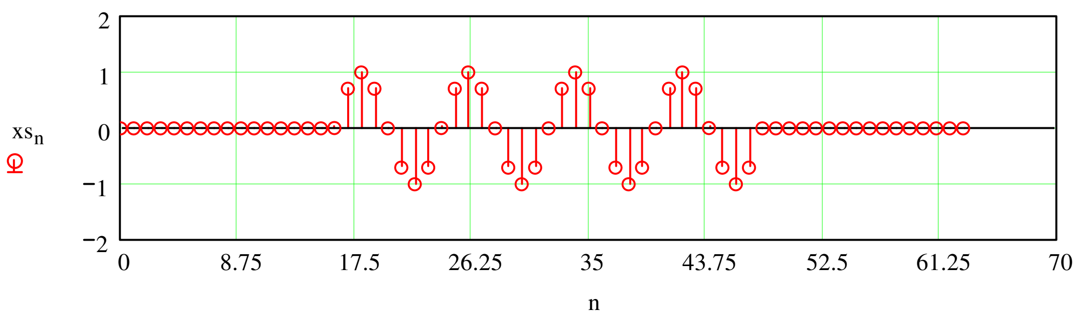
$npts = 64$ $npts\cdot Ts = 4$ This is actual width of time to be sampled

$n := 0..npts - 1$

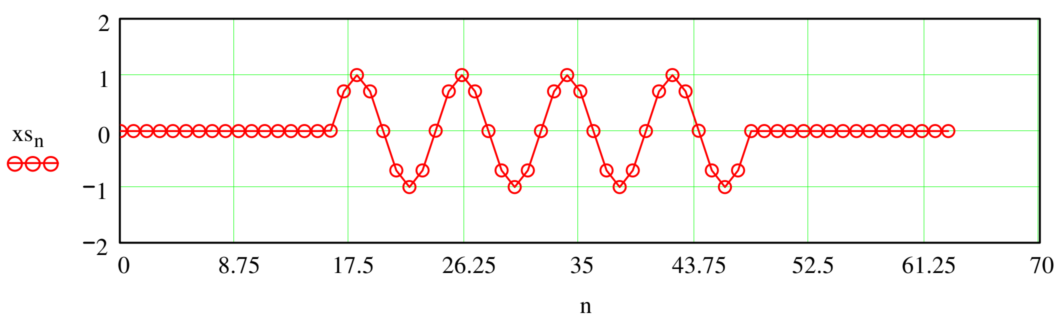
$k := 0..npts - 1$

$$xs_n := x\left(n \cdot Ts - npts \cdot \frac{Ts}{2}\right)$$

Finally, xs_k is our sampled signal.
It is a vector consisting of the value of x(t) at npts equally spaced time intervals, spaced Ts apart.



Or connecting the "dots"



Q1: Double the carrier frequency and replot the results.

Q2: When the carrier frequency doubled, did the sample rate change?

Q3: When the carrier frequency doubled, did number of samples change?

Q4: When the carrier frequency doubled, was the sample rate more than twice the highest frequency?

Part 2
Aliasing

If the signal is not sampled at a rate fs greater than twice the highest frequency coomponent in x(t), then aliasing occurs,
as illustrated below.

$fs := 1.5 \cdot fc$. Sampling frequency fs samples/second.

$Ts := \frac{1}{fs}$ $Ts = 0.333$ Sampling period Ts seconds

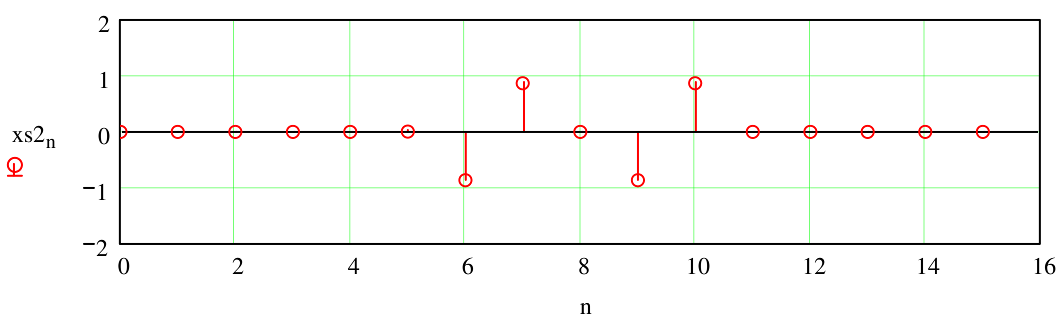
$npts := 2^{\left(floor\left(log\left(floor\left(\frac{2tmax}{Ts}\right), 2\right)\right)+1\right)}$ Set number of points to a power of two for the FFT. (The equation is a bit cumbersome.)

$npts = 16$ $npts \cdot Ts = 5.333$ This is actual width of time to be sampled

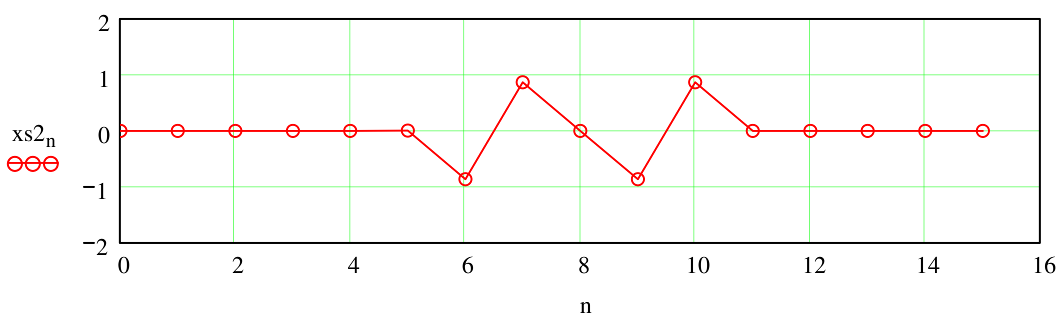
$n := 0 .. npts - 1$ $k := 0 .. npts - 1$

$xs2_n := x\left(n \cdot Ts - npts \cdot \frac{Ts}{2}\right)$

Finally, xs_k is our sampled signal.
It is a vector consisting of the value of x(t) at npts equally spaced time intervals, spaced Ts apart.



Redrawing, except connecting the "dots," it is clear to see that the improperly sampled signal now has severe aliasing and looks like a signal at half the original frequency!



Q5: Change the sampling rate to 1.75 times the carrier frequency, and replot above.

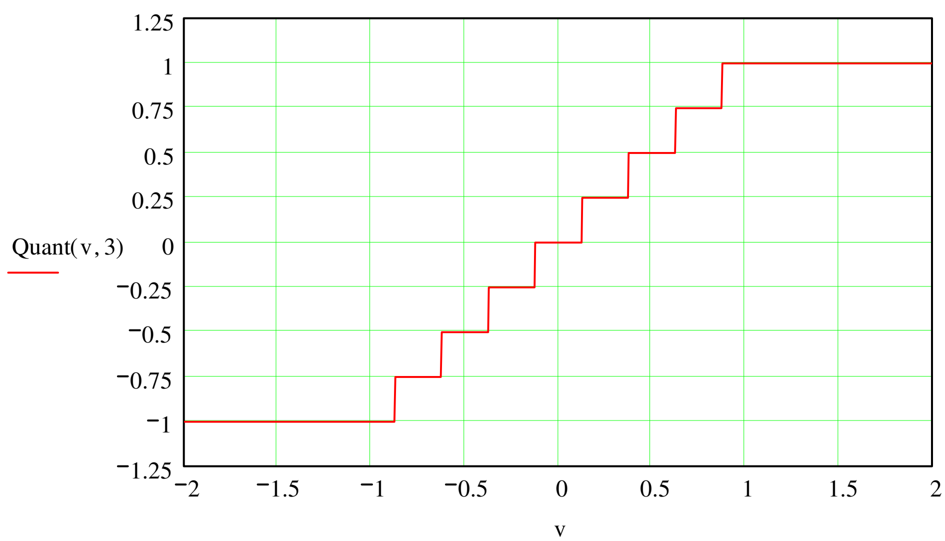
Part 3
Quantization Function

When signals ae digitized using an analog-to-digital converter (ADC), the signals are quantized in addition to being sampled in time.
The effect of quantization is the same as adding a random error to a signal.

In the case of an 8-bit ADC, the signals are quantized into 256 discrete levels, ranging from the min to max of the ADC voltage range. For an ideal ADC, the ADC selects the closest level to the analog voltage that is being digitized.

To implement quantization in matlab, the following function is used.
The example for a 3 bit dac, nbits=3, is plotted.

$$\text{Quant}(v, \text{nbits}) := \text{if}\left(v > 1, 1, \text{if}\left(v < -1, -1, \frac{\text{round}\left(2^{\text{nbits}-1} \cdot v\right)}{2^{\text{nbits}-1}}\right)\right)$$



Q6: Change the quantizer plot to 4 bits.

Q7: What is Δv the voltage step size of the 4-bit quantizer?

Part 4
Quantization Error

Returning to the sampled sinusoid,
let us now sample again but without the rect() gate, $fc := 2$
with quantization and a bit higher sample rate.

$fc := 2 \qquad \omega c := 2 \cdot \pi \cdot fc \qquad \tau := 4$

$y(t) := \sin(\omega c\, t) \qquad tmax := \frac{\tau}{2}$

$fs := 10.1 \cdot fc$. Sampling frequency fs samples/second.

$Ts := \frac{1}{fs} \qquad Ts = 0.05$ Sampling period Ts seconds

$$npts := 2^{\left(\operatorname{floor}\left(\log\left(\operatorname{floor}\left(\frac{2tmax}{Ts}\right),2\right)\right)+1\right)}$$

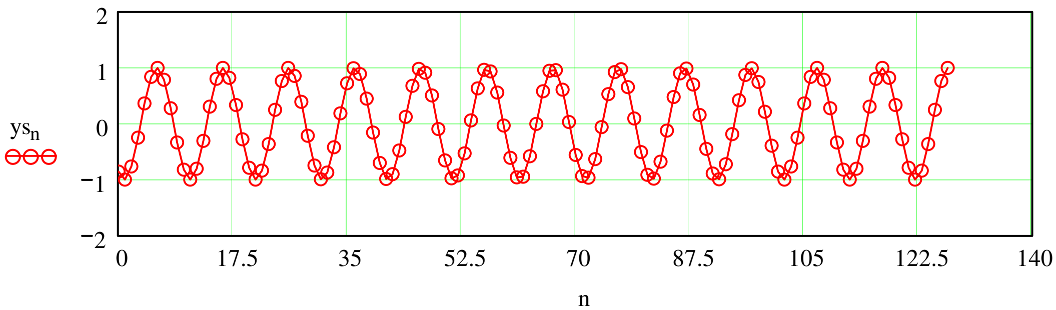
Set number of points to a power of two for the FFT.
(The equation is a bit cumbersome.)

$npts = 128 \qquad npts \cdot Ts = 6.337$ This is actual width of time to be sampled

$n := 0 .. npts - 1 \qquad k := 0 .. npts - 1$

$$ys_n := y\left(n \cdot Ts - npts \cdot \frac{Ts}{2}\right)$$

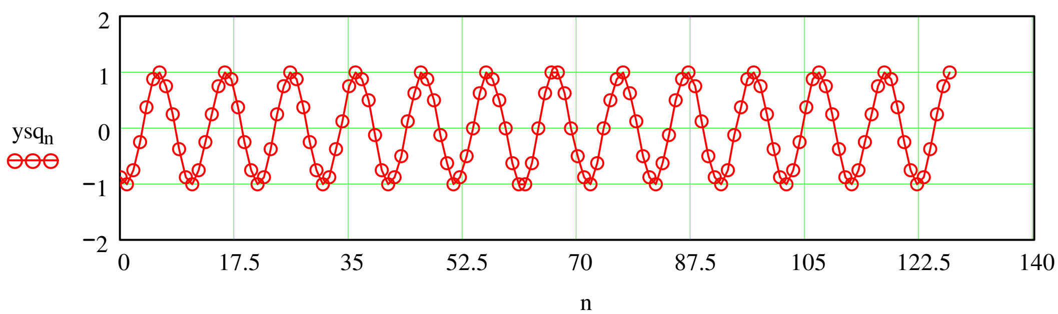
Finally, xs_k is our sampled signal.
It is a vector consisting of the value of x(t) at
npts equally spaced time intervals, spaced Ts apart.



The quantized signal is then

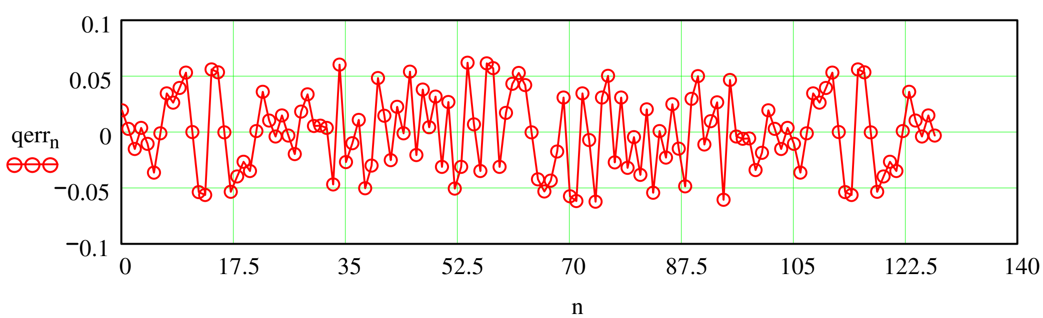
$$QuantBits := 4$$

$$ysq_n := Quant(ys_n, QuantBits)$$



The quantized error is then

$$qerr_n := ys_n - ysq_n$$



$$Qerr\_Second\_Moment := \frac{qerr \cdot qerr}{npts}$$

Use average of dot product of qerr vector to compute E[x^2].

$$Qerr\_Second\_Moment = 1.17 \times 10^{-3}$$

$$Qerr\_mean := \sum_{p=0}^{rows(qerr)-1} qerr_p \qquad Qerr\_mean = 0.0195$$

Mean should be zero, but will not exactly be zero in experiments.

$$Stepsize := \frac{2}{2^{QuantBits}} \qquad Stepsize = 0.125$$

Quantization step size

$$Theoretical\_Second\_Moment := \frac{Stepsize^2}{12} \qquad Theoretical\_Second\_Moment = 1.30208 \times 10^{-3}$$

Q8: What is the mean of the error voltage of the 4-bit quantizer?

Q9: What is the second moment of the error voltage of the 4-bit quantizer?

Q10: What is the standard deviation of the error voltage of the 4-bit quantizer?
(Add a new Mathcad formula below to compute standard deviation and show result)

Q11: What is the rms voltage of the error voltage of the 4-bit quantizer?
(Add a new Mathcad formula below to compute standard deviation and show result)

Part 5

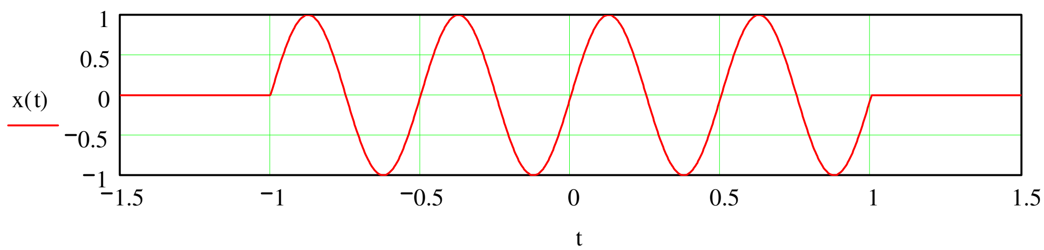
Plotting Fourier Transforms

Use explicit fourier transform definition, rather than symbolic, since numerical results are desired.
Note: you may wish **NOT** to use infinite limits, to avoid numerical problems.

Compute the fourier transform of the original signal, x(t)

$fc := 2 \qquad \omega c := 2\cdot\pi\cdot fc \qquad \tau := 2$

$$x(t) := \sin(\omega c\, t)\cdot \mathrm{rect}\left(\frac{t}{\tau}\right)$$



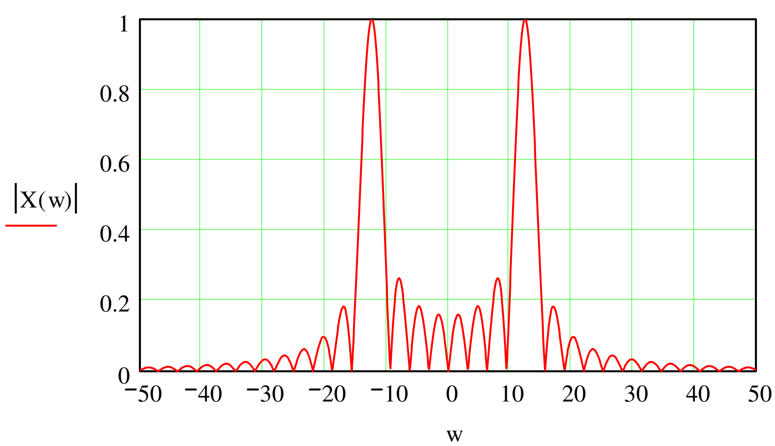
Sometimes the sine/cosine form at right converges when the exponetial form of Fourier transform does not.

$$\int_{-\infty}^{\infty} x(t)\cdot(\cos(\omega\cdot t))\,dt - i\cdot\int_{-\infty}^{\infty} x(t)\cdot(\sin(\omega\cdot t))\,dt$$

Also note that in some versions of Mathcad, the exponential fom below does not give correct results if the limits are infinity, or even a number much larger than 2. The sinusoudal form above seems a bit more robust, and will work with infinite limits.

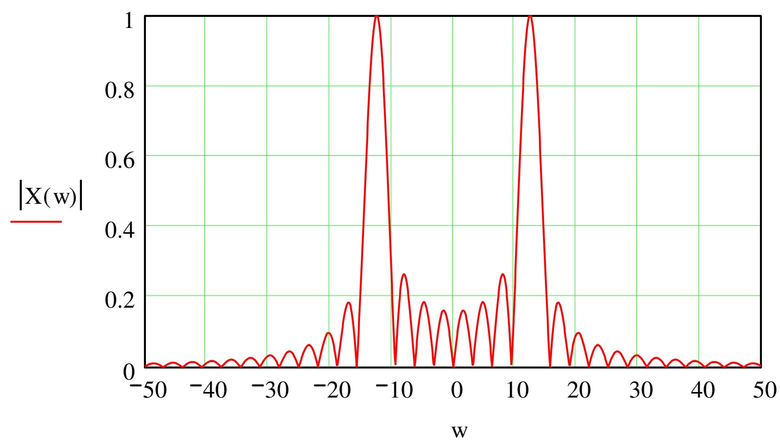
$$X(\omega) := \int_{-2}^{2} x(t)\cdot\left(e^{-i\cdot\omega\cdot t}\right)dt$$

$|X(1)| = 0.135$



Here is the fourier transform in the sinusoidal form with infinite limits (should give same result).

$$X(\omega) := \int_{-\infty}^{\infty} x(t)\cdot(\cos(\omega\cdot t))\,dt - i\cdot\int_{-\infty}^{\infty} x(t)\cdot(\sin(\omega\cdot t))\,dt \qquad |X(1)| = 0.135$$



Q12: Double the carrier frequency fc, and replot the fourier transform above.

Part 6

Discrete Time Fourier Transforms (DTFT)

The DTFT (discrete time Fourier transform) is defined for the discrete time sequence x[n] as X(ω) given below

$$X(\omega) := \sum_{n=-\infty}^{\infty} x(n)\cdot e^{-i\cdot\omega\cdot n}$$

Recall that the DTFT is periodic, with period = 2π .

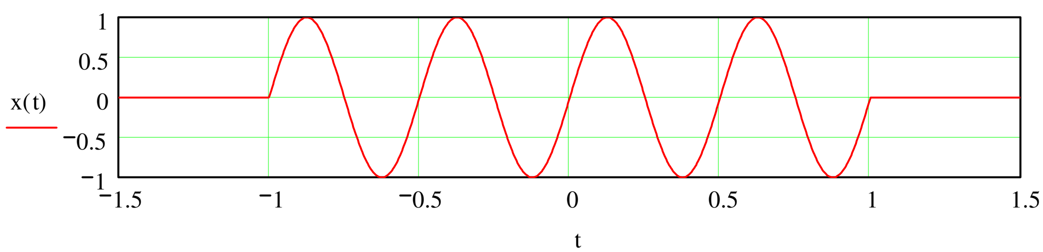
For finite length nonzero sequence, the limits become finite.
If x(n) is a vector in Mathcad holding the samples, the equation is rewritten as:

$$X(\omega) := \sum_{n=0}^{rows(x)-1} x_n\cdot e^{-i\cdot\omega\cdot n}$$

Returning to the example signal x(t)

$fc := 2$ $\omega c := 2\cdot\pi\cdot fc$ $\tau := 2$

$$x(t) := \sin(\omega c\, t)\cdot rect\left(\frac{t}{\tau}\right)$$



Sample the signal again

$fs := 8\cdot fc$. Sampling frequency fs samples/second.

$tmax := \tau$ $tmax = 2$ set the sampling window width relative to the length of the rect()

$Ts := \frac{1}{fs}$ $Ts = 0.063$ Sampling period Ts seconds

$\Omega s := 2\cdot\frac{\pi}{Ts}$ $\Omega s = 100.531$ Sampling frequency fs radians/second

$$npts := 2^{\left(floor\left(\log\left(floor\left(\frac{tmax}{Ts}\right),2\right)\right)+1\right)}$$

Set number of points to a power of two for the FFT. (The equation is a bit cumbersome.)

$npts = 64$

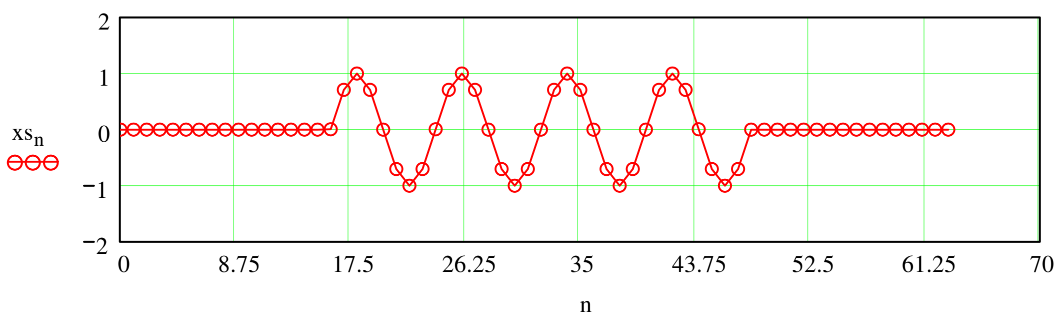
$npts \cdot Ts = 4$

This is actual width of time to be sampled

$n := 0 .. npts - 1$ $\quad$ $k := 0 .. npts - 1$

$$xs_n := x\left(n \cdot Ts - npts \cdot \frac{Ts}{2}\right)$$

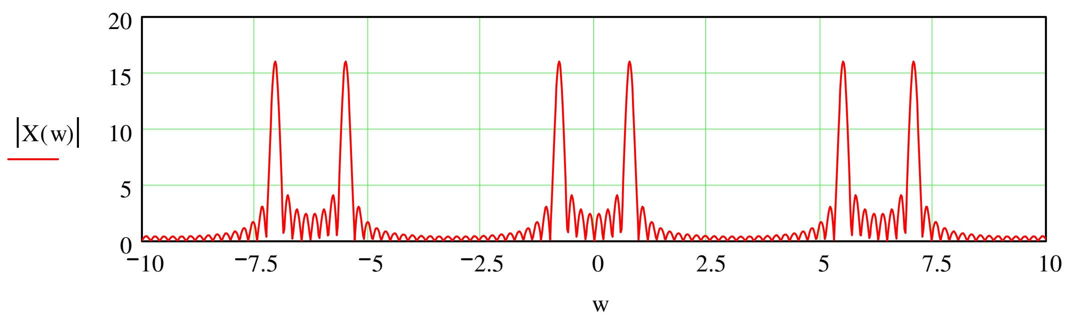
Finally, xs_k is our sampled signal.
It is a vector consisting of the value of x(t) at npts equally spaced time intervals, spaced Ts apart.



Now take the DTFT of the sampled signal (note that xs(n) is used)

$$X(\omega) := \sum_{n=0}^{rows(xs)-1} xs_n \cdot e^{-i \cdot \omega \cdot n}$$

$X(0) = 0$ $\quad$ $xs_1 = 0$



Recall that the DTFT equals 1/Ts times the Fourier transform, but is periodic with period 2π.

$$\frac{1}{Ts} = 16$$

Q13: Double the carrier frequency fc, and replot the fourier transform above.)

Part 7
Inverse DTFT

Most commonly, inverse DTFT of closed form type is done by a look-up table, just as for Laplace transform.

The following method is a numerical implementstion of the inverse z-transform integral

$$IDTFT(f,n) := \left(\frac{1}{2\cdot\pi}\right)\cdot\int_{-\pi}^{\pi} f\left(e^{i\omega}\right)\cdot e^{i\cdot\omega\cdot n}\,d\omega$$

$$IDTFT(n) := \left(\frac{1}{2\cdot\pi}\right)\cdot\int_{-\pi}^{\pi} X(\omega)\cdot e^{i\cdot\omega\cdot n}\,d\omega$$

Which may be approximated as

$nw := 50$ $\qquad dw := \frac{\pi}{nw}$ $\qquad nw\cdot dw = 3.142$

$$IDTFT2(n) := \left(\frac{1}{2\cdot\pi}\right)\sum_{p=-nw}^{nw-1} X\left(p\cdot dw + \frac{dw}{2}\right)\cdot e^{i\cdot\left(p\cdot dw+\frac{dw}{2}\right)\cdot n}\cdot dw$$

and checking out the inverse transfom against the sampled signal xs[n].

$IDTFT(35) = 0.707$ $\qquad xs_{35} = 0.707$ $\qquad IDTFT2(35) = 0.707$

$IDTFT(36) = 1.998\times 10^{-15}$ $\qquad xs_{36} = 0$

$IDTFT(37) = -0.707$ $\qquad xs_{37} = -0.707$

$IDTFT(38) = -1$ $\qquad xs_{38} = -1$

Q14: Do the inverse DTFT values equal the original values of x[n]?

Part 8
Discrete-time Convolution
(Linear Convolution, not Circular Convolution)

Suppose a system has length 3 impulse response,h[n]

$$h := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Convolution for a discrete time sequence y[n] = x[n] * h[n] = h[n] * x[n]is given below

$$y(n) := \sum_{p=-\infty}^{\infty} x(p)\cdot h(n-p)$$ or $$y(n) := \sum_{p=-\infty}^{\infty} h(p)\cdot x(n-p)$$

For finite length causal sequence h, and the limits become finite.
:

$$y(n) := \sum_{p=0}^{\text{rows}(h)-1} h_p \cdot x_{n-p}$$

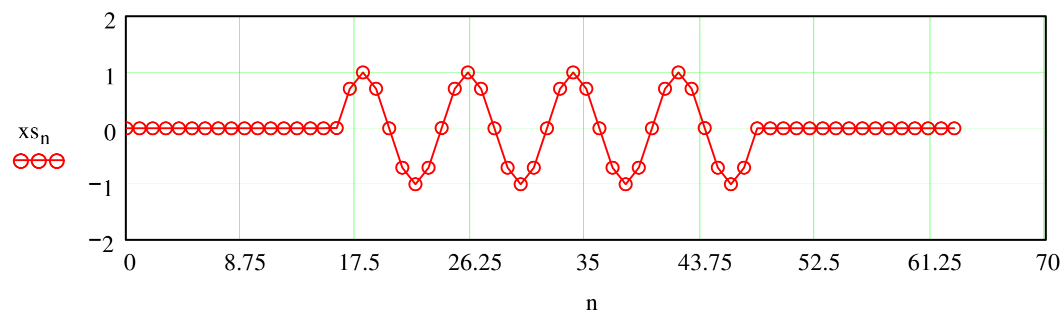
or even better written below, for causal x[n] without samples for n<0

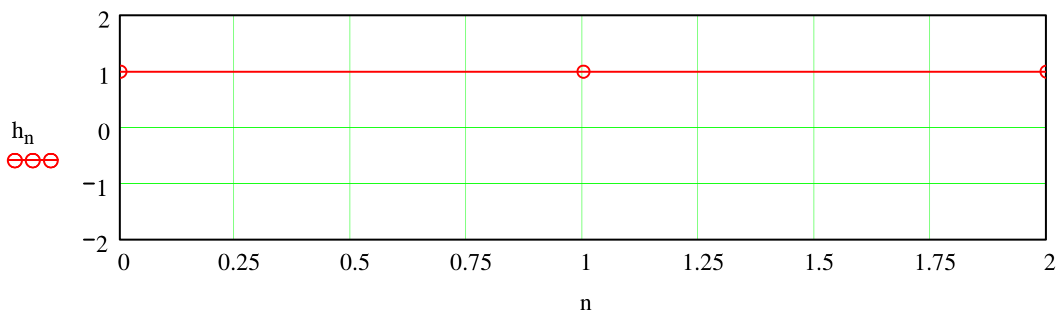
$$y_n := \sum_{p=0}^{\text{rows}(h)-1} \text{if}\left(n-p>-1, h_p\cdot xs_{n-p}, 0\right)$$

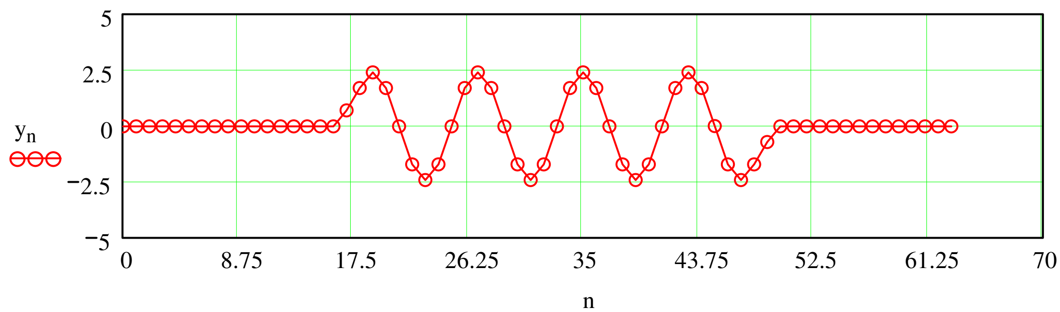
for the above h, and for the earlier sampled signal xs[n]

$$h = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$y_n := \sum_{p=0}^{rows(h)-1} if\left(n - p > -1, h_p \cdot xs_{n-p}, 0\right)$$







Q15: Increase the width of h[n] from {1,1,1} to {1,1,...1,1} such that the numver of 1's in the vector h equals the number of samples in 1 period of the sinusoid.
zreplot y[n] above for this new h vector. The sinusoid should disappear, except at endpoints. This is effectively the convolution of a rectangular box, h[n], with the original windowed sinusoid.