```
% SpecAnalysisV21.m
% Version 2.1, (9/7/09) jra
% J.R.Andrews, KH6HTV, Picosecond Pulse Labs, Boulder, Colorado USA
% V1.0, 22 nov 2002 - original program
% V1.1 2/27/04 & 4/20/04 revs. for plotting & data output, used in AN-16
% V2.0, 24 Aug. 2004, jra -- major modification of step-like waveform
% 8/25/04 added choice of data in/out from either A: or C: drive
% analysis.  Incorporated Riad's composite FFT of Nicholson ramp
% subtraction along with Gan's square wave.
% Also corrected scaling error in V1. sq.wave spectrum.
% V2.1, 7 Sept 2009, jra -- minor mod. -- added semilog plots
% note: This works with floppy disc data files written by the HP-54750
% oscilloscope.  Verbose data headers need to be first stripped off. The HP
% o'scope data is organized as one voltage data point per line.
%
% Technical References:
% [1] J.R.Andrews & M.G.Arthur, "SPECTRUM AMPLITUDE --- Definition,
% Generation and Measurement", NBS Tech.Note 699, NBS, Boulder, Colo.USA,
% Oct. 1977, 92 pages.  see in particular pp. 22-24
% [2] W.L.Gans & J.R.Andrews, "Time Domain Automatic Network Analyzer for
% Measurement of RF & Microwave Components", NBS Tech.Note 672, NBS,
% Boulder, Colo.USA, Sept.1975, 165 pages.  see chp. 3
% [3] A.Shaarawi & S.Riad, "Computing the Complete FFT of a Step-Like
% Waveform", IEEE Trans. Inst.& Meas., vol IM-35, no.1, pp.91-92,March 1986
% [4] A.M.Nicolson, "Forming the fast Fourier transform of a step response
% in time-domain metrology", Electron.Lett., vol.9,pp.317-318, July 1973
% [5] W.L.Gans & N.S.Nahman, "Continuous and discrete Fourier tranform of
% step-like waveforms", IEEE Trans. Inst. & Meas., vol.IM-31, pp.97-101,
% June 1982
% [6] J.Waldmeyer, "Fast Fourier transform for step-like functions: The
% synthesis of three apparently different methods", IEEE Trans. Inst. &
% Meas., vol, IM-29, pp.36-39, March 1980
%
disp('  ')
disp('SpecAnalysisV21.m --- MatLab Program, V2.1, 7 Sept. 2009')
disp('J.R. Andrews, Picosecond Pulse Labs')
disp('SPECTRUM ANALYZER of Periodic or Transient Signals')
disp('the input comes as .txt file from an oscilloscope')
disp('FFT output ploted in dBm or dBuV/MHz vs. Frequency')
clear
disp(' ')
drive = input('Which drive is used for data in/out? (1=A: 3=C:)   ');
if drive == 1
    disp('Data in/out will be as .txt files via floppy disc in drive A:')
end
if drive == 3
```

```
    disp('Data in/out will be as .txt files in current C: drive directory')
end
disp('Signal Type?')
disp('1 = PERIODIC, time window must include 1 or more cycles')
disp('2 = IMPULSE, time window must include entire transient')
disp('3 = STEP, time window must include flat top of step')
type = input('Enter Signal Type -- 1,2 or 3    ');
fname = input('Enter Data File Name:   ','s');
if drive == 1
    dname = ['A:', fname, '.txt'];
end
if drive == 3
    dname = [fname, '.txt'];
end
v = load(dname);
Tw = input('Time Window in NanoSeconds (ns)?  ');
N = length(v);  %(# of data points)
f0=1/Tw; %(freq. resolution in GHz)
dt=Tw/N; %(sample spacing in ns)
for i=1:N
    t(i)=(i-1)*dt;
end
plot(t,v)
grid
xlabel('time in ns')
ylabel('Volts')
title('Input from Oscilloscope')
disp('plot displayed - press any key to continue')
pause
% CALCULATE  -- FFT of v,  note: not valid for type 3 (step)
% note: for proper scaling of periodic waveforms, need to divide FFT by N
% to get correct V(f) in Volts, i.e. for type 1 (periodic)
% For transient, non-periodic signals, need to convert from periodic Volts
% to spectrum amplitude in Volt-Picoseconds.  Correct scaling is required
% To do this multiply the periodic FFT array, V, by 2 x the time window,
% Tw.  For Tw in ns, to convert to ps, also need to multiply by 1000x
%
if type ~= 3  % i.e. do following for types 1 (periodic) or 2 (impulse)
    V = (1/N)*fft(v);   % units are Volts
% for plotting purposes, throw away dc, nyquest & neg. freqs (i.e. >N/2)
    for i=1:(N/2 - 1)
        Vrms(i) = sqrt(2)*abs(V(i+1));  % units are Volts (rms)
        SA(i) = (2*Tw*1000)*abs(V(i+1));  % units are V-ps
        f(i) = i*f0;
    end
    P = 0.02*(Vrms .^2); % units are watts
```

```
    Pdbm = 10*log10(P ./ 0.001); % units are dBm
    SAdb = 20*log10(SA);  % units are dBuV/MHz
end
if type == 3  % i.e. step-like waveform -- special processing required
%   Create new waveform, vsr, by subtracting Nicolson [4] linear ramp from
%   step-like pulse waveform.
for i=1:N
    vramp(i) = v(1) + (i-1)*(v(N)-v(1))/(N-1);
    vsr(i) = v(i) - vramp(i);
end
%  Convert step pulse of N pts. to a Gans' [5] square wave, vsq, of 2N pts.
vsq = v;
for i= 1:N
    vsq(i+N) = (v(1)+v(N)) - v(i);
end
Nsq=2*N;
Twsq=2*Tw;
f0sq=1/Twsq; %(in GHz)
for i=1:Nsq
    tsq(i)=i*dt;
end
%   calculate the FFTs
% note: for proper scaling of periodic waveforms, need to divide FFT by N
% to get correct V(f) in units of Volts
Vsr = fft(vsr);
Vsr = Vsr/N;
Vsq = fft(vsq);
Vsq = Vsq/Nsq;
% note: f0 of ramped waveform is 1/Tw.  All harmonics of f0 are present up
% to Nyquist freq, fny = 1/2*dt.  The dc value is not valid.
% For sq.wave waveform the fundamental is 1/2Tw and only the odd harmonics
% are present up to the Nyquist freq, fny.  The even harmonics of 1/2Tw are
% zero due to symmetry.  The dc value is valid.  The frequency lines of the
% ramped waveform and the square wave are interleaved and are all valid.
%
% Mesh Vsr & Vsq arrays into single array V. See Shaarawi & Riad [3]
V = Vsq;
for i=3:2:Nsq-1
    V(i) = Vsr((i+1)/2);  % i.e. insert ramp spec. into nulls of sq.spec.
end
% for transient, non-periodic signals, need to convert from periodic Volts
% to spectrum amplitude in Volt-Picoseconds.  Correct scaling is required
% To do this multiply the periodic FFT array, V, by 2 x the time window,
% Tw, of the step, NOT the sq.wave.  For Tw in ns, to convert to ps, also
% need to multiply by 1000x
%
```

```
% for Spectrum Amplitude plotting purposes only, throw away dc, Nyquist
% freq. & neg freqs
for i=1:(Nsq/2)-1
    SA(i) = (2*Tw*1000)*abs(V(i+1));  % units are Volts-picoseconds
    SAdb(i) = 20*log10(SA(i));  % units are dBuV/MHz
    f(i) = i*f0sq;
end
end % end of step-like waveform processing
% plot results
if type == 1
    dbdisc = Pdbm;
    plot(f,Pdbm)
    ylabel('Power in dBm')
    grid
    xlabel('Frequency in GHz')
    title('Calculated FFT of Signal')
    pause
    semilogx(f,Pdbm)
    ylabel('Power in dBm')
    grid
    xlabel('Frequency in GHz')
    title('Calculated FFT of Signal')
    pause
end
if type > 1
    SAdb = 20*log10(SA); % dBuV/MHz or dB(V-ps)
    dbdisc = SAdb;
    plot(f,SAdb)
    ylabel('Spectrum Amplitude in dBuV/MHz')
    grid
    xlabel('Frequency in GHz')
    title('Calculated FFT of Signal')
    pause
    semilogx(f,SAdb)
    ylabel('Spectrum Amplitude in dBuV/MHz')
    grid
    xlabel('Frequency in GHz')
    title('Calculated FFT of Signal')
    pause
end

% output results to floppy disc in drive A:
dataout = input('Do you want to save results to disc? (1=yes, 0=no)  ');
if dataout == 1
    fname = input('Enter Output Data File Prefix Name:   ','s');
% '\r' or '\n' is delimiter for carriage return (newline), i.e. 'enter' key
```

```
% this writes output file with one data point per line
% note: this doesn't appear correct in NotePad,
% but is correct in WordPad or Word
if drive == 1 % i.e. drive A:
    disp('writing output.txt files to drive A:')
    disp('writing complex V(f), i.e fft(v), as Vfft.txt')
    dname = ['A:', fname, 'Vfft.txt'];
    dlmwrite(dname,V,'\r');
    disp('writing plot frequency file as freq.txt')
    dname = ['A:', fname, 'freq.txt'];
    dlmwrite(dname,f,'\r');
    if type == 1 % i.e. periodic
        disp('writing Power in dbm file as Pdbm.txt')
        dname = ['A:', fname, 'Pdbm.txt'];
        dlmwrite(dname,Pdbm,'\r');
    end
    if type ~= 1 % i.e. transient, impulse or step
        disp('writing Spectrum Amp. in dBuV/MHz as SAdb.txt')
        dname = ['A:', fname, 'SAdb.txt'];
        dlmwrite(dname,SAdb,'\r');
    end
end
if drive == 3 % i.e. drive C:
    disp('writing output.txt files to drive C: in current directory')
    disp('writing complex V(f), i.e fft(v), as Vfft.txt')
    dname = [fname, 'Vfft.txt'];
    dlmwrite(dname,V,'\r');
    disp('writing plot frequency file as freq.txt')
    dname = [fname, 'freq.txt'];
    dlmwrite(dname,f,'\r');
    if type == 1 % i.e. periodic
        disp('writing Power in dbm file as Pdbm.txt')
        dname = [fname, 'Pdbm.txt'];
        dlmwrite(dname,Pdbm,'\r');
    end
    if type ~= 1 % i.e. transient, impulse or step
        disp('writing Spectrum Amp. in dBuV/MHz as SAdb.txt')
        dname = [fname, 'SAdb.txt'];
        dlmwrite(dname,SAdb,'\r');
    end
end
end
disp('end of SpecAnalysisV21.m program')
```