# Experiments in Gaussian White-noise Generation

**By:** Zsolt Papay
Tech University of Budapest (TUB)
Dept of Telecommunications
Budapest, Hungary

## Purpose:

To study numerical signal synthesis and analysis in practice.

## Description:

Linking software files between Agilent's BenchLink and Mathcad[1] can result in a very powerful analysis/synthesis technique. This experiment will examine the use of Mathcad to create band-limited white noise, which is then loaded into the arbitrary waveform generator via Agilent BenchLink software.

## Equipment:

- Agilent 33120A Function/ARB generator
- Agilent 54600-series oscilloscope with FFT (and GPIB) option
- Personal computer with GPIB card & cables
- Agilent 34820A BenchLink/Suite Software (Consists of Agilent 34810B BenchLink/Scope + Agilent 34811A BenchLink/Arb +Agilent 34812A BenchLink/Meter)
- Mathcad ® (MathSoft Inc.) Software

## Procedure (see Fig 1):

It is possible to analyze the performance of a DUT (Device Under Test) by examining its response to well-designed test signals. In this case, we will generate a waveform with a signal buried in noise. This is a frequent occurrence in real life, and it makes for an interesting study to see how an FFT can extract the signal from the combined set of (signal+noise). Digital synthesis and analysis techniques are the powerful tools that make it possible (**Fig. 1**):

**1**) **CREATE** a waveform numerically by using Mathcad[1] software.

**2**) Transfer that waveform in a file to the **Agilent 34811A** BenchLink/Arb Software

**3)** Download the waveform to the **Agilent 33120A** Function/ARBitrary Waveform Generator and **GENERATE** the signal.

**4**) **MEASURE** the system response using the **Agilent 54602B** Digitizing Storage Oscilloscope with a plug-on **Agilent 54657A (FFT/GPIB)** Measurement/Storage Module, and capture the screen data with **Agilent 34810B** BenchLink/Scope software.

[1] [Note: Mathcad is a trademark of Mathsoft Corporation.]

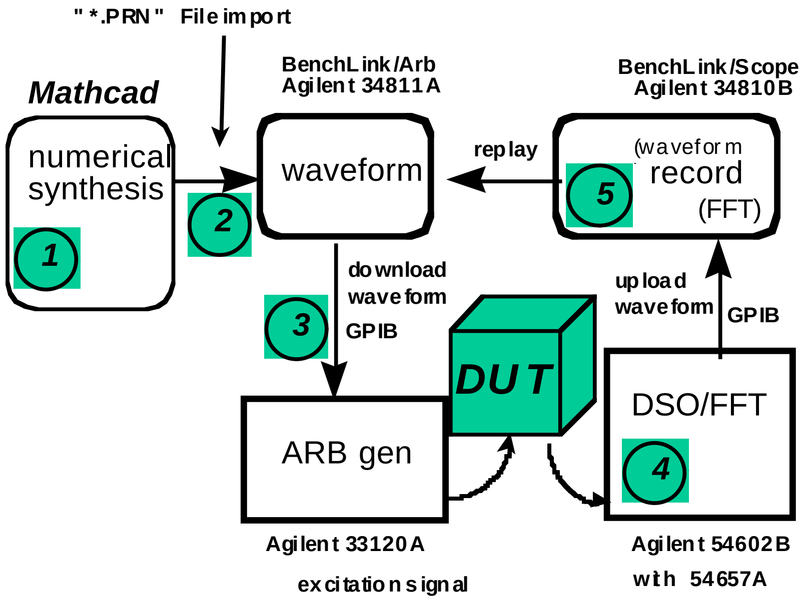



**Fig. 1: Setup of experiment**

**Experiment:**

**Part I: Synthesize the Gaussian noise with Mathcad:**

Synthesize nearly **Gaussian noise** with **flat** (band-limited white) **spectrum** by means of phase spectrum randomizing **in the frequency domain.** **(***Reference:* K. Kafadar, "Gaussian White-noise Generation for Digital Signal Synthesis" IEEE Trans on Instr and Meas, Vol. IM-35, No. 4, Dec. 1986, 492-495.**)** Then transform into **time domain** with the inverse FFT. (See Fig 2)

Gaussian White-noise synthesized in frequency domain, and transformed into time domain *(Mathcad)*

$p := 13$ $n := 0..2^{10}$ $z_n := 0$ $m := 0..2^{11} - 1$ $w_m := \frac{1}{2}$

$k := 0..2^{p-1}$ $\beta := \mathrm{stack}(z, w)$

constant (bandlimited) magnitude vector: $\alpha := \mathrm{stack}(\beta, z)$

random (uniform noise) phase vector: $\phi_k := \mathrm{rnd}(2 \cdot \pi)$ $j := \sqrt{-1}$

FOURIER components (in frequency domain): $c_k := \alpha_k \cdot e^{j \cdot \phi_k}$

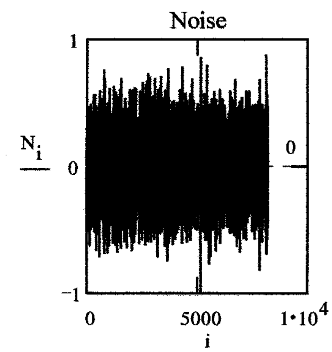
inverz FFT: $x := \mathrm{IFFT}(c)$ $u := \mathrm{last}(x)$ $i := 0..u$ $P := u + 1$ $P = 8192$

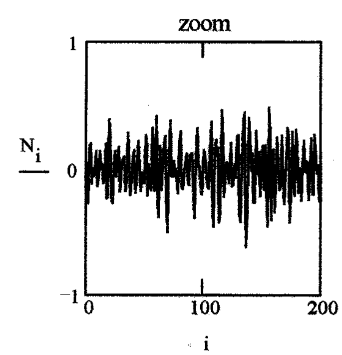
$HI := \max(x)$ $LO := \min(x)$ $\mathrm{stdev}(x) = 32$

"noise" (in time domain): $norm := \mathrm{if}(HI > |LO|, HI, |LO|)$

$N_i := \frac{x_i}{norm}$ $hi := \max(N)$ $low := \min(N)$ $avg := \mathrm{mean}(N)$ $\sigma := \mathrm{stdev}(N)$

$hi = 0.9$ $low = -1$ $avg = 0$ $\sigma = 0.2$ (rms noise level)

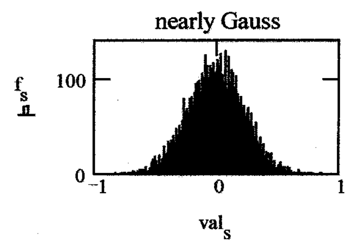



amplitude histogram:

$V := 256$ $h := 0..V$ $val_h := -1 + \frac{2}{V} \cdot h$

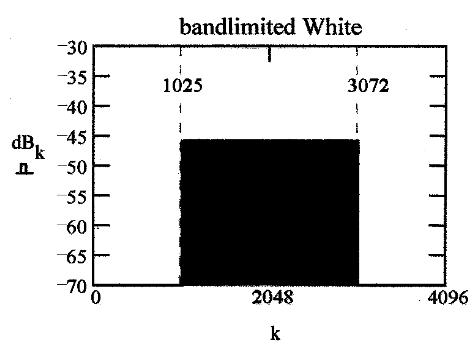
$f := \mathrm{hist}(val, N)$ $s := 0..V - 1$




amplitude spectrum:

$c1 := \mathrm{FFT}(N)$ $M := \mathrm{last}(c1)$ $M = 4096$

$dB_k := 20 \cdot \log\left(\sqrt{2} \cdot |c1_k| + 10^{-4}\right)$ $dB_{1995} = -45.9$




--------------------------

Waveform ["*.PRN" file]:

WRITEPRN( * ) := N

--------------------------

**Fig. 2: Mathcad creates "noise" waveform (N, "*.PRN" file) with 8K sample points and analyzes amplitude distribution (histogram)**

### Part Ia: The real measurement

Now let's transform the simulation into real noise. Import the Mathcad "*.PRN" file with BenchLink/ARB, download it into the Agilent 33120A Function/Arb Gen. and **generate** the noise waveform at **10.1** Hz base frequency. Then **measure** the spectrum of the noise using the Agilent 54602B DSO. (See Fig 3 example.)

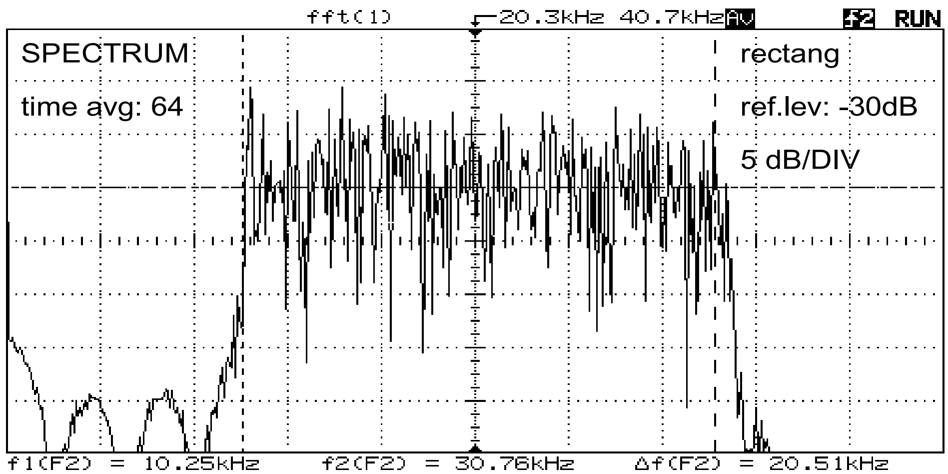



**Fig. 3: Import, download and generate waveform at 10.1 Hz base frequency, and measure spectrum**

### Part II: Combine (signal + noise):

Generate sinusoidal **signal +** flat **noise** with predetermined signal-to-noise ratio (SNR).

Sinusoidal SIGNAL + Gaussian White-NOISE (continued from Fig.4) *(Mathcad)*

SNR [in dB]: $SNR := 7$

signal amplitude: $A := \sqrt{2} \cdot \sigma \cdot 10^{\frac{SNR}{20}}$ $A = 0.7$

signal frequency [in-band]: $F := 1995$

"signal+noise" waveform:

$S_i := A \cdot \cos\left(2 \cdot \pi \cdot \frac{F}{P} \cdot i\right) + N_i$

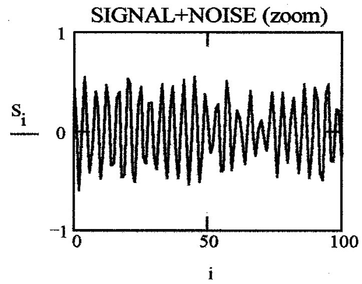
$Up := \max(S)$ $Lo := \min(S)$

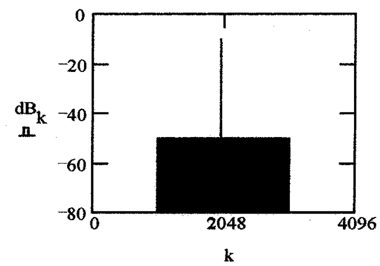
$nor := \mathrm{if}(Up > |Lo|, Up, |Lo|)$

$S_i := \frac{S_i}{nor}$

spectrum:

$c2 := FFT(S)$ $dB_k := 20 \cdot \log\left(\sqrt{2} \cdot |c2_k| + 10^{-4}\right)$




$dB_F = -10.3$

$dB_{F+1} = -50.1$

---------------------------

Waveform ["*.PRN" file]:

WRITEPRN( * ) := S

---------------------------

**Fig. 4: Create "signal + noise" waveform (S, "*.PRN" file) with SNR = 7 dB and 8K sample points**

## Part IIa: Add the signal to the noise for a real measurement

Import the Mathcad "*.PRN" file from Fig 4 with BenchLink/ARB, download it into the Agilent 33120A Function/Arb Gen. and **generate** the combined (signal plus noise) waveform at **10.1** Hz base frequency. (Remember, the Arb treats the entire downloaded waveform as a single cycle, so when the Agilent 33120A Arb's frequency is set to 10.1 Hz, it is repeating everything in the Mathcad waveform window 10.1 times per second.) Then **measure** the spectrum of the (signal+noise) using the Agilent 54602B Oscilloscope. (See Fig 5 example.)

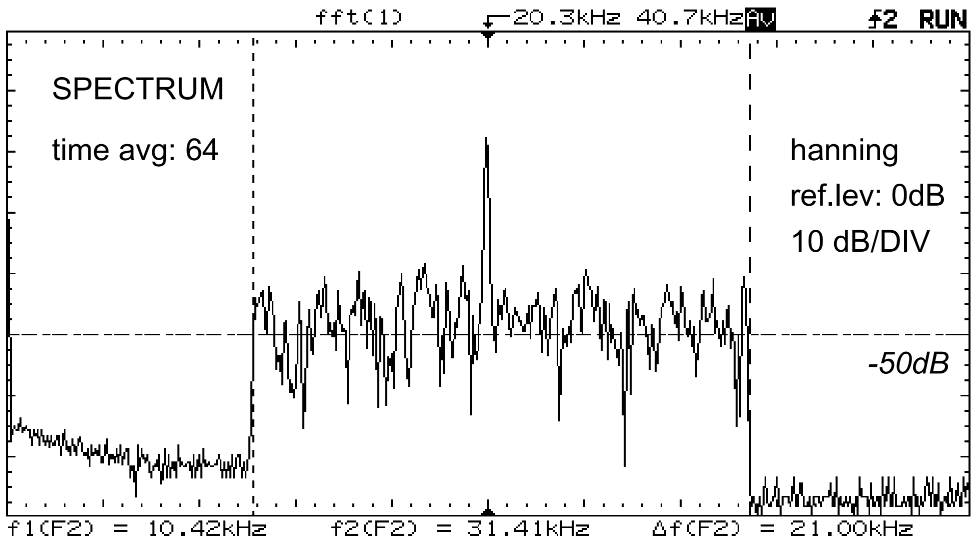



**Fig. 5: Generate waveform at 10.1 Hz base arb. frequency, and measure the spectrum of live "signal + noise"**

***Exercise: Compare the simulated and directly measured spectra***