Digital Programmable Gaussian Noise Generator

Karen Fitch, Kathryn Gillis, Abby Harrison

Project Proposal Presentation

October 14th, 2015

This work is sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.
Outline

- Introduction
- Background
- Methods
- Results
- Conclusion
Group 108 is interested in studying Gaussian vs. structured noise and its effect on various threat radar systems.
Jamming Overview

No Jamming

False Target Jamming

Noise Jamming
Digital Gaussian noise generator is a desirable expansion to existing DRFM capabilities

**DRFM**
- High Configurability
- Plays back recorded radar waveform
  - Power concentrated into narrower frequency band
  - Radar matched filter gives additional power gain
  - Gaussian noise is possible but requires work

**Analog Noise**
- Limited Configurability
- Always wideband Gaussian noise
  - Power level dependent on external amplification
Project Overview

Focus of this talk

White Box Lite (DRFM Repeater)

Software Defined Radar*

*This system will be built up in the 1718 Dev System lab, operating at IF, and will not be tested over-the-air

Digital Noise Generator

Bandwidth
Center Frequency
Amplitude
Digital Noise Generator Requirements

- Standalone hardware
- User programmable center frequency, bandwidth, and amplitude
- Analog output
- Digital in-phase and quadrature output through Ethernet
- Pass test suite for Gaussian distribution
Outline

- Introduction
- Background
- Methods
- Results
- Conclusion
Noise Characteristics

White Noise

**Autocorrelation**

![Autocorrelation Graph](image)

**Power Spectrum**

![Power Spectrum Graph](image)

Band-Limited White Noise

**Autocorrelation**

![Autocorrelation Graph](image)

**Power Spectrum**

![Power Spectrum Graph](image)

*Simulated in MATLAB
Gaussian Characteristics

*Normally distributed 10e6 samples from simulation

<table>
<thead>
<tr>
<th>Test</th>
<th>IQ Average</th>
<th>Rayleigh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-Squared</td>
<td>0.54</td>
<td>0.42</td>
</tr>
<tr>
<td>Anderson-Darling</td>
<td>0.20</td>
<td>N/A</td>
</tr>
</tbody>
</table>

MATLAB Samples

0  p-value  1
No fit  Baseline Goal  Perfect fit
Outline

- Introduction
- Background
- Methods
- Results
- Conclusion
## Considered Hardware Platforms

<table>
<thead>
<tr>
<th>Category</th>
<th>Weight</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Over 10k</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5-10k</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Under 5k</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Bandwidth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 200 MHz</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>&gt; 200 MHz</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RF Range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does not contain 20 MHz</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Contains 20 MHz</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DAC Width</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;=12 bits</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>&gt;12 bits</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DAC Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 100 MS/s</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>100-500 MS/s</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>&gt; 500 MS/s</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Lead Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Over 2 weeks</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1-2 weeks</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>less than a week</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Digital Interface?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Ethernet</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FPGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Xilinx</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Xilinx</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td># Logic Cells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;250</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>250-400</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>400+</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Considered Hardware Platforms

<table>
<thead>
<tr>
<th>Platform</th>
<th>Per Vices Crimson</th>
<th>Beecube Nano Bee</th>
<th>Nutaq uSDR 420</th>
<th>Nutaq PicoSDR</th>
<th>Epic Maveriq</th>
<th>Nuand BladeRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ettus USRP</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Xilinx Kintex-7 DSP Dev Kit</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Price</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RF Range</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DAC Width</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DAC Rate</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lead Time</td>
<td>2</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Digital Interface?</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FPGA</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td># Logic Cells</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Weighted Total**: 36

---

**Score Value**

- **0** (Over 2 weeks)
- **1** (1-2 weeks)
- **2** (less than a week)
- **3** (Over 10k)
- **4** (100-500 MS/s)
- **5** (100-500 MS/s)
- **6** (Contains 20 MHz)
- **7** (Does not contain 20 MHz)
- **8** (Over 10k)
- **9** (Under 5k)
- **10** (< 200 MHz)
- **11** (> 200 MHz)
- **12** (Does not contain 20 MHz)
- **13** (Contains 20 MHz)
- **14** (Over 2 weeks)
- **15** (1-2 weeks)
- **16** (less than a week)
- **17** (No Ethernet)
- **18** (Ethernet)
- **19** (Not Xilinx)
- **20** (Xilinx)
- **21** (<250)
- **22** (250-400)
- **23** (400+)

**Price**

- **0** (Over 10k)
- **1** (5-10k)
- **2** (Under 5k)

**Bandwidth**

- **0** (< 200 MHz)
- **1** (> 200 MHz)

**RF Range**

- **0** (Does not contain 20 MHz)
- **1** (Contains 20 MHz)

**DAC Width**

- **0** (<=12 bits)
- **1** (>12 bits)

**DAC Rate**

- **0** (< 100 MS/s)
- **1** (100-500 MS/s)
- **2** (> 500 MS/s)

**Lead Time**

- **0** (Over 2 weeks)
- **1** (1-2 weeks)
- **2** (less than a week)

**Digital Interface?**

- **0** (No Ethernet)
- **1** (Ethernet)

**FPGA**

- **0** (Not Xilinx)
- **1** (Xilinx)

**# Logic Cells**

- **0** (<250)
- **1** (250-400)
- **2** (400+)

**Weighted Total**

- **36**
Selected Hardware Platform: KC705

- Power
- UART Serial Port
- Ethernet
- Dual-Channel DAC
Methods for Digitally Generating Gaussian Noise

- Inversion
- Box Muller Transform
- Central Limit Theorem
- Recursion
- Analog Sampling
- Rejection

\[ n_1 = \sqrt{-2\ln(u_1)} \times \sin(u_2) \]
\[ n_2 = \sqrt{-2\ln(u_1)} \times \cos(u_2) \]

\( u_1, u_2 \) Uniform
\( n_1, n_2 \) Normal
Defining Bandwidth and Center Frequency

Filter broadband noise

Before

After

VS

Generate at baseband and mix to higher frequencies

Before

After
Generator Flowchart

*Uniform Random Number Generator

**User Inputs**

- **Tausworthe URNG*1**
- **Tausworthe URNG*2**

**Box Muller Transform**

**FIR Filter**

**Output Select**

**DAC Channel 1**

**DAC Channel 2**
Box-Muller Algorithm and Implementation

Fixed Point Operations

User Inputs

Tausworthe Uniform Random Number Generator 1

FIR Filter

Output Select

DAC Channel 1

Tausworthe Uniform Random Number Generator 2

Box Muller Transform

Box Muller Transform

User Inputs

48-bit Log Coeffs

Shift Left by 1 bit

31-bit Sqrt Coeffs

Multiply

Quadrature Data Out

Multiply

In-Phase Data Out

Quadrant and Sign

Sine

Cosine

16-bit Sine and Cosine Coefficient Look-up
Box-Muller Algorithm and Implementation
Floating Point Operations

1. Tausworthe Uniform Random Number Generator 1
   - Convert to Single Precision Float
   - Natural Log
   - Square Root
   - Multiply
   - Convert to 16-bit Integer
   - Quadrature Data Out

2. Tausworthe Uniform Random Number Generator 2
   - Look up table for Sine and Cosine
   - Sine
   - Cosine
   - Convert to Single Precision Float
   - Multiply
   - Convert to 16-bit Integer
   - In-Phase Data Out
Programmability

Terminal Window

- Output Select
- Amplitude
- Bandwidth and Center Frequency

Soft processor calculates coefficients that define an FIR filter

Broadband, Filtered, or Off, and Ethernet

Multiplied by the output on the FPGA

Output Select

DAC Channel 1

DAC Channel 2
Filter Implementation

Unscaled 127 Order FIR Filter Response: 30MHz BW, 30MHz CF

Scaled 127 Order FIR Filter Response: 30MHz BW, 30MHz CF
Outline

• Introduction
• Background
• Methods
• Results
• Conclusion
## Noise Comparison

<table>
<thead>
<tr>
<th>Platform</th>
<th>Data</th>
<th>Test</th>
<th>Chi-Squared</th>
<th>Anderson-Darling</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB</td>
<td>I&amp;Q Average</td>
<td>0.54</td>
<td>0.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>0.2</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>White Box Lite</td>
<td>I&amp;Q Average</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analog Generator</td>
<td>I&amp;Q Average</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KC705 Fixed Point Xilinx Simulation</td>
<td>I&amp;Q Average</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KC705 Floating Point Xilinx Simulation</td>
<td>Rayleigh</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### p-value

- **No fit**
- **Baseline Goal**
- **Perfect fit**

**Legend:**
- No fit
- Baseline Goal
- Perfect fit
Characterizing Existing Noise Sources
White Box Lite Frequency Sweep*

Test Data Distribution Fit to Normal (I)  
Test Data Distribution Fit to Normal (Q)

Test Data fit to Rayleigh

Welch's Power Spectral Density Estimate

<table>
<thead>
<tr>
<th>WBL Freq Sweep</th>
<th>Test</th>
<th>I and Q</th>
<th>Average</th>
<th>Rayleigh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-Squared</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anderson-Darling</td>
<td>0</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Data from 1718

No fit  p-value  1
Baseline Goal
Perfect fit

Noise Generator - 23
KF, KG, AH 10/14/15
Characterizing Existing Noise Sources
Analog Noise Generator*

### Analog 0dB

<table>
<thead>
<tr>
<th>Test</th>
<th>I and Q Average</th>
<th>Rayleigh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-Squared</td>
<td>0.46</td>
<td>0.64</td>
</tr>
<tr>
<td>Anderson-Darling</td>
<td>0.47</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Data from 1718
KC705 Results*

Test Data Distribution Fit to Normal (I)

Test Data Distribution Fit to Normal (Q)

Test Data fit to Rayleigh

<table>
<thead>
<tr>
<th>KC705 Full Amplitude</th>
<th>Test I and Q Average</th>
<th>Rayleigh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-Squared</td>
<td>.44</td>
<td>.55</td>
</tr>
<tr>
<td>Anderson-Darling</td>
<td>.42</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*data from Xilinx Test Bench
KC705 Results*
Power Spectrum

Broadband Output

Filtered Output

*data from spectrum analyzer
<table>
<thead>
<tr>
<th>Platform</th>
<th>Data</th>
<th>Chi-Squared</th>
<th>Anderson-Darling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MATLAB</strong></td>
<td>I&amp;Q Average</td>
<td>0.54</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>0.2</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>White Box Lite</strong></td>
<td>I&amp;Q Average</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Analog Generator</strong></td>
<td>I&amp;Q Average</td>
<td>0.46</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>0.64</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>KC705 Fixed Point Xilinx Simulation</strong></td>
<td>I&amp;Q Average</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>KC705 Floating Point Xilinx Simulation</strong></td>
<td>I&amp;Q Average</td>
<td>0.44</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>0.42</td>
<td>N/A</td>
</tr>
</tbody>
</table>
WBL vs Final Implementation
Outline

• Introduction
• Background
• Methods
• Results
• Conclusion
Future Work

- Filter Improvements
- Change Platforms
  - Double Bandwidth
  - Analog Mixing
  - Enclosed Platform
Conclusion

• Characterized Gaussian noise
• Evaluated techniques for generating noise
• Benchmarked existing technology
• Successfully designed and implemented a digital, programmable, Gaussian noise generator
  – Improvement over existing technology
  – Immediately available for Group 108 testing
Acknowledgements

Mentors
• Lisa Basile
• Ted Clancy
• Sarah Curry
• Emily Fenn
• Chris Massa

MVPs
• Dave Baur
• Brent Dennis
• Dave McQueen

And the rest of group 108!
Thank You

Questions?
Simulation vs 1718
Data Collection