Detection and Analysis of Low-Probability of Intercept (LPI) Radar Waveforms

Presented by: Alvaro Bonilla

Academic Advisor: Prof. Joel Harley

Sponsored by: L-3 Communications

L-3 Communications Liaison: Dr. David Landon

Team Members: Samuel Kingston, Riley Leigh, Ming Gao, Minh Nguyen
Are those who protect us ever safe?
How can we protect them?
What is Radar?
Radar Types

- CW = continuous wave
- FMCW = frequency modulated CW
- PRF = pulse repetition frequency
- MTI = moving target indicator
Low Probability of Intercept (LPI) Radar

• GOAL: To See and Not be Seen
  - Antenna Considerations
  - Achieving Ultra-low Side Lobes
  - Antenna Scan Patterns for Search Processing
  - Advanced Multifunction RF Concept
  - Transmitter Considerations
  - Power Management
  - Carrier Frequency Considerations
Radar Waveforms

**Continuous Wave (CW) Radars**
- Frequency Modulated Continuous Waveform (FMCW)

**Pulsed Radars**
- Phase-Shift Keying Technique
  - Frank Codes
- Frequency-Shift Keying Technique
  - Costas Codes
Project Goal

Design a process to detect and identify LPI radar waveforms
Project Overview

- Digital Receiver
  - Choi-Williams Processing
  - Cyclostationary Processing
  - Signal Detection
- Image Analysis
  - Parameter Extraction
- Classification
  - Type of signal
Research Team

Alvaro Bonilla, Samuel Kingston and Riley Leigh

VHDL Team

Riley Leigh, Ming Gao and Minh Nguyen
Understanding Choi-Williams Distribution (CWD) and Parameter Extraction

By: Alvaro Bonilla
Project Overview

Digital Receiver → Choi-Williams Processing → Image Analysis → Classification

Cyclostationary Processing

Signal Detection
Detection Method

- Choi-Williams Distribution (CWD):

\[ C_f(t, \omega, \phi) = \frac{1}{2\pi} \iiint e^{j(\xi \mu - \tau \mu - \xi t)} \phi(\xi, \tau)A(\mu, \tau)d\mu d\tau d\xi \]

- Start with Wigner-Ville Distribution:

\[ W_x(t, \omega) = \int_\infty^{-\infty} x(t + \frac{\tau}{2})x^*(t - \frac{\tau}{2})e^{-j\omega\tau}d\tau \]
Wigner-Ville Result

$W_x$

Auto-term

Cross Term

Transform Relationship

$F_\tau F_\eta^{-1}$

Ambiguity Function

$F_t F_f^{-1}$

Wigner-Ville Function

$f$

$t$
Wigner-Ville Transform

\( W_x \)

Auto-term

Cross Term

\( A_x \)

Cross Term

\( F_t F_f^{-1} \)
Kernel Function

- Weight function applied to Ambiguity Function
  - Serves as a filter to remove cross-terms

\[
\phi(\xi, \tau) = e^{-\frac{\xi^2}{\sigma^2}}
\]

\[
F_{\tau}F_{\eta}^{-1}
\]
Project Overview

- Digital Receiver
- Cyclostationary Processing
- Costas Codes Parameter Extraction
- Frank Code Parameter Extraction
- FMCW Parameter Extraction
- Processing
- Classification
- Parameter Extraction

Classification
Frequency Modulation Continuous Waveform (FMCW)
FMCW

\[ \frac{dF}{dt} = \frac{2\Delta F}{T_m} \]

\[ R = \frac{c \Delta t}{2} \]

\[ 4\Delta F - f_{IF} \]
Image Analysis

- Detection Result for FMCW
Image Analysis

$$F_c = E[C_f]$$
Image Analysis

\[ B = \Delta F \]

Time-Frequency plot of CWD
Parameter Extraction

\[ t_m = \frac{Max_2 - Max_1}{2} \]
Parameter Extraction
## Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Generated LPI Parameters</th>
<th>Estimated Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier Frequency ((F_c))</td>
<td>1000 Hz</td>
<td>1056 Hz</td>
</tr>
<tr>
<td>Bandwidth ((\Delta B))</td>
<td>500 Hz</td>
<td>496 Hz</td>
</tr>
<tr>
<td>Time Modulation Period ((t_m))</td>
<td>20 ms</td>
<td>20 ms</td>
</tr>
</tbody>
</table>
Phase-Shift Keying Technique

Frank Codes
Frank Codes

For N-phase Frank code the phase of each sub-pulse is computed from:

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & \ldots & 0 \\
0 & 1 & 2 & 3 & \ldots & N-1 \\
0 & 2 & 4 & 6 & \ldots & 2(N-1) \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & (N-1) & 2(N-1) & 3(N-1) & \ldots & (N-1)^2 \\
\end{pmatrix} \Delta \varphi
\]

We divide 360° by N number of codes in order to obtain \( \Delta \varphi \)
Frank Codes

For example, if we want $N = 4$ number of codes, we will get the following result:

$$F_{16} = \{1, 1, 1, 1, j, -1, -j, 1, -1, 1, -1, 1, -j, -1, j\}$$
LPI Waveforms Analyzed
Image Analysis

- Detection Result for Frank Codes LPI
Image Analysis

- Detection Result for Frank Codes LPI
Parameter Extraction

• Radon Transform

\[ R(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) \, dx \, dy \]
Radon Transform

$$R(\rho, \theta)$$
Radon Transform
Parameter Extraction

Using the results, and the following equations:

\[ T = -\frac{1}{f_s} \left[ \frac{d}{\cos(\theta_s)} \right] \]

\[ B = \Delta f \left[ \frac{d}{\cos(\theta_s)} \right] \frac{1}{\tan(\theta_s)} \]

\[ N_c = T \times B \]

\[ cpp = \frac{f_c}{B} \]

Parameters can be successfully recovered.
Results

Projection Vector at angle $\theta$
Results
\[ T = -\frac{1}{f_s} \left[ \frac{d}{\cos(\theta_s)} \right] \]

\[ B = \Delta f \cdot \left[ \frac{d}{\cos(\theta_s)} \right] \div \tan(\theta_s) \]

\[ N_c = T \cdot B \]

\[ cpp = \frac{f_c}{B} \]
# Results

<table>
<thead>
<tr>
<th></th>
<th>Generated LPI Parameters</th>
<th>Estimated Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Carrier Frequency</strong> ($F_c$)</td>
<td>1000 Hz</td>
<td>1125 Hz</td>
</tr>
<tr>
<td><strong>Bandwidth</strong> ($\Delta B$)</td>
<td>1000 Hz</td>
<td>1022 Hz</td>
</tr>
<tr>
<td><strong>Time Modulation Period</strong> ($t_m$)</td>
<td>68 ms</td>
<td>68.4 ms</td>
</tr>
<tr>
<td><strong>Number of sub-pulses</strong> ($N_c^2$)</td>
<td>64</td>
<td>64</td>
</tr>
</tbody>
</table>
Conclusion

- Research and Simulation part successfully completed
- Choi-Williams Distribution will be used for implementation on hardware
References

References

• Images used in presentation
  o https://en.wikipedia.org/wiki/Bilinear_time%E2%80%93frequency_distribution#Choi.2880.93Williams_distribution_function
  o http://www.st-andrews.ac.uk/~mmwave/mm-waves/avtis/theory-mmw-imaging/radar/
  o http://www.slideshare.net/solohermelin/5-pulse-compression-waveform
Questions?

• Contact Information:

Alvaro Bonilla
Email - u0664523@utah.edu
Phone - (801) 318-2107
Low Probability of Intercept (LPI) Radar Detection and Classification Through Image Analysis

Presented by: Samuel Kingston

Academic Advisor: Prof. Joel Harley

Sponsored by: L-3 Communications

L-3 Communications Liaison: Dr. David Landon

Team Members: Alvaro Bonilla, Riley Leigh, Minh Nguyen, Ming Gao
Presentation Outline

• Goal of Project
• Experiments Performed
• Results of The Experiments
• Conclusion
Goal of Project

• Identify LPI Signal
  o Parameter Extraction
  o Reconstruction of Signal
  o Signal Classification
Identifying LPI Signal

- Time Frequency Analysis
- FMCW Parameter Extraction
- Frank Code Parameter Extraction
- Costas Codes Parameter Extraction
- FMCW Signal Reconstruction
- Frank Code Signal Reconstruction
- Costas Codes Signal Reconstruction
- Parameter Extraction
- Signal Reconstruction
- Cross Correlator
- Signal Classification
FMCW signal

- Carrier Frequency
- Bandwidth
- Modulation Period
Frank Codes Signal

- Carrier Frequency
- Bandwidth
- Number of Phase Codes
- Number of Cycles/Phase
Costas Codes Signal

- Carrier Frequencies with their Sequence
- Frequency Time Duration

![Power Spectrum Magnitude vs Frequency](image)
Identifying LPI Signal

- Time Frequency Analysis
- FMCW Parameter Extraction
- Frank Code Parameter Extraction
- Costas Codes Parameter Extraction
- Parameter Extraction
- FMCW Signal Reconstruction
- Frank Code Signal Reconstruction
- Costas Codes Signal Reconstruction
- Signal Reconstruction
- Cross Correlator
- Signal Classification
LPI Waveforms After Time Frequency Analysis

- **FMCW waveform**
- Bandwidth (BW)
- Modulation Period (Tm)
- Carrier Frequency (fc)
Parameter Extraction for FMCW Signal

Mesh of CWD for FMCW

Mesh of CWD for FMCW

fc

bw

T

Tm

X: 0.02271
Y: 2669
Z: 2.154

X: 0.02029
Y: 1094
Z: 67.76

X: 0.08829
Y: 984.4
Z: 55.19

X: 0.01929
Y: 1094
Z: 81.55

X: 0.09043
Y: 1001
Z: 255.2

X: 0.1097
Y: 2018
Z: 0.2581

X: 0.02943
Y: 2611
Z: 0

X: 0.1116
Y: 2721
Z: 0
LPI Waveforms

- **Costas Codes**
- Frequency Hop Sequence (3kHz, 2kHz, 6kHz, 4kHz, 5kHz, 1kHz)
- Frequency Duration (T)
Parameter Extraction for Costas Codes Signal
Costas Code Parameter Extraction

\[ \overline{S} = \frac{1}{m} \sum_{j=1}^{m} S_j \]

Carrier Frequency

Frequency value at given index

# of Elements in row
FMCW Signal Reconstruction

- Original param: $F_c = 1\text{kHz}, \ B_W = 450\text{Hz}, \ T_m = 0.05\ \text{sec}$
- Extracted param: $F_c = 978\text{Hz}, \ B_W = 479\text{Hz}, \ T_m = 0.054\ \text{sec}$
FMCW Signal Reconstruction in Noise

- **Original param:** $F_c = 1\text{kHz}$, $BW = 450\text{Hz}$, $T_m = 0.05\text{ sec}$ \(w/\ SNR = 0\text{dB}\)
- **Extracted param:** $F_c = 993\text{Hz}$, $BW = 506\text{Hz}$, $T_m = 0.0477\text{ sec}$
FMCW Signal Reconstruction in Noise

- Original param: $F_c = 1\text{kHz}$, $BW = 450\text{Hz}$, $T_m = 0.05\text{sec}$ w/ $\text{SNR} = -6\text{dB}$
- Extracted param: $F_c = 1.2\text{kHz}$, $BW = 1261\text{Hz}$, $T_m = 0.0691\text{sec}$
Frank Code Signal Reconstruction (No Noise)

- Original param: $F_c = 1\text{kHz}$, $T_m = 0.064$, Phase Codes = 8, Cycles = 1 w/ no noise
- Extracted param: $F_c = 1015\text{Hz}$, $T_m = 0.0642$ Phase Codes = 8, Cycles = 1
Frank Code Signal Reconstruction in Noise

- Original param: $F_c = 1\text{khz}$, $T_m = 0.064$, Phase Codes = 8, Cycles = 1
  w/ noise: SNR = 0dB
- Extracted param: $F_c = 1125\text{hz}$, $T_m = 0.068$, Phase Codes = 8, Cycles = 1
Costas Code Signal Reconstruction

- Original param: $F_c = [3\text{kHz}, 2\text{kHz}, 6\text{kHz}, 4\text{kHz}, 5\text{kHz}, 1\text{kHz}]$, $T_m = .01 \text{ sec}$
- Extracted param: $F_c = [2.99\text{kHz}, 2\text{kHz}, 5.99\text{kHz}, 3.99\text{kHz}, 5\text{kHz}, 1\text{kHz}]$, $T_m = .009 \text{ sec}$
Costas Code Signal Reconstruction in Noise

- Original param: $F_c = [3\text{kHz}, 2\text{kHz}, 6\text{kHz}, 4\text{kHz}, 5\text{kHz}, 1\text{kHz}]$, $T_m = 0.01 \text{ sec}$
  - w/ Noise: SNR = $0\text{dB}$
- Extracted param: $F_c = [3\text{kHz}, 1.998\text{kHz}, 6\text{kHz}, 3.99\text{kHz}, 4.98\text{kHz}, 995\text{Hz}]$, $T_m = 0.0079 \text{ sec}$
Identifying LPI Signal

- Time Frequency Analysis
  - FMCW Parameter Extraction
  - Frank Code Parameter Extraction
  - Costas Codes Parameter Extraction
- Parameter Extraction
- Signal Reconstruction
- FMCW Signal Reconstruction
- Frank Code Signal Reconstruction
- Costas Codes Signal Reconstruction
- Signal Classification
- Cross Correlator
Classification of Signals

- **Using Matlab Function: XCORR**
- **Test Results with 80 different signals**
  - 88.7% classified correctly
  - 11.25% classified incorrectly
  - 1.25% False Alarm

```
Command Window

LPI signal has been detected
FMCW LPI signal has been detected with parameters:

Carrier Frequency: 9.960531e+02
Bandwidth: 287
Time Modulation Period: 0.02 (s)
```

```
Command Window

This is not an LPI Signal
```
Conclusion

• Extraction of Parameters was successful
• Reconstruction of signal
• Cross Correlated reconstructed signal with original signal
• **LPI radar detected**
We Found Waldo


Questions?

• Contact Information:

  • Samuel Kingston
  • u0763921@Utah.edu
Cyclostationary Processing Research and VHDL Implementation of the Choi-Williams TF Distribution

Presented by: Riley Leigh

Academic Advisor: Prof. Joel Harley

Sponsored by: L-3 Communications

L-3 Communications Liaison: Dr. David Landon

Team Members: Minh Nguyen, Alvaro Bonilla, Samuel Kingston, Ming Gao
Presentation Outline

1 – Cyclostationary Processing Research

2 – VHDL Implementation of the Choi-Williams Time Frequency Distribution
Cyclostationary Processing Research
Radar Chirp
Frequency Modulated CW

Costas Codes

Frank Codes
Cyclostationary Processing Goal

Determine if Cyclostationary Processing is the most viable technique in detecting LPI Radar waveforms
Project Overview

- Digital Receiver
- Research
- Cyclostationary Processing
- Choi-Williams Processing
- Image Analysis
- Matlab Simulations
- Classification
Cyclostationary Theory

![Graph showing cyclostationary signal processing](image)
Cyclostationary Theory
Cyclostationary Theory

\[ \Delta f = \frac{1}{T_w} \]

\[ \Delta \alpha = \frac{1}{\Delta t} \]
Direct Frequency Smoothing Method (DFSM)

\[ S_{x_N}^\gamma (n, k)_{\Delta k} = \sum_{m} X_N \left( n, k + \frac{\gamma}{2} \right) X_N^* \left( n, k - \frac{\gamma}{2} \right) \]
Simulation

Phillip E. Pace - Detecting and Classifying Low Probability of Intercept Radar

Frank Code Modulation

• Carrier Frequency – 1000 Hz
• Number of Phase Codes – 8
• Number of Cycles Per Phase - 1
Frequency Smoothing SCD, df = 128, N = 128

- Frequency (Hz)
- Cycle frequency (Hz)

- BW
- Code Rate
Frank Code Modulation with SNR = 0 dB
Research Conclusions

• Viable Solution
• Computationally Demanding
  • Difficult Parameterization
• Proceed with Choi-Williams
VHDL Implementation of the Choi-Williams TF Distribution
Project Overview

- Digital Receiver
- Signal Sampler
- Specialized Shifter
- Cyclostationary Processing
- Choi-Williams Processing
- Image FFT
- Mathematical Operations
- Classification

Still In Progress
Specialized Shifter

Input(Signal, Shift Amount)

Positive Shift Amount
- Shift Right by Amount
  - Zero Fill Shifted Indexes

Negative Shift Amount
- Shift Left by Amount
  - Zero Fill Shifted Indexes
FFT

Xilinx IP Core

Input Signal

FFT
Signal Sampler → Specialized Shifter → FFT → Mathematical Operations

→

Output Processed Signal to Image Analysis
Conclusion

• C y l o s t a t i o n a r y P r o c e s s i n g
• T h e o r y b e c o m i n g a r e a l i t y
Frequency Modulated CW

Frank Codes

Costas Codes
References

http://web.ewu.edu/groups/technology/Claudio/ee430/Cad/AccoladeVHDLref.pdf
Questions?

Contact Information:

Riley Leigh
riley.j.leigh@gmail.com
u0564969@utah.edu
(801)-644-4112
VHDL Implementation of LPI Signal Reconstruction

Presented by: Ming Gao

Academic Advisor: Prof. Joel Harley
Sponsored by: L-3 Communications
L-3 Communications Liaison: Dr. David Landon
Team Members: Minh Nguyen, Alvaro Bonilla, Samuel Kingston, Riley Leigh
Building a Bookshelf
Why VHDL?

- Fast
- Design & Simulation before translate into hardware
- Hardware (FPGA)
Classification

Pulse Generator

Sine Wave Generator

NCO

LPI Signal Reconstruction
VHDL Simulation - Pulse Generator
VHDL Simulation – Sine Wave Generator
Classification

- Pulse Generator
- Sine Wave Generator
- NCO
- LPI Signal Reconstruction
Numerically Controlled Oscillator (NCO) Block Diagram
32-bit Phase Accumulator

- Fix a sampling frequency: 100 MHz (clk on off every 5ns)

- Select a frequency of the output waveform

- Calculate a constant phase value (input to the NCO)

\[ \Phi_{\text{increment}} = \left( \frac{F_{\text{out}}}{F_s} \right) \times 2^{32} \]
Example (Output 1.7 MHz)

\[ \Phi_{\text{increment}} = \left( \frac{F_{\text{out}}}{F_s} \right) \times 2^{32} = \left( \frac{1.7 \text{MHz}}{100 \text{MHz}} \right) \times 2^{32} = 73014444 \]

Frequency Resolution = \( \frac{F_s}{2^{32}} = \frac{100 \text{MHz}}{2^{32}} = 0.00233 \text{Hz} \)

Phase Resolution = \( \frac{2\pi}{2^{12}} = \frac{2\pi}{4096} = 0.088^\circ \)
### Example (Output 1.7 MHz)

<table>
<thead>
<tr>
<th>Phase (Integer)</th>
<th>Phase (32-bit binary)</th>
<th>First 12-bit</th>
<th>Decimal</th>
</tr>
</thead>
</table>
| 73014444        | 0000010001011010001111 | 00000100010 1
|                 | 0010101100             |              | 69*n\textsuperscript{th} index in the LUT |

Phase Resolution = \[ \frac{2\pi}{2^{12}} = \frac{2\pi}{4096} = 0.088^\circ \]

1\textsuperscript{st} Sine Wave output = 2046\times \text{sine}(69\times 0 \times 0.088) = 0

2\textsuperscript{nd} Sine Wave output = 2046\times \text{sine}(69\times 1 \times 0.088) = 216

3\textsuperscript{rd} Sine Wave output = 2046\times \text{sine}(69\times 2 \times 0.088) = 433
VHDL Simulation Output (1.7MHz)

1\textsuperscript{st} Sine Wave output = 2046\times \text{sine}(69\times 0\times 0.088) \neq 0

2\textsuperscript{nd} Sine Wave output = 2046\times \text{sine}(69\times 1\times 0.088) = 216

3\textsuperscript{rd} Sine Wave output = 2046\times \text{sine}(69\times 2\times 0.088) = 433
VHDL Simulation Output (1.7MHz)
Classification

Pulse Generator

Sine Wave Generator

NCO

LPI Signal Reconstruction
Reconstructing LPI Signals Using Extracted Parameters
Time-Frequency Plot of Choi-William Distribution (FMCW)
Specifications

- **Running Time:** 80ms (2 periods)
- **Number of Data:** 800
- **Clock Rate:** \( \frac{80\text{ms}}{800} = 0.1\text{ms} \)
- **Sampling Frequency:** 0.01MHz
- **Number of Clock Cycles:** \( \frac{20\text{ms}}{0.1\text{ms}} = 200 \)
- **Bandwidth:** 500Hz
- **Slope:** \( \frac{500\text{Hz}}{200} = 2.5\text{Hz/clock cycle} \)
\[
\Phi_{inc(F_{total})} = \Phi_{inc(F_{total})} - \Phi_{inc(F_{slope})}
\]

\[
\Phi_{inc(F_{start})} = \frac{F_{start}}{F_s} \times 2^{32} = \frac{750\text{Hz}}{0.01\text{MHz}} \times 2^{32} = 322122547_{dec}
\]

\[
\Phi_{inc(F_{slope})} = \frac{F_{slope}}{F_s} \times 2^{32} = \frac{2.5\text{Hz}}{0.01\text{MHz}} \times 2^{32} = 1073741_{dec}
\]
William Distribution (Frank Codes)
Specifications

- Running Time: 80ms
- Number of Data: 400
- Clock Rate: \( \frac{80\text{ms}}{400} = 0.2\text{ms} \)
- Sampling Frequency: \( 0.005\text{MHz} = 5\text{KHz} \)
- Number of Clock Cycles: \( \frac{40\text{ms}}{0.2\text{ms}} = 200 \)
- Bandwidth: 500Hz
- Number of Clock Cycles Stalled Per Frequency: 10
- Slope: \( \frac{500\text{Hz}}{(200/10)} = 25\text{Hz/10 clock cycles} \)
Reconstructed Frank's Code Signals

Time (0.1 ms)
Time-Frequency Plot of Choi-William Distribution (Costas Code Signals)
Specifications

- **Clock Rate**: 0.5ms
- **Sampling Frequency**: $0.002 \text{MHz} = 2 \text{KHz}$
- **Frequency Hoping**: 750Hz, 2500Hz, 1250Hz, 3300Hz, 2800Hz, 2000Hz
- **Number of Clock Cycles Stalled Per Frequency**: 40
Conclusion

• Pulse Generator
• Sine Wave Generator
• NCO
• LPI Signal Reconstruction
  ▪ FMCW
  ▪ Frank Codes
  ▪ Costas Codes
• Software Optimization
References

References (Images)

• http://www.designandhome.xyz/wp-content/uploads/2015/12/small-white-bookcase.jpg
Questions?

Contact Information:

Ming Gao
minggao92@gmail.com
u0673564@utah.edu
VHDL Implementation of a Signal Comparator

Presented by: Minh Nguyen

Academic Advisor: Prof. Joel Harley

Sponsored by: L-3 Communications

L-3 Communications Liaison: Dr. David Landon

Team Members: Ming Gao
Riley Leigh, Alvaro Bonilla, Samuel Kingston
How Are They Related?
Comparator

• The goal is to detect and classify an incoming LPI signal
  o The comparator module provided a means to compare two signals
Expected Results

- Case 1: Both \( x_1 \) and \( x_2 \) are the same
- Expect comparator to return true (1)
Comparator Module

VHDL Simulation
Expected Results

- Case 2: \( x_1 \) differs from \( x_2 \) by one value
- Expect comparator to return false (0)
### Comparator Module

#### VHDL Simulation

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>140 ns</th>
<th>150 ns</th>
<th>160 ns</th>
<th>170 ns</th>
<th>180 ns</th>
<th>190 ns</th>
<th>200 ns</th>
<th>210 ns</th>
<th>220 ns</th>
<th>230 ns</th>
<th>240 ns</th>
<th>250 ns</th>
<th>260 ns</th>
<th>270 ns</th>
<th>280 ns</th>
<th>290 ns</th>
<th>300 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rst</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x1[0:4]</td>
<td>[12, 10, 25, 1]</td>
<td>0,0,0,0,0</td>
<td>0,0,0,0,0</td>
<td>[12, 10, 25, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2[0:4]</td>
<td>[12, 10, 25, 1]</td>
<td>0,0,0,0,0</td>
<td>0,0,0,0,0</td>
<td>[12, 10, 25, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>verdict</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

X1: 250.087 ns
Cross-Correlation

0 Lag Position

\[(0.25)(0.18) + (0.45)(0.8) + (0.15)(0.65) + (0.75)(0.48) + (0.65)(0.26) + (0.5)(0.58) = 1.32\]
(0.45)(0.18) + (0.15)(0.8) + (0.75)(0.65) + (0.65)(0.48) + (0.5)(0.26) + (0.3)(0.58) = 1.31
Cross-Correlation

2nd Lag Position

(0.15)(0.18) + (0.75)(0.8) + (0.65)(0.65) + (0.5)(0.48) + (0.3)(0.26) + (0.6)(0.58) = 1.72
Matlab vs. VHDL

Matlab

- `xcorr(x1, x2)`

VHDL

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;
use work.my_data_types.all;

entity cross_correlation is
  -- declaring ports
  port( clk : in std_logic;
        x1 : in input_vector;
        x2 : in input_vector;
        corr_seq : out output_std );

end cross_correlation;

architecture Behavioral of cross_correlation is
  signal l : integer := 0;
  signal k : integer range 0 to (x2'length-1) := 0;
  signal l : integer range 1 to (x2'length-1) := 1;
  signal sum : integer;
  signal temp1, temp2, tempx2 : input_vector;

begin
  process(clk)
  begin
    if clk = '0' then
      l <= 0;
      k <= 0;
      l <= 1;
      temp1 <= x1;
      temp2 <= x2;
      tempx2 <= x2;
      sum <= 0;
      mult <= (others=>'0');
    elsif clk = '1' then
      if l <= (x2'length-1) then
        sum <= sum + summul(temp1(k) * temp2(1)); -- sum all of the multiplications
        l <= i+1; -- increment do that the next multiplication can be done
      end if;
      if l = x2'length then
        l <= 0; -- reset index
        sum <= 0; -- reset the summation of multiplications
        k <= k+1; -- increment to the next element of the correlation sequence
    end if;
  end process;

  for l in 1 to (x2'length-1) loop
    temp2(1) <= tempx2(1-l); -- shift to the next lag position
  end loop;

  temp2(0) <= "000000000";
  tempx2 <= temp2;

  if (temp2(x2'length-1) = x2(0)) then
    i <= 5; -- set 1 such that the system stops
  end if;
end Behavioral;
```

**Expected Results**

- $x_1$ and $x_2$ are quite similar signals
- Should have high correlation
## VHDL Simulation Results

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>1</td>
</tr>
<tr>
<td>rst_n</td>
<td>1</td>
</tr>
<tr>
<td>i</td>
<td>111</td>
</tr>
<tr>
<td>k</td>
<td>111</td>
</tr>
<tr>
<td>x1[0:7]</td>
<td>[1, 2, 5, 4, 3, 1]</td>
</tr>
<tr>
<td>x2[0:7]</td>
<td>[2, 5, 4, 3, 10, 1]</td>
</tr>
<tr>
<td>corr_seq[0:7]</td>
<td>[886, 878, 632]</td>
</tr>
<tr>
<td>temp2[0:7]</td>
<td>[0, 0, 0, 0, 0, 0, 0]</td>
</tr>
</tbody>
</table>
VHDL Simulation Results

Cross Correlation Sequence

Correlation Value

Lag Position
Reconstructed Costas Code

Reconstructed FMCW Signals
VHDL Simulation Results

Cross Correlation Sequence

Correlation Value

Lag Position

Name   Value
clk    1
rst_n  1
x1[0:99] [0, 0, 0, 25, 1655, 1948, 1450, 633, -319, -1202, -1823, -2047, -1824, -1204, -322, 630, 1445, 1947, 2022, 1657, 932, 3, -929, -1655, -2...
x2[0:99] [0, 0, 0, 29, 1655, 1892, 1269, 241, -855, -1700, -2047, -1704, -786, 476, 1556, 2040, 1704, 633, -707, -1784, -2008, -1269, 157, 1502, ...

corr_seq[0:99] [1397239, -321268, 1397239, -321268, 7186256, 633884...

temp2[0:99] [0, 0, 0, 0, 0, 1655, 1948, 1450, 633, -319, -1202, -1823, -2047, -1824, -1204, -322, 630, 1445, 1947, 2022, 1657, 932, 3, -929, -1655, -2...
txt_values2[0:99] [0, 0, 0, 25, 1655, 1948, 1450, 633, -319, -1202, -1823, -2047, -1824, -1204, -322, 630, 1445, 1947, 2022, 1657, 932, 3, -929, -1655, -2...
Strategy Overview

Digital Receiver → Choi-Williams Processing

Cyclostationary Processing

Image Analysis → Classification
FMCW

Choi Williams

Processing

Matlab Scripts
Project Overview

- Digital Receiver
- Choi-Williams Processing
  - Cyclostationary Processing
  - Image Analysis
- Classification
# Image Analysis

<table>
<thead>
<tr>
<th></th>
<th>Generated LPI Parameters</th>
<th>Estimated Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Carrier Frequency</strong> ($F_c$)</td>
<td>1000 Hz</td>
<td>1056 Hz</td>
</tr>
<tr>
<td><strong>Bandwidth</strong> ($\Delta B$)</td>
<td>500 Hz</td>
<td>496 Hz</td>
</tr>
<tr>
<td><strong>Time Modulation Period</strong> ($t_m$)</td>
<td>20 ms</td>
<td>20 ms</td>
</tr>
</tbody>
</table>
Project Overview

Digital Receiver → Choi-Williams Processing

Cyclostationary Processing

Image Analysis → Classification
Signal Reconstruction

Original FMCW In

Signal Reconstructed from Extracted Parameters

Time-Frequency plot of CWD

Time-Frequency plot of CWD FMCW
Cross-Correlation

Signal Reconstructed from Extracted Parameters

Cross-Correlator Matlab Scripts
VHDL Strategy Overview

CWD

NCO

Cross Correlator
Conclusion

• Cross-correlation is an efficient method for comparing signals

• Apply the theory to provide a electronic support system that may protect our service men and women
Are those who protect us ever safe?
References


Questions?

• Contact information:
  o Minh Nguyen
  o u0576781@utah.edu