TIME-DOMAIN EVALUATION OF ATMOSPHERIC DUCTING EFFECTS ON
X-BAND PROPAGATION OVER WATER

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering

by
Jack Aaron Gallegos
March 2023
COMMITTEE MEMBERSHIP

TITLE: Time-Domain Evaluation of Atmospheric Ducting Effects on X-Band Propagation Over Water

AUTHOR: Jack Aaron Gallegos

DATE SUBMITTED: March 2023

COMMITTEE CHAIR: Dennis Derickson, Ph.D
Professor of Electrical Engineering

COMMITTEE MEMBER: Clay McKell, Ph.D
Electrical Engineering Lecturer

COMMITTEE MEMBER: Steve Dunton, MS
Electrical Engineering Lecturer
ABSTRACT
Time-Domain Evaluation of Atmospheric Ducting Effects on X-Band Propagation Over Water
Jack Aaron Gallegos

The marine atmospheric boundary layer (MABL) is the region of atmosphere that interacts with the ocean surface. The atmospheric variability (i.e. temperature and relative humidity) in this region can result in rapid changes in the refractive index with increasing height from the sea surface. The complex region can result in non-standard propagation of electromagnetic (EM) waves beyond the horizon under atmospheric ducting conditions. However, when ducting layers are not present, EM waves are limited to line-of-sight transmission. Atmospheric ducting research is typically conducted using radio frequencies in the X-band (around 8-12 GHz) due to its impact on performance of marine radars at those frequencies. Studies typically examine levels of received signal power or effects on radar returns in ducting conditions, but often ignore the time-domain effects of ducting which can also affect communications link performance.

In collaboration with the Coastal Observing Research and Development Center at Scripps Institution of Oceanography (SIO), the ducting research in this thesis uses a channel sounder that consists of a X-band transmitter which transmits a coded pseudorandom sequence and a software-defined radio (SDR) receiver. Both transmitter and receiver are GPS synchronized so that the time-domain cross-correlation between the TX and RX signals can be found. In theory, if atmospheric ducting is present, there will be multipath propagation, and the TX-RX cross-correlation indicates multiple “peaks”, indicating multiple arrival times. Conversely, if little to no ducting is present, then the cross-correlation indicates a single “peak”. The chan-
nel sounding was evaluated over several over-water communications links, involving fixed-path and variable range sea tests with a moving vessel to verify if this hypothesis is true. The expected ducting conditions were determined by \textit{in-situ} refractive index measurements of the atmosphere. Results from testing showed multiple peaks when strong ducting was expected, but an extensive sea test in strong ducting conditions is needed to distinguish multipath from ducting from that of terrain reflections. Further work is also needed to determine the computational model that accurately models multipath propagation through a duct, which is beyond the scope of this thesis.
ACKNOWLEDGMENTS

Thank you to my parents, for their support and encouragement along the way. Thank you to all the SIO staff and faculty who helped me, especially: Tony De Paolo, Dr. Peter Rogowski, Dr. Sophia Merrifeld, and Dr. Eric Terrill, for inspiring this thesis topic, providing scientific insight, and without whom I would have not been able to handle experimentation. I would also like to thank Dylan Shields for his assistance getting me up to speed on prior SIO research and experimentation in this area. Thank you to the various members of the Cal Poly Amateur Radio Club for their help motivating me to get work done on this thesis when I needed it. I would also like to thank my thesis committee of Cal Poly faculty: Dr. Dennis Derickson, Dr. Clay McKell, and Mr. Steve Dunton for their feedback, support, and understanding. I also would like to thank the Office of Naval Research (ONR) and ONR Program Manager Dan Eleuterio, for their financial support of this research. Thank you to everyone else who helped along the way.
# TABLE OF CONTENTS

<p>| LIST OF TABLES                          | ix |
| LIST OF FIGURES                        | x |
| 1 Introduction                        | 1 |
| 1.1 Motivation                        | 1 |
| 1.2 Atmospheric Refraction            | 2 |
| 1.3 An Examination of Methods of Ducting Observation | 9 |
| 1.3.1 Refractivity Measurement from Direct Sampling of Environment | 9 |
| 1.3.2 Propagation Loss (PL) Method    | 12 |
| 1.4 Ducting Estimation From Multipath Delay Measurement | 16 |
| 1.5 Modeling Delay Through a Duct     | 19 |
| 1.6 Explanation of Pseudorandom Sequence Choice | 22 |
| 1.7 Project History and My Contribution | 25 |
| 1.8 Content of This Thesis            | 26 |
| CHAPTER                                | |
| 2 System Design and Construction      | 27 |
| 2.1 System Overview                   | 27 |
| 2.2 PN Transmitter Design             | 29 |
| 2.3 PN Receiver Design                | 37 |
| 3 System Simulations                  | 43 |
| 3.1 Phase Shifter Jitter Simulation   | 45 |
| 3.2 Emulating Multipath Effects With Time Delay | 52 |
| 3.3 Effect of Frequency Offset on Received Signal | 55 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>System Testing and Results</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>July 2022 Fixed Path Test</td>
<td>57</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Post-Processing Methodology</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>July 2022 Sea Test</td>
<td>68</td>
</tr>
<tr>
<td>4.2.1</td>
<td>COAMPS Ducting Forecast</td>
<td>72</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Unmanned Aerial Vehicle Measurements</td>
<td>73</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Sea Test Data Analysis</td>
<td>76</td>
</tr>
<tr>
<td>4.3</td>
<td>September 2022 Over-Water Fixed Path Test</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>87</td>
</tr>
<tr>
<td>5.1</td>
<td>Discussion of System Goals</td>
<td>87</td>
</tr>
<tr>
<td>5.2</td>
<td>Interpretation of Experimental Results</td>
<td>87</td>
</tr>
<tr>
<td>5.3</td>
<td>Future Work</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td><strong>BIBLIOGRAPHY</strong></td>
<td>92</td>
</tr>
<tr>
<td>A</td>
<td>C++ Receive Program Code to Interface with Ettus Universal Hardware Driver (UHD)</td>
<td>95</td>
</tr>
<tr>
<td>B</td>
<td>MATLAB Post-Processing Code</td>
<td>118</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Desired system specifications/design goals.</td>
</tr>
<tr>
<td>2.1</td>
<td>Statistical analysis of TX pipeline delay.</td>
</tr>
<tr>
<td>2.2</td>
<td>Statistical analysis of RX sampling latency.</td>
</tr>
<tr>
<td>4.1</td>
<td>Statistical analysis of the main peak correlation delay from the fixed-path link in the Cardiff test.</td>
</tr>
<tr>
<td>5.1</td>
<td>Actual system specifications.</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Bending of EM wave according to Snell’s Law [1].</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Example bending of EM wave across a gradient resulting from a stack of thin slabs with homogeneous refractive index [1].</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Assumptions of “standard atmosphere” model [2].</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Real atmospheric conditions near the ocean surface when there is a temperature inversion present [2]. The temperature inversion shown here creates a trapping layer that can trap EM waves, starting near the surface and spanning the MABL.</td>
<td>6</td>
</tr>
<tr>
<td>1.5</td>
<td>Plots of modified refractivity versus altitude for different refractive conditions: (a) sub-refractive layer denoted by dashed line; (b) normal refraction; (c) elevated duct denoted by dashed line; (d) surface duct denoted by dashed line; (e) surface duct (dashed line) due to elevated region of strongly negative vertical refractivity gradient; (f) evaporation duct denoted by dashed line [3].</td>
<td>7</td>
</tr>
<tr>
<td>1.6</td>
<td>Effect of radio wave “bending” due to inflection points in the refractivity profile [4].</td>
<td>7</td>
</tr>
<tr>
<td>1.7</td>
<td>Refractivity profiles for various ducting phenomena. Profiles shown: a) evaporation duct, b) surface-based duct, and c) elevated duct [5].</td>
<td>8</td>
</tr>
<tr>
<td>1.8</td>
<td>Left: InterMet iMet-54 Research Radiosonde. Right: A balloon for hosting a radiosonde [6].</td>
<td>9</td>
</tr>
<tr>
<td>1.9</td>
<td>Balloon radiosonde observations of (a) Temperature, (b) Relative Humidity (RH), (c) Refractivity (M), (d) Wind Speed and (e) Wind Direction at San Clemente Island on 4 November 2019 during two profiles at 2311 UTC and 2328 UTC [7]. Note the slope of trace (c) which indicates the presence of ducting as well as temporal variation in the refractivity profile.</td>
<td>10</td>
</tr>
<tr>
<td>1.10</td>
<td>3D Robotics Solo UAV with iMet weather sensor [6].</td>
<td>11</td>
</tr>
<tr>
<td>1.11</td>
<td>Visual representation of gains and losses in a communication link, along with power levels [8]. If TX power and both antenna gains are held constant, then then the propagation/path loss can be measured via the RX power.</td>
<td>13</td>
</tr>
</tbody>
</table>
1.12 Experimental setup of CASPER East platforms and X-Band propagation measurement links, including a Controlled Towed Vehicle (CTV), R/V Sharp, R/V Atlantic Explorer, and the US Army Corps of Engineers Field Research Facility (FRF) pier in Duck, NC, USA [5].

1.13 Picture detailing the horn antenna and receiver configuration of the XBBR by J. Pozderac at the SIO pier [1].

1.14 Cross-correlation result from an ideal communications channel.

1.15 Multipath communications channel [9].

1.16 Time domain received signal in a multipath channel [10].

1.17 Cross-correlation result from an ideal multipath communications channel.

1.18 Simplistic model of a duct based on ray-tracing philosophy.

1.19 “Best case” approximation of the first bounce off a duct.

1.20 “Worst case” approximation of the first bounce off a duct.

1.21 Block diagram of the linear sequence generator [11].

1.22 Autocorrelation function of an m-sequence as detailed in [12].

Figure Page

2.1 System block diagram showing locations of PN transmitter and receiver separated by a large distance.

2.2 PN Transmitter block diagram.

2.3 PN Transmitter program flow diagram.

2.4 Inside of the PN transmitter.

2.5 Outside of the PN Transmitter showing TX and GPS antennas.

2.6 Azimuth antenna pattern of the PN transmit antenna [13].

2.7 Elevation antenna pattern of the PN transmit antenna [13].

2.8 Logic analyzer capture showing PPS and phase shifter control signals starting on the PPS rising edge and repeating every second for a 32767-length m-sequence.
2.9 Logic analyzer capture showing time to first “1-0-1” bit sequence.  

35

2.10 Simulated PN sequence symbols showing time to first “1-0-1” bit sequence.  

35

2.11 RX System block diagram.  

37

2.12 USRP-X310 Software Defined Radio [14].  

38

2.13 Horn and LNB used to receive the X-band signal.  

39

2.14 PN Receiver before installation in the rack case.  

39

2.15 PN Receiver after installation in the rack case.  

40

2.16 PN Receiver program flow diagram.  

41

3.1 A short portion of the PN sequence showing the first 50 µs of the sequence.  

44

3.2 PN sequence symbol transitions showing (a) rising edge with no jitter, (b) rising edge with jitter, (c) falling edge with no jitter, and (d) falling edge with jitter.  

45

3.3 PSD Plot showing power spectrum of the 1 GHz carrier, modulated carrier, and modulated carrier with jitter added before transmission through the channel, centered at 1 GHz with a 10 MHz span. The peak power of the carrier is approximately 35 dB greater than the spread spectrum signal. The jitter does not affect the “main lobe” of the signal.  

47

3.4 BPSK I/Q plot showing ideal location of “0” and “1” symbols on the complex plane.  

48

3.5 RX I/Q data for the first 100 µs of the demodulated signal.  

50

3.6 Cross-correlation result from the non-delayed signal.  

51

3.7 RX I/Q data for first 100 µs of the simulated multipath channel with a secondary path delay of 20 µs and 25% amplitude of the LOS path. Here, the delayed signal can be clearly seen, but that is not the case for real-world links at far distances, where the SNR is much lower (or even below 0 dB).  

53

3.8 Cross-correlation result for the simulated multipath channel with a secondary path delay of 20 µs and 25% amplitude of the LOS path in the secondary path.  

54
3.9 Cross-correlation result for the simulated multipath channel, with a 100 Hz frequency offset. .................................................. 56

4.1 Test setup for the fixed path test with a path length of approximately 560 meters. Inset left: receive horn and LNB located at the pier. Inset right: transmitter box stationed next to Isaacs Hall. .......... 58

4.2 Raw I/Q data of the received signal from the fixed path test. ...... 59

4.3 Raw I/Q data of the first 250 $\mu$s of the PN sequence from the fixed path test. .......................................................... 60

4.4 Zoomed in spectrum plot of CW portion of signal during the fixed test, showing peak at 0 Hz. ................................................. 61

4.5 I/Q constellation plot of PN sequence before phase correction. Note the skew of the symbols, indicating I/Q imbalance. ............... 62

4.6 Histogram of phase values of the PN sequence before phase correction. 63

4.7 Histogram of phase values after phase correction. ................. 64

4.8 I/Q constellation plot of PN sequence after phase correction. .... 64

4.9 The first 250 $\mu$s of I/Q data of the PN sequence after phase correction. 65

4.10 Cross-correlation of 560 meter fixed-path link. ....................... 66

4.11 One of the cross-correlations of 560 meter fixed-path link, zoomed-in on the main peak. The peak width is approximately 2.5 $\mu$s. .... 67

4.12 RX horn and LNB mounted on the tower facing westward for the sea test. ................................................................. 68

4.13 Transmitter mounted on the boat. Note: this picture was taken with the bow of the boat in front of the photographer, indicating that the transmit antenna was facing in the aft direction. Photo by Nick Rowlett of UCSD. ................................................ 69

4.14 Experimental setup of the sea test. Foreground: RX Horn. Background: Boat with transmitter. ............................................. 70

4.15 GPS track of boat path off the coast of La Jolla, California during the sea test of July 29, 2022. ............................................. 71

4.16 GPS track of boat trying to fight north/south currents to maintain position around the waypoint (shown here at 9 km). .......... 71
4.17 iMet-XQ2 sensor used for the UAV measurements in this thesis [6].

4.18 DJI Inspire 2 UAV with iMet sensor attached. Note location of sensor element in relation to propellers.

4.19 Refractivity vertical profiles from the iMet sensor of the July 2022 sea test. Times are in UTC.

4.20 Cross-correlations from the sea test, d = 31 m, d = 1.06 km, and d = 4.8 km.

4.21 Plot of actual correlation peak time delays, expected peak delays (based only on time of flight from distance), and total TX-RX delays for every trial in the sea test.

4.22 Plot of CNR with respect to distance.

4.23 Cross-correlation result from distance = 18 km showing the duration and no “flattening” of the main peak.

4.24 Path for the Cardiff fixed path measurements.

4.25 RX horn pointed towards Cardiff.

4.26 Transmitter mounted on the railing.

4.27 Transmitter facing towards the SIO pier.

4.28 Refractivity profiles captured at the SIO pier during the Cardiff test.

4.29 Cross-correlation of trial 15 of 47 from the Cardiff fixed path test, distance = 17.1 km, with the main correlation peak at a delay of 71.31 µs and a secondary peak of 72.93 µs. The expected delay for the main peak is 71 µs.

5.1 Measurement of the multipath delay from the cross-correlation of figure 4.29. The delay behind the main peak was measured to be approximately 1.6 µs.

5.2 Zoomed in view of the main peak of the correlation from trial 36 of 47 in the Cardiff test.
Chapter 1

INTRODUCTION

1.1 Motivation

Atmospheric ducting can offer greatly extended communications ranges for maritime communications links. Under the right conditions, transmissions which would normally only propagate in a straight line can propagate beyond line of sight (BLOS). Understanding when ducting is occurring is useful for communications systems operators as it can guide operational decisions (such as choosing a different frequency for communication when ducting is causing poor link performance). While a trained radio operator may be able to determine by ear the effects of multipath in an analog communications link, a digital link may not necessarily have insight into whether or not errors in received data are from multipath effects without time-domain inspection of received signals.

A method for knowing when ducting is occurring is also useful when reducing the electromagnetic signature associated with a transmitter is desired (for example, turning down transmitter power when ducting has been observed to reduce possible interference to other stations). This may be of concern to radio station operators, who are often compelled by regulatory entities to reduce unnecessary interference to other stations.

Additionally, unwanted multipath effects can reduce the performance of radar systems and create false targets [3]. To understand the phenomena of ducting, it is first important to understand electromagnetic (EM) wave propagation and its characteristics in atmosphere.
1.2 Atmospheric Refraction

According to Snell’s Law (Equation 1.1), EM waves bend when they enter a medium of differing refractive index (velocity of propagation). The resulting bend direction ($\theta_2$) is a function of the entrance angle ($\theta_1$) and the respective refractive indices ($n_1, n_2$) of the entrance and exit media.

$$\sin(\theta_1)n_1 = \sin(\theta_2)n_2 \quad (1.1)$$

Figure 1.1: Bending of EM wave according to Snell’s Law [1].
Even small variations in the refractive index over a gradient can have a significant impact on EM wave propagation. Figure 1.2 shows an example of this\textsuperscript{1}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.2.png}
\caption{Example bending of EM wave across a gradient resulting from a stack of thin slabs with homogeneous refractive index [1].}
\end{figure}

Near the ocean surface, the rapid evaporation of water creates a region of high humidity which tapers off as altitude increases. Although the refractive index of air is very close to 1 (the refractive index of free space), the air in this gradient varies slightly, with values starting around $n = 1.0003$ near the surface [15]. Even small changes in $n$ have a large impact on the propagation of radio waves over typical ocean communication link distances (tens to hundreds of km). To simplify the math when changes in $n$ are so small, Refractivity, $N$, is often used as a proxy for the refractive index:

\begin{equation}
N = (n - 1) \times 10^6,
\end{equation}

where $n$ is the index of refraction at any particular point in the atmosphere [16]. It should be noted that like the refractive index, refractivity is unitless, although values are often given in “N-units”.

\textsuperscript{1}Here, $n_1 > n_2 > n_3 > n_4 > n_5$. Note the direction of ray bending with respect to $n$. 

3
Refractivity in atmosphere can be calculated at any point as a function of empirical measurements of barometric pressure, water vapor pressure, and temperature:

\[ N = \frac{77.6}{T} \left( p + 4810 \frac{e}{T} \right), \]  

(1.3)

where \( T \) is the air temperature (K), \( p \) is the partial atmospheric pressure (hPa), and \( e \) is water vapor pressure (hPa) [15]. This equation was empirically derived from dielectric constant measurements and is considered valid for radio frequencies between 1 and 100 GHz [3]. The equation is not frequency-dependent.

Over typical ocean communications links, the distance between transmitter and receiver can be on the order of tens to hundreds of kilometers. Given that most transmitters and receivers in these links are shipboard, the transmitter and receiver heights are no more than a few hundred feet above the surface at most. This means that communication is likely happening beyond visual line of sight and the curvature of the earth must be taken into account to understand how the radio wave can bend beyond visual line of sight. In order to account for this, the Modified Refractivity, \( M \), is used to account for changes in \( N \) with height in a curved-earth representation:

\[ M = N + \frac{z}{a} \times 10^6 \approx N + 0.157z, \]  

(1.4)

where \( N \) is the refractivity, \( z \) is the height above the Earth’s surface in meters where \( N \) was measured, and \( a \) is the radius of the earth in kilometers [17].

In a “standard atmosphere” (no ducting feature present), EM waves above 1-2 GHz will experience “normal” or “standard” refraction; that is to say, the refractive index, and thus \( M \), linearly increases with altitude and there is little to no bending of the

\[ ^2 \text{For brevity, the derivation of equation 1.4 is not shown here, but readers interested in its derivation should consult Chapter 3 of [17].} \]
EM wave beyond visual horizon. Figure 1.3 shows a picture of standard atmospheric conditions assumed by propagation modelers. Such a model does not account for water vapor, which has substantial effects on RF propagation [2]. Figure 1.4 shows the more realistic atmospheric conditions present near the surface. It can be seen that instead of a constant linear decrease in temperature, in reality, there can be layers of warm air sitting on top of cold air. This is called a temperature inversion and is one of the main contributors to strong ducting behavior [5]. Most of this ducting happens in what is called the Marine Atmospheric Boundary Layer (MABL), which is the lowest region of the atmosphere that is in contact with the ocean surface.

Figure 1.3: Assumptions of “standard atmosphere” model [2].
Figure 1.4: Real atmospheric conditions near the ocean surface when there is a temperature inversion present [2]. The temperature inversion shown here creates a trapping layer that can trap EM waves, starting near the surface and spanning the MABL.

If weather conditions (such as layers of warm air sitting on top of cold water) permit, the refractivity can differ greatly as altitude increases, and even have inflection points. These inflection points are where “trapping” layers and ducts are said to form. Figure 1.5 shows refractivity profiles for various ducting conditions. The evaporation duct (f) is the most common type of surface duct, as it often results from the large humidity gradient immediately above the sea surface and is almost always present [3]. However, other types of ducts can occur depending on the weather conditions.

Figures 1.7 and 1.6 show how the values of $\frac{dN}{dz}$ and $\frac{dM}{dz}$ result in bending of EM waves. The discrepancy in $M$ between inflection points as shown in figure 1.7 determines the strength of the duct (higher strength = more reflection of EM energy).
Figure 1.5: Plots of modified refractivity versus altitude for different refractive conditions: (a) sub-refractive layer denoted by dashed line; (b) normal refraction; (c) elevated duct denoted by dashed line; (d) surface duct denoted by dashed line; (e) surface duct (dashed line) due to elevated region of strongly negative vertical refractivity gradient; (f) evaporation duct denoted by dashed line [3].

Figure 1.6: Effect of radio wave “bending” due to inflection points in the refractivity profile [4].
Figure 1.7: Refractivity profiles for various ducting phenomena. Profiles shown: a) evaporation duct, b) surface-based duct, and c) elevated duct [5].
1.3 An Examination of Methods of Ducting Observation

1.3.1 Refractivity Measurement from Direct Sampling of Environment

Weather measurements for computing refractivity profiles are conventionally made via radiosonde by inflating a balloon or launching a rocket attached to said radiosonde. Radiosondes are considered the “gold standard” of refractivity estimation. A radiosonde typically contains sensors for measuring temperature, pressure, and relative humidity alongside a GPS receiver to log position and time of samples, as well as a radio transmitter which reports sensor readings in real-time.

![InterMet iMet-54 Research Radiosonde](image1)

![A balloon for hosting a radiosonde](image2)

Figure 1.8: Left: InterMet iMet-54 Research Radiosonde. Right: A balloon for hosting a radiosonde [6].

Figure 1.9 shows an example of weather measurements made via radiosonde near San Clemente Island in southern California in November 2019. Note that the refractivity profile (c) shows what appears to be a surface duct, which occurred at approximately 150 meters altitude. Surface ducts differ from evaporation ducts in that they generally appear at much higher altitudes (hundreds to thousands of meters) and can be hard to predict from atmospheric models.
Figure 1.9: Balloon radiosonde observations of (a) Temperature, (b) Relative Humidity (RH), (c) Refractivity (M), (d) Wind Speed and (e) Wind Direction at San Clemente Island on 4 November 2019 during two profiles at 2311 UTC and 2328 UTC [7]. Note the slope of trace (c) which indicates the presence of ducting as well as temporal variation in the refractivity profile.

Recent advances in technology have allowed unmanned aerial vehicles (UAVs) with a weather sensor package to be used to perform on-demand collection of weather measurements. UAVs are very convenient for measuring weather parameters because they are low-cost (compared to a radiosonde, rocket, or helicopter-based measurement system) and allow rapid vertical profile collection. Figure 1.10 shows a 3D Robotics Solo UAV with a meteorological sensor package attached. UAV operation requires careful consideration of the turbulence in air flow from the propellers as it will disturb the meteorological measurements. Additionally, refractivity profile data near the ocean surface can only be gathered down to a minimum safe UAV operation altitude (depending on the sea state and pilot comfort), leaving gaps in profile data from the surface up to a few meters in altitude. This was explored by [7] during the measurement campaign at San Clemente Island.
Figure 1.10: 3D Robotics Solo UAV with iMet weather sensor [6].

Further examination of the UAV measurement for providing insight into the measurement made via radio is discussed in Section 4.2.2.
1.3.2 Propagation Loss (PL) Method

A wireless communications link can be modeled at the simplest level as a transmitter and receiver separated by some distance, each with antenna of varying gain. In an ideal LOS communications link, the relationship between the power of transmitted and received signals follows the following equation:

\[ P_r = \left( \frac{\lambda}{4\pi R} \right)^2 G_t G_r P_t \]  \hspace{1cm} (1.5)

where \( P_r \) is the received power out of the antenna, \( R \) is the distance between transmit and receive antennas, \( \lambda \) is the wavelength, \( G_r \) is the receive antenna gain, \( G_t \) is the transmit antenna gain, and \( P_t \) is the transmit power into the antenna [18]. In reality, there are inefficiencies in antennas, losses in receive/transmit chain devices, and attenuation in the channel which make this equation not 100% accurate for all cases, but it is generally accepted for calculating received power in air when propagation is LOS. If the parameters of equation 1.5 are known save for the received power, then the expected received power can be computed. Figure 1.11 shows a visual representation of the gains and losses in equation 1.5.
Figure 1.11: Visual representation of gains and losses in a communication link, along with power levels [8]. If TX power and both antenna gains are held constant, then the propagation/path loss can be measured via the RX power.

The work of J. Pozderac of The Ohio State University showed that with a vertical array of X-band transmitters and receivers, if the received power is much higher than expected for any given LOS path (of course, with appropriate knowledge of the transmit/receive gains and losses, as well as distance), then the presence of a duct can be inferred heuristically [1]. Measurements of sea level pressure, temperature, and humidity are made to compare the expected duct height (EDH) against the observed value from radio measurement. This was validated in the Coupled Air-Sea Processes and Electromagnetic Ducting Research (CASPER) campaign, a measurement project to explore the effects on environmental conditions in the MABL on low-altitude EM propagation [5]. Figure 1.12 shows the CASPER East experimental setup, with beacons placed at different heights.

---

[3] Some authors refer to this process as “Refractivity Inversion”.

13
Figure 1.12: Experimental setup of CASPER East platforms and X-Band propagation measurement links, including a Controlled Towed Vehicle (CTV), R/V Sharp, R/V Atlantic Explorer, and the US Army Corps of Engineers Field Research Facility (FRF) pier in Duck, NC, USA [5].

The so-called X-Band Beacon Receiver (XBBR) is capable of estimating duct height when multiple beacons were used, but is sensitive to the fact that variations in propagation loss could be from factors other than ducting [1],[19]. The system requires extensive and detailed calibration of the gains and losses of all RF elements in the system. Several receive antennas were switched through the receive path in the process of capturing data. Figure 1.13 shows the XBBR configuration during a measurement campaign at the SIO Pier. Multiple horn antennas are mounted at different heights and their height is recorded using a laser height recorder.
Figure 1.13: Picture detailing the horn antenna and receiver configuration of the XBBR by J. Pozderac at the SIO pier [1].
1.4 Ducting Estimation From Multipath Delay Measurement

The method investigated in this work uses a time delay measurement to measure multipath effects directly. This method differs from the PL method in that it is in theory insensitive to variations in signal power, as long as the signal can be received with adequate signal to noise ratio (SNR).

In an ideal communications channel, there is a single copy of the transmitted signal that is received, as shown in Figure 1.14. Performing a cross-correlation of the TX and RX signals (represented by the value $R_{TX-RX}$) will give a peak at the time delay of the received signal (provided the receiver starts sampling when the transmitter sends a signal; some synchronization method is required).

![Figure 1.14: Cross-correlation result from an ideal communications channel.](image)
In reality, reflections off of objects and the terrain in the environment in which a signal travels can produce multipath effects. A multipath channel can be represented as the sum of reflected paths and (if applicable) a line of sight path, such as in Figure 1.15:

![Multipath communications channel](image)

**Figure 1.15: Multipath communications channel [9].**

This will mean that multiple copies of the signal are received in quick succession, as shown in Figure 1.16:

![Time domain received signal in a multipath channel](image)

**Figure 1.16: Time domain received signal in a multipath channel [10].**

In theory, if the transmitted signal through a multipath communications channel is a known and unique waveform, the TX-RX cross-correlation should show multiple peaks occurring at the successive arrivals of the transmitted signal at the receiver. Figure 1.17 shows an example of this. A fast sampling interval on the receiver is also required in order to see multipath effects that occur on a similar time scale as (or less than!) the time between symbols of the transmitted signal. This can indicate
a channel’s propensity for generating inter-symbol interference (ISI), which occurs when received symbols interfere with subsequent symbols. Being able to view cross-correlations at such precise intervals is important for designing digital communication systems that are resistant to this distortion in the channel. The desired sample rate of the receiver is then dependent on the symbol rate of the transmitted signal.

Figure 1.17: Cross-correlation result from an ideal multipath communications channel.
1.5 Modeling Delay Through a Duct

The correlation behavior shown in the previous section is a good example of discrete multipath delays, but does not cover the phenomenon of continuous reflections, as shown in figure 1.18. Continuous reflections “shape” the main peak to be more rounded than that of figure 1.17. In order to understand the potential multipath delays observed under ducting conditions, a simple model was constructed. In this model, the transmitted signal is modeled as a ray. The primary signal path is line of sight, and the secondary arrival is assumed to only make one bounce off a duct. The expected delay can be found given the geometry of the duct (i.e. height and TX-RX distance).

![Simplistic model of a duct based on ray-tracing philosophy.](image)

Figure 1.18: Simplistic model of a duct based on ray-tracing philosophy.

In the “best case”, the ray reflects off the duct with total internal reflection, and the angle normal to the duct as the ray bounces is the same as when it is leaving. Figure 1.19 shows this geometry of a duct with height $h$ and a TX-RX distance $d$ in the “best case” for a bounce.
Figure 1.19: “Best case” approximation of the first bounce off a duct.

The distance traveled by the primary path is simply $d$, but for the secondary path, the distance traveled $d_{\text{prop}}$ is dependent on $d_1$ and $d_2$:

$$d_{\text{prop}} = 2 \sqrt{h^2 + \left(\frac{d}{2}\right)^2}$$  \hspace{1cm} (1.6)

In the “worst case” for the single bounce, the ray travels almost vertical, and any energy reflected into the duct reflects at an acute angle. This is satisfied when $d_1$ or $d_2$ is maximized, i.e. $d_1$ or $d_2 = h$. This geometry is shown in figure 1.20:
Figure 1.20: “Worst case” approximation of the first bounce off a duct.

The distance traveled is then by the ray is then:

\[ d_{\text{prop}} = \sqrt{h^2 + d^2} + h \]  \hspace{1cm} (1.7)

Further computation reveals that the case where \( d_1 = d_2 \) minimizes the travel distance (and thus travel time) of the bounce ray.

Depending on the transmit baud rate and the receive sample rate, delays that are within the width of the main arrival correlation peak may not be observable as distinct peaks. This constrains the ability of the system to measure delays when duct heights and/or TX-RX distance creates a delay that is not measurable. Nonetheless, this estimation of multipath delays was used to interpret experimental data collected and the analysis is presented in the conclusion of this thesis.
1.6 Explanation of Pseudorandom Sequence Choice

To make the transmitted signal unique enough to have as large as possible of correlation coefficient in the presence of noise and interfering signals, a pseudo-random (PN) data chirp is to be used. The unique waveform is also necessary to distinguish the signal from other signals that are being received (for example, a CW carrier or sinusoidal pulse would not be a good signal, because it is very much like spurious noise emissions from other transmitters). This is in contrast to the propagation loss measurements made in the experiments of [1], where the transmitted signal is simply a CW signal whose amplitude is measured. In theory, the PN signal method is more robust than a CW transmitter because large correlation coefficients can occur even when the received signal is below the noise floor, allowing measurement distances much farther than a CW-only system.

The PN signal selected to be used as a modulating signal is a binary sequence called a maximum length sequence (hereafter called m-sequence), which has special properties that make it desirable for use as a TX signal [12],[20]. First, it has a auto-correlation maximum only when the lag between the TX and RX is zero (and nearly zero elsewhere; see equation 1.8), and second, it contains an almost equal distribution of 0s and 1s, which minimizes any average value on the carrier signal [11].

A m-sequence is constructed from a primitive polynomial which defines a multi-stage feedback shift-register generator. The “maximum” part of “maximum length sequence” comes from the fact that they are the longest possible sequences to generate using a shift register of a given length [11]. The input to the shift register is the number of stages, \( m \), and the output sequence of is length \( L = 2^m - 1 \). Each m-sequence is a deterministic binary sequence that is solely dependent on the length of the sequence. Figure 1.21 shows a diagram of the shift register.
Figure 1.21: Block diagram of the linear sequence generator [11].

The auto-correlation function calculates the “likeness” of a signal to a time-shifted version of itself. The auto-correlation function $C(\tau)$ of a m-sequence is periodic with the period $T_p = LT_d$ and given by the following:

$$C(\tau) = \begin{cases} 
1 - \frac{1+L}{L} \frac{\tau}{T_d}, & \text{where } |\tau| \leq T_d, \\
-\frac{1}{L} & \text{otherwise.}
\end{cases} \quad (1.8)$$

where $\tau$ is the time shift between the m-sequence and the time-shifted version of itself, $L$ is the sequence length in symbols and $T_d$ is the symbol duration. The auto-correlation peak has a linear rise and fall with almost no constant level in the correlation peak [12]. Although the presence of a auto-correlation peak at $\tau = 0$ is self-evident, the peak corresponding to a signal arrival will always be a minimum of 2 symbols in duration (between $-T_d$ and $T_d$), as shown by figure 1.22. Although sampling at the symbol rate of the transmitted signal can show multipath effects, to observe faster arrivals and provide more resolution, the receiver should be oversampling the transmitted signal.
Figure 1.22: Autocorrelation function of an m-sequence as detailed in [12].

An algorithm for generating the m-sequence can be found in [21].
1.7 Project History and My Contribution

The work in this thesis builds on the thesis work of several SIO staff and graduate students before me. The CORDC effort to measure ducting via an array of receivers began circa 2015, with the first experiments using Pozderac’s setup as described in section 1.3.2. After several field-deployments of this system over several years, the time-delay method for improving the observation of ducting behavior had been hypothesized, and development work on the new system had been started by Dylan Shields of CORDC in 2019. Dylan’s work eventually led to the current system design that is discussed in (and was modified for the purposes of) this thesis.

My personal contributions to this effort were principally the integration of receiver with a 200 MSps sample rate (in place of the 15 MSps receiver first utilized by Pozderac for his propagation loss experiments, and then by Dylan Shields in 2018 for the time-delay experiments) as shown in figure 2.11, and several field measurement campaigns to evaluate the impact of a higher-sample rate receiver on channel measurements. The integration of the receiver is discussed in section 2.3, and the measurement campaigns are discussed in chapter 4. Integrating the receiver consisted of writing interface code in C++ (program flow diagram is shown in figure 2.16) to interface with the software-defined radio receiver directly using its native application programming interface (API). The C++ code for retrieving samples from the SDR is listed in appendix A. The MATLAB code for post-processing received samples is listed in appendix B.
1.8 Content of This Thesis

The content of this thesis was to investigate how a channel sounding using an X-band transmitter and a receiver with a high sample rate can be used to observe time-domain multipath effects caused by atmospheric ducts on a time scale of one symbol period (or as close to it as possible). Since the typical marine communications link can span tens to hundreds of kilometers, the system was designed with the goal that the transmitted signal is able to be received with a noticeable cross-correlation coefficient out to 100 km (beyond LOS for a shipboard link)\(^4\). The stated design goals are tabulated in table 1.1.

<table>
<thead>
<tr>
<th>Table 1.1: Desired system specifications/design goals.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of Operation</td>
</tr>
<tr>
<td>Sampling rate</td>
</tr>
<tr>
<td>Maximum TX-RX distance</td>
</tr>
</tbody>
</table>

Transmitter and receiver system design with simulations of cross-correlation results, results from field measurement campaigns, and ideas for future work/system improvements are presented in the following chapters.

\(^4\)At those distances, propagation is likely beyond LOS and subject to refractive conditions, so it is not possible to use typical link budget equations (such as equation 1.5) to find the expected signal strength at that distance. It is stated as a rather generous goal and is something that performance is best determined by real world testing.
Chapter 2

SYSTEM DESIGN AND CONSTRUCTION

2.1 System Overview

The transmit-receive system consists of a single X-band PN transmitter (located somewhere on the ocean surface—typically shipboard) and corresponding receiver (located on shore) as shown in Figure 2.1.

![System block diagram showing locations of PN transmitter and receiver separated by a large distance.](image)

Figure 2.1: System block diagram showing locations of PN transmitter and receiver separated by a large distance.

To satisfy the design goals stated in chapter 1, the first task was to choose a system frequency in the range of X-band (8-12 GHz). The center frequency was chosen to be around 11 GHz, which is allocated in the United States to satellite/terrestrial-satellite communications [22], and thus a plethora of transmit and receive equipment (oscillators, antennas, downconverters, etc) is readily available for purchase.

In order to send the binary sequence over the air, the 11 GHz carrier must be modulated. An easy way to modulate the symbols of a binary sequence is through the use of Binary Phase Shift-Keying (BPSK) modulation. BPSK maps symbols to carrier phase values.
The transmitted signal can be modeled as a sinusoid:

\[ s(t) = A\cos(2\pi f_c t + \theta) \]  

(2.1)

where \( s(t) \) is the transmitted signal, \( A \) is the amplitude, \( f_c \) is the carrier frequency in Hz, \( t \) is time in seconds, and \( \theta \) is the phase angle of the carrier. The phase angle is modified according to the following symbols for a BPSK modulation:

- Binary 0 \( \rightarrow +0^\circ \) phase shift
- Binary 1 \( \rightarrow +180^\circ \) phase shift

Since \( A\cos(2\pi f_c t + 180^\circ) \) is the same as \(-A\cos(2\pi f_c t)\) (by trigonometric identity), the transmitted symbol of the BPSK waveform can be represented by the sign of \( A \):

- Binary 0 \( \rightarrow s(t) = A\cos(2\pi f_c t + 0^\circ) = A\cos(2\pi f_c t) \)
- Binary 1 \( \rightarrow s(t) = A\cos(2\pi f_c t + 180^\circ) = -A\cos(2\pi f_c t) \)
2.2 PN Transmitter Design

The transmitter was designed by a UC San Diego Student and former SIO Staff Member, Dylan Shields, and consists primarily of an 11 GHz oscillator whose output phase is controlled by a single board computer (SBC) to perform the BPSK modulation. The SBC was designed in-house at the Coastal Observing Research and Development Center (CORDC) at SIO and contains high-speed general purpose input/output (GPIO) lines. The block diagram of the transmitter is shown in figure 2.2.

![PN Transmitter block diagram](image)

**Figure 2.2: PN Transmitter block diagram.**

The SBC’s GPIO lines are connected to a RF Lambda RFPSHT0812N6 monolithic RF phase shifter. The SBC modulates the +0/+180 degrees of RF phase on the carrier according to the symbols of the PN sequence. The chirp rate is 1Mbaud (1 µs per symbol) and the chirp length is approximately 32.767 µs long (corresponding to one burst of a 32767-length m-sequence). This chirp rate was selected to be similar to
industry standard chip rates in the satellite and cellular telecommunications industries for code-division multiple access (CDMA) schemes.

The DLCRO X-band oscillator is frequency locked using the 10 MHz reference from a Symmetricom GPS-2750 GPS Disciplined Oscillator (GPSDO). This GPSDO also has a 1 Hz Pulse Per Second (PPS) output. The outputting of the BPSK chirp signal occurs on the rising edge of the PPS signal. Upon power up, the transmitter looks for GPS lock, and once the 10 MHz reference has been adequately stabilized, the SBC starts outputting the signals to modulate the phase shifter and repeats this process every second. The reason for choosing a transmit interval of one second is that the PPS is a convenient way of signalling the start of transmitting that not does not require any coordination between the transmitter and receiver. The TX program flow diagram is shown in figure 2.3.

![PN Transmitter program flow diagram.](image)

The transmitter is contained in a weather-resistant case, and all components in the transmitter have been selected to be powered via 12V DC so that the transmitter can be run directly from a 12V battery, reducing interference from noise caused by inverters and/or switching power supplies.
Figure 2.4: Inside of the PN transmitter.
The transmit power is approximately 28 dBm and the gain of the transmit antenna (UB Corp Model AO85106BS) is approximately -20 dBi in the boresight direction.

![Image](image.jpg)

**Figure 2.5:** Outside of the PN Transmitter showing TX and GPS antennas.

Since the time-delay measurement system is correlation-based, the fluctuation of received signal (due to slight directionality in the antenna pattern) by a few dB in either direction is not too important, but the location of the transmit antenna (on the side of the enclosure) was chosen to maximize signal while at the same time being convenient for mounting the enclosure on a boat. It was decided that mounting the transmitter enclosure with its longest dimension lying flat on the boat’s roof was the best option that also allows for ventilation and minimizes attenuation of the transmitted signal. The antenna patterns of the TX antenna are shown in figures 2.6 and 2.7.
Figure 2.6: Azimuth antenna pattern of the PN transmit antenna [13].

Figure 2.7: Elevation antenna pattern of the PN transmit antenna [13].
Figure 2.8 shows the PPS input and PN sequence output of the SBC captured by a Saleae Logic Analyzer.

![Logic analyzer capture showing PPS and phase shifter control signals starting on the PPS rising edge and repeating every second for a 32767-length m-sequence.](image)

Figure 2.8: Logic analyzer capture showing PPS and phase shifter control signals starting on the PPS rising edge and repeating every second for a 32767-length m-sequence.

Although it cannot be seen from figure 2.8, there is a delay inherent to the microcontroller executing instructions (sometimes called “pipeline delay”) upon receiving input to an interrupt line, and therefore there will be a delay between the PPS input being received and the PN sequence actually being transmitted. The pipeline delay was estimated by finding when the bitstream was identical to the m-sequence and comparing the times. Figure 2.9 shows that the time to get to the first noticeable “1-0-1” sequence was approximately 54 $\mu s$. Figure 2.10 shows that the time for those bits to show up (if there were no delay) would be approximately 42 $\mu s$. Therefore, the estimate of the pipeline delay is approximately 12 $\mu s$. This estimate of the pipeline delay will be used to help understand the correlation performed on the results of testing in Chapter 4.
Figure 2.9: Logic analyzer capture showing time to first “1-0-1” bit sequence.

Figure 2.10: Simulated PN sequence symbols showing time to first “1-0-1” bit sequence.
The pipeline delay was then measured several times using the highest sample rate available on the Logic Analyzer (500 Megasamples/second). The results of the TX pipeline delay investigation are tabulated in table 2.1.

Table 2.1: Statistical analysis of TX pipeline delay.

<table>
<thead>
<tr>
<th>Number of trials</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>12.02 µs</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>99.55 ns</td>
</tr>
</tbody>
</table>
2.3 PN Receiver Design

The PN receiver principally consists of a software-defined radio (SDR) receiver and a downconverter to convert the 11 GHz signal to a frequency that the SDR can tune to. The receiver equipment was installed in a shed at the SIO pier circa June 2022. Figure 2.11 shows the block diagram of the receiver.

![RX System block diagram.](image)

As discussed in section 1.8, the higher the sample rate, the higher the temporal resolution of viewing signal arrivals. A 200 Megasamples/second (MSps) sample rate was decided as being a good trade off between the sampling interval (200 received samples per transmitted symbol), burden of sample storage, cost, and SDR availability (due to supply chain shortages at time of constructing the receiver).

A Universal Software Radio Peripheral (USRP) X310 SDR with the UBX160 daughterboard (10 MHz - 6 GHz frequency range, 200 Ms/s max sample rate) designed by Ettus Research (now a subsidiary of National Instruments) was selected as the receiver because of its high sample rate and integrated RF stage.

The SDR was connected to a host PC via a 10 Gigabit Ethernet interface through an Intel X520-DA2 Network Interface Card (NIC). The host personal computer (PC)
platform has a 8-core 64-bit processor with 32 Gigabytes of memory. The receiver system’s GPSDO’s 1 PPS signal is connected to the PPS trigger input of the SDR. A Network Attached Storage (NAS) device was networked with the PC to provide overflow storage for the data generated during a measurement campaign.

A NorSat 1007XHCN Low-Noise Block Downconverter (LNB) was used with a Paster-nack PE9855-20 X-band receive horn antenna to downconvert the 11 GHz signal to something which the USRP could receive as shown in figure 2.13. The downconverter has an internal 10 GHz Local Oscillator (LO) and mixer. The LNB requires a 10 MHz reference signal and power which is delivered via a bias tee. The LNB is powered at a voltage of 24 Volts DC from an external power supply. The transmitted X-band signal is around 11 GHz, so the intermediate frequency (IF) output from the LNB is centered around 11 GHz - 10 GHz = 1 GHz. The 1 GHz signal is then brought into the USRP. The LNB and the USRP are both frequency synchronized via a 10 MHz sine wave from the GPSDO. The GPSDO was set to “survey-in” mode which provides a more stable 10 MHz reference signal (for a stationary receiver) than if the GPSDO was left in “mobile” mode. The GPSDO antenna was mounted outdoors on the roof of the pier shed.

Figure 2.14 shows the SDR hardware before integration into a rack-mount case.
In the following months after the receiver was first deployed, it was then moved into a rack-mount case\(^1\) for organization and ease of transportation to future receive sites. Figure 2.15 shows the PN receiver after integration into the rack case.

\(^1\)The rack-mount case used was the one previously used by [1] for his XBBR experiments at the SIO Pier as referenced in figure 1.13 (hence the presence of multiple RF inputs on the rack). The GPSDO and server in that rack from those experiments were utilized for use with the PN receiver system.
Figure 2.15: PN Receiver after installation in the rack case.
After installing the Ettus Universal Hardware Driver (UHD) [23], a program was developed in C++ to capture one second’s worth of received samples to a file which is executed whenever a data point is taken. The RX program code is detailed in appendix A. After data capture, the files are offloaded onto the NAS for long-term storage. The received samples are then brought into MATLAB for post-processing. The samples are received at 200 MSps as 16-bit complex integers (16 bits of in-phase (I channel) and 16 bits of in-quadrature (Q channel)) are recorded by the program. The size of received data is then 200 MSps * (16 bits I channel + 16 bits Q channel) * 1 second = 800 Megabytes per data point. Whenever a data point is captured, the timestamp is logged on the receiver (either from a local network time server or the serial output of the GPSDO if network time is not available) for reference against a GPS tracker so that the actual distance between transmitter and receiver (and thus the time of flight) can be found. The RX program flow diagram is detailed in figure 2.16.

![PN Receiver program flow diagram.](image)

As with the PN transmitter, the receiver also has a delay (small, but measurable) between the input of the trigger and the start of sampling on the USRP. The delay was investigated in a similar manner to transmitter by creating a program to measure the delay between the rising edge of the PPS input to the USRP and the toggling of a GPIO line on the USRP (which should be synchronous with the start of sampling). A statistical analysis of the USRP sample latency behind the PPS is detailed in table 2.2.
Table 2.2: Statistical analysis of RX sampling latency.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of trials</td>
<td>53</td>
</tr>
<tr>
<td>Mean</td>
<td>243.28 ns</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>3.21 ns</td>
</tr>
</tbody>
</table>

Since the receiver in the USRP is not a calibrated measurement, this means that instead of amplitude being relative to an absolute measurement (such as in test equipment or other calibrated receivers), the amplitude of the received samples is relative to full scale of the Analog to Digital Converter (ADC). This means that computing the signal to noise ratio (SNR) is possible by measuring the noise level and computing how many decibels (dB) the signal is above the noise, but a received power measurement is not absolute without calibrating the USRP. For the purposes of this thesis, an absolute power measurement was not needed and thus the USRP was not calibrated.
Chapter 3

SYSTEM SIMULATIONS

The expected correlation behavior of the PN system was simulated using MATLAB. This chapter details the methodology used to simulate the system.

The M-sequence algorithm generates symbols of the binary sequence with the coefficient +1 or -1 in the place of 0 or 1 bits. This is used to modulate the sign $A$ (and thus the phase) of the carrier signal as discussed in equation 2.1. Since the 11 GHz signal will eventually be down-converted in the LNB to 1 GHz, 1 GHz was used as the carrier frequency for simulation. This is because the down-conversion is just a shift in carrier frequency and in theory should not affect the baseband signal. Additionally, choosing to simulate at the lower carrier frequency allows for less memory usage to generate the samples since a lower sample rate is needed to represent the signal. In the case of this simulation, the sampling rate was chosen to be 5 Gigasamples per second (GSa/s) which minimizes aliasing of the 1 GHz signal and any significant sidebands due to BPSK modulation. The 5 GSa/s signal is later decimated to 200 MSa/s to match the USRP performance after simulating the downconversion to baseband that occurs on the USRP. The PN sequence (a 32767-length m-sequence) was generated at the sample rate of 5 GSa/s. Figure 3.1 shows a portion of the PN sequence symbols representing the amplitude of the modulating signal.
Figure 3.1: A short portion of the PN sequence showing the first 50 μs of the sequence.
3.1 Phase Shifter Jitter Simulation

The phase shifter in the transmitter has a transition time between symbols that degrades the performance of the transmitter. This figure was given to be 150 ns according to the phase shifter datasheet [24]. This was simulated by adding 200ns worth of noise on every phase transition (rising and falling edge) and applying it to the modulating signal as shown in Figure 3.2.

![Figure 3.2: PN sequence symbol transitions showing (a) rising edge with no jitter, (b) rising edge with jitter, (c) falling edge with no jitter, and (d) falling edge with jitter.](image)

Figure 3.2: PN sequence symbol transitions showing (a) rising edge with no jitter, (b) rising edge with jitter, (c) falling edge with no jitter, and (d) falling edge with jitter.
A 1 GHz carrier was then modulated with the PN sequence. Figure 3.3 shows the Power Spectral Density (PSD) plot of the 1 GHz carrier, carrier modulated with PN sequence, and carrier modulated with the PN sequence with the estimated phase shifter jitter added. The effect of modulating the carrier with the PN sequence is that a “spreading” of the signal across the frequency band occurs. This spreading reduces the peak power of the modulated signal by a certain amount, which here is called the spreading loss. It can be seen from figure 3.3 that the spreading loss is around 35 dB. Even though the peak power is reduced, when the cross-correlation is performed, the processing gain from the correlation is equal to the spreading loss and raises the peak power level to what the CW carrier peak level was.
Figure 3.3: PSD Plot showing power spectrum of the 1 GHz carrier, modulated carrier, and modulated carrier with jitter added before transmission through the channel, centered at 1 GHz with a 10 MHz span. The peak power of the carrier is approximately 35 dB greater than the spread spectrum signal. The jitter does not affect the “main lobe” of the signal.

The LNB contains a 950 MHz - 1.7 GHz band-pass filter (BPF) to remove the high-frequency mixing product. Because the modulated carrier contains most of the signal power within 2 MHz of bandwidth, the BPF in the LNB is far enough away from the center frequency of 1 GHz and has no effect on the IF signal.
After passing through the LNB, the signal needs to be downconverted again to baseband. This step is accomplished by the mixer in the USRP. The mixer in the USRP produces real and imaginary components of the baseband signal. The real and imaginary components are also called the In-Phase (I) and In-Quadrature (Q) channels. Figure 3.4 shows the symbols of a BPSK waveform like the one transmitted by the PN transmitter plotted in the complex plane.

![Figure 3.4: BPSK I/Q plot showing ideal location of “0” and “1” symbols on the complex plane.](image)
The signal was down-converted to baseband and a 100 MHz wide baseband filter was used to remove mixing components (as is physically implemented in the USRP’s UBX-160 daughterboard [25]). The signal was also decimated down to 200 MSa/s to emulate the highest possible sampling rate of the USRP. Although the transmitted signal contains only real components in theory, the addition of noise and local oscillator (LO) leakage from I to Q channels (and vice versa) inside the radio will mean there will be energy present on the Q channel. This is often referred to as I/Q imbalance. In the post-processing for the samples received from the USRP, the energy on the Q channel will be removed to account for this (since transmitted signal should not have any imaginary component). Additionally, the transmitter and receiver LOs are separate (not phase synchronized) and will have some random phase difference between them, which will also be removed in post-processing. Both the I/Q imbalance and phase offset are connected issues and are compensated for by finding the phase offset from 0 degrees (this was accomplished by generating a histogram of the phase values of all samples during the PN sequence to determine the skew) and applying a phase shift in the opposite direction, to bring the skew to as close to 0 degrees as possible.

For the purposes of the simulation, the I/Q imbalance issue was not simulated, but the post-processing steps detailed above were applied in experimental measurement campaigns discussed later in the thesis, in particular, to obtain the result of figure 4.8.

In simulation, a small amount of Additive White Gaussian Noise (AWGN) was added to the modulated carrier to visually show how the signal might look in a real-world receiver. The noise power used was selected to create a SNR of 10 dB. Figure 3.5 shows a portion of the simulated I and Q channels at baseband.
Figure 3.5: RX I/Q data for the first 100 $\mu$s of the demodulated signal.

The RX signal was then cross-correlated with the transmitted symbols of the PN sequence. Figure 3.6 shows the output of the cross-correlation. It can be seen that the correlation has a peak around almost 0 symbols, with relatively small values elsewhere, which tracks with the fact that no delay was applied in the simulation. The correlation peak is also close to 2 symbols wide (2 $\mu$s) in duration, which is as predicted by the auto-correlation function of the m-sequence.
Figure 3.6: Cross-correlation result from the non-delayed signal.
3.2 Emulating Multipath Effects With Time Delay

The effect of a multipath arrival can easily be simulated by adding a delayed and attenuated copy of the received signal to itself. To demonstrate this, the amplitude of the simulated second signal arrival was attenuated to be 25% of the amplitude of the main signal. This is to emulate a potential wave guide mode formed by the duct, which is “leaky” and does not perfectly reflect energy off of the duct. The amount of reflection depends on the strength of the duct and varies greatly with weather conditions. This 25% attenuation factor is not meant to be representative of all ducting conditions (hence the experimental portion of this thesis, which seeks to find those path gains, if applicable), but is presented to show what an attenuated, delayed signal could look like.

The RX I/Q data was input to a fractional delay filter that delays the signal using a Thiran approximation to create a time-delayed version of the signal with 20 µs of delay. The two signals were added to each other to produce the multipath signal. The result is shown in figure 3.7.

The RX correlation was then performed on the delayed signal. Figure 3.8 shows the cross-correlation result from the delayed signal simulation.
Figure 3.7: RX I/Q data for first 100 $\mu$s of the simulated multipath channel with a secondary path delay of 20 $\mu$s and 25% amplitude of the LOS path. Here, the delayed signal can be clearly seen, but that is not the case for real-world links at far distances, where the SNR is much lower (or even below 0 dB).
Figure 3.8: Cross-correlation result for the simulated multipath channel with a secondary path delay of 20 µs and 25% amplitude of the LOS path in the secondary path.
3.3 Effect of Frequency Offset on Received Signal

In reality, the received signal will have a nonzero frequency offset from the transmitter for two reasons:

1. The transmitter and receiver have separate 10 MHz reference oscillators.

2. Motion of the transmitter relative to the receiver will induce a Doppler shift.

This frequency offset will degrade the correlation coefficient. The 10 MHz oscillator offset cannot be planned for, but the Doppler shift can. The Doppler shift associated with any EM wave received from a moving source is expressed by the following:

\[ f = \left( \frac{c \pm v_r}{c \pm v_s} \right) f_0, \quad (3.1) \]

where \( f \) is the observed (received) frequency, \( f_0 \) is the emitted (transmitted) frequency, \( c \) is the velocity of propagation (assumed to be the speed of light), \( v_r \) is the velocity of the receiver, and \( v_s \) is the velocity of the transmitter. The effect of a transmitter moving at approximately 5 knots at 11 GHz was found to produce a frequency shift of approximately 100 Hz. It is important to use the number for frequency shift found at 11 GHz, because the downconversion of the signal means that the Doppler shift at IF will also be 100 Hz, rather than the Doppler shift for a transmitted frequency of 1 GHz (which is 9 Hz). Figure 3.9 shows the cross-correlation result when a 100 Hz frequency shift is applied.
Figure 3.9: Cross-correlation result for the simulated multipath channel, with a 100 Hz frequency offset.

It can be seen that the effect of the frequency shift makes the correlation result very noisy and hard to pick out any correlation peaks. In post-processing of real-world receive data, the frequency offset is found by computing the PSD of the CW portion of the signal and applying a frequency offset in the opposite direction.
Chapter 4

SYSTEM TESTING AND RESULTS

4.1 July 2022 Fixed Path Test

The PN transmitter and receiver was first placed in a fixed location to determine if the PN signal was being received and correlated properly. This was to verify if the signal processing indicates a large correlation peak from the primary received signal (from a line of sight path, with little to no multipath propagation).

The receiver was located on the SIO research pier with its antenna facing towards the Isaacs Hall building at SIO. The transmitter was placed outside on the deck of the building. Figure 4.1 shows the approximate locations and distance between transmitter and receiver.
This distance of approximately 560 meters was found to have a time of flight (TOF) of approximately 186 ns. This means that the total lag (TOF + 12.02 µs of TX pipeline delay + 0.234 µs of RX sampling delay) should be approximately 14.129 µs. The receive post-processing scripts developed and demonstrated in the simulation section were applied to analyze the data captured from the stationary test.

4.1.1 Post-Processing Methodology

First, the received signal was imported into MATLAB. The samples are read in as 16-bit complex integers (values ranging from -32768 to 32767). Figure 4.2 shows the raw I/Q data over the entire second of data capture. The PN sequence data burst can be seen distinctly in the first 32 ms. Figure 4.3 shows a zoomed-in view on the first 250 µs of the PN sequence.
Figure 4.2: Raw I/Q data of the received signal from the fixed path test.
Figure 4.3: Raw I/Q data of the first 250 $\mu$s of the PN sequence from the fixed path test.

In this case, even though it can be seen visually that there is a low-frequency oscillation in envelope of the CW portion (due to Doppler shift and lack of LO synchronization on TX and RX), a Fast Fourier Transform (FFT) was performed to find the spectrum of the CW portion of the signal to compensate for the frequency offset. The spectrum is shown in figure 4.4$^1$.

$^1$The units on the vertical axis of figure 4.4 are given without context because this measurement is for only finding the frequency offset of the carrier and absolute power is not needed.
Figure 4.4: Zoomed in spectrum plot of CW portion of signal during the fixed test, showing peak at 0 Hz.
In this case, any frequency offset was so small (mainly due to no motion between transmitter and receiver) that it was rounded to 0 Hz by the FFT, so no frequency shift was applied. There is still the problem of energy being on the Q channel which must be dealt with. To perform the phase correction, the I/Q data was inspected in the complex plane to see the distribution of symbols as shown in figure 4.5.

![I/Q constellation plot of PN sequence before phase correction. Note the skew of the symbols, indicating I/Q imbalance.](image)

A histogram was performed to see the distribution of phase across the PN sequence in detail as shown in figure 4.6. It can be seen that there is a large distribution of symbols $\pi$ radians (180 degrees) apart (as would be expected in an ideal BPSK waveform per figure 3.4), however they are not centered around 0 and 180 degrees. The presence of many samples with seemingly random amplitude and almost random phase is due to the fact that the I/Q constellation plot includes all the samples of the PN sequence, including the phase transitions (which have almost random phase and amplitude).
Figure 4.6: Histogram of phase values of the PN sequence before phase correction.

This skew was fixed by applying a constant phase shift to the received PN sequence to bring the symbols towards 0 degrees of phase imbalance as shown in figure 4.7. The phase-corrected signal has much less skew as shown in figure 4.8. The result of this correction on the I/Q data can be seen in the time domain as well. It is apparent from figure 4.9 that most of the energy on the Q channel was removed and placed on the I channel.
Figure 4.7: Histogram of phase values after phase correction.

Figure 4.8: I/Q constellation plot of PN sequence after phase correction.
Figure 4.9: The first 250 µs of I/Q data of the PN sequence after phase correction.
The cross-correlation was then performed and is shown in figure 4.10. As expected, there is a peak at $t=14.39 \mu s$ (which is close to the expected delay of $14.129 \mu s$), but what was not expected was the larger peak at negative sample lag$^2$.

![Cross-correlation of 560 meter fixed-path link.](image)

**Figure 4.10: Cross-correlation of 560 meter fixed-path link.**

The negative sample peak was ignored, because the positive peak looked to be accurate. The discrepancy between the delay of the actual peak and the expected peak is mainly due to the relatively large variance in the transmit pipeline delay, as tabulated in table 2.1. Figure 4.11 shows the zoomed-in correlation result.

$^2$To investigate if the negative delay peak was an artifact or a repeating phenomenon, 50 samples were taken of the fixed path link. Of these, most contained the negative delay peak, but all of the positive delay peaks were at the correct delay. The reason for this issue is unknown, but the issue appeared to resolve itself after a USRP firmware update in the fall of 2022.
Figure 4.11: One of the cross-correlations of 560 meter fixed-path link, zoomed-in on the main peak. The peak width is approximately 2.5 $\mu$s.

Even though there were no discrete path delays observed, the peak was noticeably flat, instead of the pyramidal shape expected by the m-sequence auto-correlation. This is likely due to sub-symbol delays caused by multipath from reflections from nearby buildings.
4.2 July 2022 Sea Test

After general operation was verified, a field test was conducted in July 2022 with the transmitter on a moving vessel. The receiver LNB was positioned so that the horn antenna was pointing roughly westward as shown in figure 4.12. The transmitter was placed on the roof of the boat as shown in Figure 4.13. Upon reaching the pier’s location, the boat’s crew connected the transmitter to a 12V battery on the vessel, powering up the transmitter. The vessel then proceeded due westward at approximately 10 knots for approximately 20 kilometers. Several samples of I/Q data were captured at every 1 km waypoint and at several points between waypoints. 95 data points were taken in total over the course of approximately 4 hours with the majority of those representing stationary positions every 1 km.

Figure 4.12: RX horn and LNB mounted on the tower facing westward for the sea test.
Figure 4.13: Transmitter mounted on the boat. Note: this picture was taken with the bow of the boat in front of the photographer, indicating that the transmit antenna was facing in the aft direction. Photo by Nick Rowlett of UCSD.
Several of the data points were taken when the boat was still within visual range. Figure 4.14 shows the boat heading out to the 1 km waypoint.

Figure 4.14: Experimental setup of the sea test. Foreground: RX Horn. Background: Boat with transmitter.

The GPS track of the boat’s traveled path can be seen in figure 4.15. To minimize doppler shift at the different waypoints, the boat operators attempted to maintain position around the waypoint. This behavior can be seen in figure 4.16.
Figure 4.15: GPS track of boat path off the coast of La Jolla, California during the sea test of July 29, 2022.

Figure 4.16: GPS track of boat trying to fight north/south currents to maintain position around the waypoint (shown here at 9 km).
4.2.1 COAMPS Ducting Forecast

The Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) is an atmospheric model developed by the Naval Research Laboratory to predict atmospheric behavior, and by proxy, ducting behavior [26]. The forecast from COAMPS is not accurate enough to produce fine details about duct height, and worsens in quality at low altitudes [1], but it was used as a decision aid whether or not ducting is happening qualitatively and if a sea test should be done that day. Although the COAMPS model does not give an indication of duct strength, in theory, ducts with higher base heights will provide more prominent trapping effects at long transmitter-receiver distances. The COAMPS forecast for the day of the test showed the possibility of prominent ducting layers the day of, and it was decided shortly before the test that the experiment should be carried out.
4.2.2 Unmanned Aerial Vehicle Measurements

To help guide the data analysis, a small unmanned aerial vehicle (UAV) was used to gather a vertical refractivity profile as discussed in Section 1.2. An InterMet Systems iMet-XQ2 Meteorological Sensor (see Figure 4.17) was attached to a DJI Inspire 2 UAV to make the measurements in a column from a few meters above the ocean surface up to 120 meters above sea level. The sensor was placed on a mast approximately 1 foot above the rotors as to minimize the effects of downward air flow (prop wash) disturbing measurements during flight. Only the ascent vertical profile is used since the descent profile contains too much measurement ambiguity from the prop wash. The iMet sensor attached to the UAV is shown in figure 4.18.

![iMet-XQ2 sensor](image)

Figure 4.17: iMet-XQ2 sensor used for the UAV measurements in this thesis [6].
Figure 4.18: DJI Inspire 2 UAV with iMet sensor attached. Note location of sensor element in relation to propellers.

The data from the iMet sensor was downloaded to see if there were any surface ducts occurring. Figure 4.19 shows the plots of refractivity profiles generated from the iMet data throughout the day of 7/29/22.
Figure 4.19: Refractivity vertical profiles from the iMet sensor of the July 2022 sea test. Times are in UTC.

It can be seen that there was some weak ducting (based on the slope of the refractivity $M$) around 10-15 meters altitude that got weaker throughout the day. If the theory that ducting causes multipath is true, then these refractivity profiles indicate that there should be little to no multipath present, because the ducting is very weak and low in altitude.
4.2.3 Sea Test Data Analysis

The next step was to analyze the I/Q data from the sea test to see if multipath effects were occurring. The post-processing code developed to process the stationary test results was used on the sea test data. The correlation result of each trial was taken and inspected, and the distance estimation was compared with the GPS path distance as ground truth.

Plots of correlation peaks were individually made to demonstrate that the correlation peak is where it was expected. Figure 4.20 shows the cross-correlations for selected distances of 31 m, 1.06 km, and 4.8 km, corresponding to expected arrival times of $12.366 \mu s$, $15.796 \mu s$, and $28.263 \mu s$, respectively. These cross-correlations indicated a single peak and no discrete multipath delays. Although not all correlation time series are shown here, the rest of the trials did also not show any discrete arrivals.

![Figure 4.20: Cross-correlations from the sea test, d = 31 m, d = 1.06 km, and d = 4.8 km.](image)
In the 31 m link, the correlation peak is fairly close to the expected value. Any discrepancy could be reasonably explained by the fact that it is within the combined variance of the non-range-dependent delays, notably TX pipeline delay (table 2.1) and the RX sampling latency (table 2.2). However, the peaks for the 1.06 km and 4.8 km distances are showing up almost a whole μs (or even more) later than expected!

To investigate the range-dependent phenomenon, a plot of the measured delay of the main peak, expected delay for given distance (without pipeline delay factored in), and a total TX-RX delay estimate (calculated by subtracting the aforementioned measured delay from the expected delay) for all the trials in the sea test was produced and is shown in figure 4.21. Several trials contained correlation peaks with delays that were much higher than expected. In this case, this was the peak finding algorithm picking a random peak in the noise. Closer inspection of the I/Q data revealed that the file mysteriously only contained CW and no PN sequence. These samples returned no discernible correlation peaks (only noise) and had to be discarded (hence the spatial gaps in the data of figure 4.21, such as between 18 and 19 km). Why these trials contained only noise is unknown.
Figure 4.21: Plot of actual correlation peak time delays, expected peak delays (based only on time of flight from distance), and total TX-RX delays for every trial in the sea test.

The TX-RX delay estimate appears to increase with distance. The reason for this jump in delay is unknown.
The carrier to noise ratio (CNR) was computed in dB from the CW portion of each trial relative to the noise floor of the USRP (which itself is measured in full scale of the RX stage as dBFS) and is displayed in figure 4.22. The noise floor was approximately -76 dBFS over the course of the sea test. As expected, the CNR generally decreased with distance, with fluctuations due to motion in the transmitter causing variations in received power.

Figure 4.22: Plot of CNR with respect to distance.
As expected, the main peaks also had close to the expected 2 $\mu$s duration in the main peak, as was predicted by the auto-correlation function of the m-sequence. Figure 4.23 shows the duration of a main correlation peak at 18 km distance. The peak is greater than 2 $\mu$s in duration, but there is no “flattening” of the main peak as was seen in the test from Isaacs Hall to the pier. While only one peak is shown here, this was the case for all the trials in the sea test.

![Cross-correlation result from distance = 18 km showing the duration and no “flattening” of the main peak.](image)

**Figure 4.23:** Cross-correlation result from distance = 18 km showing the duration and no “flattening” of the main peak.
4.3 September 2022 Over-Water Fixed Path Test

Determined to discover a definite multipath signal when strong ducting conditions were observed, another field test was conducted in September 2022. This time, the channel was a fixed-path link from San Elijo State Beach in Cardiff, CA to the SIO Pier. The path distance was approximately 17.7 km and was primarily over water as shown in figure 4.24. This distance should result in a total (TOF + TX/RX delays) delay of 71.263 µs.

Figure 4.24: Path for the Cardiff fixed path measurements.

The COAMPS forecast was used to choose what day to conduct the test. The forecast for September 26, 2022 suggested strong ducting layers coinciding with a heat wave that would be present that day. More specifically, the forecast indicated very thick
ducts off the coast, which should in theory reflect more energy and provide more pronounced multipath effects. In this case, the day of the test was predicted to have visible multipath effects.

The RX horn antenna was pointed towards Cardiff as shown in figure 4.25. The transmitter was located on a railing on a cliff above the beach as shown in figures 4.26 and 4.27.
Figure 4.26: Transmitter mounted on the railing.

Figure 4.27: Transmitter facing towards the SIO pier.
The iMet sensor and Inspire UAV were also used to gather refractivity profiles during the test. Several flights were performed throughout the day to observe the temporal variations in the refractivity profile. The profile showed strong ducting conditions at the beginning of the test and increased in strength as the day continued, corresponding with an increase in temperature and decrease in humidity. Figure 4.28 shows the refractivity profiles as captured at the SIO pier the day of the Cardiff test.

Figure 4.28: Refractivity profiles captured at the SIO pier during the Cardiff test.
The Cardiff test consisted of 47 trials of time delay measurements, conducted over the course of approximately one hour. The main peak correlation peak delay for each trial was found. The average value and standard deviation of the main peak delay are tabulated in table 4.1. This was to investigate whether or not the peak was at the expected location and remained at that location, which it appears it did.

Table 4.1: Statistical analysis of the main peak correlation delay from the fixed-path link in the Cardiff test.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected value</td>
<td>71.263 µs</td>
</tr>
<tr>
<td>Number of trials</td>
<td>47</td>
</tr>
<tr>
<td>Mean</td>
<td>71.1441 µs</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.137 µs</td>
</tr>
</tbody>
</table>

The cross-correlation was then inspected for each of the trials. 28 of the 47 trials showed what looked like a discrete multipath bounce arrival. One such example is shown in figure 4.29. Each point displayed indicated a flattening of the main peak. Analysis of the delays observed is presented in the conclusion of this thesis.
Figure 4.29: Cross-correlation of trial 15 of 47 from the Cardiff fixed path test, distance = 17.1 km, with the main correlation peak at a delay of 71.31 $\mu$s and a secondary peak of 72.93 $\mu$s. The expected delay for the main peak is 71 $\mu$s.
Chapter 5

CONCLUSION

5.1 Discussion of System Goals

The goals stated in table 1.1 were accomplished, with the exception of the range metric. The actual specifications of the system were compared against the desired ones and are tabulated in 5.1. Future tests are anticipated to establish a true maximum system range.

Table 5.1: Actual system specifications.

<table>
<thead>
<tr>
<th></th>
<th>Desired</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of Operation</td>
<td>X-band (8-12 Ghz)</td>
<td>11 GHz</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>≥ 1 Sample/symbol</td>
<td>200 Samples/symbol</td>
</tr>
<tr>
<td>Maximum TX-RX distance</td>
<td>100 km</td>
<td>20 km (measured)</td>
</tr>
</tbody>
</table>

5.2 Interpretation of Experimental Results

The lack of any real multipath arrivals in the trials of the sea test agreed with the forecast from the refractivity measurements made via UAV during that test (little to no ducting observed). Unlike the Isaacs Hall-Pier measurements (in which it is possible that there was multipath from buildings and terrain nearby, even though the dominant mode of propagation was LOS), the trials in the sea test did not show any flattening of the main peak when inspected closely.

If the surface ducting of 10 m altitude is used for calculating delay based on equations 1.6 and 1.7, at the close end (30 m), the bounce should in the best case be 20 ns behind the main peak, and in the worst case will be 39 ns behind the main peak. At the far
end (20 km), the delay ranges from 1 ns to 33 ns behind the main peak. Since no 
flattening of the main peak or discrete path delays were observed in the sea test, there 
appears to be little evidence that the model proposed in section 1.5 offers valuable 
analysis to predicting delays through a duct.

In the Cardiff fixed-path test, there were signs of multipath propagation, which agreed 
with the refractivity measurements that strong elevated ducting was occurring. On 
the day of the test, the TX-RX distance was 17.71 km, and the duct height observed 
was around 240 to 300 m in altitude. The delay behind the main peak should be 
in the range of 22 ns (shortest bounce) to 805 ns (longest bounce) behind the main 
peak for a 240 m duct height. The numbers for a 300 m duct height are very similar. 
Regardless of which duct height is used, the delays observed are much longer than 
predicted from the duct model, as shown in figure 5.1. If the relative delay seen is 
used to estimate the duct height at this range, the estimated duct height is found to 
be within the range of 450 m to 2050 m. Some points in the Cardiff test were even 
seen to have sub-symbol period delays in the main peak as shown in figure 5.2.

Although further testing is required to verify how stronger ducting conditions will 
produce different results, preliminary testing indicates this system is capable of de-
tecting multipath propagation at 11 GHz, and that strong ducting in a channel may 
cause discrete and continuous multipath arrivals. This conclusion was reached after 
realizing that the Cardiff test had flattened main peaks as well as discrete discernible 
arrivals, whereas the July 2022 sea test had neither (when ducting was not observed). 
Further investigation is also required to determine if nearby terrain played a role in 
the flattened peaks of the Cardiff test.

With regards to the question of if the model of bounce propagation presented earlier in 
this thesis is valuable, the arrivals observed in the Cardiff test did not seem to match 
up with those predicted by the model, or if they did, they were likely too attenuated
Figure 5.1: Measurement of the multipath delay from the cross-correlation of figure 4.29. The delay behind the main peak was measured to be approximately 1.6 $\mu$s.

to be seen in the correlation. The model also does not explain the presence of discrete arrivals 1.6 $\mu$s later, like that of figure 5.1.

The design goal of being able to monitor multipath on the order of a symbol period was satisfied (within the constraints of the two symbols worth of duration of the correlation peak), as shown in figure 5.2. If a lower sample rate was used, it would not be possible to see the distinct “humps” in the main peak, and it would be harder to see the detail of the discrete arrivals. Although the experiments conducted in this thesis were not exhaustive, it appears the theory of ducting causing multipath may be true.
Figure 5.2: Zoomed in view of the main peak of the correlation from trial 36 of 47 in the Cardiff test.

5.3 Future Work

Further measurement campaigns are anticipated in response to strong ducting forecasts and rapid evaluation of refractivity conditions in the southern California area. Additionally, effort is being put forth using this system to characterize the marine propagation environment over a long period of time (on the order of several months) and look for seasonal patterns in multipath propagation with respect to local environmental conditions.

Future improvements to the system/ideas for investigation could involve the following:
• Upgrading the RF phase shifter to a different modulation method: The jitter in the phase shifter is thought to degrade the correlation coefficient. A solution to this would be to use a very fast switching I/Q modulator such as a double-balanced mixer, or a Direct Digital Synthesis (DDS) RF integrated circuit (RFIC) that can generate the waveform in a cleaner fashion. This should in theory improve the correlation coefficient.

• Use of a dedicated circuit to output the PN sequence, instead of using GPIO on the SBC (which has a rather large standard deviation in pipeline delay). Minimizing the standard deviation of any transmit pipeline delays will give more insight into environmental effects on propagation delay.

• Sea tests at longer ranges beyond visual line of sight to see if multipath effects are occurring.

• Characterization of the USRP’s receive chain and the LNB to perform a detailed link budget.

• Investigation into using a longer m-sequence, which will increase the processing gain and improve the correlation coefficient (assuming the channel properties can still assumed to be constant over the longer PN sequence period).

• Using an even higher sample-rate digitizer, such as an oscilloscope, to capture samples.

• Investigation of the early correlation peak issue, which would require reverse engineering the USRP’s architecture and firmware in past revisions, as well as determining if the transmitter is at fault.
BIBLIOGRAPHY


https://www.oc.nps.edu/NWDC_EM_Course/course_materials/.


[19] Q. Wang, R. J. Burkholder, C. Yardim, L. Xu, J. Pozderac, A. Christman,
H. J. Fernando, D. P. Alappattu, and Q. Wang, “Range and height
measurement of x-band em propagation in the marine atmospheric
boundary layer,” IEEE Transactions on Antennas and Propagation, vol. 67,


[22] U.S. Government Publishing Office, Table of Frequency Allocations. As


mesoscale prediction system (coamps),” Monthly Weather Review, vol. 125,
pp. 1414–1430, 7 1997. COAMPS model data retrieved from
Dependency note: This program requires a UHD version of at least 4.1.0.

/*

rx_v1

Author: Jack Gallegos

July/August 2022

Coastal Observing Research And Development Center (CORDC)
Scripps Insitution of Oceanography

This program samples from an Ettus Research (now NI) USRP
→ Software-Defined Radio (SDR)
Purpose is to sample 1 second of IQ data starting on GPS Pulse Per
→ Second (PPS) signal
The program starts, connects to the USRP, stores the requested
→ number of samples in the specified path. Default is in the
→ executable folder.
See program options section for default options and config.

Most of the code in this example was taken from the UHD examples
→ folder.
Therefore, most of the credit goes to:

Ettus Research

Copyright 2016 Ettus Research, a National Instruments Company

SPDX-License-Identifier: GPL-3.0-or-later

*/

#include <uhd/exception.hpp>
#include <uhd/types/tune_request.hpp>
#include <uhd/usrp/multi_usrp.hpp>
#include <uhd/utils/safe_main.hpp>
#include <uhd/utils/thread.hpp>
#include <boost/format.hpp>
#include <boost/lexical_cast.hpp>
#include <boost/program_options.hpp>
#include <chrono>
#include <complex>
#include <csignal>
#include <fstream>
#include <iostream>
#include <thread>
#include <stdio.h>
#include <time.h>
#include <string>

namespace po = boost::program_options;

static bool stop_signal_called = false;
```cpp
void sig_int_handler(int)
{
    stop_signal_called = true;
}

template<typename samp_type>
void recv_to_file(uhd::usrp::multi_usrp::sptr usrp,
                  const std::string& cpu_format,
                  const std::string& wire_format,
                  const size_t& channel,
                  const std::string& file,
                  const std::string& path,
                  size_t samps_per_buff,
                  unsigned long long num_requested_samples,
                  double time_requested = 0.0,
                  bool bw_summary = false,
                  bool stats = false,
                  bool null = false,
                  bool enable_size_map = false,
                  bool continue_on_bad_packet = false)
{
    unsigned long long num_total_samps = 0;
    // create a receive streamer
    uhd::stream_args_t stream_args(cpu_format, wire_format);
    std::vector<size_t> channel_nums;
    channel_nums.push_back(channel);
    stream_args.channels = channel_nums;
    //...
```
```cpp
uhd::rx_streamer::sptr rx_stream =
    usrp->get_rx_stream(stream_args);

uhd::rx_metadata_t md;
std::vector<samp_type> buff(samps_per_buff);
std::ofstream outfile;
// get unix time stamp from GPSDO
unsigned long seconds;

seconds = (unsigned long)time(NULL);

printf("Seconds since January 1, 1970 00:00 UTC= %ld\n", seconds);
if (not null)

    outfile.open((path+file+"_"+std::to_string(seconds)+".cs16").c_str(),
        std::ofstream::binary);

bool overflow_message = true;

// force 1 second
num_requested_samples = usrp->get_rx_rate();
std::cout << boost::format("# of requested samples: %d") %
    numRequestedSamples << std::endl;

// setup streaming
// uhd::stream_cmd_t stream_cmd((numRequestedSamples == 0)
// ?
//    uhd::stream_cmd_t::STREAM_MODE_START_CONTINUOUS
```
//

uhd::stream_cmd_t::STREAM_MODE_NUM_SAMPS_AND_DONE);

uhd::stream_cmd_t
stream_cmd(uhd::stream_cmd_t::STREAM_MODE_NUM_SAMPS_AND_MORE);

stream_cmd.num_samps = size_t(num_requested_samples);

stream_cmd.stream_now = false;

usrp->set_time_next_pps(0.0);

//Wait for it to apply

//Sleep for 1.5 seconds

std::this_thread::sleep_for(std::chrono::milliseconds(1500));

//Check times

//This is causing a lot of crashes

//std::cout << boost::format("USRP time: %d") %

(usrp->get_time_last_pps()).get_frac_secs << std::endl;

usrp->clear_command_time();

//uhd::time_spec_t time_now = usrp->get_time_now();

//std::cout << boost::format("USRP time: %d.%d") %

(time_now.get_full_secs()) % (time_now.get_frac_secs())<<

std::endl;
stream_cmd.time_spec = uhd::time_spec_t(2.00000000); // wait 1 second before stream command should execute

rx_stream->issue_stream_cmd(stream_cmd);

std::cout << boost::format("Receiving %d samples at %f Sps...") %
num_requested_samples
% (usrp->get_rx_rate())
<< std::endl;

typedef std::map<size_t, size_t> SizeMap;
SizeMap mapSizes;
const auto start_time = std::chrono::steady_clock::now();
const auto stop_time =
start_time + std::chrono::milliseconds(int64_t(1000 *
  time_requested));
// Track time and samps between updating the BW summary
auto last_update = start_time;
unsigned long long last_update_samps = 0;

// Run this loop until either time expired (if a duration was
given), until
// the requested number of samples were collected (if such a
number was
while (not stop_signal_called
    and (num_requested_samples != num_total_samps or
         num_requested_samples == 0)
    and (time_requested == 0.0 or
         std::chrono::steady_clock::now() <= stop_time)) {
    const auto now = std::chrono::steady_clock::now();

    size_t num_rx_samps =
        rx_stream->recv(&buff.front(), buff.size(), md, 10.0,
                         enable_size_map);

    if (md.error_code == uhd::rx_metadata_t::ERROR_CODE_TIMEOUT) {
        std::cout << boost::format("Timeout while streaming") <<
                   std::endl;
        break;
    }

    if (md.error_code == uhd::rx_metadata_t::ERROR_CODE_OVERFLOW) {
        if (overflow_message) {
            overflow_message = false;
            std::cerr
                << boost::format("Got an overflow indication. Please
                           consider the following:\n" " Your write medium must sustain a rate of
                       %fMB/s.\n")
" Dropped samples will not be written to
the file."

" Please modify this example for your
purposes."

" This message will not appear
again."

% (usrp->get_rx_rate(channel) *
sizeof(samp_type) / 1e6);

)

continue;

}

if (md.error_code != uhd::rx_metadata_t::ERROR_CODE_NONE) {
    std::string error = str(boost::format("Receiver error:
%s") % md.strerror());
    if (continue_on_bad_packet) {
        std::cerr << error << std::endl;
        continue;
    } else
        throw std::runtime_error(error);
}

if (enable_size_map) {
    SizeMap::iterator it = mapSizes.find(num_rx_samps);
    if (it == mapSizes.end())
        mapSizes[num_rx_samps] = 0;
    mapSizes[num_rx_samps] += 1;
}
num_total_samps += num_rx_samps;

if (outfile.is_open()) {
    outfile.write((const char*)&buff.front(), num_rx_samps *
                   sizeof(samp_type));
}

if (bw_summary) {
    last_update_samps += num_rx_samps;
    const auto time_since_last_update = now - last_update;
    if (time_since_last_update > std::chrono::seconds(1)) {
        const double time_since_last_update_s =
            std::chrono::duration<double>(time_since_last_update).count();
        const double rate = double(last_update_samps) /
            time_since_last_update_s;
        std::cout << "\t" << (rate / 1e6) << " Msps" <<
            std::endl;
        last_update_samps = 0;
        last_update = now;
    }
}

const auto actual_stop_time = std::chrono::steady_clock::now();

stream_cmd.stream_mode =
    uhd::stream_cmd_t::STREAM_MODE_STOP_CONTINUOUS;
rx_stream->issue_stream_cmd(stream_cmd);

if (outfile.is_open()) {
    outfile.close();
}
usrp->clear_command_time();
if (stats) {
    std::cout << std::endl;
    const double actual_duration_seconds =
        std::chrono::duration<float>(actual_stop_time -
                                   start_time).count();

    // std::cout << boost::format("Received %d samples in %f
    → seconds") % num_total_samps
    // % (actual_duration_seconds)
    // << std::endl;
    std::cout << boost::format("Received %d samples at %f Msp") %
              num_total_samps
                  % (usrp->get_rx_rate()/1e6)
                  << std::endl;
    const double rate = (double)num_total_samps /
                        (actual_duration_seconds);
    //std::cout << (rate / 1e6) << " Msp" << std::endl;
    if (enable_size_map) {
        std::cout << std::endl;
    }
std::cout << "Packet size map (bytes: count)" <<
    std::endl;
for (SizeMap::iterator it = mapSizes.begin(); it !=
    mapSizes.end(); it++)
    std::cout << it->first << "::" << it->second <<
    std::endl;
}
}

typedef std::function<uhd::sensor_value_t(const std::string&)> get_sensor_fn_t;

bool check_locked_sensor(std::vector<std::string> sensor_names,
    const char* sensor_name,
    get_sensor_fn_t get_sensor_fn,
    double setup_time)
{
    if (std::find(sensor_names.begin(), sensor_names.end(),
        sensor_name)
        == sensor_names.end())
        return false;

    auto setup_timeout = std::chrono::steady_clock::now()
        + std::chrono::milliseconds(int64_t(setup_time
        * 1000));
bool lock_detected = false;

std::cout << boost::format("Waiting for \"%s\": ") % sensor_name;
std::cout.flush();

while (true) {
    if (lock_detected and (std::chrono::steady_clock::now() > setup_timeout)) {
        std::cout << " locked." << std::endl;
        break;
    }

    if (get_sensor_fn(sensor_name).to_bool()) {
        std::cout << "+";
        std::cout.flush();
        lock_detected = true;
    } else {
        if (std::chrono::steady_clock::now() > setup_timeout) {
            std::cout << std::endl;
            throw std::runtime_error(str(boost::format(
                "timed out waiting for consecutive locks on sensor \"%s\"")
                % sensor_name));
        }
        std::cout << ",";
        std::cout.flush();
    }
}
std::this_thread::sleep_for(std::chrono::milliseconds(100));

}

std::cout << std::endl;
return true;
}

int UHD_SAFE_MAIN(int argc, char* argv[])
{
    // variables to be set by po
    std::string args, file, path, type, ant, subdev, ref, wirefmt;
    size_t channel, total_num_samps, spb;
    double rate, freq, gain, bw, total_time, setup_time, lo_offset;

    // setup the program options
    po::options_description desc("Allowed options");
    // clang-format off
    desc.add_options()
        ("help", "help message")
        ("args", po::value<std::string>(&args)->default_value(""),
        → "multi uhd device address args")
        ("file",
        → po::value<std::string>(&file)->default_value("usrp_samples"),
        → "name of the file to write binary samples to WITHOUT
        → extension. extension is .cs16")
        ("type",
        → po::value<std::string>(&type)->default_value("short"),
        → "sample type: double, float, or short")
    // clang-format on
    return 0;
}
("path", po::value<std::string>(&path)->default_value(""),
  "path of file to write to")

//("duration",
  po::value<double>(&total_time)->default_value(1.0000),
  "total number of seconds to receive")

("spb", po::value<size_t>(&spb)->default_value(10000),
  "samples per buffer")

("rate", po::value<double>(&rate)->default_value(50e6), "rate
  of incoming samples")

("nsamps",
  po::value<size_t>(&total_num_samps)->default_value(1000),
  "total number of samples to receive")

("freq", po::value<double>(&freq)->default_value(100e6), "RF
center frequency in Hz")

("lo-offset",
  po::value<double>(&lo_offset)->default_value(0.0),
  "Offset for frontend LO in Hz (optional)"

("gain", po::value<double>(&gain), "gain for the RF chain")

("ant", po::value<std::string>(&ant)->default_value("TX/RX"),
  "antenna selection")

("subdev", po::value<std::string>(&subdev), "subdevice
  specification")

("channel", po::value<size_t>(&channel)->default_value(0),
  "which channel to use")

("bw", po::value<double>(&bw), "analog frontend filter
  bandwidth in Hz")
// clang-format on
po::variables_map vm;
po::store(po::parse_command_line(argc, argv, desc), vm);
po::notify(vm);

// print the help message
if (vm.count("help")) {
    std::cout << boost::format("UHD RX samples to file %s") % desc
        << std::endl;
    std::cout << std::endl
}
This application streams data from a single channel of a USRP "device to a file."

return ~0;

bool bw_summary = vm.count("progress") > 0;
bool stats = vm.count("stats") > 0;
bool null = vm.count("null") > 0;
bool enable_size_map = vm.count("sizemap") > 0;
bool continue_on_bad_packet = vm.count("continue") > 0;

if (enable_size_map)
    std::cout << "Packet size tracking enabled - will only recv one packet at a time!"

if (vm.count("ref")) {

    std::cout << std::endl;
    std::cout << boost::format("Creating the usrp device with args: \%s...") \% args
        << std::endl;
    uhd::usrp::multi_usrp::sptr usrp = uhd::usrp::multi_usrp::make(args);

    // Lock mboard clocks
    if (vm.count("ref")) {

usrp->set_clock_source(ref);

// always select the subdevice first, the channel mapping affects the other settings
if (vm.count("subdev"))
    usrp->set_rx_subdev_spec(subdev);

std::cout << boost::format("Using Device: %s") %
        usrp->get_pp_string() << std::endl;

// set the sample rate
if (rate <= 0.0) {
    std::cerr << "Please specify a valid sample rate" <<
              std::endl;
    return ~0;
}

// force sample for 1 second
total_num_samps = rate;

// set to use ext PPS
usrp->set_time_source("external");
std::cout << boost::format("Setting RX Rate: \%f Msps...") % (rate / 1e6) << std::endl;
usrp->set_rx_rate(rate, channel);
std::cout << boost::format("Actual RX Rate: \%f Msps...")
% (usrp->get_rx_rate(channel) / 1e6)
<< std::endl
<< std::endl;

// set the center frequency
if (vm.count("freq")) { // with default of 0.0 this will always be true
    std::cout << boost::format("Setting RX Freq: \%f MHz...") % (freq / 1e6)
        << std::endl;
    std::cout << boost::format("Setting RX LO Offset: \%f MHz...")
        % (lo_offset / 1e6)
        << std::endl;
    uhd::tune_request_t tune_request(freq, lo_offset);
    if (vm.count("int-n"))
        tune_request.args = uhd::device_addr_t("mode_n=integer");
    usrp->set_rx_freq(tune_request, channel);
    std::this_thread::sleep_for(std::chrono::milliseconds(110));
    // sleep 110ms (~10ms after retune occurs) to allow LO to lock
    std::cout << boost::format("Actual RX Freq: \%f MHz...")
        % (usrp->get_rx_freq(channel) / 1e6)
// set the rf gain
if (vm.count("gain")) {
    std::cout << boost::format("Setting RX Gain: %f dB...") % gain
              << std::endl;
    usrp->set_rx_gain(gain, channel);
    std::cout << boost::format("Actual RX Gain: %f dB...")
              % usrp->get_rx_gain(channel)
              << std::endl
              << std::endl;
}

// set the IF filter bandwidth
if (vm.count("bw")) {
    std::cout << boost::format("Setting RX Bandwidth: %f MHz...")
              % (bw / 1e6)
              << std::endl;
    usrp->set_rx_bandwidth(bw, channel);
    std::cout << boost::format("Actual RX Bandwidth: %f MHz...")
              % (usrp->get_rx_bandwidth(channel) / 1e6)
              << std::endl
              << std::endl;
}
// set the antenna
if (vm.count("ant"))
    usrp->set_rx_antenna(ant, channel);

std::this_thread::sleep_for(std::chrono::milliseconds(int64_t(1000 * setup_time)));

// check Ref and LO Lock detect
if (not vm.count("skip-lo")) {
    check_locked_sensor(usrp->get_rx_sensor_names(channel),
            "lo_locked",
            [usrp, channel](const std::string& sensor_name) {
            return usrp->get_rx_sensor(sensor_name, channel);
        },
        setup_time);
    if (ref == "mimo") {
        check_locked_sensor(usrp->get_mboard_sensor_names(0),
            "mimo_locked",
            [usrp](const std::string& sensor_name) {
            return usrp->get_mboard_sensor(sensor_name);
        },
        setup_time);
    }
    if (ref == "external") {
        check_locked_sensor(usrp->get_mboard_sensor_names(0),
            "ref_locked",
            [usrp](const std::string& sensor_name) {
            return usrp->get_mboard_sensor(sensor_name);
        },
        setup_time);
    }
}
435     return usrp->get_mboard_sensor(sensor_name);
436 },
437     setup_time);
438 }
439 if (ref == "gpsdo") {
440     check_locked_sensor(usrp->get_mboard_sensor_names(0),
441                       "gpsdo_locked",
442                       [usrp](const std::string& sensor_name) {
443             return usrp->get_mboard_sensor(sensor_name);
444         },
445         setup_time);
446 }
447 }
448 
449 if (total_num_samps == 0) {
450     std::signal(SIGINT, &sig_int_handler);
451     std::cout << "Press Ctrl + C to stop streaming..." <<
452         std::endl;
453 }
454 #define recv_to_file_args(format) \\
455     (usrp, \\
456      format, \\
457      wirefmt, \\
458      channel, \\
459      file, \\
460      path, \\
461      file, \\
462      path,
spb, total_num_samps, total_time, bw_summary, stats, null, enable_size_map, continue_on_bad_packet)

// recv to file

if (wirefmt == "s16") {
    if (type == "double")
        recv_to_file<double> recv_to_file_args("f64");
    else if (type == "float")
        recv_to_file<float> recv_to_file_args("f32");
    else if (type == "short")
        recv_to_file<short> recv_to_file_args("s16");
    else
        throw std::runtime_error("Unknown type " + type);
}

else if (wirefmt == "sc8") {
    if (type == "double")
        recv_to_file<double> recv_to_file_args("fc32");
    else if (type == "float")
        recv_to_file<float> recv_to_file_args("fc16");
    else if (type == "short")
        recv_to_file<short> recv_to_file_args("sc8");
    else
        throw std::runtime_error("Unknown type " + type);
}
throw std::runtime_error("Unknown type " + type);

else {

    if (type == "double")
        recv_to_file<std::complex<double>>
            recv_to_file_args("fc64");
    else if (type == "float")
        recv_to_file<std::complex<float>>
            recv_to_file_args("fc32");
    else if (type == "short")
        recv_to_file<std::complex<short>>
            recv_to_file_args("sc16");
    else
        throw std::runtime_error("Unknown type " + type);

}

// finished
std::cout << std::endl << "Done!" << std::endl << std::endl;

return EXIT_SUCCESS;
%% Post-Processing script for processing received sea test X-band PN transmitter data from the USRP.

% Written by Tony De Paolo and Jack Gallegos
% Coastal Observing R&D Center, Scripps Institution of Oceanography
% August 2022

clear;
close all;

% for this test, maximum range was 20km, so maximum expected delay is
% 20000/3e8 = 66us, or 13333 samples at 200Msps
addpath('\path\to\current_dir\');
dpath = '\path\to\data\';
d = dir([dpath '*cs16']);
capture_time = zeros(length(d),1);
capture_sample_delay = zeros(length(d),1);
capture_time_delay = zeros(length(d),1);
capture_range = zeros(length(d),1);
rssi_pn = zeros(length(d),1);

% Capture correlation coefficient value
r_tx_rx = zeros(length(d),1);
% create matrix for correlation results for export

cross_correlation_matrix = zeros([6553400,length(d)]);

%% GPS Data

% Bring in GPS data

gps_filename = 'gpx_file.gpx';
p = gpxread(gps_filename,'FeatureType','Track');

% Create new time vector and extract date/time from filename

t_new = p.Time;
t_new = strrep(t_new,'Z','');

t_new = datetime(t_new,'InputFormat','yyyy-MM-dd''T''HH:mm:ss');
gps_dnum = datenum(t_new);
t_labels = split(cellstr(t_new(1:end)));

[C,ia,ic] = unique(gps_dnum);
gps_dnum = gps_dnum(ia);
gps_lon = p.Longitude(ia);
gps_lat = p.Latitude(ia);

% pier lat/lon for reference

olat = 32.867049;
olon = -117.257382;

% calculate distance from pier

gps_d = lonlat2km(olon,olat,gps_lon,gps_lat);
gps_d = abs(gps_d); %absolute distance from pier
% Interpolate distance to times
% gps_d = interp1(gps_dnum, d, dnum);
% gps_dnum = interp1(gps_dnum, gps_dnum, dnum, 'linear', 'extrap');
% get the m-sequence

load("tx_mseq_1x32767len_fs200e6.mat");
Tx_msequence = -Tx_msequence;

% execute over all trials in the data set
for n=1:length(d)
    % read in the samples
    fprintf('reading %s\n', d(n).name);
    fprintf('File %s of %s\n', num2str(n), num2str(length(d)));
    tic
    % Note: read_complex_binary_sc16 is a
    samples = read_complex_binary_sc16([dpath d(n).name]);
    t_whole = linspace(0, length(samples)*5e-9, length(samples));

    % figure('Name','Raw IQ, no corrections')
    % hold on
    % subplot(211)
    % plot(t_whole, real(samples))
    % ylabel('RX I Channel Amplitude')
    % xlabel('Time (s)')
    % subplot(212)
% sequence is 32767us long, at 200Msps, that's 6553400 samples
% this test went out to 20km, for a maximum delay of 66us, or
% and additional 13333 samples
pn_end = (2^15-1)*200 + round(20000/3e8)*200e6;
Ipn = real(samples(1:pn_end));
%Ipn = real(samples(1:length(samples)));
Qpn = imag(samples(1:pn_end));
%Qpn = imag(samples(1:length(samples)));
t_cw_stop = (pn_end*5e-9)+(length(Icw)*5e-9);

% clear samples % done for saving memory

% figure('Name','IQ of CW signal, no frequency correction')
% hold on
% subplot(2,1,1)
% plot(t_cw,Icw);
% ylabel('RX I Channel Amplitude')
% xlabel('Time (s)')
% subplot(2,1,2)
% plot(t_cw,Qcw);
% ylabel('RX Q Channel Amplitude')
% xlabel('Time (s)')
% hold off

% frequency vector
fs=200e6;
fftlen = 2^(nextpow2(length(Icw))-1);
fres = fs/fftlen;
f = (-fs/2:fres:fs/2-fres)';
S = fftshift(fft(Icw+1i*Qcw,fftlen));
% determine frequency error
[~,sidx]=max(abs(S.*conj(S))); % figure('Name','Spectrum of CW portion')
plot(f,10*log10(abs(S.*conj(S)))) % xlim([-2e6 2e6])
% title(['Frequency offset f = ', num2str(f(sidx))])
% drawnow;

f_offset = f(sidx);

clear Icw Qcw

% start time

t0 = epoch2UTC(str2double(d(n).name(14:23)));
fprintf('File time %s\r',datestr(t0)); % relative time vector

Ts = 5e-9;
t = (0:Ts:(pn_end-1)*Ts)';

% find associated GPS time, distance
[C,idx1] = min(abs(t0-gps_dnum));
gps_t = gps_dnum(idx1);
fprintf('GPS time %s\r',datestr(gps_t));
gps_dist = gps_d(idx1);
fprintf('GPS distance %f km\r',gps_dist);

% % Filter I and Q
% filter_len = 50;
filter = ones(1,filter_len)./filter_len;

Ipn = conv(Ipn,filter,'same');
Qpn = conv(Qpn,filter,'same');

figure('Name','IQ of PN sequence without any corrections')
hold on
subplot(2,1,1)
plot(t(1:50000),Ipn(1:50000))
ylim([-5000 5000])
ylabel('RX I Channel Amplitude')
xlabel('Time (s)')
subplot(2,1,2)
plot(t(1:50000),Qpn(1:50000))
ylabel('RX Q Channel Amplitude')
xlabel('Time (s)')
ylim([-5000 5000])
drawnow;
hold off

fs=200e6;
fftlen = 2^(nextpow2(length(Ipn))-1);
fres = fs/fftlen;
f = (-fs/2:fres:fs/2-fres)';

S = fftshift(fft(Ipn+1i*Qpn,fftlen));
\[
\text{[~,\text{sidx}]=max(abs(S.*conj(S))});
\]
\[
\text{figure('Name','PN power spectrum without any corrections')}\]
\[
\text{plot(f,10*log10(abs(S.*conj(S))));}
\]
\[
\text{xlim([-2e6 2e6])}\]
\[
\text{ylim([80 180])}\]
\[
\text{title('Uncorrected PN power spectrum')}\]
\[
\text{drawnow;}\]
\[
\text{\% take out frequency error}
\]
\[
\text{corrected_signal = (Ipn+1i*Qpn).*(exp(-1i*2*pi.*f_offset.*t));}
\]
\[
\text{\%corrected_signal_whole =}
\]
\[
\text{\quad samples.)*(exp(-1i*2*pi.*f_offset.*t_whole));}
\]
\[
\text{Ipn_corrected = real(corrected_signal);}
\]
\[
\text{Qpn_corrected = imag(corrected_signal);}
\]
\[
\text{\% figure('Name','IQ of PN sequence comparison w/wo freq}
\]
\[
\quad \text{\% correction')}
\]
\[
\text{\% hold on}
\]
\[
\text{\% subplot(211)
\]
\[
\text{\% plot(t(1:50000),Ipn_corrected(1:50000),
\]
\[
\quad ...t(1:50000),2500*Tx_msequence(1:50000))}
\]
\[
\text{\%}
\]
\[
\text{\quad plot(t(1:50000),Ipn(1:50000),t(1:50000),Ipn_corrected(1:50000))}
\]
\[
\text{\% legend('Uncorrected','Frequency corrected')}
\]
% legend('Corrected PN sequence', 'Transmitted PN sequence');
% ylim([-5000 5000])
% ylabel('RX I Channel Amplitude')
% title('PN sequence w/ frequency and phase correction');
% xlabel('Time (s)');
% hold off
% subplot(2,1,2)
% hold on
% plot(t(1:50000), Qpn(1:50000), t(1:50000), Qpn_corrected(1:50000))
% legend('Uncorrected', 'Frequency corrected')
% ylabel('RX Q Channel Amplitude')
% xlabel('Time (s)');
% ylim([-5000 5000]);
% hold off

% figure('Name', 'IQ of PN sequence w/ frequency correction')
% subplot(2,1,1)
% plot(t(1:500000), Ipnn_corrected(1:500000))
% ylim([-5000 5000])
% ylabel('RX I Channel Amplitude')
% xlabel('Time (s)')
% subplot(2,1,2)
% plot(t(1:500000), Qpn_corrected(1:500000))
% ylabel('RX Q Channel Amplitude')
% xlabel('Time (s)')
% ylim([-5000 5000])
232  \%    drawnow;
233  
234  \% figure('Name','IQ of PN sequence comparison of frequency shift')
235  \%    subplot(2,1,1)
236  \%    hold on
237  \%    plot(t(1:50000),Ipn(1:50000))
238  \%    plot(t(1:50000),Ipn_corrected(1:50000))
239  \%    legend('Uncorrected','Corrected')
240  \%    ylim([-5000 5000])
241  \%    ylabel('RX I Channel Amplitude')
242  \%    hold off
243  \%    subplot(2,1,2)
244  \%    hold on
245  \%    plot(t(1:50000),Qpn(1:50000))
246  \%    plot(t(1:50000),Qpn_corrected(1:50000))
247  \%    legend('Uncorrected','Corrected')
248  \%    ylabel('RX Q Channel Amplitude')
249  \%    ylim([-5000 5000])
250  \%    drawnow;
251  \%    hold off

252
253
254

255  \%    spectrum
256  \%    \% S = fftshift(fft(Ipn_corrected+1i*Qpn_corrected,fftlen));
257  \%    [~,sidx]=max(abs(S.*conj(S)));
258  \%    figure('Name','PN power spectrum with frequency correction')
```matlab
plot(f,10*log10(abs(S.*conj(S))))
xlim([-2e6 2e6])
ylim([80 180])
title('Frequency corrected PN power spectrum')
drawnow;

% Remove phase error, put all the energy on the I channel
% 180 degree phase ambiguity, so the resulting I channel may be ▸ 180
% degrees out of phase making for a negative correlation peak
phi = mod(angle(corrected_signal),2*pi);
figure('Name','Phase values - uncorrected')
%subplot(2,1,1)
h=histogram(phi,360);
%title('Phase distribution (deg) - without phase correction')
xlabel('Phase (radians)')
drawnow
[~,bin]=max(h.Values);
phi0=h.BinEdges(bin);

% figure('Name','Constellation of uncorrected IQ')
scatterplot(corrected_signal)
% hold on;
% drawnow;
```
corrected_signal =
    abs(Ipn_corrected+1i*Qpn_corrected).*(exp(1i*(phi-phi0)));
Ipn_corrected = real(corrected_signal);
Qpn_corrected = imag(corrected_signal);

Ppn = 20*log10(mean(abs((Ipn_corrected/32768).^2)));

rssi_pn(n) = Ppn;

phi_corrected = mod(angle(corrected_signal),2*pi);
figure('Name','Phase values - corrected')
    %subplot(2,1,2)
    h_corrected=histogram(phi_corrected,360);
    %title('Phase distribution (deg) - with phase correction ')
    xlabel('Phase (radians)')
    drawnow
    [~,bin]=max(h.Values);
    phi0=h.BinEdges(bin);

    % figure('Name','Constellation of corrected IQ')
    %scatterplot(corrected_signal)
    % hold on;
    % drawnow;

    % % eye diagram plot
    % angles = rad2degunwrap(angle(corrected_signal));
% unit_vector_pn_sequence = 1 * exp(1i*angles);

% figure('Name','Phase angle of all corrected IQ')
plot(t,angles)
xlabel('Time (s)')
ylabel('Phase angle (radians)')
hold on;
drawnow;

% figure('Name','IQ of PN sequence w/frequency and phase correction')
subplot(2,1,1)
hold on
% plot(t(1:50000),Ipn_corrected(1:50000),
...t(1:length(Ipn_corrected)),Ipn_corrected(1:length(Ipn_corrected))
% legend('Corrected PN sequence','Transmitted PN sequence');
% ylim([-5000 5000])
% ylabel('RX I Channel Amplitude')
% title('PN sequence w/ frequency and phase correction');
% xlabel('Time (s)');
hold off
subplot(2,1,2)
hold on
% plot(t(1:length(Qpn_corrected)),Qpn_corrected(1:length(Qpn_corrected)))
% apply this to whole signal for visualization

% Plot the whole signal

% figure('Name','IQ of whole thing w/frequency and phase correction')
% subplot(2,1,1)
% hold on
% plot(t(1:50000),Ipn_corrected(1:50000), t(1:50000),2500*Tx_msequence(1:50000))
% legend('Corrected PN sequence','Transmitted PN sequence');
% ylim([-5000 5000])
% ylabel('RX I Channel Amplitude')
% title('PN sequence w/ frequency and phase correction');
% xlabel('Time (s)');
% hold off
% subplot(2,1,2)
% hold on
% plot(t_whole(1:length(t_whole)),real(samples(1:length(samples))))
\% 360  \% \text{plot}(t\_whole(1:length(t\_whole)),\text{imag}(samples(1:length(samples))))
\% ylabel('RX Q Channel Amplitude')
\% xlabel('Time (s)');
\% ylim([-5000 5000]);
\% hold off

\% \% spectrum
\% S = \text{fftshift}(\text{fft}(Ipn\_corrected+1i*Qpn\_corrected,fftlen));
\% figure('Name','Phase corrected PN power spectrum')
\% plot(f,10*log10(abs(S.*conj(S))))
\% xlim([-2e6 2e6])
\% ylim([80 180])
\% drawnow;

\% correlate
\[R,\text{lags}] = \text{xcorr}(Ipn\_corrected,Tx\_msequence);

\% take absolute value of \text{R} in case we are 180 out of phase
\text{R} = \text{abs}(R);

\% \% After fixing the phase error, only correlate with the I channel
\% figure('Name','Correlation')
\% subplot(211)
\% plot(lags,R)
\% drawnow
% [C, idx] = max(R);
% xlim([-50000 50000])
%
% % adjust index to center
% sample_delay = idx-length(Ipn_corrected);
% time_delay = sample_delay.*1/fs;
% range = time_delay.*3e8;
% title(['Delay = ' num2str(sample_delay) ' samples = ' num2str(time_delay) ' seconds = ' num2str(range) ' meters']);
% drawnow
%
% delayed_Tx_msequence = [NaN.*ones(sample_delay,1); Tx_msequence];
%
% subplot(2,1,2)
% plot(t(1:50000),Ipn_corrected(1:50000),...
% t(1:50000),delayed_Tx_msequence(1:50000).*5000);
% title('Tx msequence delayed on I channel')
% drawnow
%
% After fixing the phase error, only correlate with the I channel
% figure('Name','Correlation')
% semilogy(lags,R)
% drawnow
% [C, idx] = max(R);
% xlim([-50000 50000])
% [C, idx] = max(R);
[C, idx] = max(R(length(Ipn_corrected):end)); % This one picks out the good peaks, from Tony's version of this script
figure('Name','Correlation','Position',[692 368 664 511])
plot(lags*5e-9,R)
r_tx_rx(n)=C;
% Generate vector for positive-time-delayed correlations
R_pos_time=R(length(R)/2:end);
pos_time= lags(length(lags)/2:end)*5e-9;
cross_correlation_matrix(:,n) = R_pos_time;
% R_normalized = R/C;
% drawnow

% xlim([-50000 50000])
% adjust index to center
sample_delay = idx-length(Ipn_corrected);
time_delay = sample_delay.*1/fs;
range = time_delay.*3e8;
ylabel('Correlation Coefficient (R_{TX-RX})')
% xlim([0 35e-6])
xlabel('Delay (s)')
% ylabel('Sample Lag (5ns per sample)')
% title(['Delay = ' num2str(sample_delay) ' samples = ' num2str(time_delay) ' seconds = ' num2str(range) ' meters']);
title(['Delay = ' num2str(sample_delay) ' samples, GPS distance = ' num2str(gps_dist) ' km']);
drawnow

% save results
capture_time(n) = t0;
capture_sample_delay(n) = sample_delay;
capture_time_delay(n) = time_delay;
capture_range(n) = range;

% expected
dist(n)=gps_dist;
tof(n) = gps_dist*1000/3e8;
pipeline_delay = time_delay - tof(n);

%close all % when looping so the screen doesn't fill up
toc

end
disp('Done!')

%results =
- table(capture_time, capture_sample_delay, capture_time_delay,
... capture_range, dist, tof, pipeline_delay);
% writetable(results,'results_jack_test.xlsx');

% figure(4)
% plot(capture_time,capture_range);
% datetick('x');

%% Run after loading data

% SAMP = 200; % which sample (1 Hz) should be used?
%
% % Filter I and Q
% filter_len = 15;
% filter = ones(1,filter_len);
% I = inphase;
% Q = quadrature;
%
% for i = 1:length(dnum)
%    inphase(i,:) = conv(I(i,:),filter,'same');
%    quadrature(i,:) = conv(Q(i,:),filter,'same');
% end

% phaseDeg = atan2d(quadrature,inphase);
% magnitude = sqrt(inphase.^2 + quadrature.^2);
%
% figure()
% subplot(211)

% subplot(212)
% plot(t(SAMP,:),magnitude(SAMP,:));
% title('Magnitude of IQ Samples');
% subplot(212)
% hold on;
% plot(t(SAMP,:),atan2d(Q(SAMP,:),I(SAMP,:)),'.');
% plot(t(SAMP,:),phaseDeg(SAMP,:),'.');
% title('Phase of IQ Samples');
% legend('Unfiltered','Filtered');
%
% %%% Generating known sequence
% numSeq = 137; % Number of PN sequences sent in succession
%
% base = 2;
% power = 15;
% n = base.^power - 1;
% shift = 0;
% index = 1;
% [msequence] = mseq(base,power,shift,index);
% Fs = 15e6;
%
% msequence_interp =
% interp1(1e-6:(1e-6):(n)*1e-6,msequence,1e-6:1/Fs:(n)*1e-6,'nearest');
%
% % seqtrain = zeros(1,256*numSeq);
% % for i = 1:numSeq-1
% %   if i == 0
% % segtrain(0:256) = msequence;
% % else
% % segtrain(i*256:i*256 + 255) = msequence;
% % end
% % end
%
% % for i = 1:numSeq
% % segtrain = [segtrain msequence];
% % end
%
% Fs = 15e6;
% Fmod = 1e6;
% t_new = 1/Fs:1/Fs:(1/Fs)*length(t);
% t_old = t_new(1):t_new(end)/length(segtrain):t_new(end);
% seqtrain_interp = interp1(t_old,segtrain,t_new,'nearest');
% seqtrain_interp(isnan(seqtrain_interp)) = 0;
%
% % FFT based f_off
% % finding coarse freq offsets from FFT
% Fs = 1/t(1,2);
% mult = 10;
% fftlen = mult*length(I);
% fres = Fs/fftlen;
% pf = -Fs/2+fres:fres:Fs/2;
% recSig = I+1j*Q;
% rec_sigFFT = (1/fftlen).*fftshift(fft(rec_sig(SAMP,:),fftlen));
% figure()
% plot(pf,10*log10(abs(rec_sigFFT).^2));
% xlabel('Frequency');
% ylabel('Magnitude (dB)');
% title('Received Signal Spectrum');
%
% [~,pos] = max(10*log10(abs(rec_sigFFT).^2));
% f_off = pf(pos);
% Ts = t(SAMP,2)-t(SAMP,1);
% phase_error = 2*pi*Ts*f_off:...
% 2*pi*Ts*f_off: ...  
% 2*pi*Ts*length(t(1,:))*f_off;
%
% % skip unwrapping the data
% coarse_phase = mod(mod(phaseDeg(SAMP,:),360) -
% mod(rad2deg(phase_error),360),360);  
% coarse_phase = mod(coarse_phase,360); % shift by 90 degrees
% figure()
% plot(coarse_phase,'.');
% xlabel('Samples');
% ylabel('Phase (degrees)');
% title('FFT corrected phase')
%
% corrected_signal =
% magnitude(SAMP,:).*exp(1j.*deg2rad(coarse_phase));
%
% mseqfull = [msequence_interp
    - zeros(1,length(I)-length(msequence_interp))];
% [r, lags] = xcorr(angle(corrected_signal)/(pi/2), msequence_interp);
% r = r/length(msequence_interp); % normalize to smaller sliding window
% txcorr = (-length(r)/2+1:length(r)/2)*(1/Fs);
% figure()
% plot(txcorr, r);
% title('Correlation from FFT corrected waveform');
% xlabel('Time (s)');
% ylabel('Correlation');
%
% %% Taking FFT for each column
% Fs = 1/t(1,2);
% mult = 1;
% fftlen = mult*length(I);
% fres = Fs/fftlen;
% pf = -Fs/2+fres:fres:Fs/2;
% samp = 5;
% rec_sig = inphase+1j*quadrature;
% 
% % % 11s computation for short data
% rec_sigFFT = (1/fftlen).*fftshift(fft((rec_sig.^2)', fftlen));
% [~, ind_f_off] = max(abs(rec_sigFFT), [], 1);
% f_off = pf(ind_f_off);
% 
% t_vec = 2*pi*Ts:2*pi*Ts:2*pi*Ts*length(t(:, :));
```matlab
588 % phase_error = f_off'*t_vec;
589 %
590 % coarse_phase = mod(mod(phaseDeg,360) -
591 % mod(rad2deg(phase_error),360),360);
592 % coarse_phase = mod(coarse_phase + 90,360); % shift by 90 degrees
593 %
594 % corrected_signal = magnitude.*exp(1j.*deg2rad(coarse_phase));
595 %
596 % r = zeros(size(corrected_signal,1),2*size(corrected_signal,2)-1);
597 % txcorr = (-size(r,2)/2+1:size(r,2)/2)*(1/Fs);
598 % tof = zeros(1,size(corrected_signal,1));
599 % for i = 1:size(corrected_signal,1)
600 %    r(i,:) =
601 %      xcorr(angle(corrected_signal(i,:))/(pi/2),msequence_interp);
602 % end
603 % T_CHIP = 1e-6;
604 % PIPELINE_DELAY = 12.3e-6 + T_CHIP;
605 % PIPELINE_DELAY_ESTIMATE = 13.5e-6;
606 %
607 % peak_ind = find(txcorr>0); % take peaks only for t>0
608 % for i = 1:size(r,1)
609 %    [pks(i),locs(i)] =
610 %      findpeaks(abs(r(i,peak_ind)),txcorr(peak_ind),'MinPeakDistance'
611 %        ...,250e-6,'Npeaks',1);
612 % end
613 % tof = locs;
614 % tof = tof - PIPELINE_DELAY;
```
%% figure()
%% plot(dnum, tof.*(3e8), '.');
%% xlabel('Time');
%% ylabel('Distance (m)')
%% title('Distance from Time of Flight');
%% datetick('x', 'HH:MM');

%% Comm toolbox

%% scatterplot(rec_sig(SAMP,:))

%% coarse = comm.CoarseFrequencyCompensator('SampleRate', Fs, ...
%%      'FrequencyResolution', 10);

%% syncCoarse = zeros(size(rec_sig));
%% for i=1:size(rec_sig,1)
%%  [syncCoarse(i,:), f_off_coarse(i)] = coarse(rec_sig(i,:));
%% end

%% r = zeros(size(syncCoarse,1), 2*size(syncCoarse,2)-1);
%% txcorr = (-size(r,2)/2+1:size(r,2)/2)*(1/Fs);
%% tof = zeros(1, size(syncCoarse,1));
%% for i = 1:size(syncCoarse,1)
%%  r(i,:) =
%%    xcorr(angle(syncCoarse(i,:))/(pi/2), msequence_interp);
%% end
% for i = 1:size(r,1)
% [pks(i),locs(i)] =
% findpeaks(abs(r(i,peak_inds)),txcorr(peak_inds),'MinPeakDistance',
...250e-6,'Npeaks',1);
% end
% tof = locs;
% tof = tof - PIPELINE_DELAY;
%
% figure()
% subplot(211)
% plot(txcorr,(1/length(msequence_interp))*r(SAMP,:));
% xlabel('Time (s)');
% ylabel('Cross Correlation');
% title('Corrected waveform correlation');
% xlim([0 1500e-6]);
%
% subplot(212)
% plot(dnum,tof*(3e8),'.');
% xlabel('Time');
% ylabel('Distance (m)');
% title('Distance from Time of Flight');
% datetick('x','HH:MM');
%
% figure()
% plot(txcorr,(1/length(msequence_interp))*r(SAMP,:));
% xlabel('Time (s)');
% ylabel('Cross Correlation');
% title('Corrected waveform correlation');
% xlim([0 1500e-6]);
%
% ind1 = 1:size(r,1);
% ind2 = round(length(txcorr)/2):round(length(txcorr)/2) + 2000;
%
% figure()
% waterfall(ind2,dnum,(1/length(msequence_interp))*r(ind1,ind2));
% xlabel('Correlation Time');
% ylabel('Sample Time (s)');
% zlabel('Correlation');
% datetick('y','HH:MM');
%
% figure()
% mesh(dnum,txcorr(ind2),(1/length(msequence_interp))*r(ind1,ind2),'
% ...,'edgecolor','interp');
% xlabel('Correlation Time (s)');
% ylabel('Sample Time');
% zlabel('Correlation');
% datetick('x','HH:MM');
% colorbar;
%
% figure()
% contour(dnum,txcorr(ind2),(1/length(msequence_interp))*r(ind1,ind2),')
imagesc(dnum,txcorr(ind2),(1/length(msequence_interp))*r(ind1,ind2));

ylabel('Correlation Time (s)');

xlabel('Sample Time');

zlabel('Correlation');

datetick('x','HH:MM');

colorbar;

%% reproduce comm toolbox FO

fftlen = 10*size(rec_sig,2);

fres = Fs/fftlen;

pf = -Fs/2:fres:Fs/2 - fres;

fooFFT = (1/fftlen)*fftshift(fft(rec_sig.^2,fftlen,2),2);

[~,fe] = max(abs(fooFFT),[],2);

fe = (1/2)*pf(fe);

t_vec = 2*pi*Ts:2*pi*Ts:2*pi*Ts*length(t(1,:));

phase_error = fe'*t_vec;
coarse_phase_foo = mod(mod(phaseDeg,360) -
  mod(rad2deg(phase_error),360),360);

% coarse_phase_foo = mod(coarse_phase_foo -
  (0.5)*coarse_phase_foo(:,1)',360); % shift by half of first
  sample

% corrected_signal = magnitude.*exp(1j.*deg2rad(coarse_phase_foo));

% figure()
% plot(t(SAMP,:),angle(corrected_signal(SAMP,:)))
% xlabel('Time');
% ylabel('Phase (radians)');
% title('Corrected Phase');
% xlim([-10 3000]*(1e-7))

% mseqfull = [msequence_interp
  zeros(1,length(I)-length(msequence_interp))];
% for i = 1:length(fe)
%    [r2(i,:),lags] =
%      xcorr(angle(corrected_signal(i,:))/(pi/2),msequence interp);
% end
% r2 = r2/length(msequence_interp); % normalize to smaller sliding
% window
% taccorr = (-length(r2)/2+1:length(r2)/2)*(1/Fs);
%
% for i = 1:size(r2,1)
Guardado en 

[pks(i),locs2(i)] = findpeaks(abs(r2(i,peak_inds)),txcorr(peak_inds),'MinPeakDistance',...
250e-6,'Npeaks',1);

% end
% tof2 = locs2;
% tof2 = tof2 - PIPELINE_DELAY;
%
% figure()
% subplot(211)
% plot(txcorr,r2(SAMP,:));
% title('Correlation from FFT corrected waveform');
% xlabel('Time (s)');
% ylabel('Correlation');
%
% subplot(212)
% hold on;
% plot(dnum,tof2*(3e8),'.');
% plot(gps_dnum,1000*gps_d);
% xlabel('Time');
% ylabel('Distance (m)');
% legend('Estimated time of flight','GPS distance');
% datetick('x','HH:MM');
%
% figure()
% imagesc(dnum,txcorr(ind2),(1/length(msequence_interp))
...*abs(r2(ind1,ind2))');
% ylabel('Correlation Time (s)');
xlabel('Sample Time');

zlabel('Correlation');

datetick('x', 'HH:MM');

title('Correlation Peaks throughout time');
a = colorbar;

ylabel(a, 'Correlation');