

Laboratory Number:	3
Laboratory Title:	Pulse Code Modulation
Full Name:	Leo Berman
TUID:	916027207

Introduction:

The purpose of this lab is to investigate pulse code modulation, different kinds of line coding, and investigating the power spectrums of each line coding. The different types of line coding being investigated are Unipolar Non-Return to Zero, Polar Non-Return to Zero, Unipolar Return to Zero, Bipolar Return to Zero, Manchester, and Differential. The two ones that I will be focusing on in this report are the Polar Non-Return to Zero coding and Differential encoding. We are going to be investigating the types of coding by plotting the generation of a stream of data. Then, we will investing the power spectrums and the frequency spectrum and investigating a few crucial points.

Procedure:

Before Starting:

1. Define an amplitude and bit rate to use based off of your TUID.

```
tuid = [9,1,6,0,2,7,2,0,7];  
A = tuid(9) + 2; % amplitude in amps  
rb = 1000 * (tuid(8) + 3); % Bit rate hz
```

Task 1: (Original Signal)

1. Generate the bit interval which is equal to the inverse of the bit rate.
2. Define the sampling frequency as $100 * \text{bit rate}$.
3. Declare the number of symbols as 2 and the number of bits per symbol as 1
4. Generate an X axis that is scaled to the bit interval
5. Then, we generate a random slew of zeros and ones where zeros correspond to the y-value zero and the ones correspond to the y-value A as defined above
6. Then we plot both a stem plot and a regular line plot, this is our bas signal

Task 1: (Polar Non-Return to Zero):

1. Generate a vector of values by multiplying 2 * the value of the binary vector – A/2 where A is the value defined above.
2. Then we iterate through the previous vector and append bit interval // sampling interval number of elements.
3. Then plot on a stem plot and a line plot

Task 1: (Differential)

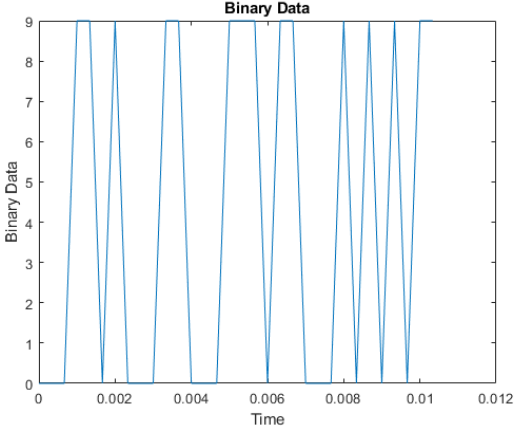
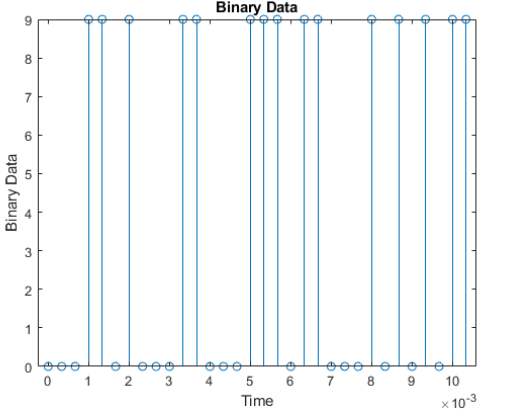
1. Generate a vector of zeros the same size as the previously generated data stream
2. Perform a differential encoding on the vector of zeros by comparing to the original and masking `not(xor(previous(not(xor)),current_element))`
3. Finally multiply that by A as defined above
4. Then plot on a stem plot and a line plot

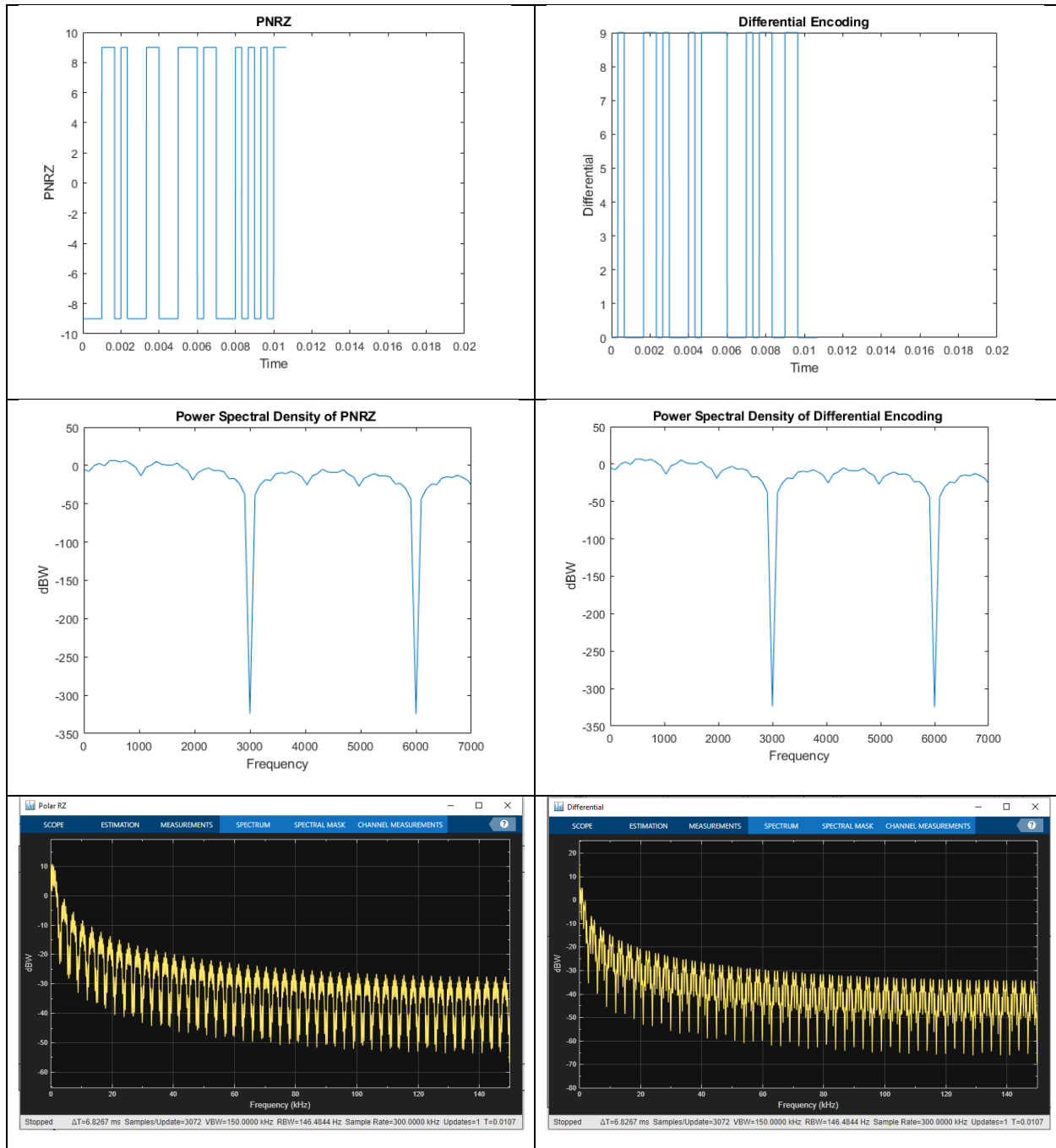
Task 2:

1. Plot a line plot of the power spectral density
2. Plot a plot utilizing the scope MATLAB function

Results:

Task 1:

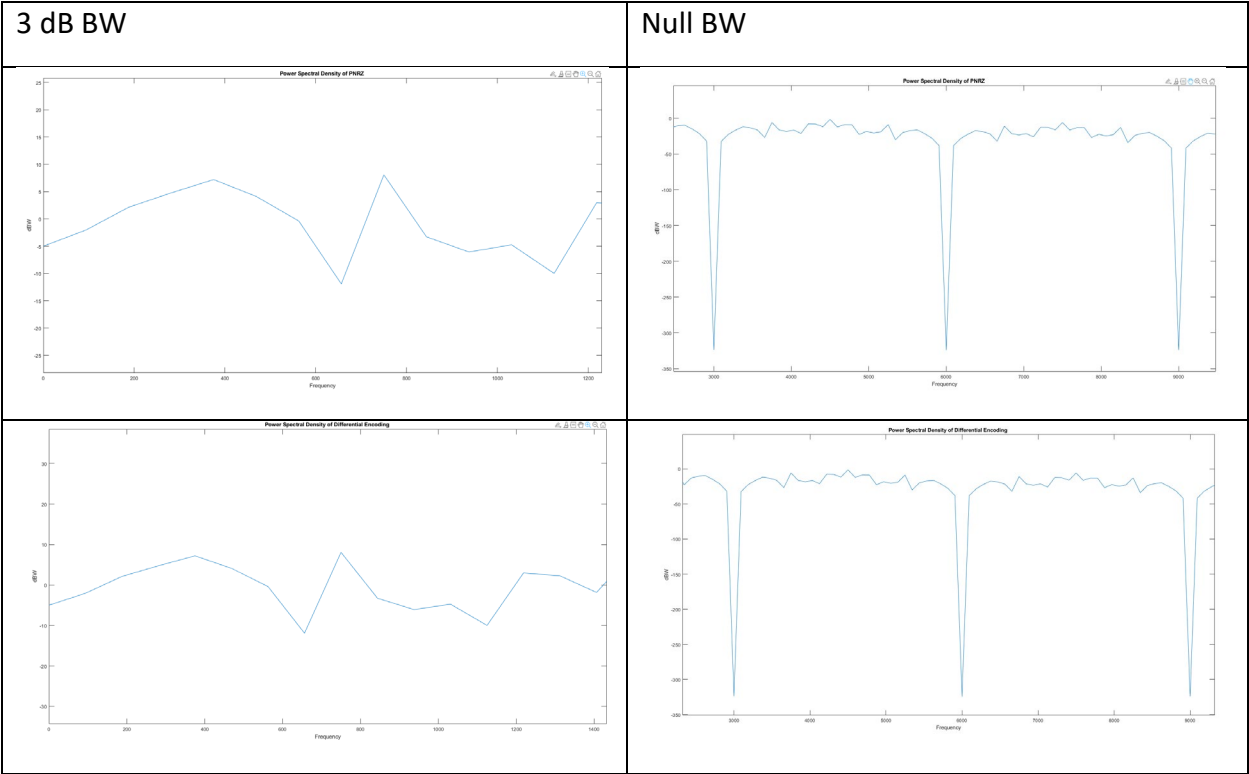
DataStream Line Plot	DataStream Stem Plot
 <p>A line plot titled 'Binary Data' showing a square wave signal. The y-axis is labeled 'Binary Data' and ranges from 0 to 9. The x-axis is labeled 'Time' and ranges from 0 to 0.012. The signal alternates between 0 and 9 at regular intervals.</p>	 <p>A stem plot titled 'Binary Data' showing a square wave signal. The y-axis is labeled 'Binary Data' and ranges from 0 to 9. The x-axis is labeled 'Time' and ranges from 0 to 10, with a multiplier of $\times 10^{-3}$ at the end. The signal alternates between 0 and 9 at regular intervals.</p>
Polar Non-Return Zero	Differential



Task 2:

Parameter	Symbol Rate	Bit Rate	3 dB BW	Null BW [hz]
PNRZ	1	3000	250	2500
Differential	1	3000	488	3027

Parameter	Symbol Rate	Bit Rate	3 dB BW	Null BW [hz]
PNRZ	1	3000	800	9000
Differential	1	3000	800	9000



Descriptive Answers to Tasks:

Polar Non-Return to Zero	
Pros	Cons
<ul style="list-style-type: none"> Simple Small bandwidth 	<ul style="list-style-type: none"> Signal droop No clock Cannot reliably transfer DC components Susceptible to noise
Differential	
Pros	Cons
<ul style="list-style-type: none"> Clock data for tracking phase 	<ul style="list-style-type: none"> Expensive

<ul style="list-style-type: none"> • Good for reliable transfer of DC components • Larger bandwidth due to double of rate for clock. 	<ul style="list-style-type: none"> • Requires a lot of storage due to having to double points for the shortest time change due to transfer of clock
--	--

Conclusion:

Polar Non-Return to Zero is a cheap way to encode binary data. Due to the inherent nature of this data being cheap and efficient to send, this data encoding is used in telecommunications and industrial settings. Differential encoding on the other hand is used in image, audio, and video processing. This is due to differential encoding's ability to interpret the data and predict the future data so it reduces the apparent time to the user. It can make a video more clear with less blips due to the fact that these predicted data points can fill in during a network interrupt.