gps

August 28, 2024

1 Pre-amble

Author: Calvin Wen

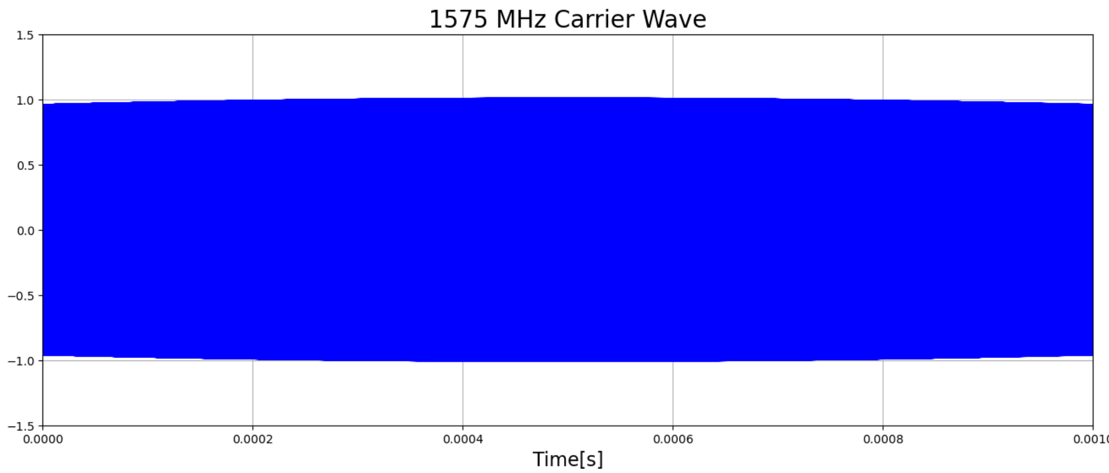
The purpose of this notebook is to prove that it is possible to retrieve information from signals (-20dB) below the noise floor. Such is the case for GPS signals, which uses CDMA & BPSK modulation techniques.

2 Import libraries

[ ]:
```
import random
import numpy as np
%matplotlib inline
from numpy import sin, pi

from matplotlib import rcParams
import matplotlib.pylab as plt
```

3 Create Carrier Wave

[ ]:
```
f = 1575e6
f_prn =  1.023e6
t=np.linspace(0,1e-3,int(10e-3*f))
# print(len(t))
carrier = lambda t,f: sin(2*pi*f*t)

plt.figure(figsize=(16,6))
plt.plot(t,carrier(t,f),'b', lw=3)
plt.title('1575 MHz Carrier Wave',size=20)
plt.xlim([0,t[-1]])
plt.ylim([-1.5,1.5])
plt.xlabel('Time[s]',size='16')
plt.grid()
```



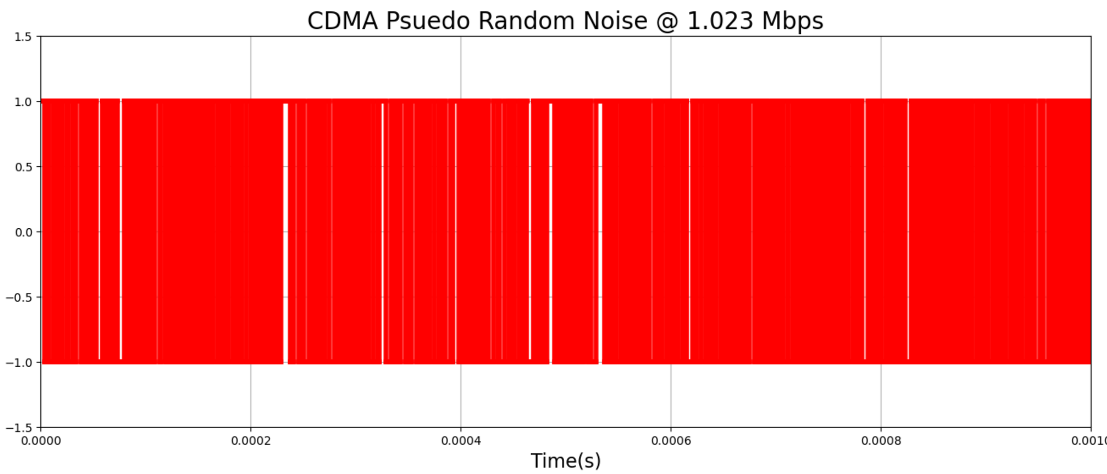
4 Create Psuedo Random Noise (PRN) code for CDMA

[ ]:
```python
random.seed("GPS PRN")
prn_seq = [random.choice([1,-1]) for i in range(1023)]

print("Our PRN chip:", str(prn_seq).replace('-1','0'))
```

```
Our PRN chip: [0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0,
0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0,
0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1,
1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0,
1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1,
1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1,
0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0,
1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0,
0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0,
0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0,
1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1,
1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0,
1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0,
1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1,
1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0,
0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1,
0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
```

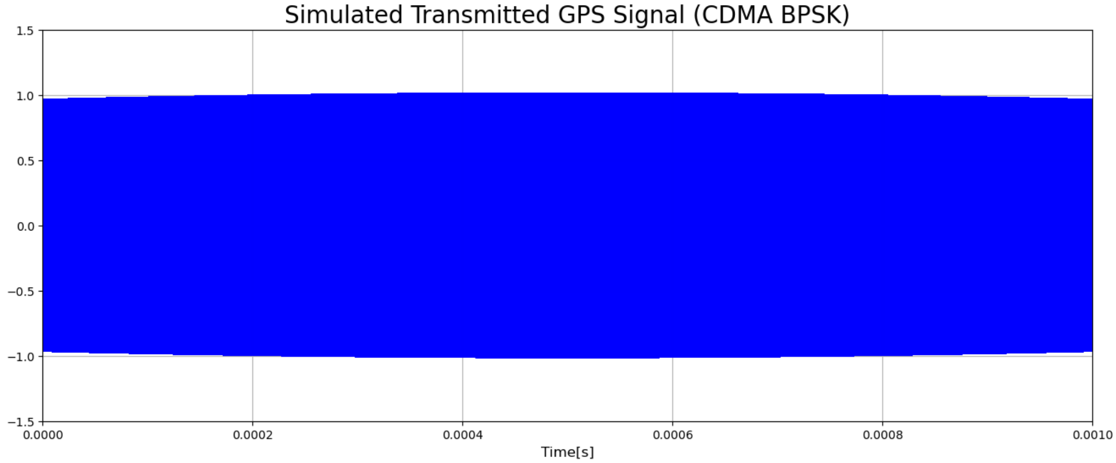
```
0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1,
0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1,
1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0,
0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0,
1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1,
1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1,
1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,
1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0,
1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1,
0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0,
1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0,
1, 0]
```

[ ]:
```
def prn_np(t, f_prn, prn_seq, bit):
    return [ prn_seq[int(ti*f_prn)%1023]*bit for ti in t]

bit = -1 # this represents the data bit taht contains GPS information
data = prn_np(t,f_prn, prn_seq, bit)

plt.figure(figsize=(16,6))
plt.plot(t,data, color='red',lw=4)
plt.title('CDMA Psuedo Random Noise @ 1.023 Mbps',size='20')
plt.xlim([0,t[-1]])
plt.ylim([-1.5,1.5])
plt.xlabel('Time(s)',size='16')
plt.grid()
```



5 Binary Phase Shift Keying (BPSK)

[ ]:
```
signal = lambda t,f_prn, carrier, data: carrier(t,f) * data
modulated_signal = signal(t,f_prn, carrier, data)

plt.figure(figsize=(16,6))
# plt.plot(t,prn_np(t,f_prn),color='red',lw=4)
plt.plot(t,modulated_signal,'b',lw=4,alpha=0.6)
plt.title('Simulated Transmitted GPS Signal (CDMA BPSK)',size=20)
plt.xlim([0,t[-1]])
plt.ylim([-1.5,1.5])
plt.xlabel('Time[s]',size=12)
plt.grid()
```



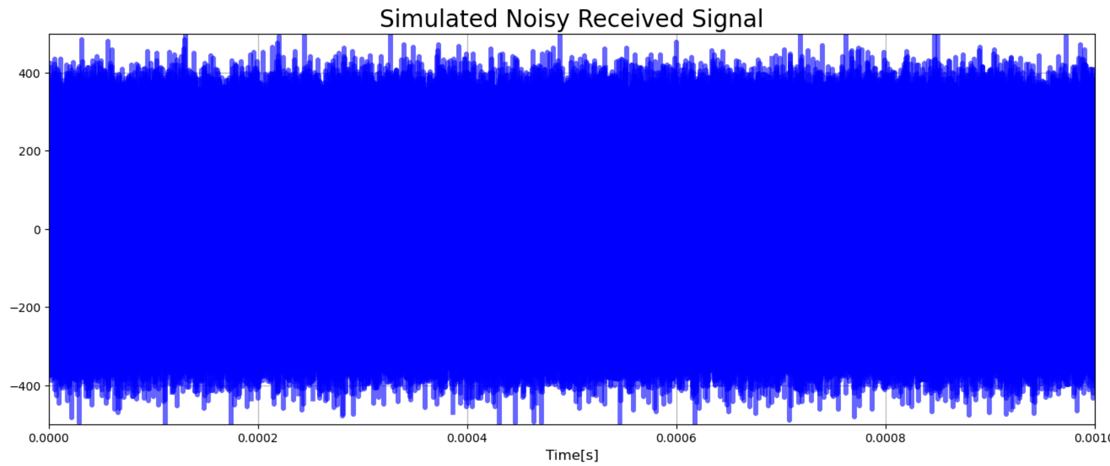
6 Correlation Study

6.1 Simulate GPS Signal -20dB below noise floor

[ ]:
```
# Generate signals
snr = -20 #db
noise_amp = 10**-(snr/10)*1
noise = noise_amp* np.random.normal(0, 1, len(t))
ref_prn = prn_np(t,f_prn, prn_seq, 1)
ref_signal = signal(t,f_prn, carrier, ref_prn)
signal_noisy = modulated_signal + noise
```

6.2 Visualise simulated noisy received GPS signal

[ ]:
```
rcParams['agg.path.chunksize'] = 101
rcParams['path.simplify_threshold'] = 0.7
plt.figure(figsize=(16,6))
plt.plot(t,signal_noisy,'b',lw=4,alpha=0.6)
plt.title('Simulated Noisy Received Signal',size=20)
plt.xlim([0,t[-1]])
plt.ylim([-500,500])
plt.xlabel('Time[s]',size=12)
plt.grid()
```



6.3 Correlation on CA signal hidden under noise floor

[ ]:
```
gps_correlation = np.corrcoef(ref_signal, signal_noisy)[0, 1]
print('CA Correlation:', gps_correlation)
```

```
CA Correlation: -0.007023575045144799
```

6.4 Correlation on Pure Noise

[ ]:
```
snr = -20 #db
noise_amp = 10**-(snr/10)*1
signal1 = noise_amp* np.random.normal(0, 1, len(t))
signal2 = np.random.normal(0, 1, len(t))

noise_correlation = np.corrcoef(signal1, signal2)[0, 1]
print('Noise Correlation:', noise_correlation)
```

```
Noise Correlation: -0.00023694233920208748
```

6.5 Comparison between correlation values of noise and GPS signal hidden below noise floor

[ ]:
```
print("Simulated CA correlation normalised to noise correlation:␣
↪",gps_correlation/noise_correlation)
print("Correlation value of GPS signal -20dB below noise floor is {:.2f} times␣
↪that of correlation value of pure noise".format(gps_correlation/
↪noise_correlation))
```

```
Simulated CA correlation normalised to noise correlation:  29.642549612690416
Correlation value of GPS signal -20dB below noise floor is 29.64 times that of
correlation value of pure noise
```