Example of Matched Filter Processing: Digital Cross Correlation

In Notes#9 we learned about the "matched filter" concept that is really very important in RF sensors of all types, THz sensors included. It defines the optimal way to do signal processing in active sensors, such as radars and spectrometers. The active sensor is assumed to have a transmit waveform encoded by some means, by either analog or digital modulation. We showed that if the noise is bandlimited Gaussian, then the optimal signal-processing is *cross correlation*. Cross correlation is easiest carried out on digitally-modulated waveforms because of the availability of sequential logic and digital-signal processing (DSP) in digital circuits that readily carry out cross correlations, convolutions, and a variety of other operations at very high speed and very high precision. And following the intuitive derivation of the matched filter concept, we desire the correlator to "compress" the waveform in the time domain as much as possible so that the sampled output of the correlator has the maximum possible "post-detection" signal-to-noise ratio. Clearly, the digital "coding" of the waveform is important. We seek a binary code that, upon correlation with a facsimile of itself, produces the greatest resemblance to a delta function in the time domain.

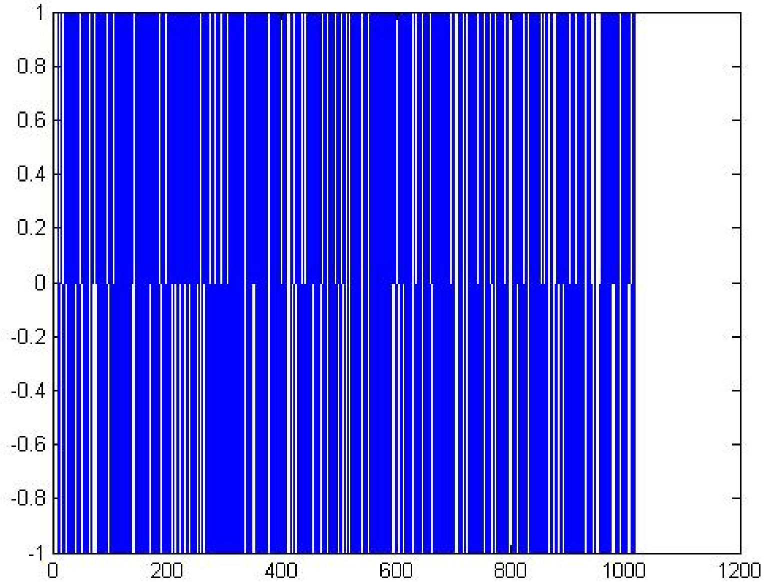
In the 1960s applied mathematicians and communications engineers made many great discoveries, one of which was the "maximum-length" (i.e., "M") binary sequence. The mathematical properties and practical means of generation is discussed widely in the literature including http://en.wikipedia.org/wiki/Maximum_length_sequence. In short, they are a sequence of 1s and 0s, each of length 2^N -1, where N is any integer. When the 0s of the sequence are offset relative to zero (i.e., all 0s become -1), the autocorrelation displays the remarkable property

$$R_{xx}(i) = \sum_{m}^{N} x_m x_{m-i} = N \text{ if } i = 0, = -1 \text{ otherwise}$$

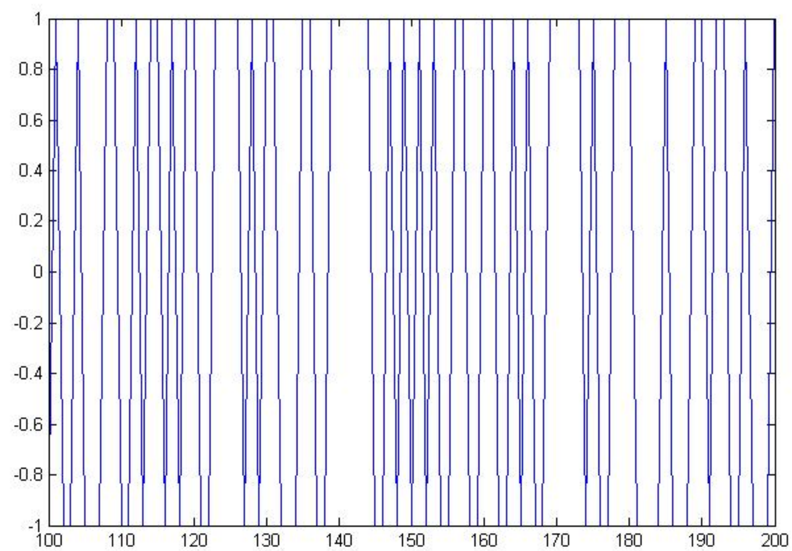
Like so many great discoveries in applied mathematics, the M sequences can be generated from simple algorithms. What follows is the MATLAB code for generating a 10-bit M sequence:

```
nbits = 10;
nprn = 2*nbits – 1;
prn = ones([nbits 1]);
for ibit = nbits+1:nprn
  %  prn(ibit)=-prn(ibit-nbits)*prn(ibit-(nbits-2));
  prn(ibit)=-prn(ibit-nbits)*prn(ibit-3);
end
figure(1)
plot(prn)
```

The 1023-bit long bipolar M sequence is shown below.



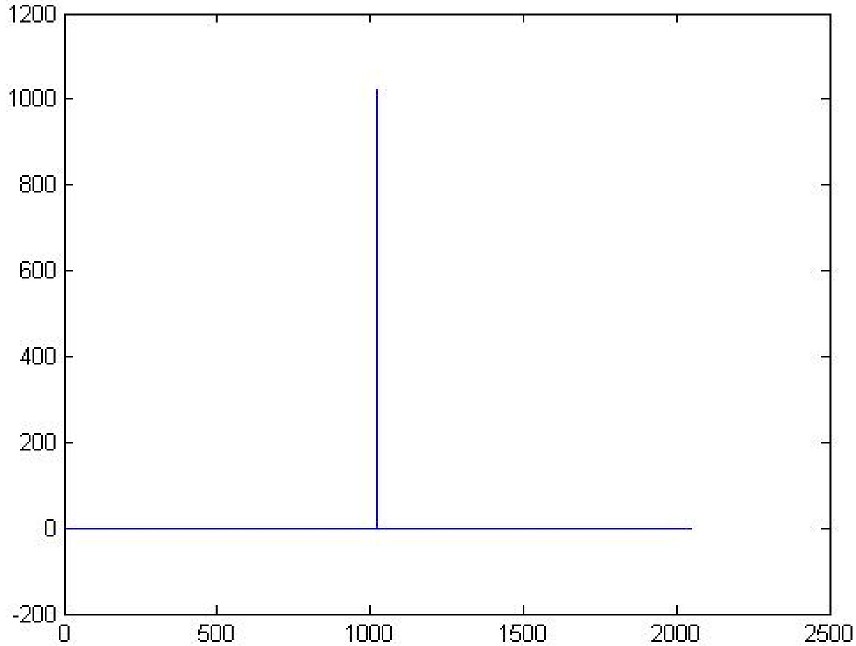
To show its “pseudorandom” nature, we zoom-in between output “chips” 100 and 200:



Next we carry out the autocorrelation function in MATLAB. Some care is required here since the autocorrelation sequence is, by definition, twice as long as the input sequence. So we have to repeat the input sequence three times as given in the following code.

```
prnprime=prn';
crosscorr1=[]
prn3 = [prn; prn; prn];
for icorr=-nprn+1:1:nprn-1
%crosscorr is the inner product of the row vector xprime and column vector prn3
%the length of both vectors is nprn
```

```
        crosscorr1(icorr+nprn) = prnprime(1:nprn)*prn3(nprn+icorr+1:2*nprn+icorr);
end
figure(2)
plot(crosscorr2)
```



On first glance, this appears to be a perfect “Kronecker” delta function. But closer inspection shows the finite “sidelobes” equal to -1, as predicted.