A Classical System for Producing “Quantum Correlations”

Robert H. McEachern

Abstract: It is almost universally supposed, that “Quantum Correlations”, as discussed in connection with Bell’s Inequality Theorem, cannot be produced by any classical, macroscopic system. Nevertheless, this paper demonstrates the actual construction of just such a system. It then discusses why this peculiar type of classical system, unlike any other, behaves in this “weird” fashion. The reason illuminates the Physics Community’s profound misunderstanding of exactly what a single, classical “bit” is, in the context of Shannon’s Information Theory, and the resulting misinterpretation of the Heisenberg Uncertainty Principle and the EPR paradox.

Summary: The two figures shown below, depict the “quantum correlations”, computed between randomized pairs of polarity measurements, made on two sets of images of “entangled coins”, that have polarized, noisy, band-limited, surface features. Each set consists of 500,000 pairs, of oppositely-polarized (entangled) coin images. Figure 1 (red) depicts the observed correlations obtained, when the surface noise on each coin is low enough to enable several bits of information to be extracted from each image. In both figures, the blue curve is the idealized “quantum correlation”.

Figure 2 (red) depicts the same correlations, when the surface noise on each coin is increased enough to enable only a single bit of information to be extracted from each image. Figure 1, is exactly as expected for a classical system[1]. Figure 2 is not. Yet the only significant difference in generating the two figures, is that one set of images had a significantly higher, but carefully selected, level of noise, than the other, along with resulting “missed-detections”. These critical differences elucidate a common, but profound misunderstanding of the nature of a “bit” of information, within the physics community.

When only a single bit of information exists within any received “message”, such as a physical particle or image, then only a single independent measurement, can ever be made on that “message”. Every subsequent attempt at another measurement, even of a supposedly independent variable, *must* be correlated with the first measurement, unlike the ubiquitous cases in which more than a single bit of information exists. The so-called “quantum correlations” are nothing more than single-bit correlations, whereas “classical correlations” result from multi-bit correlations. But true, single-bit entities, are extremely rare in the classical realm and thus unfamiliar.

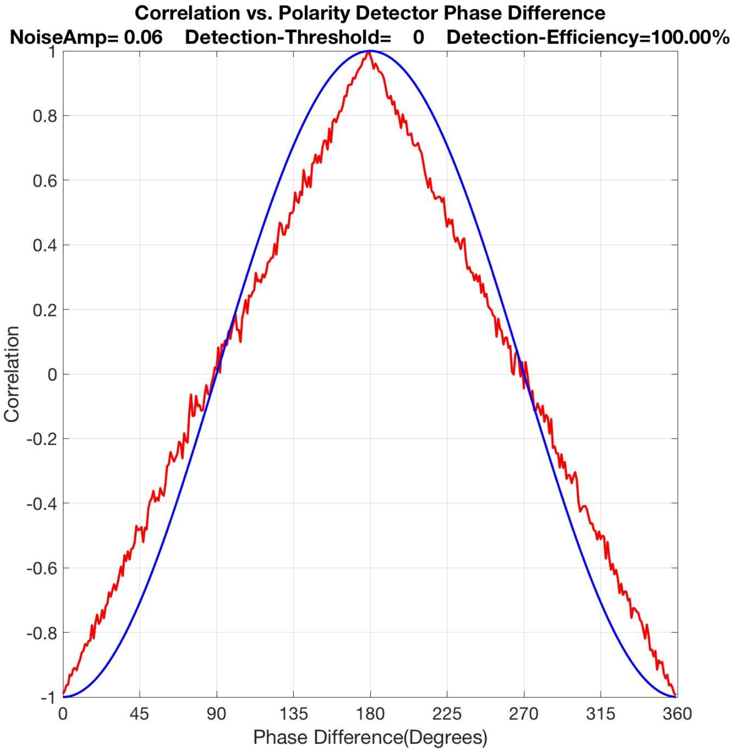


Figure 1

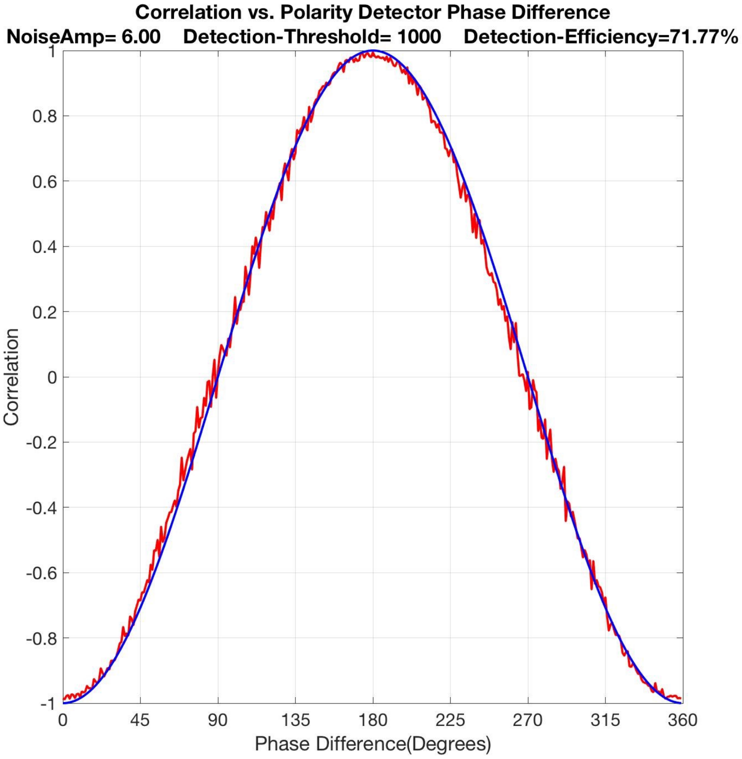


Figure 2

Figure 3

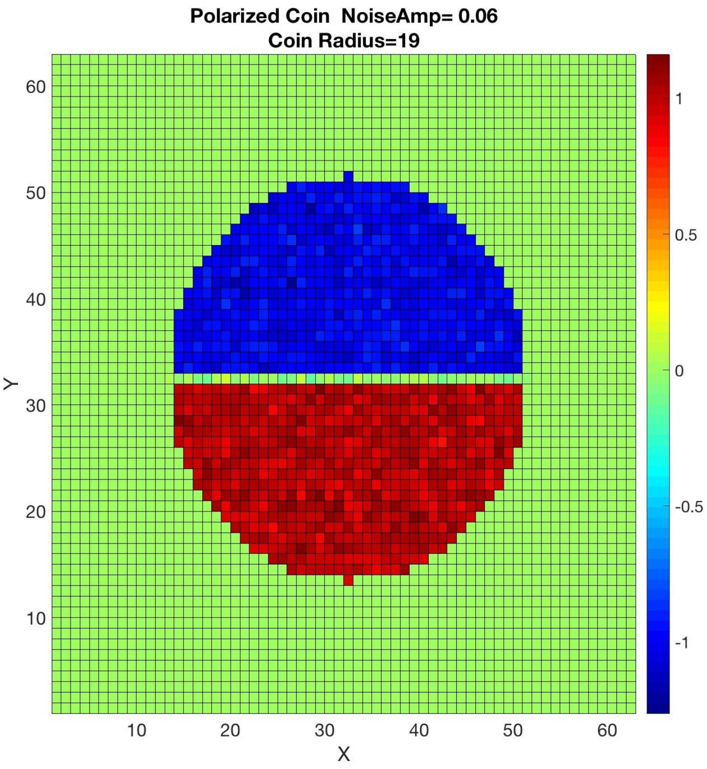
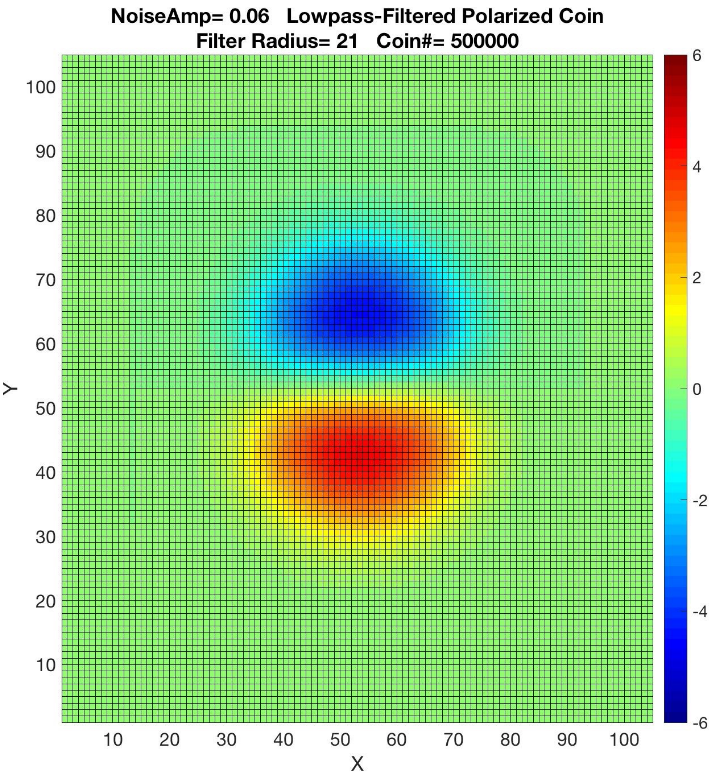


Figure 4



The previous two figures were produced by the appended MATLAB[2] simulation. To understand the nature of the phenomenon being produced, consider the two figures depicted above. Figure 3 depicts one of the 1,000,000 coin images, used to generate Figure 1. It shows the surface of the coin, polarized such that the top half has a value of -1 and the bottom half has a value of +1. The "entangled coin" has the opposite polarity. A small amount of random noise has been added. Each coin is identical, except for the added noise. Figure 4 depicts the same coin, after it has been lowpass filtered, to blur the image, and thus reduce its information content, using the two-dimensional, filter shown in Figure 5. The filtering is accomplished by performing a two dimensional convolution of each coin image with the lowpass filter.

In accordance with Shannon's Capacity Theorem, the information content of each image can be controlled by adjusting the added noise level and the bandwidth of the lowpass filter.

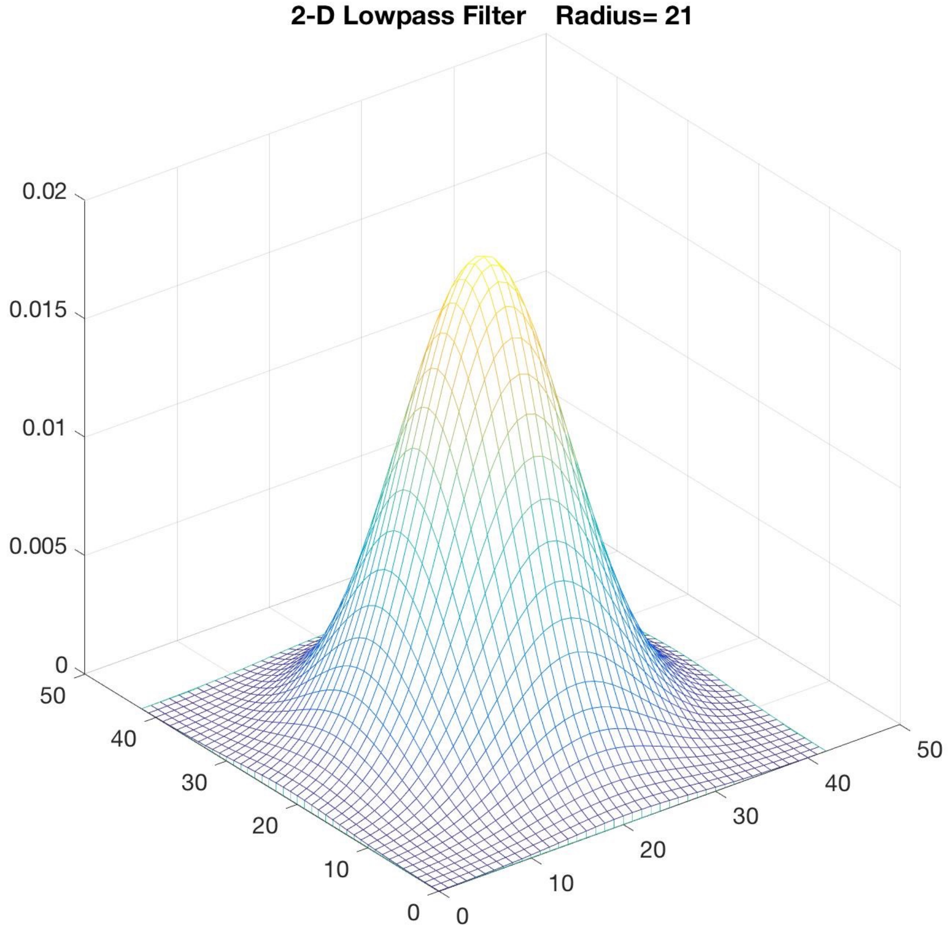
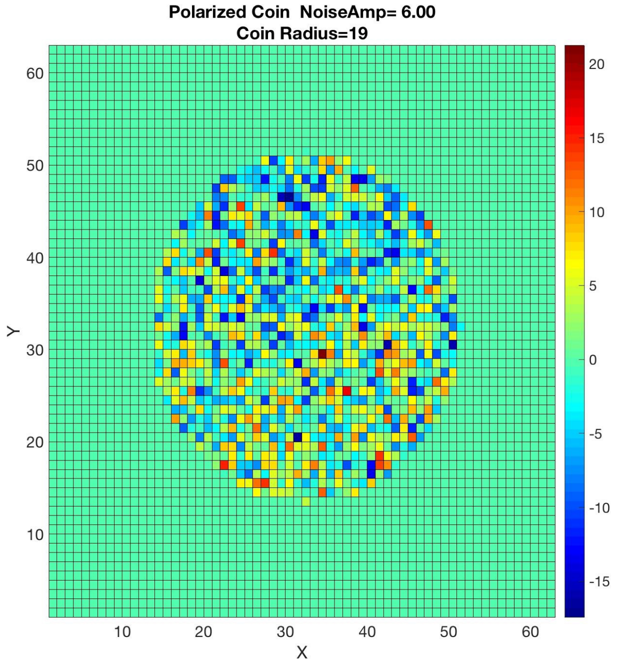
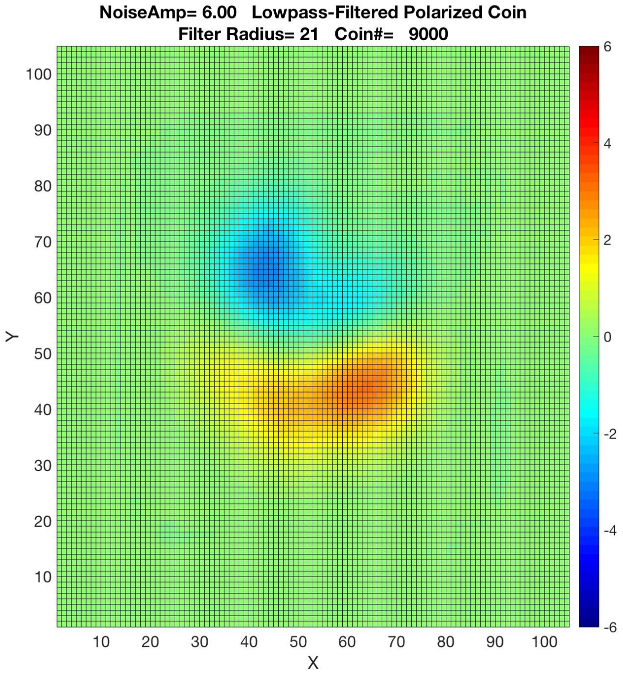


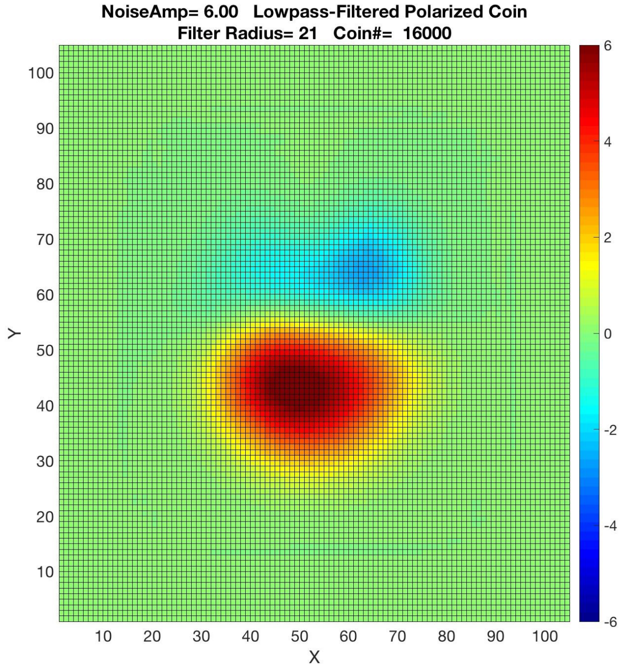
Figure 5

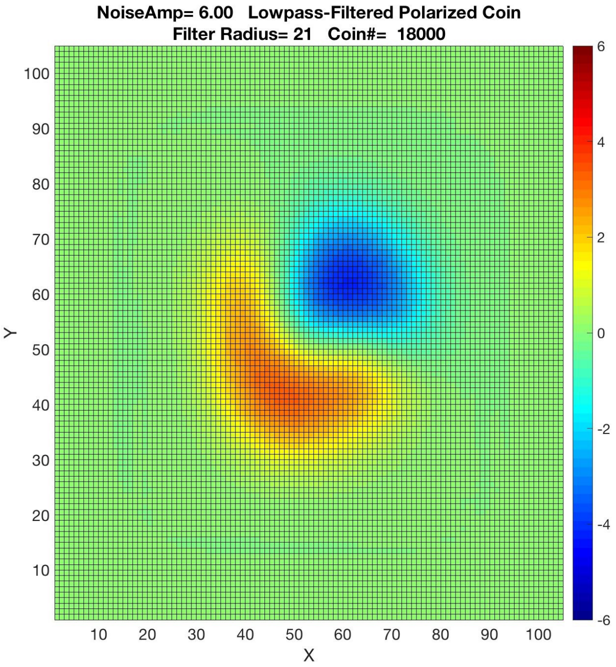
It is important to recognize, that unlike the more familiar cases of communication channels and measurement theory, the noise and bandwidth being simulated, are *not* properties of either the channel or measurement process. Instead, they are *intrinsic* to the surface of each coin. In other words, the coins have irregular surfaces. Hence, the limitations imposed by the noise and bandwidth (limited information content), cannot be altered by any measurement process.

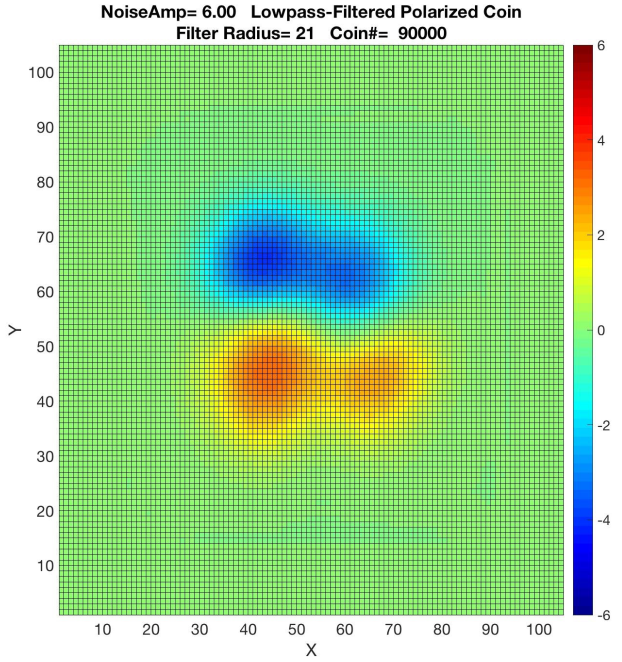
The figure on the right, depicts an unfiltered coin, with the higher noise level, used to produce Figure 2. Note that due to the noise, the polarization is not obvious. The four figures below depict four typical, lowpass-filtered coin images. The polarization is apparent, but too noisy to enable any polarization measurements, with more accuracy than approximately one, significant bit. It is this property, which results in the "quantum correlations".











The polarization is measured by multiplying each coin's image, by an angle-randomized "matched filter", such as that depicted in figure 11, then integrating over the resultant image. In other words, each pixel in the coin's image, is multiplied by the corresponding pixel in the matched filter, then all the pixels are summed together and the polarization is declared to be +1, if the sum is positive, and -1 if the sum is negative. In the simulation, there are 360 precomputed matched filters, corresponding to one degree phase increments in the polarization.

The phase difference between the pair of randomly selected matched filters, used to determine the polarity of each coin within an "entangled" pair, is then computed. Next, histograms are computed, for each possible entangled-pair polarity outcome (+ +, + -, - -, - +) versus the phase difference. The histograms are then used to compute the correlation versus phase difference, shown in Figures 1 and 2. The correlation at each angle, a, is computed by:

$$N=HistUU(a)+HistDD(a)+HistDU(a)+HistUD(a)$$
$$Corr(a)=(HistUU(a)+HistDD(a)-HistDU(a)-HistUD(a))/N$$

"D" and "U" refer to Down and UP polarizations respectively.

Note that Figure 1 was computed with a "Detection-Threshold" of 0.0, resulting in a "Detection-Efficiency" of 100%, whereas Figure 2 had a Threshold of 1000 and an Efficiency of about 72%. Figure 12 depicts the same noise level as figure 2, but with the Threshold=0 and a 100% Efficiency. The significance of these parameters is as follows:

In any real experiment, the detectors will not be able to detect the existence of every particle or coin-image. Hence, the Detection-Efficiency will be less than 100%. In the simulation, it is possible to detect every particle; that is what is depicted in Figures 1 and 12. However, when only a single bit of information can ever be extracted from a received particle/image, the detection of the particle/image existence, cannot be separated from the detection of its polarity; they are one and the same thing. At polarity detection angles of 90 and 270 degrees, the correlation is zero. That means there is no "Signal" to detect. There is only "Noise".

Consequently, if a threshold is set, such that the absolute value of the correlation, as measured by the matched filters, must exceed the threshold, for *both* images in an entangled pair, then the Detection-Efficiency will drop below 100%. But contrary to popular belief, there is no "Fair Sampling[3]"; The apparatus will systematically fail to detect only those entities with near zero correlation (no signal), which contribute nothing but noise variance, to the numerator in the detection correlation equation given above, but reduces the denominator in that equation, since that is simply equal to the number of detected entities.

Thus, when all the coins are detected (Threshold=0), "bit errors" are made, which lower the correlation peaks to values below +1 and -1. But the thresholding effectively renormalizes the correlations, by selectively failing to detect only the low correlation value polarities.

Figure 11

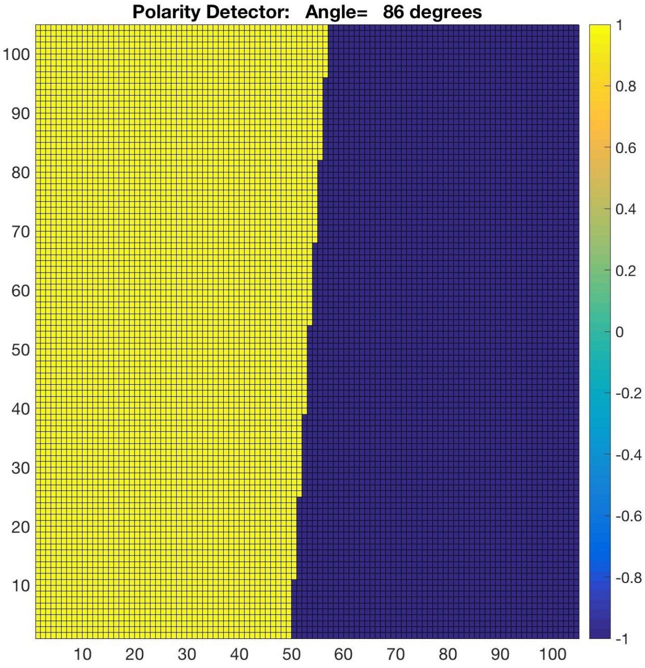
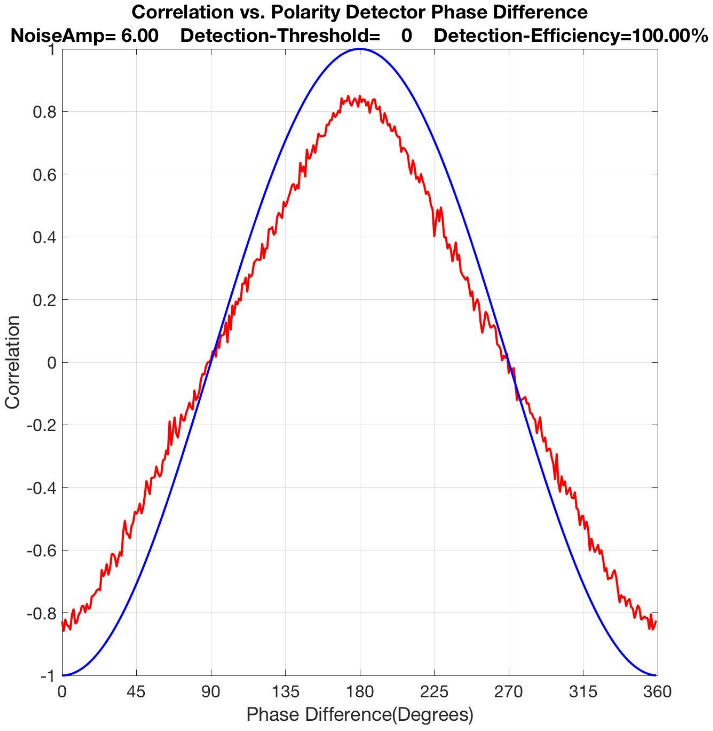


Figure 12



That is the difference between Figures 2 and 12; they have the same noise level, but the thresholding reduces the denominator in the correlation computation, without significantly altering the numerator. The Numerator effects the shape of the correlation curve, but the denominator only effects its normalization. Hence this "Unfair Sampling" effect, peculiar to the one-bit information content of the measurements, results in the Classical Correlation curve reproducing the "Quantum Correlation" curve.

Shannon, Heisenberg and Bell

More generally, this "single bit of information" phenomenon, is at the heart of the correct interpretation of the Heisenberg Uncertainty Principle, of which the EPR paradox[4] and Bell's Theorem are merely special cases. The uncertainty principle, in turn, is merely the special case of Shannon's Capacity Theorem[5], which marks the transition from the classical to the quantum realm, via the transition from measurements capable of recovering multiple bits of information (classical realm) to measurements in which only a single bit of information can ever be recovered (quantum realm).

Once it is realized that some entities only contain a single bit of recoverable information, it becomes obvious why two variables, such as those appearing in the uncertainty principle, cannot be simultaneously measured; measuring two independent (uncorrelated) variables would require extracting at least two bits of information, one for each variable, from an entity that only contains one: a self-contradiction.

Shannon's Capacity Theorem is usually discussed in regards to the number of states a system may have, or its entropy. However, there is a much simpler way to understand its significance. Shannon's Capacity, states that there is a maximum number of bits of information that can ever be recovered from a continuous "signal":

$$\text{Max number of bits of information} = T\ B\ \log_2(1+S/N)$$

where T is the signal's duration, B is the filter bandwidth and S/N is the signal-to-noise ratio.

This limit exists, because the maximum number of recoverable bits of information, cannot exceed the number of bits of data, within a set of discrete, sample measurements, that is sufficient for perfectly reconstructing the continuous signal. That maximum number of bits, is simply equal to the product of the number of required samples (TB), multiplied by the number of bits per sample ($\log_2(1+S/N)$), required to encode all the "significant" bits, above the noise level.

The maximum number of samples required is limited, because of the limited bandwidth of the signal[6]; a band-limiting filter introduces correlations between closely-spaced measurements, such that any more closely spaced measurements cannot be independent, from those sufficient for perfect reconstruction. Similarly, the noise limits the number of bits required per sample.

With these facts in mind, it is obvious that the limiting case is a signal that can be reconstructed from a single sample (T B = 1), with a single significant bit per sample ($\log_2(1+S/N) = 1$). The limiting case is thus, T B =1. This is the Uncertainty Principle. To see this, in the case of a photon, consider that:

$$c\,\Delta t = \Delta x = h/\Delta p = \Delta\lambda = c/\Delta f, \text{ hence:}$$

$$\Delta t=1/\Delta f, \text{ or } \Delta t\ \Delta f =1, \text{ or, in the notation above } T\ B =1$$

In other words, the Heisenberg Uncertainty Principle simply states that once the bandwidth and noise intrinsic to the object being measured is such that only a single bit of information can ever be recovered from any measurements of the object, then all classical independent variables, like position and momentum and multiple spin components, can no longer be independent - because they have become correlated, as the result of the limit on the information content. This is the nature of the correlations being characterized by Bell's Theorem. It has nothing to do with "spooky action at a distance", since it is simply the consequence of an intrinsically limited information content - the defining difference between the "classical" and the "quantum".

Identical Particles

Line number 82 in the script is commented out. If it is uncommented and executed, the "Quantum Correlations" disappear. Even the noise is now identical, except for a sign, for each entangled pair; causing even the bit errors (bad polarity decisions) to now be perfectly correlated. This does not happen because no bit errors are being made (as in figure 1), it happens because both detectors now *always* make *identical* errors. In other words, in the (incorrect) quantum interpretation, even when the actual detected polarity is not even a possible state of the noise-free wave-function (the detection was a total error) even the bogus detections must be perfectly correlated, and have nothing to do with the actual, noise-free wave-function, in order to explain the classical result.

This begs the question, "What makes identical particles, behave as if they are identical?" They cannot have identical noise (actually be identical), because that will fail to produce the observed quantum correlations. Thus, they are identical if and only if, only their recoverable information content is identical. If the intrinsic noise is "too identical", then they cannot behave like identical, quantum particles, instead, they will behave like identical, classical particles. Particles with more than one identical bit of information, are consequently, *too* identical to ever behave as identical, quantum particles.

Appendix A - MATLAB Simulation

The MATLAB script implementing the simulation appears in an appendix. Note that by changing the value of the variable "test", on line number 24, the three cases, used to generate Figures 1, 2 and 12, can be executed. The associated file "Quantum_Correlations.txt[7]" contains the actual script. It can be opened and viewed with any text editor. By changing the extension for ".txt" to ".m", it can be directly opened and executed in MATLAB.

[1] See for example https://en.wikipedia.org/wiki/Bell%27s_theorem

[2] http://www.mathworks.com/products/matlab/

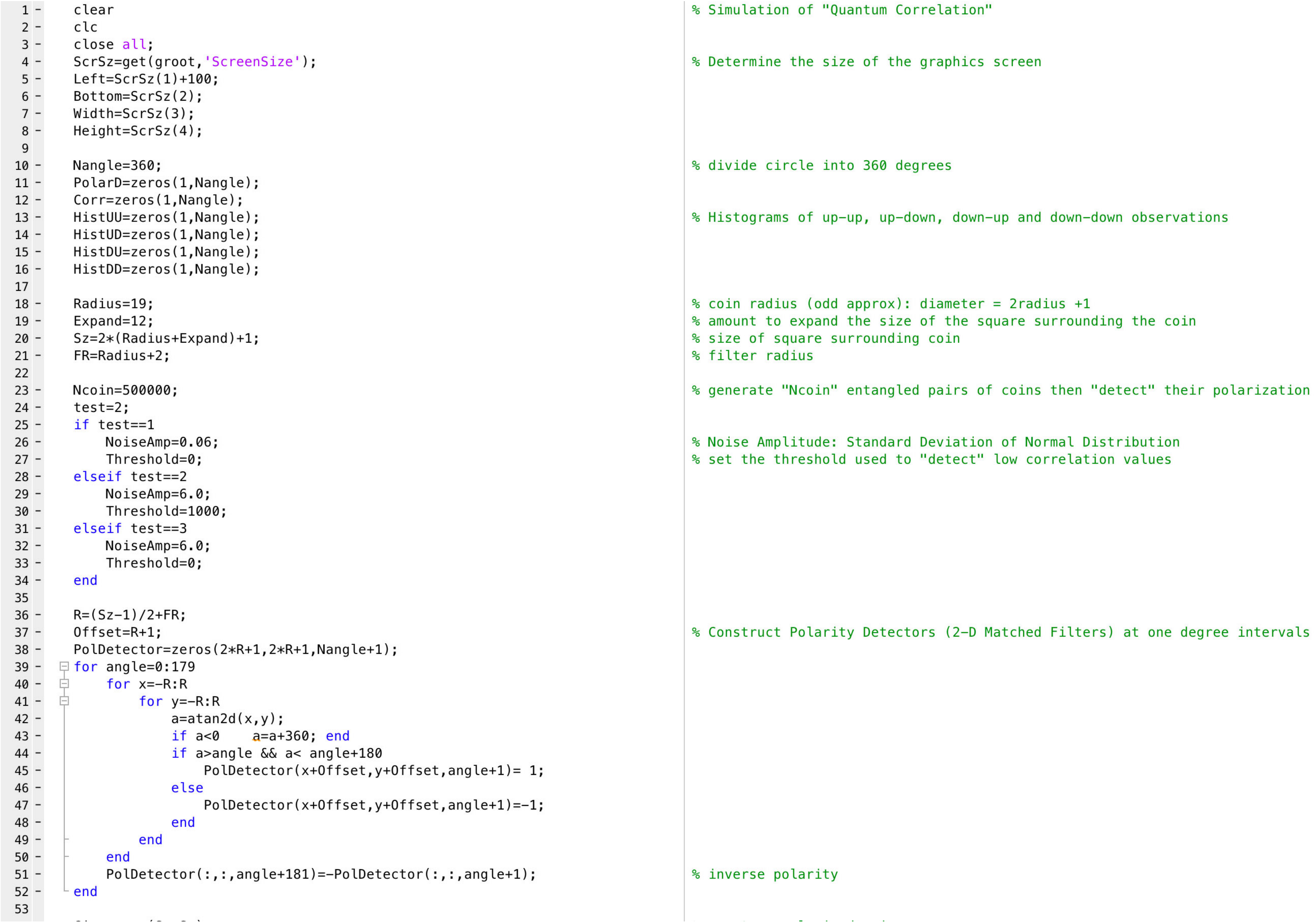
[3] https://en.wikipedia.org/wiki/Loopholes_in_Bell_test_experiments

[4] https://en.wikipedia.org/wiki/EPR_paradox

[5] https://en.wikipedia.org/wiki/Shannon%E2%80%93Hartley_theorem

[6] https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

[7] http://fqxi.org/data/forum-attachments/Quantum_Correlations.txt



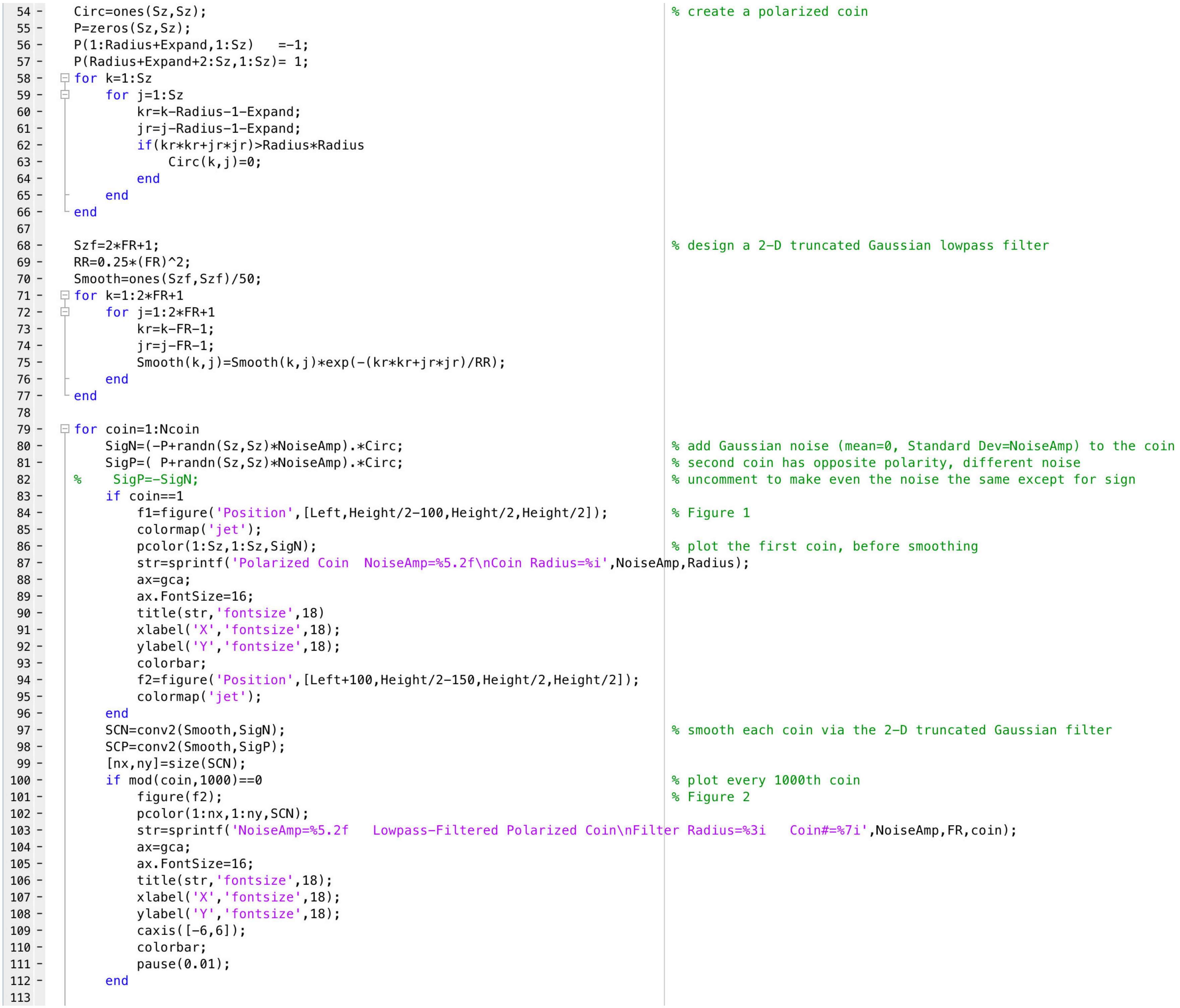
```matlab
clear                                                        % Simulation of "Quantum Correlation"
clc
close all;
ScrSz=get(groot,'ScreenSize');                               % Determine the size of the graphics screen
Left=ScrSz(1)+100;
Bottom=ScrSz(2);
Width=ScrSz(3);
Height=ScrSz(4);

Nangle=360;                                                  % divide circle into 360 degrees
PolarD=zeros(1,Nangle);
Corr=zeros(1,Nangle);
HistUU=zeros(1,Nangle);                                      % Histograms of up-up, up-down, down-up and down-down observations
HistUD=zeros(1,Nangle);
HistDU=zeros(1,Nangle);
HistDD=zeros(1,Nangle);

Radius=19;                                                   % coin radius (odd approx): diameter = 2radius +1
Expand=12;                                                   % amount to expand the size of the square surrounding the coin
Sz=2*(Radius+Expand)+1;                                      % size of square surrounding coin
FR=Radius+2;                                                 % filter radius

Ncoin=500000;                                                % generate "Ncoin" entangled pairs of coins then "detect" their polarization
test=2;
if test==1
    NoiseAmp=0.06;                                           % Noise Amplitude: Standard Deviation of Normal Distribution
    Threshold=0;                                             % set the threshold used to "detect" low correlation values
elseif test==2
    NoiseAmp=6.0;
    Threshold=1000;
elseif test==3
    NoiseAmp=6.0;
    Threshold=0;
end

R=(Sz-1)/2+FR;
Offset=R+1;                                                  % Construct Polarity Detectors (2-D Matched Filters) at one degree intervals
PolDetector=zeros(2*R+1,2*R+1,Nangle+1);
for angle=0:179
    for x=-R:R
        for y=-R:R
            a=atan2d(x,y);
            if a<0    a=a+360; end
            if a>angle && a< angle+180
                PolDetector(x+Offset,y+Offset,angle+1)= 1;
            else
                PolDetector(x+Offset,y+Offset,angle+1)=-1;
            end
        end
    end
    PolDetector(:,:,angle+181)=-PolDetector(:,:,angle+1);   % inverse polarity
end
```

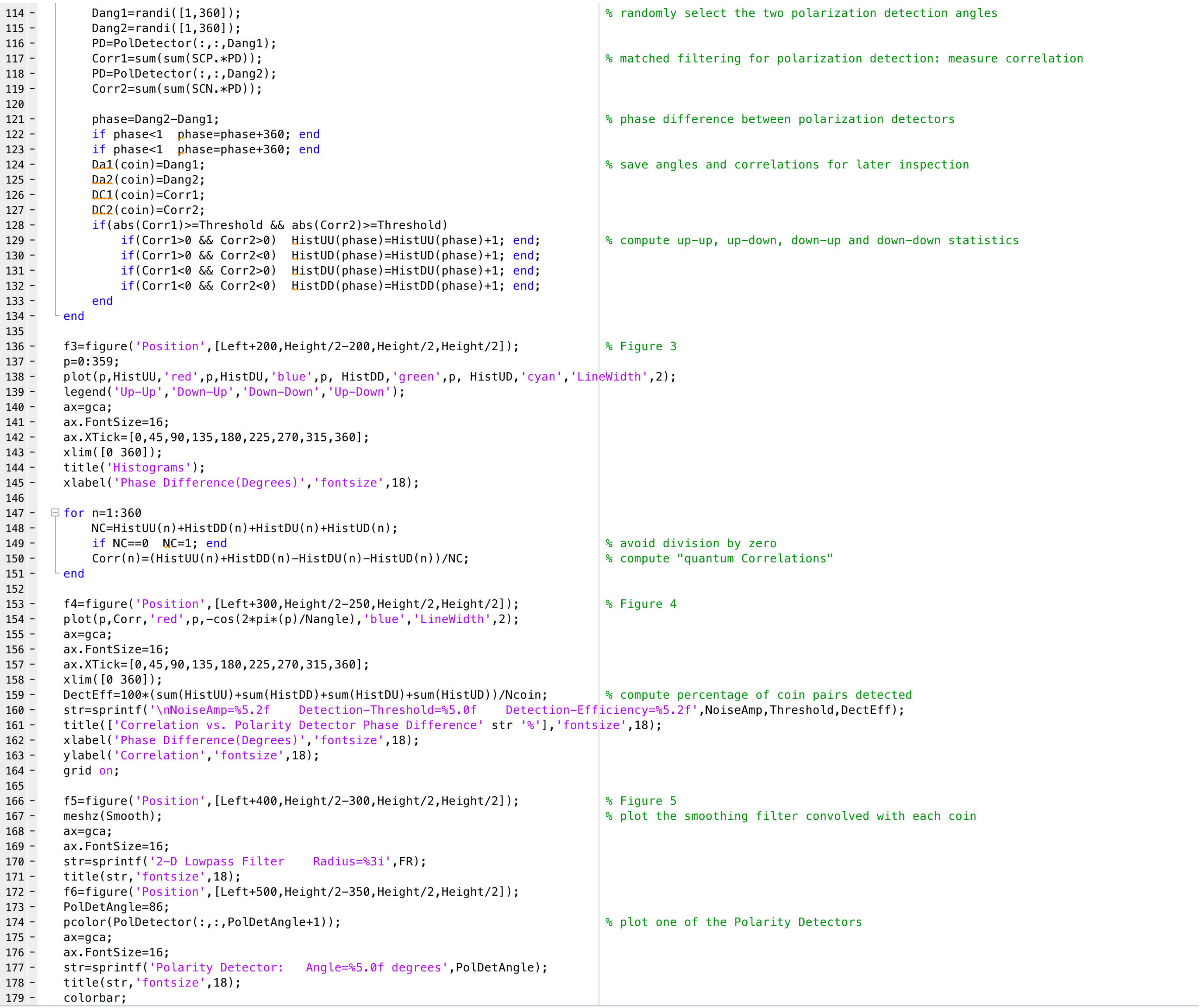


```matlab
Circ=ones(Sz,Sz);                                                           % create a polarized coin
P=zeros(Sz,Sz);
P(1:Radius+Expand,1:Sz)   =-1;
P(Radius+Expand+2:Sz,1:Sz)= 1;
for k=1:Sz
    for j=1:Sz
        kr=k-Radius-1-Expand;
        jr=j-Radius-1-Expand;
        if(kr*kr+jr*jr)>Radius*Radius
            Circ(k,j)=0;
        end
    end
end

Szf=2*FR+1;                                                                 % design a 2-D truncated Gaussian lowpass filter
RR=0.25*(FR)^2;
Smooth=ones(Szf,Szf)/50;
for k=1:2*FR+1
    for j=1:2*FR+1
        kr=k-FR-1;
        jr=j-FR-1;
        Smooth(k,j)=Smooth(k,j)*exp(-(kr*kr+jr*jr)/RR);
    end
end

for coin=1:Ncoin
    SigN=(-P+randn(Sz,Sz)*NoiseAmp).*Circ;                                  % add Gaussian noise (mean=0, Standard Dev=NoiseAmp) to the coin
    SigP=( P+randn(Sz,Sz)*NoiseAmp).*Circ;                                  % second coin has opposite polarity, different noise
%    SigP=-SigN;                                                            % uncomment to make even the noise the same except for sign
    if coin==1
        f1=figure('Position',[Left,Height/2-100,Height/2,Height/2]);        % Figure 1
        colormap('jet');
        pcolor(1:Sz,1:Sz,SigN);                                             % plot the first coin, before smoothing
        str=sprintf('Polarized Coin  NoiseAmp=%5.2f\nCoin Radius=%i',NoiseAmp,Radius);
        ax=gca;
        ax.FontSize=16;
        title(str,'fontsize',18)
        xlabel('X','fontsize',18);
        ylabel('Y','fontsize',18);
        colorbar;
        f2=figure('Position',[Left+100,Height/2-150,Height/2,Height/2]);
        colormap('jet');
    end
    SCN=conv2(Smooth,SigN);                                                 % smooth each coin via the 2-D truncated Gaussian filter
    SCP=conv2(Smooth,SigP);
    [nx,ny]=size(SCN);
    if mod(coin,1000)==0                                                    % plot every 1000th coin
        figure(f2);                                                         % Figure 2
        pcolor(1:nx,1:ny,SCN);
        str=sprintf('NoiseAmp=%5.2f   Lowpass-Filtered Polarized Coin\nFilter Radius=%3i   Coin#=%7i',NoiseAmp,FR,coin);
        ax=gca;
        ax.FontSize=16;
        title(str,'fontsize',18);
        xlabel('X','fontsize',18);
        ylabel('Y','fontsize',18);
        caxis([-6,6]);
        colorbar;
        pause(0.01);
    end

```



```
        Dang1=randi([1,360]);                                               % randomly select the two polarization detection angles
        Dang2=randi([1,360]);
        PD=PolDetector(:,:,Dang1);
        Corr1=sum(sum(SCP.*PD));                                            % matched filtering for polarization detection: measure correlation
        PD=PolDetector(:,:,Dang2);
        Corr2=sum(sum(SCN.*PD));

        phase=Dang2-Dang1;                                                  % phase difference between polarization detectors
        if phase<1  phase=phase+360; end
        if phase<1  phase=phase+360; end
        Da1(coin)=Dang1;                                                    % save angles and correlations for later inspection
        Da2(coin)=Dang2;
        DC1(coin)=Corr1;
        DC2(coin)=Corr2;
        if(abs(Corr1)>=Threshold && abs(Corr2)>=Threshold)
            if(Corr1>0 && Corr2>0)  HistUU(phase)=HistUU(phase)+1; end;     % compute up-up, up-down, down-up and down-down statistics
            if(Corr1>0 && Corr2<0)  HistUD(phase)=HistUD(phase)+1; end;
            if(Corr1<0 && Corr2>0)  HistDU(phase)=HistDU(phase)+1; end;
            if(Corr1<0 && Corr2<0)  HistDD(phase)=HistDD(phase)+1; end;
        end
end

f3=figure('Position',[Left+200,Height/2-200,Height/2,Height/2]);            % Figure 3
p=0:359;
plot(p,HistUU,'red',p,HistDU,'blue',p, HistDD,'green',p, HistUD,'cyan','LineWidth',2);
legend('Up-Up','Down-Up','Down-Down','Up-Down');
ax=gca;
ax.FontSize=16;
ax.XTick=[0,45,90,135,180,225,270,315,360];
xlim([0 360]);
title('Histograms');
xlabel('Phase Difference(Degrees)','fontsize',18);

for n=1:360
    NC=HistUU(n)+HistDD(n)+HistDU(n)+HistUD(n);
    if NC==0  NC=1; end                                                     % avoid division by zero
    Corr(n)=(HistUU(n)+HistDD(n)-HistDU(n)-HistUD(n))/NC;                   % compute "quantum Correlations"
end

f4=figure('Position',[Left+300,Height/2-250,Height/2,Height/2]);            % Figure 4
plot(p,Corr,'red',p,-cos(2*pi*(p)/Nangle),'blue','LineWidth',2);
ax=gca;
ax.FontSize=16;
ax.XTick=[0,45,90,135,180,225,270,315,360];
xlim([0 360]);
DectEff=100*(sum(HistUU)+sum(HistDD)+sum(HistDU)+sum(HistUD))/Ncoin;        % compute percentage of coin pairs detected
str=sprintf('\nNoiseAmp=%5.2f    Detection-Threshold=%5.0f    Detection-Efficiency=%5.2f',NoiseAmp,Threshold,DectEff);
title(['Correlation vs. Polarity Detector Phase Difference' str '%'],'fontsize',18);
xlabel('Phase Difference(Degrees)','fontsize',18);
ylabel('Correlation','fontsize',18);
grid on;

f5=figure('Position',[Left+400,Height/2-300,Height/2,Height/2]);            % Figure 5
meshz(Smooth);                                                              % plot the smoothing filter convolved with each coin
ax=gca;
ax.FontSize=16;
str=sprintf('2-D Lowpass Filter    Radius=%3i',FR);
title(str,'fontsize',18);
f6=figure('Position',[Left+500,Height/2-350,Height/2,Height/2]);
PolDetAngle=86;
pcolor(PolDetector(:,:,PolDetAngle+1));                                     % plot one of the Polarity Detectors
ax=gca;
ax.FontSize=16;
str=sprintf('Polarity Detector:   Angle=%5.0f degrees',PolDetAngle);
title(str,'fontsize',18);
colorbar;
```