Note: all the incoming states mentioned below will be entangled states that are not detected by $\{W\}$.

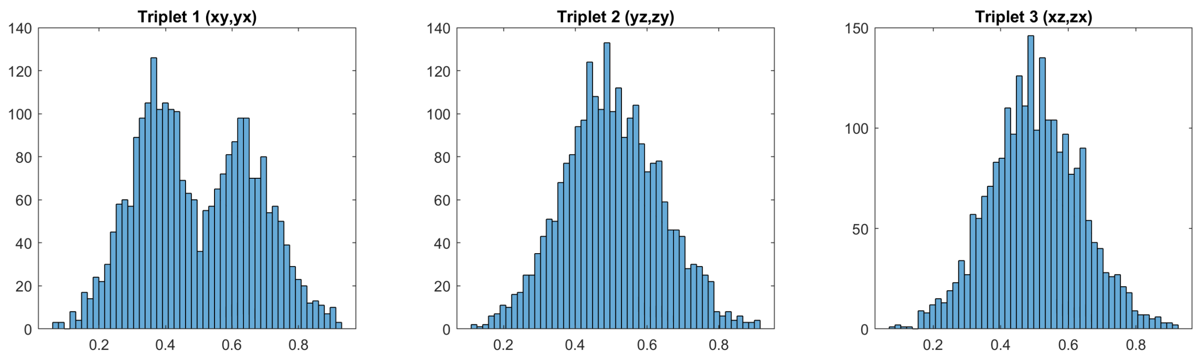
The population method

Reference: **"Improved entanglement detection with subspace witnesses"**

The sum of populations in the $|k\rangle$,$|\bar{k}\rangle$ subspace is

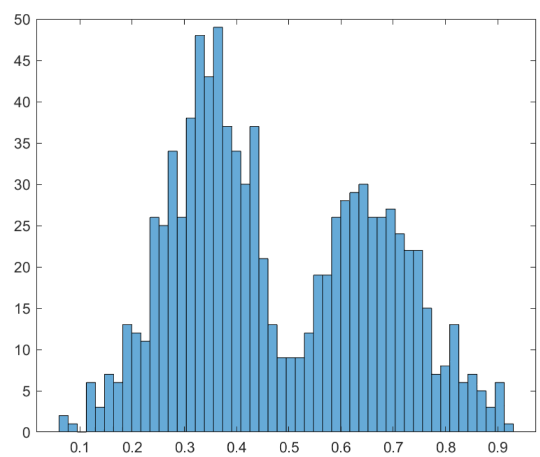
$$P = \rho_{kk} + \rho_{\bar{k}\bar{k}}.$$

For $|k\rangle = |00\rangle$, the distribution of $P = \rho_{|00\rangle|00\rangle} + \rho_{|11\rangle|11\rangle}$, of the states detected by each triplet is

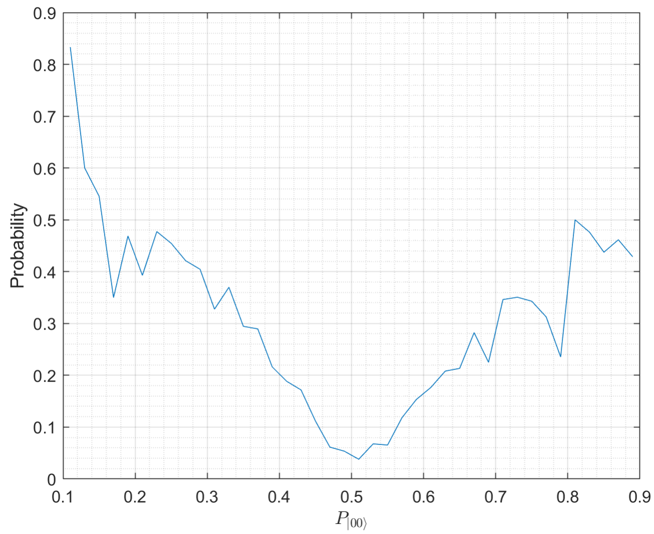


The distributions for $|k\rangle = |01\rangle$ will be the inversion about $P = 0.5$, which will look the same as the figure above.

Sometimes a state is detected by more than one triplet. For states that can be detected by ONLY Triplet 1, the count around $P = 0.5$ is very low.



If we normalize the above distribution over all the states that are detected by any of the triplets (bin size = 0.01), we get the probability that an incoming state is detected by only Triplet 1 as a function of P.

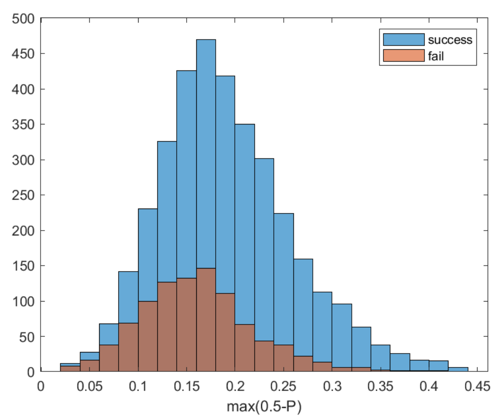


If we instead calculate the population in DA/RL basis, Triplet 2/3 will exhibit the same pattern.

Testing the performance

Plan 1: Calculate P_{HV}, P_{DA}, and P_{RL}. If P_{HV} is furthest away from 0.5, i.e., P_{DA} and P_{RL} are closer to 0.5, that means Triple 2 and 3 have lower probabilities to detect entanglement. Hence we want to choose Triplet 1.

Result:



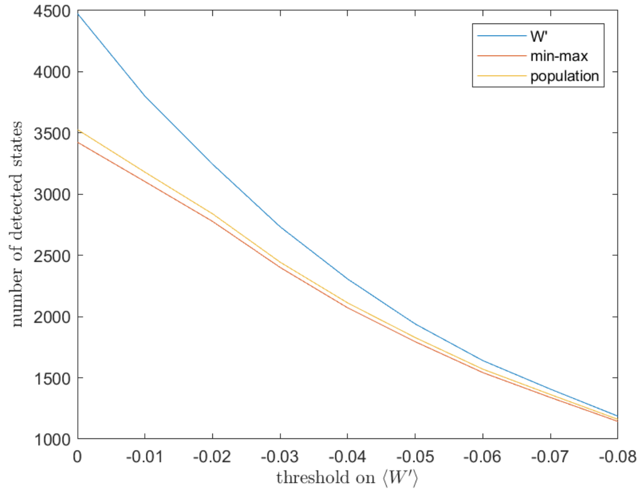
This method fails to detect 30.39% of the incoming entangled states, which is still 10% away from the ideal value.

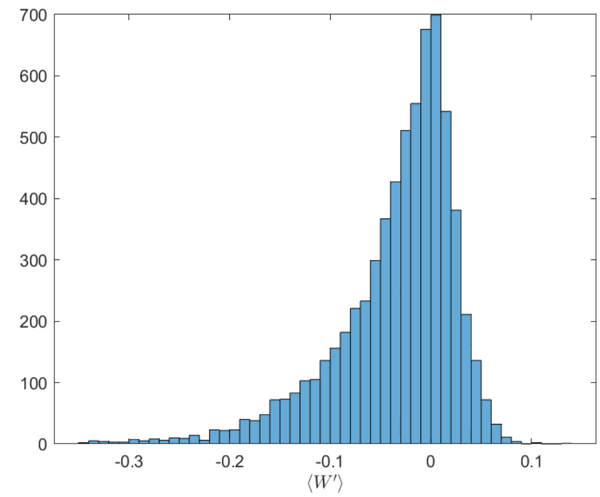
Plan 2: Check how much the predictions given by this population method overlap with the predictions given by the min-max method. If not much, we can weigh the two methods and achieve higher success detection probability.

Result: Out of the 6566 entangled states undetected by $\{W\}$, we can ideally detect 4475 of them if we pick the triplet correctly each time. The prediction success rate by the two methods are compared below

# predicted by min-max	# predicted by population	# predicted by either method	overlapping
3424	3527	3644	3100

We also guess that if the witness value to a state is only slightly negative, i.e., $\langle W' \rangle \approx 0$, then information will be lost when we apply the analytic methods and they will be equally bad at predicting the triplet. We show below how the relative success rate changes when we apply a minimum threshold value on the witness value.





Plan 3: We estimate the PFDs of the population distribution for each basis through curve-fitting (e.g. the graph above `Plan 1`). For any incoming state, we obtain the relative probabilities for each triplet using the PDFs and choose the one with the highest probability.

	HV basis	DA basis	RL basis
Triplet 1	$0.51\lvert\cos(2.97x)\rvert + 0.03$	$-0.41\lvert\cos(3.12x) + 0.33$	$-0.46\lvert\cos(3.12x)\rvert + 0.34$
Triplet 2	$-0.34\lvert\cos(3.03x)\rvert + 0.29$	$0.49\lvert\cos(3.15x)\rvert + 0.03$	$-0.31\lvert\cos(3.14x)\rvert + 0.26$
Triplet 3	$-0.35\lvert\cos(2.99x)\rvert + 0.29$	$-0.30\lvert\cos(3.08x)\rvert + 0.26$	$0.58\lvert\cos(3.15x)\rvert + 0.01$

Result: # predicted states ~ 3500. We do not see any improvement to the success rate.

So far we have seen that the population method performs slightly better than the min-max method and has more reasoning. However, all the attempts give similar success rate, and the detected states largely overlap. It seems possible that we are close to a limit.

Finding a limit to how good we can predict

A thought experiment:

For any incoming state, the 9 Stokes parameters (or equivalently the 12 measurement results) will be all the information we know. Assume that we have an infinite database of entangled states and their triplet information, and there are N states in the database that have exactly the same 9 Stokes parameters as our incoming state. If Triplet 1 detects our incoming state, and it detects n of N states in the database, then a reasonable upper bound in our ability to predict the correct triplet is n/N.

Plan 1: We create a database of 10,000 known entangled states and the corresponding triplets that detect them. For any incoming state, we construct the incomplete density matrix with the known Stokes parameters from the first 12 measurements:

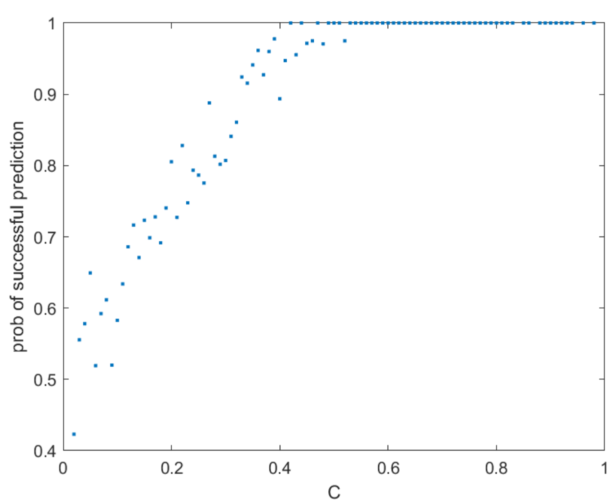
$$\rho_{part} = \frac{1}{4}(\mathbb{I} + S_{xx}\sigma_{xx} + S_{yy}\sigma_{yy} + S_{zz}\sigma_{zz} + S_{ix}\sigma_{ix} + S_{xi}\sigma_{xi} + S_{iy}\sigma_{iy} + S_{yi}\sigma_{yi} + S_{iz}\sigma_{iz} + S_{zi}\sigma_{zi})$$

We evaluate the closeness of the incoming state to any state ζ_i in our database by calculating the matrix distance between ρ_{part} and the partial density matrix $\zeta_{part,i}$ (also constructed by only 9 Stokes parameters):

$$d_i = ||\zeta_{part,i} - \rho_{part}||.$$

We then pick out the three closest states in the database to the incoming state, check their triplet information and determine the most probable one.

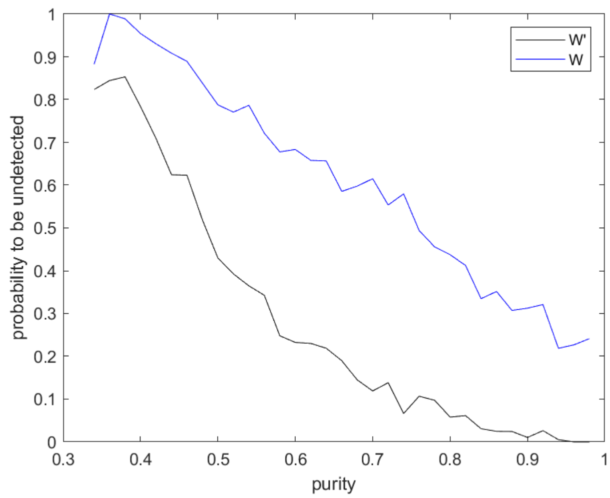
Result: # predicted ~3400 :(It is important to note that we do not have a good reasoning on how to calculate the "closeness" of two states. Using the matrix distance seems natural, but this might not give the best weighing of the Stokes parameters. But figuring out how to weigh the parameters will lead to a gradient descent problem or back to a machine learning problem, and we do not really want to proceed in this direction.



Plan 2: A more direct approach to this problem is to construct all the possible density matrices given the values of the 9 Stokes parameters. This would be possible if we can find a way to parametrize the Stokes parameters, or, equivalently, find their dependence on each other.

Result: There is no explicit way to parameterize the Bloch vector/Stokes parameters because it is a 15-dimensional space. Bayesian inference might be helpful, but it requires some known probability distribution of the density matrices.

P.S. If our random state generation method is biased towards low-purity states, then we would expect the overall detection rate to be lower.



Documentation of the code

Write P and C_{max} to `entry_info.mat`

```
load('C:\Users\qiyan\Desktop\Quantum Optics\Fall 2022\Fall 2022 QO Data\detectedTriplet.mat')
load('C:\Users\qiyan\Desktop\Quantum Optics\Fall 2022\Fall 2022 QO Data\undetectedW_6566.mat')

syms t real

entry_info = zeros(6566,7);    % P1, C1_max, P2, C2_max, detected triplet
for i = 1:6566
    rho = reshape(undetectedW(:,i),4,4);
    P1 = rho(1,1) + rho(4,4);
    C1 = real(rho(1,4))*cos(t) + imag(rho(1,4))*sin(t);
    nf1 = matlabFunction(-1*C1);
    [~,fmax] = fminbnd(nf1,-pi,pi);
    C1_max = -fmax;

    P2 = rho(2,2) + rho(3,3);
    C2 = real(rho(2,3))*cos(t) + imag(rho(2,3))*sin(t);
    nf2 = matlabFunction(-1*C2);
```

```
    [~,fmax] = fminbnd(nf2,-pi,pi);
    C2_max = -fmax;

    entry_info(i,:) = real([P1, C1_max, P2, C2_max, detectedTriplet(i,:)]);
end
```

Plot histogram of `entry_info.mat` for three triplets

```
load('C:\Users\qiyan\Desktop\Quantum Optics\Fall 2022\Fall 2022 QO Data\detectedTriplet.mat')
load('entry_info.mat')

info_1 = zeros(6566,4);
info_2 = zeros(6566,4);
info_3 = zeros(6566,4);
for i = 1:6566
    if detectedTriplet(i,1) == 1
        info_1(i,:) = entry_info(i,1:4);
    end
    if detectedTriplet(i,2) == 1
        info_2(i,:) = entry_info(i,1:4);
    end
    if detectedTriplet(i,3) == 1
        info_3(i,:) = entry_info(i,1:4);
    end
end
info_1( all(~info_1,2), : ) = [];
info_2( all(~info_2,2), : ) = [];
info_3( all(~info_3,2), : ) = [];

figure('Name','P(|00>)');
subplot(1,3,1);
histogram(info_1(:,1),50)
title('Triplet 1 (xy,yx)')
subplot(1,3,2);
histogram(info_2(:,1),50)
title('Triplet 2 (yz,zy)')
subplot(1,3,3);
histogram(info_3(:,1),50)
title('Triplet 3 (xz,zx)')
```

Extract data from histogram and plot probability against P

```
load('C:\Users\qiyan\Desktop\Quantum Optics\Fall 2022\Fall 2022 QO Data\detectedTriplet.mat')
load('entry_info.mat')
info_1 = zeros(6566,1);
info_all = zeros(6566,1);
for i = 1:6566
    if detectedTriplet(i,1) == 1 && detectedTriplet(i,2) == 0 && detectedTriplet(i,3) == 0
        info_1(i,1) = entry_info(i,1);
    end
    if detectedTriplet(i,1) == 1 || detectedTriplet(i,2) == 1 || detectedTriplet(i,3) == 1
```

```
        info_all(i,1) = entry_info(i,1);
    end
end
info_1( all(~info_1,2), : ) = [];
info_all( all(~info_all,2), : ) = [];
edges = 0.1:0.02:0.9;
[CDF1, ~] = histcounts(info_1,edges);
[CDF2, ~] = histcounts(info_all,edges);
plot(edges(1:size(CDF1,2)) + 0.01*ones(size(CDF1)), CDF1./CDF2)
```

Predicted triplets by the min-max method and the population method

```
load('C:\Users\qiyan\Desktop\Quantum Optics\Fall 2022\Fall 2022 QO Data\detectedTriplet.mat')
load('C:\Users\qiyan\Desktop\Quantum Optics\Fall 2022\Fall 2022 QO Data\undetectedW_6566.mat')

prediction = zeros(6566,4);
for i = 1:6566
    if detectedTriplet(i,1) == 1 || detectedTriplet(i,2) == 1 || detectedTriplet(i,3) == 1
        rho = reshape(undetectedW(:,i),4,4);
        pred_minMax = minMaxPrediction(rho);
        pred_pop = popPrediction(rho);
        prediction(i,:) = [pred_minMax, pred_pop, ...
            detectedTriplet(i,pred_minMax)==1, ...
            detectedTriplet(i,pred_pop)==1];
    end
end
prediction( all(~prediction,2), : ) = [];

function r = rotateRhoBasis(rho, basis)
H = [1;0]; V = [0;1];
D = (H+V)/sqrt(2); A = (H-V)/sqrt(2);
R = (H+1i*V)/sqrt(2); L = (H-1i*V)/sqrt(2);

HH = kron(H,H); VV = kron(V,V); HV = kron(H,V); VH = kron(V,H);
HD = kron(H,D); VA = kron(V,A); HA = kron(H,A); VD = kron(V,D);
HR = kron(H,R); VL = kron(V,L); HL = kron(H,L); VR = kron(V,R);
DD = kron(D,D); AA = kron(A,A); DA = kron(D,A); AD = kron(A,D);
RR = kron(R,R); LL = kron(L,L); RL = kron(R,L); LR = kron(L,R);

if basis == 'DADA'
    r = [DD'*rho*DD, DD'*rho*DA, DD'*rho*AD, DD'*rho*AA;
         DA'*rho*DD, DA'*rho*DA, DA'*rho*AD, DA'*rho*AA;
         AD'*rho*DD, AD'*rho*DA, AD'*rho*AD, AD'*rho*AA;
         AA'*rho*DD, AA'*rho*DA, AA'*rho*AD, AA'*rho*AA];
elseif basis == 'RLRL'
    r = [RR'*rho*RR, RR'*rho*RL, RR'*rho*LR, RR'*rho*LL;
         RL'*rho*RR, RL'*rho*RL, RL'*rho*LR, RL'*rho*LL;
         LR'*rho*RR, LR'*rho*RL, LR'*rho*LR, LR'*rho*LL;
         LL'*rho*RR, LL'*rho*RL, LL'*rho*LR, LL'*rho*LL];
elseif basis == 'HVDA'
    r = [HD'*rho*HD, HD'*rho*HA, HD'*rho*VD, HD'*rho*VA;
```

```
        HA'*rho*HD, HA'*rho*HA, HA'*rho*VD, HA'*rho*VA;
        VD'*rho*HD, VD'*rho*HA, VD'*rho*VD, VD'*rho*VA;
        VA'*rho*HD, VA'*rho*HA, VA'*rho*VD, VA'*rho*VA];
elseif basis == 'HVRL'
    r = [HR'*rho*HR, HR'*rho*HL, HR'*rho*VR, HR'*rho*VL;
        HL'*rho*HR, HL'*rho*HL, HL'*rho*VR, HL'*rho*VL;
        VR'*rho*HR, VR'*rho*HL, VR'*rho*VR, VR'*rho*VL;
        VL'*rho*HR, VL'*rho*HL, VL'*rho*VR, VL'*rho*VL];
end
end

function triplet = minMaxPrediction(rho)
    [~,I] = maxW(rho);
    triplet = convertInd(I,1);
end

function triplet = popPrediction(rho)
    P_hv = real(rho(1,1) + rho(4,4));
    r_da = rotateRhoBasis(rho,'DADA');
    P_da = real(r_da(1,1) + r_da(4,4));
    r_rl = rotateRhoBasis(rho,'RLRL');
    P_rl = real(r_rl(1,1) + r_rl(4,4));

    [~,triplet] = max([abs(0.5-P_hv), abs(0.5-P_da), abs(0.5-P_rl)]);
end
```