In[ ]:=
```
SetDirectory[NotebookDirectory[]];
```

In[ ]:=
```
(** Data import **)
NamData0 = Import["Pure_data/Rhos.txt", "Table"];
proj = Import["ideal_proj.txt", "Table"];
```

In[ ]:=
```
Rho[i_] := {{NamData0[[4 + 16 i, 1]], NamData0[[5 + 16 i, 1]]}, {NamData0[[6 + 16 i, 1]], NamData0[[7 + 16 i, 1]]}} + {{NamData0[[9 + 16 i, 1]], NamData0[[10 + 16 i, 1]]}, {NamData0[[11 + 16 i, 1]], NamData0[[12 + 16 i, 1]]}} * I
RhoList1 = ParallelTable[Rho[i], {i, 0, NamData0[[1, 1]] - 1}];
```

In[ ]:=
```
RhoList1[[124]] // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 0.575889 + 3.8948 \times 10^{-18}\, i & -0.49419621742625829963202477301\,8 + 0.00297828012283533069895491429\,7\, i \\ -0.49419621742625841065432723553\,4 - 0.00297828012283533199999752128\,0\, i & 0.424111 - 3.8948 \times 10^{-18}\, i \end{pmatrix}$$

In[ ]:=
```
(* HVDARL states *)
StateH = {{1}, {0}};
StateV = {{0}, {1}};
StateD = (1/(2^(1/2))) {{1}, {1}};
StateA = (1/(2^(1/2))) {{1}, {-1}};
StateR = (1/(2^(1/2))) {{1}, {I}};
StateL = (1/(2^(1/2))) {{1}, {-I}};
States = {StateH, StateV, StateD, StateA, StateR, StateL};
Dimensions[States]
```

Out[ ]=
```
{6, 2, 1}
```

In[ ]:=
```
StateToRho[i_] := States[[i]].ConjugateTranspose[States[[i]]]
Rhos = ParallelTable[StateToRho[i], {i, 1, Dimensions[States][[1]]}];
Dimensions[Rhos]
```

Out[ ]=
```
{6, 2, 2}
```

In[ ]:=
```
(* Target sates *)Rho[i_] := {{proj[[4 + 11 i, 1]], proj[[5 + 11 i, 1]]}, {proj[[6 + 11 i, 1]], proj[[7 + 11 i, 1]]}} + {{proj[[9 + 11 i, 1]], proj[[10 + 11 i, 1]]}, {proj[[11 + 11 i, 1]], proj[[12 + 11 i, 1]]}} * I
RhoListProj = ParallelTable[Rho[i], {i, 0, proj[[1, 1]] - 1}];
Dimensions[RhoListProj]
```

Out[ ]=
```
{120, 2, 2}
```

In[ ]:=
```
RhoTarget = RhoListProj;
```

In[ ]:=
```
sigma2 = {{0, 1}, {1, 0}};
sigma3 = {{0, -I}, {I, 0}};
sigma1 = {{1, 0}, {0, -1}};
sigma = {sigma1, sigma2, sigma3};(* Sigma matrices *)
stokesStates[i_, x_] := ConjugateTranspose[States[[x]]].sigma[[i]].States[[x]]
stokeslistStates = ParallelTable[Re[stokesStates[i, x][[1, 1]]], {x, 1, Dimensions[States][[1]]}, {i, 1, 3}];
```

In[ ]:=
```
Fidelity[Rho1_, Rho2_, i_, j_] := Re[Tr[MatrixPower[MatrixPower[Rho1[[i]], 1/2].Rho2[[j]].MatrixPower[Rho1[[i]], 1/2], 1/2]]]^2
```

In[ ]:=
```
(** 120 states **)FidelityList1 = ParallelTable[Fidelity[RhoTarget, RhoList1, i, j], {j, 1, Dimensions[RhoList1][[1]]}, {i, 1, Dimensions[RhoTarget][[1]]}];
FidelityListTrans1 = Transpose[FidelityList1];
PosTable1 = ParallelTable[Position[FidelityListTrans1, Max[FidelityListTrans1[[i]]]], {i, 1, Dimensions[FidelityListTrans1][[1]]}];
BestFidelitys1 = ParallelTable[FidelityList1[[PosTable1[[i, 1, 2]], PosTable1[[i, 1, 1]]]], {i, 1, Dimensions[PosTable1][[1]]}];
```

```
In[ ]:= BestRhoList1 = ParallelTable[RhoList1[[PosTable1[[i, 1, 2]]]], {i, 1, Dimensions[PosTable1][[1]]}];

In[ ]:= RhoToStokes[Rho_, i_, j_] := Re[Tr[Rho[[i]].sigma[[j]]]]

In[ ]:= StokesList1 = ParallelTable[RhoToStokes[RhoList1, i, j], {i, 1, Dimensions[RhoList1][[1]]}, {j, 1, Dimensions[sigma][[1]]}];
        stokeslistStates = ParallelTable[RhoToStokes[Rhos, i, j], {i, 1, Dimensions[Rhos][[1]]}, {j, 1, Dimensions[sigma][[1]]}];
        BestStates1 = ParallelTable[RhoToStokes[BestRhoList1, i, j], {i, 1, Dimensions[BestRhoList1][[1]]}, {j, 1, Dimensions[sigma][[1]]}];

In[ ]:= StokesListTarget = ParallelTable[RhoToStokes[RhoTarget, i, j], {i, 1, Dimensions[RhoTarget][[1]]}, {j, 1, Dimensions[sigma][[1]]}];

In[ ]:= (** Fidelity **)

In[ ]:= BestFidelitys1;
        Max[BestFidelitys1]
        Min[BestFidelitys1]
        StandardDeviation[BestFidelitys1]

Out[ ]= 0.999997

Out[ ]= 0.994605

Out[ ]= 0.00114606

In[ ]:= Sum[BestFidelitys1[[i]], {i, 1, 120}] / Dimensions[BestFidelitys1][[1]]

Out[ ]= 0.99942

In[ ]:= 0.9994204546069704` s

Out[ ]= 0.99942 s

In[ ]:= (** Purity **)
        Purity[Rho_] := Re[Tr[MatrixPower[Rho, 2]]]

In[ ]:= BestPuritys1 = ParallelTable[Purity[BestRhoList1[[i]]], {i, 1, Dimensions[BestRhoList1][[1]]}];
        Max[BestPuritys1]
        Min[BestPuritys1]
        StandardDeviation[BestPuritys1]

Out[ ]= 0.999999

Out[ ]= 0.989427

Out[ ]= 0.00226655

In[ ]:= Sum[BestPuritys1[[i]], {i, 1, Dimensions[BestPuritys1][[1]]}] / Dimensions[BestPuritys1][[1]]

Out[ ]= 0.999063

In[ ]:= (** Angles **)

In[ ]:= AngleVec[u_, v_] := ArcCos[(u[[1]] * v[[1]] + u[[2]] * v[[2]] + u[[3]] * v[[3]]) / (Sqrt[u[[1]] ^ 2 + u[[2]] ^ 2 + u[[3]] ^ 2] * Sqrt[v[[1]] ^ 2 + v[[2]] ^ 2 + v[[3]] ^ 2])]
        Angles1 = Table[AngleVec[BestStates1[[i]], StokesListTarget[[i]]], {i, 1, 120}];

In[ ]:= StandardDeviation[Angles1];

In[ ]:= UnitConvert[% rad, "AngularDegrees"];
```

In[ ]:=
```
% / 2
```
Out[ ]=
```
0.270545°
```
In[ ]:=
```
Sum[Angles1[[i]], {i, 1, Dimensions[Angles1][[1]]}] / Dimensions[Angles1][[1]];
```
In[ ]:=
```
UnitConvert[% rad , "AngularDegrees "];
```
In[ ]:=
```
% / 2
```
Out[ ]=
```
0.535725°
```
In[ ]:=
```
(** Hammer projestion **)
GetSpherical[Stokes_] := {ArcTan[Sqrt[Stokes[[2]] ^ 2 + Stokes[[1]] ^ 2], Stokes[[3]]], Limit[ArcTan[x, Stokes[[2]]], {x -> Stokes[[1]]}]}
```
In[ ]:=
```
HammerCoordinates[stokes_] :=
 ArrayReshape[{(2 * Sqrt[2] * Cos[elev] * Sin[az / 2]) / Sqrt[1 + Cos[elev] Cos[az / 2]], (Sqrt[2] * Sin[elev]) / Sqrt[1 + Cos[elev] Cos[az / 2]]} /. {elev -> GetSpherical[stokes][[1]], az -> GetSpherical[stokes][[2]]}, {2}]
```

In[ ]:=
```
Show[ListPlot[
  Table[ArrayReshape[{(2 * Sqrt[2] * Cos[elev] * Sin[az / 2]) / Sqrt[1 + Cos[elev] Cos[az / 2]], (Sqrt[2] * Sin[elev]) / Sqrt[1 + Cos[elev] Cos[az / 2]]} /. {elev -> Range[0., 2 * Pi, 2 * Pi / 120][[i]], az → Pi}, {2}], {i, 1, 121}],
  Joined → {True, False}],
 ListPlot[Table[HammerCoordinates[StokesListTarget[[i]]], {i, 1, 120}], PlotStyle → {PointSize[0.01], RGBColor[1, 0, 0], Opacity[0.9]}],
 ListPlot[Table[HammerCoordinates[BestStates1[[i]]], {i, 1, 120}], PlotStyle → {PointSize[0.01], RGBColor[0, 1, 0], Opacity[0.9]}],
 ListPlot[Table[HammerCoordinates[stokeslistStates[[i]]], {i, 1, 6}], PlotStyle → {PointSize[0.013], RGBColor[0, 0, 0]}],
 Graphics[Style[Text["H", {0, 0.2}], Black, Italic, 30]],
 Graphics[Style[Text["V", {2 √2, 0.2}], Black, Italic, 30]],
 Graphics[Style[Text["D", {2/Sqrt[1 + 1/√2], 0.2}], Black, Italic, 30]],
 Graphics[Style[Text["A", {-2/Sqrt[1 + 1/√2], 0.2}], Black, Italic, 30]],
 Graphics[Style[Text["R", {0, √2 + 0.2}], Black, Italic, 30]],
 Graphics[Style[Text["L", {0, -√2 + 0.2}], Black, Italic, 30]],
 PlotRange → All, Axes → False, ImageSize → 600]
```

Out[ ]=