Demonstration

A concise demonstration of the ultimate calculation performed in **tutorial.nb**

Get the QuEST Mathematica package

In[1]:=
```
Import["https://quest.qtechtheory.org/QuEST.m"]
```

Download and connect to a local MacOS QuEST runtime environment

In[2]:=
```
env = CreateDownloadedQuESTEnv[];
```

Allocate a 9 qubit state vector and density matrix

In[3]:=
```
numQb = 9;
ψ = CreateQureg[numQb];
ρ = CreateDensityQureg[numQb];
```

Specify a 9 qubit circuit, which includes decoherence of strength parameterized by θ

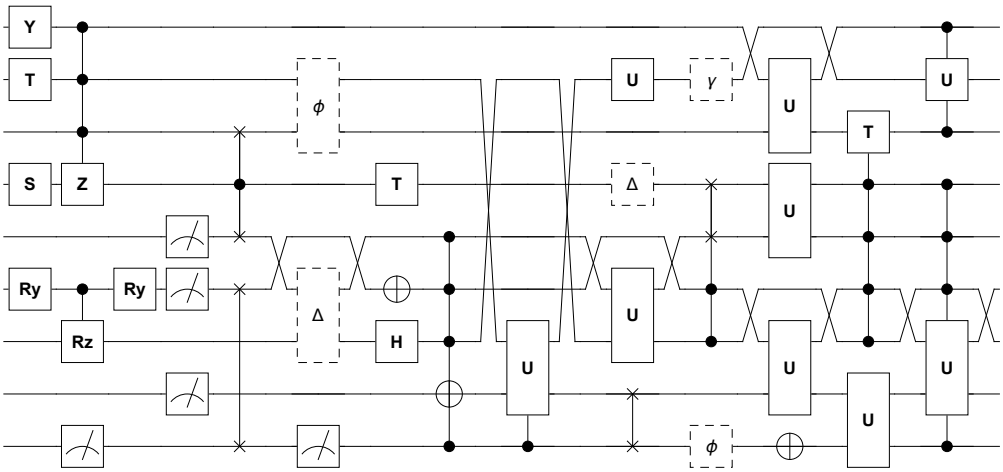
In[6]:=
```
m1 = {{0, I}, {Exp[.3 I], 0}};

m2 = {{1, 0, 0, 0}, {0, 0, 1, 0}, {0, 1, 0, 0}, {0, 0, 0, 1}};

u[θ_] := Circuit[
    S5 T7 Y8 Ry3[θ] C3[Rz2[θ]] C8,7,6[Z5] M0 Ry3[θ] M1,3,4 SWAP0,3 C5[SWAP4,6]
     Depol2,4[θ / 100] Deph7,6[θ / 400] M0 H2 X3 T5 C0,2,3,4[X1] C0[U1,7[m2]] U2,4[m2]
     U7[m1] SWAP0,1 Depol5[θ / 300] Deph0[θ / 200] Damp7[θ / 500] C2,3[SWAP4,5]
     U3,1[m2] U4,5[m2] U6,8[m2] X0 U0,1[m2] C2,3,4,5[T6] C0,2,4,5[U1,3[m2]] C6,8[U7[m1]]
  ];

DrawCircuit @ u[θ]
```

Out[9]=



Compute how smoothly varying θ affects the fidelity against the noise-free state.

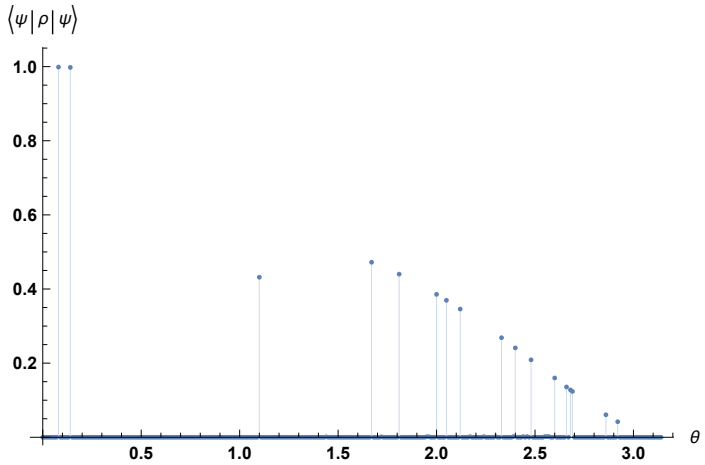
In[10]:=
```
ApplyCircuit[u[0], InitPlusState @ ψ];

params = Range[0, π, .01];
fids = Table[
       ApplyCircuit[u[θ], InitPlusState @ ρ];
       CalcFidelity[ρ, ψ],
       {θ, params}
  ];
```

Note the results here are *random* since our circuit contains projective measurement gates.

In[13]:=
```
ListPlot[
      Transpose[{params, fids}],
      AxesLabel → {"θ", "⟨ψ|ρ|ψ⟩"},
      Filling → Bottom
]
```

Out[13]=



Free QuEST memory and disconnect from QUEST environment

In[14]:=
```
DestroyAllQuregs[];
DestroyQuESTEnv[env];
```