Jaynes_Cummings model

February 28, 2021

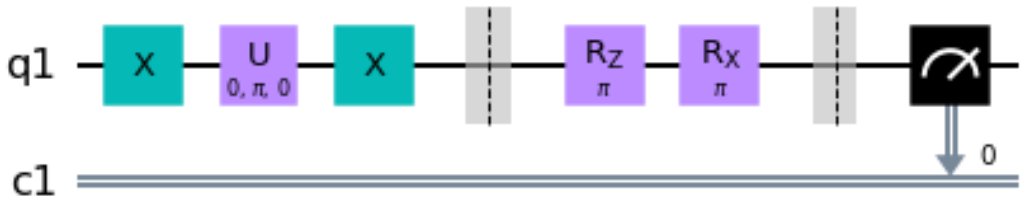
[1]:
```
from qiskit import*
from qiskit.tools.visualization import*
from numpy import*
from matplotlib.pyplot import*
```

[4]:
```
#circuit
qr=QuantumRegister(1)
cr=ClassicalRegister(1)
qc=QuantumCircuit(qr,cr)

qc.x(qr[0])
qc.u(0,pi,0,qr[0])#phase shift gate
qc.x(qr[0])
qc.barrier()
qc.rz(pi,qr[0])
qc.rx(pi,qr[0])
qc.barrier()
qc.measure(qr[0],cr[0])

qc.draw(output='mpl')
```

[4]:



[7]:
```
#data from QASM

qr=QuantumRegister(1)
cr=ClassicalRegister(1)
```

```
qc=QuantumCircuit(qr,cr)

backend=BasicAer.get_backend('qasm_simulator')
p=[]
q=[]
n=2
g=1
om=2*g*sqrt(n+1)#rabi frequency
d=0.2*pi#detuning frequency
t=arange(0,2,0.1)

for i in t:
    th=om*i*(n+1)
    #initialization
    qc.x(qr[0])
    qc.u(0,th,0,qr[0])#phase shift gate
    qc.x(qr[0])

    qc.rz(d*i,qr[0])
    qc.rx(om*i,qr[0])
    qc.measure(qr[0],cr[0])
    counts=execute(qc,backend,shots=1024).result().get_counts()
    if '0' in counts:
            p.append(counts['0'])
            q.append(1024-counts['0'])
    else:
        p.append(0)
        q.append(1024)
    qc.reset(qr)
```

[8]:
```
p=[p/1024 for p in p]
q=[q/1024 for q in q]
```

#data from IBM-BELEM IBMQ.load_account() provider=IBMQ.get_provider(hub='ibm-q') backend=provider.get_backend('ibmq_belem')

r=[] s=[] n=2 g=1 om=2*g*sqrt(n+1)#rabi frequency d=0.2*pi#detuning frequency t=arange(0,2,0.1)

for i in t: th=om*i*(n+1) #initialization qc.x(qr[0]) qc.u(0,th,0,qr[0])#phase shift gate qc.x(qr[0])

```
qc.rz(d*i,qr[0])
qc.rx(om*i,qr[0])
qc.measure(qr[0],cr[0])
job=execute(qc,backend,shots=1024)
results=job.result()
counts=results.get_counts()

if '0' in counts:
        r.append(counts['0'])
```

```
        s.append(1024-counts['0'])
else:
    r.append(0)
    s.append(1024)
qc.reset(qr)
```

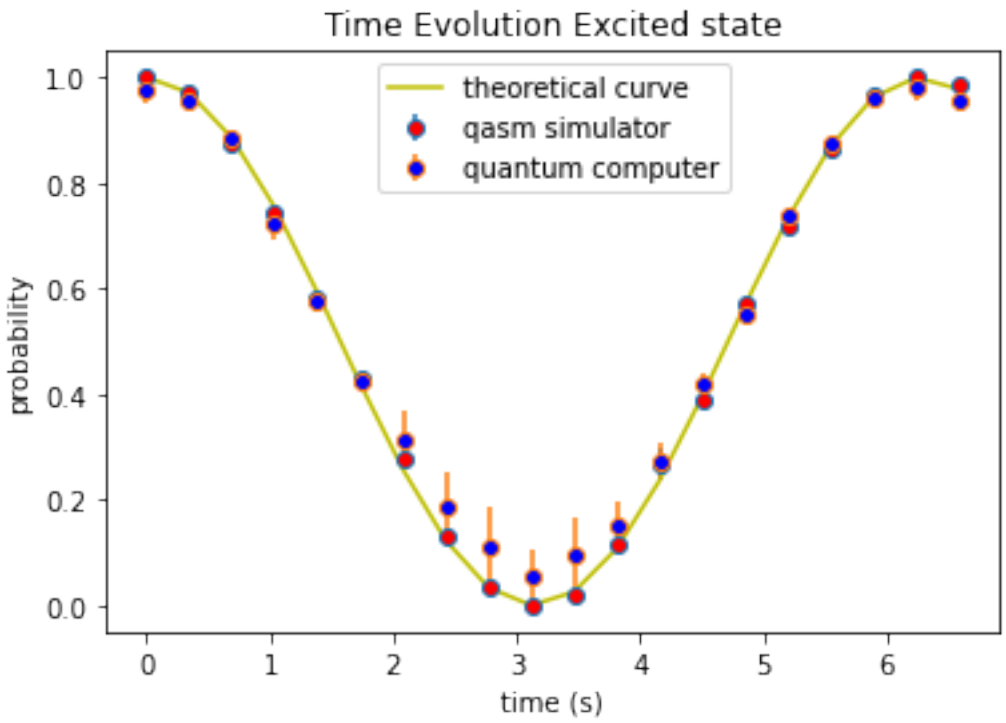
[5]:
```
r=[1000,981,907,741,588,433,321,191,114,55,98,157,280,429,565,757,896,983,1002,980]
s=[24,43,117,283,436,591,703,833,910,969,926,807,744,595,459,267,128,41,22,44]

r=[r/1024 for r in r]
s=[s/1024 for s in s]
```

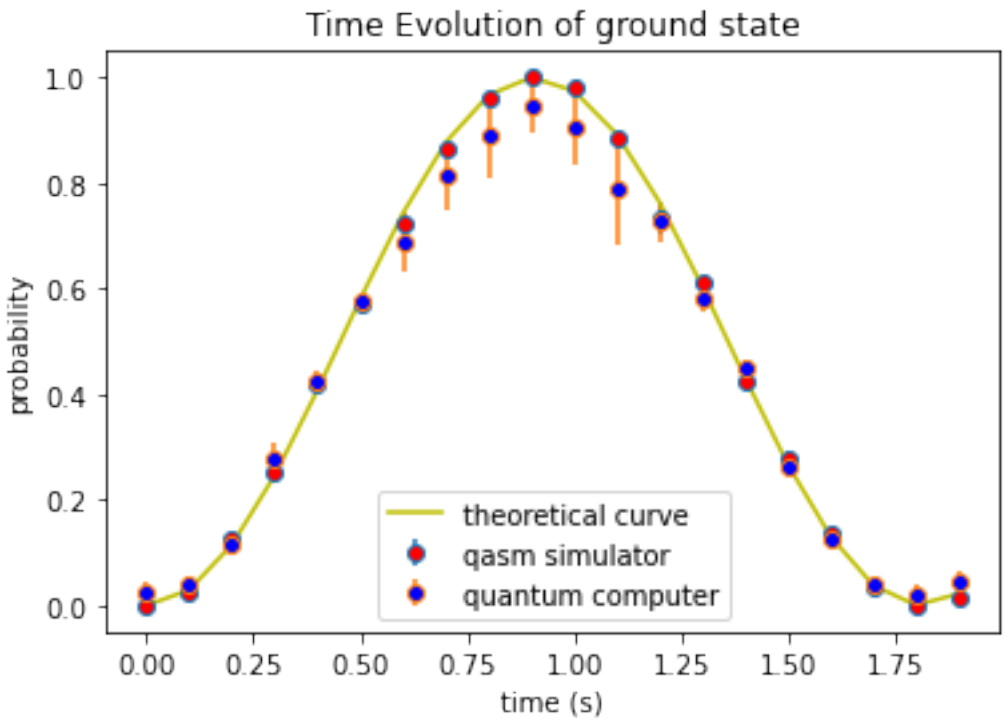
[9]:
```
#theoretical curve

e=1/2*(1+cos(om*t))
g=1/2*(1-cos(om*t))
```

[46]:
```
plot(om*t,e,'y',label='theoretical curve')
perr=[e-p for e,p in zip(e,p)]
errorbar(om*t,p,perr,fmt='o',mfc='red',label='qasm simulator')
rerr=[e-r for e,r in zip(e,r)]
errorbar(om*t,r,rerr,fmt='o', mfc='blue',label='quantum computer')
xlabel('time (s)')
ylabel('probability')
title('Time Evolution Excited state')
legend()
savefig('excitedstate.pdf')
```



[50]:
```
plot(t,g,'y',label='theoretical curve')
qerr=[g-q for g,q in zip(g,q)]
errorbar(t,q,qerr,fmt='o',mfc='red',label='qasm simulator')
serr=[g-s for g,s in zip(g,s)]
errorbar(t,s,serr,fmt='o',mfc='blue',label='quantum computer')
xlabel('time (s)')
ylabel('probability')
title('Time Evolution of ground state')
legend()
savefig('groundstate.pdf')
```


Time Evolution of ground state
probability
time (s)
theoretical curve
qasm simulator
quantum computer
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75

[ ]: