In [199…

```
%matplotlib inline
from IPython.display import Image
import cmath
import scipy
from numpy import pi
import numpy as np
from qutip import Qobj, average_gate_fidelity, rand_unitary
from qutip_qip.operations import *
from qutip_qip.circuit import QubitCircuit, Gate
from qutip_qip.decompose import decompose_one_qubit_gate
```

Gate and Array for U1

In [200…

```
H = hadamard_transform(N=1)
# U1 Array
U6 = controlled_gate(H, N=2, control=0, target=1, control_value=1)
print(U6.full)
```

```
<bound method Qobj.full of Quantum object: dims = [[2, 2], [2, 2]], shape = (4,
4), type = oper, isherm = True
Qobj data =
[[ 1.          0.          0.          0.        ]
 [ 0.          1.          0.          0.        ]
 [ 0.          0.          0.70710678  0.70710678]
 [ 0.          0.          0.70710678 -0.70710678]]>
```

In [201…

```
# U1 Circuit

# This is the default gate because it's decompsoition scheme is known from Lemma
def U_gate():
     U_prime = Qobj(U6)
     return Qobj(U_prime, dims=[[2, 2], [2, 2]])

qcU1 = QubitCircuit(2, reverse_states=False)
qcU1.user_gates = {"U_{gen}"     : U_gate}
paulix = Gate("X",targets=0, classical_controls=[0])
U1_gate_object = Gate("U_{gen}", targets=[0,1])
gate_list = [paulix,U1_gate_object,paulix]
qcU1.add_gates(gate_list)
qcU1.png
```

```
/home/purva/qutip-project/qutip-qip/src/qutip_qip/operations/gates.py:266: UserW
arning: Unknown gate U_{gen}
  warnings.warn("Unknown gate %s" % name)
```

Out[201…

In [202…

```
calculatedU1=qcU1.compute_unitary()
print(calculatedU1)
```

```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
```

```
Qobj data =
[[ 0.70710678  0.70710678  0.          0.        ]
 [ 0.70710678 -0.70710678  0.          0.        ]
 [ 0.          0.          1.          0.        ]
 [ 0.          0.          0.          1.        ]]
```

Circuit and array for U2

In [203…

```
U2 = controlled_gate(H, N=2, control=1, target=0, control_value=0)
print(U2)
```

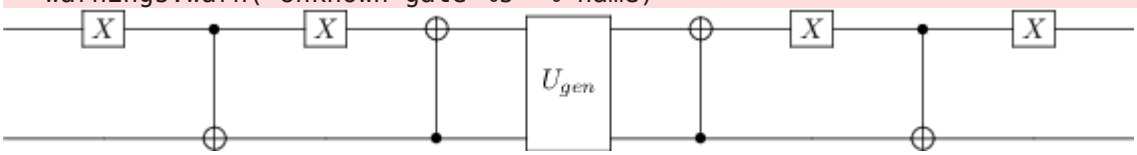
```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[ 0.70710678  0.          0.70710678  0.        ]
 [ 0.          1.          0.          0.        ]
 [ 0.70710678  0.         -0.70710678  0.        ]
 [ 0.          0.          0.          1.        ]]
```

In [204…

```
# U2 Circuit
qcU2 = QubitCircuit(2, reverse_states=False)
qcU2.user_gates = {"U_{gen}"    : U_gate}
U2_gate_object = Gate("U_{gen}", targets=[0,1])
paulix = Gate("X",targets=0, classical_controls=[0])
cnot1 = Gate("CNOT", controls=0, targets=1)
cnot2 = Gate("CNOT", controls=1, targets=0)
gate_list = [paulix,cnot1,paulix,cnot2,U2_gate_object,cnot2,paulix,cnot1,paulix]
qcU2.add_gates(gate_list)
qcU2.png
```

```
/home/purva/qutip-project/qutip-qip/src/qutip_qip/operations/gates.py:266: UserW
arning: Unknown gate U_{gen}
  warnings.warn("Unknown gate %s" % name)
```

Out[204…



In [205…

```
calculatedU2=qcU2.compute_unitary()
print(calculatedU2)
```

```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[-0.70710678  0.          0.70710678  0.        ]
 [ 0.          1.          0.          0.        ]
 [ 0.70710678  0.          0.70710678  0.        ]
 [ 0.          0.          0.          1.        ]]
```

U3 circuit and array

In [206…

```
U5 = controlled_gate(H, N=2, control=1, target=0, control_value=1)

# This is the default gate with different controls and targets
def U1_gate():
```

```
    U_prime = Qobj(U5)
    return Qobj(U_prime, dims=[[2, 2], [2, 2]])

qc3 = QubitCircuit(2, reverse_states=False)
qc3.user_gates = {"U_{gen_{ctrl}}"    : U1_gate}
new_gate = Gate("U_{gen_{ctrl}}", targets=[0,1])
paulix = Gate("X",targets=0, classical_controls=[0])
cnot1 = Gate("CNOT", controls=0, targets=1)
gate_list = [paulix,cnot1,paulix,new_gate,paulix,cnot1,paulix]
qc3.add_gates(gate_list)
qc3.png
```

```
/home/purva/qutip-project/qutip-qip/src/qutip_qip/operations/gates.py:266: UserW
arning: Unknown gate U_{gen_{ctrl}}
  warnings.warn("Unknown gate %s" % name)
```

Out[206…

In [207…

```
U3 = qc3.compute_unitary()
print(U3)
```

```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[ 0.70710678  0.          0.          0.70710678]
 [ 0.          1.          0.          0.        ]
 [ 0.          0.          1.          0.        ]
 [ 0.70710678  0.          0.         -0.70710678]]
```

U4 circuit and array

In [208…

```
U4 = controlled_gate(H, N=2, control=1, target=0, control_value=0).full()
cnot1 = cnot(N=None, control=0, target=1).full()
paulix = x_gate(N=None, target=0).full()
u_dagger = np.matmul(np.kron(paulix, np.identity(2)), np.matmul(cnot1,np.kron(pa
print(np.matmul(u_dagger,np.matmul(U4,u_dagger)))
```

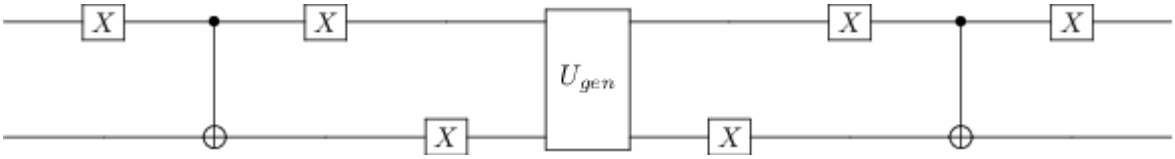
```
[[ 1.        +0.j  0.        +0.j  0.        +0.j  0.        +0.j]
 [ 0.        +0.j  0.70710678+0.j  0.70710678+0.j  0.        +0.j]
 [ 0.        +0.j  0.70710678+0.j -0.70710678+0.j  0.        +0.j]
 [ 0.        +0.j  0.        +0.j  0.        +0.j  1.        +0.j]]
```

In [209…

```
qcU4 = QubitCircuit(2, reverse_states=False)
qcU4.user_gates = {"U_{gen}"    : U_gate}
new_gate4 = Gate("U_{gen}", targets=[1,0])
paulix = Gate("X",targets=0, classical_controls=[0])
paulix1 = Gate("X",targets=1, classical_controls=[1])
cnot1 = Gate("CNOT", controls=0, targets=1)
gate_list = [paulix,cnot1,paulix,paulix1,new_gate4,paulix1,paulix,cnot1,paulix]
qcU4.add_gates(gate_list)
qcU4.png
```

```
/home/purva/qutip-project/qutip-qip/src/qutip_qip/operations/gates.py:266: UserW
arning: Unknown gate U_{gen}
  warnings.warn("Unknown gate %s" % name)
```

Out[209…



In [210…

```
U4 = qcU4.compute_unitary()
print(U4)
```

```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[ 1.          0.          0.          0.        ]
 [ 0.          0.70710678  0.70710678  0.        ]
 [ 0.          0.70710678 -0.70710678  0.        ]
 [ 0.          0.          0.          1.        ]]
```

U5 circuit and array

In [211…

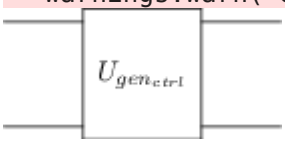
```
U5 = controlled_gate(H, N=2, control=1, target=0, control_value=1)

# This is the default gate with different controls and targets
def U1_gate():
     U_prime = Qobj(U5)
     return Qobj(U_prime, dims=[[2, 2], [2, 2]])

qc5 = QubitCircuit(2, reverse_states=False)
qc5.user_gates = {"U_{gen_{ctrl}}"    : U1_gate}
new_gate5 = Gate("U_{gen_{ctrl}}", targets=[0,1])
gate_list = [new_gate5]
qc5.add_gates(gate_list)
qc5.png
```

```
/home/purva/qutip-project/qutip-qip/src/qutip_qip/operations/gates.py:266: UserW
arning: Unknown gate U_{gen_{ctrl}}
  warnings.warn("Unknown gate %s" % name)
```

Out[211…



In [212…

```
U5 = qc5.compute_unitary()
print(U5)
```

```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[ 1.          0.          0.          0.        ]
 [ 0.          0.70710678  0.          0.70710678]
 [ 0.          0.          1.          0.        ]
 [ 0.          0.70710678  0.         -0.70710678]]
```

U6 array and circuit

In [213…

```
U6 = controlled_gate(H, N=2, control=0, target=1, control_value=1)
```

```
print(U6)
```

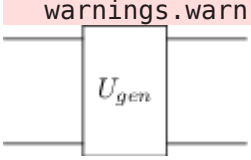
```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[ 1.          0.          0.          0.        ]
 [ 0.          1.          0.          0.        ]
 [ 0.          0.          0.70710678  0.70710678]
 [ 0.          0.          0.70710678 -0.70710678]]
```

In [214…

```
qcU6 = QubitCircuit(2, reverse_states=False)
qcU6.user_gates = {"U_{gen}"    : U_gate}
paulix = Gate("X",targets=0, classical_controls=[0])
U6_gate_object = Gate("U_{gen}", targets=[0,1])
gate_list = [U6_gate_object]
qcU6.add_gates(gate_list)
qcU6.png
```

```
/home/purva/qutip-project/qutip-qip/src/qutip_qip/operations/gates.py:266: UserW
arning: Unknown gate U_{gen}
  warnings.warn("Unknown gate %s" % name)
```

Out[214…



In [215…

```
U6 = qcU6.compute_unitary()
print(U6)
```

```
Quantum object: dims = [[2, 2], [2, 2]], shape = (4, 4), type = oper, isherm = T
rue
Qobj data =
[[ 1.          0.          0.          0.        ]
 [ 0.          1.          0.          0.        ]
 [ 0.          0.          0.70710678  0.70710678]
 [ 0.          0.          0.70710678 -0.70710678]]
```

In [ ]: