Python quantum programming languages

John Scott, Oliver Thomas

Quantum Engineering CDT
University of Bristol

September 14, 2018

Overview

- We'll focus on Python based quantum programming libraries
- We tried to program the common programs (e.g. Grover's algorithm, Shor's algorithm, etc.)
- We tried compiling a simple program for different hardware platforms (i.e. with gate restrictions, etc.)
- We've written a programming guide – it's under an internal review

```
# Do quantum stuff now we have our bit string
qvm = QVMConnection()
qprog = Program()

# do X on q1, q3, q7
# remember HZH is X
qprog.inst(H(1), Z(1), H(1))
qprog.inst(X(3))
qprog.inst(X(7))
# do measurement over all 8 qubits
for i in range(0, 8):
    qprog.measure(i, i)
```

Short comparison

What is there

- Focussed on quantum circuits
- Apply gates to specific qubits
- Classical control in the same source code
- Python syntax is beginner friendly
- Simulators are available
- Hardware compilers are available

What is lacking

- Lack of support for custom unitaries
- Compilers are not highly developed
- Some languages target specific hardware
- Some simulators are cloud based and require accounts
- No real quantum programming contructs (e.g. quantum if etc.)

What do we mean by nonlinear optics?

- Roughly processes that conserve energy but do not conserve photon number.

Gaussian Optics

- Using th
- We

Types of

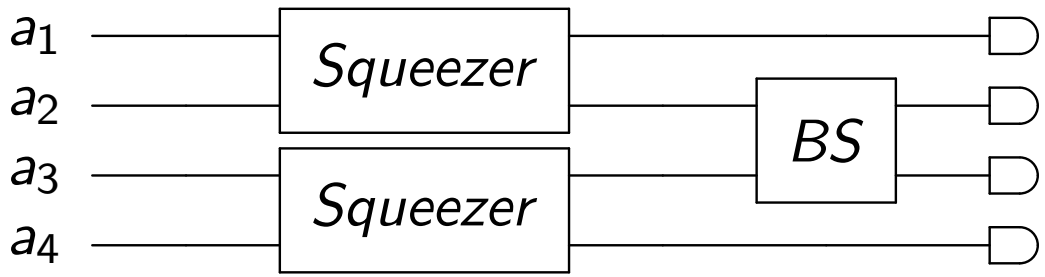

Figure: Two source HOM dip

[0]These are two-mode squeezers

Schmidt decomposition

- with $\psi_k(\omega_1)$ is the k-th row and ω_1-th column of $\mathbf{U}_{(\omega_1,k)}$,
- with $\phi_k(\omega_2)$ is the ω_2-th row and k-th column of $\mathbf{V}^{\dagger}_{(k,\omega_2)}$

Summary

k

References