

Identification sur un système quantique bruité

Théorie et démonstration expérimentale
sur un processeur quantique

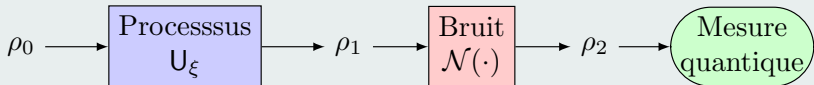
Nicolas DELANOUE & François CHAPEAU-BLONDEAU

LARIS - Université d'Angers - France

6èmes journées des Démonstrateurs en Automatique - 21-22 juin 2022



Un système quantique bruité



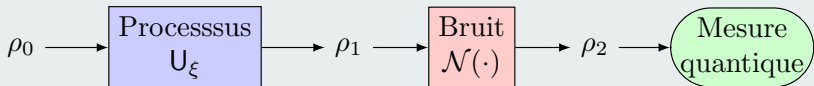
- **Signal quantique** en entrée : opérateur densité $\rho_0 = \sum_j p_j |\psi_j\rangle \langle \psi_j|$, où chaque $|\psi_j\rangle$ est un vecteur d'état de l'espace de Hilbert \mathcal{H} .
- **Système quantique** U_ξ produit le signal $\rho_1 = U_\xi \rho_0 U_\xi^\dagger$.
- **Bruit quantique** $\mathcal{N}(\cdot)$ produit le signal bruité $\rho_2 = \mathcal{N}(\rho_1)$,
- qui est mesuré pour estimer le paramètre inconnu ξ .

Problème

Comment

- 1 efficacement exciter en entrée ρ_0 et mesurer en sortie ?
- 2 mettre en oeuvre pratiquement ?
- 3 estimer le paramètre ξ ?

Un système quantique bruité



Démonstration sur un processeur quantique

The screenshot shows the Quantum Lab interface. At the top, there are two boxes: a purple one labeled U_ξ and a pink one labeled $\mathcal{N}(\cdot)$. Below them is a toolbar with various quantum gates like H, X, Z, T, S, P, RZ, and a "Add" button. The main area displays a quantum circuit with four qubits (q[0] to q[3]). Qubit 0 has an H gate, followed by an RZ gate (highlighted by a green box and a purple arrow from U_ξ), then a CNOT gate with qubit 1 as control and qubit 0 as target. Qubit 1 has an X gate, followed by a CNOT gate with qubit 0 as control and qubit 1 as target, then an H gate, and finally a measurement gate (highlighted by a green arrow from $\mathcal{N}(\cdot)$). Qubit 2 has an H gate, followed by a CNOT gate with qubit 3 as control and qubit 2 as target. Qubit 3 has an RY gate, followed by a CNOT gate with qubit 2 as control and qubit 3 as target. A green oval labeled "Mesure quantique" is positioned below the measurement gate. To the right, there is a code editor with the following code:

```
1 OPENQASM 2.0;  
2 include "qelib1.inc";  
3  
4 qreg q[4];  
5 creg c[4];  
6  
7 h q[0];  
8 h q[2];  
9 ry(pi/2) q[3];  
10 rz(pi/4) q[0];  
11 cx q[2],q[1];  
12 cswap q[3],q[0],q[1];  
13 h q[0];  
14 measure q[0] -> c[0];
```

To the right of the code editor is a Bloch sphere visualization showing several vectors representing different states, with a "Phase" indicator at the bottom right.

Accessible en ligne via <https://quantum-computing.ibm.com>