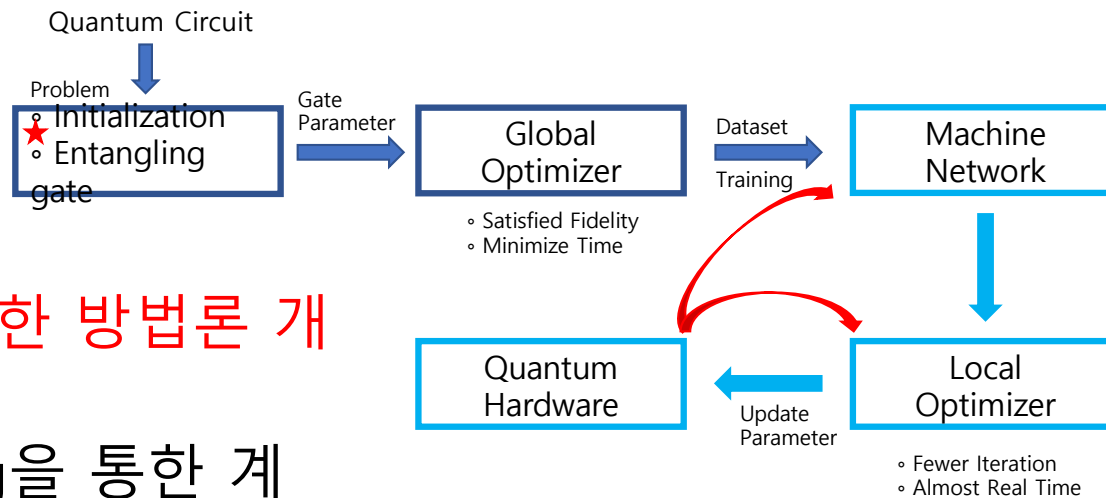


Time-Optimal Spin Control

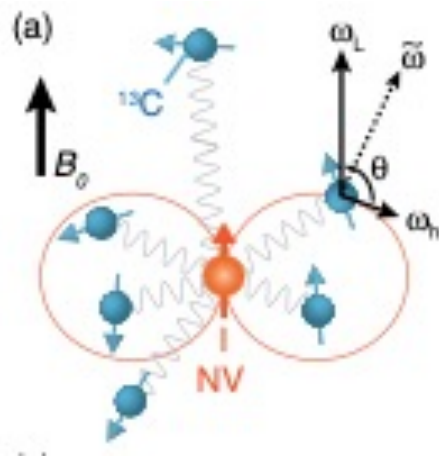
양자정보연구단 인턴 하규원

목표

- Spin control에 대한 방법론 개발
- Machine Learning을 통한 계산 속도 개선
- 실제 실험 셋업에 적용

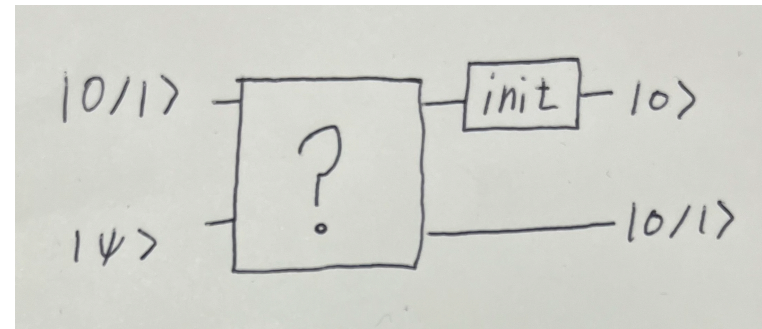
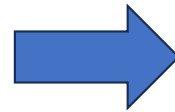
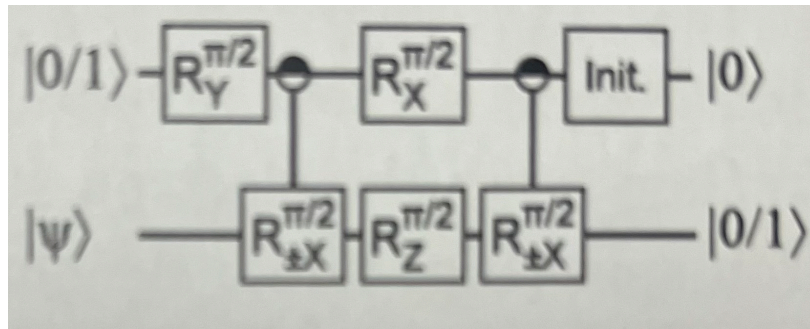


목표



- NV spin과 Hyperfine interaction을 하는 Carbon nuclear spin을 최단시간으로 초기화

목표



- Resonant한 tau값을 통해 conditional x gate를 구현
- Resonant하지 않은 tau값을 통해 unconditional z gate를 구현

- 적절한 tau값을 찾아 한번의 pulse 조합으로 초기화를 진행

Hamiltonian

$$\hat{H} = A_{||} \hat{S}_z \hat{I}_z + A_{\perp} \hat{S}_z \hat{I}_x + \omega_L \hat{I}_z$$

- A : Hyperfine interaction
- ω_L : 주어진 시스템에서, Carbon의 Larmor frequency

아이디어

- NV spin control 당시 사용했던 A* algorithm을 사용하기에는, 예상소요시간을 알 수 있는 방법을 아직 찾아내지 못함.
- Hyperfine Interaction 값이 주어졌을 때, optimal하진 않더라도, 초기화 게이트를 작은 연산시간 내에서 찾는 것 까지는 성공함.
- 위에서 찾은 초기화 게이트를 Random Search를 통해 적은 수의 펄스 조합을 찾는 것을 시도해보려고 함.

초기화 게이트 찾는 법

```
pulse_list = [U180xm, U180xmm, U180ym, U180ymm, U090xm, U090xmm, U090ym, U090ymm]

def make_unitary_2(tau, choice) :
    A1 = 2*pi*0.1*Constant_A1 #[MHz] # A_|| hyperfine term
    A_perp = 2*pi*0.1*Constant_A_perp #[MHz] # A_perp hyperfine term
    ham = A_perp*np.kron(sz, Ix) + A1*np.kron(sz, Iz) + B*gammaN*np.kron(I, Iz)
    eigvals = np.linalg.eigh(ham)[0] # diagonalizing the Hamiltonian
    eigvecs = np.linalg.eigh(ham)[1]
    E = np.diag(eigvals) # exponent of eigenvalues

    U = eigvecs @ linalg.expm(-1j*E*tau/2) @ eigvecs.conj().T

    U_e = U @ np.kron(pulse_list[choice], I) @ U
    return U_e
```

```
while point[2] < 0.99 :

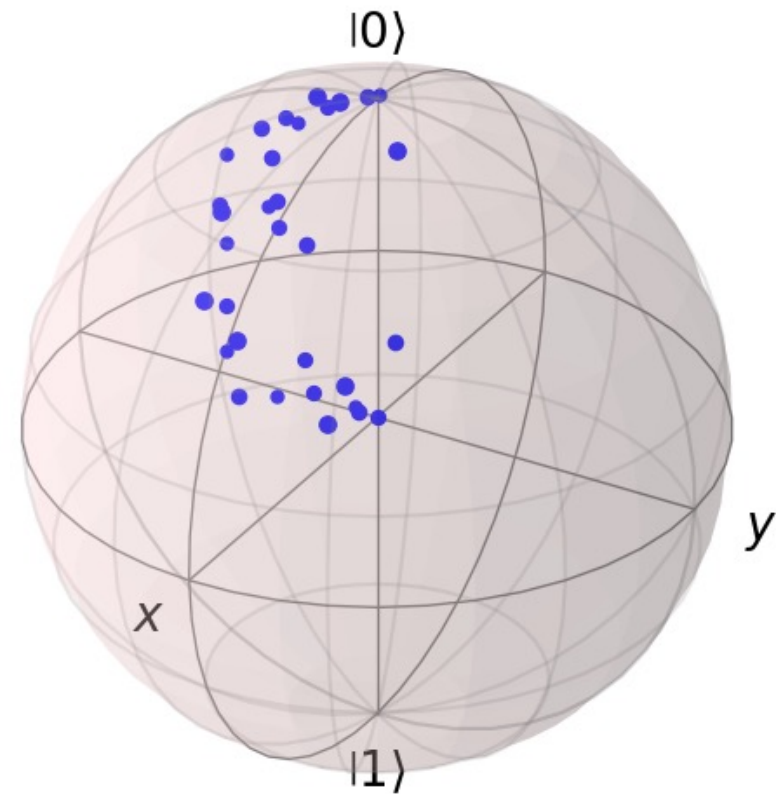
    temp_z = []
    current_z = point[2]

    for u in range(8) :
        U = U_list[u]
        rho_temp = U @ rho_0 @ U.conj().T
        delta = (np.trace(Iz@partial_trace(rho_temp, 1)).real - current_z)
        temp_z.append(delta)

    max_delta_value = max(temp_z)
    max_delta_indices = [j for j, v in enumerate(temp_z) if v==max_delta_value]
    choice = random.choice(max_delta_indices)
    U = U_list[choice]
    rho_0 = U @ rho_0 @ U.conj().T
    point = [np.trace(Ix@partial_trace(rho_0, 1)).real,
             np.trace(Iy@partial_trace(rho_0, 1)).real,
             np.trace(Iz@partial_trace(rho_0, 1)).real]
    combi.append(choice)
```

결과

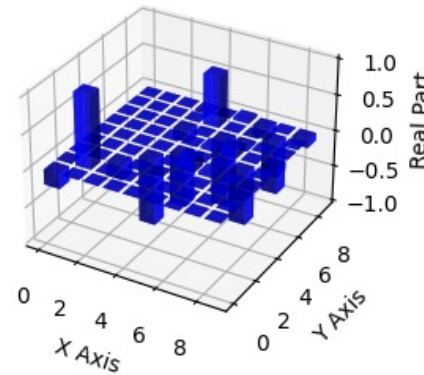
- $A_I = 2\pi \cdot 0.1$ MHz
- $A_p = 2\pi \cdot 0.1$ MHz
- Number of step = 32
- $\tau = 3.788 \mu\text{s}$
- Total time = $121.2 \mu\text{s}$
- Fidelity = 0.9971



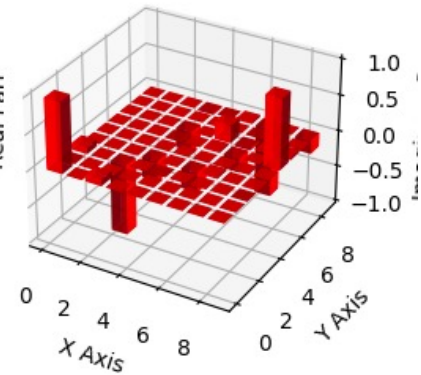
결과

- 찾아낸 Unitary gate의 최종형태
- 임의의 Hyperfine interaction 값에 대하여 각각의 형태를 찾아낼 수 있음.

3D Bar Plot - Real Part



3D Bar Plot - Imaginary Part



감사합니다