

CQS - Aula 8 - Fourier Quântica e aplicações

☰ Conteúdo e Tag	Aula
📄 Curso	Engenharia Mecatrônica
📄 Disciplina	Computação Quântica e Supercomputadores
📄 Semestre	1º Semestre

1. Introdução à Computação Quântica

Conceitos Fundamentais

- **Qubit:** Unidade de informação quântica que pode representar 0, 1 ou uma superposição de ambos. É a base de qualquer sistema quântico.
- **Superposição:** Um qubit pode estar em uma combinação linear de 0 e 1, possibilitando computação simultânea em diferentes estados.
- **Portas Lógicas Quânticas:** Operações que manipulam o estado dos qubits. As mais comuns são as portas X (NOT), H (Hadamard), e P (Porta de Fase).

Estados de Fase

- A fase é uma propriedade fundamental que determina a posição relativa dos estados de um qubit em superposição. Será essencial para o cálculo do período através da Transformada de Fourier.

2. Fundamentos para o Cálculo de Períodos em Circuitos Quânticos

A Global Solution requer que utilizemos a Computação Quântica para calcular o período de ondas marítimas. Neste caso, queremos modelar o sistema quântico para que ele "leia" esses períodos diretamente nas fases dos qubits.

- **Período e Frequência:** O período de uma onda é o tempo necessário para completar um ciclo, diretamente ligado à frequência de transferência de

energia.

- **Importância das Fases:** As fases de qubits em superposição ajudam a modelar a relação entre o tempo e o comportamento cíclico das ondas.

3. Transformada Quântica de Fourier Inversa (IQFT)

A Transformada de Fourier é uma ferramenta matemática usada para converter dados no domínio do tempo para o domínio da frequência e vice-versa. Em computação quântica, a IQFT nos permite determinar o período com base nas fases de qubits.

Implementação no Qiskit

```
from qiskit import QuantumCircuit
from qiskit.circuit.library import QFT

num_qubits = 4 # Defina o número de qubits necessários
qc = QuantumCircuit(num_qubits)
qc.append(QFT(num_qubits, inverse=True), qc.qubits)
qc.measure_all()
```

Conceito Prático

A IQFT aplicada em um circuito permite extrair o período a partir da relação entre as fases dos qubits, baseando-se na fórmula $\theta_i = \frac{2\pi \cdot i}{T}$ onde T é o período.

4. Montagem do Circuito Quântico para Captura de Períodos

Nesta seção, montaremos um circuito quântico para receber medições de período das ondas.

Passo a Passo:

1. **Obtenha medições das ondas:** Suponha medições de períodos entre cristas de ondas.
2. **Configuração das Portas P:** Use as portas P para definir a fase de cada qubit de acordo com o período medido.

3. **Aplicação da IQFT:** Após configurar as fases, aplique a IQFT para calcular o período total.

Código Exemplo

```
from qiskit import QuantumCircuit
import numpy as np

num_qubits = 4
qc = QuantumCircuit(num_qubits)

# Exemplo de fases com T = 3 e num_qubits = 4
for i in range(num_qubits):
    theta = 2 * np.pi * i / 3 # Calculando a fase
    qc.p(theta, i) # Aplicando a porta P com a fase calculada

# Aplicando a IQFT
qc.append(QFT(num_qubits, inverse=True), qc.qubits)
qc.measure_all()
```

5. Simulação e Introdução de Erros

Para validar a precisão do sistema, introduzimos um erro aleatório de 10% nas medições e comparamos os resultados.

Implementação

1. **Introduza Erros Aleatórios:** Adicione um desvio de $\pm 10\%$ às medições.
2. **Reaplique a IQFT:** Repita o processo de medição com erro e compare os resultados obtidos.

Código Exemplo para Erro

```
import random

def medicao_com_erro(medicao, erro_perc=10):
    erro = random.uniform(-erro_perc, erro_perc) / 100
    return medicao * (1 + erro)
```

```
medicoes = [3.0, 3.2, 3.1] # Exemplo de medições
medicoes_com_erro = [medicao_com_erro(m) for m in medicoes]
```

6. Automação da Simulação e Análise de Resultados

O objetivo é automatizar a construção e simulação do circuito para ajustar o número de bits até que a precisão desejada seja atingida.

Algoritmo para Iteração

1. **Inicialize com um número de bits suficiente para representar o período com erro de $\pm 10\%$.**
2. **Calcule a média e obtenha o período.**
3. **Ajuste até obter precisão:** Se a diferença entre média e período obtido for maior que 10%, adicione uma medição e continue o processo.

Código Exemplo

```
from statistics import mean, stdev

def simulacao(medicoes):
    while True:
        media = mean(medicoes)
        periodo_iqft = calcular_periodo_iqft(medicoes)
        if abs(media - periodo_iqft) / periodo_iqft < 0.1:
            break
        medicoes.append(nova_medicao()) # Suponha uma função de nova medição
    return media, periodo_iqft, len(medicoes), stdev(medicoes)

media_final, periodo_final, num_bits, desvio = simulacao(medicoes_com_erro)
```

7. Exercícios Práticos

1. **Exercício 1:** Crie um circuito quântico para representar uma sequência de fases e aplique a IQFT.
2. **Exercício 2:** Implemente o algoritmo de simulação com adição de erro e compare os resultados.
3. **Exercício 3:** Escreva um pequeno relatório discutindo o impacto dos erros nos resultados da IQFT.

VISÃO GERAL GS E DICAS

A proposta é usar computação quântica para calcular o **período das ondas do mar**, o que pode ajudar na geração de energia maremotriz. Essa energia é extraída do movimento das ondas, e o período influencia na quantidade e na qualidade da energia capturada por turbinas e flutuadores.

Objetivo

Vocês vão **criar e simular um circuito quântico** que consegue:

1. Receber medições do período das ondas (o tempo entre cristas sucessivas).
2. Aplicar uma transformação quântica para calcular o período baseado nessas medições.
3. Adicionar um erro controlado de 10% para simular imperfeições e ver como isso afeta o cálculo.
4. Automatizar a simulação para corrigir o erro ao longo de várias medições e fazer uma análise final.

Agora, vamos detalhar cada passo!

Passo 1: Construir o Circuito Quântico com Medições das Ondas

- **Coleta de Dados:** A primeira tarefa é entender como são feitas medições de período no mar. Vocês vão simular essas medições em segundos.
 - Exemplo: imaginem que registramos intervalos de tempo entre cristas de onda consecutivas. Esses valores podem ser algo como 3.0, 3.2, 3.1 segundos.
- **Representação no Circuito:** Vocês vão aplicar esses valores como **fases** nos qubits do circuito. Cada fase representa uma dessas medições.

Passo 2: Aplicar a Transformada Quântica de Fourier Inversa (IQFT)

- A **IQFT** será usada para **extrair o período** a partir das fases configuradas em cada qubit.
 - **Como funciona:** A IQFT "lê" a relação entre as fases dos qubits e, com isso, obtém o valor do período que melhor representa essas medições.
 - **Implementação:** No código, isso envolve configurar o circuito para usar a IQFT no Qiskit, com base nas medições anteriores aplicadas como fases.

Passo 3: Introduzir Erros Aleatórios de 10%

- Para simular a realidade, onde as medições têm uma margem de erro, vocês vão aplicar um **erro aleatório de 10%** (tanto positivo quanto negativo) nas medições.
 - Isso significa que cada medição deve ser ajustada para cima ou para baixo em até 10% de seu valor original.
 - Esse erro vai impactar o valor do período obtido pela IQFT, e o objetivo é ver como o erro afeta o cálculo.

Passo 4: Repetir a Simulação e Comparar Resultados

- Depois de aplicar o erro, simulem o circuito novamente para comparar o valor do período obtido com a média das medições iniciais.
- **Discussão dos Resultados:** Observem se o valor obtido pela IQFT está próximo do valor médio das medições e discutam qualquer diferença. Essa análise ajuda a entender o impacto do erro nas medições.

Passo 5: Automatizar o Processo e Melhorar a Precisão

- Agora vocês devem automatizar o processo para que o circuito faça o seguinte:
 1. **Iniciar com um número mínimo de bits:** Começar com uma quantidade de medições suficiente para representar o período com precisão (exemplo: para representar 25 segundos, são necessários 5 bits).
 2. **Calcular a média e obter o período:** Com essas medições, calcular a média e aplicar a IQFT para obter o período.
 3. **Avaliar a precisão:** Se a diferença entre o período obtido pela IQFT e a média das medições for maior que 10%, o sistema deve adicionar uma nova medição.
 4. **Repetir até atingir precisão:** O processo continua até que a diferença seja menor que 10%, e então os resultados finais são salvos.
- **Resultados a serem apresentados:** Quando a simulação terminar, vocês devem fornecer:
 - O período obtido pela média das medições.
 - O período obtido pela IQFT.
 - O número total de bits (ou medições) utilizados.
 - O desvio padrão das medições.

Passo 6: Análise Final e Discussão

- A última etapa é fazer uma análise final dos resultados. Comparar o valor médio com o valor obtido pela IQFT e discutir o impacto do erro aleatório, observando como o circuito se ajustou para corrigir o erro.