การสร้างวงจรควอนตัมเพื่อจำลองระบบการสั่นคู่ควบโดยใช้เทคนิคบล็อกเอ็นโค้ดดิง

นายณัฏฐ์ เหลืองสิรพรชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2567

BUILDING QUANTUM CIRCUIT VIA BLOCK ENCODING TECHNIQUE
FOR SIMULATING COUPLED OSCILLATION

Mr. Natt Luangsirapornchai

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2024

หัวข้อวิทยานิพนธ์	การสร้างวงจรควอนตัมเพื่อจำลองระบบการสั่นคู่ควบโดยใช้เทคนิคบล็อกเอ็นโค้ดดิง
โดย	นายณัฏฐ์ เหลืองสิริพรชัย
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร. ประภาส จงสถิตย์วัฒนา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

คณบดีคณะวิศวกรรมศาสตร์

....

(รองศาสตราจารย์ ดร.วิทยา วัณณสุโภประสิทธิ์)

คณะกรรมการสอบวิทยานิพนธ์

.........เศรษฐา......... ประธาน

(รองศาสตราจารย์ ดร. เศรษฐา ปานงาม)

ประภาส จงสถิตวัฒนา .... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ศาสตราจารย์ ดร. ประภาส จงสถิตย์วัฒนา)

.... กรรมการภายนอก

(ดร. นินนาท แดงเนียม)

ณัฏฐ์ เหลืองสิรพรชัย: การสร้างวงจรควอนตัมเพื่อจำลองระบบการสั่นคู่ควบโดยใช้เทคนิคบล็อกเอ็นโค้ดดิง. (BUILDING QUANTUM CIRCUIT VIA BLOCK ENCODING TECHNIQUE FOR SIMULATING COUPLED OSCILLATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ศ. ดร. ประภาส จงสถิตย์วัฒนา, 59 หน้า.

วิทยานิพนธ์เล่มนี้นำเสนอวงจรควอนตัมสำหรับการจำลองระบบการสั่นคู่ควบโดยอาศัยเทคนิคบล็อกเอ็นโค้ดดิงซึ่งเป็นวิธีการฝังเมทริกซ์และการดำเนินการพื้นฐานลงไปในตัวดำเนินการควอนตัมควบคู่กับการใช้วงจรแปลงค่าเอกฐานเชิงควอนตัม ภายในงานนี้อาศัยควอนตัมเกตพื้นฐานในการสร้างวงจร โดยมีการวัดความถูกต้องของวงจรเทียบกับผลลัพธ์การคำนวณเมทริกซ์บนคอมพิวเตอร์คลาสสิก และมีวัดขนาดของวงจรบล็อกเอ็นโค้ดดิงเทียบกับเฟรมเวิร์คที่มีอยู่ในปัจจุบัน วงจรที่ได้สามารถนำไปใช้ต่อยอดหรือเปรียบเทียบการจำลองระบบการสั่นคู่ควบด้วยเทคนิคอื่นหรือการจำลองระบบอื่นด้วยเทคนิคบล็อกเอ็นโค้ดดิงได้

ภาควิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต	ณัฏฐ์ เหลืองสิรพรชัย
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก	ประภาส
ปีการศึกษา	2567		

6372033021: MAJOR COMPUTER ENGINEERING

KEYWORDS: QUANTUM COMPUTING / QUANTUM CIRCUIT / BLOCK ENCODING / COUPLED OSCILLATION

NATT LUANGSIRAPORNCHAI : BUILDING QUANTUM CIRCUIT VIA BLOCK ENCODING TECHNIQUE FOR SIMULATING COUPLED OSCILLATION. ADVISOR : PROF. PRABHAS CHONGSTITVATANA, Ph.D., 59 pp.

This thesis presents a quantum circuit for simulating coupled oscillators using the block encoding technique, which is a method to embed matrices and basic operations on the quantum operator, together with the quantum singular value transformation circuit. In this work, basic quantum gates are used to construct the circuit. The accuracy is measured by comparing it with matrix calculations on a classical computer. And the block-encoded circuits' size are compared to the results of existing frameworks. The results of this thesis can be further developed and compared with other coupled oscillation simulations or other block encoding technique simulations.

Department:	Computer Engineering	Student's Signature	ณัฏฐ์ เหลืองสิราพรชัย ....
Field of Study:	Computer Engineering	Advisor's Signature	ประภาส ....
Academic Year:	2024		

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สามารถสำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์และการสนับสนุนเป็นอย่างดีจาก ศาสตราจารย์ ดร. ประภาส จงสถิตย์วัฒนา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้กรุณาเสียสละเวลาให้คำปรึกษาและให้ความรู้ ทั้งในด้านการทำวิจัย การเรียนและการใช้ชีวิต รวมถึงแนะนำแนวทางการแก้ไขปัญหาที่เกิดขึ้น และให้ความช่วยเหลือในด้านต่าง ๆ ผู้เขียนขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบคุณดร.ธิปรัชต์ โชติบุตร ที่คอยให้คำแนะนำและให้ความรู้ทั้งในด้านเนื้อหาฟิสิกส์และด้านการทำงานวิจัย และได้เปิดโอกาสให้ผู้เขียนได้เข้าร่วมกิจกรรมของแล็บ Chula Intelligent and Complex Systems (CHICS) เพื่อแลกเปลี่ยนความรู้ความคิดเห็นกับนิสิตและนักวิจัยของภาควิชาฟิสิกส์ รวมถึงช่วยเสนอแนะและแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้น และขอขอบคุณดร.กมลลักษณ์ สุขเสน ที่ให้การสนับสนุนและให้ความช่วยเหลือ รวมถึงพูดคุยแลกเปลี่ยนความรู้ความคิดเห็นด้านงานวิจัยมาตลอด

ขอขอบคุณพี่อั้ม อภิมุข สรแสง ที่คอยสอน แนะแนวทาง และแลกเปลี่ยนความรู้ทั้งในเนื้อหาด้านคณิตศาสตร์และงานวิจัยด้านควอนตัม รวมทั้งคอยสนับสนุนการทำงานวิจัยนี้มาตลอด ทำให้ผู้เขียนมีความเข้าใจทั้งในด้านฟิสิกส์และคณิตศาสตร์อย่างถ่องแท้มากขึ้น ขอขอบคุณน้องอู๋ พีระณัฐ แสงละออ ที่ทำงานวิจัยนี้ร่วมกัน และคอยแนะนำ ช่วยเหลือและแลกเปลี่ยนความรู้จนได้ไอเดียใหม่ ๆ ในงานวิจัยนี้ ขอขอบคุณพี่ต๊ะ ปฐวี ปราการกมานันท์ ที่ได้ช่วยเหลือและให้คำแนะนำทั้งในด้านคณิตศาสตร์และการทำงานวิจัย ขอขอบคุณพี่โอมัส นภันต์ เบญจสัตตบุษย์ ที่ให้คำปรึกษาในเนื้อหาทางด้านควอนตัมและแนะนำงานวิจัยที่เกี่ยวข้อง ขอขอบคุณดร.ทศพร อังสาชน ที่ให้คำแนะนำในเนื้อหาด้านกลศาสตร์ที่ใช้ในงานวิจัยนี้ และขอขอบคุณเพื่อนนักวิจัยในแล็บ Intelligent System Lab (ISL) ที่ให้การสนับสนุน

และส่วนสุดท้ายขอขอบพระคุณ บิดา มารดา ของข้าพเจ้าที่คอยให้การสนับสนุนและความช่วยเหลือในทุก ๆ ด้านตลอดระยะเวลาการศึกษา รวมถึงเพื่อนสนิทมิตรสหาย ที่คอยให้กำลังใจและให้แรงบันดาลใจ จนทำให้งานวิจัยและวิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

สารบัญ

หน้า

บทคัดย่อภาษาไทย . . . . **ง**

บทคัดย่อภาษาอังกฤษ . . . . **จ**

กิตติกรรมประกาศ . . . . **ฉ**

สารบัญ . . . . **ช**

สารบัญภาพ . . . . **ฌ**

1 บทนำ **1**

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของงานวิจัย 2

1.3 ขอบเขตการดำเนินงาน 2

1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย 3

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง **4**

2.1 หลักพื้นฐานของการคำนวณควอนตัม 4

2.2 การจำลองควอนตัม 8

2.3 การสั่นคู่ควบ 16

3 วิธีดำเนินการวิจัย **21**

3.1 แนวคิดและวิธีการดำเนินงาน 21

3.2 ขั้นตอนการดำเนินงาน 22

4 การออกแบบวงจร **23**

4.1 การสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ B 23

4.2 วงจรบล็อกเอ็นโค้ดดิงของ H 32

4.3 การสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ e^{-iHt} 33

4.4 วงจรทั้งหมดสำหรับการจำลอง 34

หน้า

5 ผลการวิจัย **35**

5.1 เปรียบเทียบความลึกวงจรเทียบกับวงจร FABLE 35

5.2 ผลการจำลองการสั่นคู่ควบ 37

6 สรุปงานวิจัย **43**

รายการอ้างอิง **46**

ประวัติผู้เขียนวิทยานิพนธ์ **49**

สารบัญภาพ

รูปที่ **หน้า**

2.1 ตัวอย่างวงจรควอนตัมสำหรับการบวกเลข 1 บิต (Single bit half-adder) 6
2.2 รายชื่อควอนตัมเกต สัญลักษณ์และเมทริกซ์ (Nielsen and Chuang, 2019) . . . 7
2.3 วงจรบล็อกเอ็นโค้ดดิงของเมทริกซ์ทแยงมุม 12
2.4 วงจรการรวมเชิงเส้นของยูนิแทรีหลายตัว 13
2.5 วงจรการรวมเชิงเส้นของ 2 ยูนิแทรี 13
2.6 วงจรบล็อกเอ็นโค้ดดิงของการคูณเมทริกซ์ 14
2.7 บล็อกเอ็นโค้ดดิงของการคูณเทนเซอร์ (Camps and Van Beeumen, 2020) . . . 14
2.8 วงจร QET(QSVT) (Lin, 2022) 15
2.9 ระบบมวลติดสปริงแบบเส้นตรง 16
2.10 ระบบมวลติดสปริงแบบวงกลม 17
4.1 วงจร (1,1,0)-block-encoding ของเมทริกซ์ $M^{-\frac{1}{2}}$ และ $W^{\frac{1}{2}}$ 24
4.2 วงจร L-shift (Camps et al., 2024) 26
4.3 วงจร (1,1,0)-block-encoding ของ $\tilde{I}$ และ (2,2,0)-block-encoding ของ Φ_F^o . 26
4.4 วงจร (2,1,0)-block-encoding ของ Φ_F^c 26
4.5 ตัวอย่างผลลัพธ์วงจรบล็อกเอ็นโค้ดดิงของ B กรณี $N \neq 2^n$ 30
4.6 วงจร (2,2,0)-block-encoding ของ $\Phi_F'^o, \Phi_F'^c$ 31
4.7 วงจร (2,4,0)-block-encoding ของ ของ B 31
4.8 บล็อกเอ็นโค้ดดิงจาก X gate และ CNOT gate 32
4.9 วงจรบล็อกเอ็นโค้ดดิงของ H 32
4.10 วงจรบล็อกเอ็นโค้ดดิงของ $\tilde{H}$ 33
4.11 วงจรบล็อกเอ็นโค้ดดิงของ e^{-iHt} 34
4.12 วงจรทั้งหมดสำหรับการจำลอง 34
5.1 ผลลัพธ์การทดลองจากการวัดความลึกวงจร : ระบบมวลเส้นตรง 36
5.2 ผลลัพธ์การทดลองจากการวัดความลึกวงจร : ระบบมวลวงกลม 36
5.3 ผลลัพธ์การทดลองระบบมวล 3 ก้อน 37
5.4 ค่าคลาดเคลื่อนสัมบูรณ์ของการทดลองระบบมวล 3 ก้อน 38
5.5 ผลลัพธ์การทดลองระบบมวล 6 ก้อน 39

หน้า

5.6 ค่าคลาดเคลื่อนสัมบูรณ์ของการทดลองระบบมวล 6 ก้อน 40
5.7 ผลลัพธ์การทดลองระบบมวล 7 ก้อน 41
5.8 ค่าคลาดเคลื่อนสัมบูรณ์ของการทดลองระบบมวล 7 ก้อน 42

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

นับตั้งแต่ทศวรรษที่ 1980 เทคโนโลยีด้านควอนตัมคอมพิวเตอร์ถูกพัฒนาอย่างรุดหน้าทั้งในเชิงทฤษฎีคณิตศาสตร์และการพัฒนาฮาร์ดแวร์ คู่ขนานไปกับการพัฒนาการประมวลผลบนทรานซิสเตอร์ในคอมพิวเตอร์คลาสสิก โดยมีจุดประสงค์เพื่อแสวงหาการประมวลผลที่ต่างไปจากการประมวลผลแบบคลาสสิกและเพื่อการจำลองระบบทางฟิสิกส์ที่มีประสิทธิภาพ อัลกอริทึมควอนตัมที่มีชื่อเสียงหลายตัว ได้รับการพิสูจน์ว่าสามารถแก้ไขปัญหาได้เร็วเหนือกว่าอัลกอริทึมแบบคลาสสิก ในปัจจุบัน บริษัทเทคโนโลยีหลายแห่งได้บูรณาการทั้งสองระบบให้ออกมาเป็นแพลตฟอร์มที่รองรับการประมวลผลวงจรควอนตัม ผ่านระบบคลาวด์ เพื่อรองรับการทำงานวิจัยทางควอนตัมคอมพิวเตอร์ทั่วโลก มีควอนตัมหลากหลายประเภทที่ถูกคิดค้นและพัฒนาขึ้นมา เช่น ตัวนำยวดยิ่ง (Superconductor), การกักไอออน (Trapped ions), อะตอมเย็น (Cold atoms), โฟตอน (Photon) เป็นต้น

ในฝั่งของการจำลองควอนตัม (Quantum simulation) เป็นการนำคุณสมบัติทางควอนตัมมาใช้จำลองระบบการคำนวณทางฟิสิกส์ที่ซับซ้อนหรือมีขนาดใหญ่มาก เช่น ระบบทางเคมี, วัสดุศาสตร์, ระบบสสารควบแน่น, ระบบสมการเชิงอนุพันธ์ ซึ่งรวมถึงระบบควอนตัมด้วย เนื่องจากระบบเหล่านี้ต้องอาศัยเวลาและทรัพยากรทางการคำนวณที่เติบโตแบบเลขชี้กำลังเทียบกับขนาดของระบบทำให้ไม่สามารถประมวลผลบนคอมพิวเตอร์คลาสสิกอย่างมีประสิทธิภาพได้ หนึ่งในระบบฟิสิกส์ที่ซับซ้อนที่มีความสัมพันธ์ในรูปของสมการเชิงอนุพันธ์ คือ ระบบการสั่นคู่ควบ (Coupled Oscillator System) ของระบบมวลหลายก้อนที่เชื่อมต่อกันด้วยสปริง ซึ่งระบบนี้ถูกนำไปใช้ในการศึกษาระบบฟิสิกส์ที่มีพฤติกรรมคล้ายกัน เช่น การสั่นของโมเลกุล

หนึ่งในวิธีการจำลองที่ทำบนควอนตัมคอมพิวเตอร์ คือ การนำตัวดำเนินการฮามิลโตเนียน (Hamiltonian operator) ของระบบที่สนใจ มาจำลองบนควอนตัมและศึกษาการวิวัฒน์ไปของระบบ ซึ่งเรียกว่า การจำลองฮามิลโตเนียน (Hamiltonian simulation) มี

เครื่องมือหลายชนิดที่ถูกพัฒนาขึ้นเพื่อช่วยในการจำลองฮามิลโตเนียนนี้ โดยส่วนสำคัญหลักคือการฝังเมทริกซ์กลไกของระบบที่ต้องการศึกษาลงบนฮามิลโตเนียน หนึ่งในวิธีที่ถูกใช้อย่างแพร่หลายคือ เทคนิคบล็อกเอ็นโค้ดดิง (Block encoding) ซึ่งเป็นการแปลงเมทริกซ์ของระบบให้อยู่ในรูปของเมทริกซ์ยูนิแทรี ซึ่งแสดงในรูปของวงจรควอนตัมก่อนนำไปจำลอง

วิทยานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อศึกษาการสร้างบล็อกเอ็นโค้ดดิงของตัวดำเนินการฮามิลโตเนียนเพื่อทำการจำลองระบบการสั่นคู่ควบโดยใช้วงจรควอนตัม การจำลองที่ใช้ในงานวิจัยนี้จะอาศัยการแปลงค่าเอกฐานเชิงควอนตัม (Quantum Singular Value Transformation) เพื่อการทำฟังก์ชั่นของเมทริกซ์ร่วมด้วย และการสร้างวงจรควอนตัมจะใช้ไลบรารี Qiskit ของ IBM เป็นหลัก จากการศึกษาจะทำให้เข้าใจผลลัพธ์การจำลองระบบการสั่นคู่ควบบนควอนตัมเมื่อเทียบกับการจำลองโดยใช้เมทริกซ์คำนวณบนคอมพิวเตอร์คลาสสิก อันนำมาสู่การเปิดมุมมองใหม่ในการจำลองระบบฟิสิกส์ที่ซับซ้อนยิ่งขึ้นโดยใช้ควอนตัมคอมพิวเตอร์

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อศึกษาการสร้างบล็อกเอ็นโค้ดดิงโดยใช้วงจรควอนตัม
2. เพื่อศึกษาการนำบล็อกเอ็นโค้ดดิงมาจำลองฮามิลโตเนียนโดยใช้การแปลงค่าเอกฐานเชิงควอนตัม
3. เพื่อศึกษาผลลัพธ์การจำลองระบบการสั่นคู่ควบโดยใช้วงจรควอนตัม

1.3 ขอบเขตการดำเนินงาน

1. งานวิจัยนี้จะพิจารณาถึงการทำบล็อกเอ็นโค้ดดิงคู่กับ QSVT
2. งานวิจัยนี้จะพิจารณาเฉพาะการสั่นควบคู่ของระบบมวลติดสปริง
3. งานวิจัยนี้จะทำในส่วนอัลกอริทึมและซอฟต์แวร์เป็นหลัก จะไม่มีการลงรายละเอียดในด้านฮาร์ดแวร์ของควอนตัมคอมพิวเตอร์

1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. สามารถอธิบายหลักการทำงานของบล็อกเอ็นโค้ดดิงได้
2. ได้ความรู้ในการสร้างวงจรบล็อกเอ็นโค้ดดิงของเมทริกซ์จากควอนตัมเกตพื้นฐาน
3. ได้แนวทางในการพัฒนาและปรับปรุงวงจรควอนตัม
4. เข้าใจการทำงานของการจำลองฮามิลโตเนียนโดยใช้การแปลงค่าเอกฐานเชิงควอนตัม
5. ได้แนวทางในการสร้างวงจรควอนตัมเพื่อการจำลองระบบมวลคู่ควบ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ จะอธิบายถึงเนื้อหาพื้นฐานทางการคำนวณควอนตัม, การสร้างวงจรควอนตัม, พื้นฐานของระบบการสั่นคู่ควบ และงานวิจัยที่เกี่ยวข้องกับการจำลองควอนตัมและการสร้างบล็อกเอ็นโค้ดดิง ดังนี้

2.1 หลักพื้นฐานของการคำนวณควอนตัม

คณิตศาสตร์ของการคำนวณควอนตัม

ในขณะที่การคำนวณบนคอมพิวเตอร์คลาสสิกอาศัยหน่วยความจำเป็นบิต (bit) ซึ่งมีค่า 0 หรือ 1 แน่นอน การคำนวณบนควอนตัม (quantum computation) อาศัยการทำงานประมวลผลและวัดค่าบนพื้นฐานของหลักกลศาสตร์ควอนตัม (quantum mechanics) และใช้คิวบิต (qubit) เป็นตัวเก็บข้อมูลในสภาวะซ้อนทับ (superposition) กล่าวคืออยู่ในสถานะ 0 กับ 1 พร้อมกัน หากเขียนสถานะทั้งสองให้อยู่ในรูปสัญลักษณ์เค็ท (ket) $|\bullet\rangle$ และเวกเตอร์ จะเขียนได้ว่า $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ และ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ตามลำดับ และสถานะควอนตัม (quantum state) จะเป็นดังนี้

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{2.1}$$

ซึ่ง α และ β แทนแอมพลิจูดของความน่าจะเป็นและมีค่าเป็นจำนวนเชิงซ้อน โดยความน่าจะเป็นของทั้ง 2 สถานะดังกล่าวมีค่า $|\alpha|^2$ และ $|\beta|^2$ ตามลำดับ ดังนั้นผลรวมความน่าจะเป็นคือ

$$|\psi|^2 = |\alpha|^2 + |\beta|^2 = 1 \tag{2.2}$$

คิวบิตจะเปลี่ยนสถานะได้ผ่านเมทริกซ์ตัวดำเนินการ (operator) M ซึ่งเขียนได้ว่า

$$M = c_{00} |0\rangle\langle 0| + c_{01} |0\rangle\langle 1| + c_{10} |1\rangle\langle 0| + c_{11} |1\rangle\langle 1| = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} \tag{2.3}$$

โดย $\langle\bullet|$ เป็นสัญลักษณ์บรา (bra) โดยแทนว่า $\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}$ และ $\langle 1| = \begin{pmatrix} 0 & 1 \end{pmatrix}$ กล่าวคือ เป็นรูปสลับเปลี่ยนสังยุค (conjugate transpose) ของสัญลักษณ์เค็ทนั่นเอง เราเรียกการเขียนสัญลักษณ์เค็ทและบรานี้ว่า สัญกรณ์บรา-เค็ทของดิแรก (Dirac bra-ket notation) ซึ่งผลลัพธ์ของสถานะหลังผ่านตัวดำเนินการจะเป็นดังนี้

$$M|\psi\rangle = (c_{00}\alpha + c_{01}\beta)|0\rangle + (c_{10}\alpha + c_{11}\beta)|1\rangle \tag{2.4}$$

ลักษณะของตัวดำเนินการในระบบควอนตัมจะเป็นเมทริกซ์ยูนิแทรี (unitary matrix) เสมอ กล่าวคือ รูปของเมทริกซ์ผกผัน (inverse matrix) จะเท่ากับรูปสลับเปลี่ยนสังยุคของเมทริกซ์ ($M^{-1} = M^{\dagger}$)

สำหรับระบบหลายคิวบิต จะสามารถเขียนความสัมพันธ์ระหว่างหลายสถานะควอนตัมได้ด้วยผลคูณเทนเซอร์ (tensor product) ทำให้เขียนสถานะรวมทั้งหมดได้ว่า

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle = \sum_{k=0}^{2^n-1} \gamma_k |k\rangle \tag{2.5}$$

และเมทริกซ์ของตัวดำเนินการ **M** ที่กระทำผ่านหลายคิวบิตสามารถเขียนได้ดังนี้ โดย i เป็นค่าของแถว (row) และ j เป็นค่าของคอลัมน์ (column)

$$\mathbf{M} = \sum_{i,j} c_{ij} |i\rangle\langle j| \tag{2.6}$$

เมื่อทำการวัดค่าสถานะ จะหาค่าจากผลคูณภายใน (inner product) โดยโอกาสที่จะวัดได้ค่าสถานะ $m; m \in \{0, 1, ..., 2^n - 1\}$ เท่ากับ

$$|\langle m|\Psi\rangle|^2 = |\gamma_m|^2 \tag{2.7}$$

และผลรวมความน่าจะเป็นของทุกสถานะที่เป็นไปได้ของทุกคิวบิตย่อมเป็น 1 เช่นกัน

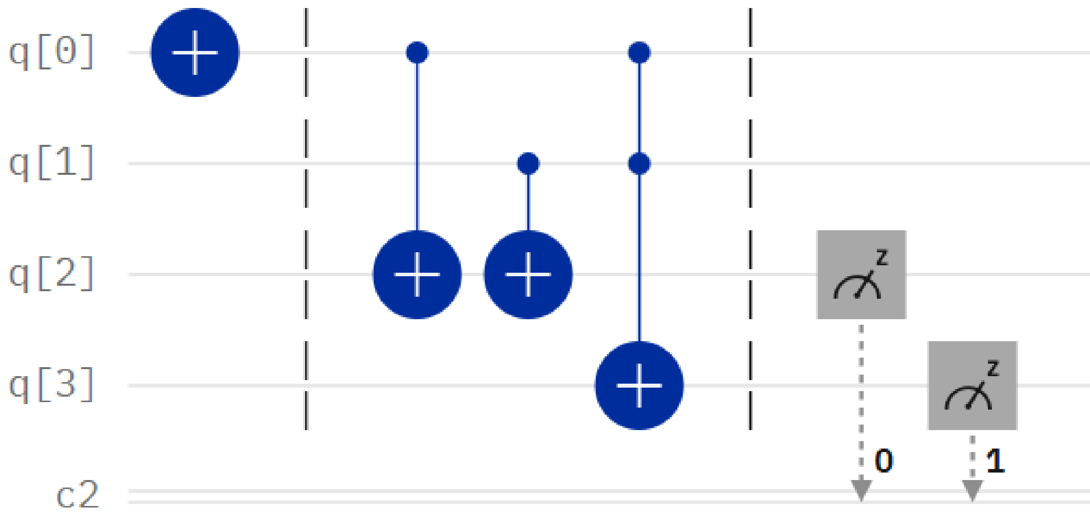
$$|\Psi|^2 = \sum_{k=0}^{2^n-1} |\gamma_k|^2 = 1 \tag{2.8}$$

สัญลักษณ์ทางพีชคณิตเหล่านี้ถูกใช้อย่างแพร่หลายในการคำนวณทางกลศาสตร์ควอนตัม และวิทยาการข้อมูลควอนตัม (quantum information science) รวมถึงช่วยแสดงถึงขั้นตอนวิธีของอัลกอริทึมควอนตัม (quantum algorithm)

อัลกอริทึมควอนตัมและวงจรควอนตัม

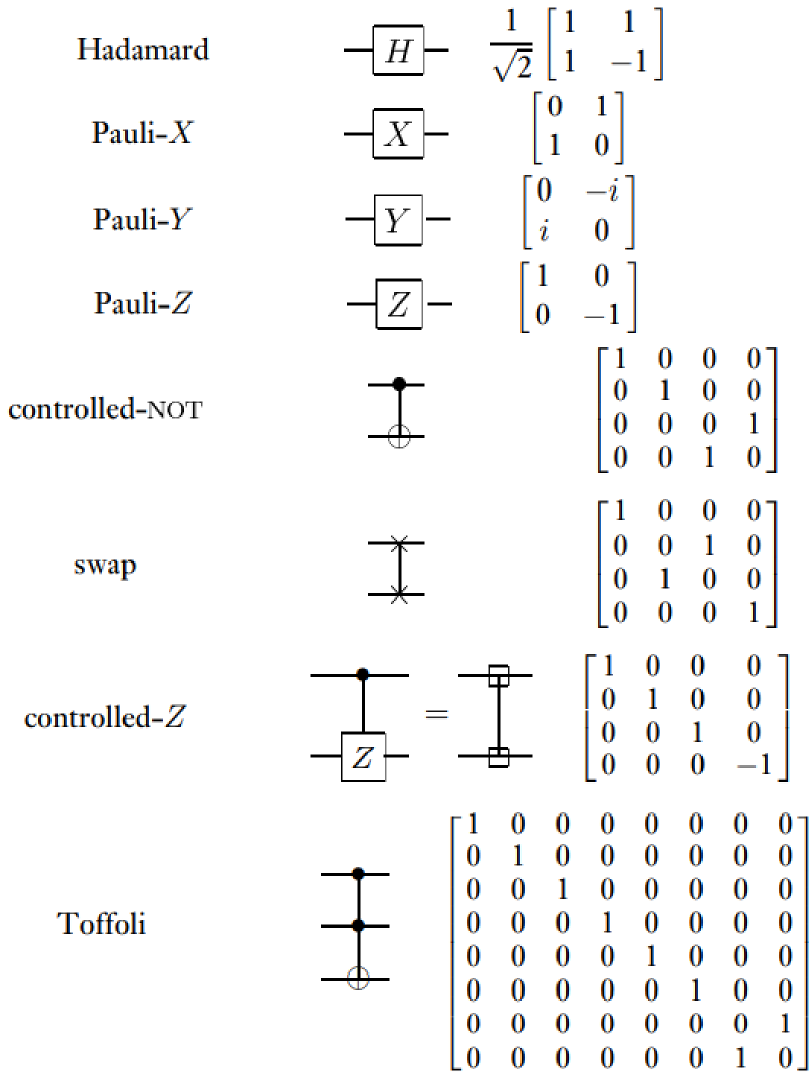
อัลกอริทึมควอนตัมเป็นอัลกอริทึมที่ถูกสร้างขึ้นโดยมักอาศัยคุณสมบัติการซ้อนทับและความพัวพัน (entanglement) เพื่อเพิ่มประสิทธิภาพให้เหนือกว่าอัลกอริทึมแบบคลาสสิกสำหรับการแก้ปัญหายากทางคณิตศาสตร์ ตัวอย่างอัลกอริทึมควอนตัมที่มีชื่อเสียง ได้แก่ Deutsch-Josza algorithm, Simon's algorithm, Bernstein-Varizani algorithm, Shor's algorithm, Grover's algorithm ฯลฯ ซึ่งองค์ประกอบในบางอัลกอริทึมถูกใช้ต่อยอดเพื่อสร้างอัลกอริทึมใหม่ ๆ เช่น การขยายแอมพลิจูดของสถานะ (amplitude amplification)

สิ่งที่สามารถแสดงอัลกอริทึมควอนตัมดังกล่าวในรูปที่ชัดเจนและได้รับความนิยม คือภาพวงจรควอนตัม (quantum circuit) ซึ่งในวงจรประกอบด้วยคิวบิตและรีจิสเตอร์แบบคลาสสิก (classical register) สำหรับการบันทึกค่าการวัด มักถูกวาดเป็นเส้นแนวนอนโดยมีควอนตัมเกต (quantum gate) เป็นตัวแทนของตัวดำเนินการวางพาดผ่านเส้นของคิวบิตเหล่านั้น และมีการวัด (measurement) ที่มีสัญลักษณ์เป็นเครื่องวัดวางบนเส้นคิวบิตและมีเส้นชี้ทางข้อมูลการอ่านไปยังรีจิสเตอร์คลาสสิก ดังตัวอย่างในรูปที่ 2.1 แสดงวงจรควอนตัมสำหรับการบวกเลข



รูปที่ 2.1: ตัวอย่างวงจรควอนตัมสำหรับการบวกเลข 1 บิต (Single bit half-adder)

ควอนตัมเกตมีอยู่หลายประเภท โดยพื้นฐานแบ่งออกคร่าว ๆ ได้แก่ เกตที่ทำบนคิวบิตเดียว (single-qubit gate) ได้แก่ Pauli-X, Pauli-Y, Pauli-Z gate, Hadamard gate, rotation gate และเกตที่ทำบนหลายคิวบิต (multiple qubit gate) เช่น CNOT gate, Toffoli gate ดังตัวอย่างในรูปที่ 2.2 เกตแต่ละเกตต่างก็แสดงถึงเมทริกซ์ของตัวดำเนินการที่แตกต่างกัน แต่มีจุดร่วมคือเมทริกซ์ทุกตัวเป็นเมทริกซ์ยูนิแทรีเหมือนกัน ทำให้มีคุณสมบัติในการย้อนกลับสถานะควอนตัมได้ (reversible) หลายควอนตัมเกตก็มีคุณสมบัติเป็นเมทริกซ์เฮอร์มิเทียน (Hermitian matrix) โดยเมทริกซ์มีค่าเท่ากับรูปสลับเปลี่ยนสังยุคของตัวมันเอง ($M = M^{\dagger}$) อีกด้วย ในงานวิจัยฉบับนี้จะเรียกชื่อควอนตัมเกตเป็นภาษาอังกฤษเพื่อความสะดวกแก่การอธิบาย อย่างไรก็ตาม มีหลายอัลกอริทึมควอนตัมที่ยังไม่สามารถสร้างวงจรที่ประกอบด้วยเกตพื้นฐานได้ ซึ่งภายในวงจรจะเขียนอยู่ในรูปกล่องสี่เหลี่ยมเปล่าแทน เรียกว่าออราเคิล (oracle)



รูปที่ 2.2: รายชื่อควอนตัมเกต สัญลักษณ์และเมทริกซ์ (Nielsen and Chuang, 2019)

ส่วนควอนตัมคอมพิวเตอร์ (quantum computer) เป็นเครื่องคำนวณที่ทำงานด้วยหลักการข้างต้นโดยรับข้อมูลวงจรควอนตัมเข้าไปประมวลผล ซึ่งอุปกรณ์ที่ใช้แทนคิวบิต เครื่องมือที่ใช้เปลี่ยนแปลงสถานะควอนตัมหรือเครื่องมือวัดจะแตกต่างกันไปในเครื่องคอมพิวเตอร์ควอนตัมแต่ละประเภท ตัวอย่างระบบที่ถูกพัฒนาขึ้นได้แก่ ตัวนำยวดยิ่ง (superconductor), โฟตอน (photon), การกักไอออน (trapped ion), อะตอมเย็น (cold atoms) เป็นต้น โดยเกณฑ์คุณสมบัติที่จำเป็นของควอนตัมคอมพิวเตอร์ถูกกำหนดตามหลักเกณฑ์ของดิวินเซนโซ (DiVincenzo's criteria) (DiVincenzo, 1996) ในปัจจุบัน งานวิจัยด้านฮาร์ดแวร์ควอนตัมคอมพิวเตอร์จะมุ่งเน้นไปที่การสร้างเครื่องที่มีจำนวนคิวบิตมากขึ้นและลดข้อผิดพลาดจากสัญญานรบกวน

2.2 การจำลองควอนตัม

การจำลองควอนตัมและการจำลองฮามิลโตเนียน

การจำลองควอนตัม (quantum simulation) เป็นศาสตร์ที่ศึกษาวิธีการนำระบบควอนตัมมาใช้จำลองระบบทางฟิสิกส์หรือปรากฏการณ์ที่สามารถอธิบายได้ทางคณิตศาสตร์ โดยเป้าหมายของการจำลองควอนตัมมุ่งเน้นไปยังระบบที่มีความซับซ้อนหรือมีข้อมูลจำนวนมากเกินกว่าที่คอมพิวเตอร์ธรรมดาหรือซุปเปอร์คอมพิวเตอร์สามารถประมวลผลเพื่อจำลองได้ ตัวอย่างเช่น การจำลองระบบควอนตัมด้วยกันเอง, การทำพันธะเคมี, การเปลี่ยนแปลงเฟส (phase transition) ในงานวัสดุศาสตร์ , พฤติกรรมทางแม่เหล็ก (Monroe et al., 2021) ฯลฯ ความคิดเริ่มแรกในการจำลองควอนตัมถูกเสนอโดย ยูริ มานิน (Yuri Manin) ในปี 1980 (Manin, 1980) และ ริชาร์ด ไฟน์แมน (Richard Feynman) ในปี 1982 (Feynman, 1982) จนถึงปัจจุบัน แนวคิดและวิธีการมากมายถูกเสนอขึ้นมาเพื่อให้สอดคล้องกับระบบที่ต้องการจำลอง การใช้ควอนตัมคอมพิวเตอร์ก็เป็นทางเลือกหนึ่งเช่นกัน (Lloyd, 1996) และเนื่องจากระบบควอนตัมจะเปลี่ยนแปลงไปตามเวลาด้วยขึ้นกับฮามิลโตเนียน (Hamiltonian) H ของระบบตามผลเฉลยของสมการชเรอดิงเงอร์ (Schrödinger's equation) ดังนี้

$$|\psi(t)\rangle = e^{-i\frac{H}{\hbar}t}|\psi(0)\rangle \tag{2.9}$$

หากเราสามารถแปลงค่าหรือเงื่อนไขเริ่มต้นของระบบที่สนใจลงในสถานะควอนตัม ณ $t = 0$ ได้ และฝังข้อมูลกลไกของระบบให้อยู่ในรูปเมทริกซ์ฮามิลโตเนียนได้ จะสามารถจำลองระบบ ณ เวลา t ใดๆได้ ซึ่งเราเรียกการจำลองโดยอิงฮามิลโตเนียนนี้ว่า การจำลองฮามิลโตเนียน

(Hamiltonian simulation) โดยรูปแบบของเมทริกซ์ข้อมูลที่ถูกฝังในฮามิลโตเนียนสามารถจำแนกได้เป็น 3 โมเดล (Dalzell et al., 2023) ได้แก่

1. โมเดลเมทริกซ์เพาลี (Pauli input model) คือ เมทริกซ์ข้อมูลที่สามารถเขียนอยู่ในรูปผลรวมของตัวดำเนินการเพาลีบนหลายคิวบิต H_l ได้ เช่น $H = \sum h_l H_l$ ซึ่ง h_l คือค่าสัมประสิทธิ์ (coefficients)
2. โมเดลเมทริกซ์มากเลขศูนย์ (Sparse matrix model) คือ เมทริกซ์ข้อมูลที่ค่าภายในส่วนใหญ่เป็นศูนย์
3. โมเดลเมทริกซ์หนาแน่น (Dense matrix model) คือ เมทริกซ์ข้อมูลที่ค่าภายในส่วนใหญ่ไม่เป็นศูนย์

การจำลองฮามิลโตเนียนอาจประพฤติตัวเป็นฟังก์ชันการทำงานย่อยของอัลกอริทึมอื่นอีกที เช่น การประมาณเฟสเชิงควอนตัม (quantum phase estimation) การแก้ระบบสมการเชิงเส้นเชิงควอนตัม (quantum linear solver system) ตัวอย่างของการจำลองฮามิลโตเนียนที่ได้รับความนิยม ได้แก่

1. ทรอตเทอไรเซชัน (Trotterization หรือ Trotter-Suzuki formulae) (Hatano and Suzuki, 2005) เป็นวิธีแยกฮามิลโตเนียนออกเป็นพจน์ฮามิลโตเนียนย่อย และนำมารวมกันโดยใช้สูตรผลคูณลี (Lie product formula)
2. qDRIFT (The quantum stochastic drift protocol) (Campbell, 2019) เป็นวิธีแยกที่คล้ายกับทรอตเทอไรเซชัน แต่มีโมเดลการแยกแบบเมทริกซ์เพาลีร่วมด้วย
3. อนุกรมเทย์เลอร์และอนุกรมไดสัน (Taylor and Dyson series) (Berry et al., 2015; Kieferová et al., 2019) เป็นวิธีแยกฮามิลโตเนียนออกมาตามสมการอนุกรมข้างต้น โดยมีการตัดปลาย (truncate) พจน์ที่มีขนาดน้อยลงไปโดยยอมรับค่าคลาดเคลื่อนในช่วงหนึ่ง เป็นโมเดลการแยกแบบเมทริกซ์เพาลี และนำมารวมกันด้วยยูนิแทรีของการรวมเชิงเส้น (Linear combination of unitaries, ย่อว่า LCU)
4. การแปลงค่าเอกฐานเชิงควอนตัม (QSVT)

สำหรับในงานนี้ได้ใช้การจำลองในแบบที่ 4 เป็นหลัก โดยจะกล่าวถึงรายละเอียดในส่วนถัดไป

จากกรณีที่ยกตัวอย่างข้างต้น จะเห็นได้ว่าจะต้องมีการแบ่งฮามิลโตเนียนออกเป็นขนาดย่อยเพื่อให้สะดวกต่อการจำลอง อย่างไรก็ตาม เมทริกซ์ของระบบที่นำมาใส่ในฮามิลโตเนียนนั้นอาจไม่ได้อยู่ในรูปเมทริกซ์ยูนิแทรี ดังนั้น จึงจำเป็นที่จะต้องแปลงเมทริกซ์ให้อยู่ในรูปเมทริกซ์ยูนิแทรีเสียก่อน โดยเทคนิคบล็อกเอ็นโค้ดดิง (block encoding) เข้ามามีบทบาท ณ จุดนี้

เทคนิคบล็อกเอ็นโค้ดดิง

เทคนิคนี้ถูกเสนอขึ้นโดย (Gilyén et al., 2019) เพื่อจุดประสงค์ในการฝังเมทริกซ์ใด ๆ ลงในบล็อกตัวดำเนินการที่เป็นเมทริกซ์ยูนิแทรีซึ่งเป็นวิธีทั่วไปสำหรับสร้างการคำนวณเมทริกซ์บนควอนตัมคอมพิวเตอร์ ลักษณะเมทริกซ์บล็อกเอ็นโค้ดเป็นดังนี้ โดยเมทริกซ์ใด ๆ A จะถูกหารด้วยค่าคงที่ซับนอร์มอลไลซ์ (subnormalized constant) α โดยค่า $\alpha \geq \|A\|$ และฝังในตำแหน่งมุมซ้ายบนของเมทริกซ์บล็อกเอ็นโค้ด U

$$U = \begin{bmatrix} \frac{A}{\alpha} & * \\ * & * \end{bmatrix} \tag{2.10}$$

โดย $*$ แทนค่าที่ทำให้ U เป็นยูนิแทรี

บทนิยาม 1 (บล็อกเอ็นโค้ดดิง (Gilyén et al., 2019)) *ถ้ากำหนดให้ A เป็นเมทริกซ์ขนาด $m \times n$ ที่ต้องใช้อย่างน้อย s คิวบิต เรียกว่า s-qubit operator กล่าวคือ $m, n \leq 2^s$ และให้ $\alpha, \epsilon \in \mathbb{R}^+$ และ $a \in \mathbb{N}$ แล้วเมทริกซ์ยูนิแทรี U ขนาด $s + a$ คิวบิต จะเป็น (α, a, ϵ)-block-encoding ของ A เมื่อ*

$$\|A_e - \alpha(\langle 0|^{\otimes a} \otimes I_s)U(|0\rangle^{\otimes a} \otimes I_s)\| \leq \epsilon \tag{2.11}$$

โดยที่ α คือ ค่าคงที่ซับนอร์มอลไลซ์, ϵ คือ ค่าคลาดเคลื่อน, a คือ จำนวนคิวบิตตัวทด (ancilla qubit), I_s คือ เมทริกซ์เอกลักษณ์ (identity matrix) ขนาด $2^s \times 2^s$ และ A_e คือ เมทริกซ์ขนาด $2^s \times 2^s$ ที่ฝัง A ไว้ชิดมุมซ้ายบน

หากกำหนดให้ $|\psi\rangle$ เป็นสถานะควอนตัมจำนวน s คิวบิต ผลลัพธ์จากการป้อน $|\psi\rangle|0\rangle^{\otimes a}$ ผ่าน U ตามนิยาม 1 จะได้ว่า

$$U|\psi\rangle|0\rangle^{\otimes a} = \frac{A}{\alpha}|\psi\rangle|0\rangle^{\otimes a} + |\phi^{+}\rangle \tag{2.12}$$

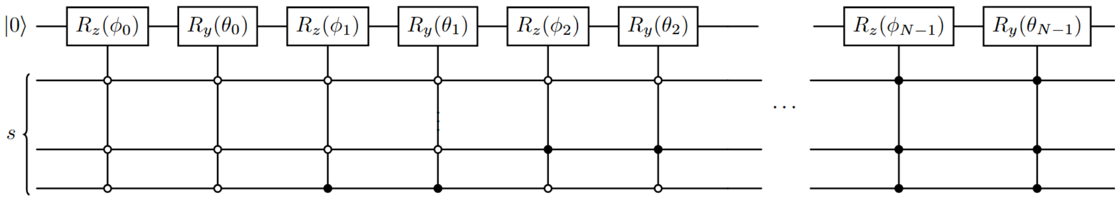
โดย $(I_s \otimes |0\rangle\langle 0|^{\otimes a})|\phi^{+}\rangle = 0$ กล่าวคือ สถานะที่คิวบิตตัวทดไม่เป็นศูนย์จะมีแอมพลิจูดเป็นศูนย์นั่นเอง (Chakraborty et al., 2023)

บทนิยาม 2 (บล็อกเอ็นโค้ดดิงโดยปริยาย (Gilyén et al., 2019)) *เมทริกซ์ยูนิแทรีใด ๆ เป็น* $(1, 0, 0)$*-block-encoding ของตัวมันเองเสมอ*

วิธีการสร้างวงจรบล็อกเอ็นโค้ดดิงมีหลายรูปแบบและไม่ตายตัว ตัวอย่างของบล็อกเอ็นโค้ดดิงที่ถูกเสนอใน (Gilyén et al., 2019) เช่น การสร้างเมทริกซ์ความหนาแน่น (Density operator), การสร้างเมทริกซ์มากเลขศูนย์ ในปัจจุบัน มีเฟรมเวิร์คที่ถูกพัฒนาขึ้นเพื่อหาวงจรบล็อกเอ็นโค้ดดิง เช่น FABLE (Camps and Beeumen, 2022) แต่อย่างไรก็ตาม งานวิจัยด้านนี้ยังคงมุ่งเน้นการหาวิธีปรับปรุงวงจรบล็อกเอ็นโค้ดดิงเพื่อให้การทำงานมีประสิทธิภาพสูงสุด ตัวอย่างของบล็อกเอ็นโค้ดดิงที่ใช้ในงานวิจัยนี้คือ บล็อกเอ็นโค้ดดิงของเมทริกซ์ทแยงมุม (diagonal matrix)

บทตั้ง 1 (บล็อกเอ็นโค้ดดิงของเมทริกซ์ทแยงมุม (Takahira et al., 2021)) *กำหนดให้* D *เป็นเมทริกซ์ทแยงมุมที่ใช้* s *คิวบิต โดยมีค่าในแนวทแยงมุมเป็น* $d_0, d_1, ..., d_{N-1}; d_k \in \mathbb{C}$ *และให้* $d_{\max}$ *แสดงค่า* d *ที่มากที่สุด และให้* $\theta_k = 2\cos^{-1}(\frac{|d_k|}{d_{\max}})$ *และ* $\phi_k = 2\arg(d_k)$ *แล้ว* $(d_{\max}, 1, 0)$*-block-encoding ของ* D *สามารถสร้างได้ด้วย* $\sum_k (R_z(\phi_k)R_y(\theta_k)) \otimes |k\rangle\langle k|$ *เมื่อ* $R_y(\theta) = e^{i(\frac{\theta}{2})\sigma_y}$ *และ* $R_z(\phi) = e^{i(\frac{\phi}{2})\sigma_z}$

สามารถอธิบายได้ว่า วงจรบล็อกเอ็นโค้ดดิงประกอบด้วย multi-controlled rotation gate ซึ่งการหมุนในแนวแกน Y แทนค่า $\mathrm{Re}(d_k)$ และการหมุนในแนวแกน Z แทนค่า $\mathrm{Im}(d_k)$ นั่นเอง โดยวงจรจะมีลักษณะเป็นดังรูป 2.3



รูปที่ 2.3: วงจรบล็อกเอ็นโค้ดดิงของเมทริกซ์ทแยงมุม

ทั้งนี้ เรายังสามารถสร้างบล็อกเอ็นโค้ดดิงของการดำเนินการพื้นฐานได้เช่นกัน เช่น การรวมเชิงเส้น (การบวก) และการคูณเมทริกซ์

บทตั้ง 2 (การรวมเชิงเส้นของยูนิแทรี (LCU) (Gilyén et al., 2019)) *กำหนดให้* $A = \sum_{j=0}^{m-1} y_j A_j$ *เป็นตัวดำเนินการที่ใช้* s *คิวบิต, ให้คู่ยูนิแทรีสำหรับการเตรียมสถานะ* P_L, P_R *เรียกว่า* (β, b, ϵ_1)*-state-preparation-pair สำหรับ* y *โดย*

$$P_L |0\rangle^{\otimes b} = \sum_{j=0}^{2^b-1} c_j |j\rangle \tag{2.13}$$

และ

$$P_R |0\rangle^{\otimes b} = \sum_{j=0}^{2^b-1} d_j |j\rangle \tag{2.14}$$

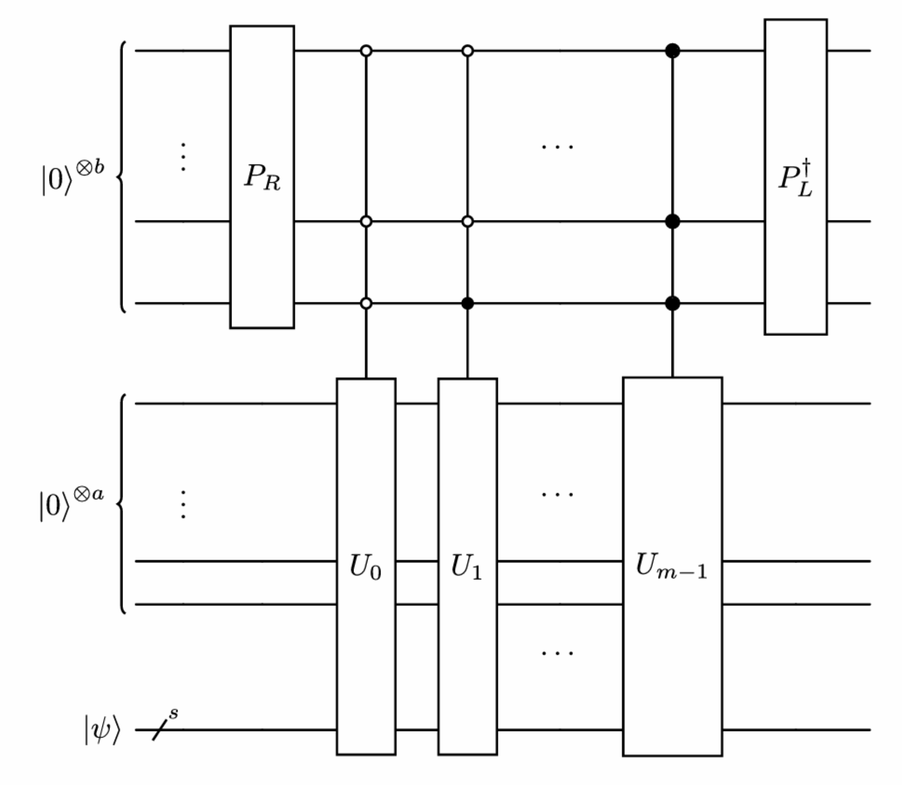
ซึ่งอยู่ภายใต้เงื่อนไข $\sum_{j=0}^{m-1} |\beta(c_j^* d_j) - y_j| \leq \epsilon$ *และ* $c_j^* d_j = 0$ *สำหรับ* $j \in \{m, ..., 2^b - 1\}$ *และให้*

$$W = \sum_{j=0}^{m-1} |j\rangle\langle j| \otimes U_j + \sum_{j=m}^{2^b-1} |j\rangle\langle j| \otimes I_a \otimes I_s \tag{2.15}$$

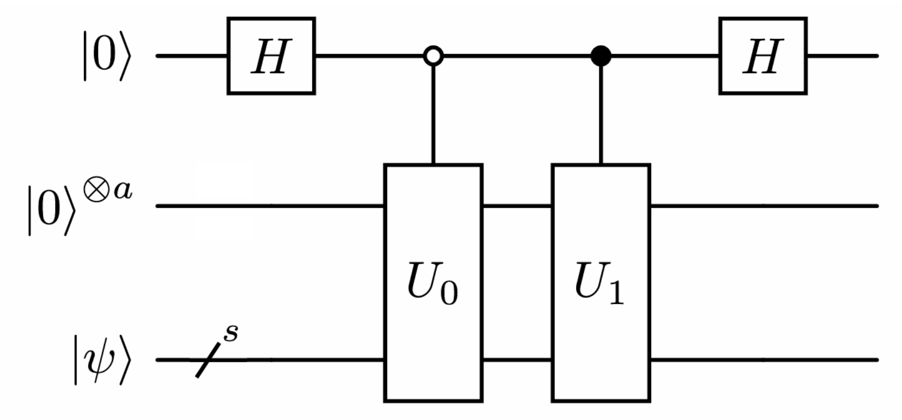
เป็นยูนิแทรีที่ใช้ $s + a + b$ *คิวบิต โดยที่* U_j *เป็น* (α, a, ϵ_2)*-block-encoding ของ* A_j *แล้วผลรวมเชิงเส้นของยูนิแทรีในรูป* $(\alpha\beta, a + b, \alpha\epsilon_1 + \alpha\epsilon_2)$*-block-encoding ของ* A *คือ* $(P_L^\dagger \otimes I_a \otimes I_s) W (P_R \otimes I_a \otimes I_s)$

หากสร้างวงจรด้วยบทตั้งดังกล่าวจะแสดงได้ดังรูปที่ 2.4

โดยเกต P_R และ P_L เป็นเกตใดก็ได้ที่สอดคล้องกับนิยามข้างต้น เช่น Hadamard gate ยกตัวอย่างวงจรที่แสดงการบวกระหว่าง 2 บล็อกเอ็นโค้ด เป็นดังรูป 2.5



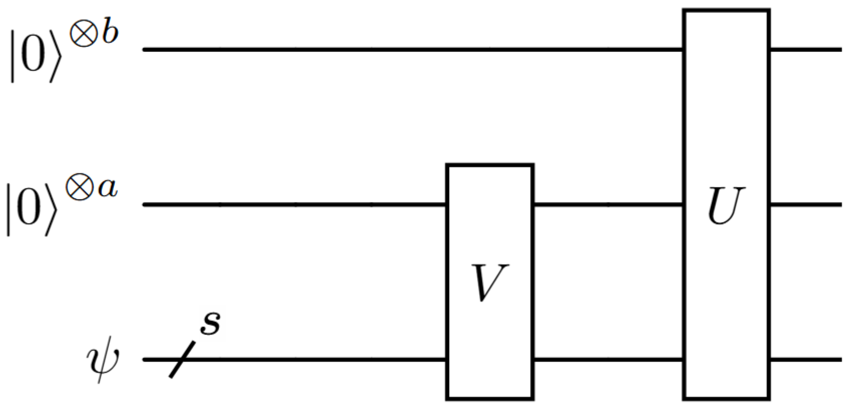
รูปที่ 2.4: วงจรการรวมเชิงเส้นของยูนิแทรีหลายตัว



รูปที่ 2.5: วงจรการรวมเชิงเส้นของ 2 ยูนิแทรี

บทตั้ง 3 (บล็อกเอ็นโค้ดดิงของการคูณเมทริกซ์ (Gilyén et al., 2019)) *กำหนดให้* U *เป็น* (α, a, δ_1)*-block-encoding ของเมทริกซ์* A *ที่ดำเนินการโดยใช้* s *คิวบิต และ* V *เป็น* (β, b, δ_2)*-block-encoding ของเมทริกซ์* B *ที่ดำเนินการโดยใช้* s *คิวบิต แล้วผลคูณ* $(\alpha\beta, a+b, \alpha\delta_2 + \beta\delta_1)$*-block-encoding ของ* AB *คือ* $(I_b \otimes U)(I_a \otimes V)$

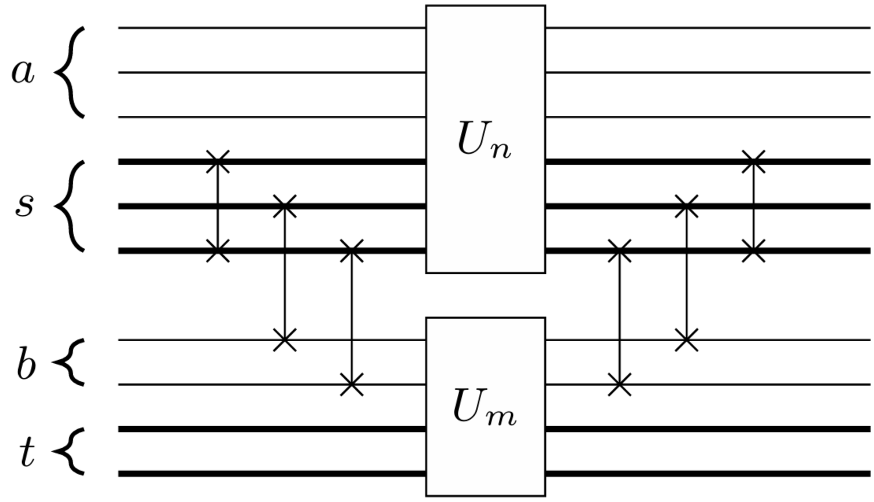
หากสร้างวงจรด้วยบทตั้งดังกล่าวจะแสดงได้ดังรูปที่ 2.6



รูปที่ 2.6: วงจรบล็อกเอ็นโค้ดดิงของการคูณเมทริกซ์

บทตั้ง 4 (บล็อกเอ็นโค้ดดิงของการคูณเทนเซอร์ (Camps and Van Beeumen, 2020)) *กำหนดให้* U_n *เป็น* (α, a, ν_1)*-block-encoding ของ* A_s *และ* U_m *เป็น* (β, b, ν_2)*-block-encoding ของ* A_t *และให้* $S_{n+m} = \prod_{i=1}^{s} \mathrm{SWAP}_{a+b+i}^{a+i}$ *แล้ว* $S_{n+m}(U_n \otimes U_m)S_{n+m}^{\dagger}$ *จะเป็น* $(\alpha\beta, a+b, \alpha\nu_2 + \beta\nu_1 + \nu_1\nu_2)$*-block-encoding ของ* $A_s \otimes A_t$

ซึ่ง SWAP ก็คือ swap gate ในรูปที่ 2.2 นั่นเอง หากสร้างวงจรด้วยบทตั้งดังกล่าวจะแสดงได้ดังรูปที่ 2.7

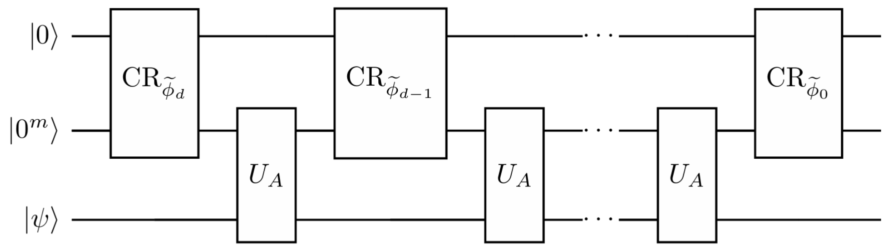


รูปที่ 2.7: บล็อกเอ็นโค้ดดิงของการคูณเทนเซอร์ (Camps and Van Beeumen, 2020)

จะเห็นได้ว่า การสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์และการดำเนินการนั้นสามารถใช้เกตพื้นฐานที่กล่าวไปก่อนหน้าได้ แต่หากต้องการป้อนฟังก์ชันลงไปในเมทริกซ์นั้น จำเป็นต้องอาศัยการแปลงค่าเอกฐานเชิงควอนตัมเข้ามาช่วย

การแปลงค่าเอกฐานเชิงควอนตัม

การแปลงค่าเอกฐานเชิงควอนตัม (Quantum Singular Value Transformation ย่อว่า QSVT) เป็นอัลกอริทึมที่ได้จากการปรับปรุงการประมวลผลสัญญาณเชิงควอนตัม (Quantum Signal Processing ย่อว่า QSP) เพื่อรองรับการแปลงให้เป็นผลลัพธ์ของฟังก์ชันพหุนามของเมทริกซ์บล็อกเอ็นโค้ด โดยอาศัยคุณสมบัติของเมทริกซ์ A ใด ๆ ที่สามารถเขียนในรูปการแยกเอกฐาน $A = WDV^\dagger$ ได้และทำการป้อนฟังก์ชันพหุนามไปยังค่าลักษณะเฉพาะ (eigenvalue) ของเมทริกซ์บล็อกเอ็นโค้ดนั้น วงจรนี้จำเป็นต้องมีการรับชุดค่าเฟสเป็นอินพุตเพื่อให้ได้มาซึ่งฟังก์ชั่นที่ต้องการแปลง โดยรายละเอียดและบทพิสูจน์ปรากฏอยู่ใน (Gilyén et al., 2019) ทั้งนี้ หากเมทริกซ์ A เป็นเมทริกซ์จตุรัสก็สามารถแปลงออกมาได้เช่นกัน ในบางงานวิจัยมีการเขียนแยกออกมาเป็นอีกสมการ ชื่อว่าการแปลงค่าลักษณะเฉพาะเชิงควอนตัม (Quantum Eigenvalue Transformation ย่อว่า QET) (Martyn et al., 2023) โดยลักษณะของวงจรจะมีลักษณะดังรูป 2.8 ซึ่ง $CR_{\tilde{\phi}_i}$ เรียกว่า ตัวดำเนินการเลื่อนเฟสควบคุมการฉาย (projector-controlled phase shift operator) เป็นวงจรที่รับค่าเฟสดังกล่าว

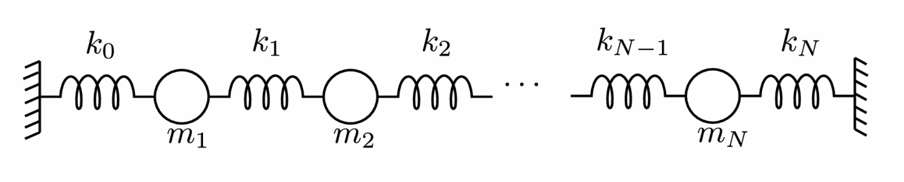


รูปที่ 2.8: วงจร QET(QSVT) (Lin, 2022)

ทั้งนี้ นอกจากฟังก์ชันพหุนามแล้ว ฟังก์ชันตรีโกณมิติอย่าง ฟังก์ชันไซน์ (sine) และฟังก์ชันโคไซน์ (cosine) ก็สามารถใช้งานร่วมกับ QSVT ได้เช่นกัน โดยอาศัยการตัดทอนสมการการกระจายจาโคบี-แองเกอร์ (Jacobi-Anger expansion) และใช้วิธีการปรับสเกลค่าผลลัพธ์เพื่อจำกัดกรอบค่าความคลาดเคลื่อน โดยวิธีการและบทพิสูจน์ถูกกล่าวไว้ใน (Gilyén et al., 2019; Martyn et al., 2021)

2.3 การสั่นคู่ควบ

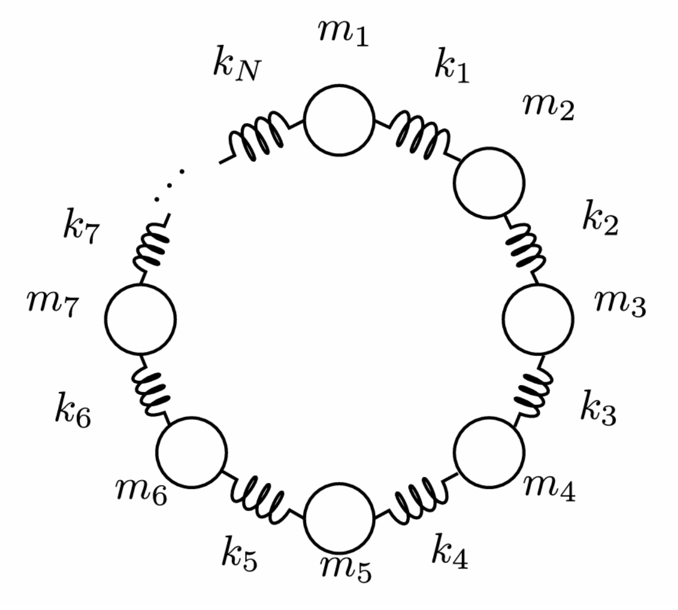
ในกลศาสตร์คลาสสิก เราเรียกกลไกการแกว่งหรือการสั่นเป็นคาบที่ไม่มีแรงภายนอกมากระทำว่า การแกว่งแบบฮาร์โมนิกกอย่างง่าย (Simple Harmonic Oscillation), การแกว่งที่การกระจัดลดลงจากการสูญเสียพลังงานว่า การแกว่งที่ถูกหน่วง (Damped Harmonic Oscillation) และการแกว่งที่มีแรงภายนอกมากระทำว่า การแกว่งที่ถูกแรงบังคับ (Forced Harmonic Oscillation) ซึ่งสามารถใช้กฎข้อที่สองของนิวตันและกฎของฮุค (Hooke's law) อธิบายพฤติกรรมของมวลลูกตุ้มหรือมวลติดสปริงก้อนเดียวได้โดยง่าย (Limkumnerd, 2016) อย่างไรก็ดี ระบบทางฟิสิกส์ที่พบเจอได้ในความเป็นจริงมักเป็นการสั่นของระบบมวลหลายก้อนที่มีปฏิสัมพันธ์กันและส่งผ่านพลังงานไปมาระหว่างมวลที่อยู่ติดกัน เราเรียกกันแกว่งหรือการสั่นแบบนี้ว่า การสั่นคู่ควบ (Coupled Oscillator System) ยกตัวอย่างระบบมวลติดสปริง N ก้อนแบบเส้นตรง ดังรูปที่ 2.9 สามารถเขียนสมการตามกฎของนิวตันอยู่ในรูปเมทริกซ์ได้ดังนี้



รูปที่ 2.9: ระบบมวลติดสปริงแบบเส้นตรง

$$\begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_N \end{bmatrix} \begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_N \end{pmatrix} = - \begin{bmatrix} k_0 + k_1 & -k_1 & & \\ -k_1 & k_0 + k_1 & -k_2 & \\ & & \ddots & \\ & & -k_{N-1} & k_{N-1} + k_N \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \tag{2.16}$$

หรือหากเปลี่ยนให้สปริงของมวลก้อนสุดท้ายเชื่อมกับมวลก้อนแรก กลายเป็นระบบมวลเชื่อมกันเป็นวงกลม ดังรูปที่ 2.10 ก็สามารถเขียนในรูปเมทริกซ์ได้ดังนี้



รูปที่ 2.10: ระบบมวลติดสปริงแบบวงกลม

$$\begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_N \end{bmatrix} \begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_N \end{pmatrix} = - \begin{bmatrix} k_N + k_1 & -k_1 & & -k_N \\ -k_1 & k_1 + k_2 & -k_2 & \\ & & \ddots & \\ -k_N & & -k_{N-1} & k_{N-1} + k_N \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \tag{2.17}$$

ซึ่งสองสมการข้างต้นสามารถนำไปจัดรูปเพิ่มเติม เพื่อนำไปจำลองโดยใช้อัลกอริทึมควอนตัมได้ โดยจากงานวิจัยของ Babbush et al. (2023) ซึ่งศึกษาการจำลองระบบการสั่นคู่ควบบนควอนตัมได้เสนออัลกอริทึมเพื่อจำลองระบบดังกล่าว โดยอาศัยการกระจัดเริ่มต้นและความเร็วเริ่มต้นของมวลในระบบมารวมกับตัวดำเนินการก่อนบรรจุใส่ในสถานะควอนตัม และคำนวณผ่านตัวดำเนินการควอนตัมที่วิวัฒน์ไปในเวลา (time-evolutional quantum operator) สถานะปลายทางจะนำมาใช้หาการกระจัดและความเร็วของมวลทุกก้อน ณ เวลา t ได้ โดยเริ่มแรก เราสามารถเขียนสมการ 2.16 และ 2.17 ในรูปของตัวแปรได้ว่า

$$M\ddot{\vec{x}}(t) = -F\vec{x} \tag{2.18}$$

โดยให้ค่าในเมทริกซ์ของ F ในแนวทแยงมุมเป็นผลบวกของค่านิจสปริงที่ติดกับมวลนั้น และให้ค่าในแนวเบี่ยงจากทแยงมุม (off-diagonal) เป็นค่านิจสปริงที่ติดลบ และให้ M เป็นเมทริกซ์ทแยงมุมของมวล ทั้ง M และ F เป็นเมทริกซ์ขนาด $N \times N$ โดย N คือ จำนวนมวลในระบบ

จากนั้น กำหนดให้ $y = M^{\frac{1}{2}}\vec{x}(t)$ แทนค่าในสมการ 2.18 ได้ว่า

$$MM^{-\frac{1}{2}}\ddot{\vec{y}}(t) = -FM^{-\frac{1}{2}}\vec{y}(t) \tag{2.19}$$

$$\ddot{\vec{y}}(t) = -M^{-\frac{1}{2}}FM^{-\frac{1}{2}}\vec{y}(t) = -A\vec{y}(t) \tag{2.20}$$

เมื่อ

$$A = M^{-\frac{1}{2}}FM^{-\frac{1}{2}} \tag{2.21}$$

เมื่อบวก $iA^{\frac{1}{2}}\dot{\vec{y}}(t)$ ทั้ง 2 ข้างของสมการจะได้ว่า

$$\ddot{\vec{y}}(t) + iA^{\frac{1}{2}}\dot{\vec{y}}(t) = -A\vec{y}(t) + iA^{\frac{1}{2}}\dot{\vec{y}}(t) = iA^{\frac{1}{2}}(\dot{\vec{y}}(t) + iA^{\frac{1}{2}}\vec{y}(t)) \tag{2.22}$$

เนื่องจาก A เป็นเมทริกซ์จตุรัส (square matrix) จึงสามารถแก้สมการเชิงอนุพันธ์ได้ โดยให้ผลเฉลยที่แสดงความสัมพันธ์ของการกระจัดและความเร็ว ณ ช่วงเริ่มต้นกับช่วงเวลา t ใดๆ ดังนี้

$$\dot{\vec{y}}(t) + iA^{\frac{1}{2}}\vec{y}(t) = e^{itA^{\frac{1}{2}}}(\dot{\vec{y}}(0) + iA^{\frac{1}{2}}\vec{y}(0)) \tag{2.23}$$

สังเกตได้ว่า สมการ 2.22 และ 2.23 มีความคล้ายกับ สมการชเรอดิงเงอร์ที่วิวัฒน์ไปตามเวลา โดยในระบบหน่วยธรรมชาติ (natural unit system) กล่าวคือ ให้ $\hbar = 1$ ซึ่งจะทำให้สมการดังกล่าวเขียนได้ดังนี้

$$\left|\dot{\psi}(t)\right\rangle = -iH\left|\psi(t)\right\rangle \tag{2.24}$$

$$\left|\psi(t)\right\rangle = e^{-iHt}\left|\psi(0)\right\rangle \tag{2.25}$$

กล่าวคือ เราสามารถแปลงสมการ 2.22 ให้อยู่ในรูป 2.24 ได้ โดยกำหนดให้
$H = -\begin{bmatrix} 0 & A^{\frac{1}{2}} \\ A^{\frac{1}{2}} & 0 \end{bmatrix}$ จะได้

$$\begin{pmatrix} \ddot{\vec{y}}(t) \\ iA^{\frac{1}{2}}\dot{\vec{y}}(t) \end{pmatrix} = i\begin{bmatrix} 0 & A^{\frac{1}{2}} \\ A^{\frac{1}{2}} & 0 \end{bmatrix}\begin{pmatrix} \dot{\vec{y}}(t) \\ iA^{\frac{1}{2}}\vec{y}(t) \end{pmatrix} = \begin{pmatrix} -A\vec{y}(t) \\ iA^{\frac{1}{2}}\dot{\vec{y}}(t) \end{pmatrix} \tag{2.26}$$

และมีผลเฉลยเป็น

$$\begin{pmatrix} \dot{\vec{y}}(t) \\ iA^{\frac{1}{2}}\vec{y}(t) \end{pmatrix} = e^{it\begin{bmatrix} 0 & A^{\frac{1}{2}} \\ A^{\frac{1}{2}} & 0 \end{bmatrix}}\begin{pmatrix} \dot{\vec{y}}(0) \\ iA^{\frac{1}{2}}\vec{y}(0) \end{pmatrix} \tag{2.27}$$

นอกจากนี้ เมื่อพิจารณา F ในสมการ 2.18 จะเห็นว่า F เป็นเมทริกซ์ที่แสดงถึงการเชื่อมต่อของมวลในระบบที่มีลักษณะเป็นกราฟ โดยมีค่านิจสปริงเป็นน้ำหนักของเส้นเชื่อม เราเรียกเมทริกซ์นี้ว่าเป็น เมทริกซ์ลาปลาซ (Laplacian matrix) หรือลาปลาเซียนแบบกราฟ (Graph laplacian) โดยเราสามารถแยกเมทริกซ์ลาปลาซให้แสดงเมทริกซ์ทแยงมุมของน้ำหนักเส้นเชื่อมออกมาได้ ดังนั้น F แยกออกมาเป็น

$$F = \Phi_F W \Phi_F^{\dagger} \tag{2.28}$$

โดย W แทนเมทริกซ์ทแยงมุมของค่านิจสปริง และ Φ_F แทนเมทริกซ์อุบัติการณ์ (incidence matrix) อีกทั้ง เมทริกซ์ลาปลาซถูกจัดว่าเป็นเมทริกซ์กึ่งบวกแน่นอน (positive semi-definite matrix) $\mathcal{M}$ และเมทริกซ์นี้มีคุณสมบัติในการแยกค่ารากที่สองออกมาได้โดย $\mathcal{M} = \Omega^\dagger\Omega$ ซึ่งสามารถเขียน Ω ในรูปของ $\mathcal{M}^{\frac{1}{2}}$ ได้ จากเหตุผลข้างต้นรวมกับสมการ 2.21 จึงกล่าวได้ว่าทั้ง F และ A เป็นเมทริกซ์กึ่งบวกแน่นอน และทำให้ A ถูกแยกออกมาได้ว่า

$$A = BB^\dagger \tag{2.29}$$

โดยทั้ง $B, B^\dagger$ ต่างเป็นคำตอบจากการแยกรากที่สองของ A (มองเทียบได้ว่า $\Omega^\dagger = B$) ซึ่งหาค่าได้โดยรวมสมการ 2.21, 2.28 และ 2.29 เข้าด้วยกัน จึงออกมาเป็น

$$\begin{aligned} BB^\dagger &= M^{-\frac{1}{2}}\Phi_F W \Phi_F^\dagger M^{-\frac{1}{2}} & (2.30)\\ &= M^{-\frac{1}{2}}\Phi_F W^{\frac{1}{2}} W^{\frac{1}{2}} \Phi_F^\dagger M^{-\frac{1}{2}} & (2.31)\\ &= M^{-\frac{1}{2}}\Phi_F W^{\frac{1}{2}} (M^{-\frac{1}{2}}\Phi_F W^{\frac{1}{2}})^\dagger & (2.32) \end{aligned}$$

$$B = M^{-\frac{1}{2}}\Phi_F W^{\frac{1}{2}} \tag{2.33}$$

$$B^\dagger = W^{\frac{1}{2}}\Phi_F^\dagger M^{-\frac{1}{2}} \tag{2.34}$$

ทำให้เราสามารถนำ $B, B^\dagger$ แทนที่ $A^{\frac{1}{2}}$ ในสมการ 2.26 และ 2.27 ได้ โดยหากกำหนดให้เป็น $H = -\begin{bmatrix} 0 & B \\ B^\dagger & 0 \end{bmatrix}$ สมการ 2.26 จะกลายเป็น

$$\begin{pmatrix} \ddot{\vec{y}}(t) \\ iB^\dagger\dot{\vec{y}}(t) \end{pmatrix} = i\begin{bmatrix} 0 & B \\ B^\dagger & 0 \end{bmatrix}\begin{pmatrix} \dot{\vec{y}}(t) \\ iB^\dagger\vec{y}(t) \end{pmatrix} = \begin{pmatrix} -A\vec{y}(t) \\ iB^\dagger\dot{\vec{y}}(t) \end{pmatrix} \tag{2.35}$$

และสมการ 2.27 จะกลายเป็น

$$|\psi(t)\rangle = \begin{pmatrix} \dot{\vec{y}}(t) \\ iB^\dagger\vec{y}(t) \end{pmatrix} = e^{it\begin{bmatrix} 0 & B \\ B^\dagger & 0 \end{bmatrix}}\begin{pmatrix} \dot{\vec{y}}(0) \\ iB^\dagger\vec{y}(0) \end{pmatrix} \tag{2.36}$$

ซึ่งเมื่อแทนค่า $\dot{\vec{y}}, \vec{y}$ และ B ใน 2.36 จะเป็น

$$|\psi(t)\rangle = \begin{pmatrix} \dot{\vec{y}}(t) \\ iB^\dagger\vec{y}(t) \end{pmatrix} = \begin{pmatrix} M^{\frac{1}{2}}\dot{\vec{x}}(t) \\ iW^{\frac{1}{2}}\Phi^\dagger\vec{x}(t) \end{pmatrix} = \begin{pmatrix} \sqrt{m_1}\dot{\vec{x}}_1(t) \\ \vdots \\ \sqrt{m_N}\dot{\vec{x}}_N(t) \\ i\sqrt{k_1}(x_1(t)-x_2(t)) \\ \vdots \\ i\sqrt{k_{N-1}}(x_{N-1}(t)-x_N(t)) \\ i\sqrt{k_{0/N}}(x_N(t)-x_1(t)) \end{pmatrix} \tag{2.37}$$

หากพิจารณารวมกับค่าพลังงานรวมของระบบ E ซึ่งประกอบด้วยพลังงานจลน์และพลังงานศักย์ ดังนี้

$$E = T + V = \left(\frac{1}{2}\sum_{i=1}^{N} m_i\dot{\vec{x}}_i^2\right) + \left(\frac{1}{2}\sum_{j=2}^{N} k_{j-1}(x_{j-1}(t)-x_j(t))^2\right) \tag{2.38}$$

ดังนั้น เราสามารถนอร์มอลไลซ์เวกเตอร์สถานะ $|\psi(t)\rangle$ ได้ว่า

$$|\psi(t)\rangle = \frac{1}{\sqrt{2E}}\begin{pmatrix} \dot{\vec{y}}(t) \\ iB^\dagger\vec{y}(t) \end{pmatrix} \tag{2.39}$$

เราสามารถนำสมการ 2.36 และ 2.39 ไปสร้างวงจรควอนตัมได้ โดยรายละเอียดจะกล่าวถึงในบทที่ 4

บทที่ 3

วิธีดำเนินการวิจัย

3.1 แนวคิดและวิธีการดำเนินงาน

งานวิจัยฉบับนี้มีเป้าหมายเพื่อหาแนวทางการจำลองการสั่นคู่ควบของระบบมวลติดสปริงบนวงจรควอนตัมโดยใช้เทคนิคบล็อกเอ็นโค้ดดิงที่อาจมีประสิทธิภาพที่ดีกว่าการจำลองระบบการสั่นควบคู่บนคอมพิวเตอร์คลาสสิก โดยอาศัยการฝังเมทริกซ์ลงในตัวดำเนินการควอนตัมและการใช้อัลกอริทึม QSVT การสร้างวงจรจะใช้ไลบรารี Qiskit ซึ่งเป็นเฟรมเวิร์กบน python ของ IBM ควบคู่กับโปรแกรม QSPPACK (Dong et al., 2021) ซึ่งเป็นโปรแกรมที่ทำงานบน Matlab วงจรนี้ประกอบด้วย 2 ส่วน คือ วงจรบล็อกเอ็นโค้ดดิงของตัวดำเนินการ e^{-iHt} และการเตรียมสถานะเริ่มต้นตามสมการ 2.39 แต่เนื่องจากวงจรเพื่อสร้างสถานะเริ่มต้นนั้นทำได้ยากและไม่อยู่ในขอบเขตของงานวิจัยนี้ งานนี้จึงอาศัยฟังก์ชันภายในไลบรารี Qiskit เพื่อสร้างออราเคิลของสถานะเริ่มต้นแทน โดยหัวใจสำคัญของงานนี้คือ การสร้างวงจรบล็อกเอ็นโค้ดดิงของเมทริกซ์ $B, B^\dagger$ ซึ่งฝังอยู่ในเมทริกซ์ H อีกทีตามสมการ 2.36 เนื่องจากรูปแบบของ $B, B^\dagger$ ขึ้นกับค่ามวล, ค่านิจสปริง, จำนวนมวล, และลักษณะการเชื่อมต่อของสปริง (แบบเส้นตรงกับแบบวงกลม) อีกทั้งจำนวนคิวบิตหลักที่ใช้มีค่าตามลอการิทึม (logarithm) ของจำนวนมวลที่ถูกปัดเศษขึ้น ทำให้เมทริกซ์บล็อกเอ็นโค้ดมีแถวและหลักที่มีค่าศูนย์จำนวนมากตามนิยามบล็อกเอ็นโค้ดดิงที่ได้กล่าวไป การสร้างวงจรควอนตัมจึงมีความซับซ้อนตามไปด้วย

ในบทที่ 4 นี้จะเสนออัลกอริทึมเพื่อสร้างวงจรบล็อกเอ็นโค้ดดิงของเมทริกซ์ B รวมถึงอธิบายหลักการทำงานประกอบว่าเกตที่ใส่ในวงจรส่งผลต่อหน้าตาของเมทริกซ์บล็อกเอ็นโค้ดอย่างไร รวมถึงการนำบล็อกเอ็นโค้ดดิงของ B ไปประกอบเพื่อสร้างวงจรบล็อกเอ็นโค้ดดิงของ H และ e^{-iHt} ตามลำดับ ก่อนนำวงจรควอนตัมที่ได้ไปรันจำลองระบบ โดยป้อนค่าเงื่อนไขเริ่มต้นและชุดค่าเฟสที่ได้จาก QSPPACK ผลจากการรันวงจรของ e^{-iHt} คู่กับออราเคิลของสถานะเริ่มต้นจะออกมาเป็นสถานะของระบบมวลติดสปริง ณ เวลา t ซึ่งเราสามารถนำไปใช้หาความเร็วและตำแหน่งของมวลทุกก้อนได้ ผลลัพธ์จากการจำลองจะแสดงในบทที่ 5

หมายเหตุ งานวิจัยฉบับนี้ทำร่วมกับ นายพีระณัฐ แสงละออ นิสิตภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย โดยส่วนที่นายพีระณัฐเป็นผู้ทดลองจะเขียนชื่อกำกับไว้

3.2 ขั้นตอนการดำเนินงาน

1. ศึกษาเนื้อหาทั่วไปของการจำลองควอนตัม
2. ศึกษาหลักการของบล็อกเอ็นโค้ดดิง
3. ศึกษาลักษณะการต่อวงจรเพื่อให้ได้มาซึ่งบล็อกเอ็นโค้ดดิงแบบต่าง ๆ
4. ศึกษาวงจรและการทำงานของ QSVT
5. ทำการสร้างวงจรบล็อกเอ็นโค้ดดิงย่อยตามสมการและตรวจสอบลักษณะเมทริกซ์ยูนิแทรี
6. ทำการรวมวงจรควอนตัม
7. ทดสอบรันวงจรจำลองโดยกำหนดจำนวนมวลและสถานะเริ่มต้นแบบต่าง ๆ
8. ทดสอบวงจรบล็อกเอ็นโค้ดดิงกับผลลัพธ์ของวงจรที่สร้างจากเฟรมเวิร์คอื่น
9. สรุปผลการทดลอง

บทที่ 4

การออกแบบวงจร

4.1 การสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ B

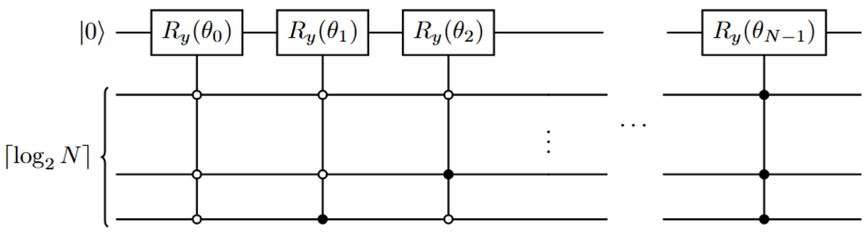
จากสมการของ B ประกอบไปด้วย 3 ส่วน คือ เมทริกซ์ของมวล $M^{-\frac{1}{2}}$, เมทริกซ์ของค่านิจสปริง $W^{\frac{1}{2}}$ และเมทริกซ์อุบัติการณ์ Φ_F ซึ่งเมทริกซ์ทั้ง 3 มีขนาด $N \times N$ เมื่อ N คือ จำนวนมวล หากแปลงเป็นตัวดำเนินการจะใช้ $n = \lceil \log_2 N \rceil$ คิวบิต (n-qubit operator)

พิจารณา $M^{-\frac{1}{2}}, W^{\frac{1}{2}}$ **:** เนื่องจากทั้งสองเป็นเมทริกซ์ทแยงมุม จึงสามารถสร้างบล็อกเอ็นโค้ดดิงและนิยามตามบทตั้ง 1 และเนื่องจากค่ามวลและค่านิจสปริงย่อมเป็นจำนวนจริงบวกเสมอ พจน์ R_z จึงสามารถตัดทิ้งได้ หากกำหนดให้ค่ามวล $m_i \in \{m_0, m_1, ..., m_{N-1}\}$ และค่านิจสปริง $w_i \in \{w_0, w_1, ..., w_{N-1}\}$ ค่ามุมที่ถูกป้อนให้กับ R_y จะมีค่าเท่ากับ $\theta_k^m = 2\cos^{-1}(\frac{1}{\sqrt{m}})$ และ $\theta_k^w = 2\cos^{-1}(\sqrt{w})$ (กำหนดให้ $d_{\max} = 1$) เนื่องจาก θ_k^m และ θ_k^w มีค่าเรนจ์ (range) ในช่วง $[0, \pi]$ ดังนั้น การจำลองจึงมีเงื่อนไขว่า $m \geq 1$ และ $0 \leq w \leq 1$ ดังนั้น ค่าภายในบล็อกของเมทริกซ์บล็อกเอ็นโค้ด $M^{-\frac{1}{2}}$ และ $W^{\frac{1}{2}}$ จะมีลักษณะดังนี้

$$U_M = \begin{bmatrix} \frac{1}{\sqrt{m_0}} & & & & & & \\ & \frac{1}{\sqrt{m_1}} & & & & & \\ & & \ddots & & & & \\ & & & \frac{1}{\sqrt{m_{N-1}}} & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix} \tag{4.1}$$

$$U_W = \begin{bmatrix} \sqrt{w_0} & & & & & & \\ & \sqrt{w_1} & & & & & \\ & & \ddots & & & & \\ & & & \sqrt{w_{N-1}} & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix} \tag{4.2}$$

และวงจรของ $M^{-\frac{1}{2}}$ และ $W^{\frac{1}{2}}$ จะมีลักษณะรูป 4.1



รูปที่ 4.1: วงจร (1,1,0)-block-encoding ของเมทริกซ์ $M^{-\frac{1}{2}}$ และ $W^{\frac{1}{2}}$

พิจารณา Φ_F **:** Φ_F เป็นเมทริกซ์อุบัติการณ์ ซึ่งค่าในเมทริกซ์ b_{ij} นิยามว่า

$$b_{ij} = \begin{cases} -1, & \text{เมื่อเส้นเชื่อม } e_j \text{ ชี้เข้าหาจุด } v_i \\ 1, & \text{เมื่อเส้นเชื่อม } e_j \text{ ชี้ออกจากจุด } v_i \\ 0, & \text{กรณีอื่น} \end{cases}$$

หากมองภาพระบบมวลติดสปริงที่เชื่อมเป็นเส้นตรงโดยให้สปริงแทนเส้นเชื่อมที่มีทิศชี้ไปทางขวาจะเขียนเมทริกซ์ขนาด $N \times N$ ได้ว่า

$$\Phi_F^o = \begin{bmatrix} 1 & & & & 0 \\ -1 & 1 & & & \\ & -1 & \ddots & & \\ & & \ddots & 1 & \\ & & & -1 & 0 \end{bmatrix} \tag{4.3}$$

และหากมองภาพระบบมวลติดสปริงที่เชื่อมเป็นวงกลมโดยให้สปริงแทนเส้นเชื่อมที่มีทิศชี้ตามเข็มนาฬิกาจะเขียนเมทริกซ์ $N \times N$ ได้ว่า

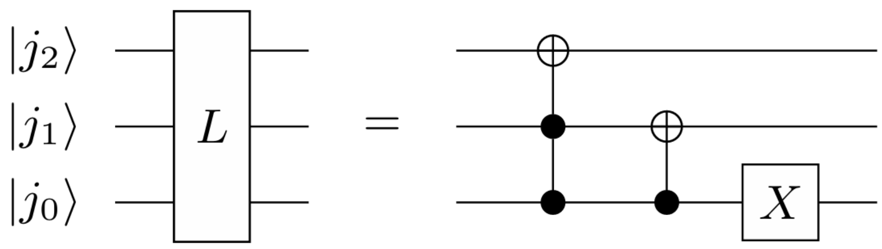
$$\Phi_F^c = \begin{bmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & -1 & \ddots & & \\ & & \ddots & 1 & \\ & & & -1 & 1 \end{bmatrix} \tag{4.4}$$

ซึ่ง Φ_F^o และ Φ_F^c สามารถเขียนแยกออกมาได้ว่า

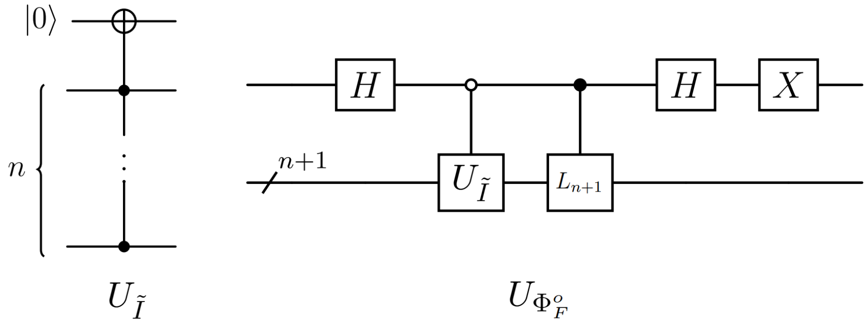
$$\Phi_F^o = \tilde{I} - L = \begin{bmatrix} 1 & & & & 0 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} - \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & 0 & \\ & & & 1 & 0 \end{bmatrix} \tag{4.5}$$

$$\Phi_F^c = I - \tilde{L} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} - \begin{bmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & 0 & \\ & & & 1 & 0 \end{bmatrix} \tag{4.6}$$

พีระณัฐได้เสนอวงจร $(2, 2, 0)$-block-encoding ของ Φ_F^o สำหรับระบบมวลที่ $N = 2^n$ เมื่อ $n \in \mathbb{I}^+$ ดำเนินการบน n คิวบิตและ $(1, 1, 0)$-block-encoding ของ $\tilde{I}$ สามารถสร้างได้โดยใช้ multi-controlled CNOT gate โดยให้ทุกคิวบิตหลักคุมคิวบิตตัวทด และ $(1, 1, 0)$-block-encoding ของ L สามารถสร้างได้โดยใช้วงจรเลื่อนซ้าย (L-shift) ที่เสนอโดย (Camps et al., 2024)ทำบนทั้งคิวบิตหลักและคิวบิตตัวทด แล้วนำมาลบกันโดยใช้ LCU ตามบทตั้ง 2 โดยใส่ X gate เพื่อเปลี่ยนเป็นลบ ภาพวงจร L-shift เป็นดังรูป 4.2 ภาพของวงจรบล็อกเอ็นโค้ดดิงของ $\tilde{I}, \Phi_F^o$ แสดงตามในรูป 4.3 ตามลำดับ

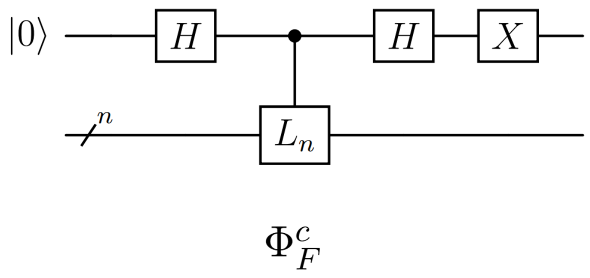


รูปที่ 4.2: วงจร L-shift (Camps et al., 2024)



รูปที่ 4.3: วงจร (1,1,0)-block-encoding ของ $\tilde{I}$ และ (2,2,0)-block-encoding ของ Φ_F^o

และพีระณัฐได้เสนอวงจร $(2,1,0)$-block-encoding ของ Φ_F^c สำหรับระบบมวลที่ $N = 2^n$ เมื่อ $n \in \mathbb{I}^+$ ดำเนินการบน n คิวบิตและ $(1,0,0)$-block-encoding ของ $\tilde{L}$ ใช้ L-shift ทำบน n คิวบิตหลัก แล้วนำมาใส่ในวงจร LCU กับ X gate เพื่อเปลี่ยนเป็นการลบ (วงจรของเมทริกซ์ I สามารถปล่อยว่างได้ เนื่องจากไม่มีการเปลี่ยนแปลงสถานะ) ภาพของวงจรบล็อกเอ็นโค้ดดิงของ Φ_F^c แสดงตามในรูป 4.4



รูปที่ 4.4: วงจร (2,1,0)-block-encoding ของ Φ_F^c

ในกรณีที่ $N \neq 2^n$ ผู้เขียนเล็งเห็นว่าลักษณะของเมทริกซ์ Φ_F^o, Φ_F^c จะเขียนใหม่ได้ว่า

$$\Phi_F^{'o} = \tilde{I}^{'} - L^{'} = \begin{pmatrix} \tilde{I} & 0 \\ 0 & 0 \end{pmatrix}_{2^n \times 2^n} - \begin{pmatrix} L & 0 \\ 0 & 0 \end{pmatrix}_{2^n \times 2^n} \tag{4.7}$$

$$\Phi_F^{'c} = I^{'} - \tilde{L}^{'} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}_{2^n \times 2^n} - \begin{pmatrix} \tilde{L} & 0 \\ 0 & 0 \end{pmatrix}_{2^n \times 2^n} \tag{4.8}$$

ลักษณะข้อสังเกตหลักในการสร้างวงจรบล็อกเอ็นโค้ดดิงนี้คือ

1. ทั้ง $\tilde{I}'$ และ I' ต่างเป็นเมทริกซ์คล้ายเมทริกซ์เอกลักษณ์ แต่มีบางช่วงที่เป็น 0
2. ทั้ง L' และ $\tilde{L}'$ ต่างเป็นเมทริกซ์คล้ายเมทริกซ์เอกลักษณ์ที่ถูกเลื่อนลงมา 1 แถว แต่มีบางช่วงที่เป็น 0
3. $\tilde{L}'$ มีค่า 1 ในตำแหน่งมุมขวาบน ในขณะที่ L' มีค่าเป็น 0
4. วงจร L-shift ข้างต้นประพฤติตัวเหมือนกับการเลื่อนเมทริกซ์ลงมา 1 แถว โดยค่าในแถวสุดท้ายขึ้นไปแทนที่ในตำแหน่งแถวบน

จากข้อสังเกตที่ว่าดังกล่าว ผู้เขียนขอนำเสนออัลกอริทึมสำหรับสร้าง $(1, 1, 0)$-block-encoding ของ $\tilde{I}', I', L', \tilde{L}'$ ดังที่ปรากฏอยู่ใน Algorithm 1

Algorithm 1: Building a block encoding circuit of $\tilde{I}', I', L', \tilde{L}'$

Input: Number of masses N, case = $\{\tilde{I}', I', L', \tilde{L}'\}$
Initiate a new circuit U with the number of qubits $n+1$ which $n = \lceil \log_2 N \rceil$
Set a stop condition depends on a case: $g = 0$ for I', and $g = 1$ for $\tilde{I}', L', \tilde{L}'$
procedure *Add gate* $\Xi_g = \sum_{i=0}^{N-g-1} |i\rangle\langle i| + \sum_{i=N-g}^{2^n-1} |i+2^n\rangle\langle i| + \sum_{i=2^n}^{2^n+N-g-1} |i\rangle\langle i| + \sum_{i=2^n+N-g}^{2^{n+1}-1} |i-2^n\rangle\langle i|$
    Set $h = 2^n$
    Set array $k[]$ for listing qubit index
    for $i = 1, 2, ...$ **to** n **do**
        if $h - 2^{n-i} \geq N - g$ **then**
            add a NOT gate to all qubits listed in array k if k is not empty
            add a multi-controlled NOT gate (MCX) with control qubits on all $a \in \{1, ..., i\}$ and target qubit is the 0-th qubit
            add a NOT gate to all qubits listed in array k if k is not empty
            add i in array k
            $h \leftarrow h - 2^{n-i}$
        end
    end
if *case is* $\tilde{L}'$ **then**
    procedure *Add gate* $M_c = \sum_{i=0}^{2^n+N-g-1} |i\rangle\langle i| + |2^{n+1}-1\rangle\langle 2^n+N-g| + |\perp\rangle\langle\perp|$
        $b_g \leftarrow$ convert $2^n + N - g$ to binary with length $n+1$
        for $i = 1, 2, ...$ **to** n **do** ▷ `Loop through binary string`
            if $b_g[i] =$ '0' **then**
                add a CNOT gate with control qubit and target qubit on 0-th and i-th qubits, respectively.
            end
        end
    add gate L-shift L_{n+1}
end
if *case is* L' **then**
    procedure *Add gate* $M_o = \sum_{i=0}^{2^n-1} |i\rangle\langle i| + |2^{n+1}-1\rangle\langle 2^n+m| + |\perp\rangle\langle\perp|$ *with any* m *in* $\{0, 1, ..., 2^n-1\}$ *such as CNOT (control on 0; target on 1) iff* $N - g \geq 2^{n-1}$
    add a L-shift gate L_{n+1}
end
Output: circuit U

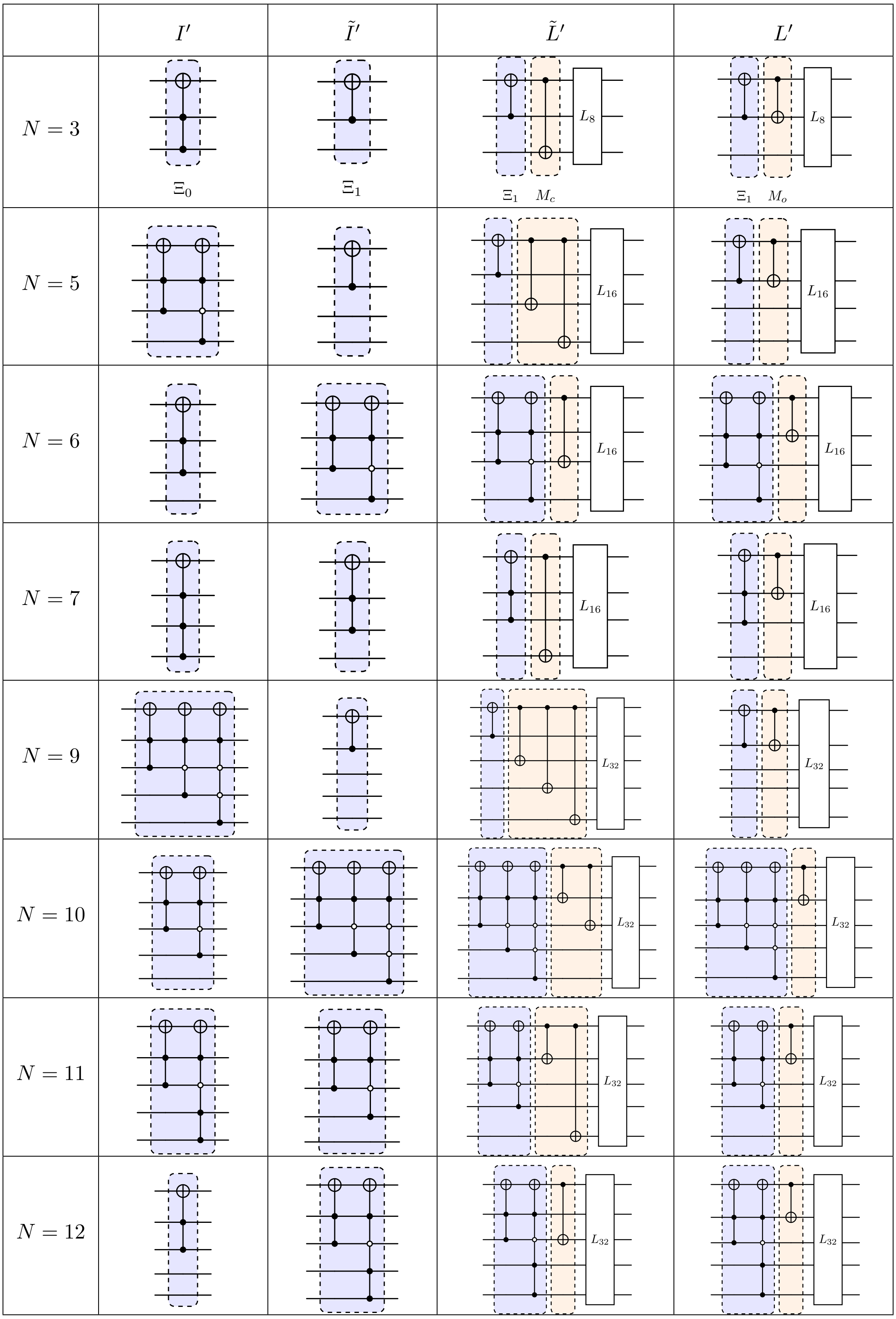
พิจารณา Ξ_g **:** การสร้างวงจรดังกล่าวเปรียบเสมือนการสลับแถวเพื่อเอาค่าออกนอกบล็อกการคำนวณ โดยหากเริ่มต้นพิจารณา CNOT gate(control:1, target:0) สำหรับคิวบิตหลัก n ตัว จะทำให้บล็อกเอ็นโค้ดดิงมีค่าเป็น $\sum_{i=0}^{2^{n-1}-1} |i\rangle\langle i| + \sum_{i=2^{n-1}}^{2^n-1} |i+2^n\rangle\langle i| + |\perp\rangle\langle\perp|$ กล่าวคือ ค่า 1 ในแนวทแยงมุมถูกสลับไปครึ่งนึงของบล็อก หากเพิ่มตัว control เป็น CNOT gate(control:1,2 , target:0) จะทำให้บล็อกเอ็นโค้ดดิงมีค่าเป็น $\sum_{i=0}^{2^{n-1}+2^{n-2}-1} |i\rangle\langle i| + \sum_{i=2^n+2^{n-2}}^{2^n-1} |i+2^n\rangle\langle i| + |\perp\rangle\langle\perp|$ กล่าวคือ ค่า 1 ในแนวทแยงมุมถูกสลับไป $\frac{1}{4}$ ของบล็อก เมื่อเพิ่ม control ไปเรื่อย ๆ จนเป็น CNOT gate(control:1,2,...,N , target:0) จะทำให้บล็อกเอ็นโค้ดดิงมีค่าเป็น $\sum_{i=0}^{2^{n-1}+2^{n-2}+\ldots+2^0-1} |i\rangle\langle i| + \sum_{i=2^n+2^{n-2}+\ldots+2^0}^{2^n-1} |i+2^n\rangle\langle i| + |\perp\rangle\langle\perp| = \sum_{i=0}^{2^n-1-1} |i\rangle\langle i| + |2^{n+1}-1\rangle\langle 2^n-1| + |\perp\rangle\langle\perp|$ กล่าวคือ ค่า 1 ในแนวทแยงมุมถูกสลับไป $\frac{1}{2^n}$ ของบล็อก ซึ่งมีแค่ตัวเดียว ดังนั้น หากทำการเลือกคิวบิตสำหรับควบคุมอย่างเหมาะสม (อาศัย X gate เพื่อเลือกช่วงสถานะ) ก็จะสามารถสลับ 1 ออกจากบล็อกได้โดยใช้จำนวนเกต $O(n)$

พิจารณา M_c **:** การสร้างวงจรดังกล่าวเปรียบเสมือนการสลับที่ 1 ที่อยู่ในตำแหน่ง $|2^n+N\rangle\langle N|$ ให้ไปอยู่ในตำแหน่ง $|2^{n+1}-1\rangle\langle N|$ โดยเราสามารถใส่ CNOT โดยให้คิวบิตตัวทดเป็นตัวควบคุมได้โดยตรง เพราะค่าอื่นจะสลับตำแหน่งในบริเวณนอกบล็อก และสุดท้ายค่าในตำแหน่ง $|2^n+N\rangle\langle N|$ จะถูกเลื่อนลงและกลับไปโผล่อยู่ ณ ตำแหน่ง $|0\rangle\langle N|$ ด้วย L-shift L_{n+1} นั่นเอง การสร้างวงจรในขั้นตอนนี้ใช้จำนวนเกต $O(n)$

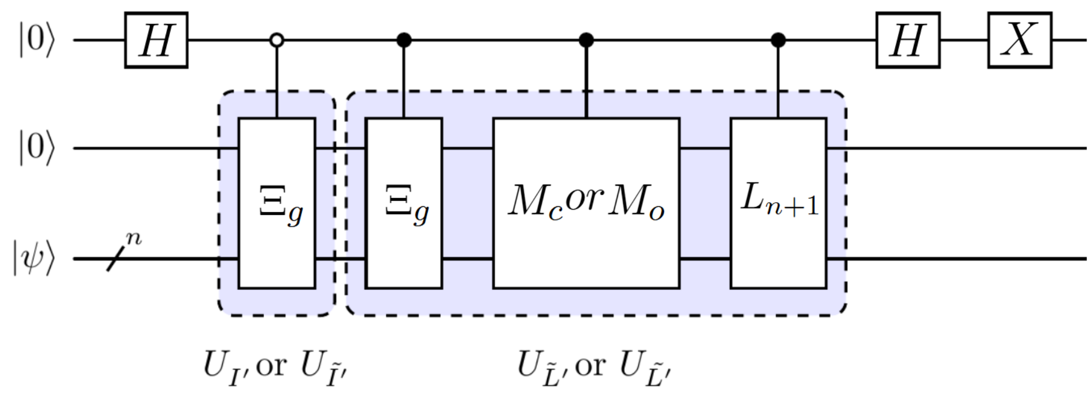
พิจารณา M_o **:** การใส่เกตดังกล่าวเพื่อเป็นการย้ายค่า 1 ไปอยู่ในบริเวณล่างขวาของเมทริกซ์บล็อกเอ็นโค้ด เพื่อไม่ให้มีค่าโผล่ในบล็อกเมื่อใช้ L-shift L_{n+1} การสร้างวงจรในขั้นตอนนี้ใช้จำนวนเกต $O(1)$

ตัวอย่างวงจรที่ได้จาก Algorithm 1 เป็นดังในรูปที่ 4.5

เมื่อนำไปรวมกับวงจร LCU พร้อมกับใส่ X gate เพื่อเปลี่ยนเป็นลบ วงจรรวมทั้งหมดจะออกมาดังรูปที่ 4.6

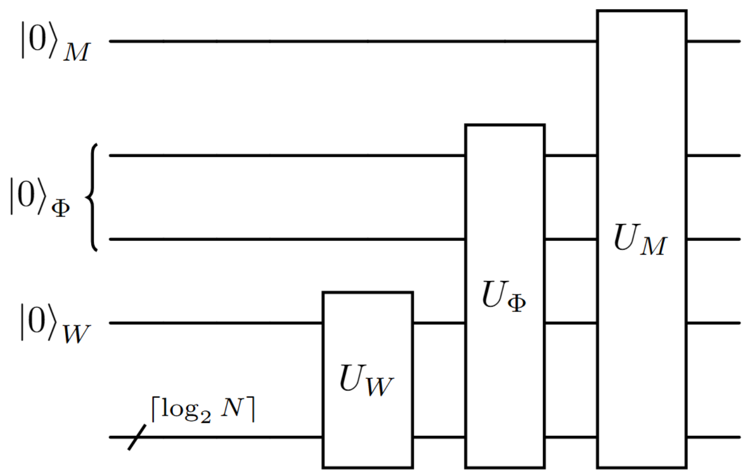


รูปที่ 4.5: ตัวอย่างผลลัพธ์วงจรบล็อกเอ็นโค้ดดิงของ B กรณี $N \neq 2^n$



รูปที่ 4.6: วงจร (2,2,0)-block-encoding ของ $\Phi'^{o}_{F}, \Phi'^{c}_{F}$

สุดท้ายจึงนำบล็อกเอ็นโค้ดดิงของ $M^{\frac{1}{2}}, \Phi_F, W^{\frac{1}{2}}$ มารวมกันด้วยบล็อกเอ็นโค้ดดิงการคูณตามบทตั้ง 3 ทำให้ได้วงจรดังรูปที่ 4.7 ทั้งนี้ เราได้แทรก 1 คิวบิตตัวทดเพื่อใช้ในการทำผลคูณเทนเซอร์ในส่วนถัดไป



รูปที่ 4.7: วงจร (2,4,0)-block-encoding ของ ของ B

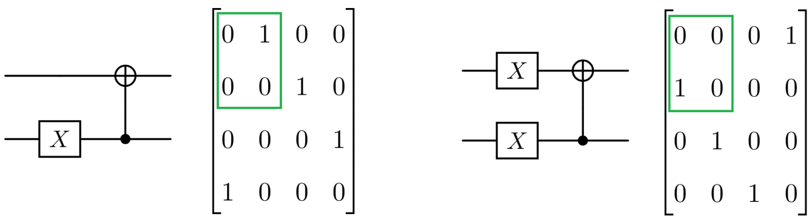
ส่วนการสร้างบล็อกเอ็นโค้ดดิงของ $B^{\dagger}$ นั้นสามารถทำได้โดยการกลับด้านวงจร B ทั้งหมด ซึ่งสาเหตุที่ทำได้เพราะเกตที่ใช้ในบล็อกเอ็นโค้ดดิงมีคุณสมบัติเป็นเมทริกซ์เฮอร์มิเทียน ซึ่งการสร้างวงจรดังกล่าวสามารถเรียกใช้คำสั่งของ Qiskit เพื่อทำการกลับวงจรได้

4.2 วงจรบล็อกเอ็นโค้ดดิงของ H

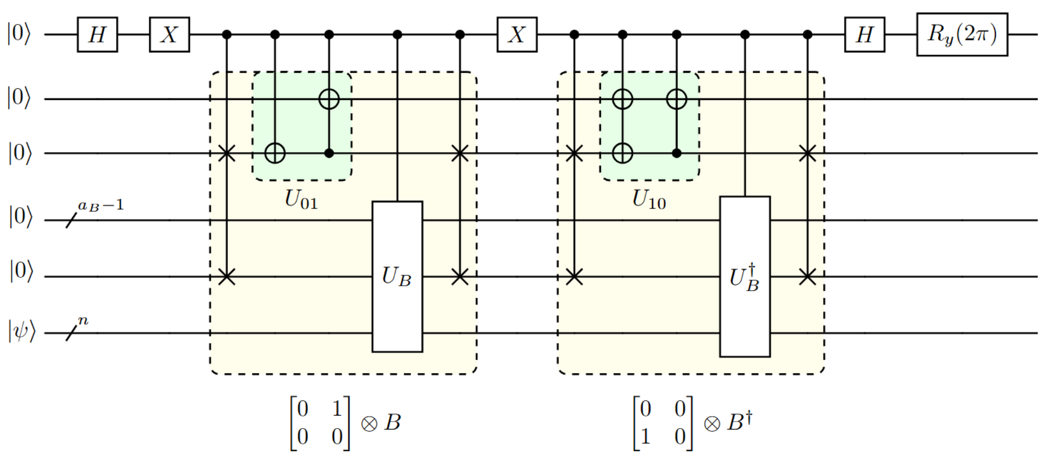
จากที่ (Babbush et al., 2023) ได้ให้ $H = -\begin{bmatrix} 0 & B \\ B^\dagger & 0 \end{bmatrix}$ ซึ่งสามารถแยกออกมาได้ว่า

$$H = -\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes B + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \otimes B^\dagger\right) \tag{4.9}$$

ซึ่งบล็อกเอ็นโค้ดดิงของ $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ กับ $\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ สามารถสร้างได้โดยใช้ X gate และ CNOT gate ตามในรูปที่ 4.8 และนำมารวมกับวงจรบล็อกเอ็นโค้ดดิงของ$B, B^\dagger$ ด้วยการคูณเทนเซอร์ตามบทตั้ง 4 ตามลำดับ จากนั้นจึงรวมทั้งสองเข้าด้วยวงจร LCU และใส่ rotation Y gate ด้วยมุม 2π ต่อท้าย เพื่อเป็นการคูณ -1 (การใส่เกตนี้ พีระณัฐเป็นคนคิดขึ้น) ทำให้ผลลัพธ์ของวงจรเป็นดังรูป 4.9



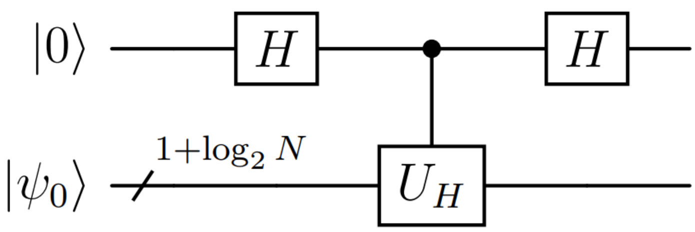
รูปที่ 4.8: บล็อกเอ็นโค้ดดิงจาก X gate และ CNOT gate



รูปที่ 4.9: วงจรบล็อกเอ็นโค้ดดิงของ H

4.3 การสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ e^{-iHt}

เนื่องจากค่า α ของวงจรบล็อกเอ็นโค้ดดิงนี้เป็น 4 (เกิดจากการรวม α จากส่วน LCU ภายนอกและการปรับสเกลผลจาก QSVT) ดังนั้นผลลัพธ์ที่ได้จากบล็อกเอ็นโค้ดดิงส่วนนี้เป็น $0.25e^{-iHt}$ ซึ่งจะหาได้จากการกำหนด $\tilde{H}$ เมื่อ $\tilde{H} = \frac{1}{2}(\frac{H}{\alpha_H} + I)$ โดยบล็อกเอ็นโค้ดดิงของ $\tilde{H}$ เป็นดังรูป 4.10



รูปที่ 4.10: วงจรบล็อกเอ็นโค้ดดิงของ $\tilde{H}$

และสามารถแยกพจน์ค่าชี้กำลังของเมทริกซ์ $\tilde{H}$ ให้อยู่ในรูปสมการของออยเลอร์ โดยเขียนได้ว่า

$$e^{-i\tilde{H}\tau} = \cos\left(\tilde{H}\tau\right) - i\sin\left(\tilde{H}\tau\right) \tag{4.10}$$

โดยที่ $0.25e^{-i\tilde{H}\tau}$ มีค่าเท่ากับ

$$0.25e^{-i\tilde{H}\tau} = 0.25e^{-\frac{iI\tau}{2}}e^{-\frac{iH\tau}{2\alpha_H}} \tag{4.11}$$

$$= 0.25e^{-\frac{i\tau}{2}}e^{-\frac{iH\tau}{2\alpha_H}} \tag{4.12}$$

หากกำหนดให้ $\tau = 2\alpha_H t$ ซึ่ง t เป็นค่าเวลาที่ต้องการจำลองระบบ จะได้ว่า

$$0.25e^{-i\tilde{H}\tau} = 0.25e^{-\frac{i\tau}{2}}e^{-iHt} \tag{4.13}$$

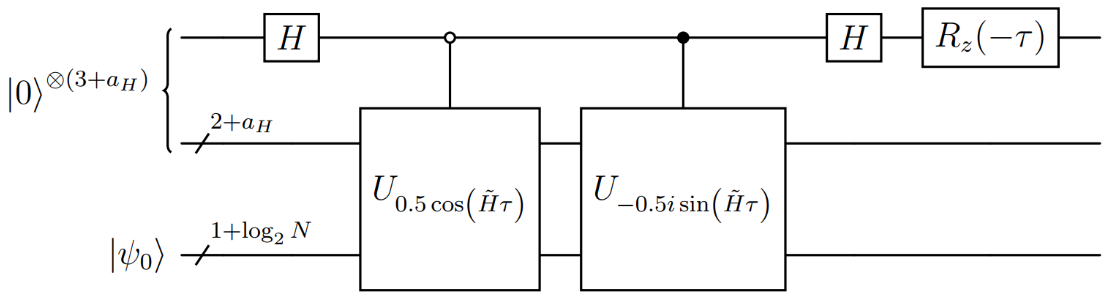
และสามารถกำจัดพจน์ $e^{-\frac{i\tau}{2}}$ โดยใส่ rotation Z gate ที่ป้อนค่า $-\tau$ ทำให้

$$R_z(-\tau)0.25e^{-i\tilde{H}\tau} = 0.25e^{-\frac{i\tau}{2}}e^{-iHt}e^{\frac{i\tau}{2}} \tag{4.14}$$

$$= 0.25e^{-iHt} \tag{4.15}$$

จากขั้นตอนตามสมการกล่าวได้ว่า เราสามารถสร้าง e^{-iHt} ได้โดยการสร้าง $\tilde{H}$ ซึ่งได้จากการทำ LCU แล้วนำมาหาฟังก์ชันไซน์และโคไซน์โดยใช้วงจร QSVT ดังรูป 2.8 โดยป้อนเฟส

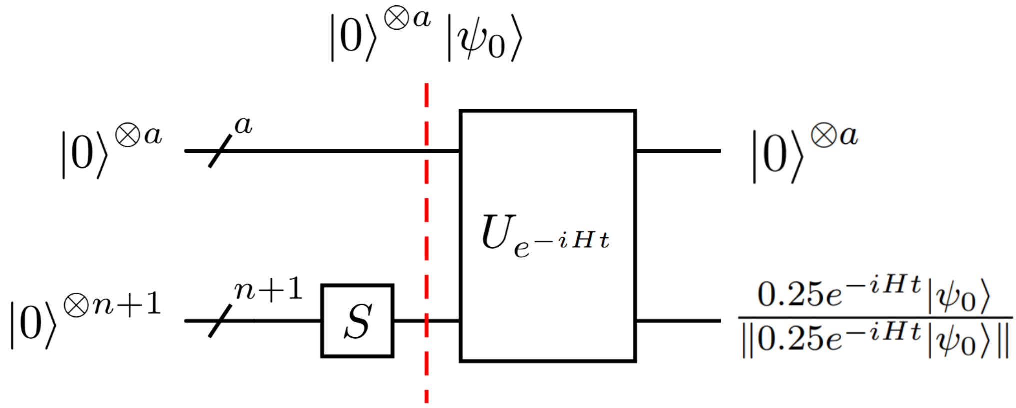
ที่ได้จาก QSPPACK ลงไป จากนั้นจึงนำมารวมกันด้วยวงจร LCU พร้อมใส่ R_z gate ไปด้วย ทำให้ภาพรวมของการสร้างวงจร e^{-iHt} ปรากฏดังในรูปที่ 4.11 ทั้งนี้มีการคูณค่า 0.5 ไปข้างหน้า เพราะเป็นการปรับสเกลจากผลลัพธ์จากการทำฟังก์ชันโคไซน์และไซน์ (ในส่วนนี้พีระณัฐเป็นผู้ทำทั้งหมด)



รูปที่ 4.11: วงจรบล็อกเอ็นโค้ดดิงของ e^{-iHt}

4.4 วงจรทั้งหมดสำหรับการจำลอง

ท้ายที่สุด จึงทำการรวมบล็อกเอ็นโค้ดดิงกับออราเคิลสำหรับการเตรียมสถานะ ให้ผลเป็นดังรูป 4.12 โดยที่ค่า 0.25 ในสถานะคิวบิตหลักจะตัดกัน ทำให้เหลือแต่ผลลัพธ์ของ $e^{-iHt}|\psi_0\rangle$



รูปที่ 4.12: วงจรทั้งหมดสำหรับการจำลอง

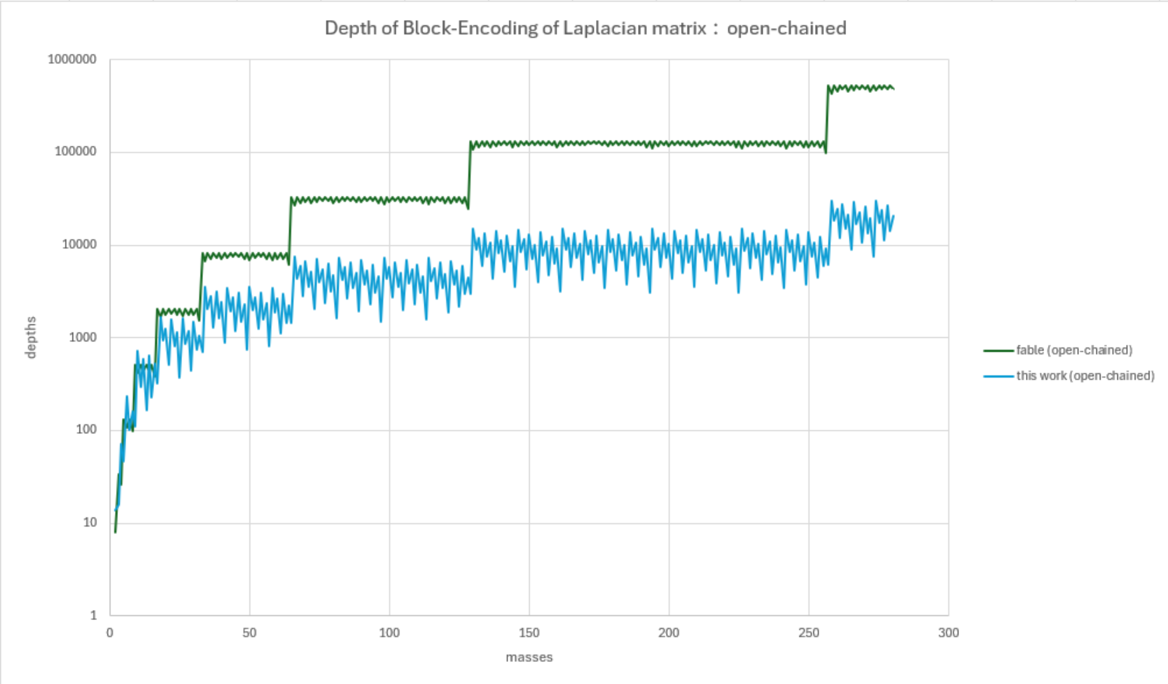
บทที่ 5

ผลการวิจัย

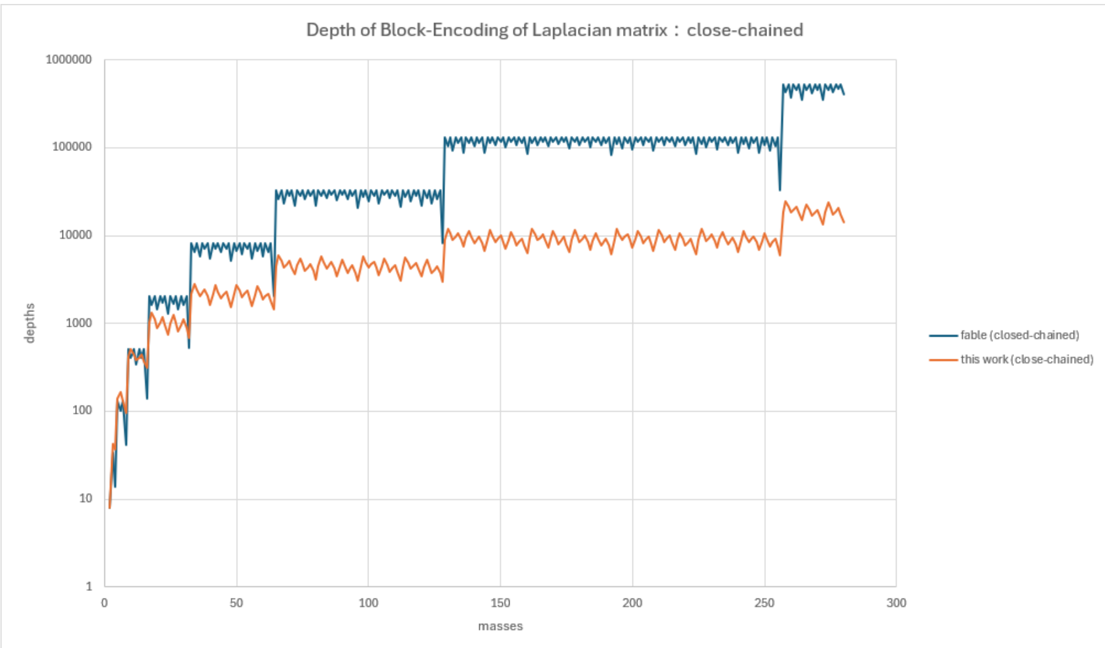
เนื้อหาในบทนี้จะแสดงถึงผลการทดลองจากการใช้อัลกอริทึมที่สร้างบล็อกเอ็นโค้ดดิง B โดยประกอบสองส่วนหลัก คือ เปรียบเทียบความลึก (depth) ของวงจรที่ได้กับวงจรที่ได้มาจากเฟรมเวิร์ค FABLE และผลการจำลองการสั่นคู่ควบจากวงจรเปรียบเทียบกับการคำนวณเมทริกซ์ปกติ และเนื่องจากข้อจำกัดทางทรัพยากรด้านหน่วยความจำของการประมวลผลและข้อจำกัดทางชุดคำสั่ง จึงไม่สามารถจำลองระบบมวลที่ใช้เกิน 13 คิวบิตได้ หรือกล่าวคือได้เฉพาะกรณีจำนวนมวลอยู่ในช่วง $N \leq 8$

5.1 เปรียบเทียบความลึกวงจรเทียบกับวงจร FABLE

FABLE (Camps and Beeumen, 2022) เป็นเฟรมเวิร์คในการสร้างบล็อกเอ็นโค้ดดิงสำเร็จรูป พัฒนาขึ้นบนไพทอนโดยอาศัย multi-controlled rotation gate ในการปรับสถานะคิวบิต FABLE มีจุดโดดเด่นที่ถูกออกแบบมาเพื่อสร้างเมทริกซ์หนาแน่นและสามารถใช้งานได้ดีในการบีบอัดเมทริกซ์ลงในบล็อก อย่างไรก็ตาม ใน (Camps et al., 2024) ได้ตั้งข้อสังเกตว่า FABLE อาจใช้จำนวนเกตถึง $O(N)$ ในบางระบบปัญหา ทั้งนี้ ความลึกของวงจร (มองว่าเป็นความยาววงจรก็ได้) สามารถใช้บอกคุณภาพของวงจรได้ หากวงจรมีความลึกมาก จะใช้ทรัพยากรในการจำลองมากและมีโอกาสเกิดข้อผิดพลาดได้มากขึ้นหากนำไปรันบนเครื่องควอนตัมคอมพิวเตอร์จริง ในงานวิจัยนี้ ผู้เขียนได้ทดลองเปรียบเทียบระหว่างวงจรจาก Algorithm 1 กับ FABLE โดยรับอินพุตเป็นจำนวนมวล N สร้างเป็นบล็อกเอ็นโค้ดดิงของเมทริกซ์ลาปลาซ F กรณีที่ไม่พิจารณาค่ามวลและค่านิจสปริง กล่าวคือ ค่า $F = BB^\dagger$ จากการวัดในช่วง $N = 2$ ถึง 280 สำหรับระบบมวลเส้นตรง ได้ผลลัพธ์เป็นดังรูป 5.1 และ สำหรับระบบมวลวงกลม ได้ผลลัพธ์เป็นดังรูป 5.2



รูปที่ 5.1: ผลลัพธ์การทดลองจากการวัดความลึกวงจร : ระบบมวลเส้นตรง

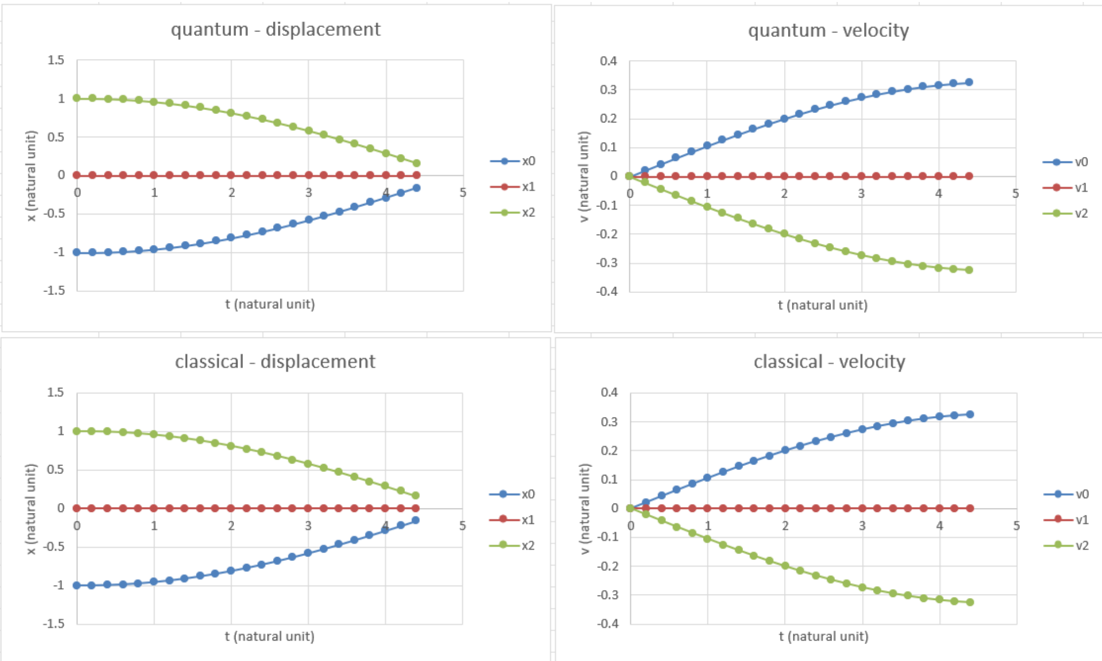


รูปที่ 5.2: ผลลัพธ์การทดลองจากการวัดความลึกวงจร : ระบบมวลวงกลม

5.2 ผลการจำลองการสั่นคู่ควบ

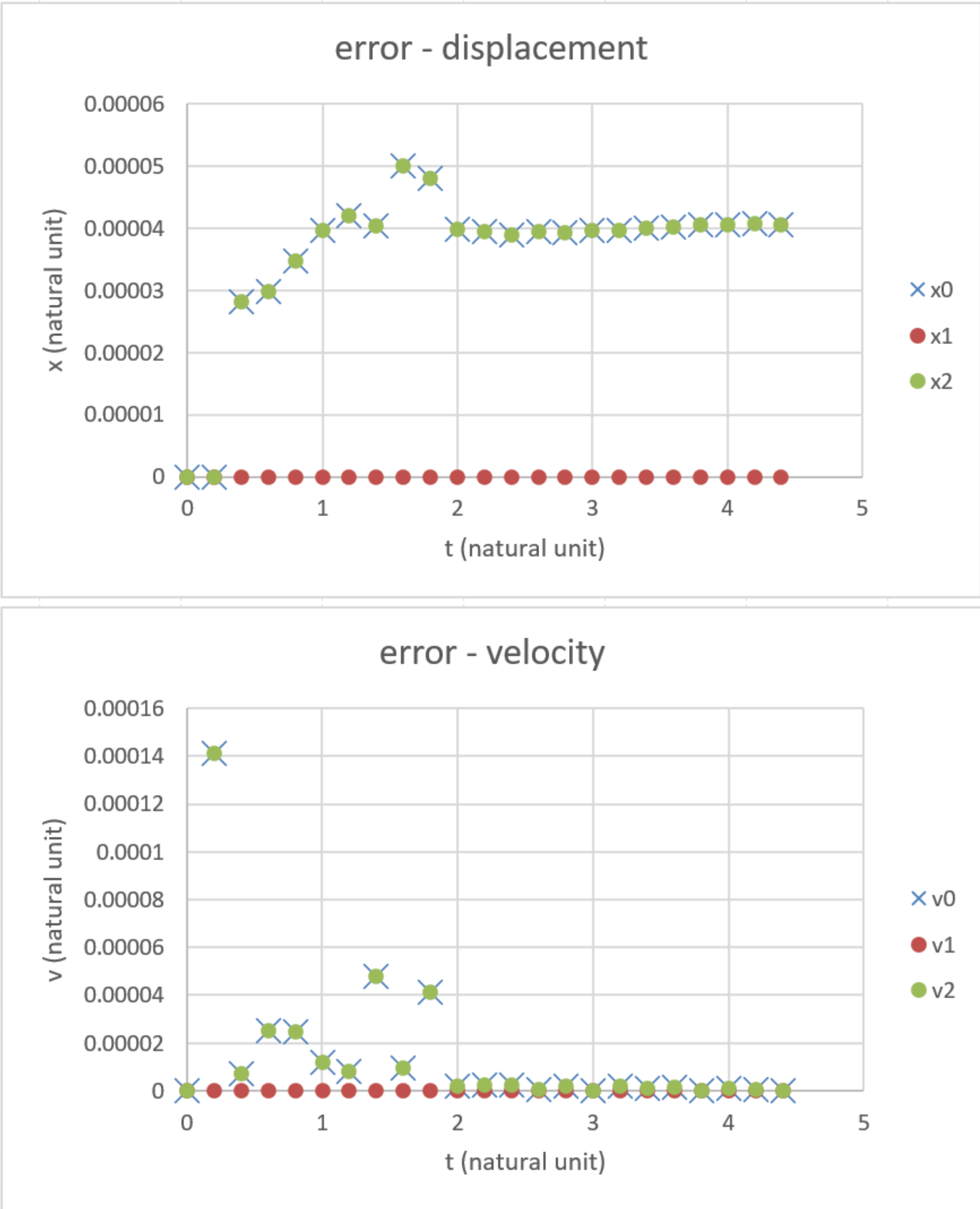
ระบบมวล 3 ก้อน

ในส่วนนี้ ผู้เขียนทดลองจำลองระบบมวล 3 ก้อน เชื่อมต่อเป็นวงกลม โดยกำหนดตัวอย่างให้ $m_0 = 12, m_1 = 30, m_2 = 12$ และให้ $k_0 = 0.89, k_1 = 0.89, k_2 = 0.2$ โดยเริ่มต้นให้ $x_0(0) = -1, x_1(0) = 0, x_2(0) = 1$ และ $v_0(0) = v_1(0) = v_2(0) = 0$ โดยจำลองตั้งแต่ $t_i = 0$ จนถึง $t_f = 4.4$ เว้นระยะแต่ละช่วง $\Delta t = 0.2$ และให้ $\epsilon = 10^{-2}$ ระบบดังกล่าวใช้เวลาประมวลผลประมาณ 12 ชม. ให้ผลดังรูปที่ 5.3



รูปที่ 5.3: ผลลัพธ์การทดลองระบบมวล 3 ก้อน

เนื่องจากผลลัพธ์จากทั้งสองฝั่งใกล้เคียงกันมาก ผู้เขียนจึงได้วัดความต่างด้วยค่าคลาดเคลื่อนสัมบูรณ์ (absolute error) ระหว่างการจำลองโดยใช้วงจรกับการคำนวณปกติ ให้ผลดังรูปที่ 5.4

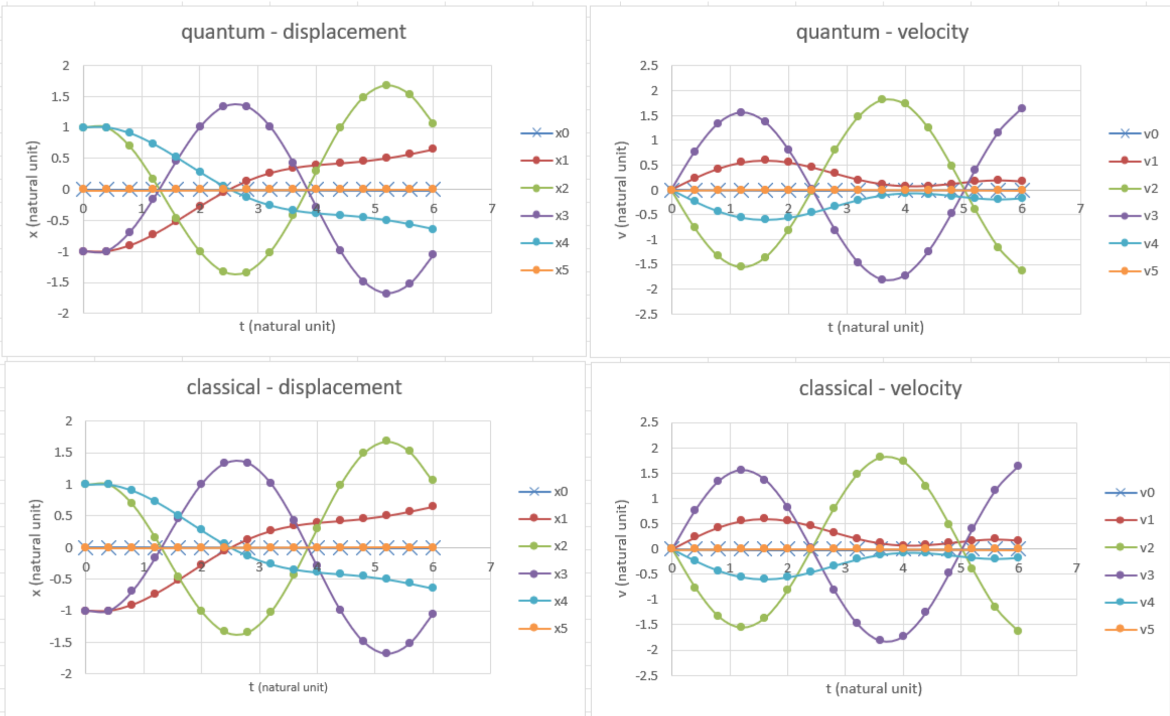


รูปที่ 5.4: ค่าคลาดเคลื่อนสัมบูรณ์ของการทดลองระบบมวล 3 ก้อน

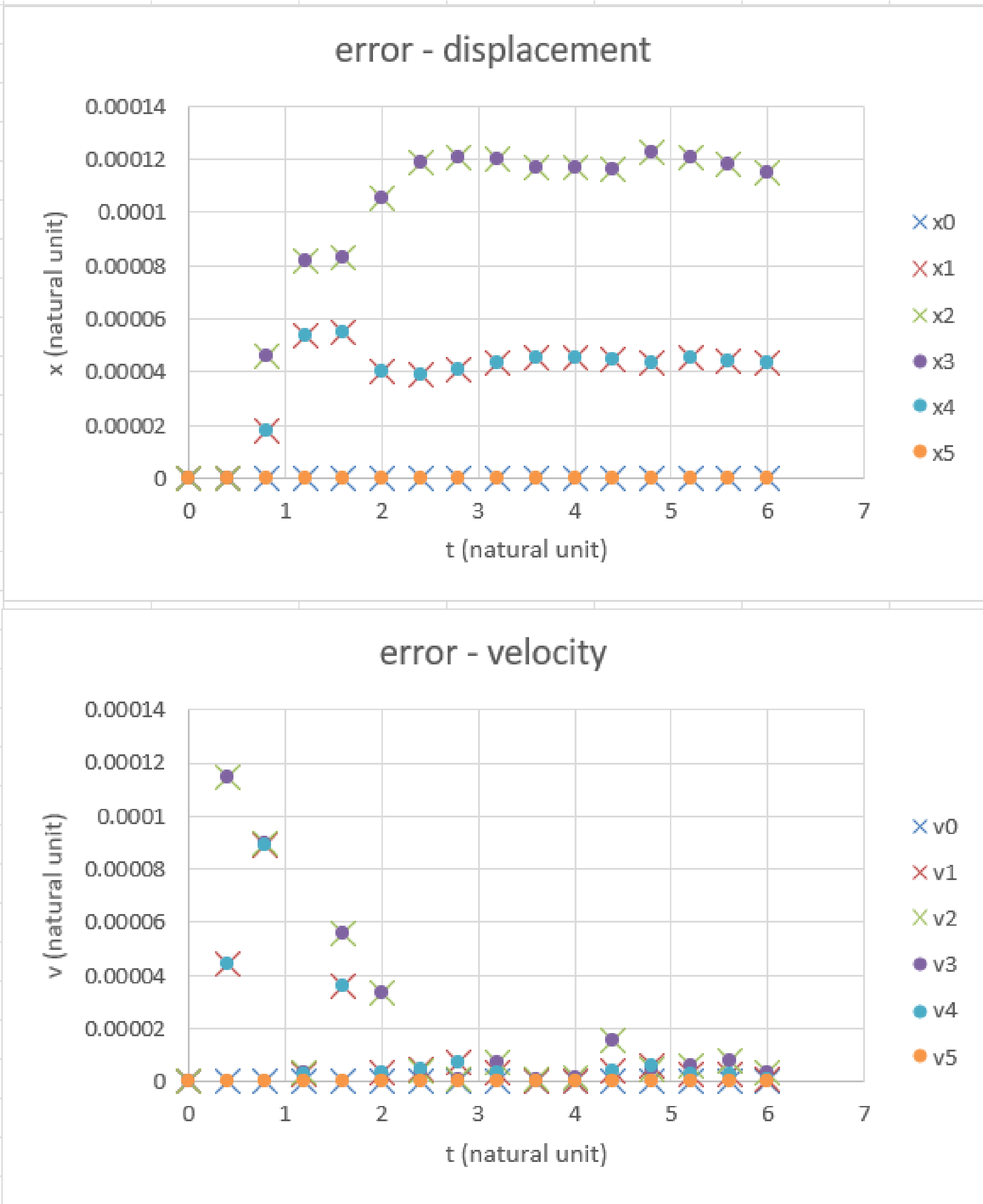
ระบบมวล 6 ก้อน

ผู้เขียนทดลองจำลองระบบมวล 6 ก้อน เชื่อมต่อเป็นเส้นตรง โดยกำหนดตัวอย่างให้ $m_0 = 99999, m_1 = 5, m_2 = 2, m_3 = 2, m_4 = 5, m_5 = 99999$ และให้ $k_0 = k_1 = k_2 =$

$k_3 = k_4 = k_5 = 1$ โดยเริ่มต้นให้ $x_0(0) = 0, x_1(0) = -1, x_2(0) = 1, x_3(0) = -1, x_4(0) = 1, x_5(0) = 0$ และ $v_0(0) = v_1(0) = v_2(0) = v_3(0) = v_4(0) = v_5(0) = 0$ โดยจำลองตั้งแต่ $t_i = 0$ จนถึง $t_f = 6$ เว้นระยะแต่ละช่วง $\Delta t = 0.4$ และให้ $\epsilon = 10^{-2}$ สาเหตุที่กำหนดมวลเช่นนี้เนื่องจากต้องการสังเกตพฤติกรรมของมวล m_0 กับ m_5 ซึ่งตามทฤษฎี ทั้งสองก้อนจะประพฤติตัวเหมือนกำแพง โดยจะขยับด้วย x ที่น้อยมาก ๆ จากการทดลอง ระบบดังกล่าวใช้เวลาในการประมวลผลประมาณ 52 ชั่วโมง ผลลัพธ์ของการรันวงจรควอนตัมให้ผลลัพธ์ดังรูป 5.5 และมีค่าคลาดเคลื่อนสัมบูรณ์ดังรูป 5.6



รูปที่ 5.5: ผลลัพธ์การทดลองระบบมวล 6 ก้อน

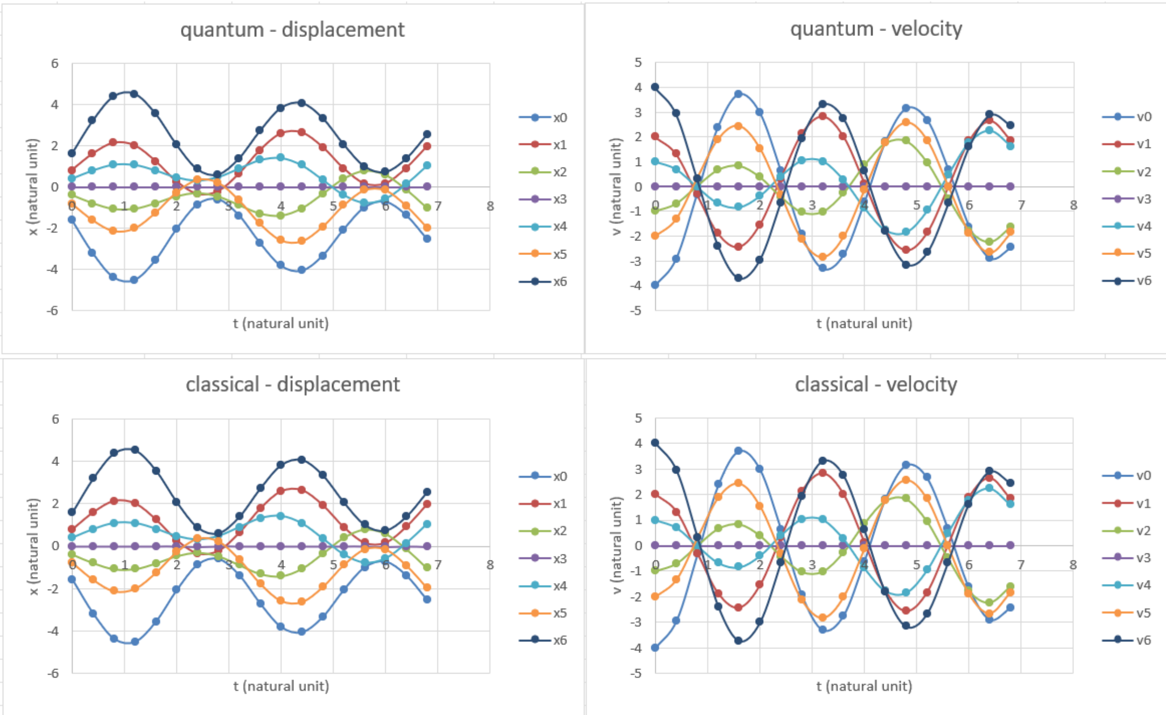


รูปที่ 5.6: ค่าคลาดเคลื่อนสัมบูรณ์ของการทดลองระบบมวล 6 ก้อน

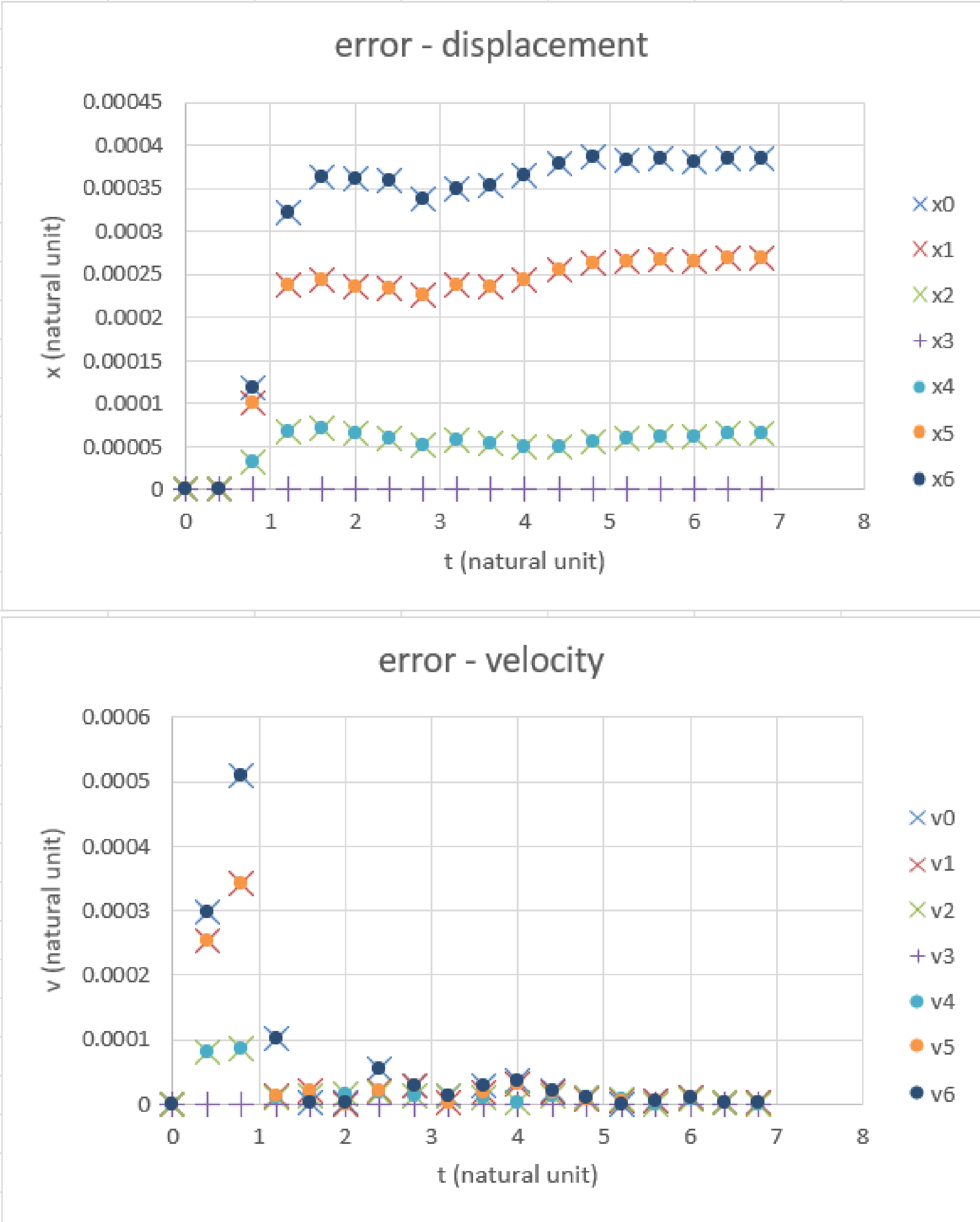
ระบบมวล 7 ก้อน

ผู้เขียนทดลองจำลองระบบมวล 7 ก้อน เชื่อมต่อเป็นวงกลม โดยกำหนดตัวอย่างให้ $m_0 = m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 1$ และให้ $k_0 = k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = 1$ โดยเริ่มต้นให้ $x_0(0) = x_1(0) = x_2(0) = x_3(0) = x_4(0) = x_5(0) = x_6(0) = 0$

และ $v_0(0) = -4, v_1(0) = 2, v_2(0) = -1, v_3(0) = 0, v_4(0) = 1, v_5(0) = -2, v_6(0) = 4$ โดยจำลองตั้งแต่ $t_i = 0$ จนถึง $t_f = 6.8$ เว้นระยะแต่ละช่วง $\Delta t = 0.4$ และให้ $\epsilon = 10^{-2}$ โดยสามารถเห็นการสั่นเป็นคาบ (periodic) ได้จากระบบนี้ ระบบดังกล่าวใช้เวลาในการประมวลผลประมาณ 60 ชั่วโมง ผลลัพธ์ของการรันวงจรควอนตัมให้ผลลัพธ์ดังรูป 5.7 และมีค่าคลาดเคลื่อนสัมบูรณ์ดังรูป 5.8



รูปที่ 5.7: ผลลัพธ์การทดลองระบบมวล 7 ก้อน



รูปที่ 5.8: ค่าคลาดเคลื่อนสัมบูรณ์ของการทดลองระบบมวล 7 ก้อน

บทที่ 6

สรุปงานวิจัย

วิทยานิพนธ์ฉบับนี้เป็นการศึกษาการสร้างวงจรควอนตัมเพื่อศึกษาและจำลองระบบการสั่นคู่ควบ โดยอาศัยหลักการของบล็อกเอ็นโค้ดดิงควบคู่กับ QSVT ในการสร้างวงจร โดยเมทริกซ์ที่นำมาใช้ได้มาจากการจัดรูปเมทริกซ์ตามสมการของระบบมวลติดสปริงและทำให้อยู่ในรูปที่เหมาะสมเพื่อจำลองระบบดังกล่าวบนควอนตัมตามแนวคิดที่ (Babbush et al., 2023) ได้วางไว้ โดยงานนี้ได้ออกแบบอัลกอริทึมสำหรับการสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์อุบัติการณ์ B เพื่อสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ H โดยอาศัยวงจรบล็อกเอ็นโค้ดดิงที่มีอยู่ในงานวิจัยอื่นมาประกอบ ภายหลังจากการประกอบรวมเข้ากับวงจรส่วนอื่นเพื่อสร้างบล็อกเอ็นโค้ดดิงของ e^{-iHt} และป้อนออราเคิลของสถานะเริ่มต้น จึงค่อยนำไปใช้ในการจำลอง ภายในงานวิจัยนี้ทดลองกับระบบมวล 3 ก้อน, 6 ก้อน และ 7 ก้อน ตามลำดับ พร้อมทั้งนำวงจรที่สร้างขึ้นจากอัลกอริทึม สร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ลาปลาซและเปรียบเทียบกับวงจรบล็อกเอ็นโค้ดดิงจากเฟรมเวิร์คที่มีอยู่แล้ว โดยสรุปผลการทดลองได้ดังนี้

เมื่อนำวงจรบล็อกเอ็นโค้ดดิงของ B จากอัลกอริทึมมาสร้างบล็อกเอ็นโค้ดดิงของเมทริกซ์ลาปลาซ และนำมาเปรียบเทียบกับวงจรบล็อกเอ็นโค้ดดิงของเมทริกซ์ลาปลาซที่ได้จากเฟรมเวิร์ค FABLE จะเห็นได้ชัดเจนว่าวงจรจากทั้งสองแหล่งมีจำนวนความลึกโตเป็นเลขชี้กำลังโดยขึ้นกับจำนวนคิวบิตหลักที่ใช้ โดยผลลัพธ์วงจรจากอัลกอริทึมของทั้งสองระบบใช้ความลึกน้อยกว่าผลลัพธ์วงจรของเฟรมเวิร์ค FABLE อย่างชัดเจน โดยในระดับที่ใช้คิวบิตหลักเท่ากัน วงจรที่ได้จากอัลกอริทึมของระบบมวลเส้นตรงจะมีความลึกแตกต่างกันมากกว่าระบบมวลแบบวงกลม ในขณะที่วงจรของทั้งสองระบบจาก FABLE ไม่เห็นความแตกต่างได้อย่างชัดเจน

ผลลัพธ์การทดลองจากมวล 3 ก้อน ด้วยช่วงและเงื่อนไขเริ่มต้นดังกล่าว ทำให้เห็นว่ามวลทั้ง 3 ก้อนเคลื่อนที่ด้วยอัตราเร็วที่ต่ำและไม่เห็นคาบของการสั่น ผลการจากการจำลองใกล้เคียงกับการคำนวณปกติ โดยการกระจัดมีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของทุก t เท่ากับ 2.41×10^{-5} และความเร็วมีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของทุก t เท่ากับ 9.65×10^{-6} (หน่วยธรรมชาติ)

ผลลัพธ์การทดลองจากมวล 6 ก้อน ด้วยช่วงและเงื่อนไขเริ่มต้นดังกล่าว ทำให้เห็นว่ามวล m_2 และ m_3 เคลื่อนที่ด้วยอัตราเร่งสูงและเห็นการแกว่งเป็นคาบชัดเจน ในขณะที่มวล m_1 และ m_4 เคลื่อนที่ด้วยอัตราเร่งต่ำและไม่เห็นการแกว่งเป็นคาบ ในทางกลับกัน m_0 และ m_5 ไม่เคลื่อนที่ ซึ่งเป็นไปตามสมมติฐานที่วางไว้ ผลการจากการจำลองใกล้เคียงกับการคำนวณปกติ โดยการกระจัดมีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของทุก t เท่ากับ 4.38×10^{-5} และความเร็วมีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของทุก t เท่ากับ 1.17×10^{-5} (หน่วยธรรมชาติ)

ผลลัพธ์การทดลองจากมวล 7 ก้อน ด้วยช่วงและเงื่อนไขเริ่มต้นดังกล่าว ทำให้เห็นการสั่นเป็นคาบของมวลทุกก้อนยกเว้น m_3 ที่ไม่เคลื่อนที่ โดยมวล m_0 และ m_6 มีการกระจัดและความเร็วที่ลดลงในแต่ละคาบ ในขณะที่มวล m_1, m_2, m_4 และ m_5 มีการกระจัดและความเร็วที่เพิ่มขึ้น ผลการจากการจำลองใกล้เคียงกับการคำนวณปกติ โดยการกระจัดมีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของทุก t เท่ากับ 1.65×10^{-4} และความเร็วมีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของทุก t เท่ากับ 3.48×10^{-5} (หน่วยธรรมชาติ)

จากการผลทดลองทั้งหมดทำให้สรุปได้ว่า การสร้างวงจรควอนตัมโดยใช้บล็อกเอ็นโค้ดดิงประกอบกับ QSVT ดังกล่าว ให้ผลลัพธ์การจำลองที่ชัดเจนโดยให้ค่าคลาดเคลื่อนในระดับที่ต่ำกว่าที่กำหนดไว้ จึงกล่าวได้ว่าวงจรบล็อกเอ็นโค้ดดิงในงานนี้สามารถจำลองโดยให้ผลลัพธ์ที่ถูกต้องภายใต้ค่าคลาดเคลื่อนที่กำหนดได้ และวงจรจากอัลกอริทึมมีจำนวนความลึกที่ต่ำกว่าวงจรจากเฟรมเวิร์ค FABLE ซึ่งคาดคะเนได้ว่าวงจรจากอัลกอริทึมอาจให้ประสิทธิภาพการจำลองที่ดีกว่าวงจรจากเฟรมเวิร์ค FABLE ด้วย สิ่งเหล่านี้เป็นการยืนยันว่าเราสามารถสร้างและใช้งานวงจรบล็อกเอ็นโค้ดดิงเพื่อการจำลองการสั่นคู่ควบของมวลติดสปริงได้

ทั้งนี้ หากพิจารณาการจำลองวิธีนี้เทียบกับทรอตเทอไรเซชัน ซึ่งเป็นการแยกออกในรูปพจน์ค่าชี้กำลังของเมทริกซ์จำนวนมากและจำเป็นต้องอาศัยขั้นตอนในการคำนวณ ซึ่งซับซ้อนกว่าเมื่อเทียบกับการแยกพจน์ให้อยู่ในรูปฟังก์ชันโคไซน์และไซน์โดยอาศัย QSVT ในการทำ การสร้างวงจรด้วยวิธีทรอตเทอไรเซชันอาจต้องอาศัยเกตจำนวนมากและอาจเกิดค่าคลาดเคลื่อนมากกว่า โดยในรายละเอียดอาจต้องศึกษาควบคู่กับอัลกอริทึมควอนตัมประเภทอื่นซึ่งอาจเข้ามาช่วยให้การจำลองโดยใช้บล็อกเอ็นโค้ดดิงมีประสิทธิภาพดีมากขึ้น

อย่างไรก็ตาม ข้อจำกัดจากงานวิจัยฉบับนี้คือ ผลลัพธ์จากการจำลองยังต้องอาศัยการประมวลผลลัพธ์ข้อมูลจากเวกเตอร์สถานะโดยตรง ซึ่งไม่ได้เกิดจากการวัดค่าของความน่า

จะเป็น ทำให้วงจรนี้ไม่สามารถจำลองเพื่อให้ได้ผลลัพธ์บนเครื่องควอนตัมคอมพิวเตอร์จริงได้ ประกอบกับหากระบบที่ต้องการศึกษามีค่ามวลหรือค่านิจสปริงแต่ละตัวที่ไม่เท่ากันโดยสิ้นเชิง อาจทำให้เราต้องป้อนค่ามวลเข้าไปในบล็อกเอ็นโค้ดดิง $M^{-\frac{1}{2}}$ และ $W^{\frac{1}{2}}$ ด้วยความลึกวงจร $O(N)$ ซึ่งทำให้เราไม่สามารถใช้ประโยชน์จากคุณสมบัติทางควอนตัมซึ่งคาดหวังให้ใช้ทรัพยากรและการคำนวณในรูปพหุนามของ n หรือต่ำกว่า และแนวทางการปรับปรุงวงจรนี้ให้มีประสิทธิภาพมากขึ้นอาจต้องศึกษากันต่อไป

รายการอ้างอิง

Babbush, R., Berry, D. W., Kothari, R., Somma, R. D., and Wiebe, N. 2023. Exponential quantum speedup in simulating coupled classical oscillators. Physical Review X 13.4 (December 2023):

Berry, D. W., Childs, A. M., Cleve, R., Kothari, R., and Somma, R. D. 2015. Simulating hamiltonian dynamics with a truncated taylor series. Physical Review Letters 114.9 (March 2015):

Campbell, E. 2019. Random compiler for fast hamiltonian simulation. Physical Review Letters 123.7 (August 2019):

Camps, D. and Beeumen, R. V. 2022. Fable: Fast approximate quantum circuits for block-encodings. 2022 IEEE International Conference on Quantum Computing and Engineering (QCE) (2022): 104–113.

Camps, D. and Van Beeumen, R. 2020. Approximate quantum circuit synthesis using block encodings. Physical Review A 102.5 (November 2020):

Camps, D., Lin, L., Van Beeumen, R., and Yang, C. 2024. Explicit quantum circuits for block encodings of certain sparse matrices. SIAM Journal on Matrix Analysis and Applications 45.1 (2024): 801–827.

Chakraborty, S., Morolia, A., and Peduri, A. 2023. Quantum Regularized Least Squares. Quantum 7 (April 2023): 988.

Dalzell, A. M., McArdle, S., Berta, M., Bienias, P., Chen, C.-F., Gilyén, A., Hann, C. T., Kastoryano, M. J., Khabiboulline, E. T., Kubica, A., Salton, G., Wang, S., and Brandão, F. G. S. L. 2023. Quantum algorithms: A survey of applications and end-to-end complexities [Online]. Available from: https://arxiv.org/abs/2310.03011 [2023,].

DiVincenzo, D. P. 1996. Topics in quantum computers [Online]. Available from: https://arxiv.org/abs/cond-mat/9612126 [1996,].

Dong, Y., Meng, X., Whaley, K. B., and Lin, L. 2021. Efficient phase-factor evaluation in quantum signal processing. Physical Review A 103.4 (April 2021):

Feynman, R. P. 1982. Simulating physics with computers. 21.6 (1982): 467–488.

Gilyén, A., Su, Y., Low, G. H., and Wiebe, N. 2019. Quantum singular value transformation and beyond: exponential improvements fquantum matrix arithmetics. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC '19. : ACM.

Hatano, N. and Suzuki, M. 2005. Finding Exponential Product Formulas of Higher Orders, pp. 37–68. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-31515-5. doi: 10.1007/11526216_2. Available from: https://doi.org/10.1007/11526216$_2$.

Kieferová, M., Scherer, A., and Berry, D. W. 2019. Simulating the dynamics of time-dependent hamiltonians with a truncated dyson series. Physical Review A 99.4 (April 2019):

Limkumnerd, S. 2016. ฟิสิกส์ของการสั่นและคลื่น. โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 1 edition.

Lin, L. 2022. Lecture notes on quantum algorithms for scientific computation [Online]. Available from: https://arxiv.org/abs/2201.08309 [2022,].

Lloyd, S. 1996. Universal quantum simulators. Science 273.5278 (1996): 1073–1078.

Manin, I. I. 1980. Vychislimoe I Nevychislimoe. "Sov. radio,", Moskva.

Martyn, J. M., Rossi, Z. M., Tan, A. K., and Chuang, I. L. 2021. Grand unification of quantum algorithms. PRX Quantum 2.4 (December 2021):

Martyn, J. M., Liu, Y., Chin, Z. E., and Chuang, I. L. 2023. Efficient fully-coherent quantum signal processing algorithms for real-time dynamics simulation. The Journal of Chemical Physics 158.2 (January 2023):

Monroe, C., Campbell, W. C., Duan, L.-M., Gong, Z.-X., Gorshkov, A. V., Hess, P. W., Islam, R., Kim, K., Linke, N. M., Pagano, G., Richerme, P., Senko, C., and Yao, N. Y. 2021. Programmable quantum simulations of spin systems with trapped ions. Rev. Mod. Phys. 93 (Apr 2021): 025001.

Nielsen, M. A. and Chuang, I. L. 2019. Quantum computation and quantum information. Cambridge Cambridge University Press.

Takahira, S., Ohashi, A., Sogabe, T., and Usuda, T. S. 2021. Quantum algorithms based on the block-encoding framework for matrix functions by contour integrals [Online]. Available from: https://arxiv.org/abs/2106.08076 [2021,].

ประวัติผู้เขียนวิทยานิพนธ์

Natt Luangsiraporncha is a Master's of Engineering student in Department of Computer Engineering at Chulalongkorn University, Thailand. He graduated with a Bachelor's Degree of Computer Engineering at the same institute in 2020. From 2021 to 2023, he was a teaching assistant in the Computer Engineering faculty. Since 2019, he has been a member of Intelligent System Laboratory. His research interest includes quantum computation and machine learning with previous works about quantum learning circuit development and quantum walk implementation.