



Establishing Polarization Qubits from A Photon-Pair Source and
A Design of An Alternative Scheme of Universal Optical
Programmable Multi-Qubit Gates

Paphon Phewkhom

A Thesis Submitted in Fulfillment of the Requirements for the
Degree of Master of Science in Physics (International Program)

Prince of Songkla University

2022

Copyright of Prince of Songkla University



Establishing Polarization Qubits from A Photon-Pair Source and
A Design of An Alternative Scheme of Universal Optical
Programmable Multi-Qubit Gates

Paphon Phewkhom

A Thesis Submitted in Fulfillment of the Requirements for the
Degree of Master of Science in Physics (International Program)

Prince of Songkla University

2022

Copyright of Prince of Songkla University

Thesis Title Establishing polarization qubits from a photon-pair source and a design of an alternative scheme of universal optical programmable multi-qubit gates

Author Mr. Paphon Phewkhom

Major Program Physics (International Program)

Major Advisor

.....
(Asst. Prof. Dr. Pruet Kalasuwan)

Examining Committee :

.....Chairperson
(Asst. Prof. Dr. Chalongrat Daengngam)

.....Committee
(Asst. Prof. Dr. Pruet Kalasuwan)

.....Committee
(Dr. Ekkarat Pongophas)

The Graduate School, Prince of Songkla University, has approved this thesis as fulfillment of the requirements for the Master of Science Degree in Physics (International Program)

.....
(Asst. Prof. Dr. Thakerng Wongsirichot)
Acting Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

.....Signature

(Asst. Dr. Pruet Kalasuwan)

Major Advisor

.....Signature

(Mr. Paphon Phewkhom)

Candidate

I hereby certify that this work has not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree.

.....Signature

(Mr. Paphon Phewkhom)

Candidate

Thesis Title	Establishing polarization qubits from a photon-pair source and a design of an alternative scheme of universal optical programmable multi-qubit gates
Author	Mr. Paphon Phewkhom
Major Program	Physics (International Program)
Academic Year	2022

ABSTRACT

Due to the potential of quantum computers to revolutionize computation by solving some types of traditionally intractable problems by the classical computer, this field of study has become increasingly active and diverse, involving work on developing quantum processing hardware as well as research into applications with a potential large speed-up compared to simulations on classical (non-quantum) computers. One of the most promising methods for processing quantum information involves the use of photonic qubits, which allow for well-established and noise-free single-qubit operations. However, since there is no photon-photon interaction, processing the qubit-interaction property requires a nonlinear optical operation. Therefore, in the experiment section of this project, we created entangled photonic qubits by beginning with photon pair generation using a nonlinear crystal, then building up the complexity of the optical setup with the Hong-Oa-Mandel dip experiment, CHSH experiments, and eventually, photonic qubit generation and tomography. By optimizing the efficiency of the system in each experiment, the resultant qubit was successfully constructed with high fidelity to the designated Bell's state. Additionally, we experimented with the optical structure of the photon pairs loop, which delivered some issues with noise and low signal in the results. Nonetheless, the structure still showed potential to be improved and utilized in future experiments. The entangled qubits and photon loop created in this project can be used as a foundation for various quantum optic algorithms, especially the photon loop structure, which can be used for the iteration of an operator, which is used regularly in the standard method for constructing multi-qubit gates. However, the method has some limitations in that only unitary operators can be created. Finally, in the second section of this project, we explored an alternative scheme for constructing quantum optical multi-qubit gates. This purposed scheme, which uses the Hilbert-space expansion technique, is capable of independently programming each matrix element of the operator, allowing many different types of quantum operators to be realized in quantum optic experiments. The advanced scheme can theoretically achieve n-polarization-qubit optical reconfigurable quantum gates by arranging linear optical elements.

ชื่อวิทยานิพนธ์	การสร้างคิวบิตโพลาริซจากแหล่งกำเนิดโฟตอนคู่และการ ออกแบบโครงสร้างทางเลือกแบบตั้งโปรแกรมได้อย่างอิสระของ เกททางแสงหลายคิวบิต
ผู้เขียน	นาย ปพน ผิวขำ
สาขาวิชา	ฟิสิกส์ (หลักสูตรนานาชาติ)
ปีการศึกษา	2565

บทคัดย่อ

เนื่องจากศักยภาพของคอมพิวเตอร์ควอนตัมในการปฏิวัติการคำนวณด้วยการแก้ปัญหาบางประเภทที่ยากโดยคอมพิวเตอร์คลาสสิก การวิจัยในสาขาวิชานี้จึงมีบทบาทและความหลากหลายมากขึ้น ตั้งแต่พัฒนาฮาร์ดแวร์การประมวลผลควอนตัมตลอดจนการวิจัยเกี่ยวกับแอปพลิเคชันที่มีศักยภาพที่มีความเร็วสูงขึ้นมาเมื่อเทียบกับการจำลองบนคอมพิวเตอร์แบบคลาสสิก หนึ่งในวิธีการที่มีแนวโน้มดีที่สุดสำหรับการประมวลผลข้อมูลควอนตัมคือการใช้คิวบิตทางแสงซึ่งช่วยให้สามารถดำเนินการแบบคิวบิตเดี่ยวได้อย่างสมบูรณ์และปราศจากสัญญาณรบกวน อย่างไรก็ตามเนื่องจากสมบัติการไม่มีปฏิสัมพันธ์กันระหว่างโฟตอนการสร้างปฏิสัมพันธ์ระหว่างคิวบิตจึงต้องใช้การเนิ่นการทางแสงแบบไม่เชิงเส้นดั่งนั้นใน ส่วนการทดลองของวิทยานิพนธ์นี้เราจึงสร้างคิวบิตทางแสงที่พัวพันกันโดยเริ่มต้นด้วยการสร้างคูโฟตอนโดยใช้คริสตัลแบบไม่เชิงเส้นจากนั้นจึงเพิ่มซับซ้อนของการทดลองทางแสงด้วยการทดลอง Hong-Oa-Mandel, การทดลอง CHSH และสุดท้ายคือการสร้างคิวบิตเชิงแสงและการตรวจวัดคุณสมบัติด้วยการเพิ่มประสิทธิภาพของระบบในการทดลองแต่ละครั้งผลลัพธ์ที่ได้จึงถูกสร้างขึ้นด้วยความแม่นยำใกล้เคียงตามสถานะของเบลล์ในเชิงทฤษฎี นอกจากนี้เรายังทดลองกับโครงสร้างเชิงแสงของลูปคูโฟตอนซึ่งยังคงพบปัญหาบางอย่างเกี่ยวกับสัญญาณรบกวนและสัญญาณต่ำในผลลัพธ์ อย่างไรก็ตามโครงสร้างดังกล่าวยังคงมีศักยภาพในการปรับปรุงและนำไปใช้ในการทดลองในอนาคต คิวบิตที่พัวพันกันและโฟตอนลูปที่สร้างขึ้นในงานวิจัยนี้สามารถใช้เป็นรากฐานสำหรับอัลกอริทึมควอนตัมออปติคต่างๆโดยเฉพาะอย่างยิ่งโครงสร้างโฟตอนลูปซึ่งสามารถใช้สำหรับการวนซ้ำของตัวดำเนินการซึ่งใช้เป็นวิธีมาตรฐานสำหรับการสร้างมัลติเกทคิวบิต อย่างไรก็ตาม วิธีการมีข้อจำกัดบางประการที่สามารถสร้างตัวดำเนินการแบบรวมได้เท่านั้น ในส่วนที่สองของโครงการนี้ เราจึงได้สำรวจรูปแบบทางเลือกสำหรับการสร้างควอนตัมออปติคหลายคิวบิตเกท โครงร่างที่มีจุดประสงค์นี้ใช้เทคนิคการขยายฮิลเบิร์ตสเปซ ทำให้สามารถโปรแกรมองค์ประกอบในเมทริกซ์แต่ละตัวของตัวดำเนินการได้อย่างอิสระ ทำให้สามารถสร้างตัวดำเนินการควอนตัมประเภทต่างๆมากมายในการทดลองด้วยควอนตัมทางแสง ในทางทฤษฎีโครงสร้างขั้นสูงสามารถบรรลุควอนตัมเกทแบบออปติคอลบนคิวบิตเชิงแสงกตัวที่กำหนดค่าใหม่ได้โดยการจัดเรียงองค์ประกอบเชิงเส้นทางแสง

Acknowledgment

I would like to express my sincere gratitude to my advisor Dr. Pruet Kalasuwan who gave me the golden opportunity to do this thesis on the topic “Establishing polarization qubits from photon pairs source and design on alternative scheme of universal optical programmable multi-qubits gate” and also provide advice in several topics regarding the thesis.

I would also like to give special thanks to the funding support from the NSRF via the Program Management Unit for Human Resources and Institutional Development, Research and Innovation and the support from the U.Reka Program, sponsored by The Siam Commercial Bank Public Company Limited and managed by Digital Ventures Co., Ltd. With these funding supports the laboratory could have access to important optical tools and equipment which facilitated the experiments.

I am grateful to all of those with whom I have had the pleasure to work on this and other related projects. Each of the members of the Nanophotonics group has provided me with extensive guidance and taught me a great deal about both scientific research and life in general, especially my lab associate, Mr. Wisit Sornkrue, with whom I exchange multiple opinions and strategies regarding the experiments.

Finally, I would also like to thank my parents and friends, who have supported me spiritually throughout this project and my life in general.

Paphon Phewkhom

CONTENTS

CONTENT	PAGE
CONTENTS	VIII
LIST OF TABLES	X
LIST OF FIGURES	XI
LIST OF PAPERS AND PROCEEDINGS	XIII
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 LITERATURE REVIEW	4
CHAPTER 3 THEORY IN THE EXPERIMENT	7
3.1 Spontaneous parametric down conversion	7
3.2 Hong-Ou-Mandel interference	10
3.3 CHSH inequality	12
3.4 Polarization density matrix	14
3.5 Accidental count rate	17
CHAPTER 4 METHODOLOGY	18
4.1 Photon pairs from type-I BBO using a CW laser pump	18
4.2 Hong-Oa-Mandel dip	19
4.3 CHSH inequality	20
4.4 Bell's state experiment	21
4.5 Photon pairs from type-I BBO using an 80 MHz pulse laser pump	22
4.6 Loop using 50/50 Beam splitter	23
4.7 Improve loop setup with lens and reduce laser pulse frequency	24
CHAPTER 5 RESULT AND DISCUSSION	25
5.1 Photon pairs from type-I BBO using a CW laser pump	25
5.2 Hong-Oa-Mandel dip	26
5.3 CHSH experiment	27
5.4 Density matrix experiment	28
5.5 Photon pairs from type-I BBO using a 80MHz pulse laser pump	30

CONTENTS (CONTINUED)

CONTENT	PAGE
5.6 Loop using a 50-50 Beam splitter	31
5.7 Improve loop setup with lens and reduce laser pulse frequency	33
CHAPTER 6 AN ALTERNATIVE SCHEME OF UNIVERSAL OPTICAL PROGRAMMABLE MULTI-QUBIT GATES FOR POLARIZATION QUBITS	35
6.1 Hilbert-space expansion technique for two polarization qubits	35
6.2 Universal optical programmable multi-qubit gates	38
6.3 Discussion	43
CHAPTER 7 CONCLUSION	46
BIBLIOGRAPHY	47
APPENDIX	50
A. M matrices	51
B. Programing for coincident count	54
C. MATLAB coding for density matrix	77
D. Laser source specifications	87
E. Paper	89
VITAE	105

LIST OF TABLES

TABLE		PAGE
4.3.1	HWP angles used in 16 measurements to calculate the S value in the CHSH experiment	21
5.2.1	Result values from plot fitting of the HOM dip experiment	26
5.3.1	Coincident count rate and its standard deviation for each projective measurement in the CHSH experiment	27
5.3.2	Calculated E and S values in the CHSH experiment	28
5.4.1	Coincidence count rate for each basis measurement in the density matrix tomography experiment	29
5.4.2	Computed physical density matrix in (a) real part and (b) imaginary part	29
5.6.1	Peak indices and their properties from the histogram in Fig. 5.6.1	32
5.7.1	Peak indices and their properties from the histogram in Fig. 5.7.1	34

LIST OF FIGURES

FIGURE	PAGE	
3.1.1	The diagram depicts the optical axis defined in a nonlinear crystal	7
3.1.2	Defined angles between wave vectors K_0 , K_1 and K_2	9
3.2.1	Simple HOM experimental setup where two photons interfere in the beam splitter and are then measured by the detectors	10
3.3.1	CHSH inequality measurement setup where the state of the input qubit is set and then measured in multiple projective states using HWPs and PBSs	12
3.3.2	S value computed by using input state $ H\rangle_{A'} V\rangle_{B'} + V\rangle_{A'} H\rangle_{B'}$, $a = -22.5^\circ$ and $a' = 0^\circ$	13
4.1.1	Experimental setup for photon pair creation	18
4.2.1	Setup for the HOM dip experiment	19
4.3.1	Experimental setup for CHSH and state tomography	20
4.5.1	Setup for creating photon pairs with a pulse laser.	22
4.6.1	Experiment setup for creating a photon loop using a 50/50 beam splitter.	23
4.7.1	Experiment setup for creating a photon loop using a 50/50 beam splitter	24
5.1.1	(a). single count rate on ports A' and B' as a function of time. (b). delay histogram showing the number of counts triggered in port B' after a varied time delay from when port A' gets triggered.	25
5.2.1	Measurement result and fitting plot of the HOM dip experiment	26
5.4.1	Real part (a) and imaginary part (b) of the density matrix plotted in 3D columns	30
5.5.1	experiment result from using pulse laser as a photon pairs source (a). single count rate on port A' (bottom line) and B' (top line). (b). Delay histogram between two ports	30
5.6.1	A delay histogram between two ports in the experiment that guides one path of photon pairs into an optical loop	31
5.7.1	Delay histogram between two ports in the improved experiment with lens	33
6.1.1	(a) SPDC source to produce two pairs of entangled photons resulting as Eq. 6.1.2 (b) Experimental scheme to realize the two-qubit operation $\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2$ according to the input state (Eq. 6.1.3), which is derived from the source (Eq. 6.1.2)	35

LIST OF FIGURES (CONTINUED)

FIGURE	PAGE
6.1.2 (a) A nonlinear single-photon source gives the superposition of photon states (Eq. 6.1.10). (b) A linear optical circuit to perform the linear combination of four tensor products of single-qubit gates in Eq. 6.1.12. The photons in the circuit are delivered from eight paths in Fig. 6.1.2a. Before entering the circuit, all photons are initialized as the superposition of the identical input polarization states on different spatial paths in Eq. 6.1.10	36
6.2.1 Two coupled notional single-photon sources (ES_1 and ES_2) provide n entangled photons. The sources give the polarization states as Eq. 6.2.1	38
6.2.2 A combination block (C_{kl}), a series of beam splitters combining photons from ES_k and ES_l , supposing each ES produces n distinguishable single photons	39
6.2.3 A cascade of n ESs, where each ES produces n entangled single photons in separate paths, resulting the initial state as expressed in Eq. 6.2.7	40
6.2.4 The example of the local operations applied to the photon state after being initialized to the state in Eq. 6.2.7 on each ES output (ES_l for this example). This example represents just one term in Eq. 6.2.6. In order to assign the position (i, j) of the single-qubit operation $\hat{\gamma}$ (i, j) on the $(2^{n-1}) \times (2^{n-1})$ all-ones matrix, the local operations representing the position are applied on the paths $O_{(l,1)}, O_{(l,2)}, \dots, O_{(l,n-1)}$. In this example, the operators relating to entries on the all-ones matrix follow from Eq. 6.2.8	41
6.2.5 A modified combination block. Paths $O_{(p,m)}$ sharing p are grouped as a composite path $O'_{(p,n')}$ with a set of local operations in Fig.6.2.4, where $m = 1, 2, \dots, n'$. In this case, the composite paths $O'_{1,n'}$ and $O'_{k',n'}$ are combined by the modified combination block $C_{1,k',n'}$, resulting in the composite path $O''_{1,n'}$. This output composite path is the post-selection of the photons leaving at the output paths $O_{(l,1)}, O_{(l,2)}, \dots, O_{(l,n-1)}$	42
6.2.6 The application of the modified combination blocks to combine entire composite paths $O'_{1,n}, O'_{2,n}, \dots, O'_{n'-1,n}, O'_{n',n}$, resulting in the post-selection of only one composite path as the output	42

LIST OF PAPERS AND PROCEEDING**PAPER**

1. Pewkhom, P., S. Suwanna, and Pruet Kalasuwan. "Alternative Scheme of Universal Optical Programmable Multi-Qubit Gates for Polarization Qubits." *Quantum Information Processing* 19 (June 8, 2020).

CHAPTER 1

Introduction

The linear transformation of inputs into outputs is the heart of information processing in computer science. In order to realize universal information processing, highly versatile logic-gate devices are required to perform linear mapping between universal sets of inputs and outputs. Quantum computers work under the laws of quantum mechanics, using gate operations and information represented by quantum bits or qubits. They only need accessible single- and two-qubit gate operations to realize any unitary quantum gate [12]. During recent decades, various quantum approaches have experimentally proved the successful operation of single- and two-qubit gates [12, 21, 23, 26, 28, 34]. Beyond single- and two-qubit gates, verification tests in recent years produced decent outcomes from quantum processors housing programmable features [8, 16]. These achievements marked a critical advance in the field as they paved the way for future practical machines.

The use of photonic qubits is one of the most promising approaches to quantum information processing. In general, photonic qubits enable free-of-noise and well established single-qubit operations. The only drawback that prevents photons from performing efficient and scalable processing is their negligible photon–photon interaction. However, the problem can be overcome by the introduction of nonlinear optical operations [28, 31]. This approach was a fundamental element of an optical quantum processor circuit based on a cascade of beam splitters and phase shifters implanted on a single-photon chip [8, 32]. The circuit could potentially be programmed to perform any quantum unitary operation on path qubits on a fixed circuit. Furthermore, this design could possibly be modified to work with other degrees of freedom of photons, such as polarization [25].

In the experimental section of this research, we started by exploring the photon pair sources and their properties. By using the SPDC (spontaneous parametric down conversion) process, photon pair sources were produced by shining the laser beam into a nonlinear crystal [7, 9]. The daughter photons from the process were collected by optical fibers and then measured. In the second step, the same fibers were used to emit photon pairs perpendicular to each other into the 50/50 beam splitter. Using the translation stage to modify displacement before two beams interfere, two photons could be established to be indistinguishable for the detectors behind the beam splitter with the help of the Hon-Oa Mandel (HOM) dip [18, 30]. In the next experiment, the system was measured to prove its quantum properties by using the CHSH experiment setup [2, 3], which can be adapted from the HOM setup. After the quantum system was confirmed, the optical setup could be used as a foundation for numerous optical quantum systems. One of which was the experiment with state tomography, where the entangled polarization of the system is set and then measured [1]. Later experiments were photon looping setups, where one of the entangled paths entered the beam splitter with a chance to go into a loop, expanding the time delay before arriving at the detector. The loop system could be used in many advanced quantum algorithms, especially as another option for iterating the operator.

In order to optimize the system and achieve maximum quantum entanglement from the system, many optical elements were varied and switched throughout each step of the experiments. Lenses, filters, laser sources, and detectors were the main components that influenced the amount of photon pairs output. We could increase the efficiency and number of

photon pairs output by adjusting their properties and position in the experimental setup, which is a crucial factor in any optical quantum experiment.

Additionally, we did an experiment with photon loop structure. By guiding one of entangled paths into a beam splitter with a 50-50 chance of looping back to the beam splitter or being detected by the photon detector, the resulting delay histogram shows multiple peaks of looped signal. The system can be implemented with various kinds of algorithms. One of which is the iteration of an operator, where instead of having multiple copies of the same operator to do the iteration, we could use only one operator and loop the qubits into it multiple times.

This research experiment heavily relies on coincident photon counters. They can count the number of photons that enter two detectors in a fixed time interval. However, the existing program that comes along with the counter contains some restrictions and lacks a few applications. Therefore, we coded new programs to be used for the general purpose of the experiment, which involves measuring output from these counters. Using the exported data of the time at which the photon incident on the detector and the detector number that triggered, the program can calculate and plot the following values: histogram of the number of photons incident in each different time between two ports, number of coincident photons in a chosen period, and coincident count variation by time (Appendix B). It can also be expanded into specific calculations, such as density matrices (Appendix C) or CHSH.

One of the important components in quantum photonic circuits is the multi-qubit gate, which can be constructed from the foundation of an entangled qubit like the one created in this experiment. However, expecting higher efficiency in photon pair generation and photon detection in future technology and experimentation, we proposed an alternative method for constructing the multi-qubit gates where the scheme results in a low probability of success, but has multiple advantages over the standard one.

The standard method [8,33] for constructing an N -qubit unitary gate is based on the Reck et al. scheme, where an arbitrary unitary matrix $U(N)$ is factorized into a product of block matrices, where successive $U(2)$ transformations can be performed on two dimensional subspaces of the full N -dimensional Hilbert space. This architecture can adequately fulfill the functions of a programmable quantum gate, where the required circuit will need to be decomposed by the optical elements into the functions of reflectivity and phase shift to provide the programming input, and has been realized experimentally with good fidelity [8]. Moreover, the algorithm is recursive, similar to the Gaussian elimination process, and can increase the complexity of realizing an arbitrary quantum gate. More importantly, such a standard scheme is limited to unitary operation where the operator is known, hence can be decomposed into single- and two-qubit gates. However, in the emergence of quantum technology, a fundamental tool in quantum metrology may require a phase estimation where the estimation algorithm U is an unknown black box, that may not be decomposed at all [5, 38], particularly when the quantum phase estimation requires a conditional gate of the operator U . Therefore, the current standard method of realizing quantum gates based on the decomposition into elementary gates may not apply for these applications.

In the theoretical section, we propose a more versatile class of polarization-encoded qubit processors. We employ the Hilbert-space expansion technique in linear optical circuits, in which the linear combination of operations on the polarization states is available. This capability was already reported in previous experiments [32, 38] for a small number of path-encoding qubits. In this work, we adapt such a technique for polarization qubits and simplify

the circuit by designing some operators to make it suitable for possibly the lowest complexity realization of programmable multi-qubit gates. The concept starts with the superposition state of photon pairs from separate nonlinear elements such as BBO to produce two pairs of photons in four spatial paths. When recombining two photons from separate nonlinearities, assuming that they have undergone different linear transformations, the signal at the detector will be a combination of those transformations. By post-selecting appropriate outputs where only one photon leaves each path, all entries of the operator can be obtained.

This schematic design is different from the standard decomposition of an operation, which mostly relies on the products of operators. Using this alternative technique, instead of combining several single-qubit and two-qubit gates to produce a universal quantum gate, we directly program every entry of the matrix representing the operation. The manipulation of a quantum gate is also possible for all kinds of quantum operations, possibly represented by complex matrices, including non-unitary ones. The detail of the technique is reviewed in Chapter 6.1, where we also describe an improved procedure for constructing an optical programmable two-qubit gate. In Chapter 6.2, we explain how to realize optical programmable multi-qubit gates by this alternative approach, and in Chapter 6.3, we discuss their advantages and technical requirements for practical implementation.

CHAPTER 2

Literature review

Spontaneous parametric down conversion

According to the study by David C. Burnham and Donald L. Weinberg [7], in the experiment, the optical photon pairs were created by emitting a pumped laser beam into an ammonium dihydrogen phosphate crystal. The coincidence between the split photons could be observed with photon multipliers when the phase matching conditions were satisfied. The experiment was set up to achieve the highest coincident count rate possible by using optical alignment, wavelength filtering, and time delay between two channels. The theory regarding type-I spontaneous parametric down conversion is described in detail by L. Caspani, E. Brambilla, and A. Gatti [9]. The literature showed the cone-shape of coincidence photon pairs, where two points on the contour can be selected to optimize the coincident count rate. Their work also shows the photon pairs' cross-correlation with different crystal structures.

There are countless factors that account for the number of photon pairs generated by SPDC processes. One of them is how the pump beam is focused before splitting into daughter photons. In the experiment by H. Di Lorenzo Pires, F. M. G. J. Coppens, and M. P. van Exter [13], the pump laser beam was strongly focused into a nonlinear crystal, and then the near and far field intensity profiles were studied. The results demonstrate the variance properties of SPDC emission under a strong focusing laser pump, as well as potential applications.

The spectral properties of entangled photon pairs are studied in the work by So-Young Baek and Yoon-Ho Kim [4]. The results show that the two-photon pair spectrum can be particularly broad, and that experimenters should exercise caution when experimenting with bandwidth properties such as Hong-Ou-Mandel interference.

Christophe Couteau's article [11] compares the classical and quantum explanations of the SPDC process and shows some practical setups for producing efficient photon pairs. The work also presents some applications of SPDC photon pairs and their experimental setups, such as entangled photons and quantum computers.

Hong-Ou-Mandel interference

The next phase of creating entangled photons is making the photon pairs generated by SPDC interfere on a beam splitter. In the experiment by C. K. Hong, Z. Y. Ou, and L. Mandel [18], by adjusting the distance of one incoming photon before the interference, its time interval could be corrected, and the indistinguishable photon pairs were formed. The paper displays and theoretically explains the relationship between the coincident count rate and the adjusted displacement.

Similar work was done by T.B. Pittman, B.C. Jacobs, and J.D. Franson [30], where photon pairs produced by SPDC were interfered with in an optical single-mode fiber and the effect of wavelength filters used in front of the detectors was studied. The work shows that to increase the visibility of the interference, the appropriate bandpass filters should be carefully selected in the experiment. The effect of filters and other variables in Hong-Ou-Mandel interference is thoroughly explained in the article by Agata M. Brańczyk [15], where several types of filters were used in a mathematical model to explore the possible outcomes of Hong-Ou-Mandel dip.

CHSH inequality

One way to prove that the quantum system is not just a usual classical state is by measuring some states to exceed the CHSH inequality. Literature by Mnacho Echenim and Mehdi Mhalla⁹ detailly explain how to classify the quantum system by using few measurement results, then calculate the output value S in the region which cannot be obtained in classical mechanic. In the experimental work by Alain Aspect, Philippe Grangier, and Gérard Roger [2,3], the systems were measured in multiple projected measurements. By choosing the valid sets of the results, the computed values successfully violated the inequality and proved the experimental setup to be quantum system.

Sets of measurements to achieve this result can be selected before the experiment by analyzing the state of the system. Work by K. Muhammed Shafi, R. S. Gayatri, A. Padhye, and C. M. Chandrashekar [35] shows a mathematical method to compute the S value from different quantum state inputs and varied projective measurements. Their work also demonstrates the effect of mixed states on result measurements, which can cause the S value to differ significantly from the pure state case.

Polarization density matrix

After the quantum system is set in the desired state, it needs to be measured to prove that the state is correct and valid for use in future applications. J. B. Altepeter, D. F. V. James, and P. G. Kwiat's article [1] describes the procedures for obtaining the density matrix, which represents the quantum state of the system, by measuring multiple qubits in projective state, as well as numerical processes for computing the physical density matrix, which is a better representative of the system state because it satisfies all the properties of being real and physically possible. Work by Daniel F. V. James, Paul G. Kwiat, William J. Munro, and Andrew G. White [21] shows some important parameter calculations that are useful quantities to comprehend the state and valuable for future usages.

In the experimental work by S. Barz, G. Cronenberg, A. Zeilinger, and P. Walther [6], the quantum state was measured, and high fidelity between the measured state and the desired state could be achieved. showing that the experiment has the potential to be a foundation setup for more complicated optical quantum systems.

Optical multi-qubit gate

Qubit(s) gates in quantum computers play a similar role as the logic gates in classical computers; they manipulate the state of qubits and are arranged together to perform algorithms. Work by A. Barenco et al. [5] shows multiple quantum gates, which are fundamental in quantum computation, and shows that the universal gate, which performs on many qubits, can be expressed as a composition of these one- and two-qubit gates. In an optical quantum system, the properties of a HWP and a QWP can be used as a single qubit gate operator, as represented by N. Peters, J. Altepeter, E. Jeffrey, D. Branning, and P. Kwiat [29]. They showed that the polarization state of the system can be manipulated in a variety of ways, such as unitary, decohering, and polarizing operations. And in the article by E. Knill, R. Laflamme, and G.J. Milburn [22], two-qubit gates can be realized using nonlinear optics and post-selection.

The experiment by X-Q. Zhou et al. [37] shows an example of the linear combination of single qubit operators to perform a two-qubit gate called the control-U gate. They demonstrate various types of CU gates practically in an optical quantum system using the same

photon-pair and post-selection principle, with high fidelity between experimental and expected results. The work also shows the possibility of universal multi-qubit gates with a greater number of entangled qubits by adapting the same circuit design.

CHAPTER 3

Theory in the experiment

3.1 Spontaneous parametric down conversion

The spontaneous parametric down conversion (SPDC) process is one of several ways to create the photon pair sources for optical quantum algorithms. It is the mechanism where one photon can be split into two with half the frequency (thus, half the energy) after interacting within a nonlinear crystal. The phenomenon is described by the dielectric polarization vector \mathbf{P} , which can be written as a power series of the electric field vector \mathbf{E} as in Eq. 3.1.1.

$$\mathbf{P} = \epsilon_0 (\chi^{(1)}\mathbf{E} + \chi^{(2)}\mathbf{E}^2 + \chi^{(3)}\mathbf{E}^3 + \dots) \quad 3.1.1$$

where $\chi^{(n)}$ is the n-th order electric susceptibility. For an ordinary linear optical material, $\chi^{(1)}$ term is large and the linear behaviors take place. But when the electric field and $\chi^{(2)}$ is sizable, the wave mixing effect such as SPDC or SHG (second-harmonic generation) can occur. By using special material designed with $\chi^{(2)}$ term, two photons can be produced from one pump photon with conserved momentum and energy. There are two categories of SPDC separated by the phase matching condition. For Type I SPDC, the two daughter photons have polarization orthogonal to the source. Whereas for Type II, they are orthogonal to each other.

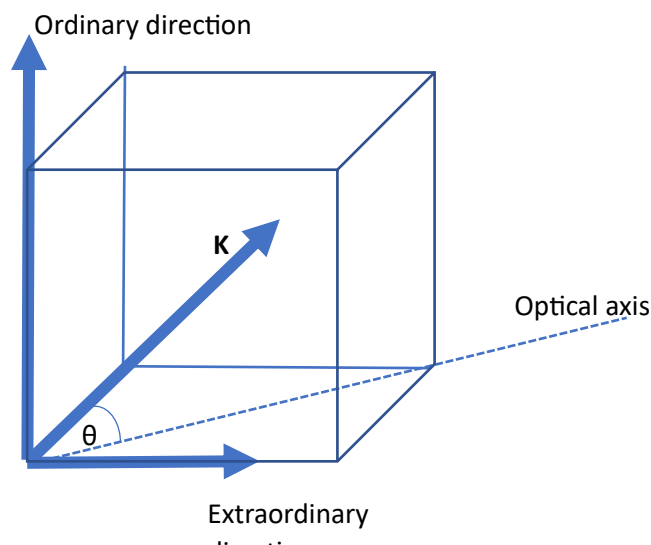


Fig. 3.1.1 The diagram depicts the optical axis defined in a nonlinear crystal

Birefringence crystal are designed with a different refractive index depending on the polarization of a pump laser beam. The crystal has the principle plane which the wave vector \mathbf{K} and the optic axis \mathbf{Z} are in plane with as shown in Fig. 3.1.1. Eq. 3.1.2 represents the refractive indices for ordinary direction (n_o), where the pump polarization is vertical to the principle plane, and Eq. 3.1.3 represents the refractive indices for extraordinary direction (n_e), where the pump polarization is in the principle plane and the beam propagation direction is perpendicular to the optical axis.

$$n_o^2 = A + \frac{B}{\lambda^2 - C} - D\lambda^2 \quad 3.1.2$$

$$n_e^2 = A' + \frac{B'}{\lambda^2 - C'} - D'\lambda^2 \quad 3.1.3$$

where the capital letters in the equation are constant values depending on the material, and λ is the pump wavelength. The refractive index in the ordinary direction is independent of the direction of propagation, whereas the refractive index in the principle plane $n_e(\theta)$ is depends on the angle (θ) between the direction of propagation and the optic axis given by Eq. 3.1.4.

$$n_e(\theta) = n_o \sqrt{\frac{1 + \tan^2 \theta}{1 + n_o^2/n_e^2 \tan^2 \theta}} \quad 3.1.4$$

For type I phase matching, a photon in ordinary polarization with wave vector \mathbf{K}_0 and frequency ω_0 is split into two photons with extraordinary polarization, wave vectors $\mathbf{K}_1, \mathbf{K}_2$ and frequencies ω_1, ω_2 . Using these definitions, the conservation of energy is shown in Eq. 3.1.5 and the conservation of momentum are shown in Eq. 3.1.6 and Eq. 3.1.7.

$$\omega_1 + \omega_2 = \omega_0 \quad 3.1.5$$

$$\mathbf{K}_0 = \mathbf{K}_1 \cos \theta_1 + \mathbf{K}_2 \cos \theta_2 \quad 3.1.6$$

$$0 = \mathbf{K}_1 \sin \theta_1 - \mathbf{K}_2 \sin \theta_2 \quad 3.1.7$$

with θ_1 and θ_2 are the half opening angles as shown in Fig. 3.1.2

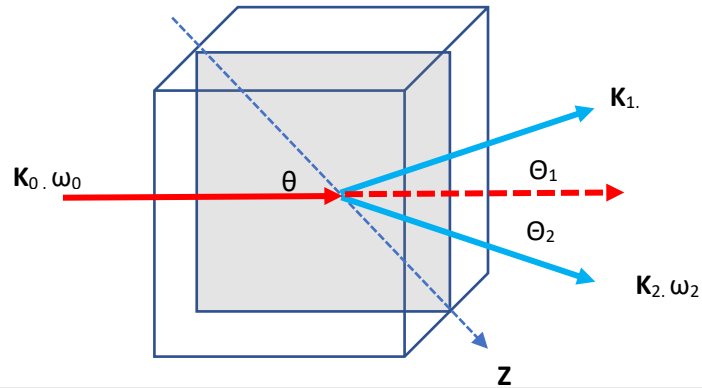


Fig 3.1.2 Defined angles between wave vectors \mathbf{K}_0 , \mathbf{K}_1 and \mathbf{K}_2

For phase matching conditions where $\omega_1 = \omega_2$, the phase matching angle θ can be calculated by Eq. 3.1.8

$$\tan^2 \theta = \frac{K_d^2 / [n_o^2(\omega_0) K_0^2] - 1}{1 - K_d^2 [n_e^2(\omega_0) K_0^2]} \quad \mathbf{3.1.8}$$

where K_d is the daughter photons's wave number. With this equation, the nonlinear crystal can be cut at the angle that is applicable to generate photon pairs at the designated opening angle. And by detecting daughter photons at both ends, the coincidence count from the photon pairs can be detected and modified.

3.2 Hong-Ou-Mandel interference

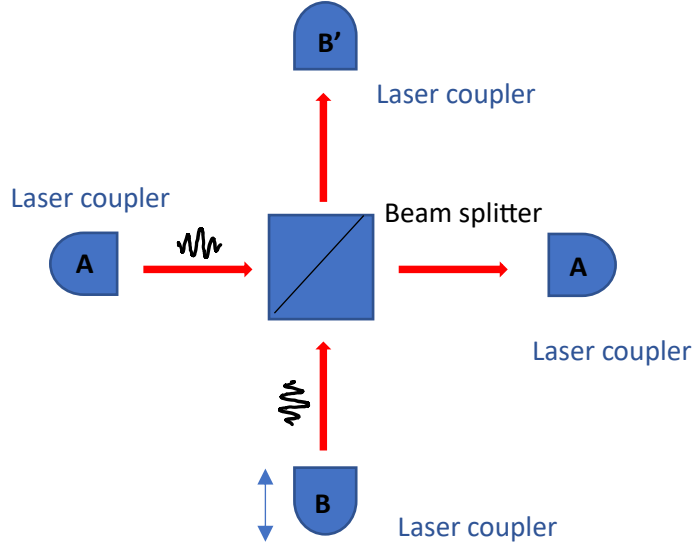


Fig. 3.2.1 simple HOM experimental setup where two photons interfere in the beam splitter and are then measured by the detectors

When photons enter a 50:50 beam-splitter as shown in Fig. 3.1.2, by using \hat{a}_A^\dagger and \hat{a}_B^\dagger as the creation operators to represent the input ports and $\hat{a}_{A'}^\dagger$, $\hat{a}_{B'}^\dagger$ for the output ports, the output operators can be written in terms of the input ports as shown in Eqs. 3.2.1 and 3.2.2

$$\hat{a}_{B'}^\dagger = \frac{1}{\sqrt{2}}(i\hat{a}_A^\dagger + \hat{a}_B^\dagger) \quad 3.2.1$$

$$\hat{a}_{A'}^\dagger = \frac{1}{\sqrt{2}}(i\hat{a}_B^\dagger + \hat{a}_A^\dagger) \quad 3.2.2$$

Here the i factors are the phase shifts resulting from the conservation of energy. If two input photons are indistinguishable, which means they have the same polarization and all other properties are equal, such as spectrum, arrival time, and transverse spatial mode. The output state, written in the number of photon states in the ports, can be calculated as in Eq. 3.2.3

$$|1\rangle_A |1\rangle_B = \hat{a}_A^\dagger \hat{a}_B^\dagger |0\rangle_A |0\rangle_B \xrightarrow{BS} \frac{1}{2} (i\hat{a}_{A'}^\dagger + \hat{a}_{B'}^\dagger)(i\hat{a}_{B'}^\dagger + \hat{a}_{A'}^\dagger) |0\rangle_{A'} |0\rangle_{B'} = \frac{i}{2} [|2\rangle_{A'} |0\rangle_{B'} + |0\rangle_{A'} |2\rangle_{B'}] \quad 3.2.3$$

The result state shows that there is no probability where photons are detected as "both reflected" or "both transmitted" because their amplitudes cancel out in the process.

In the Hong-Ou-Mandel experimental setup, the photons in the input ports are produced from a down conversion source, and the path lengths are varied with respect to each other to control the arrival time delay ($\delta\tau$) between the photons on the beam splitter and therefore adjust their level of distinguishability. Then the coincident count rate is measured by the rate at which photons are detected in both output ports in a fixed window time. This means the number of coincidences is higher when the two photons are more distinguishable and equal to zero (dip) when they are completely indistinguishable. The number of coincidences (N_C) is given by equation Eq. 3.2.4 when the photons have Gaussian spectral amplitude with bandwidth $\Delta\omega$.

$$N_C = C(T^2 + R^2) \left[1 - \frac{2RT}{R^2 + T^2} e^{-(\Delta\omega\delta\tau)^2} \right] \quad 3.2.4$$

where T and R mean the transmitted and refracted coefficients of the beam splitter, respectively. And C is a constant that is related to the coincident counts when two photons are totally distinguishable. From the graph plotted by Eq. 3.2.4 as a function of number of coincidences varied by time delay, the full width at half maximum (FWHM) is the coherence length of the photon. The visibility (V), which in this case is equal to the photon purity, can be calculated from the maximum and minimum of the coincident counts (N_{\max} and N_{\min}) given by Eq. 3.2.5.

$$V = \frac{N_{\max} - N_{\min}}{N_{\max} + N_{\min}} \quad 3.2.5$$

3.3 CHSH inequality

The statement that in quantum systems, the outcome is only determined after the measurement and has not pre-existed in some form of hidden value was one of the most controversial topics in quantum mechanics. Because when the entangled state is measured on one side, the other end will receive the information immediately and collapse into a single state. Here, the information about the state seems to be traveling faster than light speed, breaking the classical rule. According to one theory, there are some hidden variables in which the states are determined prior to measurement. which was later proven false by Bell's inequality, demonstrating a difference between quantum mechanics and the hidden variable theory. The CHSH (Clause, Horne, Shimony, and Holt) inequality is similar to the Bell's and can also be used to demonstrate the classical mechanics violation. The simple experimental setup is shown in Fig. 3.3.1. The polarization state of the input photon pairs can be assigned by HWPs on A and B paths. And the measurement state can be set using HWPs and PBSs at the output ends.

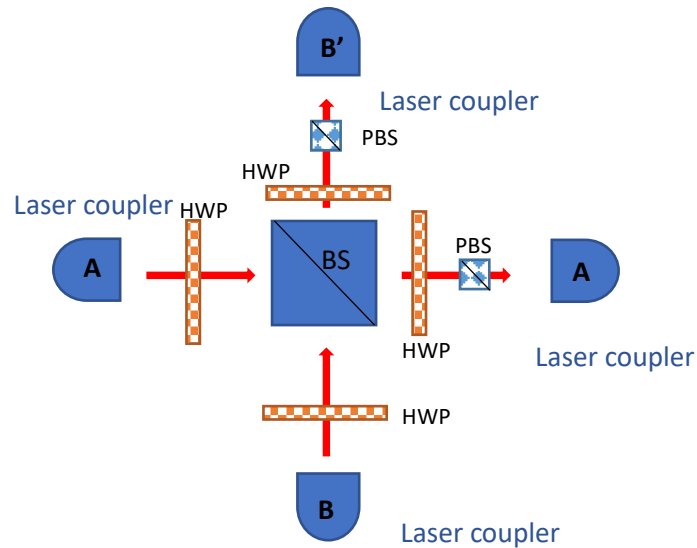


Fig 3.3.1 CHSH inequality measurement setup where the state of the input qubit is set and then measured in multiple projective states using HWPs and PBSs

Then, a series of coincidence counts are measured as a function of HWP angles on A' (a) and B' (b), and the $E(a, b)$ is calculated using Eq. 3.3.1

$$E(a, b) = \frac{N_{++} - N_{+-} - N_{-+} + N_{--}}{N_{++} + N_{+-} + N_{-+} + N_{--}} \quad 3.3.1$$

N represents the coincident counts where + and - sign designate the normal measurement and orthogonal measurement respectively. For instance, $N_{+-}(0,0)$ is the measurement with HWP angle of 0 degree on A' and 45 degrees on B'. The S value is then calculated from four sets of the E value as shown in Eq. 3.3.2

$$S = E(a, b) - E(a, b') + E(a', b) + E(a', b') \quad 3.3.2$$

where a' and b' are another set of angles in A' and B' paths, respectively. In classical mechanics, the maximum S value is 2, which is different from quantum mechanics, where the value can reach $2\sqrt{2}$ and exceed the classical boundary.

If the input polarization state is $|H\rangle_{A'}|V\rangle_{B'} + |V\rangle_{A'}|H\rangle_{B'}$, by using $a = -22.5$ degree and $a' = 0$ degree, the S value can be calculated as a function of b and b' angles, as shown in Fig. 3.3.2

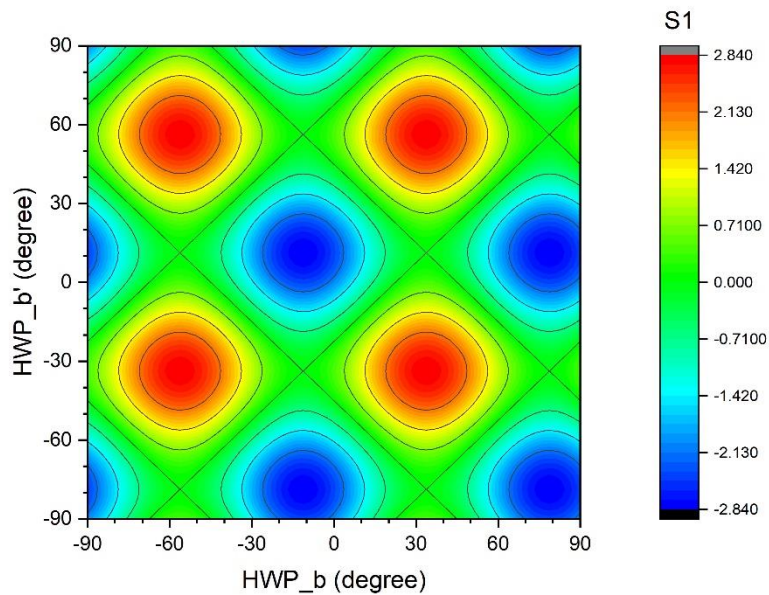


Fig. 3.3.2 S value computed by using input state $|H\rangle_{A'}|V\rangle_{B'} + |V\rangle_{A'}|H\rangle_{B'}$, $a = -22.5^\circ$ and $a' = 0^\circ$

From the contour graph, an appropriate set of angles can be selected to demonstrate the inequality in the experiment setup.

3.4 Polarization density matrix

The procedures for obtaining the density matrix, which represents the quantum state of the system, are described in the work by J. B. Altepeter, D. F. V. James, and P. G. Kwiat's [1].

Single qubit tomography

The density matrix of a single qubit ($\hat{\rho}$) can be represented by Eq. 3.4.1 where A, B, and C are real and non-negative constants, $A+B = 1$ and $C \leq \sqrt{AB}$. The matrix can also be written in terms of the sum of pure states with probabilities P_i for each orthogonal state $|\varphi_i\rangle$.

$$\hat{\rho} = \sum_i P_i |\varphi_i\rangle\langle\varphi_i| = \begin{bmatrix} A & Ce^{i\theta} \\ Ce^{-i\theta} & B \end{bmatrix} \quad 3.4.1$$

The density matrix has four important properties that are required to be physically and theoretically correct. First, it has the probability of conservation, which means $\text{Tr}(\hat{\rho}) = 1$. Secondly, the matrix is Hermitian ($\hat{\rho} = \hat{\rho}^\dagger$). The third and last requirements come from the positive semi-definiteness property, which implies that all the eigenvalues are zeroes, one, or between and the sum is 1, which causes $0 \leq \text{Tr}(\hat{\rho}^2) \leq 1$.

The definition of each polarization pure state in this thesis is defined as in equation Eq. 3.4.2, where H, V, D, A, R, and L are horizontal, vertical, diagonal, anti-diagonal, right-hand, and left-hand polarizations, respectively. and their mixed state forms can be determined by using

$$\begin{aligned} |H\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |V\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, |D\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, |A\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ |R\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}, |L\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} \end{aligned} \quad 3.4.2$$

To measure the single state in the form of a density matrix, also known as single qubit tomography, the process can be performed using stoke parameters that are defined from the set of measurements shown in Eq. 3.4.3 and Eq. 3.4.4

$$\begin{aligned} n_0 &= \frac{N}{2} (\langle H|\hat{\rho}|H\rangle + \langle V|\hat{\rho}|V\rangle), n_1 = N (\langle H|\hat{\rho}|H\rangle), \\ n_2 &= N (\langle D|\hat{\rho}|D\rangle), n_3 = N (\langle R|\hat{\rho}|R\rangle) \end{aligned} \quad 3.4.3$$

$$\begin{aligned} S_0 &= 2n_0, S_1 = 2(n_1 - n_0), \\ S_2 &= 2(n_2 - n_0), S_3 = 2(n_3 - n_0) \end{aligned} \quad 3.4.4$$

where N is a constant implied to the amplitude of photons count. n_i refer to the numbers of photons counted (or count rate) in each projective measurement, where n_0 refers to the measurement when half of the intensity of light is detected regardless of the polarization. n_1 ,

n_2 and n_3 are the measurements with horizontal, diagonal, and right circular projection. The density matrix can be calculated from the stoke parameters with Eq. 3.4.5.

$$\hat{\rho} = \frac{1}{2} \sum_{i=0}^3 \frac{S_i}{S_0} \hat{\sigma}_i \quad 3.4.5$$

where $\hat{\sigma}_i$ are Pauli spin operators defined by Eq. 3.4.6

$$\hat{\sigma}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \hat{\sigma}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \hat{\sigma}_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \hat{\sigma}_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad 3.4.6$$

Two qubits density matrix and tomography

For a two-qubit state, the density matrix can be written as in Eq. 3.4.7. It still has the same four properties to follow as in a single qubit state.

$$\hat{\rho} = \begin{bmatrix} A_1 & B_1 e^{i\phi_1} & B_2 e^{i\phi_2} & B_3 e^{i\phi_3} \\ B_1 e^{-i\phi_1} & A_2 & B_4 e^{i\phi_4} & B_5 e^{i\phi_5} \\ B_2 e^{-i\phi_2} & B_4 e^{-i\phi_4} & A_3 & B_6 e^{i\phi_6} \\ B_3 e^{-i\phi_3} & B_5 e^{-i\phi_5} & B_6 e^{-i\phi_6} & A_4 \end{bmatrix} \quad 3.4.7$$

The projective states of the matrix can be measured to perform qubit tomography. HWPs, QWPs, and PBSs on the transmitted path are used as a projective operator, as shown in Eq. 3.4.8.

$$\begin{aligned} \hat{U}_{HWP}(\theta) &= \begin{bmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{bmatrix}, \\ \hat{U}_{QWP}(\theta) &= \begin{bmatrix} \cos^2 \theta + i \sin^2 \theta & \sin \theta \cos \theta - i \sin \theta \cos \theta \\ \sin \theta \cos \theta - i \sin \theta \cos \theta & \sin^2 \theta + i \cos^2 \theta \end{bmatrix}, \\ \hat{U}_{PBS_t} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \quad 3.4.8$$

Using the experiment setup with the HWPs, QWPs and PBSs in order before the detector on both ends of the measurement, return the projection state as shown in Eq. 3.4.9.

$$\begin{aligned} \left| \Psi_{proj}^{(2)}(h_1, h_2, q_1, q_2) \right\rangle &= (\hat{U}_{HWP}(h_1) \cdot \hat{U}_{QWP}(q_1) \cdot \hat{U}_{PBS_t}) \\ &\otimes (\hat{U}_{HWP}(h_2) \cdot \hat{U}_{QWP}(q_2) \cdot \hat{U}_{PBS_t}) \end{aligned} \quad 3.4.9$$

where h_1, h_2, q_1 and q_2 are waveplate angles in path 1 and 2. The average coincident counts measured in the given state can be shown as in Eq. 3.4.10

$$n_\nu = N \langle \Psi_\nu | \hat{\rho} | \Psi_\nu \rangle \quad 3.4.10$$

Full sets of tomography states require at least 16 projective states to be measured. By using the simplest state projection for n_1 to n_{16} as HH, HV, VV, VH, RH, RV, DV, DH, DR,

DD, RD, HD, VD, VL, HL, and RL, a density matrix of 2 polarization qubits can be constructed using Eq. 3.4.11.

$$\hat{\rho} = (\sum_{\nu=1}^{16} \hat{M}_{\nu} n_{\nu}) / (\sum_{\nu=1}^4 n_{\nu}) \quad 3.4.11$$

Here \hat{M}_{ν} matrices (see Appendix A) are created to allow the density matrix to be constructed linearly. However, density matrix created by equation (2.26) still lag the properties of the positive semi-definiteness. In order to construct the density matrix where all the requirements are met, the physical density matrix $\hat{\rho}_P$ is defined by the formula in Eqs. 3.4.12 and 3.4.13.

$$\hat{T}(\mathbf{t}) = \begin{bmatrix} t_1 & 0 & 0 & 0 \\ t_5 + it_6 & t_2 & 0 & 0 \\ t_{11} + it_{12} & t_7 + it_8 & t_3 & 0 \\ t_{15} + it_{16} & t_{13} + it_{14} & t_9 + it_{10} & t_4 \end{bmatrix} \quad 3.4.12$$

$$\hat{\rho}_P(\mathbf{t}) = \hat{T}^{\dagger}(\mathbf{t}) \hat{T}(\mathbf{t}) / \text{Tr}(\hat{T}^{\dagger}(\mathbf{t}) \hat{T}(\mathbf{t})) \quad 3.4.13$$

where t_i are parameters in real numbers. The physical density matrix is constructed as a function of \mathbf{t} in such a way that $\hat{\rho}_P$ has all the constraint properties. Then the parameters are adjusted numerically by the program to ensure the maximum likelihood between the physical and measured density matrices. Which leads to the challenge of finding the minimum value of the function in Eq. 3.4.14.

$$L(t_1, t_2, t_3, \dots, t_{16}) = \sum_{\nu=1}^{16} \frac{(N \langle \Psi_{\nu} | \hat{\rho}_P(\mathbf{t}) | \Psi_{\nu} \rangle - n_{\nu})^2}{2N \langle \Psi_{\nu} | \hat{\rho}_P(\mathbf{t}) | \Psi_{\nu} \rangle} \quad 3.4.14$$

Quantities derived from density matrix

The von Neumann entropy (S) of the quantum state is defined by Eq. 3.4.15, the value of entropy is minimum at zero if $\hat{\rho}$ is a pure state and maximizes to $\ln(N_H)$ for the maximally mixed state, where N_H is the dimension of the Hilbert space. In the equation, p_a refers to an eigenvalue of $\hat{\rho}$.

$$S = -\text{Tr} \{ \hat{\rho} \log_2(\hat{\rho}) \} = -\sum_{a=1}^4 p_a \log_2(p_a) \quad 3.4.15$$

The linear entropy (P) is a quantity representing the degree of mixture of quantum states in a normalized form (maximum at 1). For two qubits state, the linear entropy is defined as in Eq. 3.4.16.

$$P = \frac{4}{3} (1 - \text{Tr} \{ \hat{\rho}^2 \}) = \frac{4}{3} (1 - \sum_{a=1}^4 p_a^2) \quad 3.4.16$$

The quantities concurrence (C), entanglement of formation (E) and tangle (T) correspond to the coherence properties of a mixed state and can be calculated by Eq. 3.4.17.

$$C = \text{Max}\{0, \sqrt{r_1} - \sqrt{r_2} - \sqrt{r_3} - \sqrt{r_4}\}$$

$$T = C^2$$

$$E = h\left(\frac{1 + \sqrt{1 - C^2}}{2}\right) \quad 3.4.17$$

where r_a are the eigenvalues of $\hat{R} = \hat{\rho}\hat{\Sigma}\hat{\rho}^T\hat{\Sigma}$ in decreasing order ($r_1 \geq r_2 \geq r_3 \geq r_4$). $\hat{\Sigma}$ here is the spin flip matrix defined by Eq. 3.4.18 and $h(x)$ is a function defined by Eq. 3.4.19.

$$\hat{\Sigma} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \quad 3.4.18$$

$$h(x) = -x \log_2 x - (1-x) \log_2 (1-x) \quad 3.4.19$$

3.5 Accidental count rate

When measuring coincident counts on two detectors receiving two photon paths, there are some probabilities where the count gets triggered by non-pair photons depending on the rate of the photon counts detected on a single port and the types of laser source used. For continuous laser, the accidental coincident count rate (R_{acc}) is shown in Eq. 3.5.1. R_A and R_B are single count rates on ports A and B, and T_w is the window time of a coincident count.

$$R_{acc} = R_A \times R_B \times T_w \quad 3.5.1$$

For pulse lasers, the computation is split into two cases: when the window time is larger or equal to the laser pulse period, the accidental coincident count rate is defined by Eq. 3.5.2, and for the opposite, it is shown in Eq. 3.5.3.

$$R_{acc} = R_A \times R_B / f_{rep} + R_d^2 \times T_w \quad 3.5.2$$

$$R_{acc} = (R_A \times R_B / f_{rep}) * (T_w / T_{pulse}) + R_d^2 \times T_w \quad 3.5.3$$

where T_{pulse} = laser pulse width, f_{rep} = laser pulse frequency and R_d = dark count rate.

CHAPTER 4

Methodology

4.1 Photon pairs from type-I BBO using a CW laser pump

Fig. 4.1.1 shows the experimental setup for generating photon pairs using type-I BBO with a CW laser pump (see Appendix D). Two mirrors are used to help with laser beam alignment from a 404-nm CW laser source. The laser couplers are mounted on 3-dimensional translation stages together with single mode fiber heads to locate the 808-nm wavelength beams produced by the SPDC process from type-I BBO crystal. The other ends of the fibers are plugged into coincidence photon counters, which can detect coincident photons that enter the receiver with time resolution in picosecond order. Time and port where the photons trigger the counters are recorded in a file, which is later processed by a program into a coincidence count histogram. (See Appendix B)

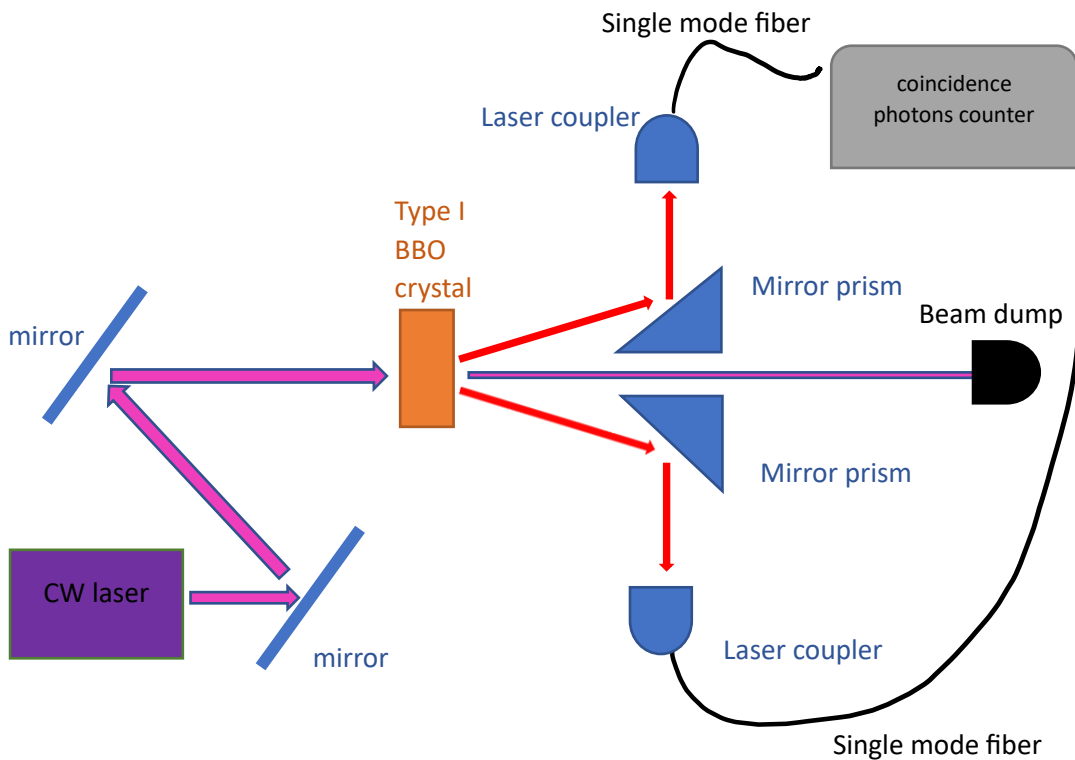


Fig. 4.1.1 Experimental setup for photon pair creation

4.2 Hong-Oa Mandel dip

Fig. 4.2.1 shows the setup for the Hong-Ou-Mandel experiment. The polarization maintain (PM) fibers are used to guide two photon pairs from part 1 into two laser couplers, which lead to two outgoing beams from another set of couplers instead of entering the photon counter. The two beam paths go through HWPs and QWPs before crossing each other in the 50-50 beam splitter. The waveplates are set to adjust the beams' polarization to horizontal in both paths. Whereas in the second part, all four couplers are mounted to the translation stages and arranged so that two outgoing beams cross perpendicularly to each other and to the beam splitter surface, as well as aligning with two receiver ends. The distance between one output coupler and the beam splitter is set to be varied in micrometer order by the translation stage along the displacement axis. The number of coincident counts per second where two photons enter both detectors by delay within three nanoseconds is recorded along with the displacement of the translation stage and then plotted into the Hong-Ou-Mandel dip fitting with Eq. 3.2.4.

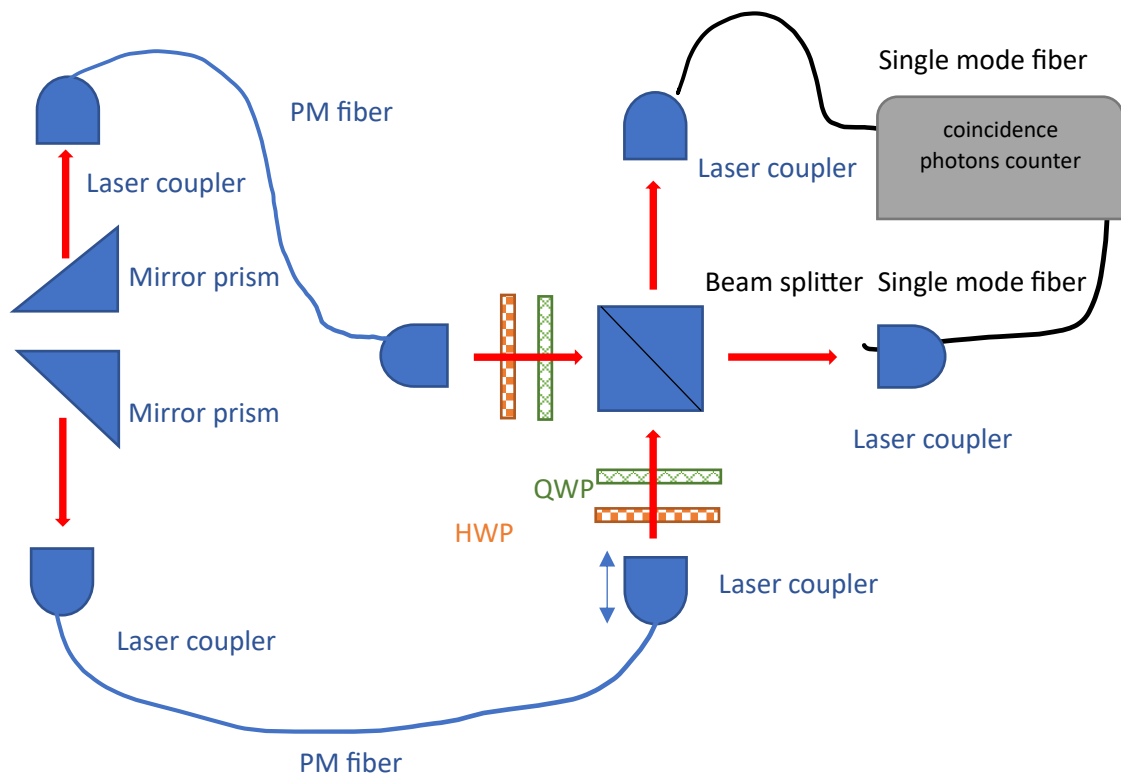


Fig. 4.2.1 Setup for the HOM dip experiment

4.3 CHSH inequality

CHSH inequality and Bell's state experiment can be operated using the same optical compositions. Fig. 4.3.1 shows the experiment setup for both studies. The distance between two laser couplers with an outgoing beam to the beam splitter is set to be equal in the second experiment. HWPs and QWPs are set after the beam emitted from the couplers to rotate one of the photon beams to vertical polarization, while another path is still horizontal polarization. By adding HWPs, QWPs, and PBSs in front of the laser couplers that sent the photons to the photon counters, two outgoing beams from the beam splitter can be measured on any polarization basis. For each experiment, there are sixteen waveplate angle sets of measurements.

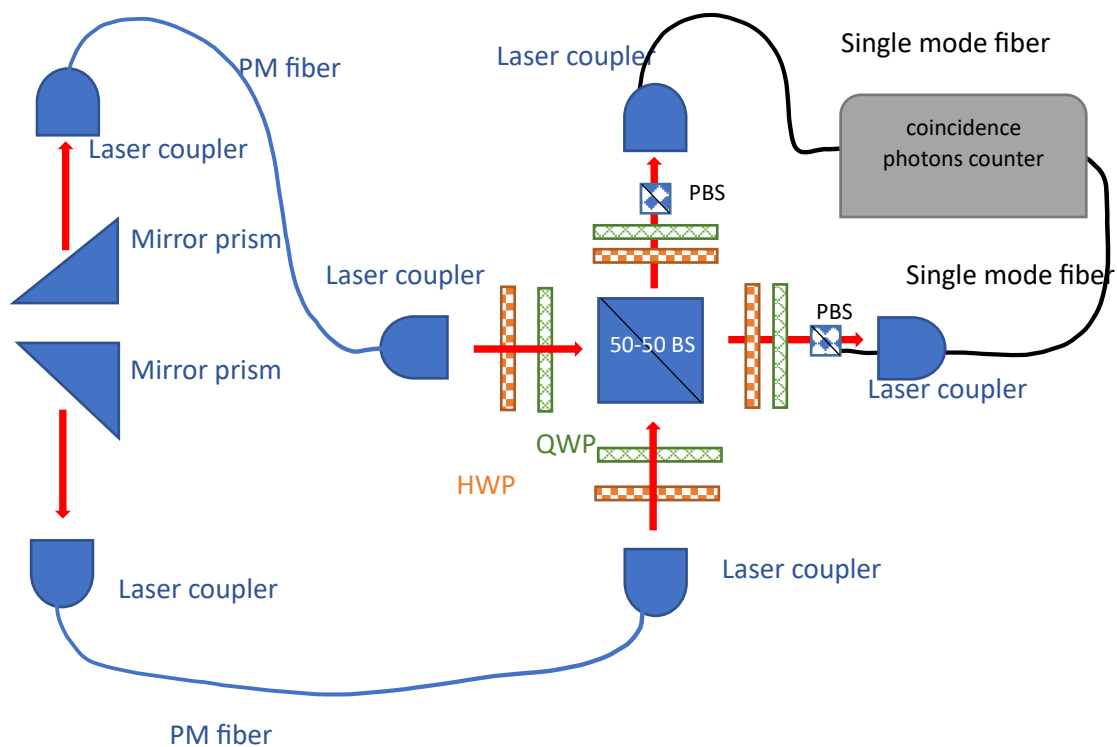


Fig. 4.3.1 Experimental setup for CHSH and state tomography

The CHSH experiment needs specific sets of measurements to achieve the inequality boundary as shown in Fig. 3.3.2. The angles of the HWPs in paths A' and B' are as shown in Table 4.3.1. Note that the angles are measured counterclockwise from the x-axis, looking in the same direction as beam propagation. Here both QWPs are set to 0 degrees from the x-axis.

Measurement number	HWP A' (degrees)	HWP B' (degrees)
1	-22.5	-12
2	-22.5	33
3	22.5	-12
4	22.5	33
5	-22.5	12.5
6	-22.5	57.5
7	22.5	12.5
8	22.5	57.5
9	0	-12
10	0	33
11	45	-12
12	45	33
13	0	12.5
14	0	57.5
15	45	12.5
16	45	57.5

Table. 4.3.1 HWP angles used in 16 measurements to calculate the S value in the CHSH experiment

The number of coincident counts per second that both counters triggered within three nanoseconds is recorded along with the experiment setup for each measurement in the experiments. Then the S value can be calculated for the CHSH experiment using Eq. 3.3.2.

4.4 Bell's state experiment

In Bell's state experiment, waveplate angles are selected to calculate the polarization density matrix. In Eq. 3.4.10, each waveplate set is used to measure the simple polarization basis for each path. Given that H is horizontal, V is vertical, D is diagonal, A is anti-diagonal, R is right-hand, and L is left-hand polarization. The sixteen basis measurements are HH, HV, VV, VH, RH, RV, DV, DH, DR, DD, RD, HD, VD, VL, HL, and RL, where the first and second letters correspond to A' and B' path measurements, respectively.

For each measurement in the experiments, the coincident counts rate is recorded in the same manner as in the CHSH experiment, and then the polarization density matrix and its valuable quantities can be computed using Eqs. 3.4.13–3.4.19 with the help of MATLAB programming in numerical computation (Appendix C).

4.5 Photon pairs from type-I BBO using an 80 MHz pulse laser pump

The experimental setup for photon pairs generated using a pulse laser pump (Appendix D) is similar to the CW pump setup as shown in Fig. 4.5.1 Due to the difference between beam profiles, the laser beam from the pulse laser source must pass through a focusing and collimating lens in order to be more efficient in generating photon pairs. The pulse laser frequency is set to its maximum at 80 MHz to achieve the highest photon pair count. The port and time bin where any photons trigger in the counter are then computed to generate the delay histogram.

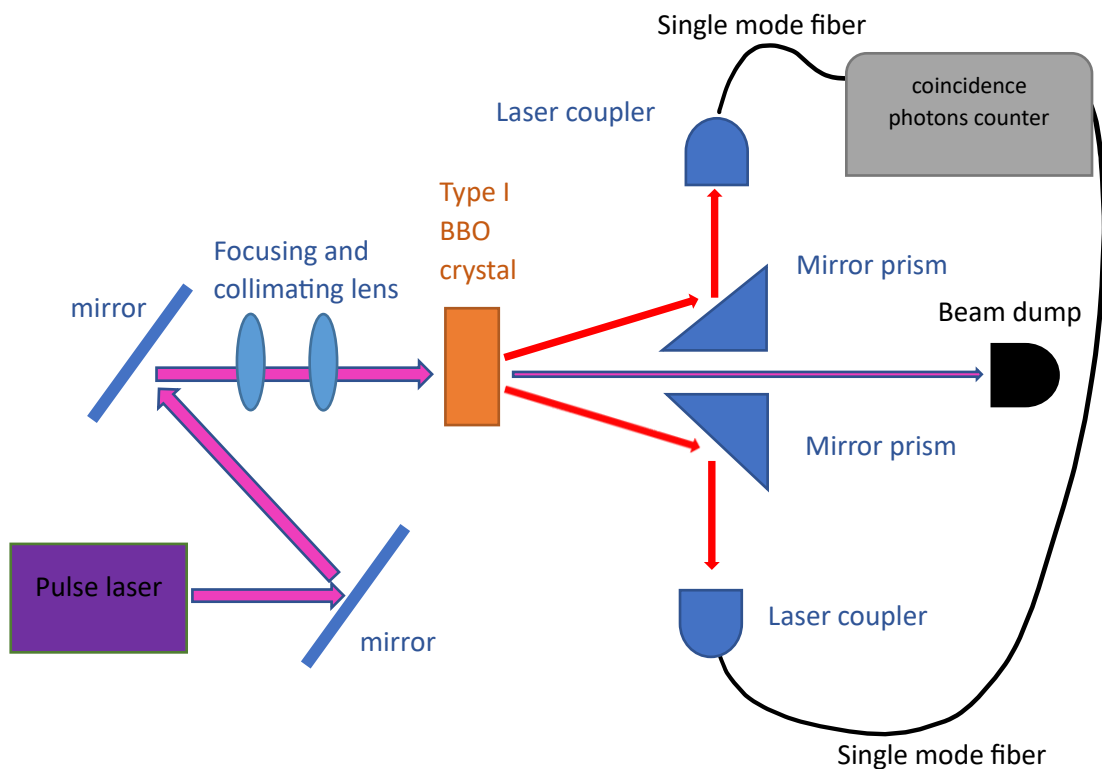


Fig 4.5.1 Setup for creating photon pairs with a pulse laser

4.6 Loop using 50/50 Beam splitter

Fig. 4.6.1 shows the experimental setup for photon loop using a beam splitter. One path of the photon pairs from experiment 4 is directly entering the photon counter. Whereas another path goes through a 50/50 beam splitter, which results in a 50% chance to enter the counter and a 50% chance to enter the 5-meter polarization maintenance fiber, where delay time is added when it triggers into the second port in the counter. The outgoing beam from the PM fiber also enters the beam splitter and therefore has a 50/50 chance of entering the port or adding more delay. Here, HWPs, QWPs, and PBSs are used to maintain the beam polarization and reduce noise from the laser source, which has perpendicular polarization compared to photon pairs. The delay histogram calculated from the recorded values of the photon counter is then analyzed to determine how many photon loops can be measured by the experimental setup.

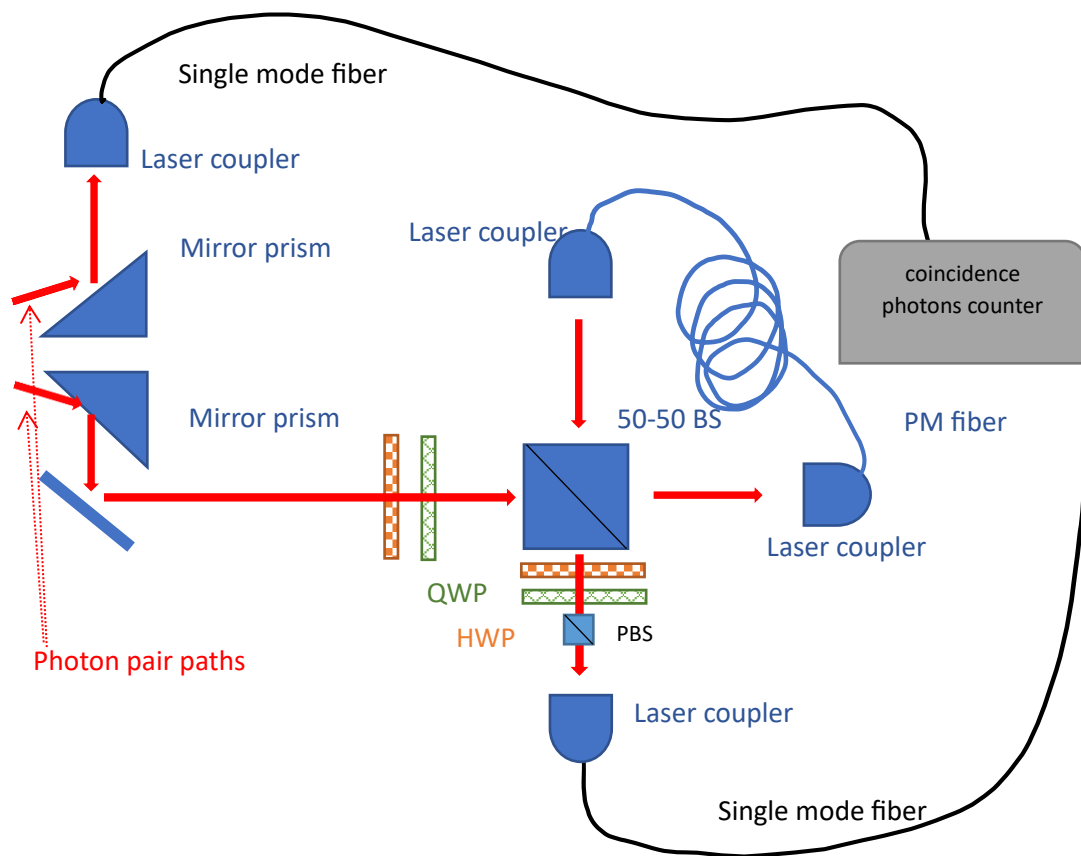


Fig. 4.6.1 Experiment setup for creating a photon loop using a 50/50 beam splitter

4.7 Improve loop setup with lens and reduce laser pulse frequency

The collimated lens is added behind the BBO crystal in experiment five, as shown in Fig. 4.7.1, to reduce the loss over long distances for the photon beam in the B path. And the laser pulse frequency is reduced to 2.5 MHz to improve visibility in the delay histogram. Then the photon counter can record the result of the counter port and time bin trigger, which can be calculated to delay histogram and analyze the photon loop visibility afterward.

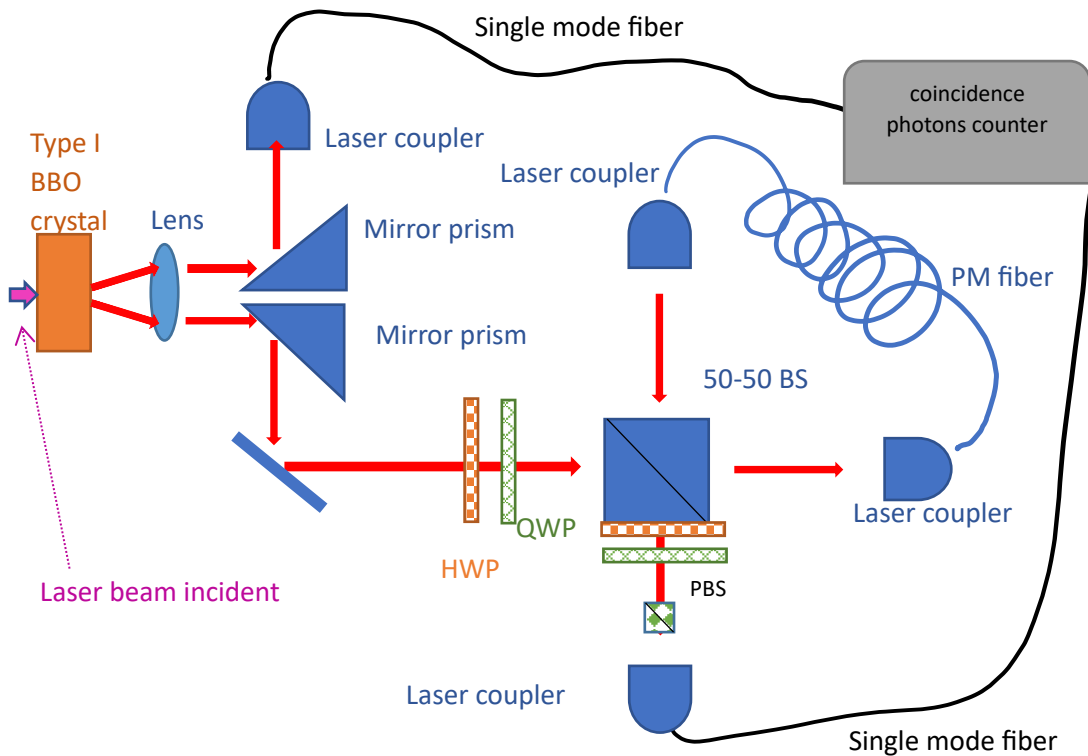


Fig 4.7.1 Experiment setup for creating a photon loop using a 50/50 beam splitter

CHAPTER 5

Result and discussion

5.1 Photon pairs from type-I BBO using a CW laser pump

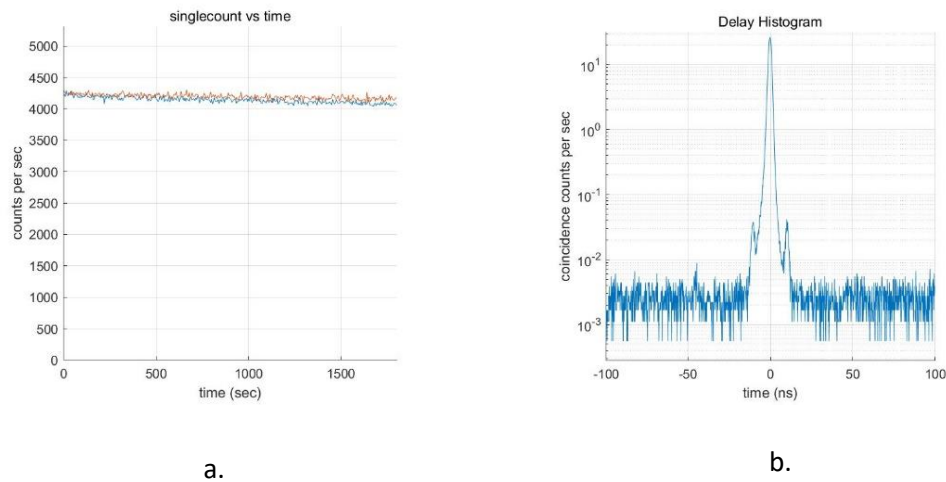


Fig. 5.1.1 (a). single count rate on ports A' and B' as a function of time. (b). delay histogram showing the number of counts triggered in port B' after a varied time delay from when port A' gets triggered

From Fig. 5.1.1, the result shows the delay histogram between two ports with a FWHM of the histogram peak around 3 nanoseconds. There are two noise peaks that appeared in the histogram that might occur if one port of the counter gets triggered by a laser source instead of the photon pairs. The average photon counts per second from ports 1 and 2 are 4,140 and 4,194, respectively, according to the single count graph. Using a window time of 6 nanoseconds, a coincident count rate of 361.2 counts per second can be calculated from the delay histogram.

Eq. 3.5.1 can be used to calculate the accidental coincident count. Using the window time of 6 nanoseconds, its value is 0.1042 count per second, which, when compared to the actual coincident count, allows us to calculate a signal-to-noise ratio of 3,468. The results show that the SPDC process successfully generates photon pairs with a delay in nanosecond order between pairs when using this experimental setup, and the number of pairs generated is noticeably higher than the number of non-pair photons.

Additionally, the accidental counts occurred consistently and appear as the baseline in the histogram plot at around 2×10^{-3} counts per time bin per second. Which is a normal phenomenon considering that with continuous laser the photons are generated continuously and can trigger the counter at any time.

5.2 Hong-Oa Mandel dip

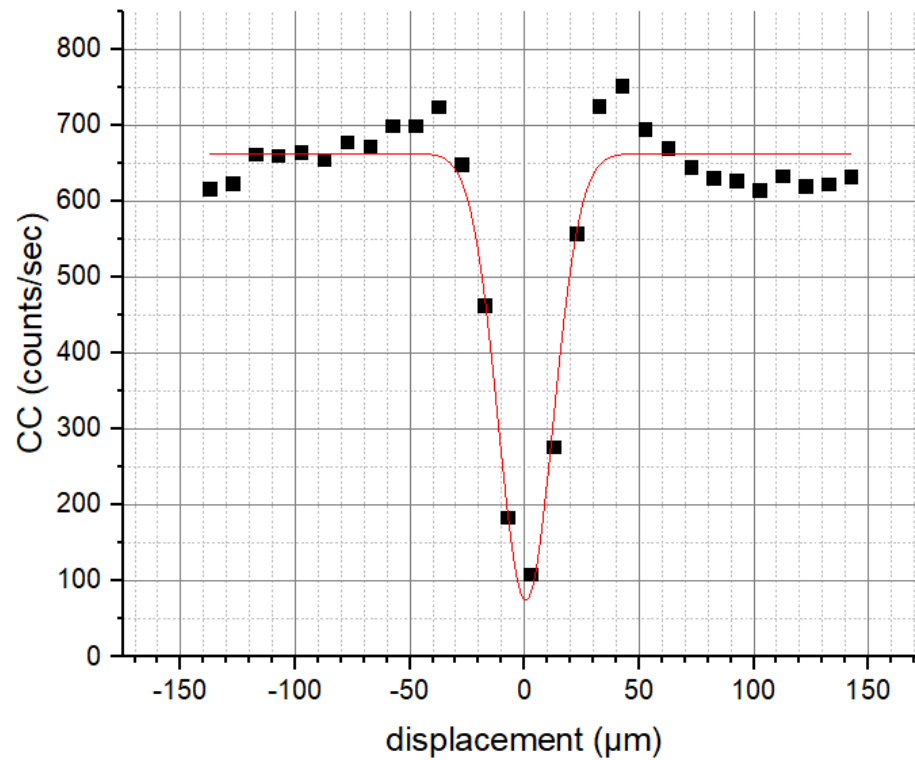


Fig 5.2.1 Measurement result and fitting plot of the HOM dip experiment

Fig. 5.2.1 shows the result from the Hon-Ou-Mandel experiment. The dots represent a coincident count per second at various translation stage displacements on one side. The red line represents the fitted graph by equation Eq. 3.2.4. Here the displacement values are shifted to zero at the lowest fitted value of coincident count to simplify the data. The fitted constant values are shown in Table. 5.2.1, and the visibility can be calculated from Eq. 3.2.5 to be about 0.89.

constant	value
C	1325 cps
T	0.51
R	0.49
w	0.060 μm
d_0	0.55 μm

Table. 5.2.1 Result values from plot fitting of the HOM dip experiment

From the measurement, there are peaks on both sides of the dip, which are possibly the result of the rectangular spectrum bandpass filter used in the setup. However, the overall result shows that there are interferences between the photon pairs generated and propagated in two perpendicular paths. And the setup can be improved for more complex experiments requiring the interference process.

5.3 CHSH experiment

HWP1	HWP2	cc net (cps)	cc std (cps)
-22.5	-12	150	14
	33	60	7
22.5	-12	49	8
	33	136	13
-22.5	12.5	44	8
	57.5	156	12
22.5	12.5	153	15
	57.5	35	7
0	-12	24	5
	33	166	15
45	-12	163	13
	33	26	5
0	12.5	43	7
	57.5	141	14
45	12.5	149	13
	57.5	43	7

Table. 5.3.1 Coincident count rate and its standard deviation for each projective measurement in the CHSH experiment

The result of the CHSH experiment is shown in Table. 5.3.1, where the first two columns are the angles of the HWPs used on paths A' and B' respectively. And the third and fourth columns are the coincident count per second and the standard deviation of its value. From Eqs. 3.3.1 and 3.3.2, E and S values can be calculated using the first four rows as a setup for E(a,b), the next four rows are for E(a,b'), the ninth to the twelfth rows are for E(a',b), and the last four rows are for E(a',b').

	avg	std
E(a,b)	0.45	0.06
E(a,b')	-0.59	0.06
E(a',b)	-0.74	0.07
E(a',b')	-0.54	0.07
S	2.32	0.13

Table. 5.3.2 Calculated E and S values in the CHSH experiment
--

The calculated E and S values are shown in Table 5.3.2. The S value is 2.3 ± 0.2 , which is greater than 2 and therefore violates the inequality, meaning that the experimental setup cannot be described by classical mechanics or by local hidden-variable theories.

5.4 Density matrix experiment

From the experiment setup, the coincident count per second on each basis measurement is shown in Table 5.4.1, where the first and second letter in the first column refer to the basis measured in path A' and path B', respectively. By using Eq. 3.4.13, the density matrix of the polarization state can be estimated numerically (Appendix C) and shown in Table. 5.4.2 and Fig. 5.4.1.

Basis measurement	Coincident count per second
HH	2.13
HV	180.30
VV	1.52
VH	173.03
RH	97.87
RV	73.84
DV	78.09
DH	96.79
DR	60.25
DD	157.17
RD	109.01
HD	93.34
VD	73.59
VL	69.47
HL	67.95
RL	16.03

Table. 5.4.1 Coincidence count rate for each basis measurement in the density matrix tomography experiment

$ \phi\rangle \backslash \langle\phi $	HH	HV	VH	VV
HH	0.01	0.01	0.01	0.00
HV	0.01	0.49	0.37	-0.03
VH	0.01	0.37	0.49	-0.03
VV	0.00	-0.03	-0.03	0.01

a.

$ \phi\rangle \backslash \langle\phi $	HH	HV	VH	VV
HH	0.00	0.04	0.02	0.00
HV	-0.04	0.00	0.16	-0.01
VH	-0.02	-0.16	0.00	0.01
VV	0.00	0.01	-0.01	0.00

b.

Table. 5.4.2 Computed physical density matrix in (a) real part and (b) imaginary part

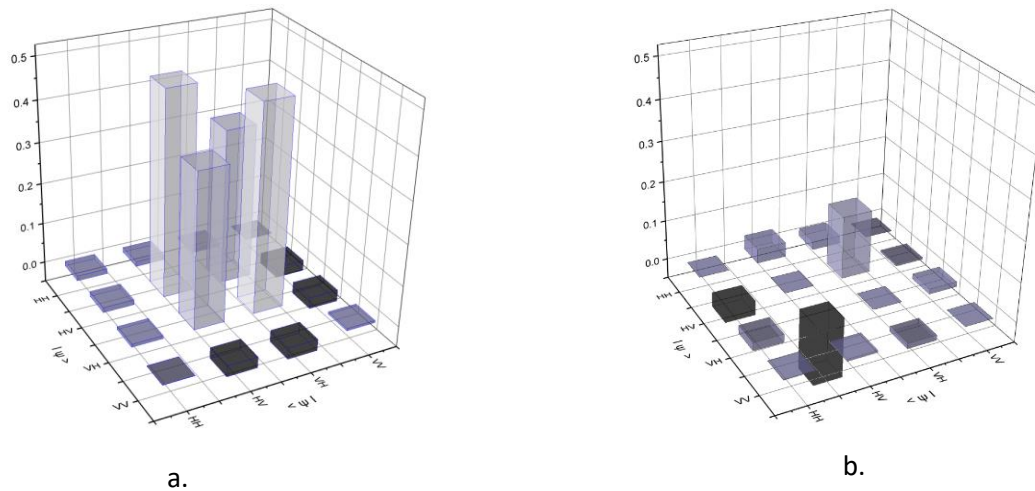


Fig. 5.4.1 Real part (a) and imaginary part (b) of the density matrix plotted in 3D columns

The result shows that the polarization state is close to the Bell state ψ^+ in this experiment setup with a fidelity of 0.8685. From Eq. 3.4.13 to Eq. 3.4.19, the derived quantities are $S = 0.431$, $P = 0.2134$, $C = 0.7449$, $T = 0.5549$, and $E = 0.6494$. S and P values show that the polarization state has some degree of mixture and is not completely in a pure state. Whereas the last three values show that the entanglement state is successfully produced in this experimental setup.

5.5 Photon pairs from type-I BBO using a 80MHz pulse laser pump

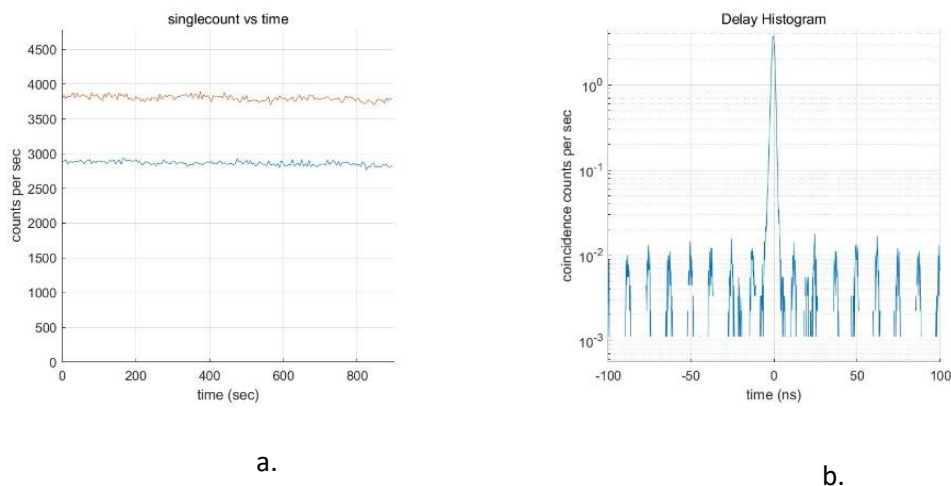


Fig. 5.5.1 The results of an experiment using a pulse laser as a photon pair source. (a) single count rate on ports A' (bottom line) and B' (top line). (b). Delay histogram between two ports.

Fig. 5.5.1 shows the result of the experimental setup for photon pairs generated by using a pulse laser pump into a type-I BBO. From the single count graph, the average photon counts per second in ports 1 and 2 are 2,863 and 3,800, respectively. The delay histogram shows a coincident peak with FWHM around 3 nanoseconds, and the coincident count can be calculated at 51.24 counts per second with a window time of 6 nanoseconds. Using Eq. 3.5.2, the accidental coincident is 0.136, which yields the signal-to-noise ratio at 376.8.

From the delay histogram, there are repeating noise peaks every 12.5 nanoseconds. These peaks are accidental coincidences where two ports are triggered by two different pulses instead of the same pulse that generated pairs of photons. Additionally, there are peaks at -20 and +20 nanoseconds that might be caused by errors in system signal processing or the detector deadtime.

From the result, the photon pairs generated by using pulse lasers have a lower signal-to-noise ratio compared to the experiment using a continuous laser source. The main reason here is the difference in laser power that the pump can generate. However, by using a pulse laser, the delay histogram can show better visibility and be easier to distinguish between a real coincidence count and an accidental coincident count peak. This shows that the pulse source is a better choice in the experiment where the delay peaks need to be analyzed carefully or there are low signal counts that need to be examined and the baseline noise is undesirable.

5.6 Loop using a 50-50 Beam splitter

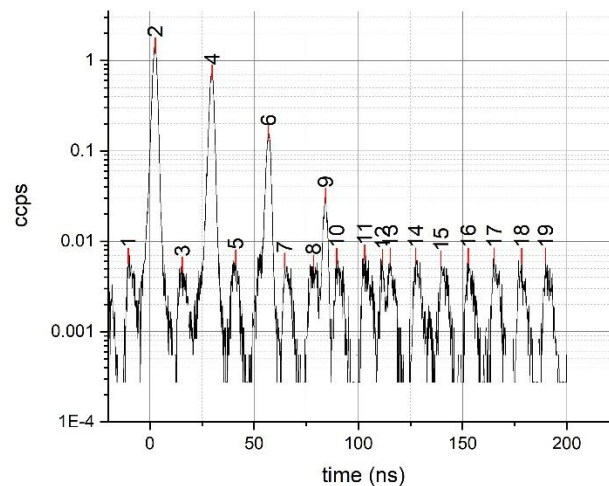


Fig 5.6.1 A delay histogram between two ports in the experiment that guides one path of photon pairs into an optical loop

Peak Index	Center Max(ns)	Max Height (cps)	FWHM (nm)
1	-9.43	3.5E-03	2.03
2	2.43	1.5E+00	2.08
3	15.31	2.1E-03	2.37
4	29.66	7.2E-01	2.11
5	40.62	3.1E-03	2.00
6	56.92	1.5E-01	2.12
7	65.63	2.6E-03	1.90
8	78.76	2.5E-03	2.74
9	84.14	2.7E-02	1.86
10	89.59	4.4E-03	0.28
11	103.01	5.4E-03	0.22
12	111.44	3.4E-03	1.31
13	115.28	3.1E-03	1.68
14	128.07	3.4E-03	2.01
15	140.45	2.8E-03	1.92
16	152.93	3.0E-03	1.65
17	165.48	3.2E-03	1.78
18	178.34	4.7E-03	0.25
19	190.22	2.6E-03	1.55

Table. 5.6.1 Peak indices and their properties from the histogram in Fig. 5.6.1

Fig. 5.6.1 and Table 5.6.1 show the result delay histogram from photon loop setup and peak properties for each peak index. The accidental peak noises repeat every 12.5 nanoseconds with a maximum height of around 0.003 counts per second and can be easily recognized from their time delay. Peak index 2,4,6 and 9 can be identified as signals with the first one accounting for photon that trigger without entering the fiber loop, and those after are for 1,2 and 3 times in the loop respectively. The histogram peak, which accounted for the fourth and higher loop, is not distinguishable from other noise peaks due to its low signal and overlap between peaks.

From the result, it appears that each loop delays the photon signal time by around 27 nanoseconds and lowers the peak height by about 80%, compared to 50% in theory. The large power loss could be a result of multiple reasons, such as loss in fiber, imperfect mirrors, an imperfect beam splitter, and inaccurate alignment.

5.7 Improve loop setup with lens and reduce laser pulse frequency

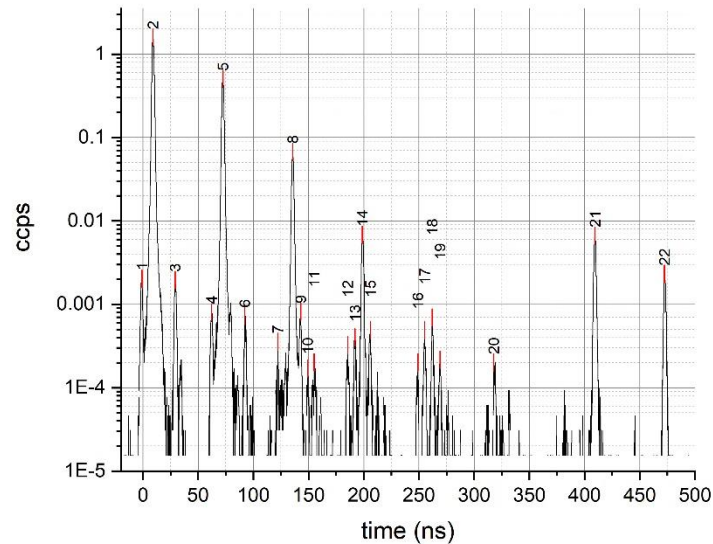


Fig. 5.7.1 Delay histogram between two ports in the improved experiment with lens

Peak Index	Center Max (ns)	Max Height (cps)	FWHM (ns)
2	9.13	1.6E+00	1.92
5	72.39	5.0E-01	1.96
8	135.66	6.4E-02	1.95
14	198.91	6.5E-03	1.94
18	262.03	2.8E-04	0.69
20	317.67	1.2E-04	0.04
21	409.36	6.4E-03	1.83

Table. 5.7.1 Peak indices and their properties from the histogram in Fig. 5.7.1

By lowering the pulse frequency, accidental coincident peaks are separated with a longer time delay and the signals can be analyzed more accurately, but in return, the laser power is lower and results in small signal counts. Therefore, in this setup, lens is added to help with photon pair beam alignment and focusing. The resulting delay histogram from the experimental setup is shown in Fig. 5.7.1, and the peak properties from the graph are shown in Table 5.7.1. With a 2.5 MHz pulse laser, the accidental coincident peaks repeat every 400 nanoseconds and appear only once at peak number 21 in the graph. Here peak numbers 5, 8, 14, 18, and 20 are accounted for signal from photons traveling in loop fiber 1, 2, 3, 4, and 5 times before entering the counter, respectively. Peak 2 shows the signal where photons trigger the counter without entering a loop.

The results show that each time photons enter the fiber loop, a delay of about 63 nanoseconds is added, with a 90% reduction in peak height. The larger power loss might come from the longer fiber used in the experiment or from more inaccurate alignment. Nonetheless,

the histogram peak can be analyzed up to the fifth loop, which is a huge improvement from the previous experiment setup.

There are some noise peaks around the main peak; these can be caused by a variety of factors but were most likely caused by the additional path that some photon pairs can take to add constant time to their delay before entering the main path and triggering the counter. For example, if one of the pair reflected back and forth inside the 5-meter fiber before entering the same path, it will add approximately 20 nanoseconds to the delay histogram and shift the peak to the left or right side depending on which port that photon triggers.

CHAPTER 6

An alternative scheme of universal optical programmable multi-qubit gates for polarization qubits

6.1 Hilbert-space expansion technique for two polarization qubits

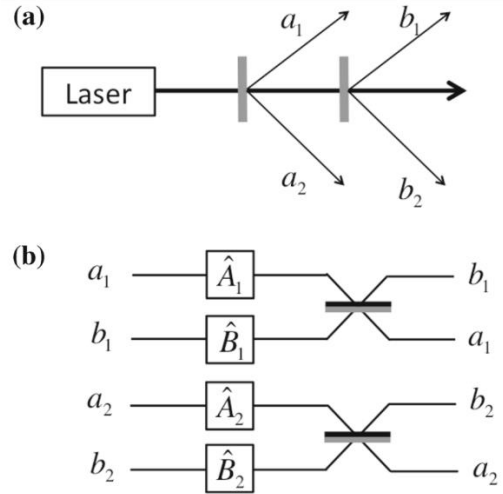
Universal quantum computers need single-qubit and two-qubit gates realized by particular physical systems [5]. The single-qubit gate of the polarization state of light is simply given by a cascade of birefringent materials, such as half-waveplates (HWP) and quarter-waveplates (QWP). Peters et al. [29] demonstrated that single-qubit gates comprising two HWPs and a QWP had great versatility, yielding all possible unitary single-qubit gates. In addition, non-unitary single-qubit gates were also possible with the use of an interferometer to introduce decoherence into the optical circuit. Two-qubit gates are also of primary interest, as they are far more difficult to achieve with photon qubits. Here, we develop optical decomposition to realize any possible two-qubit gate. First of all, any two-qubit gate can be expressed as a linear combination of tensor products of single-qubit gates, which can be written as $\sum_i \hat{O}_{1i} \otimes \hat{O}_{2i}$. Using four separate single-qubit gates $\hat{Y}_{00}, \hat{Y}_{01}, \hat{Y}_{10}, \hat{Y}_{11}$, which were realized experimentally in [29], a two-qubit gate can be mathematically decomposed as

$$|0\rangle\langle 0| \otimes \hat{Y}_{00} + |0\rangle\langle 1| \otimes \hat{Y}_{01} + |1\rangle\langle 0| \otimes \hat{Y}_{10} + |1\rangle\langle 1| \otimes \hat{Y}_{11} \quad 6.1.1$$

which implies that once programmable single qubit gates are realized, programmable two-qubit gates will be as well.

Fig. 6.1.1 (a)
SPDC source to produce two pairs of entangled photons resulting as Eq. 6.1.2

(b) Experimental scheme to realize the two-qubit operation $\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2$ according to the input state (Eq. 6.1.3), which is derived from the source (Eq. 6.1.2)



To realize such two-qubit gates using photons, we started from previous works that dealt with the linear combination of quantum operators [32, 38]. The approach requires two pairs of type-I spontaneous parametric down-conversion (SPDC) processes (Fig. 6.1.1a) and two symmetrical beam splitters (BS) to merge four spatial modes of photons resulting from the SPDCs (Fig. 6.1.1b). Once a laser beam is pumped to two nonlinear crystals, as presented in Fig. 6.1.1a, two pairs of identical photons are produced, whose states can be written in the polarization states as

$$\frac{1}{\sqrt{2}} (|\psi\rangle_{a_1} |\psi\rangle_{a_2} + |\psi'\rangle_{b_1} |\psi'\rangle_{b_2}) \quad 6.1.2$$

where $a_1, a_2, b_1,$ and b_2 denote the spatial paths of the photons, and ψ and ψ' are the polarization states of each photon pair. After the SPDCs, by priorly knowing of the possible states from SPDCs, we can transform the output states to

$$\frac{1}{\sqrt{2}}(|\psi_1\rangle_{a_1}|\psi_2\rangle_{a_2} + |\psi_1\rangle_{b_1}|\psi_2\rangle_{b_2}) \quad 6.1.3$$

using waveplates. The polarization states ψ_1 on the paths a_1 and b_1 are set identically and constitute the states of the first qubit. Similarly, the states ψ_2 on the paths a_2 and b_2 constitute the states of the second qubit. Afterward, the photon on each spatial path is operated by the single-qubit gates $\hat{A}_1, \hat{A}_2, \hat{B}_1, \hat{B}_2,$ and then the paths are merged by two BSs. Since each state is actually the hyper-entanglement of paths and polarization qubits, a beam splitter produces separate paths of photons but conserves their polarization. Consequently, two BSs (Fig. 6.1.1b) transform the states in Eq. 6.1.3 as follows:

$$|\psi_1\rangle_{a_1} \rightarrow \frac{1}{\sqrt{2}}(\hat{A}_1|\psi_1\rangle_{a_1} + \hat{A}_1|\psi_1\rangle_{b_1}) \quad 6.1.4$$

$$|\psi_1\rangle_{b_1} \rightarrow \frac{1}{\sqrt{2}}(\hat{B}_1|\psi_1\rangle_{a_1} - \hat{B}_1|\psi_1\rangle_{b_1}) \quad 6.1.5$$

$$|\psi_2\rangle_{a_2} \rightarrow \frac{1}{\sqrt{2}}(\hat{A}_2|\psi_2\rangle_{a_2} + \hat{A}_2|\psi_2\rangle_{b_2}) \quad 6.1.6$$

$$|\psi_2\rangle_{b_2} \rightarrow \frac{1}{\sqrt{2}}(\hat{B}_2|\psi_2\rangle_{a_2} - \hat{B}_2|\psi_2\rangle_{b_2}) \quad 6.1.7$$

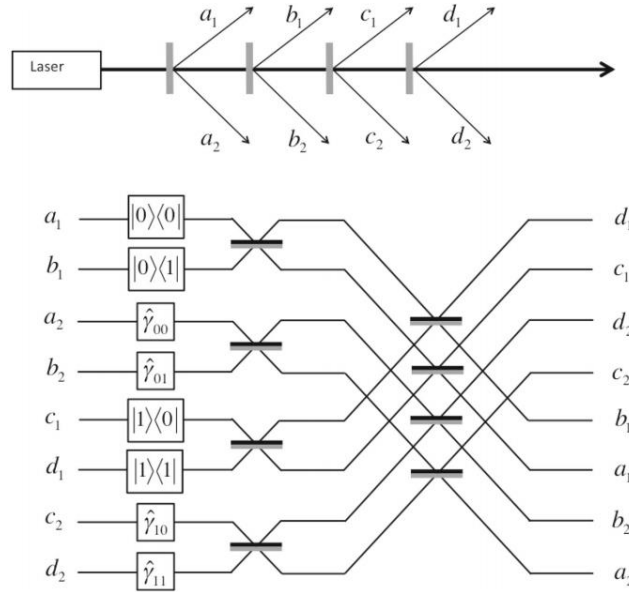


Fig. 6.1.2 (a) A nonlinear single-photon source gives the superposition of photon states (Eq. 6.1.10). **(b)** A linear optical circuit to perform the linear combination of four tensor products of single-qubit gates in Eq. 6.1.12. The photons in the circuit are delivered from eight paths in Fig. 6.1.2a. Before entering the circuit, all photons are initialized as the superposition of the identical input polarization states on different spatial paths in Eq. 6.1.10

As a result of post-selecting photons in the paths a_1, a_2 or $b_1, b_2,$ the final state of photons becomes

$$\frac{1}{\sqrt{2}}(\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2)|\psi_1\rangle|\psi_2\rangle \quad 6.1.8$$

The quantum operator $\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2$ is realized with a probability of $\frac{1}{4}$

In order to achieve Eq. 6.1.1, the construct of four SPDCs, eight spatial paths, and eight BSs is presented in Fig. 6.1.2. The circuit is indeed a scaled-up version of Fig. 6.1.1. Similar to Eq. 6.1.2, we can expect the state

$$\frac{1}{2}(|\psi_a\rangle_{a_1}|\psi_a\rangle_{a_2} + |\psi_b\rangle_{b_1}|\psi_b\rangle_{b_2} + |\psi_c\rangle_{c_1}|\psi_c\rangle_{c_2} + |\psi_d\rangle_{d_1}|\psi_d\rangle_{d_2}) \quad 6.1.9$$

from four SPDCs (Fig. 6.1.2a). If we assume that the polarization states in (Eq. 6.1.9) are known, we can generate an output state for the two qubits before entering the circuit of Fig. 6.1.2b as

$$\frac{1}{2}(|\psi_1\rangle_{a_1}|\psi_2\rangle_{a_2} + |\psi_1\rangle_{b_1}|\psi_2\rangle_{b_2} + |\psi_1\rangle_{c_1}|\psi_2\rangle_{c_2} + |\psi_1\rangle_{d_1}|\psi_2\rangle_{d_2}) \quad 6.1.10$$

In the first stage of the circuit, by using $\hat{A}_1 = |0\rangle\langle 0|$, $\hat{B}_1 = |0\rangle\langle 1|$, $\hat{C}_1 = |1\rangle\langle 0|$, $\hat{D}_1 = |1\rangle\langle 1|$, $\hat{A}_2 = \hat{\gamma}_{00}$, $\hat{B}_2 = \hat{\gamma}_{01}$, $\hat{C}_2 = \hat{\gamma}_{10}$, $\hat{D}_2 = \hat{\gamma}_{11}$, functioning as the programmable apparatus, (Eq. 6.1.10) leads to

$$\frac{1}{2}[(|0\rangle\langle 0||\psi_1\rangle_{a_1}) \otimes (\hat{\gamma}_{00}|\psi_2\rangle_{a_2}) + (|0\rangle\langle 1||\psi_1\rangle_{b_1}) \otimes (\hat{\gamma}_{01}|\psi_2\rangle_{b_2}) + (|1\rangle\langle 0||\psi_1\rangle_{c_1}) \otimes (\hat{\gamma}_{10}|\psi_2\rangle_{c_2}) + (|1\rangle\langle 1||\psi_1\rangle_{d_1}) \otimes (\hat{\gamma}_{11}|\psi_2\rangle_{d_2})] \quad 6.1.11$$

Finally, using the relation between inputs and outputs expressed in Eqs. 6.1.4 - 6.1.7 together with the post-selection measurement of photon coincidences in the paths (a_1, a_2) , (b_1, b_2) , (c_1, c_2) or (d_1, d_2) , the final state becomes

$$\frac{1}{2}(|0\rangle\langle 0| \otimes \hat{\gamma}_{00} + |0\rangle\langle 1| \otimes \hat{\gamma}_{01} + |1\rangle\langle 0| \otimes \hat{\gamma}_{10} + |1\rangle\langle 1| \otimes \hat{\gamma}_{11}) \quad 6.1.12$$

with a probability of 1/4. In particular, this yields a two-qubit operation, namely

$$\begin{bmatrix} \hat{\gamma}_{00} & \hat{\gamma}_{01} \\ \hat{\gamma}_{10} & \hat{\gamma}_{11} \end{bmatrix} |\psi_1\rangle|\psi_2\rangle \quad 6.1.13$$

where all the entries of the 4×4 matrix are programmable

6.2 Universal optical programmable multi-qubit gates

The recent experimental creation of three photon polarization GHZ states from SPDC [17] suggests that we can expect to access more photonic quantum processing with polarization-entangled photons in the coming years. Apart from other sophisticated approaches to photon entanglement, SPDC from nonlinear crystals in bulk optics setups, as presented here, can play a role as the simplest possible source of entangled photons. Suppose that the n -photon entangled state

$$\frac{1}{\sqrt{2}}(|\psi_1\rangle_{o_{11}}|\psi_2\rangle_{o_{12}} \cdots |\psi_n\rangle_{o_{1n}} + |\psi_1\rangle_{o_{21}}|\psi_2\rangle_{o_{22}} \cdots |\psi_n\rangle_{o_{2n}}) \quad 6.2.1$$

is accessible from the source depicted in Fig. 6.2.1 and that the n -photon coincidence counting operates with high efficiency, our architecture is able to accommodate programmable n -qubit gates. The scheme in Fig. 6.2.1 is actually the n -photon version of the paired-photon source in Fig. 6.1.1a, in which the paths a_1 , a_2 , b_1 , and b_2 are replaced, respectively, by O_{11} , O_{12} , O_{21} , and O_{22} for generalization; and ES represents the entanglement source. For clarity in notation, two indices, say m and p of the path O_{mp} , indicate photons from the p^{th} source (ES_p) and the m^{th} qubit, which will be later encoded into the polarization state $|\psi_m\rangle$.

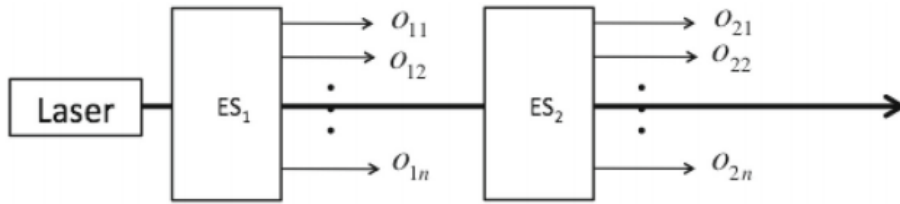


Fig. 6.2.1 Two coupled notional single-photon sources (ES_1 and ES_2) provide n entangled photons. The sources give the polarization states as Eq. 6.2.1

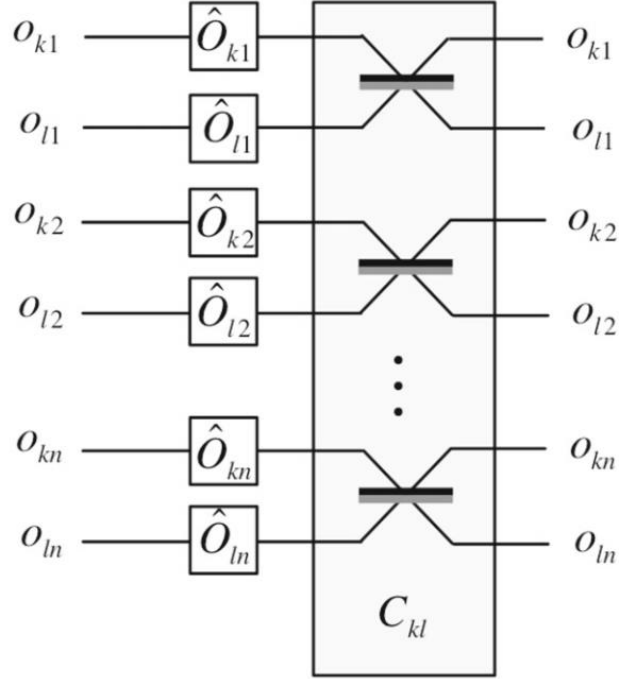


Fig. 6.2.2 A combination block (C_{kl}), a series of beam splitters combining photons from ES_k and ES_l , supposing each ES produces n distinguishable single photons

In order to generalize all operations in the previous section to the n -qubit case, we introduce a combination block (C_{kl}); see Fig. 6.2.2. Here, the C_{kl} block consists of BSs, where each BS fuses the two paths sharing the second index, e.g., from sources ES_k and ES_l in this case. When the single-qubit gate O_{mp} operates on each of its corresponding paths O_{mp} , the C_{kl} block transforms the initial state

$$\frac{1}{\sqrt{2}} (|\psi_1\rangle_{o_{k1}} |\psi_2\rangle_{o_{k2}} \cdots |\psi_n\rangle_{o_{kn}} + |\psi_1\rangle_{o_{l1}} |\psi_2\rangle_{o_{l2}} \cdots |\psi_n\rangle_{o_{ln}}) \quad 6.2.2$$

to

$$\begin{aligned} \frac{1}{(\sqrt{2})^{n+1}} [& (\hat{O}_{k1} |\psi_1\rangle_{o_{k1}} + \hat{O}_{k1} |\psi_1\rangle_{o_{l1}}) \otimes (\hat{O}_{k2} |\psi_2\rangle_{o_{k2}} + \hat{O}_{k2} |\psi_2\rangle_{o_{l2}}) \otimes \cdots \otimes (\hat{O}_{kn} |\psi_n\rangle_{o_{kn}} + \\ & \hat{O}_{kn} |\psi_n\rangle_{o_{ln}}) + (\hat{O}_{l1} |\psi_1\rangle_{o_{k1}} - \hat{O}_{l1} |\psi_1\rangle_{o_{l1}}) \otimes (\hat{O}_{l2} |\psi_2\rangle_{o_{k2}} - \hat{O}_{l2} |\psi_2\rangle_{o_{l2}}) \otimes \cdots \otimes \\ & (\hat{O}_{ln} |\psi_n\rangle_{o_{kn}} - \hat{O}_{ln} |\psi_n\rangle_{o_{ln}})] \quad 6.2.3 \end{aligned}$$

Thereafter, by post-selecting the case where only one photon leaves each path of O_{kl} - O_{kn} simultaneously, we will obtain the final state

$$(\hat{O}_{k1} \otimes \hat{O}_{k2} \otimes \cdots \otimes \hat{O}_{kn} + \hat{O}_{l1} \otimes \hat{O}_{l2} \otimes \cdots \otimes \hat{O}_{ln}) \times |\psi_1\rangle_{o_{k1}} |\psi_2\rangle_{o_{k2}} \cdots |\psi_n\rangle_{o_{kn}} \quad 6.2.4$$

with a success probability of $(1/\sqrt{2})^{n+1}$. Likewise, by post-selection on all l -paths, we obtain

$$(\hat{O}_{k_1} \otimes \hat{O}_{k_2} \otimes \cdots \otimes \hat{O}_{k_n} - \hat{O}_{l_1} \otimes \hat{O}_{l_2} \otimes \cdots \otimes \hat{O}_{l_n}) \times |\psi_1\rangle_{o_{l_1}} |\psi_2\rangle_{o_{l_2}} \cdots |\psi_n\rangle_{o_{l_n}} \quad 6.2.5$$

with a success probability of $(1/\sqrt{2})^{n+1}$. The realization of Eq. 6.2.4 is a cornerstone of the complete realization of the n -qubit programmable operation.

First of all, any n -qubit operation is represented by a $2^n \times 2^n$ matrix and can be decomposed into a linear combination of tensor products written as

$$\sum_{i=1}^{2^{n-1}} \sum_{j=1}^{2^{n-1}} \hat{\gamma}(i,j) \otimes |i\rangle\langle j| \quad 6.2.6$$

where $\hat{\gamma}(i,j)$ is a 2×2 operator as defined in Eq. 6.1.1, and all of its matrix elements are programmed by the programmable single-qubit gates. Here, $|i\rangle\langle j|$ represents a basis for each matrix element of the $2^{n-1} \times 2^{n-1}$ all-ones matrix.

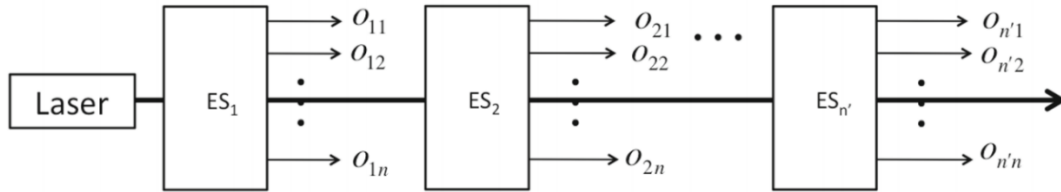
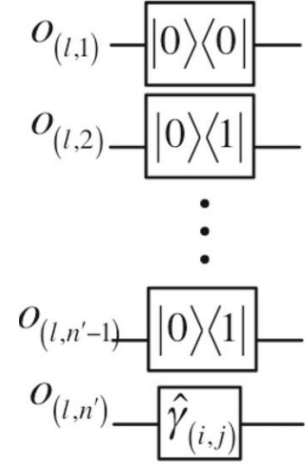


Fig. 6.2.3 A cascade of n ESs, where each ES produces n entangled single photons in separate paths, resulting the initial state as expressed in Eq. 6.2.7

As the initial state in Eq. 6.2.1 from the source in Fig. 6.2.1 is necessary to realize the summation of two terms of tensor products of n single-qubit gates, we will, at least, need the kind of source illustrated in Fig. 6.2.3 to complete the summation of $n' = (2^{n-1})^2$ terms following Eq. 6.2.6. The quantum state is then written as

$$\frac{1}{\sqrt{n'}} \sum_{k=1}^{n'} \bigotimes_{l=1}^n |\psi_l\rangle_{o_{kl}} \quad 6.2.7$$

Fig. 6.2.4 The example of the local operations applied to the photon state after being initialized to the state in Eq. 6.2.7 on each ES output (ES_i for this example). This example represents just one term in Eq. 6.2.6. In order to assign the position (i, j) of the single-qubit operation $\hat{\gamma}(i, j)$ on the $(2^{n-1}) \times (2^{n-1})$ all-ones matrix, the local operations representing the position are applied on the paths $O_{(l,1)}, O_{(l,2)}, \dots, O_{(l,n'-1)}$. In this example, the operators relating to entries on the all-ones matrix follow from Eq. 6.2.8



From this stage of the scheme, photons from each ES play a role as each term in Eq. 6.2.6 arising from applying the local operator (Fig. 6.2.4). The local operators applied on paths $O_{l,1}, O_{l,2}, \dots, O_{l,n-1}$ are the members of $|0\rangle\langle 0|, |0\rangle\langle 1|, |1\rangle\langle 0|, |1\rangle\langle 1|$. A tensor product of $n - 1$ terms from this set of operators results in a $(2^{n-1}) \times (2^{n-1})$ matrix with all zero entries except the element (i, j) . The relation between the tensor product and (i, j) follows the relation

$$|0\rangle\langle 1| \otimes |1\rangle\langle 0| \otimes \dots \otimes |1\rangle\langle 0| \equiv |01 \dots 1\rangle\langle 10 \dots 0| = |i\rangle\langle j| \quad 6.2.8$$

where i and j correspond to decimal equivalents of the binary representations $01 \dots 1$ and $10 \dots 0$, respectively. The summation of two separate terms building from two ES s and the state preparation schemes in Fig. 6.2.4 is presented in Fig. 6.2.5. Finally, using the combination blocks to merge all paths (Fig. 6.2.6) and post-selecting only photons in path $O''(1, n)$, the operation in Eq. 6.2.6 is realized, obtaining the n -qubit operation represented by the matrix

$$\begin{bmatrix} \hat{\gamma}_{(1,1)} & \hat{\gamma}_{(1,2)} & \hat{\gamma}_{(1,3)} & \dots & \hat{\gamma}_{(1,2^{n-1})} \\ \hat{\gamma}_{(2,1)} & \hat{\gamma}_{(2,2)} & \hat{\gamma}_{(2,3)} & \dots & \hat{\gamma}_{(2,2^{n-1})} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{\gamma}_{(2^{n-1},1)} & \hat{\gamma}_{(2^{n-1},2)} & \hat{\gamma}_{(2^{n-1},3)} & \dots & \hat{\gamma}_{(2^{n-1},2^{n-1})} \end{bmatrix} \quad 6.2.9$$

with a success probability of $(1/\sqrt{2})^n$

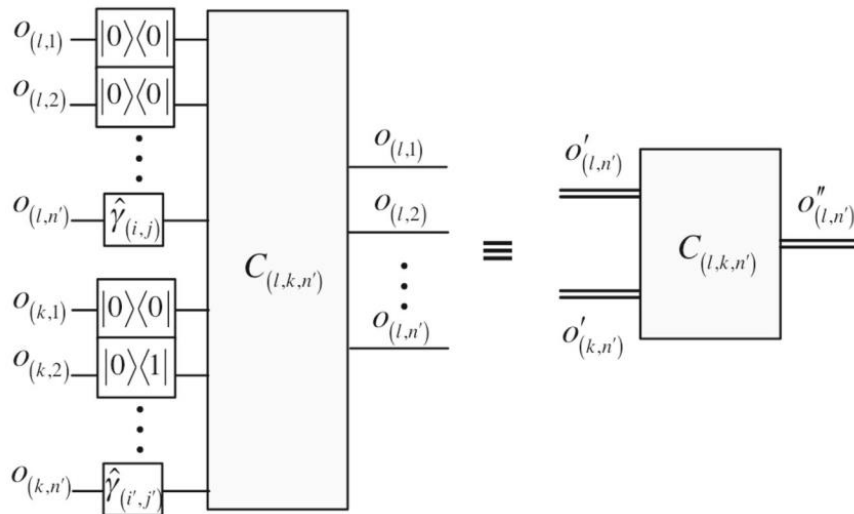


Fig. 6.2.5 A modified combination block. Paths $O_{(p,m)}$ sharing p are grouped as a composite path $o'_{(p,n')}$ with a set of local operations in Fig. 6.2.4, where $m = 1, 2, \dots, n'$. In this case, the composite paths $o'_{l,n'}$ and $o'_{k,n'}$ are combined by the modified combination block $C_{l,k,n'}$, resulting in the composite path $o''_{l,n'}$. This output composite path is the post-selection of the photons leaving at the output paths $O_{(l,1)}, O_{(l,2)}, \dots, O_{(l,n-1)}$

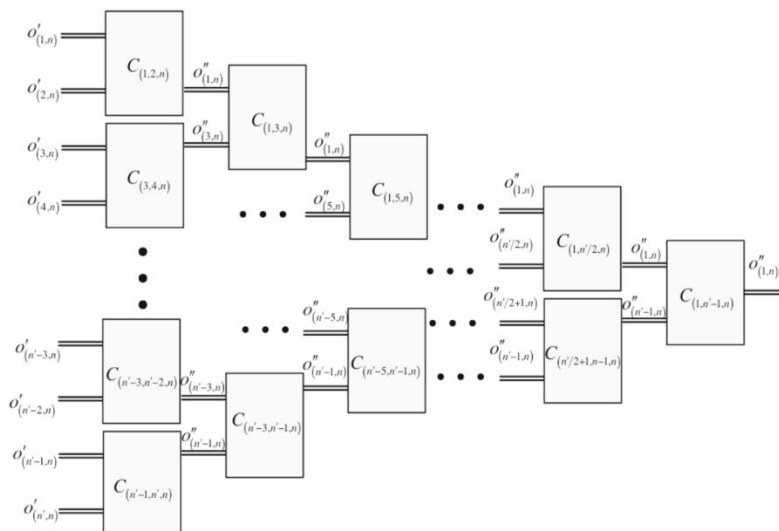


Fig. 6.2.6 The application of the modified combination blocks to combine entire composite paths $o'_{1,n}, o'_{2,n}, \dots, o'_{n'-1,n}, o'_{n',n}$, resulting in the post-selection of only one composite path as the output

6.3 Discussion

We have presented an alternative optical circuit design that can achieve complete versatility in a programmable multi-qubit gate. By directly programming the entries of the matrix with a single-qubit gate, our proposed scheme circumvents the need to decompose the required gates into elementary single- and two-qubit gates in the standard method of realizing any unitary quantum operator. Accordingly, this alternative approach has the potential to accelerate quantum simulation and information processing because it does not require iteration as does the standard method. Due to the increased feasibility of controlling every entry of the matrix independently by linear optics, it also provides great capacity to generate whole sets of operations, including complex and non-unitary operations.

However, the demonstrated quantum gate based on this scheme is probabilistic and will undergo an exponentially small probability of success. In this sense, we can regard this alternative scheme as a way of trading the decoherence for the power of generating any operator. One can argue that ideally the alternative scheme can be used to probe noise or decoherence since the matrix entries are independently programmed and can be complex; in particular, the off-diagonal terms can be probed systematically. This arguably can yield great benefits in decoherence suppression and noise spectroscopy in metrology, and possibly in errors correction in quantum communication. Furthermore, gate-based computation using a small number of qubits is rather possible, and such a class of computation can shed some light on problems in the fundamental concepts of quantum information requiring an arbitrary quantum gate. Examples of such scenarios, such as controlling operation of an unknown $U(2)$, the complete quantum phase estimation algorithm via the semiclassical Fourier transform, entanglement filters, the quantum approximate optimization algorithm, and Szegedy quantum walks, have been demonstrated already [32, 37, 38].

Perhaps, it is more insightful and useful to compare our proposed scheme to the standard method [8, 33], which relies on a decomposition of an $N \times N$ unitary matrix into a product of 2×2 matrices, which can then be determined by using successive 2×2 transformations. Both approaches require linear optical devices to program a multi-qubit gate; the standard method uses phase shifters and beam splitters on two-dimensional subspaces, and iteratively expands to the full N -dimensional Hilbert space, while our scheme uses entanglement photon sources and beam splitters to create all possible linear combinations of tensor products in the entire Hilbert space and employs post-selection to program the operator. To quantitatively analyze the complexity of the two approaches, we firstly note that, in order to realize a $U(2^m)$, for $m \geq 2$, it will need to be factorized into a multiplication of $2^m(2^m - 1)/2$ of $U(4)$:

$$U(2^m) = \left[\prod_{i=1}^{2^m(2^m-1)/2} U_i(4) \right] D \quad 6.3.1$$

where D is a diagonal matrix with elements of modulus 1. There are several classical algorithms to perform this type of factorization. However, preparing any arbitrary $N \times N$ gates could endure exponential growth complexity as a result of this requirement. Compared to our scheme, which is built from the summation of matrices $U(4)$ given in Eq. 6.3.2, the number of required $U(4)$ operations is 4^{m-1} which is just a slight improvement.

$$U(2^m) = \sum_{i=1}^{4^{m-1}} \widehat{U}_i(4) \quad 6.3.2$$

or

equivalently

$$\begin{aligned}
U(2^m) = & \begin{bmatrix} \hat{U}_1(4) & & & \\ \hat{U}_{(2^{m-1}+1)}(4) & & & \\ \vdots & & & \\ \hat{U}_{(4^{m-1}-2^{m-1})}(4) & \hat{U}_{(4^{m-1}-2^{m-1}+1)}(4) & \cdots & \hat{U}_{(4^{m-1})}(4) \end{bmatrix} = \\
& \begin{bmatrix} \hat{U}_1(4) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} + \begin{bmatrix} 0 & \hat{U}_2(4) & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} + \cdots + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{U}_{(4^{m-1})}(4) \end{bmatrix}
\end{aligned}
\tag{6.3.3}$$

However, with no requirement for factorization of the matrix, our scheme can avoid a large growth in complexity. The linear combination is the main strength of gate structure based on X. Qiang et al.'s scheme in Ref. [32], like our scheme. The prepared $U'(4)$ s can be simply placed directly in the circuit as shown in Figs. 6.2.4 and 6.2.5 to realize a whole matrix of $U(2^m)$.

Clearly, there is a trade-off between iteration, which can increase complexity, and post-selection, which yields low probability. For a small number of qubits, both approaches have been realized experimentally with good fidelity [32, 37]. In particular, a similar linear expansion technique for arbitrary two-qubit processing has been demonstrated [32] on a silicon photonics chip that could generate two photonic qubits, on which it could perform arbitrary two-qubit unitary operations, including arbitrary entanglement operations. As emphasized, our proposed scheme circumvents the decomposition issue by directly programming matrix elements of the operator, which can be applied to non-unitary matrices or the operation that is governed by an unknown matrix, such as the phase estimation algorithm. For the latter, there is no operator to decompose; hence, our method can have an advantage over the standard method. Concerning the unknown quantum operation, our work can be regarded as the theoretical extension of the work by Zhou et al. [38], where only 4×4 matrix was concerned. In some sense, our analysis is similar to that of Reck et al. [33] but emphasizes the alternative approach to their decomposition method. To our best knowledge, neither has been realized experimentally for arbitrary $N \times N$ operators.

To further remark on the experimental realization of this architecture, our proposed scheme demands extremely strong nonlinearity and highly efficient single-photon detection. Unfortunately, the demonstration of these entirely programmable two-qubit gates is perhaps beyond the capabilities of existing nonlinear optics technology. However, some previous studies using the smallest possible configuration have proven the advantages of this approach [37, 38]. In fact, photon pairs from the process of spontaneous four-wave mixing in strip waveguides on a photonic chip could also produce polarization-entangled pairs [24]. In addition to polarization-encoded qubits, micro-ring resonators for a path entangled two-qubit state also have their potential. The Bell's state of multi-photon pairs was realized in photonics-integrated circuits [36]. Regardless of relying on the nonlinear optics as the single-photon source, the future potential of a semiconductor quantum dots platform is currently envisaged [1, 10, 14, 19, 25].

Finally, we comment that, like the standard method, this alternative scheme is valid to realize discrete operations. Once all $N \times N$ matrices can be implemented, it is possible to measure the continuous observable. This can be done by truncating Fourier transforms into an arbitrary large but discrete operator. In the context of quantum simulation, this can be very beneficial for time-evolution operators of some physical systems, especially with random interaction with the environment or fluctuation.

CHAPTER 7

Conclusion

In the experiment section, several experiments are used to measure the properties of the system, and multiple optical elements are adjusted in each step to ensure the maximum outcome. As a result, the photonic qubit is successfully created starting from the foundation concept of photon pairs generated by a nonlinear process. The qubits created in the experiment can be used as an entangle source in several quantum photonic experiments, such as teleportation or phase estimation. However, because the qubits are created continuously by a continuous laser source, they lose the possibility of being used in the algorithm, which relies on altering the extensive time delay between the pairs.

In the experiment with the photon loop, there are still some issues with noise and signal losses that need to be considered before adding more components to the system. The setup gave the highest identifiable loop at the fifth one, which is insufficient considering that the planned experiment includes more path splitting and therefore inevitably reduces more signal. In the future study, it is of the utmost importance to identify the sources of system noise and signal loss and then improve the signal-to-noise ratio to some degree before progressing to the next experiment.

In the theoretical section, the alternative scheme for polarization qubit gates, which adopted the Hilbert-space expansion technique, has multiple advantages over the standard scheme, which relies on matrix decomposing. Those advantages are:

- The process of decomposing the required gates into elementary single- and two-qubit gates can be avoided.
- The scheme has the potential to accelerate quantum simulation and information processing due to the lack of iteration required.
- A non-unitary matrix, or unknown matrix, can be programmed into the operator.
- By using linear combination, the circuit structure can avoid a large growth in complexity with a larger matrix operator.
- The scheme can be used to probe noise or decoherence, which makes it potentially useful in decoherence suppression, noise spectroscopy, and error correction.

The disadvantages of the alternative scheme are:

- As a result of post-selection, the probability of success decreases exponentially with a larger matrix operator.
- The system requires strong nonlinearity and highly efficient single-photon detection. A large operator is unlikely to be built successfully with current technology.

With these advantages, the demonstrated scheme is a convenient method to construct numerous quantum algorithms, which is not possible or impractical for the standard scheme, and the disadvantage part could be weak with advancing technologies in nonlinear optics such as photonics-integrated circuits or higher efficiency photon pairs sources.

BIBLIOGRAPHY

- 1 Altepeter, J. B., E. R. Jeffrey, and P. G. Kwiat. “Photonic State Tomography.” In *Advances In Atomic, Molecular, and Optical Physics*, edited by P. R. Berman and C. C. Lin, 52:105–59. Academic Press, 2005.
- 2 Aspect, Alain, Philippe Grangier, and Gérard Roger. “Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell’s Inequalities.” *Phys. Rev. Lett.* 49 (July 1, 1982): 91–94.
- 3 Aspect, Alain, Philippe Grangier, and Gérard Roger. “Experimental Tests of Realistic Local Theories via Bell’s Theorem.” *Physical Review Letters* 47, no. 7 (August 17, 1981): 460–63.
- 4 Baek, So-Young, and Yoon-Ho Kim. “Spectral Properties of Entangled Photon Pairs Generated via Frequency-Degenerate Type-I Spontaneous Parametric down-Conversion.” *Physical Review A* 77, no. 4 (April 4, 2008): 043807.
- 5 Barenco, Adriano, Charles Bennett, Richard Cleve, David DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, and Harald Weinfurter. “Elementary Gates for Quantum Computation.” *Physical Review A* 52 (March 23, 1995).
- 6 Barz, Stefanie, Gunther Cronenberg, Anton Zeilinger, and Philip Walther. “Heralded Generation of Entangled Photon Pairs.” *Nature Photonics - NAT PHOTONICS* 4 (July 16, 2010).
- 7 Burnham, David C., and Donald L. Weinberg. “Observation of Simultaneity in Parametric Production of Optical Photon Pairs.” *Physical Review Letters* 25, no. 2 (July 13, 1970): 84–87.
- 8 Carolan, Jacques, Christopher Harrold, Chris Sparrow, Enrique Martín-López, Nicholas J. Russell, Joshua W. Silverstone, Peter J. Shadbolt, et al. “Universal Linear Optics.” *Science* 349, no. 6249 (August 14, 2015): 711–16.
- 9 Caspani, L., E. Brambilla, and A. Gatti. “Tailoring the Spatiotemporal Structure of Biphoton Entanglement in Type-I Parametric down-Conversion.” *Physical Review A* 81, no. 3 (March 8, 2010): 033808.
- 10 Claudon, Julien, Joël Bleuse, Nitin Singh Malik, Maela Bazin, Périne Jaffrennou, Niels Gregersen, Christophe Sauvan, Philippe Lalanne, and Jean-Michel Gérard. “A Highly Efficient Single-Photon Source Based on a Quantum Dot in a Photonic Nanowire.” *Nature Photonics* 4, no. 3 (March 2010): 174–77.
- 11 Couteau, Christophe. “Spontaneous Parametric Down-Conversion.” *Contemporary Physics* 59, no. 3 (July 3, 2018): 291–304.
- 12 Deutsch, David, and Roger Penrose. “Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer.” *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400, no. 1818 (July 8, 1985): 97–117.
- 13 Di Lorenzo Pires, H., F. M. G. J. Coppens, and M. P. van Exter. “Type-I Spontaneous Parametric down-Conversion with a Strongly Focused Pump.” *Physical Review A* 83, no. 3 (March 31, 2011): 033837.

- 14 Dousse, Adrien, Jan Suffczyński, Alexios Beveratos, Olivier Krebs, Aristide Lemaître, Isabelle Sagnes, Jacqueline Bloch, Paul Voisin, and Pascale Senellart. “Ultrabright Source of Entangled Photon Pairs.” *Nature* 466, no. 7303 (July 2010): 217–20.
- 15 Echenim, Mnacho, and Mehdi Mhalla. *Quantum Projective Measurements and the CHSH Inequality in Isabelle/HOL*, 2021.
- 16 Geller, Michael R., John M. Martinis, Andrew T. Sornborger, Phillip C. Stancil, Emily J. Pritchett, Hao You, and Andrei Galiautdinov. “Universal Quantum Simulation with Prethreshold Superconducting Qubits: Single-Excitation Subspace Method.” *Physical Review A* 91, no. 6 (June 8, 2015): 062309.
- 17 Hamel, Deny R., Lynden K. Shalm, Hannes Hübel, Aaron J. Miller, Francesco Marsili, Varun B. Verma, Richard P. Mirin, Sae Woo Nam, Kevin J. Resch, and Thomas Jennewein. “Direct Generation of Three-Photon Polarization Entanglement.” *Nature Photonics* 8, no. 10 (October 2014): 801–7.
- 18 Hong, C. K., Z. Y. Ou, and L. Mandel. “Measurement of Subpicosecond Time Intervals between Two Photons by Interference.” *Physical Review Letters* 59, no. 18 (November 2, 1987): 2044–46.
- 19 Hudson, A. J., R. M. Stevenson, R. J. Young, P. Atkinson, K. Cooper, D. A. Ritchie, and A. J. Shields. “Biphoton Interference with a Quantum Dot Source of Entangled Light.” *Physica E: Low-Dimensional Systems and Nanostructures* 40, no. 6 (April 1, 2008): 1888–90.
- 20 James, Daniel, Paul Kwiat, William Munro, and Andrew White. “Measurement of Qubits.” *Phys. Rev. A* 64 (October 15, 2001): 052312.
- 21 Kane, B. E. “A Silicon-Based Nuclear Spin Quantum Computer.” *Nature* 393, no. 6681 (May 1998): 133–37.
- 22 Knill, E., R. Laflamme, and G. J. Milburn. “A Scheme for Efficient Quantum Computation with Linear Optics.” *Nature* 409, no. 6816 (January 2001): 46–52.
- 23 Leibfried, D., B. DeMarco, V. Meyer, D. Lucas, M. Barrett, J. Britton, W. M. Itano, et al. “Experimental Demonstration of a Robust, High-Fidelity Geometric Two Ion-Qubit Phase Gate.” *Nature* 422, no. 6930 (March 2003): 412–15.
- 24 Matsuda, Nobuyuki, Hanna Le Jeannic, Hiroshi Fukuda, Tai Tsuchizawa, William John Munro, Kaoru Shimizu, Koji Yamada, Yasuhiro Tokura, and Hiroki Takesue. “A Monolithically Integrated Polarization Entangled Photon Pair Source on a Silicon Chip.” *Scientific Reports* 2, no. 1 (November 12, 2012): 817.
- 25 Miller, David. “Self-Configuring Universal Linear Optical Component [Invited].” *Photonics Research* 1 (June 1, 2013): 1–15.
- 26 Monroe, C., D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. “Demonstration of a Fundamental Quantum Logic Gate.” *Physical Review Letters* 75, no. 25 (December 18, 1995): 4714–17.
- 27 Müller, M., H. Vural, C. Schneider, A. Rastelli, O. G. Schmidt, S. Höfling, and P. Michler. “Quantum-Dot Single-Photon Sources for Entanglement Enhanced Interferometry.” *Physical Review Letters* 118, no. 25 (June 22, 2017): 257402.

- 28 O'Brien, J. L., G. J. Pryde, A. G. White, T. C. Ralph, and D. Branning. "Demonstration of an All-Optical Quantum Controlled-NOT Gate." *Nature* 426, no. 6964 (November 2003): 264–67.
- 29 Peters, Nicholas, Joseph Altepeter, Evan Jeffrey, David Branning, and Paul Kwiat. "Precise Creation, Characterization, and Manipulation of Single Optical Qubits." *Quantum Information & Computation* 3, no. 7 (October 1, 2003): 503–17.
- 30 Pittman, T. B., B. C. Jacobs, and J. D. Franson. "Heralding Single Photons from Pulsed Parametric Down-Conversion." *Optics Communications* 246, no. 4 (February 15, 2005): 545–50.
- 31 Politi, Alberto, Martin J. Cryan, John G. Rarity, Siyuan Yu, and Jeremy L. O'Brien. "Silica-on-Silicon Waveguide Quantum Circuits." *Science* 320, no. 5876 (May 2, 2008): 646–49.
- 32 Qiang, Xiaogang, Xiaoqi Zhou, Jianwei Wang, Callum M. Wilkes, Thomas Loke, Sean O'Gara, Laurent Kling, et al. "Large-Scale Silicon Quantum Photonics Implementing Arbitrary Two-Qubit Processing." *Nature Photonics* 12, no. 9 (September 2018): 534–39.
- 33 Reck, Michael, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. "Experimental Realization of Any Discrete Unitary Operator." *Physical Review Letters* 73, no. 1 (July 4, 1994): 58–61.
- 34 Schmidt-Kaler, Ferdinand, Hartmut Häffner, Mark Riebe, Stephan Gulde, Gavin P. T. Lancaster, Thomas Deuschle, Christoph Becher, Christian F. Roos, Jürgen Eschner, and Rainer Blatt. "Realization of the Cirac-Zoller Controlled-NOT Quantum Gate." *Nature* 422, no. 6930 (March 2003): 408–11.
- 35 Shafi, K. Muhammed, R. S. Gayatri, A. Padhye, and C. M. Chandrashekar. "Bell-Inequality in Path-Entangled Single Photon and Purity Test," December 2021.
- 36 Wang, Jianwei, Stefano Paesani, Yunhong Ding, Raffaele Santagati, Paul Skrzypczyk, Alexia Salavrakos, Jordi Tura, et al. "Multidimensional Quantum Entanglement with Large-Scale Integrated Optics." *Science* 360, no. 6386 (April 20, 2018): 285–91.
- 37 Zhou, Xiao-Qi, Pruet Kalasuwan, Timothy C. Ralph, and Jeremy L. O'Brien. "Calculating Unknown Eigenvalues with a Quantum Algorithm." *Nature Photonics* 7, no. 3 (March 2013): 223–28.
- 38 Zhou, Xiao-Qi, Timothy C. Ralph, Pruet Kalasuwan, Mian Zhang, Alberto Peruzzo, Benjamin P. Lanyon, and Jeremy L. O'Brien. "Adding Control to Arbitrary Unknown Quantum Operations." *Nature Communications* 2, no. 1 (August 2, 2011): 413.

APPENDIX

APPENDIX A

M matrices

$$\widehat{M}_1 = \frac{1}{2} \begin{bmatrix} 2 & -1+1i & 1-1i & 1 \\ -1-1i & 0 & 1i & 0 \\ -1+1i & -1i & 0 & 0 \\ 1 & \mathbf{0} & 0 & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_2 = \frac{1}{2} \begin{bmatrix} 0 & -1+1i & 0 & 1 \\ -1-1i & 2 & 1i & -1-1i \\ 0 & -1i & 0 & 0 \\ 1 & -1+1i & 0 & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_3 = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1i & 0 \\ 0 & -1i & 0 & -1+1i \\ 1 & \mathbf{0} & 0 & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_4 = \frac{1}{2} \begin{bmatrix} 0 & 0 & -1-1i & 1 \\ 0 & 0 & 1i & 0 \\ -1+1i & -1i & 2 & -1+1i \\ 1 & \mathbf{0} & 1-1i & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_5 = \frac{1}{2} \begin{bmatrix} 0 & 0 & 2i & -1-1i \\ 0 & 0 & 1-1i & 0 \\ -2i & 1+1i & 0 & 0 \\ -1+1i & \mathbf{0} & 0 & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_6 = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -1-1i \\ 0 & 0 & 1-1i & 2i \\ 0 & 1+1i & 0 & 0 \\ -1+1i & -2i & 0 & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_7 = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -1-1i \\ 0 & 0 & -1+1i & 2 \\ \mathbf{0} & -1-1i & 0 & 0 \\ -1+1i & 2 & 0 & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_8 = \frac{1}{2} \begin{bmatrix} \mathbf{0} & \mathbf{0} & 2 & -1-1i \\ \mathbf{0} & \mathbf{0} & -1+1i & 2 \\ 2 & -1-1i & \mathbf{0} & \mathbf{0} \\ -1+1i & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_9 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & 1i \\ \mathbf{0} & \mathbf{0} & -1i & \mathbf{0} \\ \mathbf{0} & 1i & \mathbf{0} & \mathbf{0} \\ -1i & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_{10} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \\ \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\widehat{M}_{11} = \begin{bmatrix} 0 & 0 & 0 & 1i \\ 0 & 0 & 1i & 0 \\ 0 & -1i & 0 & 0 \\ -1i & 0 & 0 & 0 \end{bmatrix}$$

$$\widehat{M}_{12} = \frac{1}{2} \begin{bmatrix} 0 & 2 & 0 & -1-1i \\ 2 & 0 & -1-1i & 0 \\ 0 & -1+1i & 0 & 0 \\ -1+1i & 0 & 0 & 0 \end{bmatrix}$$

$$\widehat{M}_{13} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -1-1i \\ 0 & 0 & -1-1i & 0 \\ 0 & -1+1i & 0 & 2 \\ -1+1i & 0 & 2i & 0 \end{bmatrix}$$

$$\widehat{M}_{14} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -1+1i \\ 0 & 0 & 1-1i & 0 \\ 0 & 1+1i & 0 & -2i \\ -1-1i & 0 & 2i & 0 \end{bmatrix}$$

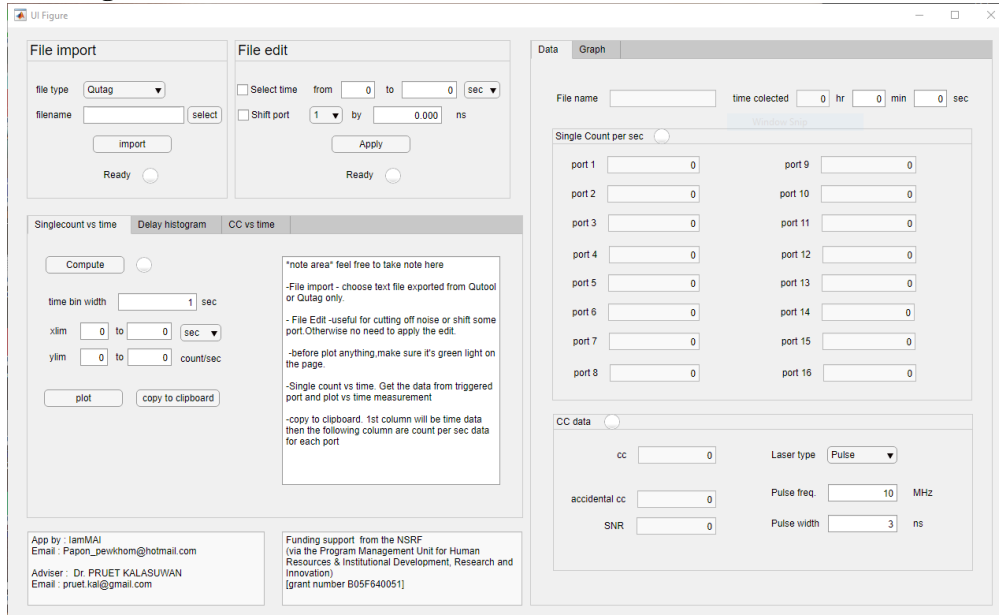
$$\widehat{M}_{15} = \frac{1}{2} \begin{bmatrix} 0 & -2i & 0 & -1+1i \\ 2i & 0 & 1-1i & 0 \\ 0 & 1+1i & 0 & 0 \\ -1-1i & 0 & 0 & 0 \end{bmatrix}$$

$$\widehat{M}_{16} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

APPENDIX B

Programing for coincident count

- Program first interface



- Coding for initial value

% Code that executes after component creation

```
function startupFcn(app)
    %initial value
    app.delayx = zeros(1,16);
    app.arraysinglecount = [];
    app.filename = "";
    app.filepath = "";
    app.timebinsingle = 1;
    app.singletimediv = 1;
    app.delaychange = 0;
    app.currentgraph = 0;
    app.cctimediv = 1;
end
```

- File import and edit

% Button pushed function: selectButton

```
function selectButtonPushed(app, event)
    %file name and filepath select
    [filenamehandle,filepathhandle] = uigetfile('*.*');
    %the handle will be 0 if cancle or exit ,else we get text value
    if filenamehandle ~= 0
        app.filename = filenamehandle;
    end
    if filepathhandle ~=0
        app.filepath = erase(filepathhandle,filenamehandle);
    end
end
```

```

end
app.exportpath = app.filepath;
%this cause bug that defocus the app
%refocus with next two commands
drawnow;
figure(app.UIFigure)
app.filenameEditField.Value = app.filename;
app.ReadyLamp_Import.Color = 'red';
end
% Value changed function: filenameEditField
function filenameEditFieldValueChanged(app, event)
    app.filename = app.filenameEditField.Value;
    app.ReadyLamp_Import.Color = 'red';
end
% Button pushed function: importButton
function importButtonPushed(app, event)

    %import command
    if strcmp('Qutag',app.filetypeDropDown.Value) == true
        %complier
        fileID = fopen(join([app.filepath,app.filename]));
        C = textscan(fileID,'%f %f ','Delimiter',';', 'headerlines', 5);
        fclose(fileID);
        app.A=C{1}; %bin where the count triggered

        %in case timetag overflow
        overflowtimetag = 2^52; %if bin go over this it'll reset to 0
        difneg = find(diff(app.A)<0) ; %find where time tag overflow and reset
        if numel(difneg) > 0
            for i = 1:numel(difneg)
                app.A((difneg(i)+1):end) = app.A((difneg(i)+1):end)+overflowtimetag;
            end
        end
        app.A=app.A-app.A(1); %set initial time to 0

        app.B=C{2}; %port where the count triggered
        app.Asave = app.A;
        app.Bsave = app.B;
        app.nA = numel(app.A);
        app.binwidth = 1; %for qutag
        app.mergebinEditField.Value = 100; %for qutag

    elseif strcmp('Qutool',app.filetypeDropDown.Value) == true
        %complier
        fileID = fopen(join([app.filepath,app.filename]));
        C = textscan(fileID,'%f %f ','Delimiter','');
    end
end

```

```

fclose(fileID);
app.A=C{1}; %bin where the count triggered
app.B=C{2}; %port where the count triggered
app.Bsave = app.B;
app.A=app.A-app.A(1); %set initial time to 0
app.Asave = app.A;
app.nA = numel(app.A);
app.binwidth = 81; %for qutool
app.mergetbinEditField.Value = 2; %for qutool
end

%reset all value
app.timebinsingle = 1;
app.singletimediv = 1;
app.delaychange =0;
app.delayx = zeros(1,16);
%reset in singlecount
app.arraysinglecount =[];

%%%calculate value
app.alltimerec = app.A(end)*app.binwidth*1e-12;
app.captimerec = app.alltimerec;
for i = 1:16
    app.xport(i) = numel(find(app.Bsave == i))/app.captimerec; %set port count
    app.(jjoin(['port',num2str(i),'EditField'])).Value = app.xport(i); %show on display
end
app.FilenameEditField.Value = app.filename;

app.timecollectedhour.Value = floor(app.alltimerec/3600);
app.timecollectedmin.Value = floor(rem(app.alltimerec,3600)/60);
app.timecollectedsec.Value = rem(app.alltimerec,60);

app.nport = find(app.xport ~=0); %array of which port triggered
for i = 1:numel(app.nport)
    nporthandle(i) = {num2str(app.nport(i))}; %convert to list
end
%set the dropdown
app.StoportDropDown.Items = nporthandle;
app.StartportDropDown.Items = nporthandle;
app.StartportDropDown.Value = app.StartportDropDown.Items(1);
app.StoportDropDown.Value = app.StoportDropDown.Items(2);
app.Portshift.Items = nporthandle;

%lamp unready
app.ReadyLamp_CC.Color = 'red';

```



```

app.ReadyLamp_Delay.Color = 'red';
app.ReadyLamp_Single.Color = 'red';
app.ReadyLamp_FileEdit.Color = 'white';
app.CCDataLamp.Color = 'white';

%Clear File Edit
app.SelecttimeCheckBox.Value = 0;
app.toEditField.Value = app.alltimerec;
app.fromEditField.Value = 0;
app.ShiftportCheckBox.Value = 0;
app.PortShiftEditField.Value = 0;
app.SelecttimeDropDown.Value = app.SelecttimeDropDown.Items(1);
app.selecttimediv = 1;

%lamp ready
app.ReadyLamp_Import.Color = 'green';
app.SingleDataLamp.Color = 'green';

end
% Callback function
function test2ButtonPushed(app, event)
    app.txttestEditField.Value = app.exportpath;
end
% Value changed function: SelecttimeDropDown
function SelecttimeDropDownValueChanged(app, event)
    %remember timediv b4 change
    timedivb4 = app.selecttimediv;

    %set timediv according to dropdown value
    if strcmp('sec',app.SelecttimeDropDown.Value) == true
        app.selecttimediv = 1;
    elseif strcmp('min',app.SelecttimeDropDown.Value) == true
        app.selecttimediv = 60;
    elseif strcmp('hour',app.SelecttimeDropDown.Value) == true
        app.selecttimediv = 60*60;
    end

    %change number in time box
    app.fromEditField.Value = app.fromEditField.Value*timedivb4/app.selecttimediv;
    app.toEditField.Value = app.toEditField.Value*timedivb4/app.selecttimediv;

end
% Value changed function: PortShiftEditField
function PortShiftEditFieldValueChanged(app, event)
    port2shift = str2num(app.Portshift.Value);
    app.delayx(port2shift) = app.PortShiftEditField.Value; %change delay value from input
end

```

```

% Value changed function: Portshift
function PortshiftValueChanged(app, event)
    port2shift = str2num(app.Portshift.Value);
    app.PortShiftEditField.Value = app.delayx(port2shift) ; %change input box according to
drop down pick
end
% Button pushed function: ApplyButton
function ApplyButtonPushed(app, event)
    app.ReadyLamp_FileEdit.Color = 'red';
    drawnow

%timeEdit
    app.A = app.Asave;
    app.B = app.Bsave;
    if app.SelecttimeCheckBox.Value == 1
        cutfrom = find(app.A >=
app.fromEditField.Value*app.selecttimediv*1e12/app.binwidth,1);
        if or(cutfrom == [],cutfrom == [1])
            cutfrom = [1];
        end
        cutto = find(app.A <=
app.toEditField.Value*app.selecttimediv*1e12/app.binwidth,1,'last');
        app.A = app.Asave(cutfrom:cutto);
        app.B = app.Bsave(cutfrom:cutto);
    end

%delayEdit
    if app.ShiftportCheckBox.Value == 1
        for i =1:16
            locate = [];
            if app.delayx(i) ~= 0
                locate = find(app.Bsave == i);
                app.Asave(locate) = app.Asave(locate) + app.delayx(i)*1000/app.binwidth;
            end
        end
    end

%%%calculate value
app.captimerec = app.A(end)*app.binwidth*1e-12;
for i = 1:16
    app.xport(i) = numel(find(app.B== i))/app.captimerec; %set port count
    app.(join(['port',num2str(i),'EditField'])).Value = app.xport(i); %show on display
end

%lamp unready
app.ReadyLamp_CC.Color = 'red';

```

```

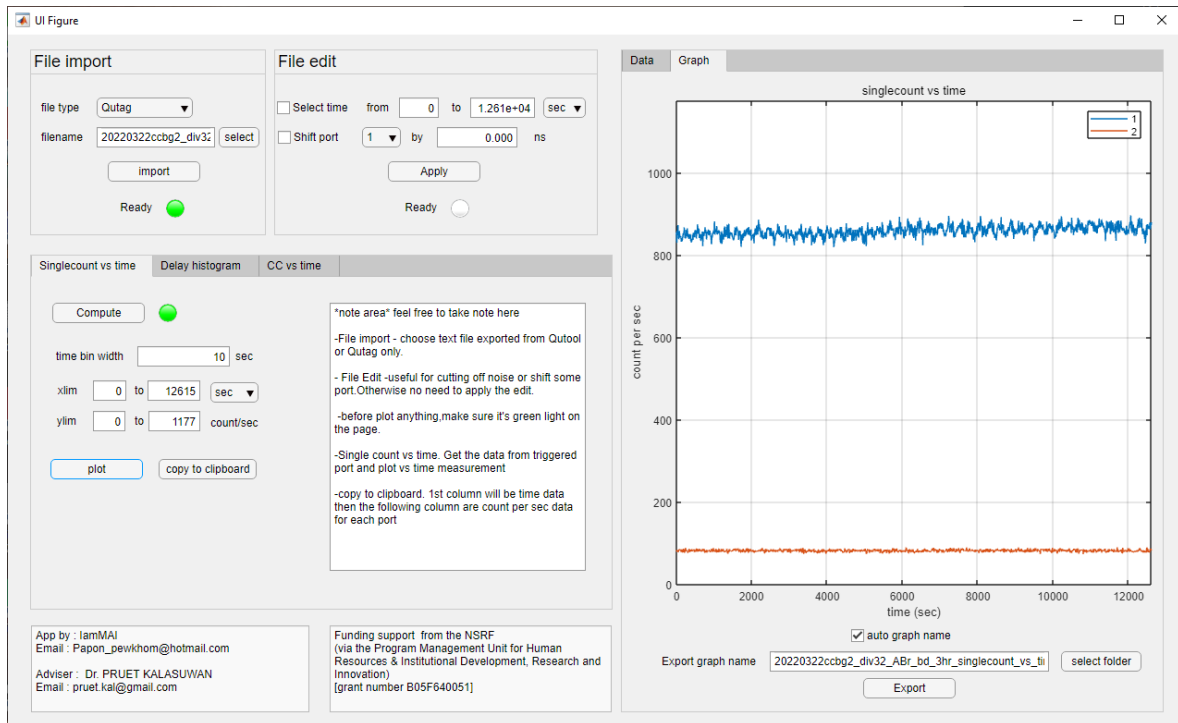
app.ReadyLamp_Delay.Color = 'red';
app.ReadyLamp_Single.Color = 'red';
%lamp ready
app.ReadyLamp_FileEdit.Color = 'green';
end

```

- **Interface single count vs time computed and plot.**

The screenshot displays the 'UI Figure' application interface, which is divided into several functional sections:

- File import:** Includes a dropdown for 'file type' (set to 'Outtag'), a text field for 'filename' (20220322ccbg2_div32), and an 'import' button. A green 'Ready' indicator is present.
- File edit:** Features checkboxes for 'Select time' and 'Shift port', with input fields for time ranges (0 to 1.261e+04 sec and 1 by 0.000 ns) and an 'Apply' button. A white 'Ready' indicator is present.
- Data/Graph:** Shows 'File name' (20220322ccbg2_div32_) and 'time collected' (3 hr 30 min 14.61 sec). Below this is a 'Single Count per sec' section with a green indicator and a grid of 16 ports. Ports 1 and 2 show values of 859.5 and 82.79 respectively, while others are 0.
- Singlecount vs time:** Contains a 'Compute' button with a green indicator, 'time bin width' (10 sec), and 'xlim' (0 to 12615 sec) and 'ylim' (0 to 1177 count/sec) settings. It includes 'plot' and 'copy to clipboard' buttons.
- CC data:** Includes a radio button, 'cc' (0), 'accidental cc' (0), 'SNR' (0), 'Laser type' (Pulse), 'Pulse freq.' (10 MHz), and 'Pulse width' (3 ns).
- Footer:** Contains contact information for the app developer (IamMAI) and funding support from the NSRF.



- Coding for single count vs time

% Button pushed function: ComputeButton_Single

function ComputeButton_SinglePushed(app, event)

hissecnum = zeros(16, floor(((app.A(end)-app.A(1))*app.binwidth*(10⁻¹²)/app.timebinsingle+1)); **%row = port and column = amount of that port trigger in that sec**
for t = 1:numel(app.A)

seccount = floor(((app.A(t)-app.A(1))*app.binwidth*(10⁻¹²)/app.timebinsingle+1);
 hissecnum(app.B(t),seccount) = hissecnum(app.B(t),seccount)+1;

end

hissecnum(:,end) = [];

app.arraysinglecount = [];

for i = 1:numel(app.nport)

app.arraysinglecount(i,:) = hissecnum(app.nport(i,:)); **%this will only save the triggered**

port

end

%reset value seting

app.timebinsingle = 1;

app.timebinwidthEditField.Value = app.timebinsingle;

app.x1single.Value = 0;

app.x2single.Value = app.captimerec;

app.singletimediv = 1;

app.DropDown_xlimsingle.Value = 'sec';

app.y1single.Value = 0;

app.y2single.Value = max(max(app.arraysinglecount))*1.2;

```

%reset the rate
app.rxvstime = app.arraysinglecount;
app.ratetime = 0:size(app.arraysinglecount,2)-1 ;

%lamp on
app.ReadyLamp_Single.Color = 'green';
end
% Value changed function: DropDown_xlimsingle
function DropDown_xlimsingleValueChanged(app, event)
%remember timediv b4 change
timedivb4 = app.singletimediv;

%set timediv according to dropdown value
if strcmp('sec',app.Dropdown_xlimsingle.Value) == true
    app.singletimediv = 1;
elseif strcmp('min',app.Dropdown_xlimsingle.Value) == true
    app.singletimediv = 60;
elseif strcmp('hour',app.Dropdown_xlimsingle.Value) == true
    app.singletimediv = 60*60;
end

%change number in time box
app.x1single.Value = app.x1single.Value*timedivb4/app.singletimediv;
app.x2single.Value = app.x2single.Value*timedivb4/app.singletimediv;

end
% Button pushed function: plotButton
function plotButtonPushed(app, event)

app.rxvstime = [];
%caculate x y data
if app.timebinwidthEditField.Value ==1
    app.timebinsingle = 1;
    app.rxvstime = app.arraysinglecount;
    app.ratetime = 0:size(app.arraysinglecount,2)-1 ;
else
    app.timebinsingle = app.timebinwidthEditField.Value;
    %the folowing code cal mean for every time bin
    numadd = app.timebinsingle-mod(size(app.arraysinglecount,2),app.timebinsingle);
    A2 = app.arraysinglecount;
    if numadd ~= 0
        A2(:,end+1:end+numadd)=missing; %add nan to fill b4 reshape
    end
    for i = 1:size(A2,1)
        S = reshape(A2(i,:),app.timebinsingle,[]);

```

```

        app.rxvstime(i,:) = mean(S,'omitnan'); %here we get the mean
    end
    app.ratetime = 0:app.timebinsingle:(size(app.rxvstime,2)-1)*app.timebinsingle;
end

%ploting
for i =1:numel(app.nport)
    plot(app.UIAxes,app.ratetime/app.singletimediv,app.rxvstime(i,:))
    hold(app.UIAxes,'on')
end
hold(app.UIAxes,'off')

%plot detail
app.UIAxes.XLim = [app.x1single.Value,app.x2single.Value];
app.UIAxes.YLim = [app.y1single.Value,app.y2single.Value];
app.UIAxes.Title.String = 'singlecount vs time';
app.UIAxes.Title.FontWeight = 'normal';
if app.singletimediv == 1
    app.UIAxes.XLabel.String = 'time (sec)';
elseif app.singletimediv ==60
    app.UIAxes.XLabel.String = 'time (min)';
elseif app.singletimediv == 3600
    app.UIAxes.XLabel.String = 'time (hour)';
end
app.UIAxes.YScale = 'linear';
app.UIAxes.YLabel.String = 'counts per sec';

%add legend
for i = 1:numel(app.nport)
    nporthandle(i) = {num2str(app.nport(i))}; %convert to list
end
legend(app.UIAxes,nporthandle);

%change tab to graph
app.TabGroup2.SelectedTab = app.GraphTab;

%change graph name
if app.autographnameCheckBox.Value ==1
    filenamenodot = erase(app.filename, '.txt');
    app.ExportgraphnameEditField.Value = join([filenamenodot, '_singlecount_vs_time']);
end
app.currentgraph = 1;
end
% Button pushed function: copytoclipboardButton
function copytoclipboardButtonPushed(app, event)
    %arraytocoppy = [app.ratetime/app.singletimediv;app.rxvstime]; %this will copy all data
    %instead we want only data in plot range

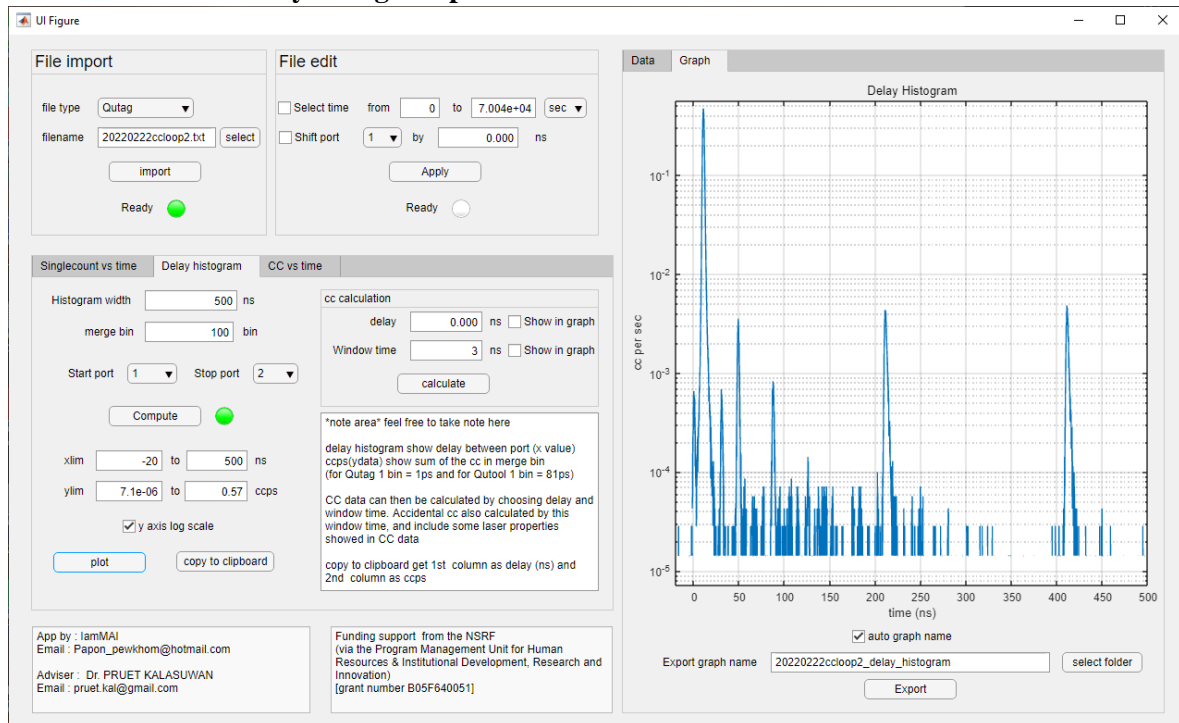
```

```

xfirst = find(app.ratetime >= app.x1single.Value*app.singletimediv,1);
xlast = find(app.ratetime <= app.x2single.Value*app.singletimediv,1,'last');
arraytocopy = [app.ratetime(xfirst:xlast)/app.singletimediv;app.rxvstime(:,(xfirst:xlast))];
num2clip(app,transpose(arraytocopy));
end

```

- **Interface with delay histogram plotted.**



- **Coding for delay histogram**

```

% Button pushed function: ComputeButton_Delay
function ComputeButton_DelayPushed(app, event)
    %lamp unready
    app.ReadyLamp_Delay.Color = 'red';
    drawnow

    %computation
    hw = app.HistogramwidthEditField.Value*1000/app.binwidth;
    diffA=diff(app.A); %dif of time in adjacent tile
    difflo = find(diffA < hw+1); %find tile where dif < width
    nlo = numel(difflo); %number of tiles in such condition

    startp = str2num(app.StartportDropDown.Value);
    stopp = str2num(app.StoppportDropDown.Value);

```

```

%cal. ccdif = bindif(bin) which is in range of histogram
p=1;
for i =1:nlo
    j=1; %tile seperation
    while app.A(difflo(i)+j)-app.A(difflo(i)) < hw+1
        if and(app.B(difflo(i)+j) == stopp,app.B(difflo(i)) == startp)
            ccdif(p)= (app.A(difflo(i)+j)-app.A(difflo(i))); %if start -> stop, collect delay
            p = p+1;
        elseif and(app.B(difflo(i)+j) == startp,app.B(difflo(i)) == stopp)
            ccdif(p)= (app.A(difflo(i)+j)-app.A(difflo(i)))*(-1); %if stop -> start, collect -delay
            p = p+1;
        end
        j=j+1;
        if difflo(i)+j > numel(app.nA)
            break
        end
    end
end

%histogram cal
hw_m = app.mergebinEditField.Value ; %merge histogram width
edges = -1*hw-0.1:hw_m:hw-0.1; %(bin)
xdata = -1*hw+hw_m/2:hw_m:hw-hw_m/2; %(bin)
app.timedata = xdata*app.binwidth/1000; %ns
Nhis = histcounts(ccdif,edges);
app.ccps = Nhis./app.captimerec; %photon per sec that trigger in stop port in range ~[start
time +delay-hw_m/2:start time +delay+hw_m/2]

%set graph limit value

app.x1delay.Value = -1*app.HistogramwidthEditField.Value;
app.x2delay.Value = app.HistogramwidthEditField.Value;
if startp == stopp
    app.x1delay.Value =0; %set xlim left to 0 when compute the self trigger
end

app.y1delay.Value = (1./app.captimerec)*0.5;
app.y2delay.Value = max(app.ccps)*1.2;

%lamp ready
app.ReadyLamp_Delay.Color ='green';

end
% Button pushed function: plotButton_Delay
function plotButton_DelayPushed(app, event)

```



```

plot(app.UIAxes,app.timedata,app.ccps)

if app.yaxislogscaleCheckBox.Value == 1
    app.UIAxes.YScale = 'log';
elseif app.yaxislogscaleCheckBox.Value == 0
    app.UIAxes.YScale = 'linear';
end

app.UIAxes.XLim = [app.x1delay.Value,app.x2delay.Value];

app.UIAxes.YLim = [app.y1delay.Value,app.y2delay.Value];

app.UIAxes.Title.String = 'Delay Histogram';
app.UIAxes.Title.FontWeight = 'normal';
app.UIAxes.XLabel.String = 'time (ns)';
app.UIAxes.YLabel.String = 'coincidence counts per sec';

%remove legend
legend(app.UIAxes,'off');

%change tab to graph
app.TabGroup2.SelectedTab = app.GraphTab;

%change graph name
if app.autographnameCheckBox.Value == 1
    filenamenodot = erase(app.filename, '.txt');
    app.ExportgraphnameEditField.Value = join([filenamenodot, '_delay_histogram']);
end

%change cc cal plot
app.ShowingraphDelay.Value = 0;
app.ShowingraphWt.Value = 0;

app.currentgraph = 2;
end
% Value changed function: HistogramwidthEditField
function HistogramwidthEditFieldValueChanged(app, event)
    %lamp unready
    app.ReadyLamp_Delay.Color = 'red';
end
% Value changed function: mergebinEditField
function mergebinEditFieldValueChanged(app, event)
    %lamp unready
    app.ReadyLamp_Delay.Color = 'red';
end
% Value changed function: StartportDropDown
function StartportDropDownValueChanged(app, event)

```

```

    %lamp unready
    app.ReadyLamp_Delay.Color = 'red';
end
% Value changed function: StopportDropDown
function StopportDropDownValueChanged(app, event)
    %lamp unready
    app.ReadyLamp_Delay.Color = 'red';
end
% Button pushed function: copytoclipboardButton_Delay
function copytoclipboardButton_DelayPushed(app, event)
    xfirst = find(app.timedata >= app.x1delay.Value,1);
    xlast = find(app.timedata <= app.x2delay.Value,1,'last');
    arraytocoppy = [app.timedata(xfirst:xlast);app.ccps(xfirst:xlast)];
    num2clip(app,transpose(arraytocoppy));
end
% Value changed function: ShowingraphDelay
function ShowingraphDelayValueChanged(app, event)
    %vertical line for middle cc calculate
    delete(app.showmidcc);
    if app.currentgraph == 2
        if app.ShowingraphDelay.Value == 1
            hold(app.UIAxes,'on')
            xdata = [app.delayEditField.Value app.delayEditField.Value];
            ydata = [app.y1delay.Value app.y2delay.Value];
            app.showmidcc = plot(app.UIAxes,xdata,ydata,'--');
            hold(app.UIAxes,'off')
        end
    end
end

end
% Value changed function: delayEditField
function delayEditFieldValueChanged(app, event)
    %vertical line for middle cc calculate
    delete(app.showmidcc);
    if app.currentgraph == 2
        if app.ShowingraphDelay.Value == 1
            hold(app.UIAxes,'on')
            xdata = [app.delayEditField.Value app.delayEditField.Value];
            ydata = [app.y1delay.Value app.y2delay.Value];
            app.showmidcc = plot(app.UIAxes,xdata,ydata,'--');
            hold(app.UIAxes,'off')
        end
    end
end

%vertical line for cc edge
delete(app.showleftcc);

```

```

delete(app.showrightcc);
if app.currentgraph == 2
    if app.ShowingraphWt.Value == 1
        hold(app.UIAxes,'on')
        xdata1 = [app.delayEditField.Value-app.WindowtimeEditField.Value/2
app.delayEditField.Value-app.WindowtimeEditField.Value/2];
        xdata2 = [app.delayEditField.Value+app.WindowtimeEditField.Value/2
app.delayEditField.Value+app.WindowtimeEditField.Value/2];
        ydata = [app.y1delay.Value app.y2delay.Value];
        app.showleftcc = plot(app.UIAxes,xdata1,ydata,'r');
        app.showrightcc = plot(app.UIAxes,xdata2,ydata,'r');
        hold(app.UIAxes,'off')
    end
end
end
% Value changed function: ShowingraphWt
function ShowingraphWtValueChanged(app, event)
    %vertical line for cc edge
    delete(app.showleftcc);
    delete(app.showrightcc);
    if app.currentgraph == 2
        if app.ShowingraphWt.Value == 1
            hold(app.UIAxes,'on')
            xdata1 = [app.delayEditField.Value-app.WindowtimeEditField.Value/2
app.delayEditField.Value-app.WindowtimeEditField.Value/2];
            xdata2 = [app.delayEditField.Value+app.WindowtimeEditField.Value/2
app.delayEditField.Value+app.WindowtimeEditField.Value/2];
            ydata = [app.y1delay.Value app.y2delay.Value];
            app.showleftcc = plot(app.UIAxes,xdata1,ydata,'r');
            app.showrightcc = plot(app.UIAxes,xdata2,ydata,'r');
            hold(app.UIAxes,'off')
        end
    end
end
end
% Value changed function: WindowtimeEditField
function WindowtimeEditFieldValueChanged(app, event)
    %vertical line for cc edge
    delete(app.showleftcc);
    delete(app.showrightcc);
    if app.currentgraph == 2
        if app.ShowingraphWt.Value == 1
            hold(app.UIAxes,'on')
            xdata1 = [app.delayEditField.Value-app.WindowtimeEditField.Value/2
app.delayEditField.Value-app.WindowtimeEditField.Value/2];
            xdata2 = [app.delayEditField.Value+app.WindowtimeEditField.Value/2
app.delayEditField.Value+app.WindowtimeEditField.Value/2];

```

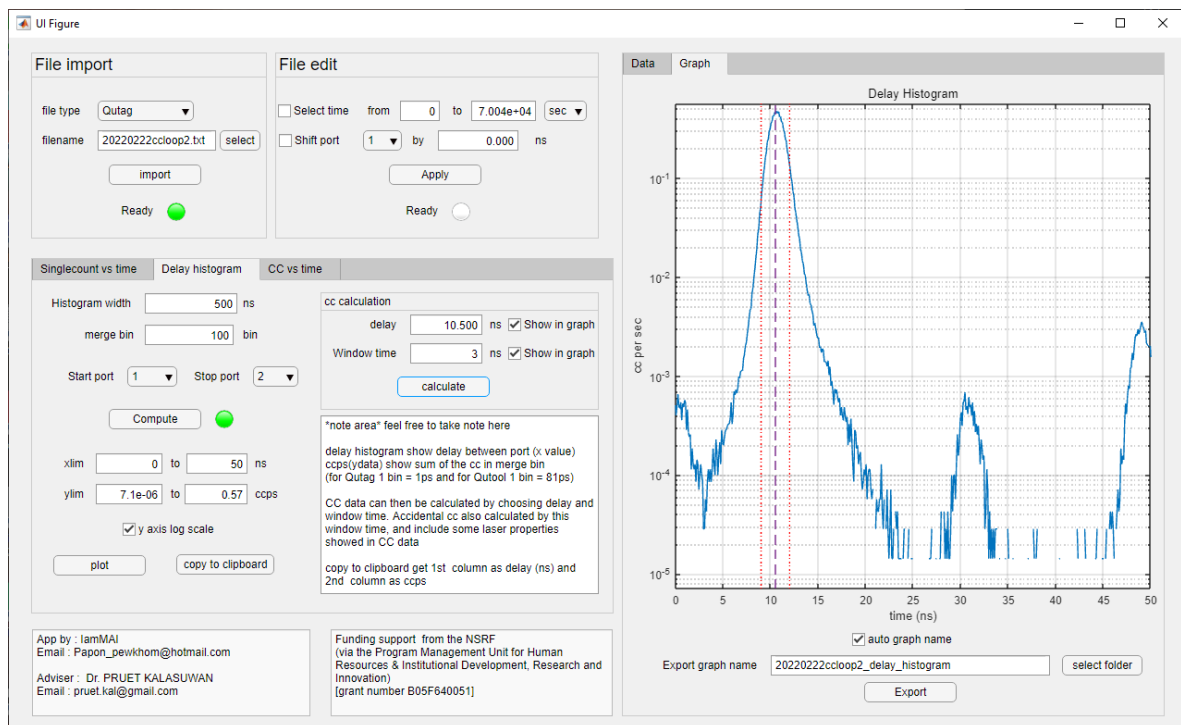
```

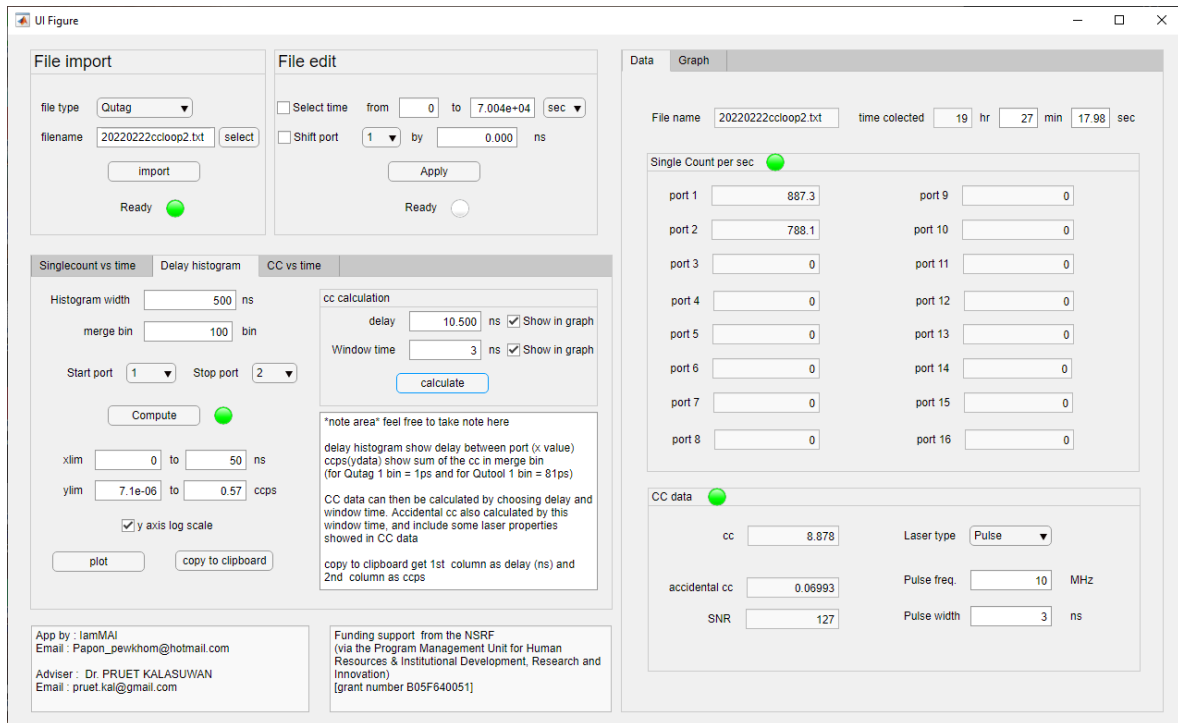
ydata = [app.y1delay.Value app.y2delay.Value];
app.showleftcc = plot(app.UIAxes,xdata1,ydata,'r');
app.showrightcc = plot(app.UIAxes,xdata2,ydata,'r');
hold(app.UIAxes,'off')
end
end

end
% Button pushed function: calculateButton

```

- **Interface with coincident count computed by selecting windows time range from the histogram**





- Coding for CC computation

% Button pushed function: calculateButton

function calculateButtonPushed(app, event)

%change tab to Data

app.TabGroup2.SelectedTab = app.DataTab;

ccleft = find(app.timedata < app.delayEditField.Value-

app.WindowtimeEditField.Value/2,1,'last')+ 1 ;

ccright = find(app.timedata >

app.delayEditField.Value+app.WindowtimeEditField.Value/2,1)- 1;

sumcc = sum(app.ccps(ccleft:ccright));

app.ccEditField.Value = sumcc;

%calculate accidental cc

startp = str2num(app.StartportDropDown.Value);

stopp = str2num(app.StoppportDropDown.Value);

n1 = app.xport(startp);

n2 = app.xport(stopp);

if strcmp('CW',app.LasertypeDropDown.Value) == true

app.accidentalccEditField.Value = n1*n2*app.WindowtimeEditField.Value*1e-9;

elseif strcmp('Pulse',app.LasertypeDropDown.Value) == true

if app.WindowtimeEditField.Value>= app.PulsewidthEditField.Value

app.accidentalccEditField.Value = (n1*n2/(app.PulsefreqEditField.Value*1e6));

else

```

        app.accidentalccEditField.Value =
(n1*n2/(app.PulsefreqEditField.Value*1e6))*(app.WindowtimeEditField.Value/app.PulsewidthE
ditField.Value);
    end
end

app.SNREditField.Value = app.ccEditField.Value/app.accidentalccEditField.Value;

%lamp
app.CCDDataLamp.Color = 'green';

end
% Value changed function: LasertypeDropDown
function LasertypeDropDownValueChanged(app, event)
    %calculate accidental cc
    startp = str2num(app.StartportDropDown.Value);
    stopp = str2num(app.StoppportDropDown.Value);
    n1 = app.xport(startp);
    n2 = app.xport(stopp);

    if strcmp('CW',app.LasertypeDropDown.Value) == true
        app.accidentalccEditField.Value = n1*n2*app.WindowtimeEditField.Value*1e-9;
    elseif strcmp('Pulse',app.LasertypeDropDown.Value) == true
        if app.WindowtimeEditField.Value>= app.PulsewidthEditField.Value
            app.accidentalccEditField.Value = (n1*n2/(app.PulsefreqEditField.Value*1e6));
        else
            app.accidentalccEditField.Value =
(n1*n2/(app.PulsefreqEditField.Value*1e6))*(app.WindowtimeEditField.Value/app.PulsewidthE
ditField.Value);
        end
    end
end

app.SNREditField.Value = app.ccEditField.Value/app.accidentalccEditField.Value;

end
% Value changed function: PulsefreqEditField
function PulsefreqEditFieldValueChanged(app, event)
    %calculate accidental cc
    startp = str2num(app.StartportDropDown.Value);
    stopp = str2num(app.StoppportDropDown.Value);
    n1 = app.xport(startp);
    n2 = app.xport(stopp);

    if strcmp('CW',app.LasertypeDropDown.Value) == true
        app.accidentalccEditField.Value = n1*n2*app.WindowtimeEditField.Value*1e-9;
    elseif strcmp('Pulse',app.LasertypeDropDown.Value) == true
        if app.WindowtimeEditField.Value>= app.PulsewidthEditField.Value

```

```

        app.accidentalccEditField.Value = (n1*n2/(app.PulsefreqEditField.Value*1e6));
    else
        app.accidentalccEditField.Value =
(n1*n2/(app.PulsefreqEditField.Value*1e6))*(app.WindowtimeEditField.Value/app.PulsewidthE
ditField.Value);
    end
end

app.SNREditField.Value = app.ccEditField.Value/app.accidentalccEditField.Value;

end
% Value changed function: PulsewidthEditField
function PulsewidthEditFieldValueChanged(app, event)
    %calculate accidental cc
    startp = str2num(app.StartportDropDown.Value);
    stopp = str2num(app.StoppportDropDown.Value);
    n1 = app.xport(startp);
    n2 = app.xport(stopp);

    if strcmp('CW',app.LasertypeDropDown.Value) == true
        app.accidentalccEditField.Value = n1*n2*app.WindowtimeEditField.Value*1e-9;
    elseif strcmp('Pulse',app.LasertypeDropDown.Value) == true
        if app.WindowtimeEditField.Value >= app.PulsewidthEditField.Value
            app.accidentalccEditField.Value = (n1*n2/(app.PulsefreqEditField.Value*1e6));
        else
            app.accidentalccEditField.Value =
(n1*n2/(app.PulsefreqEditField.Value*1e6))*(app.WindowtimeEditField.Value/app.PulsewidthE
ditField.Value);
        end
    end
end

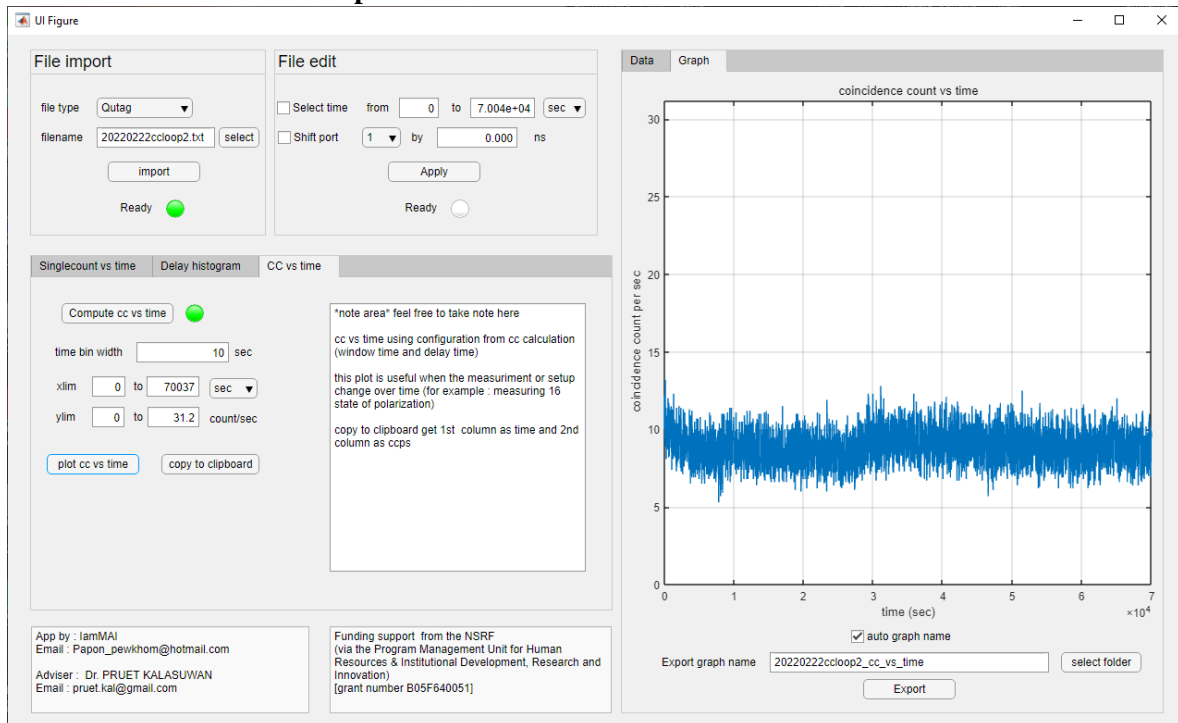
app.SNREditField.Value = app.ccEditField.Value/app.accidentalccEditField.Value;

end

```

-
-

Interface in CC vs time plot



- Coding for cc vs time

```

% Button pushed function: ComputeButton_CC
function ComputeButton_CCPushed(app, event)
    hw = app.HistogramwidthEditField.Value*1000/app.binwidth;
    diffA=diff(app.A); %dif of time in adjacent tile
    difflo = find(diffA < hw+1); %find tile where dif < width
    nlo = numel(difflo); %number of tiles in such condition
    startp = str2num(app.StartportDropDown.Value);
    stopp = str2num(app.StoppportDropDown.Value);
    app.hisseccc = zeros(1 ,floor((app.A(end)-app.A(1))*app.binwidth*(10^(-12))+1) );
    p=1;
    ccdif=[];
    for i=1:nlo
        j=1; %tile seperation
        while app.A(difflo(i)+j)-app.A(difflo(i)) < hw+1
            if and(app.B(difflo(i)+j) == stopp,app.B(difflo(i)) == startp)
                ccdif(p)= (app.A(difflo(i)+j)-app.A(difflo(i)))*app.binwidth*(10^(-3)); %here
                % ccdif cal in unit of ns instead of bin
                if abs(ccdif(p)-app.delayEditField.Value) <= app.WindowtimeEditField.Value/2
                    seccc = floor((app.A(difflo(i)+j)-app.A(1))*app.binwidth*(10^(-12)))+1;
                    app.hisseccc(seccc) = app.hisseccc(seccc) +1;
                end
                p = p+1;
            elseif and(app.B(difflo(i)+j) == startp,app.B(difflo(i)) == stopp)

```



```

        ccdif(p)= (app.A(difflo(i)+j)-app.A(difflo(i)))*app.binwidth*(10^(-3));
        if abs(ccdif(p)+app.delayEditField.Value) <= app.WindowtimeEditField.Value/2
            seccc = floor((app.A(difflo(i)+j)-app.A(1))*app.binwidth*(10^(-12)))+1;
            app.hisseccc(seccc) = app.hisseccc(seccc) +1;
        end
        p = p+1;
    end
    j=j+1;
end
end

%Lamp on
app.ReadyLamp_CC.Color = 'green';

%set the Value
app.timebinwidthCCplot.Value = 1;
app.x1cc.Value = 0;
app.x2cc.Value = numel(app.hisseccc)-1;
app.y1cc.Value = 0;
app.y2cc.Value = max(app.hisseccc)*1.2;
app.ccxDropDown.Value = 'sec';

end
% Value changed function: ccxDropDown
function ccxDropDownValueChanged(app, event)
    %remember timediv b4 change
    timedivb4 = app.cctimediv;

    %set timediv according to dropdown value
    if strcmp('sec',app.ccxDropDown.Value) == true
        app.cctimediv = 1;
    elseif strcmp('min',app.ccxDropDown.Value) == true
        app.cctimediv = 60;
    elseif strcmp('hour',app.ccxDropDown.Value) == true
        app.cctimediv = 60*60;
    end

    %change number in time box
    app.x1cc.Value = app.x1cc.Value*timedivb4/app.cctimediv;
    app.x2cc.Value = app.x2cc.Value*timedivb4/app.cctimediv;

end
% Button pushed function: plotccvstimeButton
function plotccvstimeButtonPushed(app, event)
    app.ccvstime =[];
    %caculate x y data
    if app.timebinwidthCCplot.Value ==1

```

```

app.ccvstime = app.hisseccc;
app.cctime = 0:numel(app.hisseccc)-1 ;
else
timebin = app.timebinwidthCCplot.Value;
%the following code cal mean for every time bin
numadd = timebin-mod(numel(app.hisseccc),timebin);
A2 = app.hisseccc;
if numadd ~= 0
    A2(end+1:end+numadd)=missing; %add nan to fill b4 reshape
end
S=reshape(A2,timebin,[]);
app.ccvstime = mean(S,'omitnan');
app.cctime = 0:timebin:(numel(app.ccvstime)-1)*timebin;
end

%ploting
plot(app.UIAxes,app.cctime/app.cctimediv,app.ccvstime)

%plot detail
app.UIAxes.XLim = [app.x1cc.Value,app.x2cc.Value];
app.UIAxes.YLim = [app.y1cc.Value,app.y2cc.Value];
app.UIAxes.Title.String = 'coincidence count vs time';
app.UIAxes.Title.FontWeight = 'normal';
if app.cctimediv == 1
    app.UIAxes.XLabel.String = 'time (sec)';
elseif app.cctimediv ==60
    app.UIAxes.XLabel.String = 'time (min)';
elseif app.cctimediv == 3600
    app.UIAxes.XLabel.String = 'time (hour)';
end
app.UIAxes.YScale = 'linear';
app.UIAxes.YLabel.String = 'coincidence counts per sec';

legend(app.UIAxes,'off');

%change tab to graph
app.TabGroup2.SelectedTab = app.GraphTab;

%change graph name
if app.autographnameCheckBox.Value ==1
    filenameodot = erase(app.filename,'.txt');
    app.ExportgraphnameEditField.Value = join([filenameodot,'_cc_vs_time']);
end
app.currentgraph = 3;
end
% Button pushed function: copytoclipboardCC
function copytoclipboardCCButtonPushed(app, event)

```

```

% data in plot range
xfirst = find(app.cctime >= app.x1cc.Value*app.cctimediv,1);
xlast = find(app.cctime <= app.x2cc.Value*app.cctimediv,1,'last');
arraytocopy = [app.cctime(xfirst:xlast)/app.cctimediv;app.ccvstime(:,(xfirst:xlast))];
num2clip(app,transpose(arraytocopy));
end

```

- **Coding in exporting part**

```

% Button pushed function: selectfolderButton
function selectfolderButtonPushed(app, event)
    pathhandle = uigetdir(app.exportpath);
    if pathhandle ~= 0
        app.exportpath = pathhandle;
    end

    %this cause bug that defocus the app
    %refocus with next two commands
    drawnow;
    figure(app.UIFigure)
end

% Button pushed function: ExportButton
function ExportButtonPushed(app, event)
    % Create a temporary figure with axes.
    fig = figure;
    fig.Visible = 'off';
    figAxes = axes(fig);
    % Copy all UIAxes children, take over axes limits and aspect ratio.
    allChildren = app.UIAxes.XAxis.Parent.Children;
    copyobj(allChildren, figAxes)
    figAxes.XLim = app.UIAxes.XLim;
    figAxes.YLim = app.UIAxes.YLim;
    figAxes.Title.String = app.UIAxes.Title.String;
    figAxes.Title.FontWeight = app.UIAxes.Title.FontWeight;
    figAxes.XLabel.String = app.UIAxes.XLabel.String;
    figAxes.YLabel.String = app.UIAxes.YLabel.String;
    figAxes.DataAspectRatio = app.UIAxes.DataAspectRatio;
    figAxes.XGrid = app.UIAxes.XGrid;
    figAxes.YGrid = app.UIAxes.YGrid;
    figAxes.YScale = app.UIAxes.YScale;

    app.graphname = app.ExportgraphnameEditField.Value;
    % Save as jpg files.
    saveas(fig, fullfile(app.exportpath, app.graphname), 'jpg');
    % Delete the temporary figure.
    delete(fig);
end

```

APPENDIX C

MATLAB coding for density matrix

```

clear

filename = 'sixteen_states_density_matrix';
hw = 100; %histogram_width
hw_m = 2; %merge hw
tau2 = 6; %delay correction
wt = 39; %windows time (bin)

timecollectcc = [0 340 675 1020 1360 1695 2070 2400 2755 3130 3470 3820 4170 4525 4880
5310];
ccbinwidth = 300;

%complier
fileID = fopen(filename);
C = textscan(fileID, '%f %f', 'Delimiter', ',');
fclose(fileID);
A=C{1}; %bin where the count triggered
B=C{2}; %port where the count triggered
nA = numel(A);
A=A-A(1); %set initial time to 0

%delay correction
for k = 1:nA
    if B(k) == 2
        A(k) = A(k)+tau2;
    end
end

diffA=diff(A); %dif of time in adjecent tile
difflo = find(diffA < hw+1); %find tile where dif < width
nlo = numel(difflo); %number of tiles in such condition

hisseccc = zeros(1, floor((A(end)-A(1))*81*(10^(-12))+1));
p=1;
for i = 1:nlo
    j=1; %tile seperation
    while A(difflo(i)+j)-A(difflo(i)) < hw+1
        if B(difflo(i)+j) ~= B(difflo(i))
            ccdif(p) = (A(difflo(i)+j)-A(difflo(i)))*(B(difflo(i)+j) - B(difflo(i)));
            if abs(ccdif(p)) <= wt
                seccc = floor((A(difflo(i)+j)-A(1))*81*(10^(-12))+1);
                hisseccc(seccc) = hisseccc(seccc) + 1;
            end
            p = p+1;
        end
        j=j+1;
    end
end

```

```

end

%calculate cc/sec
cc = nlo/((A(end)-A(1))*81*(10^(-12)));
cc12 = numel(find(abs(ccdif) <= wt))/((A(end)-A(1))*81*(10^(-12)));

%histogram(ccdif,100)
edges = -1*hw:hw_m:hw;
xdata = -1*hw+hw_m/2:hw_m:hw-hw_m/2;
timedata = xdata*0.081;
Nhis = histcounts(ccdif,edges);
r1 = sum(B(:) == 1);
r2 = sum(B(:) == 2);
div = r1*r2;
gtwo = Nhis./(div*81*10^(-12));
ccps = Nhis./((A(end)-A(1))*81*(10^(-12)));
%plot(timedata,ccps)
%xlabel('tau(ns)')
%ylabel('cc/sec')
%title('cc per sec vs time delay between port 1 and 2')
%text(max(timedata)*0.6,max(gtwo)*0.6,join([num2str(cc), ' cc/sec' ]))
%text(max(timedata)*0.6,max(ccps)*0.6,join([num2str(cc12), ' cc12/sec' ]))

%calculate single count
p1c = numel(find(B == 1))/((A(end)-A(1))*81*(10^(-12)));
p2c = numel(find(B == 2))/((A(end)-A(1))*81*(10^(-12)));

hisseccc(end) = [];
hisminutecc = zeros(1 ,floor((A(end)-A(1))*81*(10^(-12))/60) );
for m = 1:floor((seccc-1)/60)
    for n = 1:60
        hisminutecc(m) = hisminutecc(m) + hisseccc(((m-1)*60)+n);
    end
end
%plot(hisseccc)
%plot(hisminutecc)

%calculate cc bin
ncollecbin = numel(timecollectcc);
for l = 1:ncollecbin
    ccavg(l) = mean(hisseccc(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth));
    ccstd(l) = std(hisseccc(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth));
end

%calculate singlecount vs time
hissecnum1 = zeros(1 ,floor((A(end)-A(1))*81*(10^(-12))+1) );
hissecnum2 = zeros(1 ,floor((A(end)-A(1))*81*(10^(-12))+1) );
for t = 1:numel(A)

```

```

seccount = floor((A(t)-A(1))*81*(10^(-12))+1);
if B(t) == 1
    hisseccnum1(seccount) = hisseccnum1(seccount)+1;
elseif B(t) == 2
    hisseccnum2(seccount) = hisseccnum2(seccount)+1;
end
end
hisseccnum1(end)=[];
hisseccnum2(end)=[];
for l = 1:ncollecbin
    single1(l) = mean(hisseccnum1(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth));
    single1std(l) = std(hisseccnum1(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth));
    single2(l) = mean(hisseccnum2(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth));
    single2std(l) = std(hisseccnum2(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth));
end

netcc = hisseccc-hisseccnum1.*hisseccnum2*wt*81e-12;
for q = 1: numel(netcc)
    if or(or(hisseccnum1(q) < 3500,hisseccnum1(q) > 5000),or(hisseccnum2(q) < 3500,hisseccnum2(q)
> 5000))
        netcc(q) = 0;
        hisseccnum1(q) = 0;
        hisseccnum2(q) = 0;
    end
end
%plot(nonzeros(netcc))
%plot(nonzeros(hisseccnum1))
%plot(nonzeros(hisseccnum2))

for l = 1:ncollecbin
    netccavg(l) = mean(nonzeros(netcc(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth)));
    netccstd(l) = std(nonzeros(netcc(timecollectcc(l)+1:timecollectcc(l)+ccbinwidth)));
end

%%%%%%%%density matrix calculation
n = netccavg;

%%%%%%%%%defination
part%%%%%%%%%
%%%%%%%%%
%define polarization matrix
H = [1 0;0 0]; Hp = [1 ; 0];
V = [0 0;0 1]; Vp = [0;1];
D = (1/2)*[1 1;1 1]; Dp = (1/sqrt(2))*[1;1];
A = (1/2)*[1 -1;-1 1]; Ap = (1/sqrt(2))*[1;-1];
R = (1/2)*[1 1i;-1i 1]; Rp = (1/sqrt(2))*[1;-1i];
L = (1/2)*[1 -1i;1i 1]; Lp = (1/sqrt(2))*[1;1i];

```

```

phi(:,1) = kron(Hp,Hp);phi(:,2) = kron(Hp,Vp);phi(:,3) = kron(Vp,Vp);phi(:,4) = kron(Vp,Hp);
phi(:,5) = kron(Rp,Hp);phi(:,6) = kron(Rp,Vp);phi(:,7) = kron(Dp,Vp);phi(:,8) = kron(Dp,Hp);
phi(:,9) = kron(Dp,Rp);phi(:,10) = kron(Dp,Dp);phi(:,11) = kron(Rp,Dp);phi(:,12) =
kron(Hp,Dp);
phi(:,13) = kron(Vp,Dp);phi(:,14) = kron(Vp,Lp);phi(:,15) = kron(Hp,Lp);phi(:,16) =
kron(Rp,Lp);
%define M matrix (density matrix = sum ni*Mi)
M = (1/2)*[2 -1+1i -1-1i 1;-1-1i 0 1i 0;-1+1i -1i 0 0;1 0 0 0];
M(:,2) = (1/2)*[0 -1+1i 0 1; -1-1i 2 1i -1-1i; 0 -1i 0 0; 1 -1+1i 0 0];
M(:,3) = (1/2)*[0 0 0 1; 0 0 1i -1-1i; 0 -1i 0 -1+1i; 1 -1+1i -1-1i 2];
M(:,4) = (1/2)*[0 0 -1-1i 1;0 0 1i 0; -1+1i -1i 2 -1+1i;1 0 -1-1i 0];
M(:,5) = (1/2)*[0 0 2i -1-1i;0 0 1-1i 0;-2i 1+1i 0 0; -1+1i 0 0 0];
M(:,6) = (1/2)*[0 0 0 -1-1i;0 0 1-1i 2i; 0 1+1i 0 0;-1+1i -2i 0 0];
M(:,7) = (1/2)*[0 0 0 -1-1i;0 0 -1+1i 2;0 -1-1i 0 0;-1+1i 2 0 0];
M(:,8) = (1/2)*[0 0 2 -1-1i;0 0 -1+1i 0;2 -1-1i 0 0;-1+1i 0 0 0];
M(:,9) = [0 0 0 1i; 0 0 -1i 0;0 1i 0 0;-1i 0 0 0];
M(:,10) = [0 0 0 1; 0 0 1 0; 0 1 0 0; 1 0 0 0];
M(:,11) = [0 0 0 1i; 0 0 1i 0;0 -1i 0 0; -1i 0 0 0];
M(:,12) = (1/2)*[0 2 0 -1-1i; 2 0 -1-1i 0; 0 -1+1i 0 0; -1+1i 0 0 0];
M(:,13) = (1/2)*[0 0 0 -1-1i;0 0 -1-1i 0; 0 -1+1i 0 2; -1+1i 0 2 0];
M(:,14) = (1/2)*[0 0 0 -1+1i;0 0 1-1i 0;0 1+1i 0 -2i;-1-1i 0 2i 0];
M(:,15) = (1/2)*[0 -2i 0 -1+1i;2i 0 1-1i 0;0 1+1i 0 0;-1-1i 0 0 0];
M(:,16) = [ 0 0 0 1;0 0 -1 0;0 -1 0 0; 1 0 0 0];
%define measurement operator (measurment to get ni) %kron = tensor product
O = kron(Hp,Hp);
O(:,2) = kron(Hp,Vp);
O(:,3) = kron(Vp,Vp);
O(:,4) = kron(Vp,Hp);
O(:,5) = kron(Rp,Hp);
O(:,6) = kron(Rp,Vp);
O(:,7) = kron(Dp,Vp);
O(:,8) = kron(Dp,Hp);
O(:,9) = kron(Dp,Rp);
O(:,10) = kron(Dp,Dp);
O(:,11) = kron(Rp,Dp);
O(:,12) = kron(Hp,Dp);
O(:,13) = kron(Vp,Dp);
O(:,14) = kron(Vp,Lp);
O(:,15) = kron(Hp,Lp);
O(:,16) = kron(Rp,Lp);
%define fvui (for error calculation) for more info see error_003 file
fvui(:,1) = ...
[-2.0, 0, 0, -2.0, 0, 0, 0, 4.0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, -2.0, -2.0, 0, 0, 0, 4.0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 2.0, 2.0, 0, 0, 0, -4.0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
2.0, 0, 0, 2.0, 0, 0, 0, -4.0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...

```



```

0, -2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
-2.0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
-2.0, -2.0, 2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, -2.0, 2.0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0, 2.0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 4.0, 0, -2.0, -2.0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, -4.0, 0, 2.0, 2.0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, -4.0, -4.0, 4.0, 0, 0, 0, 2.0, 2.0, 0;...
0, 0, 0, 0, 0, 0, 4.0, 4.0, -4.0, 0, 0, 0, -2.0, -2.0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

```

fvui(:, :, 2) = ...

```

[ 0, 0, 0, 0, 2.0, 0, 0, -2.0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 2.0, -2.0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, -2.0, -2.0, 0, 0, 2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0;...
-2.0, 0, 0, -2.0, 2.0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0;...
-1.0, 0, 0, -1.0, 0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0;...
0, -1.0, -1.0, 0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, -2.0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, -2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, -2.0, -2.0, 2.0, 2.0, 0, 0, 0, 0, 0, 0, 2.0, 0, -2.0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, 0, -2.0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, 0, -1.0, -1.0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0, 2.0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, 2.0, -2.0, -2.0, 0, 0;...
0, 0, 0, 0, 0, 0, 2.0, 2.0, -2.0, 0, 0, 0, -2.0, -2.0, 2.0;...
0, 0, 0, 0, 0, 0, -2.0, -2.0, 2.0, 0, 0, 0, 0, 0, 2.0;...
0, 0, 0, 0, 0, 0, 2.0, 2.0, -2.0, 0, 0, 0, 0, -1.0, -1.0, 0];

```

fvui(:, :, 3) = ...

```

[ -2.0, -2.0, 0, 0, 0, 0, 0, 0, 0, 0, 4.0, 0, 0, 0, 0;...
2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, -4.0, 0, 0, 0, 0;...
0, 0, 2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, -4.0, 0, 0, 0;...
0, 0, -2.0, -2.0, 0, 0, 0, 0, 0, 0, 0, 4.0, 0, 0, 0;...
0, 0, 0, 0, -2.0, -2.0, 0, 0, 0, 0, 4.0, 0, 0, 0, 0;...
0, 0, 0, 0, 2.0, 2.0, 0, 0, 0, 0, -4.0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 2.0, 2.0, 0, -4.0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, -2.0, -2.0, 0, 4.0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 2.0, -2.0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, -2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
-2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 2.0, -2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

```

fvui(:, :, 4) = ...

```
[ 2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0, 0, 0, -2.0, 0;...
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, 0, 0, -2.0, 0;...
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, -2.0, 0, 0;...
  0, 0, 2.0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0, -2.0, 0, 0;...
  0, 0, 0, 0, 2.0, 2.0, 0, 0, 0, 0, -2.0, 0, 0, 0, 0, -2.0;...
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, 0, 0, 0, 0, -2.0;...
  0, 0, 0, 0, 0, 0, -2.0, -2.0, 2.0, 2.0, 0, 0, 0, 0, 0, 0;...
  0, 0, 0, 0, 0, 0, 0, 0, 2.0, -2.0, 0, 0, 0, 0, 0, 0;...
  0, 0, 0, 0, 0, 0, -1.0, -1.0, 0, 2.0, 0, 0, 0, 0, 0, 0;...
  0, 0, 0, 0, 0, 0, -2.0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0;...
  0, 0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0;...
  2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0, 0;...
  0, 0, 0, 2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2.0, 0, 0;...
  0, 0, -1.0, -1.0, 0, 0, 0, 0, 0, 0, 0, 2.0, 0, 0, 0, 0;...
  -1.0, -1.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.0, 0, 0, 0, 0;...
  0, 0, 0, 0, -1.0, -1.0, 0, 0, 0, 0, 2.0, 0, 0, 0, 0, 0];
```

```
sumop = [0 0 0 -1; 0 0 1 0; 0 1 0 0; -1 0 0 0];
```

```
%%%%%%%%%%end of definition
```

```
part%%%%%%%%%
```

```
%%%%%%%%%%Calculation
```

```
part%%%%%%%%%
```

```
%calculate (experimentally) density matrix
```

```
for i=1:16
```

```
    nM(:,i) = n(i)*M(:,i);
```

```
end
```

```
N = sum(n(1:4));
```

```
den = sum(nM,3)/N;
```

```
%create T0(T_zero)
```

```
T0(1,1) = sqrt(det(den)/minor(den,1,1));
```

```
T0(2,1) = minor(den,1,2)/sqrt(minor(den,1,1)*minor2(den,1,1,2,2));
```

```
T0(2,2) = sqrt(minor(den,1,1)/minor2(den,1,1,2,2));
```

```
T0(3,1) = minor2(den,1,2,2,3)/(sqrt(den(4,4))*sqrt(minor2(den,1,1,2,2)));
```

```
T0(3,2) = minor2(den,1,1,2,3)/(sqrt(den(4,4))*sqrt(minor2(den,1,1,2,2)));
```

```
T0(3,3) = sqrt(minor2(den,1,1,2,2)/den(4,4));
```

```
T0(4,1) = den(4,1)/sqrt(den(4,4));
```

```
T0(4,2) = den(4,2)/sqrt(den(4,4));
```

```
T0(4,3) = den(4,3)/sqrt(den(4,4));
```

```
T0(4,4) = sqrt(den(4,4));
```

```
%t0Cal
```

```
t0(1) = real(T0(1,1));
```

```
t0(2) = real(T0(2,2));
```

```
t0(3) = real(T0(3,3));
```

```
t0(4) = real(T0(4,4));
```

```
t0(5) = real(T0(2,1));
```

```

t0(6) = imag(T0(2,1));
t0(7) = real(T0(3,2));
t0(8) = imag(T0(3,2));
t0(9) = real(T0(4,3));
t0(10) = imag(T0(4,3));
t0(11) = real(T0(3,1));
t0(12) = imag(T0(3,1));
t0(13) = real(T0(4,2));
t0(14) = imag(T0(4,2));
t0(15) = real(T0(4,1));
t0(16) = imag(T0(4,1));

%fine the minimum of L function with t0 initial
options = optimset('PlotFcns',@optimplotfval,'TolFun',1.e-
13,'MaxFunEvals',10000000,'MaxIter',5e5);
tans = fminsearch(@(t)genL(t,O,n), t0, options);
%create physically density matrix
Tans = [tans(1) 0 0 0;tans(5)+tans(6)*1i tans(2) 0 0;tans(11)+tans(12)*1i ...
        tans(7)+tans(8)*1i tans(3) 0;tans(15)+tans(16)*1i tans(13)+tans(14)*1i ...
        tans(9)+tans(10)*1i tans(4)];
den_p = (Tans'*Tans)/trace(Tans'*Tans)

%calSv
for i = 1:16
    Sv(i) = real(phi(:,i)*den_p*phi(:,i));
end
%error calculation
for v = 1:16
    sumu1u2i =0;
    for ir = 1:4
        sumu1u2 = 0;
        sumfortest(v,ir) = 0;
        for u1 = 1:16
            for u2 = 1:16
                sumu1u2 = sumu1u2 + (0.5*pi/180)^2*fvui(v,u1,ir)*fvui(v,u2,ir)*n(u1)*n(u2)/(N^2);
                sumfortest(v,ir) = sumfortest(v,ir) +
(0.5*pi/180)^2*fvui(v,u1,ir)*fvui(v,u2,ir)*n(u1)*n(u2)/(N^2);
            end
        end
        sumu1u2i = sumu1u2i + sumu1u2;
    end
    lamV(v) = (n(v)/(N^2))+sumu1u2i;
end
%density matrix error
for i =1:4
    for j=1:4
        deldensqr(i,j) = 0;
    end
end

```

```

        deldensqr(i,j) = deldensqr(i,j)+ (M(i,j,v))^2*lamV(v);
    end
end
end
delden = sqrt(deldensqr);

[evect,eval] = eig(den_p);
%Von Neumann Entropy
S = 0;
for a=1:4
    S = S-eval(a,a)*log2(eval(a,a));
end
delSsqr = 0;
for v = 1:16
    sumphiM = 0;
    for a = 1:4
        sumphiM = sumphiM + (evect(:,a)*M(:,v)*evect(:,a)*(1+log(eval(a,a)))/log(2));
    end
    delSsqr = delSsqr + lamV(v)*sumphiM^2;
end
delS = sqrt(delSsqr);

%linear entropy
P = 0;
for a = 1:4
    P = P + (eval(a,a))^2;
end
P = (4/3)*(1-P);
delPsqr = 0;
for v = 1:16
    sumtrS = 0;
    for u = 1:16
        sumtrS = sumtrS + trace(M(:,v)*M(:,u))*n(u)/N;
    end
    delPsqr = delPsqr + (sumtrS*(8/3))^2*lamV(v);
end
delP = sqrt(delPsqr);

%concurrence
Rmat = zeros(4);
for u = 1:16
    for v = 1:16
        qmat(:,u,v) = M(:,u)*sumop*transpose(M(:,v))*sumop...
            + M(:,v)*sumop*transpose(M(:,u))*sumop;
        Rmat = Rmat + qmat(:,u,v)*Sv(u)*Sv(v);
    end
end
Rmat = Rmat/2;

```

```

[revec, reval] = eig(Rmat);
[r, ind] = sort(diag(reval), 'descend');
c = max([0, sqrt(r(1))-sqrt(r(2))-sqrt(r(2))-sqrt(r(3))]);
T = c^2; %tangle
E = h((1+sqrt(1-c^2))/2); %entangle

%HVpVH = [0 0 0 0; 0 0.5 0.5 0; 0 0.5 0.5 0; 0 0 0 0];
%fidelity = (trace(sqrtm(sqrtm(HVpVH)*den_p*sqrtm(HVpVH))))^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%End of calculation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%underneath here are function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

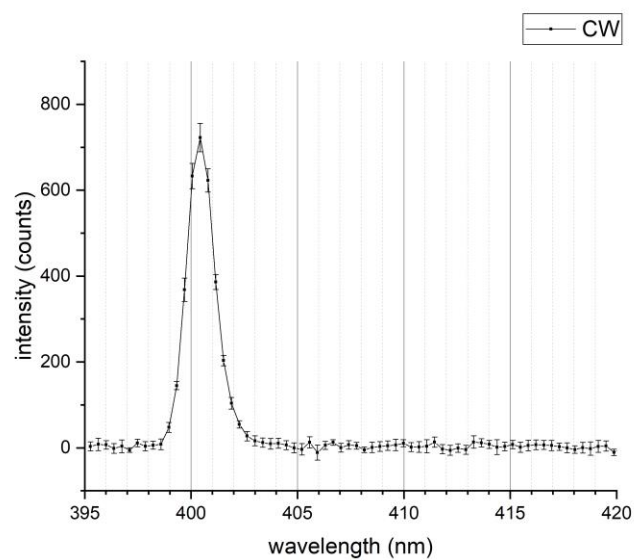
%define likelihood fuction L(t1,t2,...,t16)
function L = genL(t,O,n)
T = [t(1) 0 0 0;t(5)+t(6)*1i t(2) 0 0;t(11)+t(12)*1i t(7)+t(8)*1i t(3) ...
    0;t(15)+t(16)*1i t(13)+t(14)*1i t(9)+t(10)*1i t(4)];
den_p = (T'*T)/trace(T'*T);
N = sum(n(1:4));
for j = 1:16
    den_p_measure(j) = real(O(:,j)*den_p*O(:,j));
    L_mat(j) = ((N*den_p_measure(j)-n(j))^2)/(2*N*den_p_measure(j));
end
L = sum(L_mat);
end
%define minor function
function Det_minor=minor(x,p,q) %det of x matrix without p-row and q column
x0= x;
x0(p,:) = [];
x0(:,q) = [];
Det_minor = det(x0);
end
%define minor2 function
function Det_minor2=minor2(x,p,q,r,s) %without p,r row and q,s column
x0= x;
x0([p r],:) = [];
x0(:, [q s]) = [];
Det_minor2 = det(x0);
end
function hcal = h(x)
hcal = -x*log2(x)-(1-x)*log2(1-x);
end

```

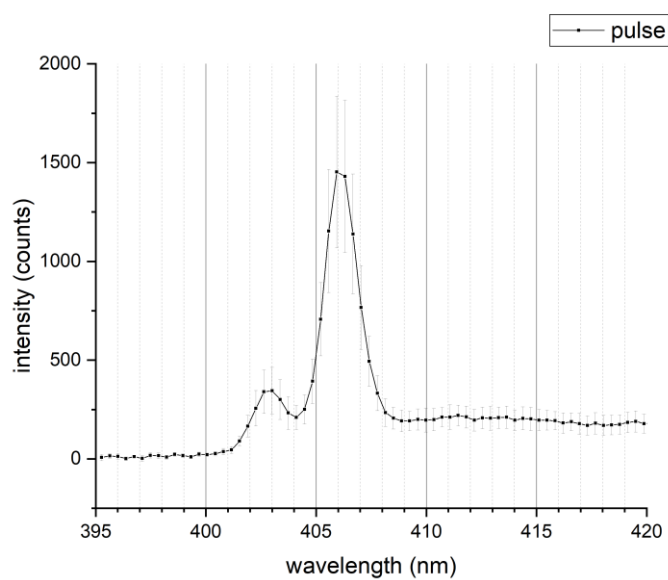
APPENDIX D

Laser source specifications

CW laser spectrum



Pulse laser spectrum



APPENDIX E

Paper : Pewkhom, P., S. Suwana, and Pruet Kalasuwan. "Alternative Scheme of Universal Optical Programmable Multi-Qubit Gates for Polarization Qubits." *Quantum Information Processing* 19 (June 8, 2020).



Alternative scheme of universal optical programmable multi-qubit gates for polarization qubits

P. Pewkhom^{1,2} · S. Suwanna³ · P. Kalasuwan^{1,2} 

Received: 7 August 2019 / Accepted: 12 May 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

We propose an alternative scheme for programmable quantum gates specifically designed for polarization qubits of single photons. Unlike the conventional approach based on the scheme of Reck et al., we adopt the Hilbert-space expansion technique to enable a novel feature in the design of arbitrary quantum operations. We present a scheme that independently programs each matrix element of an operator. The proposed scheme can support the realization of many kinds of quantum operations. In anticipation of advances in the current quantum photonics technology, we assume the manipulation and detection of many photon qubits and propose a theoretical scheme to achieve n -polarization-qubit optical reconfigurable quantum gates by linear optical elements.

Keywords Programmable gates · Qubit processing · Universal linear optics

1 Introduction

The linear transformation of inputs to outputs is the heart of information processing in computer science. In order to realize universal information processing, highly versatile logic-gate devices are required to perform linear mapping between universal sets of inputs and outputs. Quantum computers work under the laws of quantum mechanics, using gate operations and information represented by quantum bits or *qubits*. They only need accessible single- and two-qubit gate operations to realize any unitary quantum gate [5]. During recent decades, various quantum approaches have experimentally

✉ P. Kalasuwan
pruet.k@psu.ac.th

¹ Department of Physics, Faculty of Science, Prince of Songkla University, Hat-Yai, Songkhla 90112, Thailand

² ThEP, Commission of Higher Education, Bangkok 10400, Thailand

³ Optical and Quantum Physics Laboratory, Department of Physics, Faculty of Science, Mahidol University, Bangkok 10400, Thailand

proved the successful operation of single- and two-qubit gates [9,10,13,15,20]. Beyond single- and two-qubit gates, verification tests in recent years produced decent outcomes from quantum processors housing programmable features [3,7]. These achievements marked a critical advance in the field as they paved the way forward to future practical machines.

The use of photonic qubits is one of the most promising approaches to quantum information processing. In general, photonic qubits enable free-of-noise and well-established single-qubit operations. The only drawback that prevents photons from performing efficient scalable processing is their negligible photon–photon interaction. However, the problem can be overcome by the introduction of nonlinear optical operations [15,17]. This approach was a fundamental element of an optical quantum processor circuit based on a cascade of beam splitters and phase shifters implanted on a single-photon chip [3,18]. The circuit could potentially be programmed to perform any quantum unitary operation of path qubits on a fixed circuit. Furthermore, this design could possibly be modified to work with other degrees of freedom of photons, such as polarization [12].

The standard method [3,19] for constructing an N -qubit unitary gate is based on the Reck et al. scheme, where an arbitrary unitary matrix $U(N)$ is factorized into a product of block matrices, where successive $U(2)$ transformations can be performed on two-dimensional subspaces of the full N -dimensional Hilbert space. This architecture can adequately fulfill the functions of a programmable quantum gate, where the required circuit will need to be decomposed by the optical elements into the functions of reflectivity and phase shift to provide the programming input, and has been realized experimentally with good fidelity [3]. Moreover, the algorithm is recursive, similar to the Gaussian elimination process, and can increase the complexity of realizing an arbitrary quantum gate. More importantly, such the standard scheme is limited to unitary operation where the operator is known, hence can be decomposed into single- and two-qubit gates. However, in the emergence of quantum technology, a fundamental tool in quantum metrology may require a phase estimation where the estimation algorithm U is an unknown black box, which may not be decomposed at all [2,24], particularly when the quantum phase estimation requires a conditional gate of the operator U .¹ Therefore, the current standard method of realizing quantum gates based on the decomposition into elementary gates may not apply for these applications.

In this paper, we propose a more versatile class of polarization-encoded qubit processors. We employ the Hilbert-space expansion technique in linear optical circuits, in which the linear combination of operations on the polarization states is available. This capability was already reported in previous experiments [18,24] for small number of path-encoding qubits. Here, we adapt such technique for the polarization qubits and simplify the circuit by designing some operators to make the circuit suitable for possibly the lowest complexity realization of the programmable multi-qubit gates. The concept starts from the superposition state of photon pairs from separate nonlinear elements such as BBO to produce two pairs of photons in four spatial paths. When recombining two photons from separate nonlinearities, assuming that they have under-

¹ In quantum metrology, normally the form of the operator is available, though it is parametrized by an unknown parameter (e.g., $U(x)$ with the unknown parameter x). In such case, one could still attempt the decomposition, even when in general this is probably impractical.

gone different linear transformations, the signal at the detector will be a combination of those transformations. By post-selecting appropriate outputs where only one photon leaves each path, all entries of the operator can be obtained.

This schematic design is different from the standard decomposition of an operation, which mostly relies on the products of operators. Using this alternative technique, instead of combining several single-qubit and two-qubit gates to produce a universal quantum gate, we directly program every entry of the matrix representing the operation. The manipulation of a quantum gate is also possible for all kinds of quantum operations, possibly represented by complex matrices, including non-unitary ones. The detail of the technique is reviewed in Sect. 2, where we also describe an improved procedure for constructing an optical programmable two-qubit gate. In Sect. 3, we explain how to realize optical programmable multi-qubit gates by this alternative approach and, in the last section, we discuss their advantages and technical requirements for practical implementation.

2 Hilbert-space expansion technique for two polarization qubits

Universal quantum computers need single-qubit and two-qubit gates realized by particular physical systems [2]. The single-qubit gate of the polarization state of light is simply given by a cascade of birefringent materials, such as half-waveplates (HWP) and quarter-waveplates (QWP). Peters et al. [16] demonstrated that single-qubit gates comprising two HWPs and a QWP had great versatility, yielding all possible unitary single-qubit gates. In addition, non-unitary single-qubit gates were also possible with the use of an interferometer to introduce decoherence into the optical circuit.

Two-qubit gates are also of primary interest as they are far more difficult to achieve with photon qubits. Here, we develop optical decomposition to realize any possible two-qubit gate. First of all, any two-qubit gate can be expressed as a linear combination of tensor products of single-qubit gates, which can be written as $\sum_i \hat{O}_{1i} \otimes \hat{O}_{2i}$. Using four separate single-qubit gates $\hat{\gamma}_{00}$, $\hat{\gamma}_{01}$, $\hat{\gamma}_{10}$, $\hat{\gamma}_{11}$, which were realized experimentally in [16], a two-qubit gate can be mathematically decomposed as

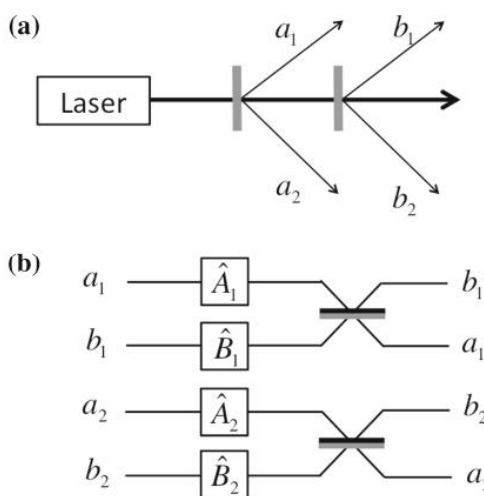
$$|0\rangle\langle 0| \otimes \hat{\gamma}_{00} + |0\rangle\langle 1| \otimes \hat{\gamma}_{01} + |1\rangle\langle 0| \otimes \hat{\gamma}_{10} + |1\rangle\langle 1| \otimes \hat{\gamma}_{11}, \quad (1)$$

which implies that once programmable single qubits are realized, the programmable two-qubit gates are also.

To realize such two-qubit gates using photons, we started from previous works that dealt with the linear combination of quantum operators [18,24]. The approach requires two pairs of type-I spontaneous parametric down-conversion (SPDC) processes (Fig. 1a) and two symmetrical beam splitters (BS) to merge four spatial modes of photons resulting from the SPDCs (Fig. 1b). Once a laser beam is pumped to two nonlinear crystals, as presented in Fig. 1a, two pairs of identical photons are produced, whose states can be written in the polarization states as

$$\frac{1}{\sqrt{2}}(|\psi\rangle_{a_1}|\psi\rangle_{a_2} + |\psi'\rangle_{b_1}|\psi'\rangle_{b_2}) \quad (2)$$

Fig. 1 **a** SPDC source to produce two pairs of entangled photons resulting as (2). **b** Experimental scheme to realize the two-qubit operation $\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2$ according to the input state (3), which is derived from the source (2)



where $a_1, a_2, b_1,$ and b_2 denote the spatial paths of the photons, and ψ and ψ' are the polarization states of each photon pair. After the SPDCs, by priorly knowing of the possible states from SPDCs, we can transform the output states to

$$\frac{1}{\sqrt{2}}(|\psi_1\rangle_{a_1}|\psi_2\rangle_{a_2} + |\psi_1\rangle_{b_1}|\psi_2\rangle_{b_2}) \tag{3}$$

using waveplates. The polarization states ψ_1 on the paths a_1 and b_1 are set identically and constitute the states of the first qubit. Similarly, the states ψ_2 on the paths a_2 and b_2 constitute the states of the second qubit. Afterward, the photon on each spatial path is operated by the single-qubit gates $\hat{A}_1, \hat{A}_2, \hat{B}_1, \hat{B}_2,$ and then the paths are merged by two BSs. Since each state is actually the hyper-entanglement of paths and polarization qubits, a beam splitter produces separate paths of photons but conserves their polarization. Consequently, two BSs (Fig. 1b) transform the states in (3) as follows:

$$|\psi_1\rangle_{a_1} \rightarrow \frac{1}{\sqrt{2}} \left(\hat{A}_1|\psi_1\rangle_{a_1} + \hat{A}_1|\psi_1\rangle_{b_1} \right), \tag{4}$$

$$|\psi_1\rangle_{b_1} \rightarrow \frac{1}{\sqrt{2}} \left(\hat{B}_1|\psi_1\rangle_{a_1} - \hat{B}_1|\psi_1\rangle_{b_1} \right), \tag{5}$$

$$|\psi_2\rangle_{a_2} \rightarrow \frac{1}{\sqrt{2}} \left(\hat{A}_2|\psi_2\rangle_{a_2} + \hat{A}_2|\psi_2\rangle_{b_2} \right), \tag{6}$$

$$|\psi_2\rangle_{b_2} \rightarrow \frac{1}{\sqrt{2}} \left(\hat{B}_2|\psi_2\rangle_{a_2} - \hat{B}_2|\psi_2\rangle_{b_2} \right). \tag{7}$$

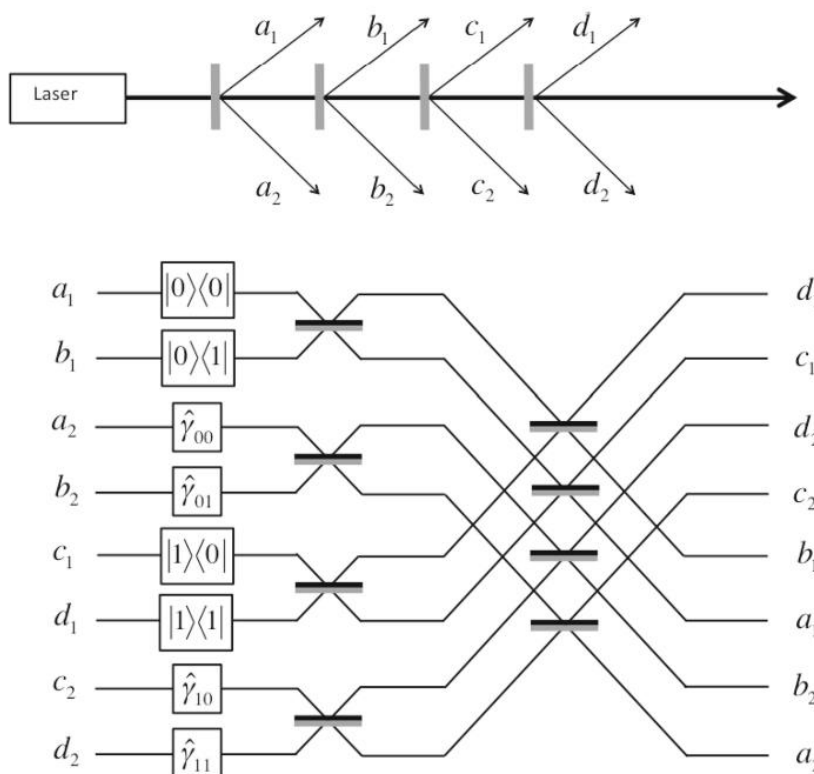


Fig. 2 **a** A nonlinear single-photon source gives the superposition of photon states (10). **b** A linear optical circuit to perform the linear combination of four tensor products of single-qubit gates in (12). The photons in the circuit are delivered from eight paths in Fig. 2a. Before entering the circuit, all photons are initialized as the superposition of the identical input polarization states on different spatial paths in (10)

As a result of post-selecting photons in the paths a_1, a_2 or b_1, b_2 , the final state of photons becomes

$$\frac{1}{\sqrt{2}}(\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2)|\psi_1\rangle|\psi_2\rangle \tag{8}$$

The quantum operator $\hat{A}_1 \otimes \hat{A}_2 + \hat{B}_1 \otimes \hat{B}_2$ is realized with a probability of $\frac{1}{4}$.

In order to achieve (1), the construct of four SPDCs, eight spatial paths, and eight BSs is presented in Fig. 2. The circuit is indeed a scaled-up version of Fig. 1. Similar to (2), we can expect the state

$$\frac{1}{2}(|\psi_a\rangle_{a_1}|\psi_a\rangle_{a_2} + |\psi_b\rangle_{b_1}|\psi_b\rangle_{b_2} + |\psi_c\rangle_{c_1}|\psi_c\rangle_{c_2} + |\psi_d\rangle_{d_1}|\psi_d\rangle_{d_2}) \tag{9}$$

from four SPDCs (Fig. 2a). If we assume that the polarization states in (9) are known, we can generate an output state for the two qubits before entering the circuit of Fig. 2b

as

$$\frac{1}{2}(|\psi_1\rangle_{a_1}|\psi_2\rangle_{a_2} + |\psi_1\rangle_{b_1}|\psi_2\rangle_{b_2} + |\psi_1\rangle_{c_1}|\psi_2\rangle_{c_2} + |\psi_1\rangle_{d_1}|\psi_2\rangle_{d_2}). \quad (10)$$

In the first stage of the circuit, by using $\hat{A}_1 = |0\rangle\langle 0|$, $\hat{B}_1 = |0\rangle\langle 1|$, $\hat{C}_1 = |1\rangle\langle 0|$, $\hat{D}_1 = |1\rangle\langle 1|$, $\hat{A}_2 = \hat{\gamma}_{00}$, $\hat{B}_2 = \hat{\gamma}_{01}$, $\hat{C}_2 = \hat{\gamma}_{10}$, and $\hat{D}_2 = \hat{\gamma}_{11}$, functioning as the programmable apparatus, (10) leads to

$$\begin{aligned} & \frac{1}{2} \left[(|0\rangle\langle 0|_{\psi_1})_{a_1} \otimes (\hat{\gamma}_{00}|\psi_2\rangle_{a_2}) + (|0\rangle\langle 1|_{\psi_1})_{b_1} \otimes (\hat{\gamma}_{01}|\psi_2\rangle_{b_2}) \right. \\ & \left. + (|1\rangle\langle 0|_{\psi_1})_{c_1} \otimes (\hat{\gamma}_{10}|\psi_2\rangle_{c_2}) + (|1\rangle\langle 1|_{\psi_1})_{d_1} \otimes (\hat{\gamma}_{11}|\psi_2\rangle_{d_2}) \right]. \end{aligned} \quad (11)$$

Finally, using the relation between inputs and outputs expressed in (4)–(7) together with the post-selection measurement of photon coincidences in the paths (a_1, a_2) , (b_1, b_2) , (c_1, c_2) or (d_1, d_2) , the final state becomes

$$\frac{1}{2}(|0\rangle\langle 0| \otimes \hat{\gamma}_{00} + |0\rangle\langle 1| \otimes \hat{\gamma}_{01} + |1\rangle\langle 0| \otimes \hat{\gamma}_{10} + |1\rangle\langle 1| \otimes \hat{\gamma}_{11})|\psi_1\rangle|\psi_2\rangle, \quad (12)$$

with a probability of 1/4. In particular, this yields a two-qubit operation, namely

$$\begin{bmatrix} \hat{\gamma}_{00} & \hat{\gamma}_{01} \\ \hat{\gamma}_{10} & \hat{\gamma}_{11} \end{bmatrix} |\psi_1\rangle|\psi_2\rangle, \quad (13)$$

where all the entries of the 4×4 matrix are programmable.

3 Universal optical programmable multi-qubit gates

The recent experimental creation of three photon polarization GHZ states from SPDC [8] suggests that we can expect to access more photonic quantum processing with polarization-entangled photons in the coming years. Apart from other sophisticated approaches to photon entanglement, SPDC from nonlinear crystals in bulk optics setups, as presented here, can play a role as the simplest possible source of entangled photons. Suppose that the n -photon entangled state

$$\frac{1}{\sqrt{2}}(|\psi_1\rangle_{o_{11}}|\psi_2\rangle_{o_{12}} \cdots |\psi_n\rangle_{o_{1n}} + |\psi_1\rangle_{o_{21}}|\psi_2\rangle_{o_{22}} \cdots |\psi_n\rangle_{o_{2n}}) \quad (14)$$

is accessible from the source depicted in Fig. 3 and that the n -photon coincidence counting operates with high efficiency, our architecture is able to accommodate programmable n -qubit gates. The scheme in Fig. 3 is actually the n -photon version of the paired-photon source in Fig. 1a, in which the paths a_1, a_2, b_1 , and b_2 are replaced, respectively, by o_{11}, o_{12}, o_{21} , and o_{22} for generalization; and ES represents the entanglement source. For clarity in notation, two indices, say m and p of the path o_{mp} ,

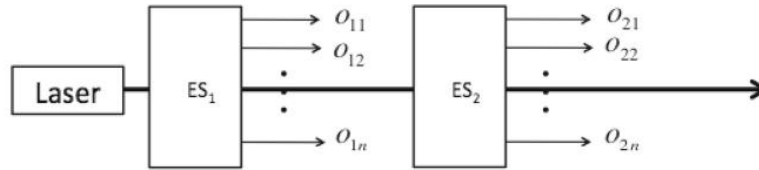
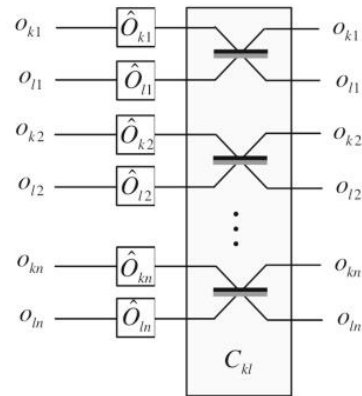


Fig. 3 Two coupled notional single-photon sources (ES_1 and ES_2) provide n entangled photons. The sources give the polarization states as (14)

Fig. 4 A combination block (C_{kl}), a series of beam splitters combining photons from ES_k and ES_l , supposing each ES produces n distinguishable single photons



indicate photons from the p^{th} source (ES_p) and the m^{th} qubit, which will be later encoded into the polarization state $|\psi_m\rangle$.

In order to generalize all operations in the previous section to the n -qubit case, we introduce a combination block (C_{kl}); see Fig. 4. Here, the C_{kl} block consists of BSs, where each BS fuses the two paths sharing the second index, e.g., from sources ES_k and ES_l in this case. When the single-qubit gate \hat{O}_{mp} operates on each of its corresponding paths o_{mp} , the C_{kl} block transforms the initial state

$$\frac{1}{\sqrt{2}}(|\psi_1\rangle_{o_{k1}}|\psi_2\rangle_{o_{k2}} \cdots |\psi_n\rangle_{o_{kn}} + |\psi_1\rangle_{o_{l1}}|\psi_2\rangle_{o_{l1}} \cdots |\psi_n\rangle_{o_{ln}}) \tag{15}$$

to

$$\begin{aligned} & \frac{1}{(\sqrt{2})^{n+1}} [(\hat{O}_{k1}|\psi_1\rangle_{o_{k1}} + \hat{O}_{l1}|\psi_1\rangle_{o_{l1}}) \otimes (\hat{O}_{k2}|\psi_2\rangle_{o_{k2}} \\ & + \hat{O}_{l2}|\psi_2\rangle_{o_{l2}}) \otimes \cdots \otimes (\hat{O}_{kn}|\psi_n\rangle_{o_{kn}} + \hat{O}_{ln}|\psi_n\rangle_{o_{ln}}) \\ & + (\hat{O}_{l1}|\psi_1\rangle_{o_{k1}} - \hat{O}_{l1}|\psi_1\rangle_{o_{l1}}) \otimes (\hat{O}_{l2}|\psi_2\rangle_{o_{k2}} \\ & - \hat{O}_{l2}|\psi_2\rangle_{o_{l2}}) \otimes \cdots \otimes (\hat{O}_{ln}|\psi_n\rangle_{o_{kn}} - \hat{O}_{ln}|\psi_n\rangle_{o_{ln}})]. \end{aligned} \tag{16}$$

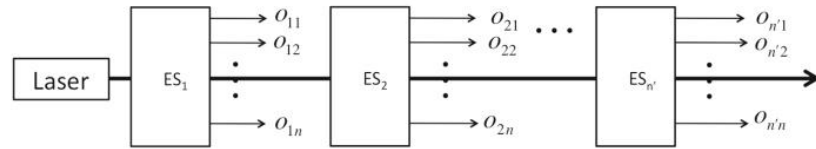


Fig. 5 A cascade of n' ESs, where each ES produces n entangled single photons in separate paths, resulting the initial state as expressed in (20)

Thereafter, by post-selecting the case where only one photon leaves each path of $o_{k1} - o_{kn}$ simultaneously, we will obtain the final state

$$(\hat{O}_{k1} \otimes \hat{O}_{k2} \otimes \cdots \otimes \hat{O}_{kn} + \hat{O}_{l1} \otimes \hat{O}_{l2} \otimes \cdots \otimes \hat{O}_{ln}) \times |\psi_1\rangle_{o_{k1}} |\psi_2\rangle_{o_{k2}} \cdots |\psi_n\rangle_{o_{kn}}, \tag{17}$$

with a success probability of $(1/\sqrt{2})^{n+1}$. Likewise, by post-selection on all l -paths, we obtain

$$(\hat{O}_{k1} \otimes \hat{O}_{k2} \otimes \cdots \otimes \hat{O}_{kn} - \hat{O}_{l1} \otimes \hat{O}_{l2} \otimes \cdots \otimes \hat{O}_{ln}) \times |\psi_1\rangle_{o_{l1}} |\psi_2\rangle_{o_{l2}} \cdots |\psi_n\rangle_{o_{ln}} \tag{18}$$

with a success probability of $(1/\sqrt{2})^{n+1}$. The realization of (17) is a cornerstone of the complete realization of the n -qubit programmable operation.

First of all, any n -qubit operation is represented by a $2^n \times 2^n$ matrix and can be decomposed into a linear combination of tensor products written as

$$\sum_{i=1}^{2^{n-1}} \sum_{j=1}^{2^{n-1}} \hat{\gamma}_{(i,j)} \otimes |i\rangle\langle j|, \tag{19}$$

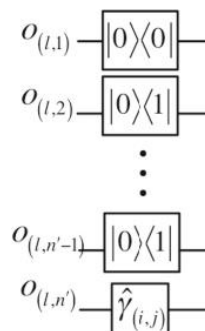
where $\hat{\gamma}_{(i,j)}$ is a 2×2 operator as defined in (1), and all of its matrix elements are programmed by the programmable single-qubit gates. Here, $|i\rangle\langle j|$ represents a basis for each matrix element of the $2^{n-1} \times 2^{n-1}$ all-ones matrix.

As the initial state in (14) from the source in Fig. 3 is necessary to realize the summation of two terms of tensor products of n single-qubit gates, we will, at least, need the kind of source illustrated in Fig. 5 to complete the summation of $n' = (2^{n-1})^2$ terms following (19). The quantum state is then written as

$$\frac{1}{\sqrt{n'}} \sum_{k=1}^{n'} \bigotimes_{l=1}^n |\psi_l\rangle_{o_{kl}}. \tag{20}$$

From this stage of the scheme, photons from each ES play a role as each term in (19) arising from applying the local operator (Fig. 6). The local operators applied on paths $o_{l,1}, o_{l,2}, \dots, o_{l,n-1}$ are the members of $\{|0\rangle\langle 0|, |0\rangle\langle 1|, |1\rangle\langle 0|, |1\rangle\langle 1|\}$. A tensor

Fig. 6 The example of the local operations applied to the photon state after being initialized to the state (20) on each ES output (ES_l for this example). This example represents just one term in (19). In order to assign the position (i, j) of the single-qubit operation $\hat{Y}_{(i,j)}$ on the $(2^{n-1}) \times (2^{n-1})$ all-ones matrix, the local operations representing the position are applied on the paths $o_{(l,1)}, o_{(l,2)}, \dots, o_{(l,n'-1)}$. In this example, the operators relating to entries on the all-ones matrix follow from (21)



product of $n' - 1$ terms from this set of operators results in a $(2^{n-1}) \times (2^{n-1})$ matrix with all zero entries except the element (i, j) . The relation between the tensor product and (i, j) follows the relation

$$|0\rangle\langle 1| \otimes |1\rangle\langle 0| \otimes \dots \otimes |1\rangle\langle 0| \equiv |01 \dots 1\rangle\langle 10 \dots 0| = |i\rangle\langle j|, \tag{21}$$

where i and j correspond to decimal equivalents of the binary representations $01 \dots 1$ and $10 \dots 0$, respectively. The summation of two separate terms building from two ESS and the state preparation schemes in Fig. 6 is presented in Fig. 7.

Finally, using the combination blocks to merge all paths (Fig. 8) and post-selecting only photons in path $o''(1, n)$, the operation (19) is realized, obtaining the n -qubit operation represented by the matrix

$$\begin{bmatrix} \hat{Y}_{(1,1)} & \hat{Y}_{(1,2)} & \hat{Y}_{(1,3)} & \dots & \hat{Y}_{(1,2^{n-1})} \\ \hat{Y}_{(2,1)} & \hat{Y}_{(2,2)} & \hat{Y}_{(2,3)} & \dots & \hat{Y}_{(2,2^{n-1})} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{Y}_{(2^{n-1},1)} & \hat{Y}_{(2^{n-1},2)} & \hat{Y}_{(2^{n-1},3)} & \dots & \hat{Y}_{(2^{n-1},2^{n-1})} \end{bmatrix} \tag{22}$$

with a success probability of $(1/\sqrt{2})^n$.

4 Discussion and conclusion

We have presented an alternative optical circuit design which can achieve the complete versatility in a programmable multi-qubit gate. By directly programming the entries of the matrix by a single-qubit gate, our proposed scheme circumvents the need to decompose the required gates into elementary single- and two-qubit gates in the standard method of realizing any unitary quantum operator. Accordingly, this alternative approach has potential to accelerate quantum simulation and information processing because it does not require iteration as does the standard method. Due to

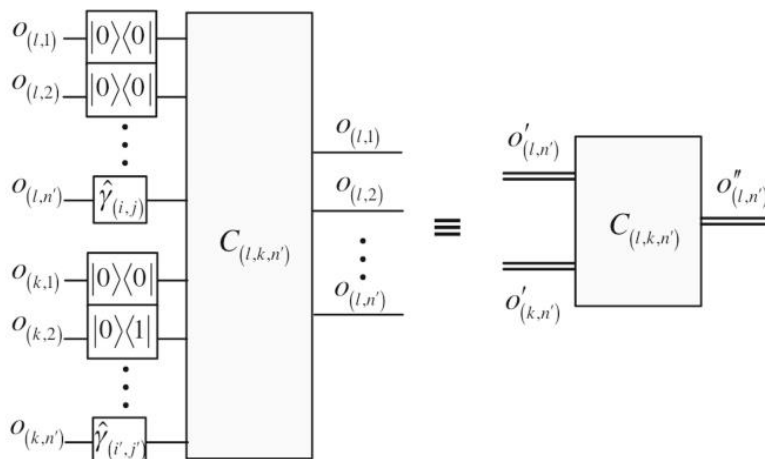


Fig. 7 A modified combination block. Paths $o_{(p,m)}$ sharing p are grouped as a composite path $o'_{(p,n')}$ with a set of local operations in Fig. 6, where $m = 1, 2, \dots, n'$. In this case, the composite paths $o'_{(l,n')}$ and $o'_{(k,n')}$ are combined by the modified combination block $C_{(l,k,n')}$, resulting in the composite path $o''_{(l,n')}$. This output composite path is the post-selection of the photons leaving at the output paths $o_{(l,1)}, o_{(l,2)}, \dots, o_{(l,n'-1)}$

the increased feasibility of controlling every entry of the matrix independently by the linear optics, it also provides great capacity to generate the whole sets of operations, including complex and non-unitary operations.

However, the demonstrated quantum gate based on this scheme is probabilistic and will undergo an exponentially small probability of success. In this sense, we can regard this alternative scheme as a way of trading the decoherence for the power of generating any operator. One can argue that ideally the alternative scheme can be used to probe noise or decoherence since the matrix entries are independently programmed and can be complex; in particular, the off-diagonal terms can be probed systematically. This arguably can yield great benefits in decoherence suppression and noise spectroscopy in metrology, and possibly in errors correction in quantum communication. Furthermore, the gate-based computational using a small number of qubits is rather possible, and such the class of computation can shed some light on problems in fundamental concepts of quantum information requiring arbitrary quantum gate. Examples of such scenarios, such as controlling operation of an unknown $U(2)$, the complete quantum phase-estimation algorithm via semiclassical Fourier transform, entanglement filters, the quantum approximate optimization algorithm, and Szegedy quantum walks, have been demonstrated already [18,23,24].

Perhaps, it is more insightful and useful to compare our proposed scheme to the standard method [3,19] which relies on a decomposition of an $N \times N$ unitary matrix into a product of 2×2 matrices, which then can be determined by using successive 2×2 transformations. Both approaches require linear optical devices to program a multi-qubit gate; the standard method uses phase shifters and beam splitters on

two-dimensional subspaces, and iteratively expands to the full N -dimensional Hilbert space, while our scheme uses entanglement photon sources and beam splitters to create all possible linear combinations of tensor products in the entire Hilbert space, and employ post-selection to program the operator. To quantitatively analyze the complexity of the two approaches, we firstly note that, in order to realize a $U(2^m)$, for $m \geq 2$, it will need to be factorized into a multiplication of $2^m(2^m - 1)/2$ of $U(4)$:

$$U(2^m) = \left[\prod_{i=1}^{\frac{2^m(2^m-1)}{2}} U_i(4) \right] D \tag{23}$$

where D is a diagonal matrix with elements of modulus 1. There are several classical algorithms to perform this type of factorization. However, preparing any arbitrary $N \times N$ gates could endure exponential growth complexity as a result of this requirement. Comparing to our scheme which is build from summation of matrices $U'(4)$ given in (24), the number of required $U(4)$ operations is 4^{m-1} which is just slight improvement.

$$U(2^m) = \sum_{i=1}^{4^{m-1}} \hat{U}'_i(4), \tag{24}$$

or, equivalently

$$\begin{aligned} U(2^m) &= \begin{bmatrix} \hat{U}_1(4) & \hat{U}_2(4) & \dots & \hat{U}_{(2^{m-1})}(4) \\ \hat{U}_{(2^{m-1}+1)}(4) & \hat{U}_{(2^{m-1}+2)}(4) & \dots & \hat{U}_{(2^m)}(4) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{U}_{(4^{m-1}-2^{m-1})}(4) & \hat{U}_{(4^{m-1}-2^{m-1}+1)}(4) & \dots & \hat{U}_{(4^{m-1})}(4) \end{bmatrix} \\ &= \begin{bmatrix} \hat{U}_1(4) & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 0 & \hat{U}_2(4) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} + \dots \\ &\quad + \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{U}_{(4^{m-1})}(4) \end{bmatrix}. \end{aligned}$$

However, with no requirement of factorization of the matrix, our scheme can avoid a large growth in complexity. The linear combination is the main strength of gate structure based on X. Qiang et al.'s scheme in Ref. [18], like our scheme. The prepared $U'(4)$ s to realize a whole matrix of $U(2^m)$ can be simply putted directly in the circuit as shown in Figs. 6 and 7.

Clearly, there is a trade-off between iteration which can increase complexity and post-selection which yields low probability. For a small number of qubits, both

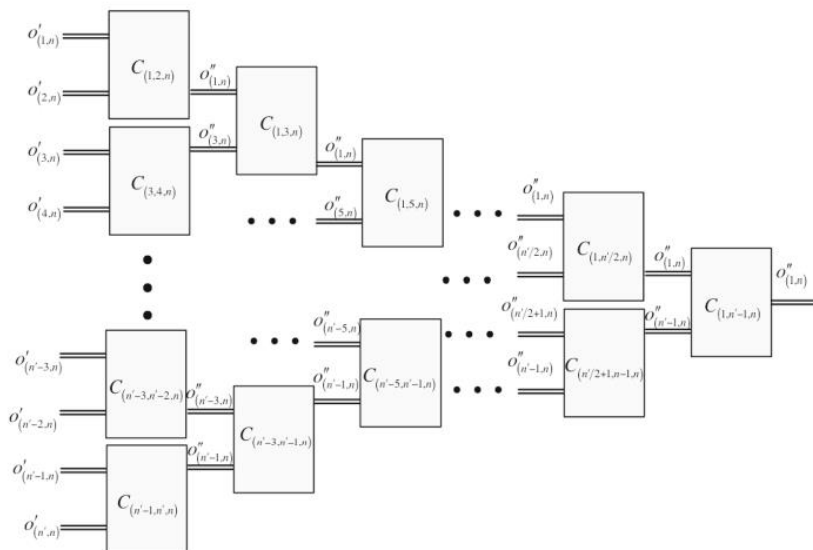


Fig. 8 The application of the modified combination blocks to combine entire composite paths $o'_{1,n}, o'_{2,n}, \dots, o'_{n'-1,n}, o'_{n,n}$, resulting in the post-selection of only one composite path as the output $o''_{(1,n)}$

approaches have been realized experimentally with good fidelity [18,24]. In particular, similar linear expansion technique for arbitrary two-qubit processing have been demonstrated [18] on silicon photonics chip that could generate two photonic qubits, on which it could perform arbitrary two-qubit unitary operations, including arbitrary entangling operations. As emphasized, our proposed scheme circumvents the decomposition issue by directly programming matrix elements of the operator which can be applied to non-unitary matrices or the operation that is governed by an unknown matrix, such as the phase estimation algorithm. For the latter, there is no operator to decompose; hence, our method can have advantage over the standard method. Concerning the unknown quantum operation, our work can be regarded as the theoretical extension of the work by Zhou et al. [24], where only 4×4 matrix was concerned. In some sense, our analysis is similar to that of Reck et al. [19], but emphasizes on the alternative approach to their decomposition method. To our best knowledge, both have not yet been realized experimentally for arbitrary $N \times N$ operators.

To further remark on the experimental realization of this architecture, our proposed scheme demands extremely strong nonlinearity and highly efficient single-photon detection. Unfortunately, the demonstration of these entirely programmable two-qubit gates is perhaps beyond the existing nonlinear optics technology. However, some previous studies using the smallest possible configuration have proven the advantages of this approach [23,24]. In fact, photon pairs from the process of spontaneous four-wave mixing in strip waveguides on a photonic chip could also produce polarization-entangled pairs [11]. In addition to polarization-encoded qubits, micro-ring resonators for a path-

entangled two-qubit state also have their potential. The Bell's state of multi-photon pairs were realized in photonics-integrated circuits [22]. Regardless of relying on the nonlinear optics as the single-photon source, the future potential of a semiconductor quantum dots platform is currently envisaged [1,4,6,14,21].

Finally, we comment that, like the standard method, this alternative scheme is valid to realize discrete operations. Once all $N \times N$ matrices can be implemented, it is possible to measure the continuous observable. This can be done by truncating Fourier transforms to an arbitrary large but discrete operator. In the context of quantum simulation, this can be very beneficial for time-evolution operator of some physical system, especially with random interaction with environment or fluctuation.

Acknowledgements We would like to thank Mr. Thomas D. Coyne and Mr. Kittisak Thowsombat for their kind assistance in editing the English, Assist. Prof. Dr. Kwan Arayathanitkul and Assist. Prof. Dr. Worasak Sukkabot for helpful discussions. Grant No. 035/2557 from the Development and Promotion of Science and Technology Talents project (DPST) scholarship, research fund for DPST graduate with first placement and Grant No. ThEP-60-PHY-PSU4 by the Thailand Center of Excellence in Physics are acknowledged. Moreover, this project is under U.Reka Program sponsored by The Siam Commercial Bank Public Company Limited and managed by Digital Ventures Co., Ltd.

References

1. Akopian, N., Lindner, N.H., Poem, E., Berlatzky, Y., Avron, J., Gershoni, D., Gerardot, B.D., Petroff, P.M.: Entangled photon pairs from semiconductor quantum dots. *Phys. Rev. Lett.* **96**, 130501 (2006)
2. Barendo, A., Bennett, C.H., Cleve, R., Vincenzo, D.P.D., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. *Phys. Rev. A* **52**, 3457 (1995)
3. Carolan, J., Harrold, C., Sparrow, C., Martín-López, E., Russell, N.J., Silverstone, J.W., Shadbolt, P.J., Matsuda, N., Oguma, M., Itoh, M., Marshall, G.D., Thompson, M.G., Matthews, J.C.F., Hashimoto, T., O'Brien, J.L., Laing, A.: Universal linear optics. *Science* **349**(6249), 711–716 (2015)
4. Claudon, J., Bleuse, J., Malik, N.S., Bazin, M., Jaffrennou, P., Gregersen, N., Sauvan, C., Lalanne, P., Gérard, J.-M.: A highly efficient single-photon source based on a quantum dot in a photonic nanowire. *Nat. Photonics* **4**, 174 (2010)
5. Deutsch, D.: Quantum theory, the church-turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A* **400**, 97 (1985)
6. Dousse, A., Suffczyński, J., Beveratos, A., Krebs, O., Lemaître, A., Sagnes, I., Bloch, J., Voisin, P., Senellart, P.: Ultrabright source of entangled photon pairs. *Nature* **466**, 217 (2010)
7. Geller, M.R., Martinis, J.M., Sornborger, A.T., Stancil, P.C., Pritchett, E.J., You, H., Galiatdinov, A.: Universal quantum simulation with prethreshold superconducting qubits: single-excitation subspace method. *Phys. Rev. A* **91**, 062309 (2015)
8. Hamel, D.R., Shalm, L.K., Hübel, H., Miller, A.J., Marsili, F., Verma, V.B., Mirin, R.P., Nam, S.W., Resch, K.J., Jennewein, T.: Direct generation of three-photon polarization entanglement. *Nat. Photonics* **8**(10), 801–807 (2014)
9. Kane, B.: A silicon-based nuclear spin quantum computer. *Nature* **393**, 133 (1998)
10. Leibfried, D., DeMarco, B., Meyer, V., Lucas, D., Barrett, M., Britton, J., Itano, W.M., Jelenkovi, B., Langer, C., Rosenband, T., Wineland, D.J.: Experimental demonstration of a robust and high-fidelity geometric two ion-qubit phase gate. *Nature* **422**, 412–415 (2003)
11. Matsuda, N., Le Jeannic, H., Fukuda, H., Tsuchizawa, T., Munro, W.J., Shimizu, K., Yamada, K., Tokura, Y., Takesue, H.: A monolithically integrated polarization entangled photon pair source on a silicon chip. *Sci. Rep.* **2**, 817 (2012)
12. Miller, D.A.B.: Self-configuring universal linear optical component [invited]. *Photonics Res.* **1**(1), 1–15 (2013)
13. Monroe, C., Meekhof, D.M., King, B.E., Itano, W.M., Wineland, D.J.: Demonstration of a fundamental quantum logic gate. *Phys. Rev. Lett.* **75**, 4714 (1995)

14. Müller, M., Vural, H., Schneider, C., Rastelli, A., Schmidt, O.G., Höfling, S., Michler, P.: Quantum-dot single-photon sources for entanglement enhanced interferometry. *Phys. Rev. Lett.* **118**, 257402 (2017)
15. O'Brien, J.L., Pryde, G.J., White, A.G., Ralph, T.C., Branning, D.: Demonstration of an all-optical quantum controlled-NOT gate. *Nature* **426**(6964), 264–267 (2003)
16. Peters, N., Altepeter, J., Jeffrey, E., Branning, D., Kwiat, P.: Precise creation, characterization, and manipulation of single optical qubits. *Quantum Inf. Comput.* **3**(7), 503–517 (2003)
17. Politi, A., Cryan, M.J., Rarity, J.G., Yu, S., O'Brien, J.L.: Silica-on-silicon waveguide quantum circuits. *Science* **320**, 646 (2008)
18. Qiang, X., Zhou, X., Wang, J., Wilkes, C.M., Loke, T., O'Gara, S., Kling, L., Marshall, G.D., Santagati, R., Ralph, T.C., Wang, J.B., O'Brien, J.L., Thompson, M.G., Matthews, J.C.F.: Large-scale silicon quantum photonics implementing arbitrary two-qubit processing. *Nat. Photonics* **12**(9), 534–539 (2018)
19. Reck, M., Zeilinger, A., Bernstein, H.J., Bertani, P.: Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* **73**, 58 (1994)
20. Schmidt-Kaler, F., Häffner, H., Riebe, M., Gulde, S., Lancaster, G.P.T., Deuschle, T., Becher, C., Roos, C.F., Eschner, J., Blatt, R.: Realization of the Cirac-Zoller controlled-NOT quantum gate. *Nature* **422**, 408–411 (2003)
21. Stevenson, R.M., Hudson, A.J., Young, R.J., Atkinson, P., Cooper, K., Ritchie, D.A., Shields, A.J.: Biphoton interference with a quantum dot entangled light source. *Opt. Express* **15**(10), 6507–6512 (2007)
22. Wang, J., Paesani, S., Ding, Y., Santagati, R., Skrzypczyk, P., Salavrakos, A., Tura, J., Augusiak, R., Mančinska, L., Bacco, D., Bonneau, D., Silverstone, J.W., Gong, Q., Acín, A., Rottwitz, K., Oxenløwe, L.K., O'Brien, J.L., Laing, A., Thompson, M.G.: Multidimensional quantum entanglement with large-scale integrated optics. *Science* **360**(6386), 285–291 (2018)
23. Zhou, X.-Q., Kalasuwan, P., Ralph, T.C., O'Brien, J.L.: Calculating unknown eigenvalues with a quantum algorithm. *Nat. Photonics* **7**, 223–228 (2013)
24. Zhou, X.-Q., Ralph, T.C., Kalasuwan, P., Zhang, A., Peruzzo, M., Lanyon, B.P., O'Brien, J.L.: Adding control to arbitrary unknown quantum operations. *Nat. Commun.* **2**(413), 2011 (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

VITAE

Name Mr. Paphon Phewkhom

Student ID 6110220133

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Science (Physics)	Prince of Songkla university	2017

List of Publication and Proceeding

1. Phewkhom, P., S. Suwanna, and Pruet Kalasuwan. "Alternative Scheme of Universal Optical Programmable Multi-Qubit Gates for Polarization Qubits." Quantum Information Processing 19 (June 8, 2020).

-