Digital Simulation of Single Qubit Markovian Open Quantum Systems: A Tutorial

Ian Joel David [1,*], **Ilya Sinayskiy** [1,2,†] **& Francesco Petruccione** [1,2,3,‡]

[1] School of Chemistry and Physics, University of KwaZulu-Natal, Durban, South Africa
[2] National Institute for Theoretical and Computational Sciences (NITheCS), South Africa
[3] School of Data Science and Computational Thinking and Department of Physics, Stellenbosch University, Stellenbosch, South Africa
E-mails: ian.david@nithecs.ac.za*, sinayskiy@ukzn.ac.za†, petruccione@sun.ac.za‡

Editors: *Vinayak Jagadish & Danko D. Georgiev*

Article history: Submitted on May 10, 2023; Accepted on September 11, 2023; Published on September 18, 2023.

One of the first proposals for the use of quantum computers was the simulation of quantum systems. Over the past three decades, great strides have been made in the development of algorithms for simulating closed quantum systems and the more complex open quantum systems. In this tutorial, we introduce the methods used in the simulation of single qubit Markovian open quantum systems. It combines various existing notations into a common framework that can be extended to more complex open system simulation problems. The only currently available algorithm for the digital simulation of single qubit open quantum systems is discussed in detail. A modification to the implementation of the simpler channels is made that removes the need for classical random sampling, thus making the modified algorithm a strictly quantum algorithm. The modified algorithm makes use of quantum forking to implement the simpler channels that approximate the total channel. This circumvents the need for quantum circuits with a large number of CNOT gates.

Quanta 2023; 12: 131–163.

1 Introduction

The simulation of complex quantum systems is central to many important problems in drug discovery, chemistry, material science and more. Although simulating these systems remains a computationally challenging task for classical computers, the natural resources of quantum computers can be leveraged to more efficiently simulate these systems.

Generally, simulating a quantum system involves approximating a mathematical object describing the total evolution of a system by some combination of simpler objects. When the simulation is performed on a quantum computer, these simpler objects define the operations that will be performed on the quantum computer. Over the past three decades, great strides have been made in the development of algorithms for simulating closed and open quantum systems. These algorithms often strive to achieve a low enough error in the approximation of the mathematical object and find an efficient set of operations to perform the evolution.

The simulation of closed quantum systems, often called Hamiltonian Simulation, has been extensively explored [1–12]. For Hamiltonian Simulation, the mathematical object that must be approximated is the total unitary evolution generated by some Hamiltonian. The total unitary evolution is approximated by some combination of simpler unitaries, generated by the components of the decomposed Hamiltonian. Once the simpler unitaries are obtained, efficient quantum circuits for these simpler unitaries which minimise gate count and number of qubits must be constructed. Algorithms for Hamiltonian simulation are typically characterised by the recombination methods used to combine the simpler unitaries. The most commonly used recombination method is the Suzuki–Lie–Trotter (SLT) Product Formulas [13, 14].

While closed quantum systems are relevant to many problems in quantum mechanics, open quantum systems describe a more complex, commonly encountered system: a quantum system that can interact with its surrounding environment, sharing information and energy [15, 16]. These systems are crucial to our understanding of nonequilibrium dynamics and thermalization in a wide range of systems, from damped-driven spin-boson models to complex many fermion-boson models [17, 18]. In this tutorial, we shall only consider the simulation of only Markovian open quantum systems for which we can neglect memory effects.

Several methods that allow us to simulate open quantum systems have been developed [19–27]. For the simulation of open quantum systems, the mathematical object that must be approximated is a quantum channel: a completely positive and trace preserving (CPTP) map, also called a dynamical map. Like the Hamiltonian which generates the total unitary evolution for closed quantum systems, a superoperator called the Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) generator [28,29] generates the quantum channel. The quantum channel can then be approximated by a combination of simpler channels generated by the components of the decomposed GKSL generator. These simpler channels are then executed on the quantum computer.

While the spectral decomposition was the natural choice for decomposing the Hamiltonians in Hamiltonian simulation, the method used to decompose the GKSL generator for open system simulation was developed by Bacon *et al.* [20]. This method involves decomposing the GKSL generator into simpler components via linear combination of semi-groups and unitary conjugation. It was also shown by Bacon *et al.* that all quantum channels on finite dimensional systems can be simulated by unitary evolutions and quantum channels from a universal semigroup. The decomposition method developed was used by Sweke *et al.* to develop an algorithm for simulating open quantum systems [19, 21]. This algorithm makes use of SLT product formulas for the recombination of the simpler quantum channels. Sweke *et al.* also derived the bounds on the error in the approximation as well as the bounds on the gate counts.

Despite the advancements made in the simulation of open systems, a standard or widely accepted notation is lacking. This has hindered progress in the field. This tutorial will introduce the methods used in the simulation of single qubit open quantum systems. It combines various existing notations into a common framework that can be extended to more complex open system simulation problems. The only currently available algorithm for the digital simulation of single qubit open quantum systems [19] will be discussed in detail. A modification that makes use of quantum forking [30] in the implementation of the simpler channels is made that removes the need for classical random sampling. This makes the modified algorithm a strictly quantum algorithm. This modification also circumvents the need for quantum circuits with a large number of CNOT gates.

This tutorial is structured as follows: Section 2 formally states the problem of quantum simulation and fixes the notation used throughout the text. Sections 3 and 4 present the decomposition and recombination methods used by the algorithm, respectively. Section 5 presents the quantum circuits that can implement any channel from the universal semigroup of quantum channels. Lastly in Section 6, concluding remarks are made and open problems in open quantum system simulation are discussed.

2 Problem and Setting

The state space of a single qubit is a two dimensional complex Hilbert space $\mathcal{H}_s \cong \mathbb{C}^2$. A quantum state of a single qubit is described by a density matrix $\rho \in \mathcal{M}_2(\mathbb{C}) \cong \mathcal{B}(\mathcal{H}_s)$, where $\mathcal{B}(\mathcal{H}_s)$ is the set of all bounded linear operators acting on the Hilbert space $\mathcal{H}_s$ and $\mathcal{M}_2(\mathbb{C})$ is the set of 2×2 complex matrices, such that:

$$\rho \geq 0, \qquad \mathrm{tr}(\rho) = 1, \qquad \rho = \rho^\dagger. \tag{1}$$

Quantum channels provide a general framework for describing the evolution of quantum states. These are completely positive and trace preserving (CPTP) maps [15],

$$T : \mathcal{B}(\mathcal{H}_s) \rightarrow \mathcal{B}(\mathcal{H}_s). \tag{2}$$

It is a well known fact that T has a Kraus representation [31],

$$T(\rho) = \sum_{j=1}^{r} K_j \rho K_j^\dagger, \tag{3}$$

where K_j are called the Kraus operators that satisfy the completeness relation $\sum_{j=1}^{r} K_j^\dagger K_j = \mathbb{1}_s$, with $\mathbb{1}_s$ being the identity operator acting on the space $\mathcal{H}_s$, and $r = \mathrm{rank}(\tau) \leq 4$. Here τ is the Choi matrix defined through the Choi–Jamiołkowski isomorphism [32, 33] as,

$$\tau = (T \otimes \mathbb{1}_s)|\Omega\rangle\langle\Omega|, \tag{4}$$

where $|\Omega\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is a maximally entangled state. The Choi matrix is a representation of the quantum channel T and it is defined through the Choi–Jamiołkowski isomorphism. It is a matrix description of the quantum channel T and it contains all the properties of T. For a quantum channel T, we can define the dual or adjoint as $\tilde{T}$ and it can be calculated through the relation $\mathrm{tr}(A\tilde{T}(B)) = \mathrm{tr}(T(A)B)$ for any $A, B \in \mathcal{B}(\mathcal{H}_s)$. Using this relation we can also find the Kraus representation of the dual of the channel,

$$\tilde{T}(\rho) = \sum_{j=1}^{r} K_j^\dagger \rho K_j. \tag{5}$$

So far we have only considered the state of a single qubit ρ that is time independent, but for most practical cases we must consider a time dependent state $\rho(t)$ for some $t \geq 0$.

The channels above describe discrete time evolution as the channel T does not depend on some continuous parameter and are specified by time independent Kraus operators. However we are interested in Markovian continuous time evolution. Which is described by a continuous single parameter semigroup of quantum channels $\{T_t\}$, which satisfy:

$$T_t T_s = T_{t+s}, \qquad T_0 = \mathbb{1} \qquad t, s \in \mathbb{R}_+. \tag{6}$$

Also if we introduce time dependence to the state of our single qubit, then we write the density matrix, $\rho(t)$ which describes the quantum state at some time $t \geq 0$ then we have that,

$$\rho(t) = T_t(\rho(0)). \tag{7}$$

Every semigroup $\{T_t\}$ has a unique generator $\mathcal{L} : \mathcal{B}(\mathcal{H}_s) \to \mathcal{B}(\mathcal{H}_s)$ such that,

$$T_t = e^{t\mathcal{L}} = \sum_{j=0}^{\infty} \frac{t^j}{j!}\mathcal{L}^j, \tag{8}$$

where $\mathcal{L}$ satisfies the master equation,

$$\frac{d}{dt}\rho(t) = \mathcal{L}(\rho(t)). \tag{9}$$

The generator $\mathcal{L}$ is the generator of a continuous one parameter Markovian semigroup $\{T_t\}$ if and only if it can be written in the celebrated Gorrini–Kossakowski–Sudarshan–Lindblad (GKSL) form [28, 29],

$$\mathcal{L}(\rho) = -i[H, \rho] + \sum_{i,j=1}^{3} A_{ij}([F_i, \rho F_j^\dagger] + [F_i\rho, F_j^\dagger]), \tag{10}$$

where $H = H^\dagger \in \mathcal{M}_2(\mathbb{C})$ is the Hamiltonian and $A \in \mathcal{M}_3(\mathbb{C})$ is a positive semi-definite matrix called the GKS matrix. The matrices $\{F_i\}$ are a basis for the space of traceless matrices in $\mathcal{M}_2(\mathbb{C})$ and without loss of generality we chose the basis $\{F_i\}$ to be the normalized Pauli matrices, i.e. $\{F_i\} = \frac{1}{\sqrt{2}}\{\sigma_i\}_{i=1}^{3} = \frac{1}{\sqrt{2}}\{\sigma_1, \sigma_2, \sigma_3\}$.

One of the main aspects of digital quantum simulation of an open quantum system is finding an approximation to the quantum channels that describe the evolution of the system. To quantify the error in approximations of quantum channels we make use of the 1 → 1 superoperator norm defined in terms of the general $p \to q$ Schatten norm. One should note quantum channels are also called superoperators since they are operators which act on other operators. The general $p \to q$ Schatten norm for some superoperator T is defined as:

$$\|T\|_{p\to q} = \sup_{\|A\|_p = 1} \|T(A)\|_q \tag{11}$$

where $\|A\|_p := (\mathrm{tr}(|A|^p))^{\frac{1}{p}}$ and $|A| = \sqrt{A^\dagger A}$. Using the above equation we see that the 1 → 1 superoperator norm is:

$$\|T\|_{1\to 1} = \sup_{\|A\|_1 = 1} \|T(A)\|_1, \tag{12}$$

where $\|A\|_1 = \mathrm{tr}(\sqrt{A^\dagger A})$. The 1 → 1 superoperator norm also satisfies the following standard properties. For any superoperators T and V we have that,

$$\begin{aligned} \|T + V\|_{1\to 1} &\leq \|T\|_{1\to 1} + \|V\|_{1\to 1}, \\ \|TV\|_{1\to 1} &\leq \|T\|_{1\to 1}\|V\|_{1\to 1}. \end{aligned} \tag{13}$$

Now that we have outlined some notation and defined some basic objects we can state the problem:

Problem Statement

Given a continuous one parameter semigroup $\{T_t\}$, generated by $\mathcal{L}$, specified by a GKS matrix $A \geq 0 \in \mathcal{M}_3(\mathbb{C})$ and a Hamiltonian $H = H^\dagger \in \mathcal{M}_2(\mathbb{C})$. Find a quantum circuit, using a polynomial number of gates, that approximates $T_t = \exp(t\mathcal{L})$ such that the maximum error in the final state, as quantified by the 1 → 1 superoperator norm is at most ϵ.

The strategy to solve this problem consists of three steps. First, we make use of linear combination of semigroups to decompose the channel T_t into constituent channels. Then we use unitary conjugation to further decompose the constituent channels into unitary transformations and a channel from some universal semigroup [20]. The second step makes use of SLT product formulas [13, 19] to approximate the channel T_t using combinations of the constituent channels. Lastly, we convexly decompose the channels from the universal semigroup into extreme channels [34] and use quantum forking [30] to implement the quantum circuits for the constituent channels.

3 Decomposition of the arbitrary generator

The first step is to decompose $\mathcal{L}$ into a combination of generators of simpler semigroups. We do this using linear combination of semigroups and unitary conjugation [20]. Given the GKSL generator for an arbitrary single qubit channel,

$$\mathcal{L}(\rho) = -i[H, \rho] + \frac{1}{2}\sum_{i,j=1}^{3} A_{ij}([\sigma_i, \rho\sigma_j] + [\sigma_i\rho, \sigma_j]), \tag{14}$$

where $A \geq 0 \in \mathcal{M}_3(\mathbb{C})$ and $H = H^\dagger \in \mathcal{M}_2(\mathbb{C})$. We first need to decompose this generator using linear combination of semigroups. Since $A \geq 0$ we can use the spectral decomposition to write

$$A = \sum_{k=1}^{3} \lambda_k A_k. \tag{15}$$

By substituting equation (15) into equation (14) and by letting $\mathcal{L}_0(\rho) = -i[H, \rho]$ we have,

$$\begin{aligned}\mathcal{L}(\rho) &= \mathcal{L}_0(\rho) + \frac{1}{2}\sum_{i,j=1}^{3}\left(\sum_{k=1}^{3}\lambda_k A_k\right)_{ij}([\sigma_i, \rho\sigma_j] + [\sigma_i\rho, \sigma_j]),\\ &= \mathcal{L}_0(\rho) + \sum_{k=1}^{3}\lambda_k\frac{1}{2}\left(\sum_{i,j=1}^{3}(A_k)_{ij}([\sigma_i, \rho\sigma_j] + [\sigma_i\rho, \sigma_j])\right).\end{aligned} \tag{16}$$

Defining,

$$\mathcal{L}_k(\rho) = \frac{1}{2}\sum_{i,j=1}^{3}(A_k)_{ij}([\sigma_i, \rho\sigma_j] + [\sigma_i\rho, \sigma_j]), \tag{17}$$

we get,

$$\mathcal{L}(\rho) = \mathcal{L}_0(\rho) + \sum_{k=1}^{3}\lambda_k\mathcal{L}_k(\rho). \tag{18}$$

Letting $\lambda_0 = 1$ we can then write equation (18) more compactly as

$$\mathcal{L}(\rho) = \sum_{k=0}^{3}\lambda_k\mathcal{L}_k(\rho). \tag{19}$$

This allows us to write T_t as,

$$T_t = e^{t\mathcal{L}} = \exp\left(t\sum_{k=0}^{3}\lambda_k\mathcal{L}_k\right). \tag{20}$$

At this point it is useful for one to split the exponential in equation (20) into simpler parts which we use in the recombination section to approximate the total channel. This can be achieved via the Lie–Trotter product formula which states that, for any sum of operators $\sum_k V_k$ acting on some vector space,

$$\exp\left(\sum_k V_k\right) = \lim_{n\to\infty}\left(\prod_k \exp(V_k/n)\right)^n. \tag{21}$$

If we define $T_t^{(k)} := \exp(t\mathcal{L}_k)$ and via a straightforward application of the Lie–Trotter product formula [13] we have that,

$$T_t = \lim_{n\to\infty}\left[\prod_{k=0}^{3} e^{\left(\frac{t}{n}\lambda_k\mathcal{L}_k\right)}\right]^n = \lim_{n\to\infty}\left[\prod_{k=0}^{3} T^{(k)}_{\left(\frac{t\lambda_k}{n}\right)}\right]^n. \tag{22}$$

In the language of [20] we say that $\{T_t\}$ can be constructed via a linear combination of semigroups $\{T_t^{(k)}\}$. The phrase linear combination of semigroups comes from the fact that we write $\mathcal{L}$ as a linear combination of $\mathcal{L}_k$ which generate the semigroups $\{T_t^{(k)}\}$. Equation (22), tells us that one can simulate T_t if one can efficiently simulate $T_t^{(k)}$ (i.e. the constituent channels) and use $T_t^{(k)}$ in some recombination strategy to approximate T_t.

Armed with this realisation that one has to be able to efficiently simulate $T_t^{(k)}$ to simulate the channel T_t. We ask ourselves, can we decompose the channel $T_t^{(k)}$ into simpler operations that are easier to efficiently implement on a quantum computer? The answer is yes, Bacon *et al.* proposed the use of an operation called unitary conjugation to decompose the channel $T_t^{(k)}$ into unitary transformations and a channel from some universal semigroup [20]. We now define the operation of unitary conjugation and then prove some results that describe the action of unitary conjugation on a quantum channel.

We define unitary conjugation (UC) of a channel T_t as the procedure of transforming T_t according to $\mathcal{U}^\dagger T_t\mathcal{U}$, i.e.

$$\mathrm{UC} : T_t \mapsto \mathcal{U}^\dagger T_t \mathcal{U} \tag{23}$$

where $\mathcal{U}(X) = UXU^\dagger$ for some unitary operator U and any operator X. UC preserves all Markovian semigroup properties and it is can be shown that UC effectively applies T_t in an alternative basis, however this is not necessary for this tutorial. We recall that $\mathcal{L}_0$ $(k = 0)$, generates Hamiltonian evolution which can be simulated using a single qubit unitary operation. We are interested in the generators of dissipative evolution $\mathcal{L}_k$ for $k \in \{1, 2, 3\}$ and want to use unitary conjugation to further decompose the channels $T_t^{(k)}$ for $k \in \{1, 2, 3\}$. To use UC to decompose $T_t^{(k)}$, we need to understand how UC of T_t affects the corresponding GKS matrix defining the generator of T_t. This is made clear by the following theorem from [20].

Theorem 1. For a single qubit channel, unitary conjugation of T_t by $U \in \mathrm{SU}(2)$ results in conjugation of the GKS matrix A by a corresponding element in SO(3) which is the adjoint representation of SU(2).

To prove Theorem 1 we need the following two lemmas below, which outline some properties of the map $\mathcal{U}(\rho)$.

Lemma 1. Given $\mathcal{U}(\rho) = U\rho U^\dagger$ (with $\mathcal{U}^\dagger(\rho) = U^\dagger\rho U$) for some unitary operator U. $\mathcal{U}(\rho)$ satisfies,

$$\mathcal{U}(\mathcal{U}^\dagger(\rho)) = \mathbb{1}(\rho) = \mathcal{U}^\dagger(\mathcal{U}(\rho)). \tag{24}$$

Proof. We can prove this directly from the definition of $\mathcal{U}(\rho)$. Recalling that $UU^\dagger = U^\dagger U = \mathbb{1}$ we get,

$$\mathcal{U}(\mathcal{U}^\dagger(\rho)) = \mathcal{U}(U^\dagger\rho U) = UU^\dagger\rho UU^\dagger = \rho = \mathbb{1}(\rho). \tag{25}$$

This implies that, $\mathcal{U}\mathcal{U}^\dagger = \mathbb{1}$. Similarly for $\mathcal{U}^\dagger\mathcal{U}$ we can show that,

$$\mathcal{U}^\dagger(\mathcal{U}(\rho)) = \mathcal{U}^\dagger(U\rho U^\dagger) = U^\dagger U\rho U^\dagger U = \rho = \mathbb{1}(\rho), \tag{26}$$

which implies that $\mathcal{U}^\dagger\mathcal{U} = \mathbb{1}$. □

Lemma 2. Given $\mathcal{U}(\rho) = U\rho U^\dagger$ for some unitary operator U, and $p \geq 0$ for $k \in \mathbb{Z}$. We have that,

$$(\mathcal{U}^\dagger\mathcal{L}\mathcal{U})^p = \mathcal{U}^\dagger\mathcal{L}^p\mathcal{U}. \tag{27}$$

Proof. We prove the formula above using induction. For the base case, i.e. $p = 0$, we have,

$$(\mathcal{U}^\dagger\mathcal{L}\mathcal{U})^0 = \mathbb{1}, \tag{28}$$

and

$$\mathcal{U}^\dagger\mathcal{L}^0\mathcal{U} = \mathcal{U}^\dagger\mathbb{1}\mathcal{U} = \mathcal{U}^\dagger\mathcal{U} = \mathbb{1}, \tag{29}$$

where we make use of lemma 1 in the last equality in equation (29). From equations (28) and (29) we see that the formula holds for the base case. Now we assume that the formula is true for $p = m$, that is,

$$(\mathcal{U}^\dagger\mathcal{L}\mathcal{U})^m = \mathcal{U}^\dagger\mathcal{L}^m\mathcal{U}, \tag{30}$$

and we proceed to show that the formula holds for $p = m + 1$. We start by writing,

$$\begin{aligned}(\mathcal{U}^\dagger\mathcal{L}\mathcal{U})^{m+1} &= (\mathcal{U}^\dagger\mathcal{L}\mathcal{U})^m(\mathcal{U}^\dagger\mathcal{L}\mathcal{U}),\\ &= \mathcal{U}^\dagger\mathcal{L}^m\mathcal{U}(\mathcal{U}^\dagger\mathcal{L}\mathcal{U}), \quad \text{(By eq. (30))}\\ &= \mathcal{U}^\dagger\mathcal{L}^m\mathcal{U}\mathcal{U}^\dagger\mathcal{L}\mathcal{U},\\ &= \mathcal{U}^\dagger\mathcal{L}^m\mathcal{L}\mathcal{U}, \quad \text{(By Lemma 1)}\\ &= \mathcal{U}^\dagger\mathcal{L}^{m+1}\mathcal{U}.\end{aligned} \tag{31}$$

Hence, by induction the formula holds. □

We now have the necessary results to prove Theorem 1.

Proof of Theorem 1. Suppose $\{T_t\}$ has a generator $\mathcal{L}$ and a GKS matrix A then $T_t = \exp(t\mathcal{L})$. Now for some $U \in \mathrm{SU}(2)$ we want to see what unitary conjugation does to the channel T_t,

$$\begin{aligned}\mathcal{U}^\dagger T_t\mathcal{U} &= \mathcal{U}^\dagger\exp(t\mathcal{L})\mathcal{U},\\ &= \mathcal{U}^\dagger\sum_{p=0}^{\infty}\frac{t^p}{p!}\mathcal{L}^p\mathcal{U},\\ &= \sum_{k=0}^{\infty}\frac{t^p}{p!}\mathcal{U}^\dagger\mathcal{L}^p\mathcal{U},\\ &= \sum_{k=0}^{\infty}\frac{t^p}{p!}(\mathcal{U}^\dagger\mathcal{L}\mathcal{U})^p, \quad \text{(By Lemma 2)}\\ &= \exp(t\mathcal{U}^\dagger\mathcal{L}\mathcal{U}).\end{aligned} \tag{32}$$

So unitary conjugation of T_t effectively applies unitary conjugation to the generator $\mathcal{L}$, i.e. $\mathcal{L} \mapsto \mathcal{U}^\dagger\mathcal{L}\mathcal{U}$. For the remainder of this proof we make use of the index summation convention where repeated indicies are summed over and the summation range shall always be from 1 to 3 unless otherwise specified. Given the generator in the GKSL form we can always write $\mathcal{L}(\rho) = \mathcal{L}_H(\rho) + \mathcal{L}_D(\rho)$, where $\mathcal{L}_H(\rho) = -i[H,\rho]$ is the Hamiltonian part and $\mathcal{L}_D(\rho) = \frac{1}{2}A_{ij}([\sigma_i,\rho\sigma_j] + [\sigma_i\rho,\sigma_j])$ the dissipative part. We are only interested in how unitary conjugation effects the dissipative part $\mathcal{L}_D$ since it contains the GKS matrix A. Applying the unitary conjugation map to $\mathcal{L}_D$ yields,

$$\begin{aligned}(\mathcal{U}^\dagger\mathcal{L}_D\mathcal{U})(\rho) &= U^\dagger\Big(\frac{1}{2}A_{ij}([\sigma_i, U\rho U^\dagger\sigma_j] + [\sigma_i U\rho U^\dagger,\sigma_j])\Big)U,\\ &= \frac{1}{2}A_{ij}\Big(U^\dagger[\sigma_i, U\rho U^\dagger\sigma_j]U + U^\dagger[\sigma_i U\rho U^\dagger,\sigma_j]U\Big).\end{aligned} \tag{33}$$

We can simplify the first term in equation (33) as follows,

$$\begin{aligned}U^\dagger[\sigma_i, U\rho U^\dagger\sigma_j]U &= U^\dagger(\sigma_i U\rho U^\dagger\sigma_j - U\rho U^\dagger\sigma_j\sigma_i)U,\\ &= U^\dagger\sigma_i U\rho U^\dagger\sigma_j U - \rho U^\dagger\sigma_j\sigma_i U,\\ &= U^\dagger\sigma_i U\rho U^\dagger\sigma_j U - \rho U^\dagger\sigma_j U U^\dagger\sigma_i U,\\ &= [U^\dagger\sigma_i U, \rho U^\dagger\sigma_j U].\end{aligned} \tag{34}$$

In a similar method to the one used in (34) we show that,

$$U^\dagger[\sigma_i U\rho U^\dagger,\sigma_j]U = [U^\dagger\sigma_i U\rho, U^\dagger\sigma_j U]. \tag{35}$$

By substituting (34) and (35) into (33) we get,

$$(\mathcal{U}^\dagger \mathcal{L}_D \mathcal{U})(\rho) = \frac{1}{2} A_{ij}([U^\dagger \sigma_i U, \rho U^\dagger \sigma_j U] + [U^\dagger \sigma_i U \rho, U^\dagger \sigma_j U]). \tag{36}$$

So unitary conjugation induces a change of basis $\frac{1}{\sqrt{2}}\sigma_i \mapsto U^\dagger \frac{1}{\sqrt{2}}\sigma_i U$, this basis is still Hermitian, orthonormal and traceless. We can expand this basis in terms of the old one as follows,

$$U^\dagger \frac{1}{\sqrt{2}} \sigma_\alpha U = c_{\alpha\gamma} \frac{1}{\sqrt{2}} \sigma_\gamma. \tag{37}$$

By multiplication of two operators from this basis we can gain more insight into the nature of the matrix $c_{\alpha\gamma}$. First taking the product of two elements in the new basis yields,

$$\frac{1}{2} U^\dagger \sigma_\alpha U U^\dagger \sigma_\beta U = \frac{1}{2} c_{\alpha\gamma} c^*_{\beta\nu} \sigma_\gamma \sigma_\nu. \tag{38}$$

Taking the trace of equation (38) and using the fact that $\mathrm{tr}(\sigma_\alpha \sigma_\beta) = 2\delta_{\alpha\beta}$ we have,

$$\mathrm{tr}(\frac{1}{2} U^\dagger \sigma_\alpha U U^\dagger \sigma_\beta U) = \frac{1}{2} c_{\alpha\gamma} c^*_{\beta\nu} \mathrm{tr}(\sigma_\gamma \sigma_\nu) = \frac{1}{2} c_{\alpha\gamma} c^*_{\beta\nu} 2\delta_{\gamma\nu} = c_{\alpha\gamma} c^*_{\beta\gamma}. \tag{39}$$

But we also know that the transformed basis $U^\dagger \frac{1}{\sqrt{2}} \sigma_\alpha U$ is still orthonormal, i.e. $\mathrm{tr}(\frac{1}{2} U^\dagger \sigma_\alpha U U^\dagger \sigma_\beta U) = \delta_{\alpha\beta}$, leading to the observation that,

$$c_{\alpha\gamma} c^*_{\beta\gamma} = \delta_{\alpha\beta}. \tag{40}$$

In other words $c_{\alpha\gamma}$ is a unitary matrix. Equipped with this information about the matrix $c_{\alpha\beta}$ we can substitute equation (38) into (37),

$$\begin{aligned}(\mathcal{U}^\dagger \mathcal{L}_D \mathcal{U})(\rho) &= \frac{1}{2} A_{\alpha\beta}([c_{\alpha\gamma}\sigma_\gamma, \rho c^*_{\beta\nu}\sigma_\nu] + [c_{\alpha\gamma}\sigma_\gamma \rho, c^*_{\beta\nu}\sigma_\nu]), \\ &= \frac{1}{2} c_{\alpha\gamma} A_{\alpha\beta} c^*_{\beta\nu} ([\sigma_\gamma, \rho\sigma_\nu] + [\sigma_\gamma \rho, \sigma_\nu]), \\ &= A'_{\gamma\nu}([\sigma_\gamma, \rho\sigma_\nu] + [\sigma_\gamma\rho, \sigma_\nu]).\end{aligned} \tag{41}$$

Here A' is the transformed GKS matrix and $A'_{\gamma,\nu} = c_{\alpha\gamma} A_{\alpha\beta} c^*_{\beta\nu}$. Defining the matrix $C = c_{\gamma\alpha}$ then we have,

$$A' = C^T A C^*. \tag{42}$$

So the effect of unitary conjugation is to conjugate the GKS matrix by a matrix C^T. We note that C^T is arbitrary but determined by $U \in \mathrm{SU}(2)$ in the following way. Suppose we choose $\{\frac{1}{2}\sigma_\alpha\}_{\alpha=1}^3$ to be the generators of SU(2) then the set $\{\frac{1}{2}\sigma_\alpha\}_{\alpha=1}^3$ forms a basis for the Lie algebra $\mathfrak{su}(2)$ of the Lie group SU(2). The elements from the basis of $\mathfrak{su}(2)$ also satisfy,

$$\frac{1}{4}[\sigma_\alpha, \sigma_\beta] = i\epsilon_{\alpha\beta\gamma} \frac{1}{2} \sigma_\gamma, \tag{43}$$

where $\epsilon_{\alpha\beta\gamma}$ is the Levi-Cevita symbol and are the structure constants for $\mathfrak{su}(2)$. By setting $U = \exp(ir_\gamma \sigma_\gamma/2)$ we can expand U in a Taylor series, about zero, to 1^{st} order in an infinitesimal r_γ and substitute this into the new basis $U^\dagger \frac{1}{\sqrt{2}} \sigma_\alpha U$,

$$\begin{aligned}U^\dagger \frac{1}{\sqrt{2}} \sigma_\alpha U &= \left(\mathbb{1} - i\frac{r_\gamma}{2}\sigma_\gamma\right) \frac{1}{\sqrt{2}} \sigma_\alpha \left(\mathbb{1} + i\frac{r_\gamma}{2}\sigma_\gamma\right), \\ &= \frac{1}{\sqrt{2}}\sigma_\alpha + \frac{1}{2\sqrt{2}} i r_\gamma \sigma_\alpha \sigma_\gamma - \frac{1}{2\sqrt{2}} i r_\gamma \sigma_\gamma \sigma_\alpha + O\left(r_\gamma^2\right), \\ &= \frac{1}{\sqrt{2}}\sigma_\alpha - \frac{1}{2\sqrt{2}} i r_\gamma [\sigma_\gamma, \sigma_\alpha], \\ &= \frac{1}{\sqrt{2}}\sigma_\alpha - i r_\gamma \left(i\epsilon_{\gamma\alpha\beta} \frac{1}{\sqrt{2}} \sigma_\beta\right).\end{aligned} \tag{44}$$

But from equation (37) $U^\dagger \frac{1}{\sqrt{2}}\sigma_\alpha U = c_{\alpha\beta}\frac{1}{\sqrt{2}}\sigma_\beta$ allowing us to write,

$$c_{\alpha\beta}\frac{1}{\sqrt{2}}\sigma_\beta = \frac{1}{\sqrt{2}}\sigma_\alpha - ir_\gamma(i\epsilon_{\gamma\alpha\beta}\frac{1}{\sqrt{2}}\sigma_\beta). \tag{45}$$

Multiplying both sides of equation (45) from the left by $\frac{1}{\sqrt{2}}\sigma_{\beta'}$ we have,

$$\frac{1}{2}c_{\alpha\beta}\sigma_{\beta'}\sigma_\beta = \frac{1}{2}\sigma_{\beta'}\sigma_\alpha - \frac{1}{2}ir_\gamma(i\epsilon_{\gamma\alpha\beta}\sigma_{\beta'}\sigma_\beta). \tag{46}$$

Finally taking the trace of equation (46) and using the fact that $\mathrm{tr}(\sigma_\beta\sigma_\alpha) = 2\delta_{\alpha\beta}$ we obtain,

$$c_{\alpha\beta}\delta_{\beta'\beta} = \delta_{\beta'\alpha} - ir_\gamma(i\epsilon_{\gamma\alpha\beta}\delta_{\beta'\beta}) \implies c_{\alpha\beta'} = \delta_{\alpha\beta'} - ir_\gamma(i\epsilon_{\gamma\alpha\beta'}) = (C)_{\alpha\beta'}. \tag{47}$$

It is clear from (47) that C^T is,

$$\left(C^T\right)_{\alpha\beta} = (C)_{\beta\alpha} = \delta_{\beta\alpha} - ir_\gamma(i\epsilon_{\gamma\beta\alpha}), \tag{48}$$

but the Levi-Cevita symbol is totally anti-symmetric allowing one to write,

$$\left(C^T\right)_{\alpha\beta} = \delta_{\alpha\beta} + ir_\gamma(i\epsilon_{\gamma\alpha\beta}). \tag{49}$$

We observe that C^T is in a Lie group generated by $(G_\gamma)_{\alpha\beta} = i\epsilon_{\gamma\alpha\beta}$ which implies that $C^T = \exp(ir_\gamma G_\gamma)$. Since we know how each G_γ is defined by picking $\gamma, \alpha, \beta \in \{1, 2, 3\}$ we can explicitly calculate the matrix elements for each G_γ as,

$$G_1 = i\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \quad G_2 = i\begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad G_3 = i\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{50}$$

We recognize the above matrices as the generators of SO(3), hence $C^T \in$ SO(3) which is the adjoint representation of SU(2). As a brief mathematical aside: it is elementary fact of group theory that the adjoint representation of the Lie algebra of SU(N) is generated by the structure constants of the Lie algebra of SU(N) (For more information on the adjoint representation one could refer to the following books [35–37]). Therefore, unitary conjugation of the channel T_t by $U \in$ SU(2) leads to conjugation of the GKS matrix A by $C^T \in$ SO(3), i.e. $\mathcal{U}^\dagger T_t \mathcal{U}$ leads to a new GKS matrix $A' = C^T AC$. □

So far we have seen how linear combination of semigroups allows us to write channels from the semigroup $\{T_t\}$ as a limit of a product of constituent channels from the semigroup $\{T_t^{(k)}\}$. It was also shown how unitary conjugation affects the channel T_t. Theorem 2 below will show how using linear combination of semigroups and unitary conjugation one can simulate a channel T_t, it is a modification of Theorem 3 in [20]. This modification was made so that we can account for the differing notation in the quantum information community as well as the open quantum systems community. As we know researchers working in the field of open quantum systems define the ground state of a single qubit as the column vector $(0, 1)^T$, while the quantum information community defines the excited state with this column vector. The range of the parameter θ in Theorem 2 below accounts for this differing definitions for ground and excited states.

Theorem 2. To simulate, using linear combination and unitary conjugation, an arbitrary Markovian semigroup generated by $\mathcal{L} \in \mathcal{B}(\mathcal{B}(\mathcal{H}_s))$ with $\mathcal{H}_s \cong \mathbb{C}^2$, it is necessary and sufficient to be able to simulate all Markovian semigroups whose generators are specified by the GKS matrix $A(\theta) = \vec{a}(\theta)\vec{a}(\theta)^\dagger$ where $\vec{a}(\theta) = (\cos(\theta), -i\sin(\theta), 0)^T$ and $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$.

Proof. (Sufficiency) First, without any loss of generality we assume $H = 0$. Let $A \geq 0 \in \mathcal{M}_3(\mathbb{C})$, be the GKS matrix specifying the generator of the Markovian semigroup we wish to simulate. We fix the basis $\{F_i\} = \frac{1}{\sqrt{2}}\{\sigma_i\}_{i=1}^3$ without any loss of generality. Since $A \geq 0$ we use the spectral decomposition to express A as in equation (15). Since we can always write each projector A_k as an outer product, i.e. $A_k = \vec{a}_k\vec{a}_k^\dagger$, we have,

$$A = \sum_{k=1}^{3} \lambda_k \vec{a}_k\vec{a}_k^\dagger, \tag{51}$$

where $\lambda_k \geq 0$ and $\vec{a}_k^\dagger \cdot \vec{a}_k = 1$ for $k = 1, 2, 3$. By linear combination of semigroups it is sufficient to simulate all GKS matrices $\vec{a}_k \vec{a}_k^\dagger$ with $|\vec{a}_k| = 1$, for simplicity from this point onwards we drop the subscript k and just write $\vec{a}\vec{a}^\dagger$ and $|\vec{a}| = 1$. Any vector $\vec{a}$ can be split into real and imaginary parts, $\vec{a}^R$ and $\vec{a}^I$ respectively,

$$\vec{a} = \vec{a}^R + i\vec{a}^I. \tag{52}$$

Since $\vec{a}$ only appears in outer products multiplying $\vec{a}$ by an overall phase $e^{i\psi}$ leaves $\vec{a}\vec{a}^\dagger$ invariant, i.e. if $\vec{a}' = e^{i\psi}\vec{a}$ then,

$$\vec{a}'\vec{a}'^\dagger = e^{i\psi}\vec{a}e^{-i\psi}\vec{a}^\dagger = \vec{a}\vec{a}^\dagger. \tag{53}$$

Multiplying by an overall phase changes $\vec{a}$ as follows,

$$\vec{a}' = e^{i\psi}\vec{a} = e^{i\psi}\vec{a}^R + ie^{i\psi}\vec{a}^I = \vec{a}'^R + i\vec{a}'^I, \tag{54}$$

where $\vec{a}'^R = \vec{a}^R\cos(\psi) - \vec{a}^I\sin(\psi)$ and $\vec{a}'^I = \vec{a}^R\sin(\psi) + \vec{a}^I\cos(\psi)$. So if we wish to simulate $\vec{a}\vec{a}^\dagger$ we could simulate $\vec{a}'\vec{a}'^\dagger$ for any value of ψ. Defining the following two parameters,

$$k_1 = |\vec{a}^R|^2 - |\vec{a}^I|^2 \quad \text{and} \quad k_2 = 2\vec{a}^R \cdot \vec{a}^I, \tag{55}$$

and multiplying $\vec{a}$ by an overall phase changes k_1 and k_2 as follows,

$$\begin{aligned} k_1' &= |\vec{a}'^R|^2 - |\vec{a}'^I|^2 = |\vec{a}^R\cos(\psi) - \vec{a}^I\sin(\psi)|^2 - |\vec{a}^R\sin(\psi) + \vec{a}^I\cos(\psi)|^2 \\ &= |\vec{a}^R|^2\cos(2\psi) - |\vec{a}^I|\cos(2\psi) - \sin(2\psi)(2\vec{a}^R \cdot \vec{a}^I) \\ &= \cos(2\psi)k_1 - \sin(2\psi)k_2, \end{aligned} \tag{56}$$

and

$$\begin{aligned} k_2' &= 2\vec{a}'^R \cdot \vec{a}'^I = 2(\vec{a}^R\cos(\psi) - \vec{a}^I\sin(\psi)) \cdot (\vec{a}^R\sin(\psi) + \vec{a}^I\cos(\psi)) \\ &= \sin(2\psi)(|\vec{a}^R|^2 - |\vec{a}^I|^2) + \cos(2\psi)(2\vec{a}^R \cdot \vec{a}^I) \\ &= \sin(2\psi)k_1 + \cos(2\psi)k_2. \end{aligned} \tag{57}$$

We can express the transformation of k_1 and k_2 as a matrix equation,

$$\begin{pmatrix} k_1' \\ k_2' \end{pmatrix} = \begin{pmatrix} \cos(2\psi) & -\sin(2\psi) \\ \sin(2\psi) & \cos(2\psi) \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}. \tag{58}$$

Since we can choose ψ arbitrarily we can make the choice,

$$\tan(2\psi) = -\frac{k_2}{k_1}, \tag{59}$$

such that $k_2' = 0$, in which case $\vec{a}'^R \cdot \vec{a}'^I = 0$. In addition we can choose, $k_1' = k_1/\cos(2\psi) \geq 0$, such that $|\vec{a}'^R| \geq |\vec{a}'^I|$. Hence due to the phase freedom in $\vec{a}$ we can assume, without loss of generality, that $\vec{a}^R \cdot \vec{a}^I = 0$ and $|\vec{a}^R| \geq |\vec{a}^I|$. Since $\vec{a}^R$ and $\vec{a}^I$ are orthogonal and $|\vec{a}| = 1$ we can parameterize $\vec{a}$ as,

$$\vec{a} = \vec{a}^R + i\vec{a}^I = \cos(\theta)\hat{a}^R + i\sin(\theta)\hat{a}^I, \tag{60}$$

where $|\hat{a}^R| = |\hat{a}^I| = 1$ are unit vectors and $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. Now performing a unitary transformation on $\vec{a}\vec{a}^\dagger$ by $G \in \mathrm{SO}(3)$ we get,

$$\vec{a}\vec{a}^\dagger \xrightarrow{G} G\vec{a}\vec{a}^\dagger G^T = (G\vec{a})(G\vec{a})^\dagger. \tag{61}$$

Since $G \in \mathrm{SO}(3)$ is real it does not mix the real and imaginary parts of $\vec{a}$, so G simultaneously rotates the vectors $\vec{a}^R$ and $\vec{a}^I$. Hence G can be chosen to align the unit vectors $\hat{a}^R$ and $\hat{a}^I$ to the $+x$ and $-y$ axis respectively, that is,

$$G\hat{a}^R = \hat{x} \quad \text{and} \quad G\hat{a}^I = -\hat{y}. \tag{62}$$

So we can write,

$$\vec{a}(\theta) = G\vec{a} = (\cos(\theta), -i\sin(\theta), 0)^T, \tag{63}$$

where $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and,

$$A(\theta) = \vec{a}(\theta)\vec{a}(\theta)^{\dagger} = \begin{pmatrix} \cos^2(\theta) & i\cos(\theta)\sin(\theta) & 0 \\ -i\cos(\theta)\sin(\theta) & \sin^2(\theta) & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{64}$$

Hence we see that using the methods of linear combination of semigroups and unitary conjugation to simulate a Markovian semigroup, whose generator is specified by the GKS matrix A, it suffices to be able to simulate all Markovian semigroups whose generator is specified by the GKS matrix $A(\theta)$.

(Necessity) We now show that using linear combination and unitary conjugation it is not possible to simulate the Markovian semigroup whose generator is specified by the GKS matrix $A(\theta)$, through simulation of some other combination and/or transformation of Markovian semigroups whose generators are specified by GKS matrices of the form of $A(\theta')$.

Firstly we note that $A(\theta) = \vec{a}(\theta)\vec{a}(\theta)^{\dagger}$ is a rank 1 matrix and a projector onto the eigenspace of a single eigenvector of A. Since rank 1 matrices are extreme in the cone of positive semidefinite matrices [38], $A(\theta)$ cannot be written as a convex combination of positive semidefinite matrices. Hence no such $A(\theta)$ can be simulated using linear combination of semigroups that are specified by other such matrices $A(\theta')$.

We now note that multiplying the vector $\vec{a}$ by an overall phase $e^{i\psi}$ and the rotation $G \in \mathrm{SO}(3)$ commute. This tells us that to show that we cannot use unitary conjugation to simulate the Markovian semigroup whose generator is specified by $A(\theta)$, it suffices to show that, for an $\psi \in [0, 2\pi]$ and $G \in \mathrm{SO}(3)$ if

$$e^{i\psi} G\vec{a}(\theta) = \vec{a}(\theta'), \tag{65}$$

where $\theta, \theta' \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, then $\theta = \theta'$. To see this let $k_2' = k_2 = 0$ and $k_1', k_1 \geq 0$ in equation (58), this leads to the following two equations,

$$k_1' = \cos(2\psi)k_1 \quad \text{and} \quad \sin(2\psi)k_1 = 0. \tag{66}$$

Since by assumption $k_1', k_1 \geq 0$ we see that these equations are only satisfied when $\psi = 0$, hence the phase transformation $e^{i\psi}$ is trivial and leaves θ unchanged. Now we know that $\vec{a}(\theta) = \cos(\theta)\hat{x} - i\sin(\theta)\hat{y}$ so,

$$G\vec{a}(\theta) = \cos(\theta)G\hat{x} + i\sin(\theta)G(-\hat{y}) = \cos(\theta)\hat{x} - i\sin(\theta)\hat{y}. \tag{67}$$

From the above equation we see that G does not change θ, also since G was chosen to align the unit vectors $\hat{a}^R$ and $\hat{a}^I$ to the unit vectors $+\hat{x}$ and $-\hat{y}$ respectively. We see that $G\vec{a}(\theta) = \vec{a}(\theta)$. Hence we see that,

$$e^{i\psi} G\vec{a}(\theta) = a(\theta), \tag{68}$$

and hence $\theta = \theta'$. Therefore it is not possible to use linear combination of semigroups and unitary conjugation to simulate the Markovian semigroup, whose generator is specified $A(\theta)$, with Markovian semigroups whose generators are specified by matrices of the same form $A(\theta')$. □

Now by making use of Theorem 2 we can decompose the constituent channels $T_t^{(k)}$. We start with the GKS matrix $A_k = \vec{a}_k\vec{a}_k^{\dagger}$ then by choosing some $C_k \in \mathrm{SO}(3)$, and by performing a unitary transformation on $\vec{a}_k\vec{a}_k^{\dagger}$ with C_K we get:

$$C_k A_k C_k^T = C_k \vec{a}_k \vec{a}_k^{\dagger} C_k^T = (C_k\vec{a}_k)(C_k\vec{a}_k)^{\dagger} = \vec{a}(\theta_k)\vec{a}(\theta_k)^{\dagger} = A(\theta_k). \tag{69}$$

The above equation implies that,

$$A_k = C_k^T A(\theta_k) C_k. \tag{70}$$

Now we define the channel $T_t^{(\theta_k)} := \exp(t\mathcal{L}_{\theta_k})$, where,

$$\mathcal{L}_{\theta_k}(\rho) = \frac{1}{2}\sum_{i,j=1}^{3} (A(\theta_k))_{ij}([\sigma_i, \rho\sigma_j] + [\sigma_i\rho, \sigma_j]), \tag{71}$$

and the GKS matrix $A(\theta_k)$ is given in equation (64). Using the definitions above we see that unitary conjugation of the channel $T_t^{(\theta_k)}$ by $U_k \in \mathrm{SU}(2)$ conjugates the GKS matrix $A(\theta_k)$ by some $C_k \in \mathrm{SO}(3)$, i.e. equation (70). Hence unitary conjugation of the channel $T_t^{(\theta_k)}$ yields the channel $T_t^{(k)}$, i.e.

$$T_t^{(k)}(\rho) = (\mathcal{U}_k^{\dagger} T_t^{(\theta_k)} \mathcal{U}_k)(\rho) = U_k^{\dagger}[T_t^{(\theta_k)}(U_k\rho U_k^{\dagger})]U_k. \tag{72}$$

This tells us that if we wish to simulate the channels from the semigroup $\{T_t^{(k)}\}$ we need to be able to efficiently simulate channels from the universal semigroup $\{T_t^{(\theta_k)}\}$ and apply a unitary transformation $U_k \in \mathrm{SU}(2)$. In the next section we shall discuss a recombination strategy to use the constituent channels $T_t^{(k)}$ in second order SLT product formulas to approximate the channel T_t.

4 Recombination Strategy for Approximating T_t

In this section, using methods developed for simulating Hamiltonian dynamics [7, 11], we wish to show that we can use the second order Suzuki–Lie–Trotter (SLT) product formulas [13, 14] to simulate the channel T_t up to an arbitrary accuracy ϵ. In particular we wish to use a finite product of elements from the semigroup $\{T_t^{(k)}\}$ with $T_t^{(k)} := e^{t\mathcal{L}_k}$, to approximate the channel T_t and we also want to place a bound on the number of implementations of $T_t^{(k)}$ required in this product.

Given the generator of the channel T_t in the form as in equation (19) we can absorb the scalars λ_k into the superoperators $\mathcal{L}_k$ by defining, $\hat{\mathcal{L}}_k := \lambda_k \mathcal{L}_k$ and writing the generator as,

$$\mathcal{L} = \sum_{k=0}^{3} \hat{\mathcal{L}}_k. \tag{73}$$

Using the $1 \to 1$ superoperator norm we define the quantity $\Lambda := \max_{k\in\{0,1,2,3\}} \left\|\hat{\mathcal{L}}_k\right\|_{1\to1}$, this allows us to bound the $1 \to 1$ norm of the generator $\|\mathcal{L}\|_{1\to1}$ in the following way,

$$\|\mathcal{L}\|_{1\to1} = \left\|\left(\sum_{k=0}^{3} \hat{\mathcal{L}}_k\right)\right\|_{1\to1} \leq \sum_{k=0}^{3} \left\|\hat{\mathcal{L}}_k\right\|_{1\to1} \leq 4\Lambda. \tag{74}$$

We consider the total evolution $T_t = e^{t\mathcal{L}}$ and by defining the parameter $\tau := t/N$ we can write one N^{th} of the total evolution as,

$$T_\tau = e^{\tau\mathcal{L}} = \exp\left(\tau \sum_{k=0}^{3} \hat{\mathcal{L}}_k\right). \tag{75}$$

Now we define the second order SLT product formula $S_2(\alpha)$ for some $\alpha \geq 0$ [13, 14] as,

$$S_2(\alpha) = S_2(\hat{\mathcal{L}}_0, \hat{\mathcal{L}}_1, \hat{\mathcal{L}}_2, \hat{\mathcal{L}}_3, \alpha) = \prod_{k=0}^{3} e^{\frac{\alpha}{2}\hat{\mathcal{L}}_k} \prod_{k'=3}^{0} e^{\frac{\alpha}{2}\hat{\mathcal{L}}_{k'}}. \tag{76}$$

Now to calculate the bound on the error in the approximation of the total channel T_t we need to find the bound on the difference $\|T_\tau - S_2(\tau)\|_{1\to1}$ and then consider N applications of the SLT integrator $S_2(\tau)$. The following theorem shall encapsulate the bound on the error in our approximation as well as the number of implementations N required to approximate the total channel.

Theorem 3. Given a quantum channel $T_t = e^{t\mathcal{L}}$ with a generator $\mathcal{L} = \sum_{k=0}^{3} \hat{\mathcal{L}}_k$, and for $0 \leq \epsilon \leq 1$ then there exists some N such that,

$$\left\|\exp\left(\sum_{k=0}^{3} \hat{\mathcal{L}}_k\right) - S_2(\tau)^N\right\|_{1\to1} \leq \epsilon, \tag{77}$$

with $\epsilon \geq \frac{(4t\Lambda)^3}{3N^2}$ and $\tau = \frac{t}{N}$.

To prove Theorem 3 we require some results which are stated in the following lemmas below.

Lemma 3. For $\tau = \frac{t}{N} > 0$ we have that,

$$\left\|\exp\left(\sum_{k=0}^{3} \hat{\mathcal{L}}_k\right) - S_2(\tau)\right\|_{1\to1} \leq \frac{(4t\Lambda)^3}{3N^3}. \tag{78}$$

Proof. We know that $S_2(\tau)$ will approximate T_τ up to order τ^2, i.e. $\|T_\tau - S_2(\tau)\|_{1\to1} \in O(\tau^3)$, so we have that in the difference $T_\tau - S_2(\tau)$ the only remaining terms in the Taylor expansion are of order τ^3 so we have

$$\exp\left(\tau \sum_{k=0}^{3} \hat{\mathcal{L}}_k\right) - S_2(\tau) = \sum_{l=3}^{\infty} R_l(\tau) - W_l(\tau), \tag{79}$$

where $R_l(\tau)$ are the remaining terms in the Taylor expansion of $\exp\left(\tau \sum_{k=0}^{3} \hat{\mathcal{L}}_k\right)$ and $W_l(\tau)$ are the remaining terms in the Taylor expansion of $S_2(\tau)$. We can then write $R_l(\tau)$ as,

$$R_l(\tau) = \frac{\tau^l}{l!}\mathcal{L}^l = \frac{\tau^l}{l!}\left(\sum_{k=0}^{3} \hat{\mathcal{L}}_k\right)^l. \tag{80}$$

We can then bound this quantity by using the fact that $\|\mathcal{L}\|_{1\to1} \leq 4\Lambda$, so that we are able to write,

$$\|R_l(\tau)\|_{1\to1} = \left\|\frac{\tau^l}{l!}\mathcal{L}^l\right\|_{1\to1} = \frac{\tau^l}{l!}\left\|\mathcal{L}^l\right\|_{1\to1} \leq \frac{\tau^l}{l!}\|\mathcal{L}\|_{1\to1}^l \leq \frac{\tau^l}{l!}(4\Lambda)^l. \tag{81}$$

To write the form of the term $W_l(\tau)$ we need to consider the Taylor expansion of $S_2(\tau)$, which is given in terms of the product of Taylor expansions of each exponential in equation (76),

$$S_2(\tau) = \sum_{j_1,\ldots,j_8=0}^{\infty} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\hat{\mathcal{L}}_0^{j_1}\hat{\mathcal{L}}_1^{j_2}\hat{\mathcal{L}}_2^{j_3}\hat{\mathcal{L}}_3^{j_4}\hat{\mathcal{L}}_3^{j_5}\hat{\mathcal{L}}_2^{j_6}\hat{\mathcal{L}}_1^{j_7}\hat{\mathcal{L}}_0^{j_8}, \tag{82}$$

$$= \sum_{l=0}^{\infty} \sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\hat{\mathcal{L}}_0^{j_1}\hat{\mathcal{L}}_1^{j_2}\hat{\mathcal{L}}_2^{j_3}\hat{\mathcal{L}}_3^{j_4}\hat{\mathcal{L}}_3^{j_5}\hat{\mathcal{L}}_2^{j_6}\hat{\mathcal{L}}_1^{j_7}\hat{\mathcal{L}}_0^{j_8}, \tag{83}$$

where we have converted the eight infinite sums in (83) to eight finite sums by restricting $j_1, \ldots, j_8$ such that they sum to l and one infinite sum. This allows us to write the term $W_l(\tau)$ as,

$$W_l(\tau) = \sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\hat{\mathcal{L}}_0^{j_1}\hat{\mathcal{L}}_1^{j_2}\hat{\mathcal{L}}_2^{j_3}\hat{\mathcal{L}}_3^{j_4}\hat{\mathcal{L}}_3^{j_5}\hat{\mathcal{L}}_2^{j_6}\hat{\mathcal{L}}_1^{j_7}\hat{\mathcal{L}}_0^{j_8}. \tag{84}$$

To bound the term $W_l(\tau)$ we use the fact that $\|\cdot\|_{1\to1}$ is sub-multiplicative and sub-additive, so that we can write,

$$\begin{aligned}
\|W_l(\tau)\|_{1\to1} &= \left\|\sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\hat{\mathcal{L}}_0^{j_1}\hat{\mathcal{L}}_1^{j_2}\hat{\mathcal{L}}_2^{j_3}\hat{\mathcal{L}}_3^{j_4}\hat{\mathcal{L}}_3^{j_5}\hat{\mathcal{L}}_2^{j_6}\hat{\mathcal{L}}_1^{j_7}\hat{\mathcal{L}}_0^{j_8}\right\|_{1\to1}, \\
&\leq \sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\left\|\hat{\mathcal{L}}_0^{j_1}\hat{\mathcal{L}}_1^{j_2}\hat{\mathcal{L}}_2^{j_3}\hat{\mathcal{L}}_3^{j_4}\hat{\mathcal{L}}_3^{j_5}\hat{\mathcal{L}}_2^{j_6}\hat{\mathcal{L}}_1^{j_7}\hat{\mathcal{L}}_0^{j_8}\right\|_{1\to1}, \\
&\leq \sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\left\|\hat{\mathcal{L}}_0^{j_1}\right\|_{1\to1}\left\|\hat{\mathcal{L}}_1^{j_2}\right\|_{1\to1}\cdots\left\|\hat{\mathcal{L}}_0^{j_8}\right\|_{1\to1}, \\
&\leq \sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\left\|\hat{\mathcal{L}}_0\right\|_{1\to1}^{j_1}\left\|\hat{\mathcal{L}}_1\right\|_{1\to1}^{j_2}\cdots\left\|\hat{\mathcal{L}}_0\right\|_{1\to1}^{j_8}, \\
&\leq \sum_{\substack{j_1,\ldots,j_8=0 \\ \sum_m j_m = l}}^{l} \frac{(\tau/2)^{j_1+\ldots+j_8}}{j_1!j_2!\ldots j_8!}\Lambda^{j_1+j_2+\ldots+j_8}.
\end{aligned} \tag{85}$$

To calculate the sum in equation (85) we must consider the Taylor expansion of the following expression,

$$\begin{aligned}\exp\left(\frac{x}{2}\right)^8 &= \exp\left(\frac{x}{2}\right)\exp\left(\frac{x}{2}\right)...\exp\left(\frac{x}{2}\right),\\ &= \Big(\sum_{j_1=0}^{\infty}\frac{(x/2)^{j_1}}{j_1!}\Big)\Big(\sum_{j_2=0}^{\infty}\frac{(x/2)^{j_2}}{j_2!}\Big)...\Big(\sum_{j_8=0}^{\infty}\frac{(x/2)^{j_8}}{j_8!}\Big),\\ &= \sum_{j_1,j_2,...,j_8=0}^{\infty}\frac{1}{j_1!j_2!...j_8!}\left(\frac{x}{2}\right)^{j_1+j_2+...+j_8},\\ &= \sum_{p=0}^{\infty}\sum_{\substack{j_1,j_2,...,j_8=0\\ \sum_m j_m=p}}^{p}\frac{1}{j_1!j_2!...j_8!}\left(\frac{x}{2}\right)^{j_1+j_2+...+j_8}.\end{aligned} \tag{86}$$

But we know that,

$$\exp\left(\frac{x}{2}\right)^8 = \exp(4x) = \sum_{p=0}^{\infty}\frac{4^p x^p}{p!}. \tag{87}$$

Now by equating equations (86) and (87) we see that,

$$\sum_{\substack{j_1,j_2,...,j_8=0\\ \sum_m j_m=p}}^{p}\frac{1}{j_1!j_2!...j_8!}\left(\frac{x}{2}\right)^{j_1+j_2+...+j_8} = \frac{4^p x^p}{p!}. \tag{88}$$

By using $x = \tau\Lambda$ in equation (88)we can write the sum in equation (85) as,

$$\sum_{\substack{j_1,...,j_8=0\\ \sum_m j_m=l}}^{l}\frac{(\tau/2)^{j_1+...+j_8}}{j_1!j_2!...j_8!}\Lambda^{j_1+j_2+...+j_8} = \frac{4^l\tau^l\Lambda^l}{l!}, \tag{89}$$

this implies that the bound on $||W_l(\tau)||_{1\to1}$ is,

$$||W_l(\tau)||_{1\to1} \leq \frac{4^l\tau^l\Lambda^l}{l!}. \tag{90}$$

Now, using sub-additive property of the 1→1 superoperator norm, we can write

$$\begin{aligned}\left\|\exp\left(\tau\sum_{k=0}^{3}\hat{\mathcal{L}}_k\right) - S_2(\tau)\right\|_{1\to1} &= \left\|\sum_{l=3}^{\infty}R_l(\tau) - W_l(\tau)\right\|_{1\to1},\\ &\leq \sum_{l=3}^{\infty}||R_l(\tau)||_{1\to1} + ||W_l(\tau)||_{1\to1},\\ &\leq 2\sum_{l=3}\frac{(4\tau\Lambda)^l}{l!}.\end{aligned} \tag{91}$$

Now by using the Lemma F.2 from the supplementary information of [4] we have that for $y > 0$,

$$\sum_{n=k}^{\infty}\frac{y^n}{n!} \leq \frac{y^k}{k!}\exp(y). \tag{92}$$

Using this upper bound on the remainder terms in the Taylor expansion of the exponential function we can write,

$$\left\|\exp\left(\tau\sum_{k=0}^{3}\hat{\mathcal{L}}_k\right) - S_2(\tau)\right\|_{1\to1} \leq 2\frac{(4\tau\Lambda)^3}{3!}\exp(4\tau\Lambda) = \frac{(4t\Lambda)^3}{3N^3}\exp\left(\frac{4t\Lambda}{N}\right). \tag{93}$$

For large enough N we can approximate $\exp\left(\frac{4t\Lambda}{N}\right) \approx 1$ which gives the desired result,

$$\left\| \exp\left(\tau \sum_{k=0}^{3} \hat{\mathcal{L}}_k\right) - S_2(\tau) \right\|_{1\to 1} \leq \frac{(4t\Lambda)^3}{3N^3}, \tag{94}$$

completing the proof. □

Lemma 4. For quantum channels T and V and for some $N \geq 0 \in \mathbb{Z}$,

$$\left\| T^N - V^N \right\|_{1\to 1} \leq N \left\| T - V \right\|_{1\to 1}. \tag{95}$$

Proof. We prove this by induction. For $N = 0, 1$ the equality in equation (95) holds and to show this would be trivial, so for the base case in our inductive proof we choose $N = 2$,

$$\begin{aligned} \left\| T^2 - V^2 \right\|_{1\to 1} &= \left\| T^2 - TV + TV - V^2 \right\|_{1\to 1}, \\ &= \left\| T(T - V) + (T - V)V \right\|_{1\to 1}, \\ &\leq \left\| T(T - V) \right\|_{1\to 1} + \left\| (T - V)V \right\|_{1\to 1}, \\ &=\leq \left\| T \right\|_{1\to 1} \left\| T - V \right\|_{1\to 1} + \left\| T - V \right\|_{1\to 1} \left\| V \right\|_{1\to 1}. \end{aligned} \tag{96}$$

By recalling that for any quantum channel T by definition $\|T\|_{1\to 1} = 1$, this allows us to write,

$$\left\| T^2 - V^2 \right\|_{1\to 1} \leq \left\| T - V \right\|_{1\to 1} + \left\| T - V \right\|_{1\to 1} = 2 \left\| T - V \right\|_{1\to 1}. \tag{97}$$

Hence we have verified that the inequality in equation (95) holds for $N = 2$, we now assume that it holds for $N = m$ and show that it is true for $N = m + 1$.

$$\begin{aligned} \left\| T^{m+1} - V^{m+1} \right\|_{1\to 1} &= \left\| T^{m+1} - TV^m + TV^m - V^{m+1} \right\|_{1\to 1}, \\ &= \left\| T(T^m - V^m) + (T - V)V^m \right\|_{1\to 1}, \\ &\leq \left\| T \right\|_{1\to 1} \left\| T^m - V^m \right\|_{1\to 1} + \left\| T - V \right\|_{1\to 1} \left\| V^m \right\|_{1\to 1}, \\ &\leq m \left\| T - V \right\|_{1\to 1} + \left\| T - V \right\|_{1\to 1} \left\| V \right\|_{1\to 1}^m, \\ &\leq (m + 1) \left\| T - V \right\|_{1\to 1}. \end{aligned} \tag{98}$$

Therefore by induction the inequality in (95) holds true for all integers $N \geq 0$. □

We are now able to write the proof for Theorem 3 using the above two lemmas.

Proof of Theorem 3. Given that $\tau = t/N$ we can write $\exp(t\mathcal{L}) = \exp(\tau\mathcal{L})^N$ which allows us to write,

$$\begin{aligned} \left\| \exp(t\mathcal{L}) - (S_2(\tau))^N \right\|_{1\to 1} &= \left\| \exp(\tau\mathcal{L})^N - (S_2(\tau))^N \right\|_{1\to 1}, \\ &\leq N \left\| \exp(\tau\mathcal{L}) - S_2(\tau) \right\|_{1\to 1}, \\ &\leq N \frac{(4t\Lambda)^3}{3N^3} = \frac{(4t\Lambda)^3}{3N^2}, \end{aligned} \tag{99}$$

where in the second line of (99) we used Lemma 4 and in the third line we use Lemma 3. By choosing $\epsilon \geq \frac{(4t\Lambda)^3}{3N^2}$ we get that $\left\| \exp(t\mathcal{L}) - (S_2(\tau))^N \right\|_{1\to 1} \leq \epsilon$, which completes the proof. □

Finally, using the fact that $\epsilon \geq \frac{(4t\Lambda)^3}{3N^2}$ we can obtain a bound on the number of implementations of $T_t^{(k)}$ required to simulate T_t up to a precision ϵ. We first see that,

$$N \geq \frac{(4t\Lambda)^{3/2}}{(3\epsilon)^{1/2}}, \tag{100}$$

this tells us that we have at least $\frac{(4t\Lambda)^{3/2}}{(3\epsilon)^{1/2}}$ implementations of $S_2(\tau)$, and in each $S_2(\tau)$ we have seven constituent channels $T_t^{(k)}$ which implies that the number of channels $T_t^{(k)}$ needed to simulate the channel T_t, which will be denoted by N_{exp} is,

$$N_{\text{exp}} \geq 7N = \frac{7(4t\Lambda)^{3/2}}{(3\epsilon)^{1/2}}. \tag{101}$$

This places a bound on the number of implementations of $T_t^{(k)}$ needed to simulate T_t and this bound will be used in the next sections to place a bound on the number of single qubit and CNOT gates needed to simulate the channel T_t.

5 Simulation of the Constituent Channels

We have seen from equation (72), that to simulate the channels $T_t^{(k)}$ we need to be able to simulate channels from the semigroup $\{T_t^{(\theta_k)}\}$ as well as apply some unitary transformation. We note that since the unitary transformation comes from SU(2) it can be easily implemented with a single qubit unitary gate. So we are interested in finding a quantum circuit that can simulate channels from the semigroup $\{T_t^{(\theta_k)}\}$.

We do this using the techniques from [34] to decompose the channel $T_t^{(\theta_k)}$ into a convex sum of quasi-extreme channels, which are channels that are generalised extreme points of the space of quantum channels and have only 2 Kraus operators. We can then use the Stinespring representation [39] to find quantum circuits that correspond to these quasi-extreme channels. Lastly we use the method of quantum forking [30] to implement the convex mixture of these quasi-extreme channels. This will give us a quantum circuit that simulates the channel $T_t^{(\theta_k)}$ for some $\theta_k \in [-\frac{\pi}{4}, \frac{\pi}{4}]$.

5.1 Calculation of the Choi matrix of the channel $T_t^{(\theta_k)}$

We now consider the channel $T_t^{(\theta_k)} = \exp(t\mathcal{L}_{\theta_k})$, whose generator is given in equation (71). We now make use of the matrix representation of the generator to find the matrix representation of the channel. We shall denote the matrix representation of a channel or its generator by bold face. The generator $\mathcal{L}_{\theta_k}$ has the following matrix representation,

$$(\mathbf{L}_{\theta_k})_{a,b} = \operatorname{tr}[G_a^\dagger \mathcal{L}_{\theta_k}(G_b)], \tag{102}$$

where the operators $\{G_a\}$ is a basis for the space $\mathcal{M}_2(\mathbb{C})$, and is given as

$$\{G_1 = \frac{1}{\sqrt{2}}\mathbb{1}, G_2 = \frac{1}{\sqrt{2}}\sigma_1, G_3 = \frac{1}{\sqrt{2}}\sigma_2, G_4 = \frac{1}{\sqrt{2}}\sigma_3\}. \tag{103}$$

Now we note that the choice of this basis leads to a matrix representation of the channel that is called the affine map representation. Now we can use equation (71) and (102) to find the matrix representation of the generator $\mathcal{L}_{\theta_k}$ as,

$$\mathbf{L}_{\theta_k} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -2\sin^2(\theta_k) & 0 & 0 \\ 0 & 0 & -2\cos^2(\theta_k) & 0 \\ -4\cos(\theta_k)\sin(\theta_k) & 0 & 0 & -2 \end{pmatrix}. \tag{104}$$

Now to find the matrix representation of the channel $T_t^{(\theta_k)}$ we make use of the following,

$$\mathbf{T}_t^{(\theta_k)} = \exp(t\mathbf{L}_{\theta_k}). \tag{105}$$

This leads to the following matrix representation of $T_t^{(\theta_k)}$,

$$\mathbf{T}_t^{(\theta_k)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ m_3 & 0 & 0 & \lambda_3 \end{pmatrix}, \tag{106}$$

where

$$\begin{aligned}
\lambda_1 &= \exp\Big(-2t\sin^2(\theta_k)\Big),\\
\lambda_2 &= \exp\Big(-2t\cos^2(\theta_k)\Big),\\
\lambda_3 &= \exp(-2t),\\
m_3 &= \sin(2\theta_k)(\lambda_3 - 1).
\end{aligned} \tag{107}$$

Now to obtain the desired convex decomposition of the channel $T_t^{(\theta_k)}$, we need to find the Choi matrix [33] of this channel, i.e. $\tau_{(\theta_k)} = (T_t^{(\theta_k)} \otimes \mathbb{1})|\Omega\rangle\langle\Omega|$, where $|\Omega\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. We do this using the following relation [40],

$$\tau_{(\theta_k)} = \frac{1}{4}\sum_{i,j=0}^{3} (\mathbf{T}_t^{(\theta_k)})_{i+1,j+1}(\sigma_i \otimes \sigma_j^T), \tag{108}$$

where $\sigma_0 = \mathbb{1}$. Using equation (108) we find the Choi matrix to be,

$$\tau_{(\theta_k)} = \frac{1}{4}\begin{pmatrix} a^2 & 0 & 0 & \lambda_1+\lambda_2 \\ 0 & b^2 & \lambda_1-\lambda_2 & 0 \\ 0 & \lambda_1-\lambda_2 & c^2 & 0 \\ \lambda_1+\lambda_2 & 0 & 0 & d^2 \end{pmatrix}, \tag{109}$$

with

$$\begin{aligned}
a &= (1+\lambda_3+m_3)^{\frac{1}{2}},\\
b &= (1-\lambda_3+m_3)^{\frac{1}{2}},\\
c &= (1-\lambda_3-m_3)^{\frac{1}{2}},\\
d &= (1+\lambda_3-m_3)^{\frac{1}{2}}.
\end{aligned} \tag{110}$$

5.2 Convex decomposition of $T_t^{(\theta_k)}$ into quasi-extreme channels

At this point it is useful to adopt the notation used in [34] and define the Choi matrix in terms of the matrix in terms of the matrix $\beta(T_t^{(\theta_k)}) = 2\tau_{(\theta_k)}$, this implies that $\tau_{(\theta_k)} = \frac{1}{2}\beta(T_t^{(\theta_k)})$.

The $\beta(\cdot)$ matrix defined in terms of the Choi matrix plays a significant role as this definition implies that β defines an affine isomorphism between space of all CPTP maps on the space of states of a single qubit, denoted by Q, and the image $\beta(Q) \subset \mathcal{M}_4(\mathbb{C})$. In particular, there is a one-to-one correspondence between extreme points of Q and those of the image $\beta(Q)$. This will correspondence will allow us to determine the extreme points of Q, which will aid in decomposing $T_t^{(\theta_k)}$ into quasi-extreme channels.

From this we can also find the Choi matrix of the dual channel $\tilde{T}_t^{(\theta_k)}$ using the following relation,

$$\beta(\tilde{T}_t^{(\theta_k)}) = (U_{23}^\dagger \beta(T_t^{(\theta_k)})U_{23})^*, \tag{111}$$

where U_{23} is the permutation matrix that swaps the second and third rows of a 4×4 matrix and is defined as,

$$U_{23} = U_{23}^\dagger = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{112}$$

Using equation (111) we find the matrix $\beta(\tilde{T}_t^{(\theta_k)})$,

$$\beta(\tilde{T}_t^{(\theta_k)}) = \frac{1}{2}\begin{pmatrix} a^2 & 0 & 0 & \lambda_1+\lambda_2 \\ 0 & c^2 & \lambda_1-\lambda_2 & 0 \\ 0 & \lambda_1-\lambda_2 & b^2 & 0 \\ \lambda_1+\lambda_2 & 0 & 0 & d^2 \end{pmatrix}. \tag{113}$$

Now the Choi matrix corresponding to the dual channel, $\tilde{T}_t^{(\theta_k)}$, is $\tilde{\tau}_{(\theta_k)} = \frac{1}{2}\beta(\tilde{T}_t^{(\theta k)})$. We shall now make use of the following two lemmas and theorem from [34], to obtain the convex decomposition of $\tau_{(\theta_k)}$, we state them below in a modified form to suit our notation.

Lemma 5. Any contraction in $\mathcal{M}_2(\mathbb{C})$ can be written as the convex combination of two unitary matrices.

Proof. Let $R \in \mathcal{M}_2(\mathbb{C})$ be a contraction then $\|R\| \leq 1$, and its singular value decomposition can be written in the form

$$R = V\begin{pmatrix}\cos(\theta_1) & 0\\ 0 & \cos(\theta_2)\end{pmatrix}W^\dagger, \tag{114}$$

where $V, W^\dagger \in \mathrm{U}(2)$. Now by using the fact that $\cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta})$, we have that,

$$\begin{aligned} R &= V\frac{1}{2}\begin{pmatrix}e^{i\theta_1} + e^{-i\theta_1} & 0\\ 0 & e^{i\theta_2} + e^{-i\theta_2}\end{pmatrix}W^\dagger,\\ &= \frac{1}{2}V\begin{pmatrix}e^{i\theta_1} & 0\\ 0 & e^{i\theta_2}\end{pmatrix}W^\dagger + \frac{1}{2}V\begin{pmatrix}e^{-i\theta_1} & 0\\ 0 & e^{-i\theta_2}\end{pmatrix}W^\dagger. \end{aligned} \tag{115}$$

From the above it is clear that a contraction R can be written as a convex combination of two unitary matrices. □

Lemma 6. Given a matrix

$$J = \begin{pmatrix}A & C\\ C^\dagger & B\end{pmatrix}, \tag{116}$$

(1) J is positive semidefinite if and only if $A \geq 0$, $B \geq 0$ and $C = \sqrt{A}R\sqrt{B}$ for some contraction R.
(2) Moreover the set of positive semidefinite matrices with fixed A and B is a convex set whose extreme points satisfy $C = \sqrt{A}U\sqrt{B}$, for some unitary matrix U.

Proof. The proof of (1) follows from [41], suppose that $A, B \geq 0$ then by theorem (7.7.7) in [42] the block matrix J is positive semidefinite if and only if

$$\begin{aligned} 1 \geq \varrho(C^\dagger A^{-1}CB^{-1}) &= \varrho(C^\dagger A^{-1}CB^{-\frac{1}{2}}B^{-\frac{1}{2}}), && (117)\\ &= \varrho(B^{-\frac{1}{2}}C^\dagger A^{-1}CB^{-\frac{1}{2}}), \quad \text{since } \varrho(AB) = \varrho(BA) \ \ \forall A, B \in \mathcal{M}_n(\mathbb{C}) &&\\ &= \varrho(B^{-\frac{1}{2}}C^\dagger A^{-\frac{1}{2}}A^{-\frac{1}{2}}CB^{-\frac{1}{2}}), && (118)\\ &= \varrho((A^{-\frac{1}{2}}CB^{-\frac{1}{2}})^\dagger(A^{-\frac{1}{2}}CB^{-\frac{1}{2}})), &&\\ &= \varsigma(A^{-\frac{1}{2}}CB^{-\frac{1}{2}})^2, \quad \text{since } \varrho(A^\dagger A) = \varsigma(A)^2, && (119) \end{aligned}$$

where $\varrho(\cdot)$ is the spectral radius and $\varsigma(\cdot)$ the spectral norm. Setting $R = A^{-\frac{1}{2}}CB^{-\frac{1}{2}}$ (since $\varsigma(R) \leq 1 \implies R$ is a contraction) we have $C = \sqrt{A}R\sqrt{B}$ as desired.
The proof of (2) follows from a well know result that the extreme points of the set of contractions in $\mathcal{M}_2(\mathbb{C})$ are unitary (see [41,43]). Now if we define the affine mapping $f : R \mapsto \sqrt{A}R\sqrt{B}$ for a fixed $A, B \geq 0$, where dom(f) is the set of positive semidefinite matrices in $\mathcal{M}_2(\mathbb{C})$ and Im$(f) \subset \mathcal{M}_2(\mathbb{C})$. Now the set of contractions in $\mathcal{M}_2(\mathbb{C})$ is a compact and convex set, and extreme points of the Im(f) are images of extreme points [34]. If $R \in$ Dom(f) is an extreme point then it is unitary, and setting $U = R$ we have that $C = f(R) = f(U) = \sqrt{A}U\sqrt{B}$ [34,41]. □

The following theorem, from [34], is stated without proof as the proof is lengthy and can be found in [34].

Theorem 4. A quantum channel T_t is a generalised extreme point of the set of all quantum channels of the same dimension if and only if $\beta(\tilde{T}_t)$ is of the form,

$$\beta(\tilde{T}_t) = \begin{pmatrix}A & \sqrt{A}U\sqrt{B}\\ \sqrt{B}U^\dagger\sqrt{A} & B\end{pmatrix}, \tag{120}$$

for some unitary matrix U.

We can now find the convex decomposition of $\tau_{(\theta_k)}$. We start by writing the matrix $\beta(\tilde{T}_t^{(\theta_k)})$ in block form as follows:

$$\beta(\tilde{T}_t^{(\theta_k)}) = \begin{pmatrix} A & C \\ C^\dagger & B \end{pmatrix}, \tag{121}$$

where

$$A = \frac{1}{2}\begin{pmatrix} a^2 & 0 \\ 0 & c^2 \end{pmatrix}, \qquad B = \frac{1}{2}\begin{pmatrix} b^2 & 0 \\ 0 & d^2 \end{pmatrix}, \qquad C = \frac{1}{2}\begin{pmatrix} 0 & \lambda_1 + \lambda_2 \\ \lambda_1 - \lambda_2 & 0 \end{pmatrix}. \tag{122}$$

Now since $\beta(\tilde{T}_t^{(\theta_k)}) \geq 0$, we can make use of Lemma 4 so that we can write $C = \sqrt{A}R\sqrt{B}$, for some contraction $R \in \mathcal{M}_2(\mathbb{C})$, which leads to,

$$\beta(\tilde{T}_t^{(\theta_k)}) = \begin{pmatrix} A & \sqrt{A}R\sqrt{B} \\ \sqrt{B}R^\dagger\sqrt{A} & B \end{pmatrix}. \tag{123}$$

Now we can compute the contraction R from the fact that $C = \sqrt{A}R\sqrt{B}$,

$$\begin{aligned} R &= A^{-\frac{1}{2}}CB^{-\frac{1}{2}}, \\ &= \sqrt{2}\begin{pmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{c} \end{pmatrix} \frac{1}{2}\begin{pmatrix} 0 & \lambda_1 + \lambda_2 \\ \lambda_1 - \lambda_2 & 0 \end{pmatrix} \sqrt{2}\begin{pmatrix} \frac{1}{b} & 0 \\ 0 & \frac{1}{d} \end{pmatrix}, \\ &= \begin{pmatrix} 0 & \frac{\lambda_1+\lambda_2}{ad} \\ \frac{\lambda_1-\lambda_2}{bc} & 0 \end{pmatrix}. \end{aligned} \tag{124}$$

Now by making use of Lemma 3 we know that any contraction in $\mathcal{M}_2(\mathbb{C})$ can be written as a convex sum of two unitary matrices, so if we let $V = \mathbb{1}$ and $W^\dagger = \sigma_1$ in equation (114) we have:

$$R = \begin{pmatrix} 0 & \cos(\phi_1) \\ \cos(\phi_2) & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{\lambda_1+\lambda_2}{ad} \\ \frac{\lambda_1-\lambda_2}{bc} & 0 \end{pmatrix}, \tag{125}$$

this gives the angles ϕ_1 and ϕ_2 as,

$$\phi_1 = \arccos\left(\frac{\lambda_1 + \lambda_2}{ad}\right), \qquad \phi_2 = \arccos\left(\frac{\lambda_1 - \lambda_2}{bc}\right). \tag{126}$$

By making use of equation (115) we can write R as,

$$R = \frac{1}{2}U_1 + \frac{1}{2}U_2, \tag{127}$$

where

$$U_1 = \begin{pmatrix} 0 & e^{i\phi_1} \\ e^{i\phi_2} & 0 \end{pmatrix}, \qquad U_2 = \begin{pmatrix} 0 & e^{-i\phi_1} \\ e^{-i\phi_2} & 0 \end{pmatrix}. \tag{128}$$

We can now write,

$$\begin{aligned} \beta(\hat{T}_t^{(\theta_k)}) &= \begin{pmatrix} A & \sqrt{A}(\frac{1}{2}U_1 + \frac{1}{2}U_2)\sqrt{B} \\ \sqrt{B}(\frac{1}{2}U_1^\dagger + \frac{1}{2}U_2^\dagger)\sqrt{A} & B \end{pmatrix}, \\ &= \frac{1}{2}\begin{pmatrix} A & \sqrt{A}U_1\sqrt{B} \\ \sqrt{B}U_1^\dagger\sqrt{A} & B \end{pmatrix} + \frac{1}{2}\begin{pmatrix} A & \sqrt{A}U_2\sqrt{B} \\ \sqrt{B}U_2^\dagger\sqrt{A} & B \end{pmatrix}. \end{aligned} \tag{129}$$

If we define,

$$\beta(\tilde{T}_{(t,j)}^{(\theta_k)}) = \begin{pmatrix} A & \sqrt{A}U_j\sqrt{B} \\ \sqrt{B}U_j^\dagger\sqrt{A} & B \end{pmatrix}, \tag{130}$$

for $j = 1, 2$, then we can write:

$$\beta(\tilde{T}_t^{(\theta_k)}) = \frac{1}{2}\beta(\tilde{T}_{(t,1)}^{(\theta_k)}) + \frac{1}{2}\beta(\tilde{T}_{(t,2)}^{(\theta_k)}). \tag{131}$$

Now by Theorem 4 the matrix $\beta(\tilde{T}^{(\theta_k)}_{(t,j)})$ is a generalised extreme point so the channel $T^{(\theta_k)}_{(t,j)}$ is a generalised extreme point in the space of channels [34], also known as a quasi extreme channel. We can now begin to find the convex decomposition of the Choi matrix $\tau_{(\theta_k)}$ in terms of the Choi matrices of these quasi extreme channels, as follows. By making use of equation (111) as well as the fact that $\tau_{(\theta_k)} = \frac{1}{2}\beta(T^{(\theta_k)}_t)$ we can find the desired convex decomposition of the Choi matrix $\tau_{(\theta_k)}$ in the following way. From equation (111) and (131) we have that,

$$\begin{aligned}\beta(T^{(\theta_k)}_t) &= U_{23}\beta(\tilde{T}^{(\theta_k)}_t)^* U^\dagger_{23},\\ &= U_{23}(\frac{1}{2}\beta(\tilde{T}^{(\theta_k)}_{(t,1)})^* + \frac{1}{2}\beta(\tilde{T}^{(\theta_k)}_{(t,2)})^*)U^\dagger_{23},\\ &= \frac{1}{2}U_{23}\beta(\tilde{T}^{(\theta_k)}_{(t,1)})^* U^\dagger_{23} + \frac{1}{2}U_{23}\beta(\tilde{T}^{(\theta_k)}_{(t,2)})^* U^\dagger_{23},\\ &= \frac{1}{2}\beta(T^{(\theta_k)}_{(t,1)}) + \frac{1}{2}\beta(T^{(\theta_k)}_{(t,2)}), \end{aligned} \tag{132}$$

where $\beta(T^{(\theta_k)}_{(t,j)}) = U_{23}\beta(\tilde{T}^{(\theta_k)}_{(t,j)})^* U^\dagger_{23}$, for $j = 1, 2$. Now by defining, $\tau_{(\theta_k,j)} = \frac{1}{2}\beta(T^{(\theta_k)}_{(t,j)})$, we can write the convex decomposition of $\tau_{(\theta_k)}$ as,

$$\tau_{(\theta_k)} = \frac{1}{2}\tau_{(\theta_k,1)} + \frac{1}{2}\tau_{(\theta_k,2)}. \tag{133}$$

Using equations (111), (122), (128) and (130) we can find the Choi matrices $\tau_{(\theta_k,j)}$ for $j = 1, 2$ to be,

$$\tau_{(\theta_k,1)} = \frac{1}{4}\begin{pmatrix} a^2 & 0 & 0 & ade^{-i\phi_1} \\ 0 & b^2 & bce^{i\phi_2} & 0 \\ 0 & bce^{-i\phi_2} & c^2 & 0 \\ ade^{i\phi_1} & 0 & 0 & d^2 \end{pmatrix},$$

$$\tau_{(\theta_k,2)} = \frac{1}{4}\begin{pmatrix} a^2 & 0 & 0 & ade^{i\phi_1} \\ 0 & b^2 & bce^{-i\phi_2} & 0 \\ 0 & bce^{i\phi_2} & c^2 & 0 \\ ade^{-i\phi_1} & 0 & 0 & d^2 \end{pmatrix}. \tag{134}$$

Now by the Choi–Jamiołkowski isomorphism [32] and equation (133), we have the desired convex decomposition of the channel $T^{(\theta_k)}_t$ as a convex sum of two quasi extreme channels, i.e.

$$T^{(\theta_k)}_t(\rho) = \frac{1}{2}T^{(\theta_k)}_{(t,1)}(\rho) + \frac{1}{2}T^{(\theta_k)}_{(t,2)}(\rho). \tag{135}$$

Using the Choi matrices in equation (134), we can find the Kraus maps corresponding to the channels $T^{(\theta_k)}_{(t,1)}$ and $T^{(\theta_k)}_{(t,2)}$, we do this as follows. We make use of the following formula from [40], for any Choi matrix $\tau = (T \otimes \mathbb{1})|\Omega\rangle\langle\Omega|$ we have $\tau \geq 0$ which implies,

$$\tau = \sum_j |\psi_j\rangle\langle\psi_j| = \sum_j (K_j \otimes \mathbb{1})|\Omega\rangle\langle\Omega|(K_j \otimes \mathbb{1})^\dagger, \tag{136}$$

where each $\left|\psi_j\right\rangle$ is the product of j-th eigenvector of τ with its corresponding eigenvalue. Using the formula above and equation (134) we can obtain the Kraus maps for the channels $T^{(\theta_k)}_{(t,1)}$ and $T^{(\theta_k)}_{(t,2)}$. For the channel $T^{(\theta_k)}_{(t,1)}$ we have,

$$T^{(\theta_k)}_{(t,1)}(\rho) = \sum_{j=1}^{2} K^{(1)}_j \rho K^{(1)\dagger}_j, \tag{137}$$

where

$$K^{(1)}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} ae^{-i\phi_1} & 0 \\ 0 & d \end{pmatrix}, \qquad K^{(1)}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & be^{i\phi_2} \\ c & 0 \end{pmatrix}. \tag{138}$$

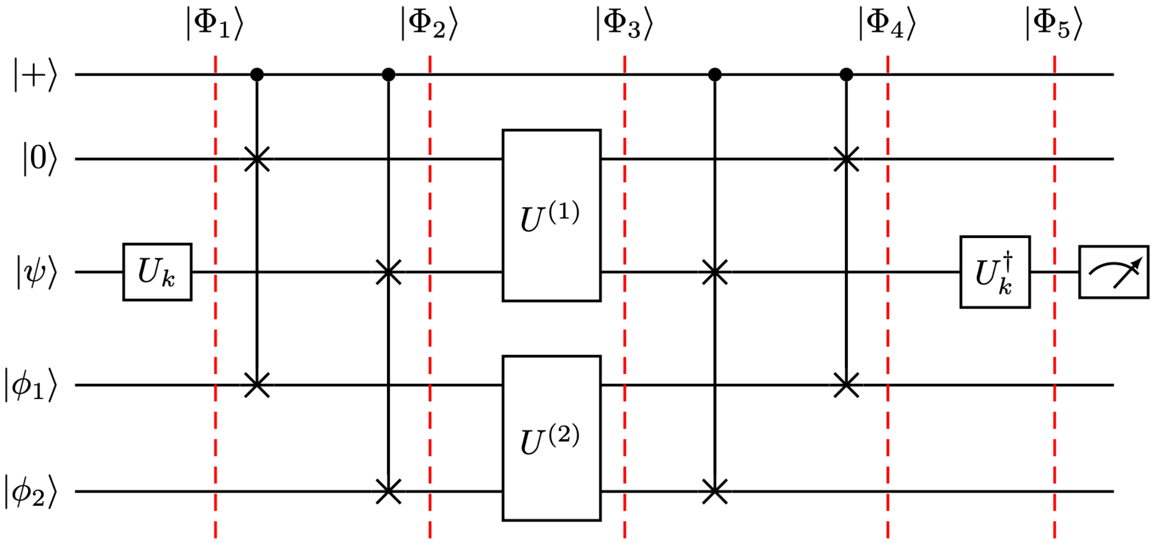


Figure 1: *The quantum circuit to implement the channel $T_t^{(k)}$. Where $|\psi\rangle$ is the system qubit, the qubit in state $|+\rangle$ is called the ancilla lablled a, the qubit in $|0\rangle$ is the environment E and the states $|\phi_1\rangle$ and $|\phi_2\rangle$ are arbitrary states needed for the quantum forking algorithm that implements the convex mixture of the extreme channels.*

For the channel $T_{(t,2)}^{(\theta_k)}$ we have,

$$T_{(t,2)}^{(\theta_k)}(\rho) = \sum_{j=1}^{2} K_j^{(2)} \rho K_j^{(2)\dagger}, \tag{139}$$

where

$$K_1^{(2)} = \frac{1}{\sqrt{2}} \begin{pmatrix} ae^{i\phi_1} & 0 \\ 0 & d \end{pmatrix}, \qquad K_2^{(2)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & be^{-i\phi_2} \\ c & 0 \end{pmatrix}. \tag{140}$$

5.3 Construction of circuits for the simulation of $T_t^{(k)}$

Now that we have obtained a decomposition of $T_t^{(\theta_k)}$ as a convex sum of quasi extreme channels we now begin to find a quantum circuit that can implement this convex sum of two quasi extreme channels so that we can simulate the channel $T_t^{(\theta_k)}$ on a quantum computer. To do this we first need to find unitary operators that correspond to the quasi extreme channels, for this we make use of the Stinespring Representation of the channel [39]. For more information on the Stinespring representation for quantum channels refer to Appendix B. From the Stinespring representation and equations (137) and (139) we have,

$$T_{(t,1)}^{(\theta_k)}(\rho) = \mathrm{tr}_E[U^{(1)}(|0\rangle\langle 0| \otimes \rho)U^{(1)\dagger}], \tag{141}$$

$$T_{(t,2)}^{(\theta_k)}(\rho) = \mathrm{tr}_E[U^{(2)}(|0\rangle\langle 0| \otimes \rho)U^{(2)\dagger}], \tag{142}$$

where,

$$U^{(1)} = \frac{1}{\sqrt{2}} \begin{pmatrix} ae^{-i\phi_1} & 0 & 0 & -c \\ 0 & d & -be^{-i\phi_2} & 0 \\ 0 & be^{i\phi_2} & d & 0 \\ c & 0 & 0 & ae^{i\phi_1} \end{pmatrix}, \tag{143}$$

and

$$U^{(2)} = \frac{1}{\sqrt{2}} \begin{pmatrix} ae^{i\phi_1} & 0 & 0 & -c \\ 0 & d & -be^{i\phi_2} & 0 \\ 0 & be^{-i\phi_2} & d & 0 \\ c & 0 & 0 & ae^{-i\phi_1} \end{pmatrix}. \tag{144}$$

Now from equation (135), we implement a convex sum of the channels in equations (141) and (142), which leads to,

$$\begin{aligned}T_t^{(\theta_k)}(\rho) &= \frac{1}{2}T_{(t,1)}^{(\theta_k)}(\rho) + \frac{1}{2}T_{(t,2)}^{(\theta_k)}(\rho),\\ &= \frac{1}{2}\mathrm{tr}_E[U^{(1)}(|0\rangle\langle 0|\otimes\rho)U^{(1)\dagger}] + \frac{1}{2}\mathrm{tr}_E[U^{(2)}(|0\rangle\langle 0|\otimes\rho)U^{(2)\dagger}].\end{aligned} \tag{145}$$

Now to simulate the constituent channel $T_t^{(k)}$ we need to unitary conjugate the channel $T_t^{(\theta_k)}$ bu U_k, so from equation (145) we have that,

$$\begin{aligned}T_t^{(k)}(\rho) &= U_k^\dagger T_t^{(\theta_k)}(U_k\rho U_k^\dagger)U_k,\\ &= U_k^\dagger(\frac{1}{2}T_{(t,1)}^{(\theta_k)}(\rho) + \frac{1}{2}T_{(t,2)}^{(\theta_k)}(\rho))U_k,\\ &= U_k^\dagger(\frac{1}{2}\mathrm{tr}_E[U^{(1)}(|0\rangle\langle 0|\otimes U_k\rho U_k^\dagger)U^{(1)\dagger}] + \frac{1}{2}\mathrm{tr}_E[U^{(2)}(|0\rangle\langle 0|\otimes U_k\rho U_k^\dagger)U^{(2)\dagger}])U_k,\\ &= \frac{1}{2}U_k^\dagger\mathrm{tr}_E[U^{(1)}(|0\rangle\langle 0|\otimes U_k\rho U_k^\dagger)U^{(1)\dagger}]U_k + \frac{1}{2}U_k^\dagger\mathrm{tr}_E[U^{(2)}(|0\rangle\langle 0|\otimes U_k\rho U_k^\dagger)U^{(2)\dagger}])U_k.\end{aligned} \tag{146}$$

Equation (146) tells us that we need a quantum circuit that can implement a convex sum of the two unitaries $U^{(1)}$ and $U^{(2)}$ and the unitary conjugation by U_k, for this we make use of quantum forking [30]. The theorem below shall outline how quantum forking is used to implement the convex sum of quasi-extreme channels.

Theorem 5. Given a convex sum of two quantum channels as in equation (135) and their Stinespring representations, i.e. equations (141) and (142). The circuit shown in Figure 1 implements the convex sum of quasi extreme channels as well as unitary conjugation in equation (146).

Proof. We can show this directly by calculating the output state of the circuit in Figure 1 and tracing out everything else except the system qubit. We note that the input states of the qubits are as follows: $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is called the ancilla qubit, $|\psi\rangle$ is the state of the system, $|0\rangle$ is the state of the environment and $|\phi_1\rangle$, $|\phi_2\rangle$ are arbitrary states that are needed to perform the quantum forking algorithm. The initial state of all the qubits is:

$$|\Phi_0\rangle = |+\rangle|0\rangle|\psi\rangle|\phi_1\rangle|\phi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle|\psi\rangle|\phi_1\rangle|\phi_2\rangle + |1\rangle|0\rangle|\psi\rangle|\phi_1\rangle|\phi_2\rangle). \tag{147}$$

We now calculate $|\Phi_1\rangle$, we do this by applying the operator U_k to the system qubit $|\psi\rangle$. Now by denoting $|\psi'\rangle = U_k|\psi\rangle$ we have,

$$|\Phi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle|\psi'\rangle|\phi_1\rangle|\phi_2\rangle + |1\rangle|0\rangle|\psi'\rangle|\phi_1\rangle|\phi_2\rangle). \tag{148}$$

Now a control swap gates swap the states $|\psi'\rangle$ and $|\phi_1\rangle$ when the ancilla is in the state $|1\rangle$ and similarly the states $|0\rangle$ and $|\phi_2\rangle$ are swapped when the ancilla is in the state $|1\rangle$, this leads to the state,

$$|\Phi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle|\psi'\rangle|\phi_1\rangle|\phi_2\rangle + |1\rangle|\phi_1\rangle|\phi_2\rangle|0\rangle|\psi'\rangle). \tag{149}$$

We now apply the unitary operations $U^{(1)}$ and $U^{(2)}$ on the respective qubits, this yields,

$$|\Phi_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle U^{(1)}(|0\rangle|\psi'\rangle)U^{(2)}(|\phi_1\rangle|\phi_2\rangle) + |1\rangle U^{(1)}(|\phi_1\rangle|\phi_2\rangle)U^{(2)}(|0\rangle|\psi'\rangle)). \tag{150}$$

Now we apply the controlled swaps to swap the second and third qubits with the fourth and fifth qubits when the ancilla qubit is in the state $|1\rangle$, so we have:

$$|\Phi_4\rangle = \frac{1}{\sqrt{2}}(|0\rangle U^{(1)}(|0\rangle|\psi'\rangle)U^{(2)}(|\phi_1\rangle|\phi_2\rangle) + |1\rangle U^{(2)}(|0\rangle|\psi'\rangle)U^{(1)}(|\phi_1\rangle|\phi_2\rangle)). \tag{151}$$

Lastly we apply the operation $U_k^\dagger$ to the system qubit which yields,

$$|\Phi_5\rangle = \frac{1}{\sqrt{2}}(|0\rangle(\mathbb{1}\otimes U_k^\dagger)U^{(1)}(|0\rangle|\psi'\rangle)U^{(2)}(|\phi_1\rangle|\phi_2\rangle) + |1\rangle(\mathbb{1}\otimes U_k^\dagger)U^{(2)}(|0\rangle|\psi'\rangle)U^{(1)}(|\phi_1\rangle|\phi_2\rangle)). \tag{152}$$

We now construct the density matrix that corresponds to the state of all the qubits after performing the circuit, denoted by $\rho_{\text{tot}} = |\Phi_5\rangle\langle\Phi_5|$, so we have,

$$\begin{aligned}\rho_{\text{tot}} = \frac{1}{2}\big(\, &|0\rangle\langle 0| \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger} \\ &+ |0\rangle\langle 1| \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger} \\ &+ |1\rangle\langle 0| \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger} \\ &+ |1\rangle\langle 1| \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger}\big).\end{aligned} \tag{153}$$

From the density matrix of the total system of qubits we can now trace out the environment E, the ancilla a and the qubits in an arbitrary state, used for the forking, ϕ_1 and ϕ_2. Doing this partial trace leaves us with the state of the system qubit after performing the circuit. We can do this in a few steps: We can start by tracing out the ancilla, we do this by taking the trace over the first tensor factor from the left in ρ_{tot} while using the fact that $\text{tr}[|a\rangle\langle b|] = \langle a|b\rangle$ this leads to,

$$\begin{aligned}\text{tr}_a[\rho_{\text{tot}}] = \frac{1}{2}\big(\, &\langle 0|0\rangle \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger} \\ &+ \langle 0|1\rangle \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger} \\ &+ \langle 1|0\rangle \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger} \\ &+ \langle 1|1\rangle \otimes (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger}\big).\end{aligned} \tag{154}$$

By using the orthonormality of the states $\{|0\rangle\,, |1\rangle\}$, we can simplify equation (154),

$$\begin{aligned}\text{tr}_a[\rho_{\text{tot}}] = \frac{1}{2}(&(\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger} \\ &+ (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k) \otimes U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger}).\end{aligned} \tag{155}$$

Now let us trace out the arbitrary qubits ϕ_1 and ϕ_2,

$$\begin{aligned}\text{tr}_{\phi_1+\phi_2+a}[\rho_{\text{tot}}] &= \text{tr}_{\phi_1+\phi_2}[\text{tr}_a[\rho_{\text{tot}}]], \\ &= \frac{1}{2}((\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \otimes \text{tr}[U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger}] \\ &+ (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k) \otimes \text{tr}[U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger}]).\end{aligned} \tag{156}$$

Using the fact that,

$$\text{tr}[U^{(2)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(2)\dagger}] = 1 = \text{tr}[U^{(1)}(|\phi_1\rangle\,|\phi_2\rangle)(\langle\phi_1|\,\langle\phi_2|)U^{(1)\dagger}], \tag{157}$$

we can simplify equation (156),

$$\begin{aligned}\text{tr}_{\phi_1+\phi_2+a}[\rho_{\text{tot}}] = \frac{1}{2}(&(\mathbb{1} \otimes U_k^\dagger)U^{(1)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(1)\dagger}(\mathbb{1} \otimes U_k) \\ &+ (\mathbb{1} \otimes U_k^\dagger)U^{(2)}(|0\rangle\,|\psi'\rangle)(\langle 0|\,\langle\psi'|)U^{(2)\dagger}(\mathbb{1} \otimes U_k)).\end{aligned} \tag{158}$$

We can now define the density matrix $\rho_{SE} = \text{tr}_{\phi_1+\phi_2+a}[\rho_{\text{tot}}]$, which denotes the state of the system qubit and environment. We can now obtain the state of the system ρ_S, after performing the circuit, by tracing out the environment. From equation (158) we have,

$$\begin{aligned}\rho_S &= \mathrm{tr}_E[\rho_{SE}],\\ &= \frac{1}{2}(\mathrm{tr}_E[(\mathbb{1}\otimes U_k^\dagger)U^{(1)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(1)\dagger}(\mathbb{1}\otimes U_k)]\\ &\quad+\mathrm{tr}_E[(\mathbb{1}\otimes U_k^\dagger)U^{(2)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(2)\dagger}(\mathbb{1}\otimes U_k)).\end{aligned}\tag{159}$$

Now by using the states $\{|0\rangle, |1\rangle\}$ as an orthonormal basis for the environment we can write equation (159) as,

$$\begin{aligned}\rho_S &= \frac{1}{2}\Big(\sum_{j=0}^{1}(\langle j|\otimes\mathbb{1})(\mathbb{1}\otimes U_k^\dagger)U^{(1)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(1)\dagger}(\mathbb{1}\otimes U_k)(|j\rangle\otimes\mathbb{1})\\ &\quad+\sum_{j'=0}^{1}(\left\langle j'\right|\otimes\mathbb{1})(\mathbb{1}\otimes U_k^\dagger)U^{(2)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(2)\dagger}(\mathbb{1}\otimes U_k)(\left|j'\right\rangle\otimes\mathbb{1})\Big).\end{aligned}\tag{160}$$

By using the fact that $(\langle j|\otimes\mathbb{1})(\mathbb{1}\otimes U_k^\dagger) = (\mathbb{1}\otimes U_k^\dagger)(\langle j|\otimes\mathbb{1})$ and $(\mathbb{1}\otimes U_k)(|j\rangle\otimes\mathbb{1}) = (|j\rangle\otimes\mathbb{1})(\mathbb{1}\otimes U_k)$, we can write ρ_S as,

$$\begin{aligned}\rho_S &= \frac{1}{2}\Big(\sum_{j=0}^{1}(\mathbb{1}\otimes U_k^\dagger)(\langle j|\otimes\mathbb{1})U^{(1)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(1)\dagger}(|j\rangle\otimes\mathbb{1})(U_k\otimes\mathbb{1})\\ &\quad+\sum_{j'=0}^{1}(\mathbb{1}\otimes U_k^\dagger)(\left\langle j'\right|\otimes\mathbb{1})U^{(2)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(2)\dagger}(\left|j'\right\rangle\otimes\mathbb{1})(U_k\otimes\mathbb{1})\Big),\\ &= \frac{1}{2}(U_k^\dagger\mathrm{tr}_E[U^{(1)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(1)\dagger}]U_k + U_k^\dagger\mathrm{tr}_E[U^{(2)}(|0\rangle\left|\psi'\right\rangle)(\langle 0|\left\langle\psi'\right|)U^{(2)\dagger}]U_k).\end{aligned}\tag{161}$$

By observing that $(|0\rangle|\psi'\rangle)(\langle 0|\langle\psi'|) = |0\rangle\langle 0|\otimes|\psi'\rangle\langle\psi'|$, and from the definition of $|\psi'\rangle$ we have that $|0\rangle\langle 0|\otimes|\psi'\rangle\langle\psi'| = |0\rangle\langle 0|\otimes U_k|\psi\rangle\langle\psi|U_k^\dagger$. If we define $\rho = |\psi\rangle\langle\psi|$, we can write $(|0\rangle|\psi'\rangle)(\langle 0|\langle\psi'|) = |0\rangle\langle 0|\otimes U_k\rho U_k^\dagger$, this allows us to write equation (161) as,

$$\begin{aligned}\rho_S &= \frac{1}{2}(U_k^\dagger\mathrm{tr}_E[U^{(1)}(|0\rangle\langle 0|\otimes U_k\rho U_k^\dagger)U^{(1)\dagger}]U_k + U_k^\dagger\mathrm{tr}_E[U^{(2)}(|0\rangle\langle 0|\otimes U_k\rho U_k^\dagger)U^{(2)\dagger}]U_k),\\ &= \frac{1}{2}(U_k^\dagger T_{(t,1)}^{(\theta_k)}(U_k\rho U_k^\dagger)U_k + U_k^\dagger T_{(t,2)}^{(\theta_k)}(U_k\rho U_k^\dagger)U_k),\\ &= U_k^\dagger(\frac{1}{2}T_{(t,1)}^{(\theta_k)}(U_k\rho U_k^\dagger) + \frac{1}{2}T_{(t,2)}^{(\theta_k)}(U_k\rho U_k^\dagger))U_k,\\ &= U_k^\dagger T_t^{(\theta_k)}(U_k\rho U_k^\dagger)U_k,\\ &= T_t^{(k)}(\rho).\end{aligned}\tag{162}$$

We get the second equality in equation (162) from the Stinespring representation of channels $T_{(t,1)}^{(\theta_k)}$ and $T_{(t,2)}^{(\theta_k)}$ and the last equality follows from equation (146). Hence we have shown that the circuit in Figure 1 implements the convex sum of quasi extreme channels and unitary conjugation in equation (146), which completes the proof. □

Now that we have a circuit that can implement the constituent channel $T_t^{(k)}$, i.e. Figure 1. We need to decompose the unitary operations $U^{(1)}$ and $U^{(2)}$ into single qubit gates and controlled not (CNOT) gates. Before we can do this we observe that, $\frac{1}{2}(a^2+c^2) = 1$ and $\frac{1}{2}(b^2+d^2) = 1$ so we can define,

$$\begin{aligned}\cos(\alpha) &= \frac{1}{\sqrt{2}}a, & \sin(\alpha) &= \frac{1}{\sqrt{2}}c,\\ \cos(\beta) &= \frac{1}{\sqrt{2}}b, & \sin(\beta) &= \frac{1}{\sqrt{2}}d.\end{aligned}\tag{163}$$

This allows us to rewrite the unitary operations $U^{(1)}$ and $U^{(2)}$ as,

$$U^{(1)} = \begin{pmatrix} e^{-i\phi_1}\cos(\alpha) & 0 & 0 & -\sin(\alpha)\\ 0 & \sin(\beta) & -e^{-i\phi_2}\cos(\beta) & 0\\ 0 & e^{i\phi_2}\cos(\beta) & \sin(\beta) & 0\\ \sin(\alpha) & 0 & 0 & e^{i\phi_1}\cos(\alpha)\end{pmatrix},\tag{164}$$

and

$$U^{(2)} = \begin{pmatrix} e^{i\phi_1}\cos(\alpha) & 0 & 0 & -\sin(\alpha) \\ 0 & \sin(\beta) & -e^{i\phi_2}\cos(\beta) & 0 \\ 0 & e^{-i\phi_2}\cos(\beta) & \sin(\beta) & 0 \\ \sin(\alpha) & 0 & 0 & e^{-i\phi_1}\cos(\alpha) \end{pmatrix}. \tag{165}$$

Now let us start by decomposing the unitary matrix $U^{(1)}$, we observe that,

$$U^{(1)} = U_A^{(1)} U_B^{(1)}, \tag{166}$$

where

$$U_A^{(1)} = \begin{pmatrix} e^{-i\phi_1}\cos(\alpha) & 0 & 0 & -\sin(\alpha) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\alpha) & 0 & 0 & e^{i\phi_1}\cos(\alpha) \end{pmatrix}, \tag{167}$$

and

$$U_B^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sin(\beta) & -e^{-i\phi_2}\cos(\beta) & 0 \\ 0 & e^{i\phi_2}\cos(\beta) & \sin(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{168}$$

Now using the methods outlined in Appendix A we can find quantum circuits for the operations $U_A^{(1)}$ and $U_B^{(2)}$. First we define the following 2×2 matrices, which will assist us in the decomposition,

$$\tilde{U}_A^{(1)} = \begin{pmatrix} e^{-i\phi_1}\cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & e^{i\phi_1}\cos(\alpha) \end{pmatrix}, \qquad \tilde{U}_B^{(1)} = \begin{pmatrix} \sin(\beta) & -e^{-i\phi_2}\cos(\beta) \\ e^{i\phi_2}\cos(\beta) & \sin(\beta) \end{pmatrix}. \tag{169}$$

Now for the unitary matrix $U_A^{(1)}$ we can write,

$$\begin{aligned} U_A^{(1)} &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-i\phi_1}\cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\alpha) & 0 & e^{i\phi_1}\cos(\alpha) \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= (|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1})(\mathbb{1} \otimes |0\rangle\langle 0| + \tilde{U}_A^{(1)} \otimes |1\rangle\langle 1|)(|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1}). \end{aligned} \tag{170}$$

From equation (170) we can find the circuit that corresponds to $U_A^{(1)}$, and it is shown in Figure 2(a). Now using the same method above we can find a circuit that implements the operation $U_B^{(1)}$,

$$\begin{aligned} U_B^{(1)} &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sin(\beta) & 0 & -e^{-i\phi_2}\cos(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ e^{i\phi_2}\cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\ &= (|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1})(\tilde{U}_B^{(1)} \otimes |0\rangle\langle 0| + \mathbb{1} \otimes |1\rangle\langle 1|)(|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1}). \end{aligned} \tag{171}$$

The circuit that implements $U_B^{(1)}$ can be seen in Figure 2(b). Now we can put these circuits together to get the circuit that implements the unitary operation $U^{(1)}$, this can be seen in Figure 3. Since $(|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1})^2 = \mathbb{1}$, we drop the two open CNOT gates in between the circuits for $U_B^{(1)}$ and $U_A^{(1)}$, and get the simplified circuit for $U^{(1)}$ in Figure 3.

We now want to decompose the controlled $\tilde{U}_A^{(1)}$ operation into single qubit gates and CNOT gates. We use the results

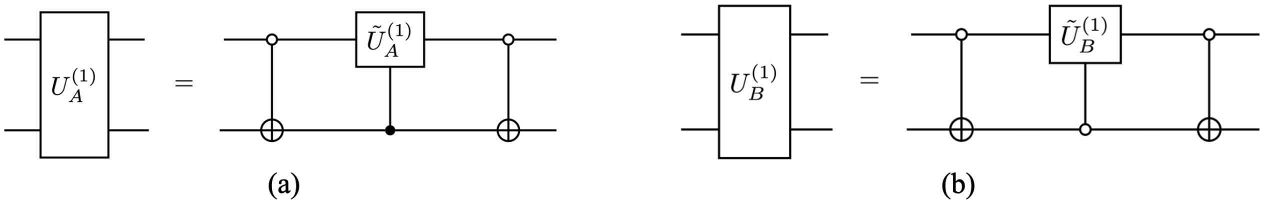


Figure 2: Circuit that implements the operators $U_A^{(1)}$ and $U_A^{(2)}$.

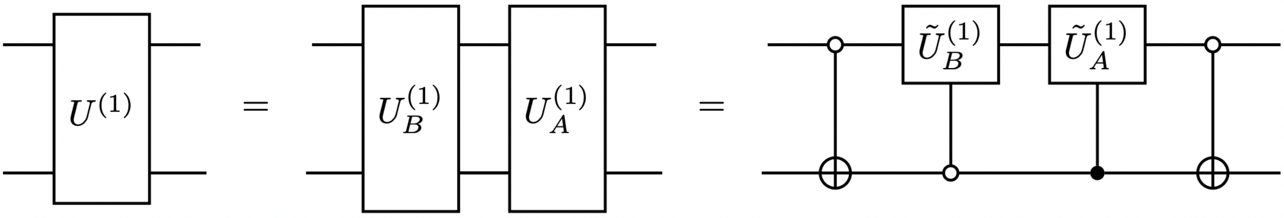


Figure 3: Circuit that implements the opertation $U^{(1)}$.

from Appendix A to decompose this operation as follows. Since $\det(\tilde{U}_A^{(1)}) = 1$, this implies that $\tilde{U}_A^{(1)} \in \mathrm{SU}(2)$, then by Lemma 7 we have,

$$\tilde{U}_A^{(1)} = \begin{pmatrix} e^{\frac{-i\phi_1}{2}} & 0 \\ 0 & e^{\frac{i\phi_1}{2}} \end{pmatrix} \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} e^{\frac{-i\phi_1}{2}} & 0 \\ 0 & e^{\frac{i\phi_1}{2}} \end{pmatrix},$$

$$= R_z(\phi_1) R_y(2\alpha) R_z(\phi_1). \tag{172}$$

Now from Lemma 9 we can decompose the controlled $\tilde{U}_A^{(1)}$ operation into single qubit and CNOT gates as seen in Figure 4. We can now decompose the open controlled $\tilde{U}_B^{(1)}$ operation into single qubit gates and CNOT gates. This is done in the same way as for the controlled $\tilde{U}_A^{(1)}$. Since $\det(\tilde{U}_B^{(1)}) = 1$, this implies that $\tilde{U}_B^{(1)} \in \mathrm{SU}(2)$, then by Lemma 7 we have,

$$\tilde{U}_B^{(1)} = \begin{pmatrix} e^{\frac{-i\phi_2}{2}} & 0 \\ 0 & e^{\frac{i\phi_2}{2}} \end{pmatrix} \begin{pmatrix} \sin(\beta) & -\cos(\beta) \\ \cos(\beta) & \sin(\beta) \end{pmatrix} \begin{pmatrix} e^{\frac{i\phi_2}{2}} & 0 \\ 0 & e^{\frac{-i\phi_2}{2}} \end{pmatrix},$$

$$= \begin{pmatrix} e^{\frac{-i\phi_2}{2}} & 0 \\ 0 & e^{\frac{i\phi_2}{2}} \end{pmatrix} \begin{pmatrix} \cos\left(\frac{\pi}{2} - \beta\right) & -\sin\left(\frac{\pi}{2} - \beta\right) \\ \sin\left(\frac{\pi}{2} - \beta\right) & \cos\left(\frac{\pi}{2} - \beta\right) \end{pmatrix} \begin{pmatrix} e^{\frac{i\phi_2}{2}} & 0 \\ 0 & e^{\frac{-i\phi_2}{2}} \end{pmatrix},$$

$$= R_z(\phi_2) R_y(2(\frac{\pi}{2} - \beta)) R_z(-\phi_2). \tag{173}$$

Now from Lemma 9 we have the circuit for the open controlled $\tilde{U}_B^{(1)}$ operation in Figure 5. Let us now decompose the unitary operation $U^{(2)}$. From equation (165) we observe that,

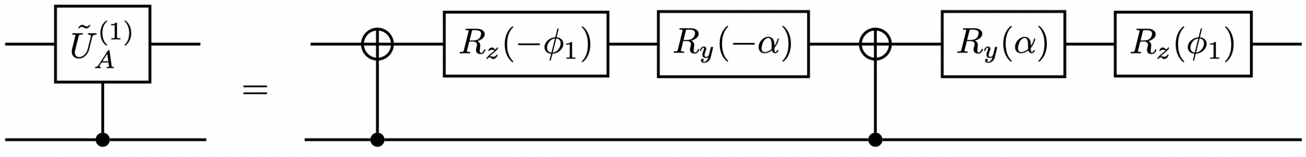


Figure 4: Circuit that implements the controlled $\tilde{U}_A^{(1)}$operation.

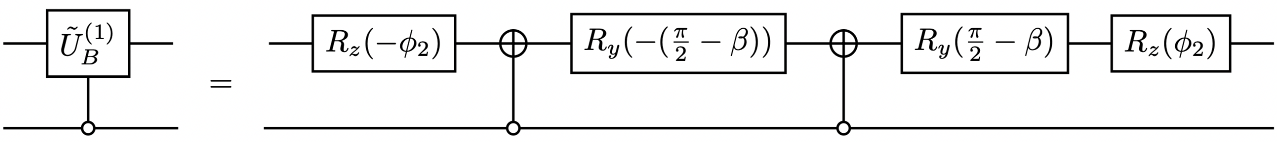
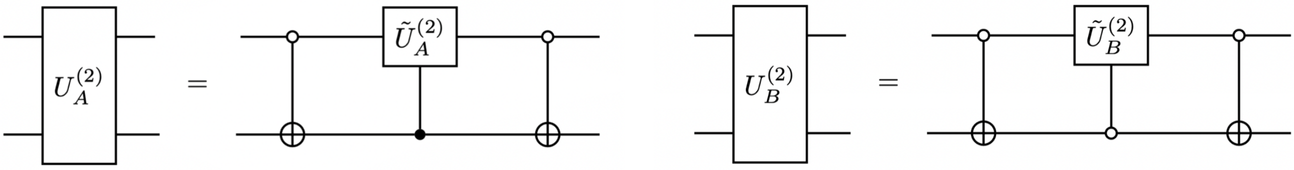


Figure 5: *Circuit that implements the open controlled $\tilde{U}_B^{(1)}$operation.*



(a)

(b)

Figure 6: *Circuit that implements the $U_A^{(2)}$ and $U_B^{(2)}$ unitary operations.*

$$U^{(2)} = U_A^{(2)} U_B^{(2)}, \tag{174}$$

where

$$U_A^{(2)} = \begin{pmatrix} e^{i\phi_1}\cos(\alpha) & 0 & 0 & -\sin(\alpha) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\alpha) & 0 & 0 & e^{-i\phi_1}\cos(\alpha) \end{pmatrix}, \tag{175}$$

and

$$U_B^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sin(\beta) & -e^{i\phi_2}\cos(\beta) & 0 \\ 0 & e^{-i\phi_2}\cos(\beta) & \sin(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{176}$$

We can now define the following matrices,

$$\tilde{U}_A^{(2)} = \begin{pmatrix} e^{i\phi_1}\cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & e^{-i\phi_1}\cos(\alpha) \end{pmatrix}, \qquad \tilde{U}_B^{(2)} = \begin{pmatrix} \sin(\beta) & -e^{i\phi_2}\cos(\beta) \\ e^{-i\phi_2}\cos(\beta) & \sin(\beta) \end{pmatrix}. \tag{177}$$

From the matrices in equation (177), we can now decompose $U_A^{(2)}$ as follows,

$$U_A^{(2)} = (|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1})(\mathbb{1} \otimes |0\rangle\langle 0| + \tilde{U}_A^{(2)} \otimes |1\rangle\langle 1|)(|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1}). \tag{178}$$

Figure 6(a) shows the circuit that implements $U_A^{(2)}$. Now let us decompose the unitary operation $U_{B^{(2)}}$,

$$U_B^{(2)} = (|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1})(\mathbb{1} \otimes |1\rangle\langle 1| + \tilde{U}_B^{(2)} \otimes |0\rangle\langle 0|)(|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes \mathbb{1}). \tag{179}$$

Figure 6(b) shows the circuit that implements $U_B^{(2)}$. We can put this together and make the same simplification to the circuit as we did for the unitary operation $U^{(1)}$, where the two open controlled CNOT gates are replaced by the identity operation. Now Figure 7 shows the circuit that implements the unitary operation $U^{(2)}$.

We can now decompose the controlled $\tilde{U}_A^{(2)}$ operation and the open controlled $\tilde{U}_B^{(2)}$ operation using the methods from Appendix A in a similar way to the decomposition used above for $U^{(1)}$. We obtain the circuit for the controlled $\tilde{U}_A^{(2)}$ operation, which is shown in Figure 8 and the circuit that implements the open control $\tilde{U}_B^{(2)}$ operation is shown in Figure 9. This completes the decomposition of the operations $U^{(1)}$ and $U^{(2)}$ into single qubit and CNOT gates. We can now use these circuits in the circuit shown in Figure 1 to simulate the constituent channel $T_t^{(k)}$.

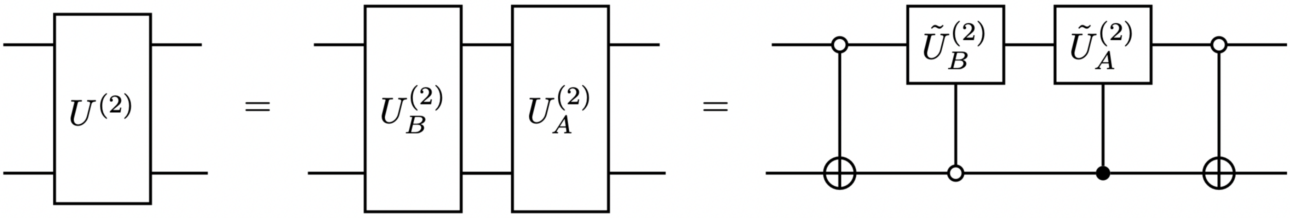


Figure 7: *Circuit that implements* $U^{(2)}$.

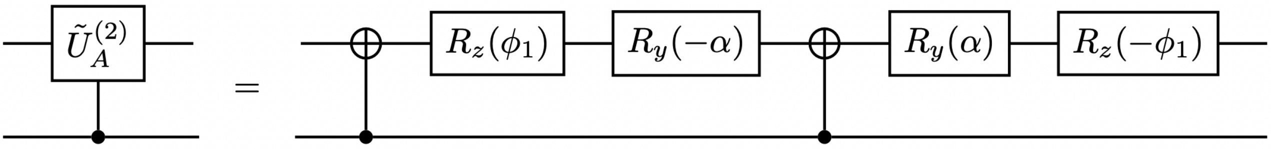


Figure 8: *Circuit that implements controled* $\tilde{U}_A^{(2)}$.

6 Summary

We now combine the results in the previous sections and state the full algorithm for the simulation of a single qubit open quantum system, which requires $O\left(\frac{(\Lambda t)^{3/2}}{\epsilon^{1/2}}\right)$ gates as a solution to the problem defined in Section 2:
(1) Given the generator $\mathcal{L}$ as defined in Section 3, the spectral decomposition of the GKS matrix A yields the decomposition of $\mathcal{L}$,

$$\mathcal{L} = \sum_{k=0}^{3} \lambda_k \mathcal{L}_k, \tag{180}$$

with $C_k \in \mathrm{SO}(3)$ and $\theta_K \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ specifying the decomposition so that,

$$A_k = C_k^T A(\theta_k) C_k. \tag{181}$$

for all k=1,2,3.
(2) Next we choose a precision $\epsilon \geq 0$ so that we can calculate the number of implementations N of $S_2(\tau)$, with

$$N \geq \frac{(4t\Lambda)^{3/2}}{(3\epsilon)^{1/2}}. \tag{182}$$

(3) Implement $S_2(\tau)$ N times using,

$$T_{t'}^{(k)}(\rho) = (\mathcal{U}_k^\dagger T_{t'}^{(\theta_k)} \mathcal{U}_k)(\rho) = U_k^\dagger [T_{t'}^{(\theta_k)}(U_k \rho U_k^\dagger)] U_k. \tag{183}$$

where λ_k, Λ and τ have been incorporated into t'. U_k is obtained from C_k as per Section 3, and $T_{t'}^{(\theta_k)}$ is implemented via quantum forking as per Section 5. One should note that to implement products of $T_{t'}^{(k)}$ we use the same circuit as in Figure 1 but for each implementation of $T_{t'}^{(k)}$ we use a new qubit for the environment and arbitrary states $|\phi_1\rangle$ and $|\phi_2\rangle$.

Now that we have summarised the full algorithm, one should be aware of the implication this has for simulating Markovian open quantum systems. This algorithm provides a framework for digitally simulating any single qubit Markovian open quantum system, up to a high enough precision allowing us to study the dynamics of systems for which their master equation cannot be analytically integrated.

This tutorial set out to pedagogically introduce the tools and techniques needed to construct algorithms for the simulation of Markovian open quantum systems by looking at an algorithm for the simplest case of the simulation of a single qubit open quantum system. Of course, the reader is urged to use the ideas and techniques taught here to

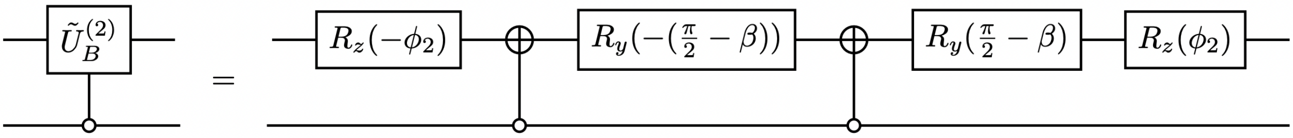


Figure 9: *Circuit that implements open control* $\tilde{U}_B^{(2)}$.

study the more general cases of simulating Markovian open quantum systems of arbitrary and finite dimension [21] as well as the case of simulating non Markovian open quantum systems [44].

In light of the results presented in this tutorial one should also mention the possible avenues of extension. The first major issue that was recognised in developing algorithms to simulate open quantum systems and an issue that plagues this algorithm as well, is the fact that we can only approximate our channel T_t up to second order with SLT product formulas so as not to violate complete positivity. This constraint arises from the fact that you cannot use higher order SLT product formulas for coefficients that are all positive, as proven by Suzuki in [13]. Given this information one could look into methods for approximating the channel T_t to higher orders that do not violate complete positivity. Another area of extension would be with regards to the simulation of Markovian open quantum systems with arbitrary dimension [21], where on could possibly tackle the open problem of convexly decomposing the universal semigroup of arbitrary dimensional channels into extreme channels [45].

7 Appendix A: Quantum Circuit Decomposition

We need some useful results from [46], for the decomposition of the unitary matrices corresponding to the quasi-extreme channels. We also make use of the quantum computing notation to denote the Pauli matrices as:

$$\sigma_1 = X, \qquad \sigma_2 = Y, \qquad \sigma_3 = Z. \tag{184}$$

Definition 1. We define the following unitary matrices,

$$R_y(\theta) \equiv \exp\left(-i\frac{\theta}{2}Y\right) = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)Y = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \tag{185}$$

$$R_z(\theta) \equiv \exp\left(-i\frac{\theta}{2}Z\right) = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}, \tag{186}$$

$$P(\delta) \equiv \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix}. \tag{187}$$

We shall make use of the following lemmas to decompose any 2×2 unitary matrix.

Lemma 7. Suppose U is a 2×2 unitary matrix. Then there exists $\alpha, \beta, \delta, \theta \in \mathbb{R}$ such that,

$$U = P(\delta)R_z(\alpha)R_y(\theta)R_z(\beta). \tag{188}$$

Proof. Since U is unitary its rows and columns are orthonormal, from which it follows that there exists $\alpha, \beta, \delta, \theta \in \mathbb{R}$ such that,

$$U = \begin{pmatrix} e^{i(\delta-\frac{\alpha}{2}-\frac{\beta}{2})}\cos\left(\frac{\theta}{2}\right) & -e^{i(\delta-\frac{\alpha}{2}+\frac{\beta}{2})}\sin\left(\frac{\theta}{2}\right) \\ e^{i(\delta+\frac{\alpha}{2}-\frac{\beta}{2})}\sin\left(\frac{\theta}{2}\right) & e^{i(\delta+\frac{\alpha}{2}+\frac{\beta}{2})}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{189}$$

Now equation (188) follows directly from the definitions of R_y, R_z, P and matrix multiplication. □

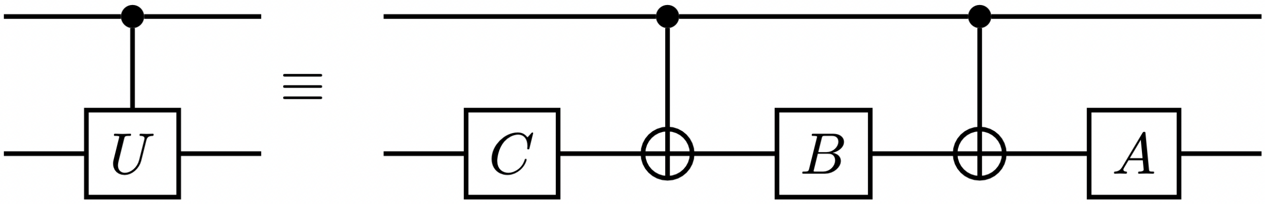


Figure 10: *Quantum circuit implementing the controlled unitary gate $c_1(U)$.*

Lemma 8. Suppose U is a 2×2 unitary matrix. Then there exists 2×2 matrices A, B, C such that $ABC = \mathbb{1}$ and $U = P(\delta)AXBXC$.

Proof. First we observe that, $XYX = -Y$ which implies that,

$$XR_y(\theta)X = X(\cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)Y)X = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)XYX = \cos\left(\frac{\theta}{2}\right)\mathbb{1} + i\sin\left(\frac{\theta}{2}\right)Y = R_y(-\theta). \tag{190}$$

Now if we set $A \equiv R_z(\alpha)R_y(\frac{\theta}{2})$, $B \equiv R_y(-\frac{\theta}{2})R_z(\frac{-(\beta+\alpha)}{2})$ and $C \equiv R_z(\frac{(\beta-\alpha)}{2})$, we have,

$$\begin{aligned} ABC &= R_z(\alpha)R_y\left(\frac{\theta}{2}\right)R_y\left(-\frac{\theta}{2}\right)R_z\left(-\frac{\beta+\alpha}{2}\right)R_z\left(\frac{\beta-\alpha}{2}\right) = R_z(\alpha)R_y\left(\frac{\theta}{2}-\frac{\theta}{2}\right)R_z\left(\frac{-\beta-\alpha}{2}+\frac{\beta-\alpha}{2}\right) \\ &= R_z(\alpha)R_y(0)R_z(-\alpha) = R_z(\alpha)R_z(-\alpha) = \mathbb{1} \end{aligned} \tag{191}$$

Since $X^2 = \mathbb{1}$ and from equation (190) we see that,

$$XBX = XR_y\left(-\frac{\theta}{2}\right)R_z\left(-\frac{(\beta+\alpha)}{2}\right)X = XR_y\left(-\frac{\theta}{2}\right)XXR_z\left(-\frac{(\beta+\alpha)}{2}\right)X = R_y\left(\frac{\theta}{2}\right)R_z\left(\frac{(\beta+\alpha)}{2}\right). \tag{192}$$

Thus,

$$\begin{aligned} AXBXC &= R_z(\alpha)R_y\left(\frac{\theta}{2}\right)XR_y\left(-\frac{\theta}{2}\right)R_z\left(-\frac{(\beta+\alpha)}{2}\right)XR_z\left(\frac{\beta-\alpha}{2}\right) \\ &= R_z(\alpha)R_y\left(\frac{\theta}{2}\right)R_y\left(\frac{\theta}{2}\right)R_z\left(\frac{(\beta+\alpha)}{2}\right)R_z\left(\frac{(\beta-\alpha)}{2}\right) = R_z(\alpha)R_y(\theta)R_z(\beta). \end{aligned} \tag{193}$$

Now by Lemma 7 we have that,

$$U = P(\delta)AXBXC, \tag{194}$$

as required. □

Now we require a way to decompose a controlled unitary operation. We have the following lemma which tells us how to do this.

Lemma 9. For any 2×2 unitary matrix U, a controlled unitary gate $c_1(U)$ can be simulated with the quantum circuit in Figure 10 with $A, B, C \in \mathrm{SU}(2)$, if and only if $U \in \mathrm{SU}(2)$.

Proof. For the "if" part let A, B and C be defined as in Lemma 8. If the value of the control qubit is 0 then $ABC = \mathbb{1}$ is applied to the target qubit. If the value of the control qubit is 1 then $AXBXC = U$ is applied to the target qubit.

For the "only if" part note that $ABC = \mathbb{1}$ must hold if the output of the circuit is correct when the control qubit is 0. Also if the circuit simulates a $c_1(U)$ gate then $AXBXC = U$. Therefore since $\det(AXBXC) = 1$, U must also be special unitary and hence $U \in \mathrm{SU}(2)$. □

8 Appendix B: Stinespring Representation of the Quantum Channel

When we constructed the unitary operators that implement the quasi extreme channels we make use of the Stinespring representation of the quantum channel T_t [39, 40, 47], this appendix shall provide a brief outline of the Stinespring representation as well as an example of its use.

Theorem 6. Let $T : \mathcal{B}(\mathcal{H}_s) \to \mathcal{B}(\mathcal{H}_s)$ be a CPTP map where $\mathcal{H}_s$ is the Hilbert space of the system. Then there exists a Hilbert space $\mathcal{H}_E$ called the environment whose and a unitary operator U acting on the joint space $\mathcal{H}_s \otimes \mathcal{H}_E$ and a quantum state $|0\rangle\langle 0| \in \mathcal{B}(\mathcal{H}_E)$ such that

$$T(\rho) = \mathrm{tr}_E[U(|0\rangle\langle 0| \otimes \rho)U^\dagger], \quad \forall \rho \in \mathcal{B}(\mathcal{H}_s) \tag{195}$$

where $\dim(\mathcal{H}_E) \geq \dim(\mathcal{H}_s)^2$ and the representation is unique up to a unitary equivalence.

Given the Kraus representation of the channel T as in equation (3) one can embed the Kraus in the first block-column of U that acts on the system as well as auxilary qubits that emulate an environment and we populate the rest of block columns so that U is unitary. Now we have,

$$U = \begin{pmatrix} K_1 & \dots & \dots \\ K_2 & \dots & \dots \\ \vdots & \ddots & \vdots \\ K_r & \dots & \dots \end{pmatrix} \tag{196}$$

where $r \leq \dim(\mathcal{H}_s)^2$. This is done so that when we apply the unitary U to an initial state $|0\rangle \otimes |\psi\rangle$ we get,

$$U(|0\rangle \otimes |\psi\rangle) = \sum_{j=1}^{r} |j\rangle \otimes K_j |\psi\rangle, \tag{197}$$

where $\{|j\rangle\}_{j=1}^{r}$ is an orthonormal basis for the environment. Now by taking the outer product of this state and tracing out the environment we get the action of the channel T on the initial state $|\psi\rangle\langle\psi|$, i.e. $T(|\psi\rangle\langle\psi|)$, we can demonstrate this as follows,

$$\begin{aligned} \mathrm{tr}_E\left\{\sum_{j,j'=1}^{r} \big|j\rangle\langle j'\big| \otimes K_j |\psi\rangle\langle\psi| K_{j'}^\dagger\right\} &= \sum_{j,j'=1}^{r} \mathrm{tr}_E\left\{\big|j\rangle\langle j'\big|\right\} K_j |\psi\rangle\langle\psi| K_{j'}^\dagger = \sum_{j,j'=1}^{r} \langle j\big|j'\rangle K_j |\psi\rangle\langle\psi| K_{j'}^\dagger \\ &= \sum_{j,j'=1}^{r} \delta_{jj'} K_j |\psi\rangle\langle\psi| K_{j'}^\dagger = \sum_{j=1}^{r} K_j |\psi\rangle\langle\psi| K_j^\dagger = T(|\psi\rangle\langle\psi|). \end{aligned} \tag{198}$$

Now that we have seen how to use the Stinespring representation of the channel to construct a unitary that acts on the total system plus environment, given the Kraus representation of the channel. Let us see an example of the Stinespring representation for the simple amplitude damping channel.

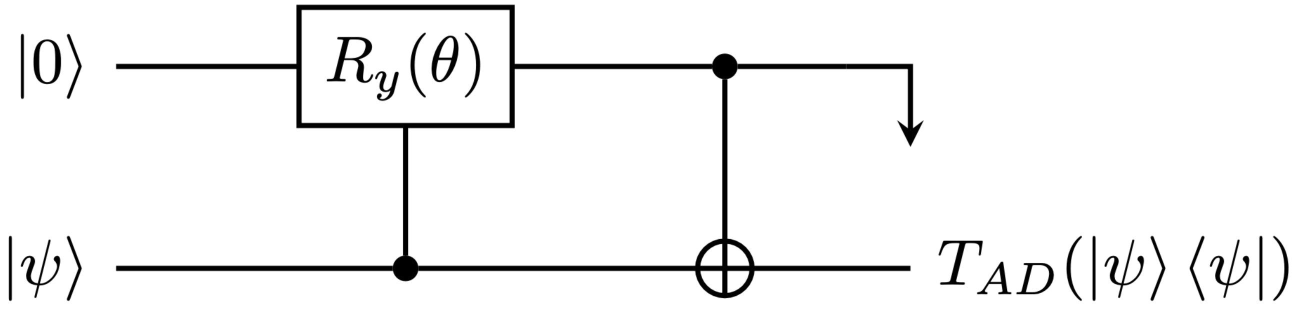


Figure 11: *The quantum circuit that implements the amplitude damping channel T_{AD}, where the first qubit is the environment in the state $|0\rangle$ and the second qubit is the system. We observe that the arrow on the environment refers to the partial trace taken over the environment.*

Example 1. (Amplitude Damping for a single qubit) Let us consider the amplitude damping channel T_{AD}, with Kraus operators,

$$K_0 = |0\rangle\langle 0| + \sqrt{1-p}\,|1\rangle\langle 1|\,, \quad K_1 = \sqrt{p}\,|0\rangle\langle 1| \tag{199}$$

where in this case $|0\rangle = (1,0)^T$ and $|1\rangle = (0,1)^T$. Using the Stinespring representation we get the unitary,

$$U_{AD} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-p} & -\sqrt{p} & 0 \\ 0 & \sqrt{p} & \sqrt{1-p} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{200}$$

for which a circuit can be constructed using the methods in Appendix A, yielding the circuit in Figure 11, with $\theta = 2\arcsin\left(\sqrt{p}\right)$ and $R_y(\theta) = \exp\left(-i\frac{\theta}{2}Y\right)$.

Acknowledgments

This work is based upon research supported by the National Research Foundation of the Republic of South Africa. Support from the National Integrated Cyber Infrastructure System (NICIS) e-research grant QICSA is kindly acknowledged. We would like to thank Ms. S. M. Pillay for her assistance in proofreading the manuscript.

References

[1] S. Lloyd. Universal quantum simulators. *Science* 1996; **273**(5278):1073–1078. doi:10.1126/science.273.5278.1073.

[2] E. Campbell. Random compiler for fast Hamiltonian simulation. *Physical Review Letters* 2019; **123**(7):070503. doi:10.1103/PhysRevLett.123.070503.

[3] A. M. Childs, N. Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information & Computation* 2012; **12**(11-12):901–924. doi:10.26421/qic12.11-12.

[4] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, Y. Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences* 2018; **115**(38):9456–9461. doi:10.1073/pnas.1801723115.

[5] A. M. Childs, A. Ostrander, Y. Su. Faster quantum simulation by randomization. *Quantum* 2019; **3**:182. doi:10.22331/q-2019-09-02-182.

[6] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, S. Zhu. Theory of Trotter error with commutator scaling. *Physical Review X* 2021; **11**(1):011020. doi:10.1103/PhysRevX.11.011020.

[7] D. W. Berry, G. Ahokas, R. Cleve, B. C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics* 2007; **270**(2):359–371. doi:10.1007/s00220-006-0150-x.

[8] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters* 2015; **114**(9):090502. doi:10.1103/PhysRevLett.114.090502.

[9] D. W. Berry, A. M. Childs, R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. in: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. IEEE, 2015. pp. 792–809. doi:10.1109/FOCS.2015.54.

[10] D. W. Berry, A. M. Childs, Y. Su, X. Wang, N. Wiebe. Time-dependent Hamiltonian simulation with L^1-norm scaling. *Quantum* 2020; **4**:254. doi:10.22331/q-2020-04-20-254.

[11] A. Papageorgiou, C. Zhang. On the efficiency of quantum algorithms for Hamiltonian simulation. *Quantum Information Processing* 2012; **11**(2):541–561. doi:10.1007/s11128-011-0263-9.

[12] G. H. Low, I. L. Chuang. Hamiltonian simulation by qubitization. *Quantum* 2019; **3**:163. doi:10.22331/q-2019-07-12-163.

[13] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A* 1990; **146**(6):319–323. doi:10.1016/0375-9601(90)90962-N.

[14] M. Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics* 1991; **32**(2):400–407. doi:10.1063/1.529425.

[15] H.-P. Breuer, F. Petruccione. The Theory of Open Quantum Systems. Oxford University Press, Oxford, 2002. doi:10.1093/acprof:oso/9780199213900.001.0001.

[16] A. Rivas, S. F. Huelga. Open Quantum Systems: An Introduction. SpringerBriefs in Physics. Springer, Heidelberg, 2012. doi:10.1007/978-3-642-23354-8.

[17] I. M. Georgescu, S. Ashhab, F. Nori. Quantum simulation. *Reviews of Modern Physics* 2014; **86**(1):153–185. doi:10.1103/RevModPhys.86.153.

[18] K. L. Brown, W. J. Munro, V. M. Kendon. Using quantum computers for quantum simulation. *Entropy* 2010; **12**(11):2268–2307. doi:10.3390/e12112268.

[19] R. Sweke, I. Sinayskiy, F. Petruccione. Simulation of single-qubit open quantum systems. *Physical Review A* 2014; **90**(2):022331. doi:10.1103/PhysRevA.90.022331.

[20] D. Bacon, A. M. Childs, I. L. Chuang, J. Kempe, D. W. Leung, X. Zhou. Universal simulation of Markovian quantum dynamics. *Physical Review A* 2001; **64**(6):062302. doi:10.1103/PhysRevA.64.062302.

[21] R. Sweke, I. Sinayskiy, D. Bernard, F. Petruccione. Universal simulation of Markovian open quantum systems. *Physical Review A* 2015; **91**(6):062308. doi:10.1103/PhysRevA.91.062308.

[22] A. M. Childs, T. Li. Efficient simulation of sparse Markovian quantum dynamics. *Quantum Information & Computation* 2017; **17**(11-12):901–947. doi:10.26421/qic17.11-12-1.

[23] R. Cleve, C. Wang. Efficient quantum algorithms for simulating Lindblad evolution. in: I. Chatzigiannakis, P. Indyk, F. Kuhn, A. Muscholl (Eds.), 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017). Vol. 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017. p. 17. arXiv:1612.09512. doi:10.4230/LIPIcs.ICALP.2017.17.

[24] Z. Hu, R. Xia, S. Kais. A quantum algorithm for evolving open quantum dynamics on quantum computing devices. *Scientific Reports* 2020; **10**(1):3301. doi:10.1038/s41598-020-60321-x.

[25] A. Gaikwad, Arvind, K. Dorai. Simulating open quantum dynamics on an NMR quantum processor using the Sz.-Nagy dilation algorithm. *Physical Review A* 2022; **106**(2):022424. doi:10.1103/PhysRevA.106.022424.

[26] H. Kamakari, S.-N. Sun, M. Motta, A. J. Minnich. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. *PRX Quantum* 2022; **3**(1):010320. doi:10.1103/PRXQuantum.3.010320.

[27] N. Suri, J. Barreto, S. Hadfield, N. Wiebe, F. Wudarski, J. Marshall. Two-unitary decomposition algorithm and open quantum system simulation. *Quantum* 2023; **7**:1002. doi:10.22331/q-2023-05-15-1002.

[28] G. Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics* 1976; **48**(2):119–130. doi:10.1007/bf01608499.

[29] V. Gorini, A. Kossakowski, E. C. G. Sudarshan. Completely positive dynamical semigroups of *N*-level systems. *Journal of Mathematical Physics* 1976; **17**(5):821–825. doi:10.1063/1.522979.

[30] D. K. Park, I. Sinayskiy, M. Fingerhuth, F. Petruccione, J.-K. K. Rhee. Parallel quantum trajectories via forking for sampling without redundancy. *New Journal of Physics* 2019; **21**(8):083024. doi:10.1088/1367-2630/ab35fb.

[31] K. Kraus. General state changes in quantum theory. *Annals of Physics* 1971; **64**(2):311–335. doi:10.1016/0003-4916(71)90108-4.

[32] M.-D. Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications* 1975; **10**(3):285–290. doi:10.1016/0024-3795(75)90075-0.

[33] A. Jamiołkowski. Linear transformations which preserve trace and positive semidefiniteness of operators. *Reports on Mathematical Physics* 1972; **3**(4):275–278. doi:10.1016/0034-4877(72)90011-0.

[34] M. B. Ruskai, S. Szarek, E. Werner. An analysis of completely-positive trace-preserving maps on $\mathcal{M}_2$. *Linear Algebra and its Applications* 2002; **347**(1):159–187. doi:10.1016/S0024-3795(01)00547-X.

[35] B. C. Hall. Lie groups, Lie algebras, and representations. in: Quantum Theory for Mathematicians. Vol. 267 of *Graduate Texts in Mathematics*. Springer, New York, 2013. pp. 333–366. doi:10.1007/978-1-4614-7116-5_16.

[36] R. Gilmore. Lie Groups, Physics, and Geometry: An Introduction for Physicists, Engineers and Chemists. Cambridge University Press, Cambridge, 2008. doi:10.1017/cbo9780511791390.

[37] F. P. Schuller. Lectures on the Geometric Anatomy of Theoretical Physics. Institute for Theoretical Physics III, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany, 2015. https://www.youtube.com/playlist?list=PLPH7f_7ZlzxTi6kS4vCmv4ZKm9u8g5yic.

[38] J. Agler, W. Helton, S. McCullough, L. Rodman. Positive semidefinite matrices with a given sparsity pattern. *Linear Algebra and its Applications* 1988; **107**:101–149. doi:10.1016/0024-3795(88)90240-6.

[39] W. F. Stinespring. Positive functions on C^*-algebras. *Proceedings of the American Mathematical Society* 1955; **6**(2):211–216. doi:10.2307/2032342.

[40] M. M. Wolf. Quantum Channels and Operations: Guided Tour. Technical University of Munich, Munich, Germany, 2012. https://mediatum.ub.tum.de/node?id=1701036.

[41] R. A. Horn, C. R. Johnson. Topics in Matrix Analysis. Cambridge University Press, Cambridge, 1994. doi:10.1017/cbo9780511840371.

[42] R. A. Horn, C. R. Johnson. Matrix Analysis. 2nd Edition. Cambridge University Press, Cambridge, 2012. doi:10.1017/cbo9781139020411.

[43] G. K. Pedersen. C^*-Algebras and Their Automorphism Groups. 2nd Edition. Academic Press, London, 2018. doi:10.1016/C2016-0-03431-9.

[44] R. Sweke, M. Sanz, I. Sinayskiy, F. Petruccione, E. Solano. Digital quantum simulation of many-body non-Markovian dynamics. *Physical Review A* 2016; **94**(2):022317. doi:10.1103/PhysRevA.94.022317.

[45] M. B. Ruskai. Some open problems in quantum information theory. 2007; arXiv:0708.1902.

[46] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter. Elementary gates for quantum computation. *Physical Review A* 1995; **52**(5):3457–3467. doi:10.1103/PhysRevA.52.3457.

[47] M. A. Nielsen, I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, 2010. doi:10.1017/cbo9780511976667.